

USERS

★★★★★
¡INCLUYE
CONSEJOS PARA
SER UN
EMPRENDEDOR
EXITOSO!

FLASH

CREACIÓN DE SITIOS WEB

ESTRUCTURA: INTERFACES, MENÚ, LAYOUTS,
DIAGRAMACIÓN LÍQUIDA

IMÁGENES: INCORPORACIÓN Y OPTIMIZACIÓN

ANIMACIÓN: CONCEPTOS Y MÉTODOS

APIS Y COMPLEMENTOS: GOOGLE MAPS, YOUTUBE,
PAPERVISION 3D, FACEBOOK

MANUALES USERS MANUALES USERS MANUAL

MA

SITIOS ATRACTIVOS CON ADOBE CREATIVE SUITE

Red**USERS**.com

FLASH

CREACIÓN DE SITIOS WEB





TÍTULO: Flash: creación de sitios Web

COLECCIÓN: Manuales USERS

FORMATO: 17 x 24 cm

PÁGINAS: 320

Copyright © MMXII. Es una publicación de Fox Andina en coedición con DALAGA S.A. Hecho el depósito que marca la ley 11723. Todos los derechos reservados. Esta publicación no puede ser reproducida ni en todo ni en parte, por ningún medio actual o futuro sin el permiso previo y por escrito de Fox Andina S.A. Su infracción está penada por las leyes 11723 y 25446. La editorial no asume responsabilidad alguna por cualquier consecuencia derivada de la fabricación, funcionamiento y/o utilización de los servicios y productos que se describen y/o analizan. Todas las marcas mencionadas en este libro son propiedad exclusiva de sus respectivos dueños. Impreso en Argentina. Libro de edición argentina. Primera impresión realizada en Sevagraf, Costa Rica 5226, Grand Bourg, Malvinas Argentinas, Pcia. de Buenos Aires en III, MMXII.

ISBN 978-987-1857-40-1

Flash: creación de sitios Web / coordinado por Daniel Benchimol.

1a ed. - Buenos Aires: Fox Andina; Dalaga, 2012.

v. 226, 320 p.; 24x17 cm. - (Manual Users)

ISBN 978-987-1857-40-1

1. Informática. I. Daniel Benchimol, coord.
CDD 005.3



ANTES DE COMPRAR

EN NUESTRO SITIO PUEDE OBTENER, DE FORMA GRATUITA, UN CAPÍTULO DE CADA UNO DE LOS LIBROS EN VERSIÓN PDF Y PREVIEW DIGITAL. ADEMÁS, PODRÁ ACCEDER AL SUMARIO COMPLETO, LIBRO DE UN VISTAZO, IMÁGENES AMPLIADAS DE TAPA Y CONTRATAPA Y MATERIAL ADICIONAL.

RedUSERS
COMUNIDAD DE TECNOLOGIA



redusers.com

Nuestros libros incluyen guías visuales, explicaciones paso a paso, recuadros complementarios, ejercicios, glosarios, atajos de teclado y todos los elementos necesarios para asegurar un aprendizaje exitoso y estar conectado con el mundo de la tecnología.



LLEGAMOS A TODO EL MUNDO VÍA  OCA* Y  **

* SÓLO VÁLIDO EN LA REPÚBLICA ARGENTINA // ** VÁLIDO EN TODO EL MUNDO EXCEPTO ARGENTINA

 usershop.redusers.com //  usershop@redusers.com

EL LIBRO DE UN VISTAZO

▼ 01 DISEÑO Y TECNOLOGÍA FLASH

En este capítulo conoceremos las ventajas de Adobe Flash, así como también todo el potencial que encierra su uso en el desarrollo de aplicaciones web. Por otra parte, aprenderemos qué es y para qué sirve ActionScript, y realizaremos las primeras acciones sencillas utilizando estas herramientas.

▼ 02 HERRAMIENTAS DE TRABAJO

Antes de comenzar a realizar nuestros proyectos con Adobe Flash, será necesario conocer las herramientas de trabajo que nos ofrece esta aplicación. En este capítulo analizaremos la interfaz de manejo de Flash y, también, las opciones disponibles en sus diversos paneles.

▼ 03 TEXTOS

Sin dudas, la posibilidad de agregar texto a los proyectos que realicemos con Adobe Flash se cuenta entre las opciones más básicas cada vez que nos disponemos a trabajar en esta aplicación. En este capítulo aprenderemos las diversas formas que podemos utilizar para agregar texto a cualquier proyecto.

▼ 04 DIBUJOS Y LOADERS

La inclusión de dibujos y loaders en nuestros proyectos nos permitirá obtener resultados finales más atractivos. Las formas y otras opciones ofrecidas por Adobe Flash serán comentadas en esta sección, y aprenderemos a usarlas.

▼ 05 ANIMACIONES BÁSICAS

La posibilidad de incluir animaciones es una tarea sencilla gracias a Flash. En este capítulo conoceremos y analizaremos la forma adecuada de utilizar animaciones básicas gracias a la gran diversidad de herramientas incluidas en Adobe Flash.

▼ 06 ANIMACIONES AVANZADAS

Las animaciones se encargan de proporcionar un carácter creativo e interactivo a la Web tal como la conocemos. Flash brinda una serie de opciones avanzadas para crearlas, las cuales explicaremos en profundidad en el presente capítulo.

▼ 07 IMÁGENES Y GALERÍAS

Hablar de imágenes y galerías nos hace recordar los fascinantes efectos proporcionados por Flash que se destacan en diversos sitios web. En este capítulo conoceremos las herramientas adecuadas para lograr la inclusión de imágenes y galerías en nuestros proyectos.

▼ 08 FORMULARIOS

Los formularios son esenciales para otorgar interactividad y comunicación con los visitantes de nuestro sitio. En este capítulo conoceremos las herramientas ofrecidas por Adobe Flash y, también, crearemos un formulario paso a paso.

▼ 09 SONIDO

En este capítulo analizaremos las opciones y diversas herramientas ofrecidas por Adobe Flash relacionadas con el manejo de archivos de audio. De esta forma, la inclusión de sonido en los sitios y aplicaciones web podrá realizarse de manera sencilla y rápida.

▼ 10 VIDEO

El video se presenta como una interesante opción para hacer más atractivos nuestros sitios. En este sentido, Adobe Flash nos entrega una serie de ventajas para que la distribución de este tipo de contenido se efectúe sin complicaciones. En este capítulo aprenderemos cómo hacerlo.

▼ 11 APIs EXTERNAS Y COMPLEMENTOS

Existe una gran cantidad de APIs y complementos encargados de dotar a Flash de funcionalidades interesantes. En este capítulo sabremos qué son y cómo utilizarlos, y además, conoceremos algunos complementos que podremos emplear.

▼ 12 SERVIDORES Y TECNOLOGÍA WEB

Una vez que nuestro proyecto esté terminado, será necesario ubicarlo en un servidor que permita ponerlo a disposición de todo el ciberespacio. En este capítulo analizaremos todos los conceptos importantes que debemos conocer para publicar un sitio en Internet.

▼ 13 VIDA LABORAL

La salida laboral para quienes desarrollan sitios y aplicaciones web utilizando Adobe Flash será mucho más ventajosa luego de leer esta obra. En este capítulo conoceremos los detalles que debemos tener en cuenta a la hora de ofrecer nuestros servicios y tratar con los potenciales clientes.

▼ SERVICIOS AL LECTOR

En esta última sección, encontraremos un índice temático que nos permitirá ubicar los términos más importantes dentro de esta obra.

PRÓLOGO

A medida que Internet creció y se hizo cada vez más popular, los usuarios poco a poco se fueron convirtiendo en expertos navegantes, por lo que comenzaron a demandar una mayor cantidad de soluciones basadas en la Web. Podemos darnos cuenta de esta evolución porque aquellas páginas en las que solo se presentaban textos y algunas animaciones sencillas ya quedaron en el pasado, y dieron lugar a complejos sitios en donde la interactividad cobra una importancia fundamental.

En este sentido, los requerimientos de la nueva generación de internautas construyeron nuevas oportunidades de trabajo que fueron aprovechadas, en un comienzo, por profesionales que tenían estudios relacionados con la programación.

Sin embargo, cuando la apariencia de los sitios web empezó a ganar casi tanta relevancia como su funcionalidad, esos técnicos se quedaron sin respuestas para dar a los clientes, y tuvieron que recurrir a los diseñadores gráficos. Junto a estas nuevas exigencias, fuimos testigos de una completa evolución en las herramientas de desarrollo y de diseño, las cuales crecieron e incorporaron más opciones destinadas a la creación

de soluciones web que integran la posibilidad de crear sitios funcionales y visualmente atractivos.

Esto se logra con un reducido número de aplicaciones, entre las que se destaca Adobe Flash. Además, el crecimiento del ancho de banda y de las velocidades de conexión nos permitió pensar en sitios con amplias funcionalidades, y con diseños sobresalientes y llamativos para hacer frente a los navegantes.

En este nuevo mundo, podemos crecer profesionalmente sin realizar más inversión que la de nuestro capital intelectual. Para eso, es preciso adquirir los conocimientos técnicos que se convertirán en un negocio rentable cuando les sumemos una cuota de nuestra propia creatividad y buenas ideas.

En este sentido, la obra que tienen entre sus manos permite adquirir todos los conocimientos necesarios para dominar la tarea de crear sitios y soluciones web exitosas usando Adobe Flash. Comencemos, entonces, el recorrido hacia el éxito en el diseño y el desarrollo de sitios profesionales. ¡Suerte!

CONTENIDO

El libro de un vistazo	004
Prólogo	006
El contenido de la obra	007
Introducción	010

▼ 01 DISEÑO Y TECNOLOGÍA FLASH 011

Flash y ActionScript	012
¿Para qué usamos ActionScript?	014
¿Por qué trabajar con AS 3.0?	018
Pensar y diseñar un sitio en Flash	020
Interfaces en Flash	022
Organizar un proyecto	026
Layouts en Flash	028
Diagramación líquida	030
Menú de navegación	036
Carga externa de películas	038
Actividades	040

▼ 02 HERRAMIENTAS DE TRABAJO 041

Adobe Cs5 Web Premium®	042
La interfaz de Adobe Flash CS5	044
Herramientas	046
Illustrator y Flash	048
Migrar un botón a Flash	050
Actividades	052

▼ 03 TEXTOS 053

Texto clásico en Flash	054
Suavizado tipográfico	057
Crear texto clásico	058
Text Layout Framework	060
TLF: flujo entre textos	062
Texto en formato HTML	064
Aplicar CSS a un campo de texto dinámico	066
Crear una scroll bar con easing	068
Actividades	072

▼ 04 DIBUJOS Y LOADERS 073

El dibujo de formas	074
Dibujo en Flash	076
La herramienta Decó	078
Entornos 3D	080
Intros en Flash	082
Las alternativas a las intros	084
Cargadores para sitios en Flash	086
Creación de un cargador simple	088
Optimización de sitios	090
Actividades	092

▼ 05 ANIMACIONES BÁSICAS 093

Axiomas de animación	094
Classic Tween	096
Classic Tween y guías de movimiento	098

Motion Tween	100
Motion Tween y Motion Preset	102
Motion Editor	104
Animaciones con Motion Editor	106
Cinemática inversa	108
Cómo animar huesos con muelles	110
Actividades	112

▼ 06 ANIMACIONES AVANZADAS 113

Efectos físicos	114
Librerías de animación	116
TweenMax y TweenLite	118
Cómo aplicar efecto a una imagen	120
Sistema de partículas	122
Cómo animar el layout de nuestro sitio	126
Actividades	130

▼ 07 IMÁGENES Y GALERÍAS 131

Incorporación de imágenes	132
Optimización de imágenes	134
Filtros dinámicos	136
Bitmap y BitmapData	138
Galería de fotos	140
Componentes	142
Crear un slideshow dinámico	144
Actividades	148

▼ 08 FORMULARIOS 149

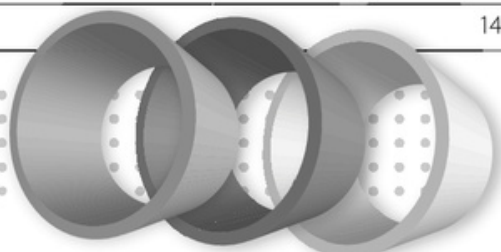
Formularios en Flash	150
Cómo crear un formulario	152
Tabulación y validación de formularios	156
Texto predictivo	158
Envío de formularios	160
Content Management System	162
Gestores de contenido	164
Flashblocks	166
Actividades	168

▼ 09 SONIDO 169

Uso de sonido	170
Sincronización de sonido	172
Sonido y ActionScript 3	174
Librerías de sonido	176
Cómo crear una visualización de sonido	178
Adobe CS5 Soundbooth	180
Cómo editar con Soundbooth	182
Sitios experimentales de sonido	184
Actividades	186

▼ 10 VIDEO 187

Video en la Web	188
Formatos de video	190
Adobe Media Encoder	192
Incorporar video a nuestras películas	194
Video streaming	196



Sitios experimentales con video	198
Adobe Flash Media Server	200
Loop de video en el fondo	202
Actividades	206

▼ 11 APIs EXTERNAS Y COMPLEMENTOS 207

¿Qué son las APIs?	208
YouTube API	210
Google Maps	212
Cómo cargar coordenadas	214
Papervision 3D	216
Facebook API	218
Adobe CS5 Flash Catalyst	220
Utilizar Flash Catalyst	222
SWFObject	224
Personalizar menú contextual	226
Cómo proteger Flash	228
Agregar elementos interactivos	230
Archivos temporales	234
Cómo aplicar efectos de partículas	236
Google Analytics para Flash	238
Actividades	240

▼ 12 SERVIDORES Y TECNOLOGÍA WEB 241

Clientes y servidores	242
Protocolos	246
DNS: sistema de nombres de dominios	247
URL y tipos de dominios	248
El sistema web y su funcionamiento	249
Elegir un nombre de dominio	250

¿Cómo registrar un dominio?	252
Servicios de hosting	256
Subir nuestra web a Internet	258
Subir un sitio al servidor	260
Actividades	262

▼ 13 VIDA LABORAL 263

La salida laboral	264
Workflow de trabajo	268
¿Emprendedor, empleado o freelance?	270
Freelance: ¿cómo iniciarse?	272
Administrar el tiempo	274
Conseguir clientes online	276
Emprendedor. ¿Yo?	280
Planificar un portfolio efectivo	282
Oportunidad en las ideas	284
Determinar el precio del trabajo	286
Modelo de negocio	288
Funcionamiento de una empresa	292
Planificar presentaciones efectivas	294
Las actitudes nocivas del diseñador	296
Gestión financiera	298
Negociaciones efectivas	300
Clientes manipuladores	304
Generar relaciones contractuales	306
Primer año del emprendimiento	308
Actividades	310

▼ SERVICIOS AL LECTOR 311

Índice temático	312
Catálogo	315

INTRODUCCIÓN

La era digital generó nuevas oportunidades de trabajo; en este sentido, esta obra se presenta como la puerta de acceso a una de las áreas que más posibilidades y satisfacciones brindan en el ámbito web: la del diseño y el desarrollo de sitios. Aquí se exponen los fundamentos de este arte y se enseñan todos los secretos para que triunfemos ante cualquier desafío que nos toque encarar, convirtiéndonos en excelentes profesionales de la Web.

A través de los capítulos de este libro, podremos conocer y dominar una de las principales tecnologías que se emplean para el diseño y el desarrollo de sitios web: Flash. En sus páginas realizaremos un recorrido por todos los elementos teóricos fundamentales para efectuar esta tarea y veremos la manera de llevar estos conceptos a la práctica. Este aprendizaje sentará bases sólidas, y nos llevará a diseñar y desarrollar sitios web profesionales muy superiores a los que podría realizar un diseñador no especializado, distinguiendo y haciendo resaltar nuestro trabajo en todo momento.

Los usuarios que se beneficiarán del contenido de esta obra son todos aquellos que quieran diseñar sitios; incluso, quienes no cuentan con educación formal previa sobre la materia. Los diseñadores gráficos aumentarán la cantidad de servicios que pueden ofrecer, y se convertirán en desarrolladores web con conocimientos avanzados de Adobe Flash, la herramienta que permite crear sitios de manera integral en su propio lenguaje de programación (ActionScript 3.0). Los aficionados a la Web podrán participar en ella de modo activo mediante el desarrollo de sus propios sitios, y serán capaces de diseñarlos también para terceros, con lo cual transformarán el objeto de su devoción en algo rentable y productivo.

En suma, nos encontramos frente a una obra imprescindible para todos aquellos que deseen sacar el máximo provecho de la creación de sus sitios para la Web. Sin dudas, luego de leer estas páginas y desarrollar los ejercicios propuestos, los lectores se convertirán en expertos en el diseño y el desarrollo de sitios web. ¡Buena suerte!



Diseño y tecnología Flash

- ▶ FLASH Y ACTIONSCRIPT
- ▶ ¿PARA QUÉ USAMOS ACTIONSCRIPT?
- ▶ ¿POR QUÉ TRABAJAR CON AS 3.0?
- ▶ PENSAR Y DISEÑAR UN SITIO EN FLASH
- ▶ INTERFACES EN FLASH
- ▶ ORGANIZAR UN PROYECTO
- ▶ LAYOUTS EN FLASH
- ▶ DIAGRAMACIÓN LÍQUIDA
- ▶ MENÚ DE NAVEGACIÓN
- ▶ CARGA EXTERNA DE PELÍCULAS



FLASH Y ACTION SCRIPT

Hoy en día, Flash y ActionScript son herramientas indispensables para crear sitios web con un alto impacto visual, donde se busque generar una experiencia en el usuario novedosa y altamente interactiva.

Flash apareció en 1997 como una herramienta de la compañía Macromedia, orientada a la creación de animaciones para el soporte web. En su versión 3, adoptó el lenguaje ActionScript, que, con un paquete básico de acciones, permitía generar cambios en la **linealidad** de la reproducción de las animaciones e **interacciones** mínimas con el usuario. A diferencia de HTML, Flash incorporó el manejo de **objetos vectoriales**, con los cuales fue posible dotar de una mayor riqueza a los gráficos e imágenes que empezaron a verse en la Web, al mismo tiempo que se lograba optimizar el peso de esos elementos. Estos dos aspectos no solo favorecieron cuestiones meramente estéticas de los sitios web, sino que también contribuyeron a mejorar el diseño y la comunicación.

EL ELEGIDO POR LOS DISEÑADORES

Flash se convirtió rápidamente en la herramienta preferida por los diseñadores al momento de crear contenidos para la Web. Por un lado, este fenómeno se debe a la gran similitud que tienen sus herramientas con aplicaciones como Illustrator o Photoshop, lo que simplifica la tarea de dar forma a contenidos visuales desde la interfaz gráfica del programa. Por otro lado, podemos decir que ActionScript es un lenguaje de programación muy potente y con infinitas posibilidades, que no requiere tener un manejo avanzado para aprovechar todo su potencial.

ACTIONSCRIPT 1.0

Macromedia creó Flash 3 y sumó un factor desconocido en el mundo de las aplicaciones para diseño: la programación. En este caso, se trataba de un primitivo paquete de **acciones**, como **stop**,

play y **gotoAndStop**. Posteriormente, con Flash 4 se incorporaron estructuras de programación y un paquete cada vez mayor de acciones, ciclos y condicionales, que permitían solo la interacción con **movie clips**. A partir de Flash 5, el lenguaje adoptó la especificación de sintaxis del estándar **ECMA-262**, al igual que el lenguaje JavaScript (de ahí sus notables similitudes). En esta versión, la programación de los botones, por ejemplo, debía agregarse sobre los elementos mismos, una labor bastante complicada que hacía prácticamente imposible la unificación en la programación de acciones.

ACTIONSCRIPT 2.0

Esta versión fue incluida a partir de Flash 6, conocido como Flash MX 2004. Aquí se produjo un hito en la generación de interactividad, ya que se incorporó una gran

cantidad de métodos y propiedades que permitían el control del video, el entorno, el sistema y el monitor; la carga dinámica de imágenes y el formateo de textos en forma dinámica; además de la posibilidad de tener un entorno cada vez más orientado a objetos. Un cambio notable en esta versión es que el código de todo el sitio puede centralizarse en un solo fotograma, lo cual ordena todos los scripts componentes del sitio y facilita la programación en elementos reiterativos.

ACTIONSCRIPT 3.0

Luego de la compra de Macromedia por parte de Adobe, apareció Flash 9, conocido como **Flash CS3**, y con él, la nueva versión de ActionScript, la 3.0. Además de ejecutarse hasta diez veces más rápido que las versiones anteriores, esta introdujo un nuevo esquema de



listeners: antes no estaba unificado el criterio o la lógica con que se programaba, por ejemplo, un evento de botón y uno de escenario. Ahora, en esta nueva evolución del lenguaje, el criterio es el mismo, hecho que simplifica el razonamiento al momento de programar.

¿CÓMO SERÁN LAS VERSIONES FUTURAS?

Es difícil saber a ciencia cierta cómo serán las versiones futuras de ActionScript, aunque podemos adivinar que, tal como sucedió en la 3.0 con los tipos de datos **int** y **uint**, se implementarán más tipos de datos y estructuras de programación, para hacer más veloz la ejecución de las líneas de código. Por otro lado, podemos observar que existen cada vez más **librerías open source**, como **tween lite**, **tween max** y **papervision 3d**, que pueden añadirse a Flash para brindar más y mejores funcionalidades respecto de las que ofrece el programa en forma nativa. En el futuro, estas funcionalidades ahora externas podrían estar incluidas en el core de ActionScript, y entonces funcionarían aún mejor de lo que lo hacen actualmente de manera externa.

GRANDES CAMBIOS

Mientras Flash pertenecía a la empresa Macromedia, aunque existió un gran salto y muchas mejoras entre las versiones 5 y 6, las posteriores entraron en una meseta de la que solo pudo salirse cuando Adobe compró a Macromedia con todo su paquete de aplicaciones.



En www.pinocchioisonfire.org encontramos una interfaz multimedia desarrollada con Flash.



Este es un juego online (www.56sagestreet.co.uk) con muchos gráficos e interacción.



En www.tweetsforhonor.com vemos un juego que relaciona Flash con tecnologías actuales.

En ese momento surgió Flash CS3, más compatible y más atento a la evolución de una Internet donde existen múltiples plataformas. Hoy en día, la Web apunta a ser una plataforma de comunicación multimedia, donde conviven cada vez en mayor medida el video en vivo, el sonido y la interacción del usuario, que genera y modifica los contenidos existentes online con un hardware cada vez más potente y con más prestaciones.

Es posible esperar que Flash y ActionScript nos ofrezcan, en el futuro, un mayor control de los distintos medios; mayor interacción con tecnologías como HTML, PHP, MySQL; y, como siempre, la posibilidad de implementar estas herramientas desde una interfaz visual que, desde el principio, fue una de sus mayores fortalezas. ◀



¿PARA QUÉ USAMOS ACTIONSCRIPT?

ActionScript aporta todo lo relativo a la interactividad entre los contenidos que hayamos creado y el usuario. Además, permite generar una gran variedad de efectos visuales muy atractivos y coloridos.

Disponer de conexiones a Internet cada vez más veloces hace que podamos disfrutar de contenidos no solo textuales, sino también videos y música. Flash y ActionScript le brindan al desarrollador una plataforma desde la que es posible generar una gran interactividad con el usuario y proveerle una enorme cantidad de información, que se traduce en videos interactivos, galerías de imágenes y sonidos, por dar algunos ejemplos.

¿QUÉ ES ACTIONSCRIPT?

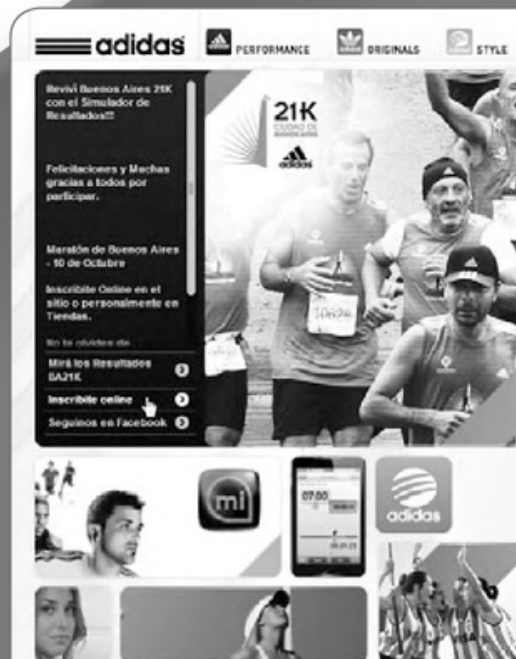
ActionScript es un lenguaje de programación **orientado a objetos** que funciona exclusivamente en Flash. En programación, cuando hablamos de objetos, nos referimos a la **instancia de una clase**. De manera genérica, el movie clip es una clase, y una vez que creamos uno en el escenario, este es una nueva **instancia** de la clase **[MovieClip]**. Todos los objetos constan de tres partes fundamentales: propiedades, métodos y eventos. Llamamos **propiedades** a las **características** de un objeto, los **métodos** son las **capaci-**

+ CDS Y DVDS INTERACTIVOS

En muchos casos, los CDs y DVDs interactivos se utilizan para complementar algún tipo de información primaria. Por ejemplo, pueden acompañar libros y electrodomésticos, o ser la presentación de algún tipo de servicio o producto. En esas oportunidades, el uso de un soporte digital que se lee en el equipo del usuario permite potenciar las prestaciones de Flash con contenidos de mucha más calidad y en mayores tamaños, ya que no será necesario descargarlos desde un servidor online. Esta modalidad ofrece una mayor velocidad de ejecución que no requiere aplicaciones extra, dado que el archivo que se reproduce incluye el player de Flash.

dades que tiene ese objeto de hacer algo, y los **eventos** son avisos en respuesta a sucesos determinados. Tomando un objeto del tipo MovieClip como ejemplo, podemos decir que algunas de sus propiedades son la altura, el ancho y la ubicación respecto de los ejes X e Y. Algunos de sus métodos son la capacidad de detener o reproducir su propia línea de tiempo (**[stop();]** y **[play();]**) o de ir a cierto fotograma y reproducir la línea de tiempo o dejarla detenida (**[gotoAndPlay();]** y **[gotoAndStop();]**). En el caso de los eventos, podemos nombrar a los

FLASH CUENTA CON UNA ENORME CANTIDAD DE OBJETOS, Y UNA IGUAL CANTIDAD DE PROPIEDADES, MÉTODOS Y EVENTOS.



Adidas (www.adidas.com/ar) muestra menús desplegables de manera creativa para ver la información.

avisos que el movie clip envía cuando el usuario hace clic sobre él o cuando el puntero del mouse se posa sobre su superficie (**[CLICK]** y **[ROLL_OVER]**, respectivamente). La librería de Flash cuenta en la actualidad con una enorme cantidad de objetos, y una igual cantidad de propiedades, métodos y eventos. Esto la convierte en una herramienta sumamente poderosa y versátil, que permite manipular todos los elementos

LAB

07

DRAG

||

Vemos en <http://lab.mathieu-badimon.com> algunas de las características más llamativas de ActionScript 3.0.

www.metropolis-media.com.pl muestra una gran cantidad de animaciones de todo tipo: permanentes y del usuario.

||



multimedia que existan en un proyecto tanto como sea necesario.

SITIOS DE PUBLICIDAD Y PROMOCIÓN

Este tipo de sitios apunta a persuadir al usuario generando una atracción y un interés en navegar y descubrir el contenido, vinculándose de manera más cercana al producto o al servicio que se esté dando a conocer. Cuando desarrollamos un sitio o una aplicación íntegramente en Flash, debemos aprovechar la capacidad que nos da el programa para crear distintos tipos

de **animaciones y efectos visuales** de manera sencilla por medio de su interfaz gráfica. Además, es muy útil la posibilidad de incorporar elementos visuales que hayamos diseñado en otros programas, como Illustrator y Photoshop, con los que tiene una excelente compatibilidad y versátiles **opciones** de importación. Los efectos dan forma a las distintas transiciones, entradas y cargas de contenidos de un sitio, a la navegación de las secciones y a la manera en la que van apareciendo los distintos tipos de información. Desde ActionScript, podemos controlar en qué momento sucede cada

animación, cuál será su duración y qué información se verá cuando estos movimientos terminen. Otros ejemplos de aplicación son los botones animados y los menús desplegables, que generan efectos llamativos y altamente funcionales al momento de delimitar de modo visible las áreas interactivas.

PORTFOLIOS, SITIOS CON CARGA EXTERNA DE CONTENIDOS

En este tipo de sitios, el eje fundamental del trabajo no pasa tanto por el impacto. En cambio, se trata de proveerle al usuario de una navegación rápida y sin tiempos de espera excesivos al recorrerlos o al acceder a imágenes o videos. En estos casos, podemos hacer que las secciones del sitio y los contenidos de gran tamaño sean archivos independientes, fuera del archivo principal, que se vayan cargando solo cuando el usuario quiere acceder a la información que está en ellos. Esto suele hacerse cuando cada sección es muy pesada y queremos evitar grandes demoras en la visualización. ActionScript no solo permite cargar esos contenidos cuando el usuario los solicite, sino que también da la posibilidad de mostrar en pantalla el progreso de la carga mediante una o varias animaciones. ▶





En www.dontclick.it encontramos una interfaz que explota al máximo las capacidades de Flash CS5.

En www.ecodazoo.com podemos hacer un recorrido virtual de un zoológico en un entorno 3D totalmente interactivo.

En www.chromazone-imaging.co.uk vemos un sitio con navegación mediante el desplazamiento del entorno.

SITIOS EXPERIMENTALES

En los sitios experimentales se busca impactar no solo desde lo visual sino también desde lo interactivo. Esto se logra explotando las capacidades de Flash, cuyo gran fuerte es la generación de **interacción**, además de la animación. Cuando hablamos de interacción, lo primero que nos viene a la mente es un botón sobre el cual el usuario hace clic para acceder a una determinada información. Este caso es solo uno de los muchos tipos de interacción que se pueden generar desde Flash para con el usuario, debido a que hay diversas alternativas posibles. Por ejemplo, obteniendo la posición del mouse, se podrían ir mostrando distintas partes del escenario; accediendo a la cámara web del usuario, podrían cambiarse elementos en la pantalla según sus movimientos; o utilizando el micrófono del equipo, sería posible registrar lo que el usuario dijera o la intensidad con la que hablara para hacer que los contenidos tuvieran distintos comportamientos o formas de visualización. La gran incorporación de Flash a partir de la versión CS4 fueron las herramientas de manipulación en **3D** de los elementos del escenario. Esto, sumado al agregado de algunas librerías externas, como **Papervision**, hizo de este un recurso muy potente para la creación de gráficos y de entornos interactivos en 3D.

+ JUEGOS

Como mencionamos, Flash es una poderosa herramienta para crear entornos interactivos de alta calidad e impacto gráfico y animaciones. En la actualidad, las comunidades de desarrolladores de Flash (foros, blogs, y otros) dan la posibilidad de descargar, en forma gratuita, muchos **motores** (la programación base de interactividad, animaciones, etcétera) para armar juegos clásicos, que pueden adaptarse al diseño que haga falta. Esto, sumado a la simplicidad de uso, hace posible que cada vez existan más entretenimientos realizados totalmente en Flash y ActionScript, tanto clásicos como creados de manera particular para campañas de difusión o publicitarias.

SITIOS Y APLICACIONES PARA DISPOSITIVOS MÓVILES

Aunque no todos los dispositivos móviles, como celulares o PDAs, son compatibles con Adobe Flash, el número de ellos que soporta esta tecnología es cada vez más elevado. Con sus pantallas táctiles, estos equipos brindan un soporte ideal para el tipo de interacción que genera Flash y para la reproducción de contenidos visuales, como imágenes, videos y sonidos, con el agregado de interactividad. Los desarrollos más comunes para estas plataformas son los juegos, entre los que es frecuente encontrar algunos de los que se usaban en los viejos **arcade**. También es posible hallar versiones resumidas de los últimos títulos para consolas como PlayStation y Wii.

E-LEARNING

Las altas velocidades de descarga que proporciona la mayoría de las conexiones a Internet hacen posible bajar contenidos cada vez más completos y útiles. Por esta razón, muchas empresas que brindan **capacitación** ponen a disposición de los usuarios diversos videos y aplicaciones interactivas destinados a realizar cursos a distancia. Estos contenidos creados con Flash permiten aprovechar al máximo la manipulación de elementos multimedia y la generación de un entorno interactivo que facilite el aprendizaje.

APLICACIONES PARA FACEBOOK

La gran cantidad de usuarios y la enorme difusión que tiene hoy en día la red social Facebook hacen que este sea un lugar privilegiado para dar a conocer y promocionar distintas marcas de productos y servicios. Para lograr este fin, se generan las llamadas campañas virales, en las que, por ejemplo, es posible jugar a un juego que, luego, podemos recomendar a nuestros amigos dentro de la red. En general, la cantidad de amigos que incorporemos al juego puede traernos algún tipo de beneficio en el puntaje. Tanto el entretenimiento como, por supuesto, la interacción con los datos de la red social son facilitados y manipulados por Flash. También podemos encontrar aplicaciones creadas en Flash para simplificar la búsqueda de paquetes turísticos, propiedades o bienes de consumo. ◀

Liquid Light (www.liquidlight.co.uk) presenta un juego para mezclar bebidas, como vemos en la imagen.

En www.letstakepictures.info, cada imagen se descarga cuando el usuario quiere verla. Esto es una notable ventaja.

¿POR QUÉ TRABAJAR CON AS 3.0?

ActionScript 3.0 surgió con la versión CS3 de Flash y lleva varios años en el mercado. Conoceremos sus características principales y analizaremos sus ventajas.

La evolución de Flash no implica, simplemente, un cambio, sino que representa una gran mejora en la reproducción de los archivos. Esta versión incorporó elementos en el lenguaje de programación, como tipos de datos nuevos y objetos que permiten una manipulación más completa de elementos, intervalos de tiempo y solicitudes al servidor. Estas novedades permiten que el **player** de



ERRORES

Una característica muy buena de AS 3, aunque parezca lo contrario, es la cantidad de errores que pueden aparecer al momento de publicar la película. Básicamente, estos nos alertan sobre situaciones que, en las versiones anteriores, pasaban inadvertidas pero que, de todas formas, hacían que nuestro desarrollo no funcionara. Por ejemplo, podemos ser advertidos por tener dos nombres de instancia o dos nombres de funciones iguales, por querer manipular un elemento que no existe o que no está en el lugar especificado, o por pasarle un número incorrecto de argumentos a una función.

FLAMPlayer

Demo version with playlist



Demo version without playlist



Many thanks to m4tv (Laurent B.) for demo tracks.

Flash (el plugin instalado en el navegador del usuario que interpreta los archivos de Flash) ejecute toda la aplicación de manera más rápida, gracias a la optimización de los recursos.

Como ejemplo, vale mencionar que, en las versiones anteriores, para lograr que el sitio se adaptara a la resolución de pantalla (técnica denominada **diagramación líquida**, como veremos más adelante), había que realizar el cálculo de la ubicación de un elemento. El resultado era un número, dato del tipo **[Number]**, con varios decimales, que Flash tenía que calcular, aunque en la mayoría de los casos hubiera bastado con obtener el entero, despreciando las otras cifras.

En ActionScript 3.0, con la incorporación del tipo de datos **[int]**, podemos operar

con números enteros sin considerar los decimales, y haciendo que Flash Player realice solo las operaciones necesarias, con lo cual se mejora el rendimiento.

UNIFICACIÓN DE CRITERIOS DE PROGRAMACIÓN

Quienes trabajaban con las versiones 1.0 y 2.0 de ActionScript estaban habituados a hacer cada tarea a su modo, ya que era posible programar de diferentes maneras. Como ejemplo, podemos mencionar tres casos en los que, en apariencia, podríamos haber usado el mismo criterio pero con una forma de programar totalmente distinta entre ellos:

- ▶ Los botones se programaban usando el evento **[onRelease]** (cuando el usuario hace clic).
- ▶ El **preloader** de un sitio web se programaba usando el evento **[enterFrame]** (que también puede usarse para hacer animaciones) y realizando un cálculo porcentual de propiedades que se obtenían mediante los métodos **[getBytesLoaded]** y **[getBytesTotal]**.

Desde www.flamplayer.com podemos descargar reproductores online de MP3 para insertar en nuestra página.



Aquí vemos una galería de videos (www.bestiario.org) generada en un ambiente 3D creado con las herramientas de Flash.

► Si queríamos cargar una imagen externa (o un archivo SWF) y mostrar a la vez el progreso de carga de ese archivo, podríamos pensar que el procedimiento era similar al de mostrar la precarga del sitio. Sin embargo, era necesario crear **listeners** que escucharan el evento **[onProgress]**, vincularlos a la clase **[MovieClipLoader]** y, recién entonces, efectuar el cálculo porcentual.

Como vemos, en las versiones anteriores existían tres maneras muy distintas de vincularnos con eventos (**[onRelease]**, **[onEnterFrame]** y **[onProgress]**), y de programar acciones que podríamos pensar que se hacían del mismo modo, ya que no había una forma intuitiva de saber cuál de ellas había que utilizar. Al contrario de lo que sucedía en las versiones 1 y 2, la 3.0 ofrece un solo modo de programar esas tres situaciones (y, en realidad, todas las que sucedan en un desarrollo con AS 3). Para hacerlo, debemos entender la estructura orientada a los eventos, que podremos trasladar para vincularnos con eventos de las clases **[Tween]** (para animaciones), **[MovieClip]** (para usarlos como botón), para manipular eventos de video o de la clase **[Loader]** (para realizar cargas externas). Ahora, todo lo que vayamos a programar que, de una forma u otra, tenga relación con un evento, se hará de la misma manera. Esto, sin dudas, es un gran beneficio tanto en el momento del aprendizaje como en el de la aplicación.

CAPACIDADES PROPIAS DE AS 3

Fuera de las técnicas de programación unificadas o de la mayor velocidad de procesamiento que tiene esta nueva versión, AS 3 posee una librería de funcionalidades y efectos mucho más grande que en las versiones anteriores. A continuación, veremos una serie de tareas que no podían hacerse en las versiones predecesoras:

- **Sonido:** el objeto de sonido posee nuevas capacidades y complementos. Es posible, por ejemplo, acceder al espectro de sonido y generar una gráfica, al igual que lo hacen los reproductores de MP3, como Winamp, Windows Media Player o iTunes.
- **Entorno 3D:** nuevas propiedades como **[rotationX]**, **[rotationY]** y **[rotationZ]**, pertenecientes a los elementos visuales, como el movie clip, permiten manipularlo y modificarlo en un entorno de tres dimensiones, con lo cual se generan situaciones visuales con elementos en profundidad.

SI NO ESTABLECEMOS
NOMBRES DE INSTANCIA,
FLASH ASIGNA UNO
AUTOMÁTICAMENTE A CADA
ELEMENTO QUE NO LO TENGA.

► **XML nativo:** si bien las versiones anteriores permitían usar XML, en esta nueva, la navegación de estos archivos se ha simplificado notablemente. Ahora, por ejemplo, podemos acceder a las etiquetas directamente por su nombre, ordenándolas en un listado.

► **Nombres de instancia:** antes, para vincularnos desde ActionScript con un elemento que estaba en el escenario, era imprescindible que este tuviese un nombre de instancia. En líneas generales, podemos decir que todo debía tener nombre de instancia, tanto lo que agregábamos al escenario en forma manual (importándolo o dibujándolo con las herramientas) como lo que incorporábamos desde la librería usando ActionScript. En esta forma de trabajo, debíamos ser cuidadosos en la asignación de nombres, que siempre debían ser distintos para que todo funcionara correctamente. Ahora, si no establecemos un nombre nosotros, Flash le asigna uno automáticamente a cada elemento que no lo tenga. De hecho, no es necesario que el desarrollador sepa cuál es ese nombre, porque, gracias a los nuevos métodos **[getChild]** y **[getChildAt]**, podemos vincularnos a cualquier elemento sabiendo, simplemente, dónde está ubicado. ◀

PENSAR Y DISEÑAR UN SITIO EN FLASH

Flash nos ofrece una amplia gama de instrumentos de diseño y desarrollo para crear un sitio web. Aprendamos a coordinarlos adecuadamente para brindarles a nuestros usuarios una sorprendente experiencia multimedia interactiva.

La Web no es más que otro soporte a través del cual comunicar un mensaje a un público determinado. Una de las particularidades que tiene este medio es que puede combinar diferentes tipos de contenido en un mismo lugar: texto, imagen, sonido y video. Esta característica **multimedia** es lo que lo hace uno de los medios de consulta e información más utilizados en la actualidad.

¿SIRVE FLASH PARA DISEÑAR NUESTRO SITIO WEB?

Una de las primeras preguntas que tendríamos que plantearnos para evaluar si debemos diseñar nuestro sitio con Flash es si la información que queremos comunicar a nuestro público objetivo abarca distintos tipos de formato (texto, imágenes, audio y video). No tendría mucho sentido desarrollar un sitio en Flash si lo único que tenemos para transmitir está en un formato escrito, por ejemplo, un blog de noticias. En este caso, lo más adecuado sería utilizar HTML y CSS. Flash se vuelve indispensable cuando un sitio necesita brindar al usuario una experiencia multimedia completa, ya que no solo nos permite combinar texto, imágenes, audio y video, sino que, además, da la posibilidad de interactuar con otros **periféricos** que posea el usuario, como **micrófono** o **webcam**.

Además, tiene la ventaja de que es considerado multiplataforma: esto significa que un sitio o aplicación desarrollado con Flash será visto correctamente por cualquiera que tenga instalado el plugin, independientemente del sistema operativo y del navegador web que posea (salvo los dispositivos con iOS, como el iPhone o el iPad).

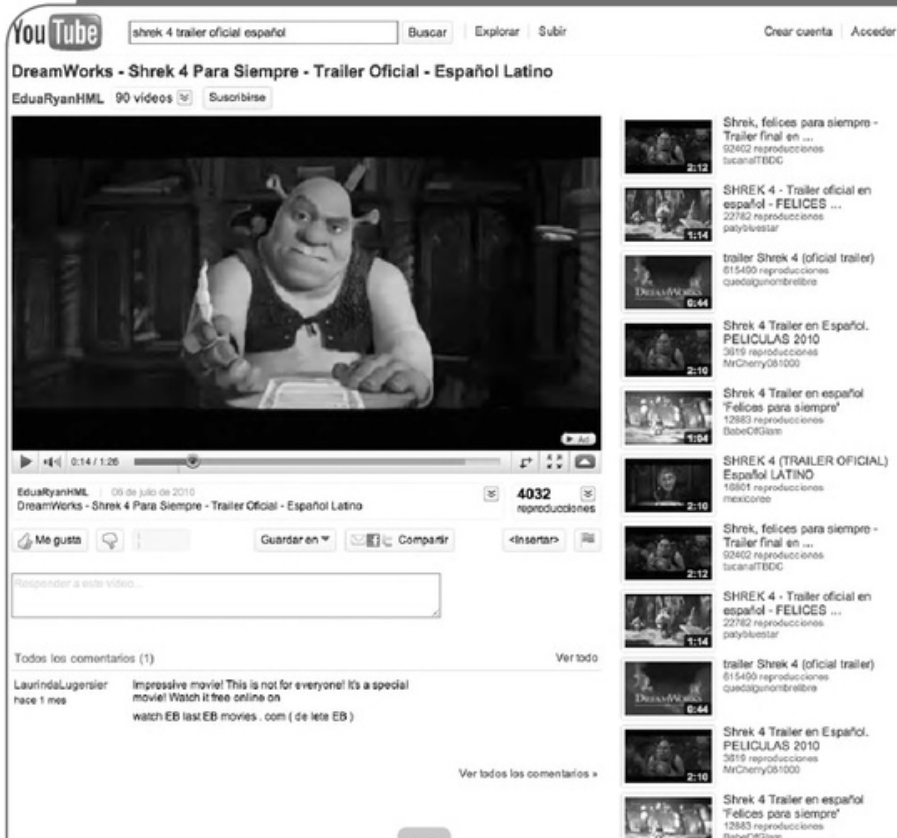
CONOCER AL VISITANTE

En general, quien visita un sitio en Flash está dispuesto a invertir bastante tiempo navegándolo, ya que espera encontrarse con información en distintos formatos (textual y audiovisual). Además, estos usuarios están bien predispuestos a investigar nuevas alternativas de navegación, y a explorar interfaces u objetos interactivos que lo sorprendan. Incluso, gracias a que Flash actualmente permite hacer sitios o aplicaciones para disfrutar a través de pantallas táctiles, el usuario podrá interactuar con el contenido usando sus dedos en vez del mouse. Aunque las personas esperan encontrar bastante contenido multimedia, muchas veces los archivos SWF generados por Flash pueden pesar varios miles de Kilobytes, lo que ocasiona demoras significa-

tivas en el momento de descargarlos. En estos casos, una buena práctica consiste en utilizar métodos de cargas parciales o proponerle al usuario alguna distracción en pantalla mientras espera. Por otro lado, debido a la alta aceptación que están teniendo los teléfonos inteligentes (**smartphones**) entre los usuarios de Internet, debemos preguntarnos si los visitantes accederán a nuestro sitio en mayor medida a través de este tipo de dispositivos o lo harán desde sus computadoras. Muchos smartphones no cuentan en la actualidad con el plugin de Flash en su navegador, y corremos el riesgo de que, entonces, esas personas no puedan ver nuestro sitio web.

El sitio www.virtuaflowers.com brinda al usuario una excelente experiencia multimedia para regalar flores.





YouTube (www.youtube.com), rey de la difusión de videos en Internet, utiliza Flash para sus reproductores.

PROS Y CONTRAS DE UN SITIO DISEÑADO EN FLASH

Entre las **ventajas** de los sitios creados en Flash, podemos mencionar las siguientes:

► **Compatibilidad entre plataformas:**

lo que diseñemos en Flash se verá correctamente, sin importar el navegador o el sistema operativo que el usuario utilice.

► **Gráficos de vector, animaciones, efectos visuales y fuentes especiales:** el potencial estético que brinda Flash permite que dejemos volar nuestra creatividad casi sin límites, con un resultado visualmente asombroso.

► **Video:** podemos integrarlo a nuestro sitio o hacer **streaming** (descargarlo a medida que se ve). En ambos casos, el usuario no necesita contar con ningún reproductor adicional en su sistema operativo (como QuickTime o MediaPlayer).

► **Juegos:** es posible desarrollar juegos muy divertidos en Flash, sin necesidad de instalar plugins adicionales que tienen los desarrollos en Java, AJAX o CGI.

El teléfono iPhone ha agregado increíbles mejoras en su versión 4. Sin embargo, no incorpora Flash.

► **Accesibilidad:** en la actualidad, Flash cuenta con muchas opciones que lo hacen accesible a personas con capacidades distintas, como el agregado de **subtítulos (Closed Caption)** a un video o el uso de **audiodescripción**.

Entre las **desventajas**, encontramos las que listamos a continuación:

► **Problema con motores de búsqueda:** si es fundamental para el éxito de nuestro sitio lograr un buen posicionamiento, debemos tener en cuenta que a los buscadores les resulta muy difícil posicionar los sitios Flash en sus bases de datos.

► **Para usar Flash se requiere un plugin:** algunas personas y compañías no permiten el plugin de Flash por decisiones de seguridad o para evitar tener que actualizarlo en miles de terminales cada vez que sea necesario. ◀

SI QUEREMOS COMUNICAR MUCHA INFORMACIÓN MULTIMEDIA, FLASH ES IDEAL PARA CREAR EL SITIO WEB QUE LA DIVULGUE.



REALIDADES DE FLASH PLAYER

Como nos informa Adobe en su sitio (www.adobe.com), el 98% de las computadoras conectadas a Internet tiene Flash Player. También lo utiliza el 85% del top 100 de sitios web, 75% del video distribuido en Internet y 70% de los juegos de la Web. Además, 3,5 millones de desarrolladores apoyan esta plataforma, y 19 de las 20 productoras más grandes de dispositivos móviles están interesadas en incorporar Flash Player en sus dispositivos.

INTERFACES EN FLASH

Los sitios y la forma en que vemos Internet van cambiando. Ese cambio es percibido por los aspectos gráficos, algo realmente notorio en los desarrollos con Flash.

Antes de comenzar a hacer nuestro análisis sobre los diferentes tipos de interfaces que se pueden lograr con un sitio desarrollado en Flash y ActionScript, definiremos a qué se denomina interfaz. Una **interfaz web** es el conjunto de elementos de un sitio que permiten realizar acciones al usuario que lo navega. Los contenidos, la navegación, los elementos de identificación y las acciones que podemos generar dentro de un sitio son parte de su interfaz. Lo más importante es la **relación** que se crea entre un sitio web y sus usuarios por medio de las interfaces. Muchos especialistas en la materia señalan que hay cuatro postulados que debemos contemplar al momento de pensar una interfaz web:

- **Dejar claro el propósito del sitio**, explicando a quién pertenece y qué permite hacer a quienes lo visitan.
- **Ayudar a los usuarios a encontrar lo que buscan**, disponiendo de una navegación intuitiva y con elementos bien visibles, que permita un rápido acceso a la información ofrecida.
- **Demostrar el contenido del sitio**, teniendo en cuenta que su disposición debe ser clara, sin muchos enlaces a material que tenga poca importancia. Hay que destacar las secciones o la información más relevante, presentando el contenido de manera jerarquizada.
- **Usar el diseño para mejorar y no para definir la interacción del sitio**. Los elementos gráficos no deben tener una finalidad meramente decorativa, sino que, por medio del diseño y sus elementos, deben crearse áreas de interés y atención que faciliten la navegación y la comprensión del contenido.

LAS INTERFACES, FLASH Y ACTIONSCRIPT

Desarrollar un proyecto web en Flash y ActionScript nos da la posibilidad de incursionar en diferentes opciones al momento de pensar y crear su interfaz de navegación. Con total libertad y casi sin límites, podemos lograr interesantes, atractivas y motivadoras formas de navegación e interacción con el usuario. Esto hace que cada sitio en Flash sea distinto de los otros, gracias a la creatividad con la que fue pensado. En las próximas páginas, haremos un recorrido por las más diversas e interesantes interfaces que se pueden crear en Flash, para lo cual las analizaremos según sus características.

INTERFACES TRADICIONALES

Cuando nos referimos a interfaces tradicionales, hablamos de una estructura de navegación en la que se brinda a los usuarios un modo de recorrer el sitio al que están acostumbrados, por ser el que comúnmente encuentran en sitios de todo tipo, ya sean desarrollados en Flash o en HTML. En ellos, por lo general hay un menú de navegación en forma de barra o lista que es fácilmente reconocible, y que suele ubicarse en el sector superior o en el lateral izquierdo. También se presenta el nombre del sitio o el logo en la zona superior, los accesos secundarios en el footer o zona inferior, y el contenido principal en el centro. Si bien podemos hallar una gran variedad de diseños de este tipo, tanto





De forma simple, el sitio Dvein (www.dvein.com) nos permite acceder al contenido rápidamente.

En www.narrowdesign.com vemos una interfaz que nos provee de un acceso rápido al contenido.



minimalistas como recargados de detalles, efectos y zonas de interactividad, estas estructuras son bien reconocibles y rápidas de utilizar. Una característica que tiene este tipo de interfaces es que, al ser tan habitual, intentan llamar la atención de los usuarios por otros medios, para diferenciar un sitio del resto.



En <http://awesome.good.is/ecosystem>, los contenidos están presentados por medio de una interfaz muy clásica.

Immersive Garden (www.immersive-garden.com) es un claro ejemplo de interfaz de navegación tradicional.

LAS INTERFACES SON IMPORTANTES MÁS ALLÁ DEL ÁMBITO DE INTERNET, Y EXISTE UN ÁREA DE ESTUDIO DEDICADA A ELLAS.

INTERFACES NO TRADICIONALES

Si nos alejamos un poco de lo clásico, encontramos sitios con interfaces en las cuales los elementos comunes a todos no están ubicados o distribuidos en donde esperamos que estén. En estos casos, los usuarios se muestran interesados en entender cómo navegar y recorrer el sitio en busca del contenido, incitados por un buen diseño e interactividad. Podemos ver un ejemplo de esto en el sitio de **Nick Jones** (www.narrowdesign.com), que carece de un menú de navegación para recorrer las diferentes secciones porque, en realidad, no tiene secciones internas. En cambio, tenemos un menú vertical con slide por medio del cual podemos ver cada uno de los trabajos realizados en forma intuitiva y no tradicional. ▶



En www.neostream.com el menú es presentado por un pequeño personaje. Es una opción interesante para nuestro sitio.

Pensar una interfaz no tradicional no implica olvidarse de la usabilidad y la accesibilidad del sitio, pero tampoco requiere que los elementos estén donde deben estar. Lo importante es que su funcionamiento pueda ser reconocido y entendido, sin caer en excesos de creatividad que resulten contraproducentes y provoquen la huida de los usuarios por no comprenderlo. Por ejemplo, cuando visitamos el sitio **PolarBearPlunge** (<http://polarbearplunge.sandiegozoo.org>), encontramos una estética y unas animaciones muy buenas y, seguramente, crearemos que existe una barra de navegación

en la parte superior. Sin embargo, el verdadero menú de navegación está escondido en una pequeña solapa en la parte inferior del sitio, junto a los accesos secundarios. Podemos alejarnos aún más de lo tradicional en sitios que nos ofrezcan una navegación diferente, donde, por ejemplo, se ponga más énfasis en la interacción y en la posibilidad de tener una experiencia entretenida. Por ejemplo, en el caso de **Neostream** (www.neostream.com), la navegación se realiza por medio de un simpático personaje que, al mismo tiempo, es la interfaz o menú de navegación principal.

INTERFACES 3D

Las interfaces 3D no necesariamente salen de lo tradicional, porque pueden ser entornos simples creados en tres dimensiones, ya sea en tiempo real o prediseñados. Sin embargo, haremos un análisis de este tipo de navegación, en donde podemos navegar el contenido no solo en forma vertical u horizontal, sino que, en este caso, también la **profundidad** juega un papel muy importante. Así, podemos encontrarnos con sitios cuyos menús de navegación están uno detrás del otro, en vez de estar uno junto al otro. Veamos algunos ejemplos:

► **EcoMagination** (<http://ge.ecomagination.com/smartgrid>): con una buena interfaz y la implementación de contenidos prediseñados en 3D, el sitio ofrece una navegación sencilla, y el contenido es visualizado correctamente. A pesar de tener el menú de navegación oculto en la zona inferior, es fácil reconocerlo y acceder a él. Además, hay una voz en off que nos da algunas indicaciones e información.

► **Ecodazoo** (www.ecodazoo.com): en este gran ejemplo, el contenido está completamente creado en tiempo de ejecución por medio de ActionScript. No solo se recurre al 3D, sino que también se



General Electric presenta su trabajo mediante el uso de 3D en <http://ge.ecomagination.com/smartgrid>.

Seguramente, nos llevará unos segundos encontrar el menú de navegación de <http://polarbearplunge.sandiegozoo.org>.



FLASH Y ACTIONSCRIPT NOS BRINDAN MUCHAS POSIBILIDADES PARA ATRAER Y COMUNICARNOS CON LOS USUARIOS DE NUESTROS SITIOS.

genera un recorrido no lineal para visualizar el material que el sitio provee.

INTERFACES Y DISPOSITIVOS

Desde hace un tiempo, podemos encontrarnos con sitios que nos permiten interactuar con su interfaz a través de dispositivos como la **webcam**, el **teclado** o el **micrófono** de la computadora. Estos nos brindan una nueva experiencia de usuario pero, en muchos casos, no son

sencillos de comprender, a menos que se indique expresamente el comportamiento esperado por parte del visitante. Un ejemplo interesante de este tipo se presenta en el sitio del estudio **JPEG Interactive (www.jpeg.cn)**, que podemos rotar según la posición de nuestra cabeza frente al monitor, detectada por medio de la webcam. Otro gran ejemplo está en **www.devozasuaimaginacao.com.br**, que utiliza el sonido por medio de ActionScript para que podamos pintar utilizando la voz.

INTERFACES EXPERIMENTALES

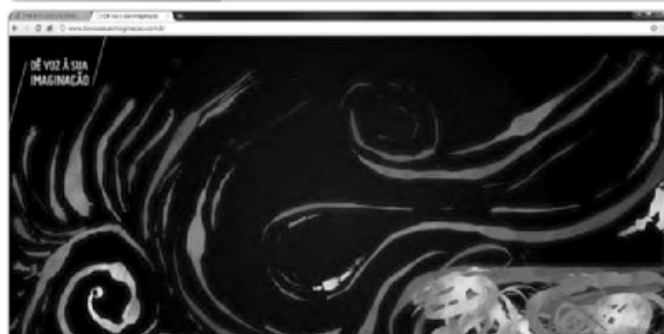
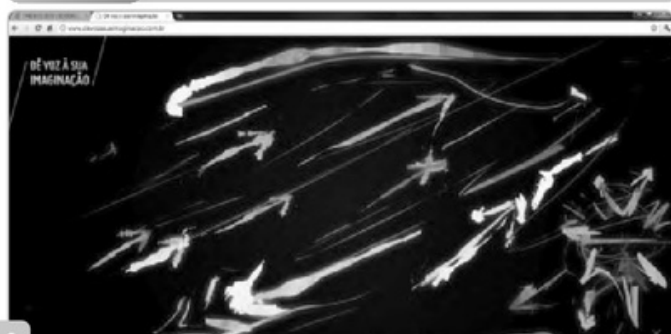
En el extremo contrario al de las interfaces tradicionales, están los sitios en los que es común encontrar cosas poco vistas y donde la disposición de la información quizá no respeta los convencionalismos sobre la manera en la que esta debe ser

presentada al usuario. Si bien esto podría generar inconvenientes en la usabilidad, por lo general, el usuario que visita estos sitios está bien definido y predispuesto, y dispone de tiempo e interés para develar cómo es la navegación. Por ejemplo, podemos recorrer el sitio **www.selftitled.ca** de modo muy práctico y minimalista, ya que mediante el círculo central, accedemos al gran contenido que se ofrece.

Como vimos, podemos recurrir no solo a lo tradicional y convencional para crear interfaces. Flash y ActionScript nos brindan un panorama muy amplio de posibilidades para crear la mejor manera de comunicar y atraer a los usuarios que visiten nuestros sitios. Es cuestión de poner a trabajar nuestra creatividad para lograr resultados que, tal vez, creíamos imposibles pero que, ahora, sabemos que no lo son. ◀



Este sitio (www.snowdin.com) nos permite ver su diseño en forma diferente si utilizamos los anteojos rojo y azul para ver en 3D.



Dê voz à sua imaginação (www.devozasuaimaginacao.com.br) es una excelente demostración del uso del sonido en Flash.

ORGANIZAR UN PROYECTO

Esta es una forma de organizar y estructurar los archivos en las distintas carpetas, en un proyecto en Flash.



LIVE

Carpeta principal del proyecto.

01

LIVE ARCHIVO INDEX.HTML

HTML donde montaremos el SWF principal del sitio por medio de la librería JavaScript Swfobject.js. Según los requerimientos, también podría ser un archivo PHP.

02

LIVE ARCHIVO SWF PRINCIPAL

SWF principal del sitio. Generalmente denominado index.swf o main.swf.

03

AS ARCHIVOS AS

Archivos AS con las funciones, librerías y clases necesarias para el funcionamiento del sitio.

04

SWF ARCHIVOS SWF

Archivos de películas SWF complementarias que cargaremos dentro del sitio.

05

FLA ARCHIVOS FLA

Estos son los archivos fuente del sitio (esta carpeta no debe ser alojada en el servidor al subir el sitio).

06

PHP ARCHIVOS PHP

Archivos PHP con las funciones utilizadas para el envío de mails y demás requerimientos, tales como logging de usuarios.



XML

07

XML
ARCHIVOS XML

Archivos XML con información de configuración y contenidos.

JS

08

JS
ARCHIVOS JS

Archivos JavaScript utilizados en el proyecto.

CSS

09

CSS
ARCHIVOS CSS

Archivos de hojas de estilo CSS utilizados en el HTML o bien para decorar textos HTML dentro de las películas.

MP3

10

SOUND
ARCHIVOS DE SONIDO

Archivos de sonido MP3 u otros formatos que pueden ser utilizados en el sitio y son cargados dinámicamente.

FLV

11

VIDEO
ARCHIVOS DE VIDEO

Archivos de video FLV u otros formatos que pueden ser utilizados en el sitio y son cargados en forma dinámica.

JPG

12

IMG
ARCHIVOS DE IMAGEN

Archivos de imagen que pueden ser cargados en forma dinámica.

LAYOUTS EN FLASH

Gracias a su interfaz gráfica, y al manejo de imágenes y elementos vectoriales, Flash es una potente herramienta para crear diseños interactivos. Analicemos las interfaces que podemos generar.

Como sabemos, los layouts son importantes en el aprendizaje de HTML, por esta razón en esta sección veremos sus propiedades enfocándonos en el diseño de interfaces de los desarrollos realizados en Flash.

NAVEGACIÓN DE LAYOUTS Y ESTRUCTURAS TRADICIONALES

Decimos que un sitio web tiene un **layout** y una **estructura de navegación tradicional** cuando posee una barra horizontal o vertical con varios botones que conforman un menú desde el cual se accede a las distintas secciones y contenidos. Estas estructuras posibilitan una **navegación no lineal**, ya que el usuario tiene acceso a cualquier parte del sitio sin importar dónde se encuentre.

Esto es de suma importancia, porque debemos ofrecerle al navegante la manera más ágil de encontrar los contenidos que está buscando. Por un lado, tenemos la **estructura de navegación** que el diseñador interactivo determine como la mejor para ese sitio en particular. Por otro lado, encontramos el diseño conceptual que se haga sobre

LAS ESTRUCTURAS DE TIPO TRADICIONAL NO TIENEN POR QUÉ SER SINÓNIMO DE SITIOS CON POCO DISEÑO, SERIOS O ABURRIDOS.

esa interfaz. En este punto, cobra gran relevancia la versatilidad de Flash para manipular sonidos, videos y elementos gráficos de distintos tipos y naturaleza; vectores y filtros generados en la misma aplicación; o imágenes desarrolladas, manipuladas y modificadas en programas como Illustrator y Photoshop. En todos los casos, debemos centrarnos en la experiencia del usuario, en hacer que este se sienta cómodo y rápidamente familiarizado con la manera en la que le presentamos los contenidos y el modo en que puede acceder a ellos. A la vez, desde el diseño, debemos transmitir alguna idea que se asocie con el concepto comunicado por la empresa que ofrece sus productos o servicios a través del sitio web. Como ya establecimos cuando trabajamos con sitios HTML, es sumamente importante que, al momento de diseñar, conozcamos cuáles serán los fines del sitio, y cuál es la información más relevante y de más interés para el usuario. De este modo, podremos ofrecer distintos puntos de acceso (a veces, varios hacia la misma información) y puntos de acceso jerarquizados, según la relevancia del material.

CREACIÓN DE LAYOUTS Y ESTRUCTURAS NO TRADICIONALES

En estos casos, los sitios presentan algo más que botones para ser navegados. Se trata de sitios que invitan al usuario a sumergirse en una experiencia de navegación distinta, en donde los contenidos se van descubriendo de modo intuitivo. Esto, por supuesto, no es recomendable para todos los sitios que tengamos que diseñar, ya que, en ciertos proyec-



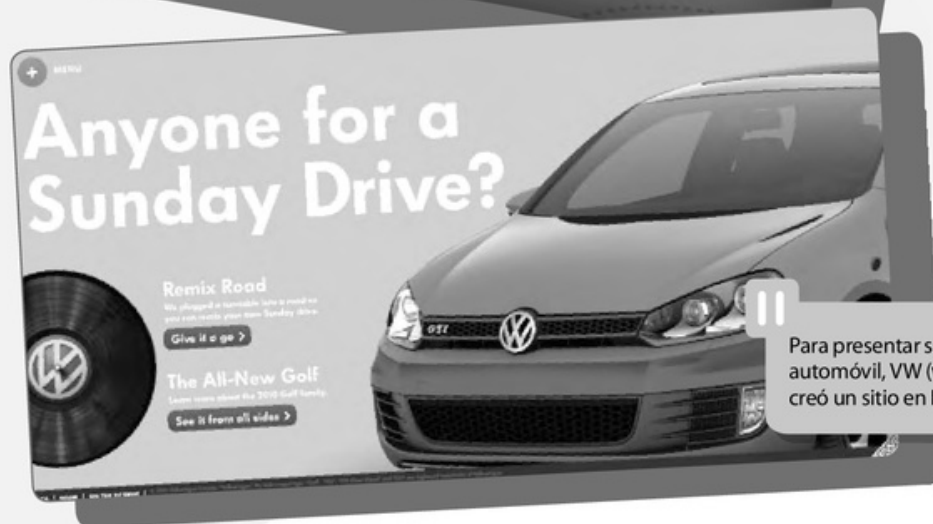
En este caso (www.gettheglass.com), una campaña contada por medio de un juego cuyos personajes son los de una familia muy particular.



En www.descubriendo.org podemos observar una estructura tradicional pero altamente semantizada.



En www.goomundo.com, la semantización de la estructura usa elementos como postales, mapas y brújulas.



Para presentar su nuevo modelo de automóvil, VW (www.vwsundaydrive.ca) creó un sitio en Flash y un video interactivo.

tos, el usuario espera encontrarse con determinados elementos de navegación que le son familiares, como los menús, los botones o las barras de scroll. Este tipo de sitios suele usarse para la difusión de productos específicos de alguna marca conocida, y en portfolios de artistas, diseñadores o desarrolladores web. La estrategia principal es lograr un sitio altamente llamativo que, a la vez, refleje las características que la empresa quiere destacar, que muestre la forma en la que se trabaja o las capacidades técnicas para resolver casos similares.

Flash ofrece características que ayudan mucho en el diseño de interfaces de este tipo, donde necesitamos generar distintas clases de jerarquías de información partiendo de estructuras no tradicionales de diseño. Es entonces cuando es posible recurrir a la animación de algún elemento para que este se destaque por sobre otro que no tiene movimiento, o utilizar sonidos. Flash trabaja especialmente bien con cualquier secuencia de sonido en MP3 (aunque no es el único formato que soporta), por lo que es posible generar jerarquías de información desde

elementos con sonido y sin sonido, o con alguno en particular. Con el advenimiento de tecnologías cada vez más avanzadas y más rápidas para conectarse a Internet, se hace posible tomar como recurso corriente la inclusión en forma externa de videos interactivos, creados con herramientas 3D, o de edición y manipulación de video. A esto podemos sumarle la interactividad que brinda Flash para lograr un notable resultado. ◀

DIAGRAMACIÓN LÍQUIDA

La diagramación líquida en Flash permite que nuestro diseño se adapte al formato y la resolución de todas las pantallas, ajustando y reubicando los objetos para que usen toda el área disponible.

Dada la diversidad de formatos y tamaños de pantalla existentes hoy en día, podemos afirmar que es indispensable hacer que nuestro sitio no solo se vea en ellas, sino que también aproveche todo el tamaño disponible que estas tengan, evitando que ocupe solo el 50% del área disponible.

A pesar de esto, algunos sitios y diseños pueden funcionar muy bien con solo quedar centrados, dando la ilusión de parecer más grandes. En un principio, se diseñaba todo el sitio y se lo programaba con una resolución de pantalla estándar (en general, 1024 x 768). Ante la eventual aparición de un usuario que manejara una pantalla con una resolución mayor, el sitio aparecía centrado en ella. Sin embargo, esta metodología no es adecuada si alguien intenta ver el sitio en un aparato con una resolución menor, por ejemplo, en un dispositivo móvil o en una netbook,

o si se utiliza un equipo con un monitor de 24 pulgadas. Escalar todos los contenidos sería un procedimiento incorrecto, ya que nuestro sitio se vería pixelado en un monitor muy grande o, por el contrario, no permitiría leer las tipografías demasiado pequeñas en un dispositivo móvil.

USAR DIAGRAMACIÓN LÍQUIDA

Sin necesidad de escalar contenidos, mediante la **diagramación líquida** podemos disponer de los elementos del sitio para ofrecerle al usuario contenidos de calidad, personalizados para su equipo y aprovechando el máximo de sus capacidades.

Cuando utilizamos diagramación líquida, hacemos, básicamente, dos cosas. Algunos elementos serán reubicados en la pantalla y otros serán escalados, con la posibilidad de variar, en este último punto, la proporción de la escala. Todo esto será decisión del desarrollador del sitio, quien no dejará ningún detalle librado al azar o a los comportamientos predefinidos de Flash.

¿CÓMO LO HACEMOS?

En primer lugar, tenemos que configurar algunas propiedades de **[stage]**. Con la propiedad **[scaleMode]**, le especificamos a Flash que no escale nada por defecto, ya que, como dijimos antes, los



En el sitio que encontramos en www.lamaldita.net, los elementos se redimensionan y acomodan para ocupar todo el espacio disponible de la pantalla.

cambios de tamaño se harán de acuerdo con lo que nosotros definamos como desarrolladores. También tendremos que configurar la **alineación** del documento respecto de la ventana del reproductor o navegador a través de la propiedad **[align]**, de modo que se ubique en el ángulo superior izquierdo. Para todo esto, en el panel **[Acciones]** tendríamos que escribir lo siguiente:

```
stage.scaleMode =
StageScaleMode.NO_SCALE;
stage.align = StageAlign.
TOP_LEFT;
```

El objeto **[stage]** es el escenario más toda la zona circundante, es decir, todo lo que está fuera de él. Luego debemos suscribirnos al evento **[RESIZE]** disparado por el objeto **[stage]**. Este evento ocurre cada vez que la ventana del navegador es redimensionada. La función que escucha el evento **[RESIZE]** se ocupará de obtener las medidas disponibles del objeto **[stage]** y de guardarlas en variables previamente creadas. Utilizando estas variables, realizaremos los cálculos requeridos (en general, sumas y restas) para ubicar los elementos como menús, imágenes, pies de página y todo lo que sea necesario, con el fin de darle al usuario una situación de máximo aprovechamiento de las capacidades de su equipo. Debemos escribir el **código 1** en el panel **[Acciones]**. Por ejemplo, en el caso del pie de página (cuyo nombre de instancia podría ser **[footer]** y estar instanciado en el escenario), debemos hacer que mida todo el ancho disponible de la pantalla y que siempre se mantenga ubicado en su parte inferior. Entonces, modificamos la propiedad **[width]** del movie clip **[footer]** que contiene al pie de página y cambiamos su posición respecto del eje **[y]**, variando la propiedad **[y]** para darle un valor igual al alto del espacio disponible menos la altura del movie clip **[footer]**. Así haremos que su extremo inferior se ubique en el extremo de la pantalla. Cabe aclarar que para que este cálculo sea correcto, el punto de registro de **[footer]** debe estar en su ubicación predefinida, que es el ángulo superior izquierdo. En el panel **[Acciones]**, la función **[diagramacionLiquida]** queda como vemos en el **código 2**. ◀

CÓDIGO 1

```
var ancho:Number;
var alto:Number;
stage.addEventListener(Event.RESIZE,
diagramacionLiquida);

function diagramacionLiquida(e:Event) {
    ancho = stage.stageWidth;
    alto = stage.stageHeight;
}
```

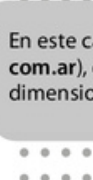
CÓDIGO 2

```
function diagramacionLiquida(e:Event) {
    ancho = stage.stageWidth;
    alto = stage.stageHeight;

    footer.width = ancho;
    footer.y = alto - footer.height;
}
```



Este diseño de Ford (www.ford.com.ar) no tiene diagramación líquida, sino que se mantiene centrado.



En este caso (www.parquedelacosta.com.ar), el sitio desaprovecha las dimensiones de la pantalla del usuario.



DIAGRAMACIÓN LÍQUIDA

PASO A PASO

En las próximas páginas veremos cómo implementar diagramación líquida en nuestro sitio de forma sencilla.

Configuración del documento

Dimensiones: 1024 px (anchura) x 768 px (altura)

☒ Ajustar ángulo de perspectiva 3D para conservar proyección actual del escenario

Unidades de regla: Píxeles

Colores:

☒ Predeterminado

☐ Contenido

☐ Impresora

Color de fondo:

Velocidad de fotogramas: 24,00

Transformar en predeterminado

01 | TAMAÑO DOCUMENTO

Lo importante a la hora de realizar una diagramación líquida es tratar de trabajar con la mayor centralización posible para que los resultados sean los esperados, aprovechando al máximo y de una manera inteligente el espacio visual de cada pantalla. ►

PASO | 01 En primer lugar, creamos un documento nuevo y vamos a **[Modificar/Documento...]**, o pulsamos **<Ctrl + J>**. No es necesario que este tenga una medida determinada, porque al trabajar con diagramación líquida, el tamaño se verá afectado cuando el documento sea embebido en el HTML. Sin embargo, para hacerlo con comodidad, es recomendable que tenga al menos 1024 x 768, ya que de esa forma sabremos cómo se verá el sitio en sus dimensiones más estándar.



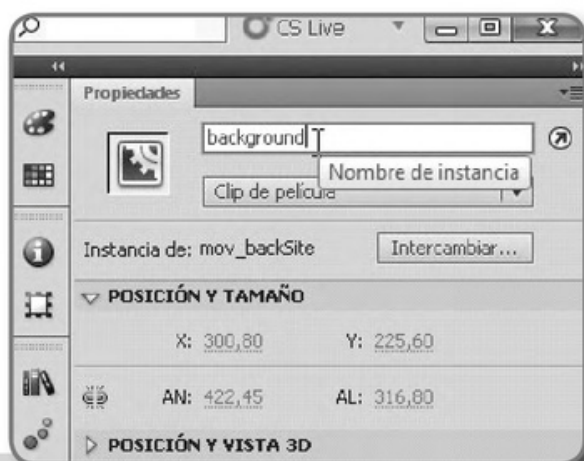
AGREGAR ELEMENTOS

02

PASO | 02 Una vez preparado el documento, lo guardamos y comenzamos con la creación y el agregado al escenario de todos los objetos que vamos a incorporar, según el diseño que hayamos planificado.

PASO | 03 Para referirnos a cada uno de estos objetos, tenemos que instanciarlos. Es importante que el nombre de instancia tenga alguna relación con el contenido del objeto. Por ejemplo, en nuestro caso serán **[footer]**, **[header]**, **[background]**, etcétera. El nombre de instancia puede establecerse desde el panel **[Propiedades]**, que podemos habilitar yendo al menú **[Ventana/Propiedades]** o presionando la combinación de teclas **<Ctrl + F3>**.

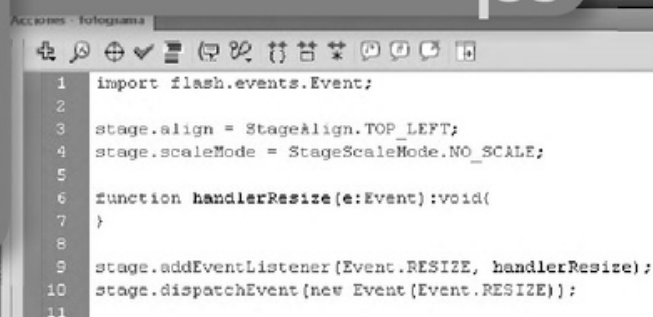
PASO | 04 Para que todo funcione correctamente, los movie clips que creamos deben estar alineados con **x = 0** e **y = 0**. Para verificarlo, entramos en cada uno, seleccionamos el objeto que contiene y comprobamos, en



04 | ALINEAR OBJETOS



ALINEAR SIN ESCALAR | 05



DETECCIÓN DE CAMBIOS | 06

el panel **[Propiedades]**, que el valor de los ejes **[x]** e **[y]** sea **[0]**. Si el elemento no está alineado, hacemos clic en el valor actual del eje y ponemos el valor correcto. Otra forma de alinear objetos es mediante el panel **[Alinear]**, que podemos habilitar yendo a **[Ventana/Alinear]** o presionando la combinación **<Ctrl + K>**. Luego, seleccionamos el objeto que vamos a alinear y presionamos en las opciones de alineación con los bordes superior e izquierdo del escenario.

PASO | 05 Con todos los objetos instanciados y en el escenario, presionamos **<F9>** y, en el panel **[Acciones]**, comenzamos a agregar el código necesario para que funcione la diagramación líquida. En primer lugar, debemos indicarle al player de Flash que alinee el escenario con el ángulo superior izquierdo y que de ninguna forma el contenido se escale. De esta manera, frente a cualquier variación en el tamaño del SWF, el escenario mantendrá su ubicación y tamaño.

Las primeras dos líneas para lograr esto son las siguientes:

```
stage.align = StageAlign.TOP_LEFT;
stage.scaleMode = StageScaleMode.NO_SCALE;
```

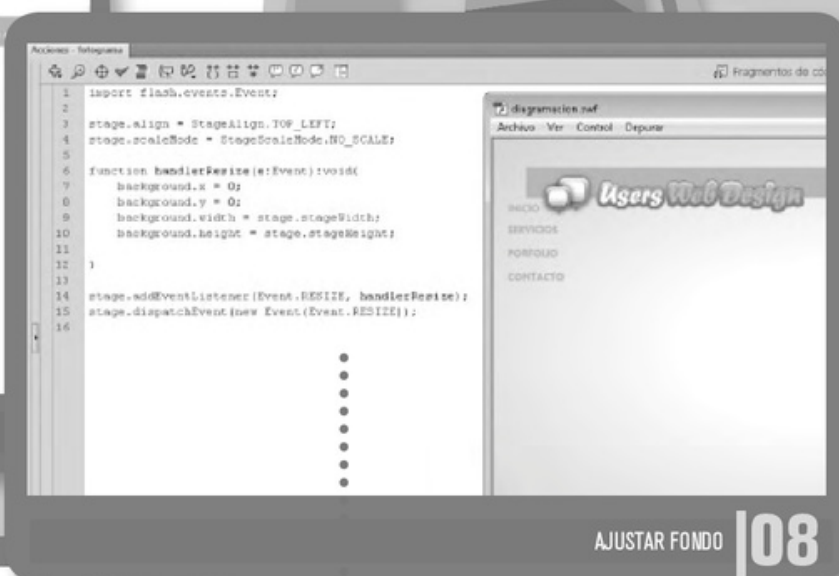
PASO | 06 Como necesitamos tener control total de la posición de los objetos y de su tamaño, vamos a introducir un evento que nos indique lo que está sucediendo en tiempo real con el escenario. Este **listener** detecta si en algún momento cambia el tamaño del escenario y, si esto ocurre, automáticamente invoca a la función **[handlerResize]**.

DIAGRAMACIÓN LÍQUIDA

PASO A PASO



07 | PROBAR LA PELÍCULA



AJUSTAR FONDO **08**



09 | ALINEAR HEADER

PASO | 07 Exportamos la película haciendo **<Ctrl + Enter>** o yendo a **[Control/Probar película/Probar]** para verificar su funcionamiento. Hasta aquí, nuestro código es un bucle que detecta la variación en el tamaño del escenario y ejecuta una función hasta que este permanezca fijo. Todo el código que escribamos dentro de la función **[handlerResize]** se actualizará cada vez que exista una variación y, para esos casos, escribiremos el código que se ocupará de alinear y modificar el tamaño de los movie clips.

PASO | 08 Para modificar los movie clips del escenario cada vez que se ejecute la función **[handlerResize]**, usamos las propiedades **[stage.stageWidth]** y **[stage.stageHeight]**, que nos permitirán saber el alto y el ancho del escenario. Para tomar el background (previamente instanciado con el nombre **[background]**) y alinearlo al punto cero de los ejes

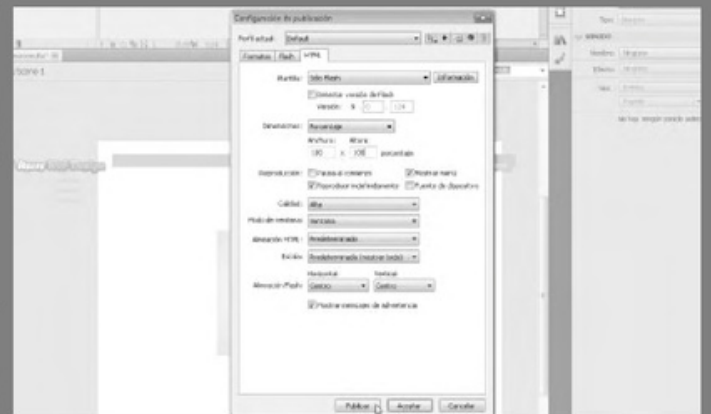
X e Y, aplicamos el alto y el ancho del escenario al alto y ancho del movie clip. Con el código que vemos en la imagen, el **[background]** cubrirá todo el escenario.

PASO | 09 Con el fin de alinear el **[header]**, usamos el código que aparece a continuación. En este caso, no solo estamos alineando o cambiando la morfología de los objetos ubicados en el escenario, sino que también estamos trabajando con los objetos que están dentro del movie clip, como **[back]** y **[menuHeader]**.

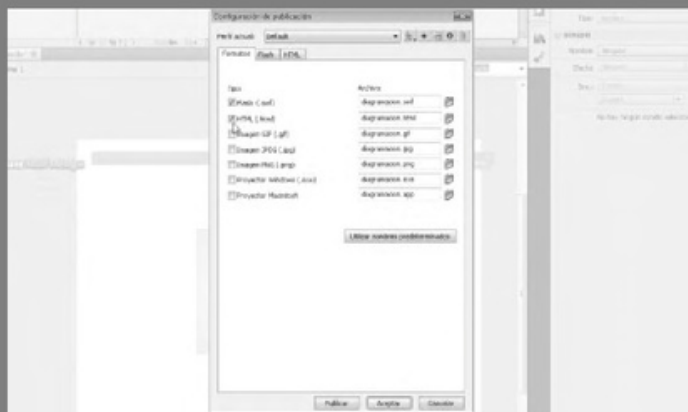
```
header.back.width = stage.stageWidth;
header.x = 0;
header.y = 0;
header.menuHeader.x = stage.stageWidth - header.menuHeader.width;
```




10 | FOOTER



HTML | 12



11 | PUBLICACIÓN



RESULTADO | 13

PASO | 10 Hasta aquí, el footer de la página no aparecía porque no estaba alineado con el tamaño del escenario. Para alinearlo, tomamos el movie clip **[back]** como referencia. Al considerar su alto, dentro de **[footer]** podría haber un sinfín de movies sin que afectaran la alineación del clip principal **[footer]**. Para ver el resultado, probamos la película.

PASO | 11 Vamos al menú **[Archivo/Configuración de publicación...]** y, en la solapa **[Formatos]**, marcamos **[Flash]** y **[HTML]**.

PASO | 12 Ingresamos en la solapa **[HTML]** y establecemos la opción **[Porcentaje]** dentro de **[Dimensiones]**. Luego, colocamos el valor **[100]** en los campos **[Anchura]** y **[Altura]**. Para terminar, presionamos **[Publicar]** para que se generen los archivos de nuestro trabajo.

PASO | 13 Al hacer clic en el archivo con extensión **HTML**, veremos el resultado del trabajo. Si redimensionamos el navegador, los contenidos ajustarán su tamaño y su posición según lo establecido en los pasos anteriores.

RESUMEN

La diagramación líquida es uno de los métodos para maquetar una web desarrollada en Flash. Es, quizás, el más atractivo por hacer un buen uso de la pantalla, aunque debemos aplicarlo siempre y cuando el sitio o el ambiente creado sea apto para este tipo de diagramación. Si no tenemos en cuenta este detalle, podemos convertir un proyecto simple en algo demasiado complicado.

MENÚ DE NAVEGACIÓN



El menú es uno de los elementos más importantes en una página web, porque permite al usuario interactuar con el sitio. Conozcamos algunos tips para que su creación sea una misión sencilla.

Un menú puede ser todo lo que nuestra imaginación pretenda, aunque su base será siempre la misma: una animación de inicio, una de cierre y unas cuantas acciones que le darán vida.

Uno de los métodos más utilizados a la hora de elaborar un menú es emplear movie clips para representar cada opción, en vez de usar un símbolo del tipo botón. Esta forma de trabajo resulta más flexible, ya que es más fácil manipular los movie clips para esta función.

USAR MOVIE CLIPS COMO BOTONES

Esta elección dependerá totalmente de la clase de menú que hagamos, ya que si nuestra idea es implementar algo muy simple, lo más sencillo será utilizar el símbolo botón proporcionado por Flash. Si trabajamos con símbolos de este tipo, el esfuerzo necesario será menor, dado que, por defecto, vienen definidas las instancias **[over]**, **[out]**, **[down]** y **[hit]** del botón. Esto, que ahorra trabajo, también en cierta forma nos quita la posibilidad de extender el atractivo de nuestro botón, y por eso lo más común es emplear movie clips.

UTILIZAR **MOVIE CLIPS** PARA REPRESENTAR LOS **BOTONES** RESULTA MÁS FLEXIBLE QUE EMPLEAR **SÍMBOLOS** DE TIPO **BOTÓN**.

+ BOTÓN BASE

A la hora de crear un menú de navegación, debemos comenzar con el primer botón de nuestro menú. A ese botón le agregaremos la gráfica que hayamos definido y, luego, le aplicaremos las acciones necesarias para que se quede esperando el clic del usuario. El siguiente paso es reproducir ese botón base para obtener el resto de los que tendrá el menú.

EL MENÚ EN EL SITIO

Un menú no solo es la forma de acceder a las diferentes secciones de nuestro sitio web, sino que va más allá: tiene que ir de la mano del diseño, y también, del resto de la navegación. Por ejemplo, no quedaría bien realizar un menú en el que, al hacer clic en un botón, este brillara, se agrandara, se sacudiera hacia los lados, rebotara, cambiara de color y, cuando terminara de hacer todos estos efectos, apareciera la página solicitada de repente, sin la más mínima animación. En un caso así, convendría evaluar si estamos aplicando demasiadas animaciones al menú o si, en realidad, sucede muy poco con las transiciones de las secciones. En un sitio tiene que haber armonía en cada acción, en cada movimiento de todos los elementos, no solo de una parte. Para mejorar la creación de menús de navegación, analizaremos algunos ejemplos presentes en Internet. Para empezar, en el sitio www.palazzograssi.it/italics,

veremos una forma de navegación totalmente distinta de las convencionales. Básicamente, no hay un menú para navegar la página, aunque sí hay secciones marcadas o módulos que muestran contenido diverso. La navegación en sí se lleva a cabo por medio de un brazo que cambia su mano, dependiendo del tipo de acceso en el que apoyemos el mouse. En este caso, la navegación y el menú son parte de la estética del sitio; cada elemento está comprometido con el otro, y ambos funcionan de maravilla.

Si analizamos <http://experience.jetblue.com>,

encontramos otro ejemplo de navegación totalmente distinta. Este es otro de los métodos interesantes que, con el pasar del tiempo, se está viendo

En www.expeditiontitanic.com encontramos una barra de navegación que sigue al mouse verticalmente.



más a menudo: consiste en esconder el menú detrás de una pequeña solapa que contiene una frase o palabra de referencia para que el usuario pueda localizarla con facilidad.

En este sitio, cuando hacemos clic en la solapa, se despliega el menú en todo el ancho de la página, y los botones textuales se reemplazan por imágenes. Aquí podemos apreciar el uso de movie clips para mejorar el menú, ya que el rollover y el rollout de cada opción traen consigo un grupo de animaciones armadas de tal manera, que nos llevan a pasar de una opción a otra solo porque resulta entretenido ver lo que está sucediendo.

DEFINIR EL MENÚ

Como vimos, para armar un menú de navegación, tenemos que considerar el buen gusto y la conexión entre todas las partes que conforman la página web, ya que un buen resultado no depende solo del menú, del fondo, de la navegación o de las transiciones.

No es incorrecto armar un menú convencional con animaciones simples, texto y transiciones comunes. Estos aspectos son variables, y tienen que concordar con el estilo de diseño y animación que hagamos nosotros y, también, con el cliente que nos está encargando el proyecto. No a todos les gustará que un botón se vuelva de

goma y que, al hacer clic, se vea aplastado contra la pantalla. Si prestamos atención al menú que creamos en el sitio de la colección, vemos un ejemplo claro de que no siempre es necesario hacer grandes cosas con un menú. Lo importante es que el efecto esté acorde a lo que es el resto del sitio web. ◀

En esta sección de www.jetblue.com, el pasaje entre botones resulta divertido porque aparecen nuevos elementos.

En el sitio de **The Influence Project**, dentro de www.fastcompany.com, veremos librerías 3D en funcionamiento, como **Sandy3D** y **Papervision**.



Una sección del sitio www.palazzograssi.it nos permite navegar arrastrando el mouse para movernos.

CARGA EXTERNA DE PELÍCULAS

Decidir si vamos a utilizar una sola película o varias dependerá no solo del tipo de proyecto, sino también de nuestra comodidad a la hora de programar, animar y maquetar, entre otros factores.

La elección de la cantidad de películas que se usarán es una de las decisiones más importantes que debemos tomar al iniciar un proyecto. El usuario tiene poca paciencia, y querrá ver algo rápido, por lo que nuestra tarea debe tener en cuenta la satisfacción de ese deseo. Un sitio en Flash puede pesar muy poco o realmente mucho, un parámetro clave que nos ayudará a tomar esta decisión. Pero, además, hay otros factores que determinarán cómo conviene trabajar; por ejemplo, el contenido del que disponemos y la forma en la que lo mostramos. Lógicamente, no pesa lo mismo un SWF

con una galería de imágenes con todos los archivos JPG dentro de la librería, que otra que carga las fotos de manera externa.

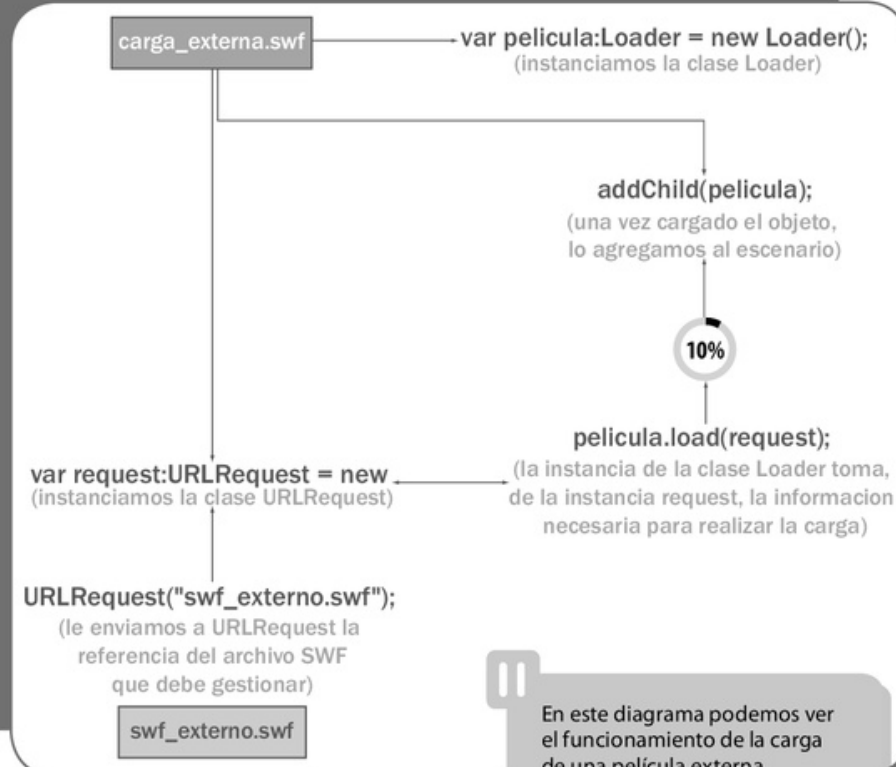
ANALIZAR EL SITIO

Antes de decidir, lo importante es evaluar cada una de las secciones de nuestro sitio para conocer la cantidad de subsecciones que contendrá, la información que vamos a mostrar, de qué forma la cargaremos (externa o internamente), cómo se compone gráficamente, si vamos a necesitar muchos o pocos archivos del tipo bitmap (como PNG o JPG) y el resto serán vectores, si cargaremos audio o video, y

si estos estarán embebidos o no. Una vez que hemos definido todos estos datos, volvemos sobre la cantidad de secciones. Si estamos hablando de un sitio pequeño (la home y 3 o 4 secciones interiores), lo recomendable es trabajar con un solo SWF, creando un movie clip por sección y manipulándolos según nuestras necesidades. De esta manera, nos aseguramos de cargar una sola vez el archivo para que el resto de la navegación sea fluida. Esto le brindará comodidad al usuario, porque no sufrirá tiempos de espera al cambiar de sección. Por otra parte, a nosotros nos dará la facilidad de tener toda la información en un mismo lugar, a excepción del contenido multimedia, como imágenes, audio y video, que al trabajar con un solo archivo, debe ser cargado de modo externo para que el proceso no sea tan extenso. Al manejarnos de esta manera, el usuario sentirá la fluidez de la navegación hasta que quiera ver el contenido externo, ya que tendrá que esperar su carga.

ELEGIR LA TÉCNICA IDEAL

Es fundamental que, como diseñadores, evaluemos cada caso particular para resolverlo del mejor modo. No existe una fórmula que diga que un sitio con determinada cantidad de secciones tiene que armarse de cierta forma. Si bien hay recomendaciones y tendencias, cada sitio se basará en una evaluación personal considerando todas las pruebas y errores que hayamos tenido. Nuestra experiencia nos ayudará a crear una estructura propia, que, aunque empecemos de cero en cada proyecto, mantendremos a lo largo de nuestros trabajos con características similares. A medida que avancemos en nuestro aprendizaje, adquiriremos más



En este diagrama podemos ver el funcionamiento de la carga de una película externa.



Sony (www.sonystyle.com) incluye videos anidados y cargadores por secciones, ya que el sitio es bastante pesado.



En www.thierrymugler.com hay videos como background y navegación automática por medio de un temporizador.

dinamismo y velocidad para concretar los proyectos. Y aunque los clientes y los emprendimientos sean distintos, tendremos en mente una estructura de creación concreta antes de iniciar cada caso.

CARGA EXTERNA DE PELÍCULAS

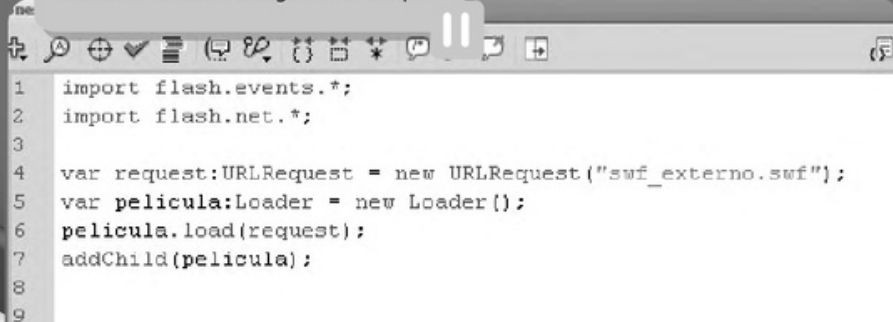
En ActionScript 2, la forma más simple de cargar una película externa era `[loadMovie("película.swf", _level);]`. En la versión 3, el proceso es diferente, ya que necesitamos instanciar un objeto de la clase `[URLRequest]` y otro de la clase `[Loader]`. La clase `[URLRequest]` se ocupa de gestionar y almacenar toda la información referente a la URL (dirección) de la película que se va a cargar. Por su parte, `[Loader]` gestiona la carga de la URL (dirección) proporcionada por la clase `[URLRequest]`, por medio del método `[Load]`. En ActionScript 3, entonces, si tenemos dos archivos SWF – uno como contenedor (`carga_externa.swf`) y otro como contenido (`swf_externo.swf`) – pondremos el siguiente código en el contenedor:

```
var request:URLRequest = new URLRequest("swf_externo.swf");
var pelicula:Loader = new Loader();
pelicula.load(request);
addChild(pelicula);
```

EL USUARIO TIENE Poca PACIENCIA Y QUIERE VER ALGO RÁPIDAMENTE. NUESTRO DISEÑO DEBE SATISFACER ESE DESEO.

Si lo analizamos por partes, con `[var request:URLRequest = new URLRequest("swf_externo.swf");]` instanciamos la clase `[URLRequest]` y le pedimos que gestione la dirección `[swf_externo.swf]`. El próximo paso será instanciar `[Loader]`, porque necesitamos un objeto con la capacidad de ir a buscar a la dirección indicada y traer la película que solicitamos mediante `[URLRequest]`.

El escenario está vacío, porque el contenido se cargará desde una película externa mediante código ActionScript.



Una vez que el objeto `[Loader]` –es decir, `[película]`– tiene todas las habilidades para realizar el trabajo, le pediremos que realice la carga por medio de `[load]: [película.load(request);]`. Ahora, `[película]` es el nuevo contenedor de la película `swf_externo.swf`. Sin embargo, aún no logramos ver el contenido de la película externa, porque nos falta un paso importante, y es que, una vez que `[load]` termina su trabajo de carga, tenemos que pedirle que vuelque (en este caso, sobre el escenario) el contenido de lo que cargó. Para hacerlo, usamos `[addChild(película);]` y, dentro de `carga_externa.swf`, veremos el contenido de `swf_externo.swf`.

ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué son Flash y ActionScript?

02 ¿Para qué sirve ActionScript?

03 ¿Qué ventajas ofrece el trabajo con AS3?

04 ¿Cuál es la importancia de pensar y diseñar un sitio?

05 Describa las interfaces en Flash.

06 Mencione las ventajas de organizar un proyecto.

07 ¿Qué es la diagramación líquida?

08 ¿Cuál es la función de un menú de navegación?

09 ¿De qué forma podemos diseñar un menú de navegación?

10 ¿Cómo podemos cargar una película externa?

PRÁCTICAS

01 Diseñe las bases de un sitio web en Flash.

02 Organice su proyecto.

03 Utilice las ventajas de la diagramación líquida.

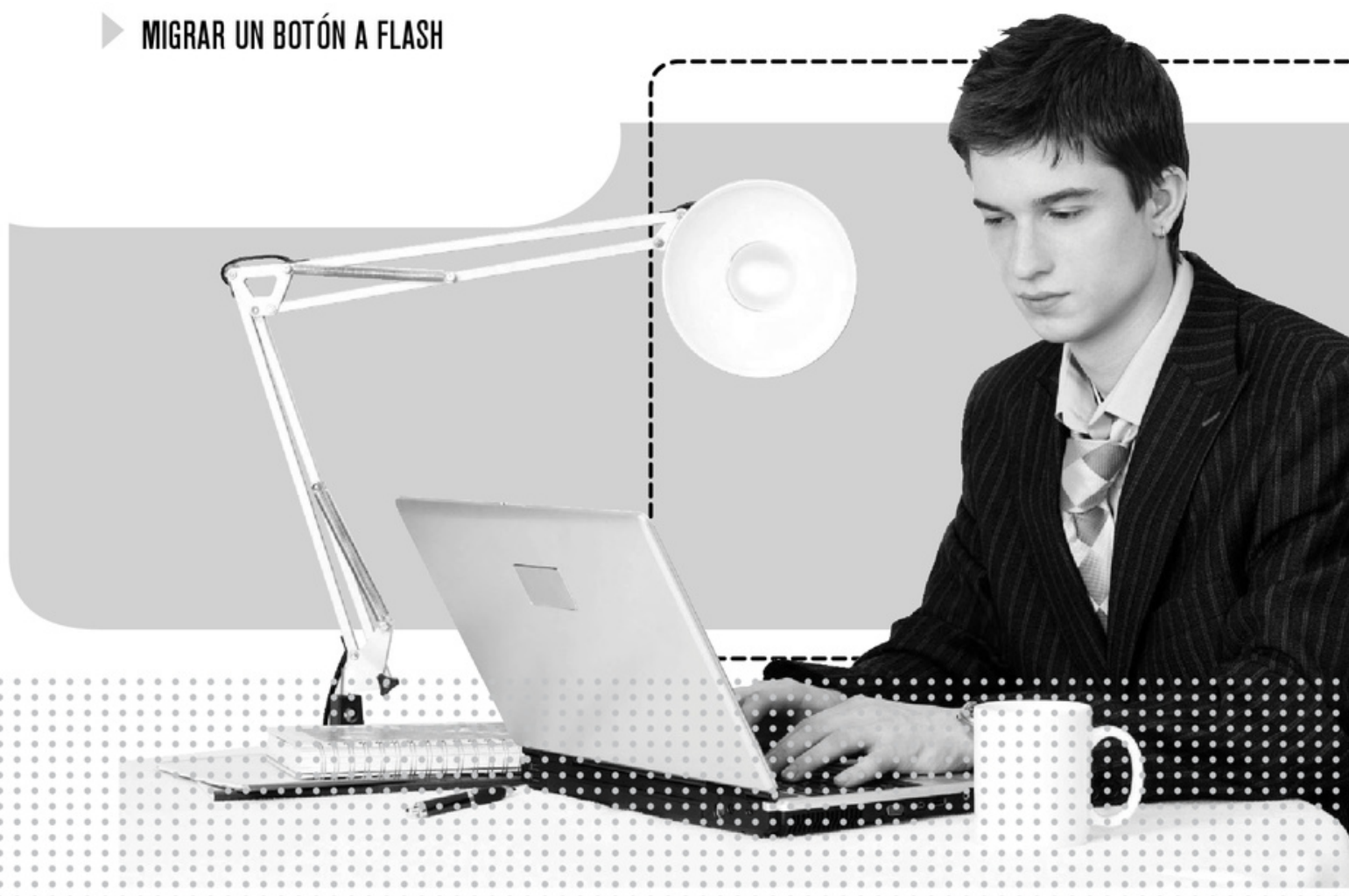
04 Realice un menú de navegación.

05 Cargue una película externa en su proyecto.



Herramientas de trabajo

- ▶ ADOBE CS5 WEB PREMIUM®
- ▶ LA INTERFAZ DE ADOBE FLASH CS5
- ▶ HERRAMIENTAS
- ▶ ILLUSTRATOR Y FLASH
- ▶ MIGRAR UN BOTÓN A FLASH



ADOBE CS5

WEB PREMIUM®

A la hora de diseñar y desarrollar sitios web de gran categoría, el paquete de aplicaciones de Adobe brinda una solución completa y profesional. Conozcamos por qué esta es la herramienta que debemos utilizar.

Adobe es sinónimo de diseño gráfico. Hace un tiempo que la calidad y la experiencia de su marca están volcadas al diseño web. El resultado es la suite más completa y aceptada del mercado, y Web Design pone en nuestras manos su última versión. CS5 trae mejoras muy notorias en sus herramientas de desarrollo web, y por eso lo utilizaremos a lo largo de toda la obra. **Adobe Creative Suite Web Premium®** contiene un conjunto de aplicaciones que nos brindan la posibilidad de abarcar cada uno de los aspectos y pasos de cualquier desarrollo web con variadas alternativas, de forma confiable y con respuestas profesionales. En esta oportunidad, Adobe prestó total atención a solucionar **el flujo de trabajo** en los proyectos web, al permitir que cada etapa sea más fácil y dinámica. Desde el diseño y la creación, hasta el desarrollo, el testeo y la finalización de un proyecto, se reduce el tiempo que se debe dedicar a la organización de cada fase de trabajo.



APLICACIONES PRINCIPALES

Las herramientas fundamentales que todo diseñador y desarrollador web necesitará para su día a día son las que nombramos a continuación:

Dw DREAMWEAVER CS5

Permite crear sitios web basados en los diferentes estándares de **maquetado y programación**, ofreciendo la posibilidad de trabajar en ellos de forma visual o escribiendo código directamente. Brinda un conjunto de utilidades destinadas a crear y aplicar hojas de estilos (CSS), verificar el funcionamiento en diferentes navegadores, validar nuestro código y subir los sitios a su alojamiento definitivo, entre otras opciones.

Fl FLASH PROFESSIONAL CS5

Este es el producto estrella de Adobe para la creación de sitios interactivos y animados con entornos virtuales y **envolventes**, en donde los usuarios pueden experimentar diferentes formas de navegación. Cuenta con características sobresalientes en sus herramientas de creación y con un lenguaje de programación cada vez más potente: ActionScript 3.0.

Ps PHOTOSHOP CS5 EXTENDED

Es la mejor aplicación para la edición y la manipulación avanzada de imágenes digitales. Este programa brinda una gran cantidad de herramientas para el retoque fotográfico, el ajuste de las propiedades de las imágenes y la composición de ilustraciones. Además, permite la inclusión de elementos en 3D y la creación de animaciones.

Ai ILLUSTRATOR CS5

Es la aplicación de creación y diseño **vectorial** más reconocida en el mundo del diseño. Ofrece una precisión y características sobresalientes al momento de trabajar en la creación de gráficas, logos, ilustraciones y otros dibujos para usar en sitios web.



APLICACIONES COMPLEMENTARIAS

Dentro de la suite de Adobe también podemos encontrar otras aplicaciones útiles que nos servirán de complemento en el desarrollo de nuestros proyectos web. Conozcamos cuáles son:

Fw FIREWORKS CS5
Aplicación especial para el desarrollo de proyectos web, brinda las herramientas necesarias para organizar contenidos, optimizar sitios y crear diseños, tanto en modo vectorial como en mapa de bits.

Fc FLASH CATALYST CS5
Permite crear contenidos interactivos y animados basados en diseños provenientes de Photoshop, Illustrator o Fireworks, sin necesidad de saber ActionScript y sin escribir código.

FB FLASH BUILDER
Anteriormente conocido como **Flex Builder**, se utiliza para crear sofisticadas aplicaciones web multiplataforma (**RIA**, *Rich Internet Applications*).

Ct CONTRIBUTE CS5
Ideal para publicar y administrar sitios web, permite probar el funcionamiento de estos en diferentes navegadores, editar sus contenidos por medio de XML, editar texto HTML y crear contenidos en forma colaborativa para ahorrar tiempo de desarrollo. ◀

ADOBE CS5 WEB PREMIUM INCLUYE LAS MEJORES APLICACIONES PARA DISEÑAR Y DESARROLLAR SITIOS WEB PROFESIONALES.



¿QUÉ ES ADOBE?

Adobe es una empresa líder a nivel mundial, con más de 20 años de experiencia en la producción de aplicaciones orientadas al diseño gráfico.

Fundada en 1982, cuenta con más de 6000 empleados y tiene una facturación que supera los 2500 millones de dólares anuales. A fines de los años 80, marca su primer hito con el lanzamiento de Illustrator, y ya en los 90 supera su propio éxito con Photoshop, con lo cual queda posicionada definitivamente como la empresa más grande en el sector. Contando ya con un marcado liderazgo, en 1999 lanza otro programa que arrasa con la competencia: InDesign. Hace unos años se vuelca al mercado del video profesional, rubro en el que también lidera, con Premiere Pro y After Effects. Es la firma creadora de un formato que hoy todos damos por sentado como un estándar: el PDF.

LA INTERFAZ DE ADOBE FLASH CS5

Para comenzar a familiarizarnos con Flash CS5, primero veremos una descripción de cuáles son y dónde están ubicadas las principales funcionalidades que nos brinda esta nueva versión del programa.

Flash, al igual que todos los programas de Adobe, cuenta con un diseño de interfaz excelente, que nos permite trabajar de forma muy cómoda y ágil. Además, nos da la posibilidad de organizar los principales paneles de varias maneras distintas, dependiendo del trabajo que estemos realizando. ◀

1

BARRA DE HERRAMIENTAS.

Donde encontraremos todo lo necesario para manipular y crear elementos en el escenario.

2

MENÚ PRINCIPAL.

Donde se encuentran todas las funciones y opciones del programa.

3

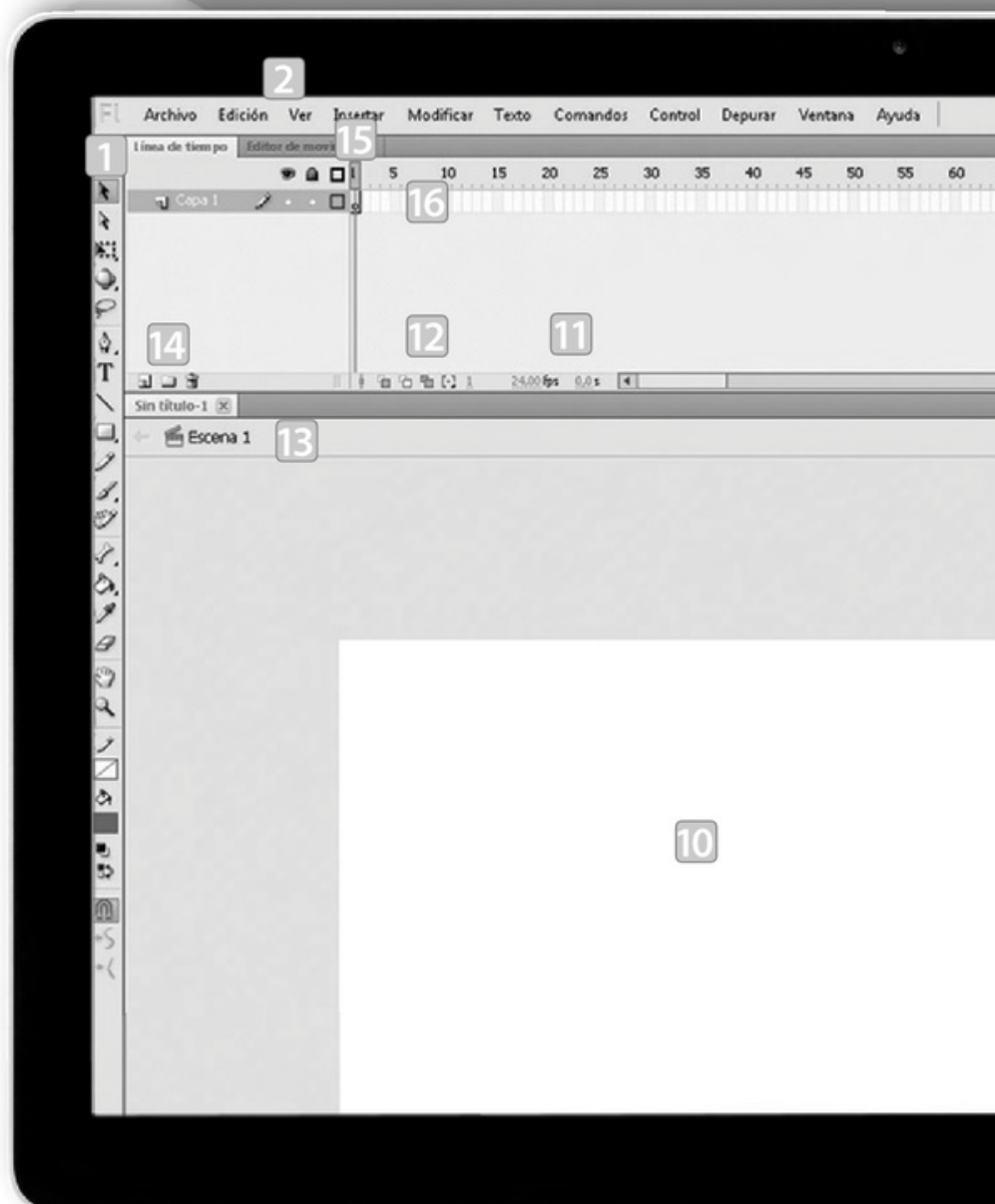
VISUALIZACIÓN DE LA LÍNEA DE TIEMPO.

Aquí establecemos todo el desarrollo temporal, el ordenamiento y las secuencias de la película Flash.

4

CONFIGURACIÓN DEL MODO DE VISUALIZACIÓN DE LA INTERFAZ.

Según lo que hagamos con Flash, podremos configurar la forma en la que veremos las opciones principales cuando estemos, por ejemplo, animando o programando.



5

Nombre del documento actual.

6

Información sobre la versión del reproductor y del ActionScript con los que se publicará la película en la Web.

7

Selector del archivo de programación externa que se utilizará.

8

Información de la película, frames por segundo, medida y color del escenario.

9

Selector de escena, editor de símbolos y escala.

10

Escenario para ubicar todo el diseño y los elementos interactivos que será visualizados por el usuario.

11

Visualización de la cantidad de fotogramas que la película recorre en 1 segundo y en cuál de ellos nos encontramos en un determinado momento.

12

Barra de opciones para configurar de qué modo visualizar las animaciones cuadro a cuadro mientras las estamos realizando.

13

Indicador de escena actual.



14

Barra desde donde podremos agregar, ordenar y borrar capas.

15

La línea de tiempo está compuesta por fotogramas. En esta sección crearemos todo lo relacionado con los movimientos y las transiciones.

16

Solapa de edición de movimiento. Cuando estamos realizando animaciones, aquí aparecen todas las opciones relacionadas con los tipos de animación y los efectos.



INTERFAZ FLASH

Si somos usuarios de otras aplicaciones de Adobe, nos encontraremos con que muchas de las opciones y herramientas que tiene Flash son iguales a las de programas como Photoshop o Illustrator. Gracias a esto, Flash es la herramienta más elegida por los diseñadores para la creación de sitios web y aplicaciones. La interfaz visual de este programa proporciona un manejo total de los comportamientos de los elementos con los que estemos trabajando. Muchos de esos comportamientos son accesibles tanto desde el menú principal como mediante la programación en el panel de acciones con ActionScript 3.0. Por lo tanto, no necesitamos ser programadores expertos para desarrollar sitios altamente interactivos y que aprovechen al máximo las capacidades del programa.

HERRAMIENTAS

Las tareas de programación, diseño gráfico, ilustración y trabajo audiovisual requieren herramientas específicas que optimicen el resultado de cada una. Veamos de qué se trata.

Flash contiene algunas herramientas que permiten realizar tareas relacionadas con distintas áreas del diseño web. Por ejemplo, posee un avanzado editor de código que se utiliza para programar en ActionScript, el lenguaje de programación de Flash. También cuenta con muchas herramientas para ilustrar y pintar, o para realizar gráficos vectoriales. Asimismo, tiene un pequeño editor de sonido destinado a recortar, cambiar el volumen o aplicar efectos básicos a los archivos de audio que incorporemos. Sin embargo, hay otras aplicaciones desarrolladas para efectuar tareas específicas, y para ese fin cuentan con herramientas mucho más avanzadas,

que permiten hacer el trabajo mejor y más rápido que si lo hiciéramos dentro de Flash. Por ejemplo, Adobe **Illustrator**, para realizar ilustraciones vectoriales; Adobe **Photoshop**, para

editar y manipular fotografías; y Adobe **Soundbooth**, para crear y editar audio.

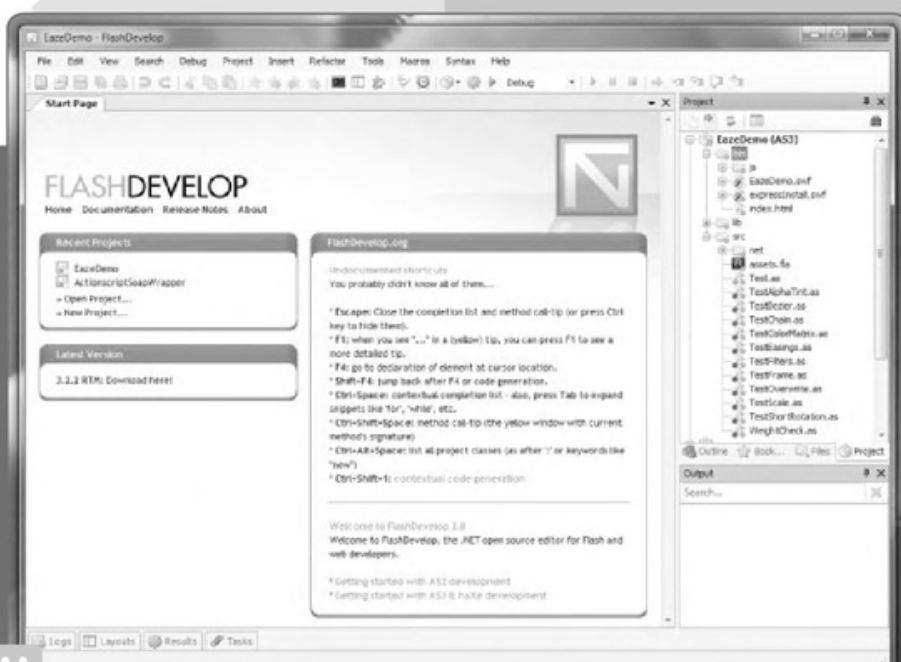
FLASHDEVELOP: PROGRAMAR DE MANERA PROFESIONAL

Cuando el proyecto que estemos realizando requiera un nivel muy avanzado de programación en ActionScript, el editor de Flash seguramente nos resultará escaso en cuanto a prestaciones. FlashDevelop es un potente editor open source muy popular entre los desarrolladores profesionales para realizar aplicaciones en ActionScript 2 y 3. Entre sus características destacadas, encontramos las siguientes: autocompletado de código, ayuda inteligente de ActionScript, exploración y manejo de clases, marcadores de líneas, y resaltado de código XML, HTML/PHP, JS, CSS. Podemos consultar información adicional y descargar gratuitamente este editor desde www.flashdevelop.org.

FLASHEFF: EFECTOS ESPECIALES PARA TEXTO E IMÁGENES

FlashEff es un componente de Flash y una herramienta interactiva fácil de utilizar, que permite crear efectos de transición de alta calidad. Se trata de una biblioteca de efectos prácticamente infinita, que podemos utilizar y personalizar para generar contenidos profesionales. Se instala fácilmente a través de Adobe Extension Manager y queda automáticamente incorporado a Flash. Con este componente es posible crear, en pocos pasos, transiciones muy llamativas entre imágenes y objetos, con efectos visuales sumamente modernos. También existe una larga lista de efectos aplicables a cualquier texto para lograr resultados similares a los que se ven en los avisos publicitarios televisivos

AUNQUE **FLASH** PERMITE TRABAJAR CON **IMÁGENES, AUDIO Y EFECTOS**, EXISTEN **APLICACIONES ESPECIALES** PARA ESAS TAREAS.



Con **FlashDevelop**, la programación en ActionScript 3.0 se vuelve mucho más fácil y profesional. Visitemos el site oficial de la aplicación: www.flashdevelop.org.



e, incluso, en títulos de cine. Esta herramienta puede adquirirse de forma online, y también posee una versión gratuita que trae una galería limitada de efectos. En **www flasheff.com** podemos obtener más información y ver demos de ella.

PAGEFLIPPER: REVISTAS Y ALBUMES DE FOTOS

PageFlipper es otro componente para Flash que, una vez instalado, nos ayuda a crear el efecto de **cambio de página** de una revista. Gracias a él, pondremos a disposición de los usuarios de nuestro sitio una interfaz muy intuitiva y natural, que les permitirá **hojear** el contenido que les presentemos.

Este componente es ideal cuando queremos presentar la información institucional de una empresa, un álbum de fotos o contenido de tipo editorial a través de Internet. Entre otras opciones, nos permite cambiar de página mediante clics o el arrastre del mouse; usar clips de película, imágenes o archivos SWF externos como páginas; y personalizar la apariencia estética.

Podemos obtener más información y ver una demostración de este componente yendo a **www.flashloaded.com**.

ADOBE SOUNDBOOTH CS5: CONSOLA PROFESIONAL DE AUDIO

Este software está pensado para trabajar en estrecha relación con los otros productos de la suite de Adobe. Permite crear o editar audio de alta calidad, realizando tareas con un alto control sobre el resultado creativo final. Aunque podemos obtener información sobre la

FlashEff permite agregar sofisticadas transiciones a las imágenes y los textos de nuestro sitio Flash.

Si queremos grabar y editar audio de manera profesional para nuestro sitio, Adobe Soundbooth CS5 es lo que necesitamos.



1 2 3 4 5 6 7 8 9 10 11 12

pageFlipper



Automated page numbering

Page numbering can be automatically applied to every page, or odd/even pages only.

Custom live preview

The built-in live preview displays dimensions in the Flash IDE reference during development.

Con PageFlipper podremos realizar revistas digitales realistas, gracias al efecto que emula el acto de **pasar la página**.

herramienta en **www.adobe.com/es**, vamos a mencionar algunas de sus características destacadas:

- ▶ Permite grabar audio en vivo con un micrófono, o importar audio de CDs y MP3.
- ▶ Brinda la posibilidad de mezclar y editar audio en formato multipista.
- ▶ Incluye una librería con más de 10.000 efectos de sonido de uso libre y más de 130 partituras personalizables.
- ▶ Simplifica la sincronización de audio y video para la Web.
- ▶ Permite arrastrar y soltar para equilibrar los niveles de sonido, recortar las seccio-

nes no deseadas y crear bucles propios de música.

ADOBE ILLUSTRATOR CS5: ILUSTRACION Y DISEÑO GRAFICO

Si lo que queremos es generar ilustraciones y gráficos vectoriales de nivel profesional, Illustrator es el programa indicado. En las próximas páginas hablaremos en detalle sobre él. ◀



ILLUSTRATOR Y FLASH

Flash contiene algunas herramientas de diseño que nos permiten crear distintos elementos para un sitio web de manera sencilla. Sin embargo, si queremos realizar un diseño profesional, la opción ideal es Adobe Illustrator.

Una de las características que hicieron que la mayoría de los diseñadores web adoptaran Flash a la hora de realizar animaciones o aplicaciones multimedia para sus sitios fue el hecho de que manejaba gráficos vectoriales, lo que se traducía en archivos de muy pocos kilobytes y muy buena calidad gráfica.

Sin embargo, las tendencias de diseño fueron cambiando y empezaron a incorporarse efectos visuales (por ejemplo, sombras proyectadas) que debían ser realizados en Photoshop y, luego, importados a Flash como archivos de mapa de bits. Esto, además de representar un doble trabajo (diseñar con un programa y exportarlo hacia otro), aumentaba el peso de los archivos debido al agregado de mapas de bits.

Otra de las desventajas de usar Photoshop junto a Flash es que el resultado no puede ser editado dentro de este último, porque no tiene herramientas para manipular mapas de



EL PROYECTO DE SITIO

Para reforzar los conocimientos adquiridos y poner en práctica nuestras habilidades, podemos visitar el sitio de los proyectos **HTML** y **Flash** de la colección (www.redusers.com/cursos/webdesign), que nos servirán para conocer todos los detalles del código y el maquetado. Además, desde allí podremos descargar un **CD interactivo** con **guías visuales**, **glosario de términos** y una completa guía de preguntas frecuentes. Mediante los **videotutoriales**, perfeccionaremos nuestras habilidades al trabajar con las aplicaciones de la suite de diseño **Adobe Web Premium CS5**, herramientas que nos permitirán convertirnos en los mejores profesionales.

bits. Cualquier cambio que necesitemos hacer, debemos realizarlo en Photoshop y, después, importarlo otra vez en Flash.

ILLUSTRATOR, EL COMPLEMENTO PERFECTO DE FLASH

Las últimas versiones de Illustrator integran lo mejor del diseño vectorial con lo atractivo de los efectos visuales, para así expresarnos creativamente sin límites. Un adicional importante es que podemos exportar nuestro diseño hacia Flash, ya

que este programa respetará fielmente lo realizado. Además, lo que importe-mos podrá ser editado libremente en él, porque sigue siendo vectorial.

De esta manera, tenemos la posibilidad de crear gráficos en Illustrator eligiendo el color de relleno, el grosor y el tipo de contorno, con la tranquilidad de saber que no se verán modificados cuando se trasladen a Flash. Incluso, los rellenos de degradé y las máscaras de recorte pueden pasar de un programa al otro sin sufrir alteraciones.

Illustrator tiene un arsenal muy vasto de herramientas para trabajar con vectores, con lo cual nos permite crear desde figuras geométricas básicas hasta otras más avanzadas, como polígonos, estrellas, espirales o grillas. Además, estas pueden combinarse fácilmente entre sí para generar otras mediante métodos de intersección, sustracción o adición.

Si es necesario, también disponemos de herramientas de edición para cada nodo individual de una figura vectorial, lo que nos permite deformarla a nuestro antojo. Flash también posee algunas de estas herramientas, pero son las de Illustrator las que nos permiten alcanzar resultados mucho más profesionales y con un mayor control sobre el producto final.



Las herramientas de dibujo de Illustrator nos permiten realizar ilustraciones vectoriales de gran detalle.

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using Content here, content here



Usted podrá administrar los contenidos de su sitio web como por ejemplo un link de novedades, consultar registros, enviar mensajes, realizar altas bajas y modificaciones en sus bases de datos, administrar usuarios.



"Por favor sientate libre de comentar de manera fugaz mis trabajos..."

ENTRAR



Todos los efectos, como las sombras y los biseles, pueden traspasarse sin problemas de Illustrator a Flash.

Otro gran adelanto de las últimas versiones de Adobe Illustrator y de Flash es que permiten utilizar e intercambiar efectos visuales entre ellos. Esto quiere decir que cualquier efecto tipo sombra, brillo o biselado aplicado a una figura realizada vectorialmente en Illustrator puede exportarse a Flash, manteniendo la forma, el color y el efecto elegido; además, puede editarse posteriormente en este último programa.

JUNTOS MEJORAN EL FLUJO DE TRABAJO

Es muy común que, en el diseño de sitios para Internet, las tareas estén divididas: una persona se ocupa de la creatividad y el diseño, y otra se encarga, posteriormente, de agregar las animaciones y la programación. En general, los diseñadores gráficos —quienes en este caso se ocuparían de la parte estética— utilizan a diario Illustrator y están muy familiarizados con sus herramientas. Cuando finalizan su trabajo, lo derivan al diseñador multimedia o al programador para que lo importe a Flash para animarlo y darle interactividad. Las últimas versiones de ambos programas pueden garantizar un alto porcentaje de éxito en el traspaso de la información de uno a otro, y de esta manera, permiten generar un flujo de trabajo mucho más dinámico.

Con Illustrator, los diseñadores pueden dar rienda suelta a su creatividad. Sin dudas, este es el complemento ideal de Flash a la hora de realizar proyectos web profesionales.

TRAZOS ESPECIALES, TEXTOS SIGUIENDO UNA RUTA Y EFECTOS VISUALES

Algunas herramientas que posee Illustrator, pero no están presentes en Flash, son los trazos artísticos y el texto siguiendo una ruta. Con la primera podemos realizar figuras geométricas o dibujos a mano alzada con un trazo que puede imitar el uso de distintos tipos de plumas. El resultado, por ejemplo, puede ser un dibujo que simula estar realizado con crayón o un ideograma chino hecho con tinta. Lo mejor de esto es que dicho resultado es vectorial y, por lo tanto, puede ser manipulado en Flash para cambiarle el color, escalarlo o aplicarle algún otro efecto, sin perder calidad. En el caso del seguimiento de rutas, el trabajo consiste en hacer que un texto que tenga cualquier longitud de caracte-

ILLUSTRATOR CS5 INTEGRA LO MEJOR DEL DISEÑO VECTORIAL CON LO ATRACTIVO DE LOS EFECTOS VISUALES.

teres pueda recorrer una ruta dibujada previamente, por ejemplo, con forma de espiral. El único inconveniente que tiene esta herramienta es que, al exportarse a Flash, el texto con la nueva forma sigue siendo vectorial, pero pierde la posibilidad de que su contenido sea editado, es decir, de modificar sus características y propiedades tipográficas.

MIGRAR UN BOTÓN

PASO A PASO

En este apartado explicaremos cómo crear un botón con texto, icono y sombra, para luego trasladarlo a Flash manteniendo su aspecto.



01 | CREAR LA BASE



AGREGAR ICONO

02

Como mencionamos anteriormente, diseñar en Illustrator nos permite tener un control más preciso de lo que hacemos, y nos brinda más herramientas de dibujo y efectos que Flash. Por ese motivo, vamos a utilizarlo para volcar nuestra creatividad al crear la interfaz de nuestra web, que luego llevaremos a Flash para agregar la interactividad. ◀



PASO | 01 Una vez que abrimos Illustrator y nos encontramos en un documento en blanco, elegimos la herramienta **[Rectángulo redondeado]** y dibujamos nuestro botón con el color de relleno y el borde deseados. Para hacer recto el lado superior, escogemos la herramienta de eliminación de nodos **[Eliminar punto de ancla]** y suprimimos los dos superiores. Para finalizar, convertimos los dos nodos superiores restantes de curvos a rectos.

PASO | 02 Con la herramienta de dibujo **[Estrella]**, creamos una figura con el color y el grosor de borde que necesitemos. En este caso, vamos a rellenarla con un tono en degradé amarillo y naranja, generado por medio de la paleta **[Degradado]**.

PASO | 03 A continuación, con la herramienta **[Texto]** escribimos la etiqueta del botón.

A FLASH

04 | APLICACIÓN DE EFECTOS

★ AGREGAR A FAVORITOS



Pegar

- ☐ Pegar como mapa de bits
- ☒ Pegar utilizando preferencias de importación de archivos de AI
- ☒ Aplicar configuración de importación recomendada para resolver incompatibilidades
- ☒ Mantener capas

COPIAR Y PEGAR

05

03 | TEXTO Y ALINEACIÓN

★ AGREGAR A FAVORITOS

EDICIÓN EN FLASH

06

Luego, para alinear el texto verticalmente centrado con el icono de la estrella, utilizamos la opción **[Centrar verticalmente]** del panel **[Alinear]**. Vemos cómo queda insertado el texto.

PASO | 04 Finalmente, seleccionamos la base del botón y le agregamos el efecto de sombra deseado. Para hacerlo, vamos al menú **[Efecto/Estilizar/Sombra Paralela...]** y modificamos los parámetros disponibles hasta conseguir el efecto que queremos darle al botón.

PASO | 05 Una vez que tenemos el diseño final del botón, lo seleccionamos completo y presionamos la combinación de teclas **<Ctrl + C>** para copiarlo. A continuación, vamos a Flash y, en una nueva capa, presionamos **<Ctrl + V>** para pegarlo. Se abrirá una ventana de diálogo con algunas opciones, donde debemos dejar las seleccionadas en la imagen.

PASO | 06 Cuando tenemos nuestro botón importado en Flash, podemos editarlo con las herramientas de este programa, ya que sigue siendo vectorial.

RESUMEN

Como vemos, actualmente no tenemos que preocuparnos por la compatibilidad de diseño entre Illustrator y Flash, porque el trabajo que hagamos en el primero podrá importarse en el segundo sin temor a perder ninguno de los atributos estéticos elegidos en el momento de crearlo.

ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué es Adobe CS5?

02 Nombre las aplicaciones principales que incorpora Adobe CS5.

03 Mencione las aplicaciones complementarias que incorpora Adobe CS5.

04 Enumere las ventajas de trabajar con Adobe CS5.

05 ¿Cuáles son las características de la interfaz de Adobe flash CS5?

06 Mencione las partes más importante de la interfaz de trabajo de Flash.

07 ¿Qué herramientas de diseño ofrece Flash?

08 ¿Para qué sirve Illustrator?

09 ¿De qué forma podemos complementar Illustrator y Flash?

10 ¿Cómo es posible mejorar el flujo de trabajo incorporando Illustrator y Flash?

PRÁCTICAS

01 Instale Adobe CS5.

02 Ejecute cada una de las aplicaciones principales de esta suite.

03 Identifique las partes integrantes de la interfaz de Adobe Flash.

04 Diseñe un botón en Illustrator.

05 Migre el botón creado a Flash.

Textos

- ▶ TEXTO CLÁSICO EN FLASH
- ▶ SUAVISADO TIPOGRÁFICO
- ▶ CREAR TEXTO CLÁSICO
- ▶ TEXT LAYOUT FRAMEWORK
- ▶ TLF: FLUJO ENTRE TEXTOS
- ▶ TEXTO EN FORMATO HTML
- ▶ APLICAR CSS A UN CAMPO DE TEXTO DINÁMICO
- ▶ CREAR UNA SCROLL BAR CON EASING



TEXTO CLÁSICO EN FLASH

En la nueva versión de Flash, el texto clásico ya no es la opción predefinida. Ahora es la segunda alternativa, o la manera antigua de editar texto, y mantiene las mismas cualidades que en las ediciones anteriores del programa.

Aunque Flash CS5 nos permite seguir usando el texto en el formato que traían las versiones anteriores, el formato predefinido actual es nuevo: se denomina **TLF (Text Layout Format)** y lo veremos en detalle más adelante.

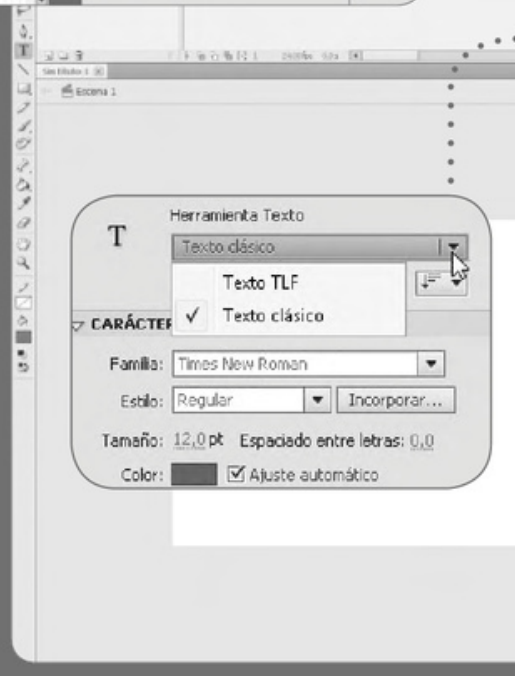
El modo clásico de trabajo con texto resulta útil si necesitamos seguir manejando las versiones antiguas del programa, para textos que no requieren ningún efecto 3D (rotación, perspectiva, zoom, etcétera) o cuando no precisamos que se cargue de manera dinámica en cajas encadenadas. De todos modos, podemos pasar de un motor de texto al otro en cualquier momento.

Trabajar con **texto clásico** nos permite utilizar fuentes PostScript® tipo 1, TrueType® y de mapa de bits (solo en Macintosh). También podemos definir atributos como el tamaño, el estilo, el color y el interlineado, además de aplicar transformaciones como la **rotación**, el cambio de **escala**, el **sesgado**, el **volteado** o la **reflexión**. Este modo nos permite orientar el texto de manera horizontal o vertical (en esta opción podemos cambiar el sentido de lectura de izquierda a derecha), regular el interlineado de un párrafo y comprobar su ortografía. Las instancias de los campos de texto clásico son como las de los símbolos de Flash, y pueden manipularse mediante ActionScript, código que también puede dar formato a los campos de texto dinámico (*Dynamic Text*) y a los de introducción de texto (*Input Text*). En los campos de texto es posible conservar el formato de **texto enriquecido**, y utilizar atributos y etiquetas HTML. Si empleamos HTML para el contenido de un campo de introducción de texto o



En este caso, un texto clásico es usado para enmascarar varias capas de imágenes por debajo de él.

uno dinámico, podemos poner el texto alrededor de una imagen, o incluirlo en un archivo SWF, JPEG o en un clip de película. Todos los campos de texto clásico admiten la codificación en formato **Unicode**. Cuando trabajamos con texto estático (*Static Text*) o dinámico (*Dynamic Text*), podemos realizar un **hipervínculo** haciendo clic en la herramienta **[Seleccionar la caja de texto]**, en caso de querer convertir todo el párrafo en un link; o en **[Texto]**, para convertir solo el carácter o los caracteres necesarios. El texto puede utilizarse con **[Capa Máscara]** para enmascarar objetos que estén por debajo de él. Esta opción



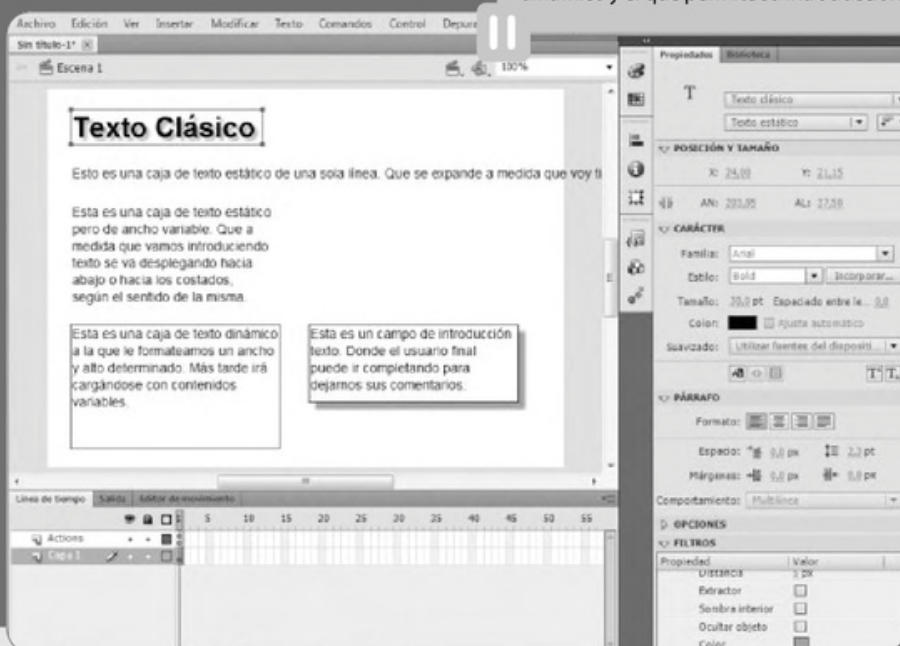
funciona con cualquiera de los casos de suavizado, a excepción del denominado **[Utilizar fuentes del dispositivo]**. El modo **[Texto clásico]** tiene tres opciones: **[Texto estático]**, **[Texto dinámico]** e **[Introducción de texto]**, que analizaremos a continuación.

TEXTO ESTÁTICO

Este es el cuadro de texto que creamos para ser leído o visto en Flash, y no puede ser modificado por medio de ActionScript, ya que no es posible asignarle un nombre de instancia que

EL FORMATO DE TEXTO PREDEFINIDO QUE UTILIZA FLASH CS5 ES NUEVO Y SE DENOMINA TLF (TEXT LAYOUT FORMAT)

Visualización de las distintas categorías del texto clásico en Flash: texto estático, texto dinámico y el que permite su introducción.



Cuando elegimos la herramienta **[Texto]**, podemos definir el tipo de texto que vamos a usar.

Vemos un posible uso del texto en modo dinámico: una caja de texto multilinea.



conecte el script al texto. Puede tener una disposición horizontal (escritura occidental) o vertical (escritura oriental), y su visualización en el archivo **SWF** no genera conflictos con las fuentes no instaladas en otros equipos, porque estas se convierten a curvas al ser exportadas. Podemos generar este tipo de texto si seleccionamos la herramienta **[Texto]**, hacemos clic en el escenario y comenzamos a escribir para que la caja crezca hacia la izquierda (disposición horizontal) o hacia abajo (disposición vertical). También podemos hacer clic en el escenario y arrastrar el mouse hasta determinar el ancho de la caja, para luego comenzar a escribir. El ancho de la caja se mantendrá, pero esta se irá expandiendo hacia abajo (disposición horizontal) o hacia la izquierda o derecha (disposición vertical) a medida que se le agrega texto. Si hacemos el **texto seleccionable**, este podrá copiarse y pegarse en otra aplicación; en tanto que si usamos las variantes de texto **[Vertical]** y **[Vertical, izquierda a derecha]**, podemos hacer que el carácter rote a 90 grados en los ejes X e Y.

TEXTO DINÁMICO (DYNAMIC TEXT)

Este tipo de cuadro de texto se usa para la visualización de datos que cambian por medio de la interacción con ActionScript, por lo que es posible asignarle un nombre de instancia. Su visualización correcta en el navegador depende de la incorporación de las fuentes utilizadas al archivo SWF o del uso de otras genéricas. Si trabajamos con un ancho fijo y una caja que va creciendo hacia abajo, no podemos cambiar la disposición del texto a horizontal o vertical, pero sí podemos redimensionar la caja en cualquiera de los dos sentidos. En esta opción, si el texto excede el marco, es posible visualizarlo al deslizar una barra de desplazamiento o, simplemente, con la rueda del mouse. Desde el panel **[PÁRRAFO]**, en el desplegable **[Comportamiento]** podemos configurar si la caja de texto será de una sola línea (**Línea única**) o de varias a modo de párrafo (**Multilínea**); hacer que el texto sea seleccionable si queremos que se pueda copiar y pegar; y conservar el formato de texto enriquecido con algunas etiquetas de HTML (como fuentes e hipervínculos) si hacemos clic en **[Generar texto como HTML]**. Si precisamos un borde fino negro y uno blanco para la caja, podemos hacer clic en **[Mostrar borde alrededor del texto]**.

INTRODUCCIÓN DE TEXTO (INPUT TEXT)

Este cuadro de texto, que se utiliza básicamente para formularios y encuestas, permite que el usuario complete con datos el campo en blanco. Como en el caso del texto dinámico, debemos incorporar las fuentes utilizadas al archivo SWF o emplear las tipografías genéricas para obtener una correcta visualización. Este tipo de texto puede tener un nombre de instancia para interactuar con ActionScript, y su construcción es igual a la de los tipos anteriores. En el panel **[PÁRRAFO]**, podemos elegir la opción **[Contraseña]** en el desplegable **[Comportamiento]**, para que el campo permita ingresar contraseñas de la forma más utilizada, es decir, ocultando los caracteres mediante pequeños círculos.

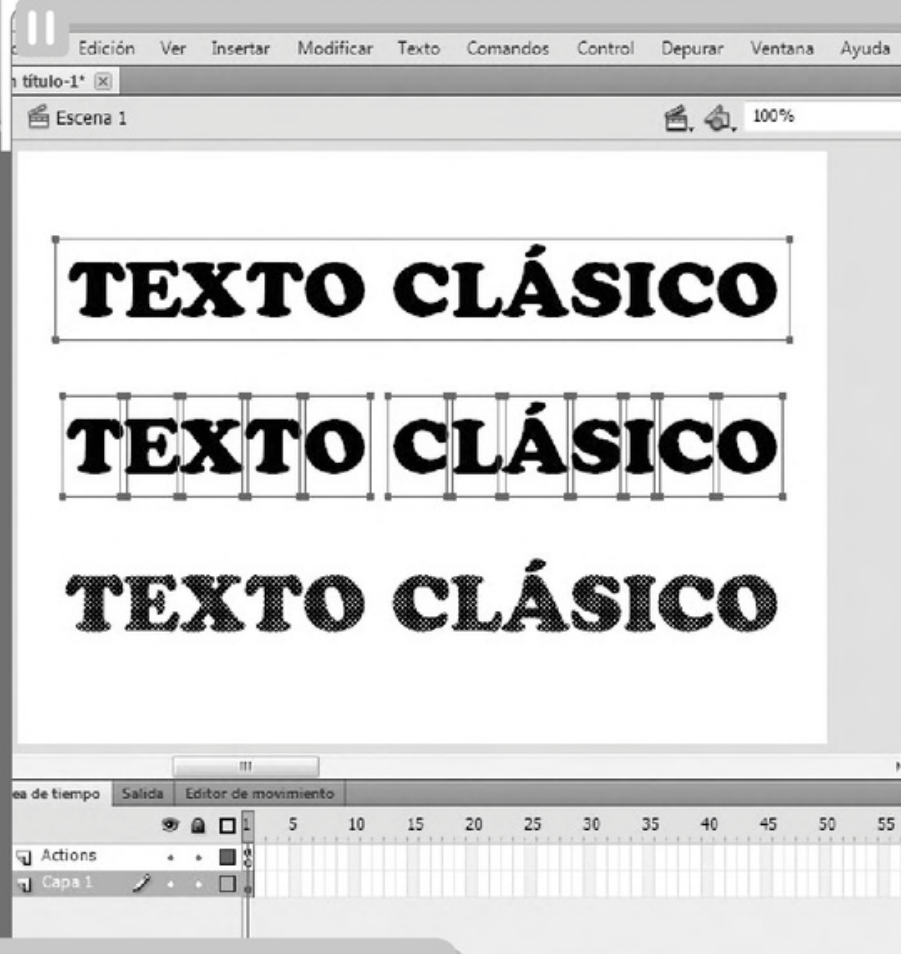
ESTABLECER LOS ATRIBUTOS EN EL TEXTO CLÁSICO

En Flash podemos elegir la tipografía que tendrá nuestro texto, su tamaño, el estilo, el color, el espaciado entre letras, el ajuste entre ellas y la posición con respecto a la línea del renglón. También podemos editar las características del párrafo, como alineación, márgenes, sangrías e interlineado. En el texto estático, las tipografías se exportan en el archivo SWF publicado y, para la opción horizontal de él, es posible utilizar fuentes de dispositivo en vez de exportarlas con el archivo. En el caso del texto dinámico y de los campos de introducción de texto, Flash guarda los nombres de las tipografías y,

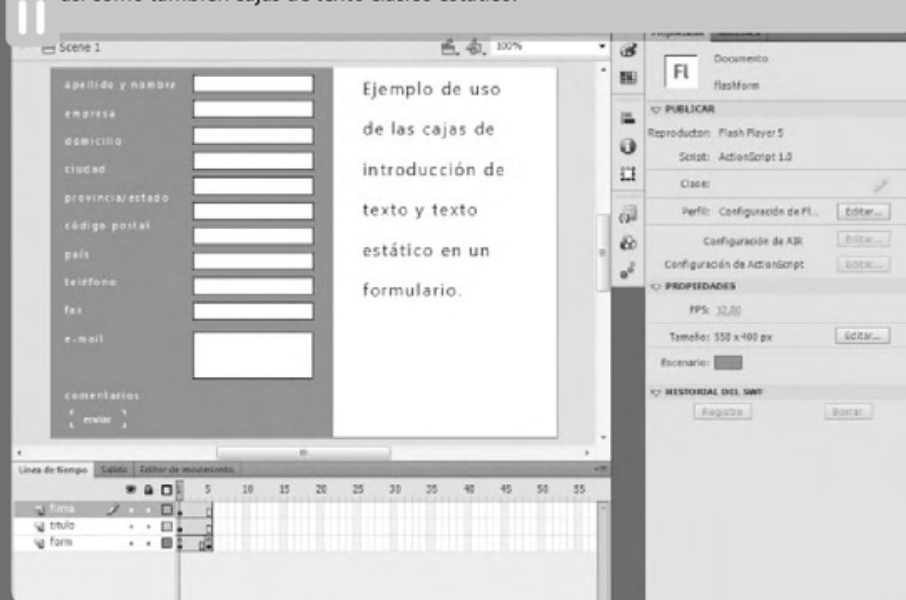
luego, el reproductor las busca en el sistema del usuario final. Tal como en los archivos HTML, primero busca la tipografía especificada y, si no la encuentra,

utiliza las similares del sistema. De todas maneras, es posible incorporar las fuentes al archivo, aunque esto aumentará su peso. ◀

Podemos convertir un texto en curva y editarlo como imagen, o solo separar los caracteres para realizar animaciones. Una vez en curvas, no lo podremos modificar.



Este formulario hecho en Flash tiene diversas clases de campos de introducción de texto, así como también cajas de texto clásico estático.



+ CONVERTIR EL TEXTO A CURVAS

Si queremos animar por separado las distintas letras de una palabra, otorgar a cada una de ellas un tipo de fuente diferente o convertir la palabra a líneas y rellenos, debemos seleccionar el campo de texto, ir al menú **[Modificar]** y allí elegir **[Separar]**. Veremos que cada carácter del texto seleccionado se convierte en un campo de texto independiente, que podemos editar de forma individual. Si repetimos la operación **[Modificar/Separar]**, el texto se convertirá en líneas y rellenos.

SUAVIZADO TIPOGRÁFICO



Existen distintas variantes para visualizar el texto clásico en Flash CS5. En este caso, analizaremos las de suavizado que tenemos disponibles para aplicar a los textos.

Flash, además de contar con una opción similar a la de los archivos HTML para renderizar la tipografía, también permite suavizar sus bordes con el fin de mejorar su legibilidad y presentación.

TIPOS DE SUAVIZADO

Existen varios tipos de suavizado disponibles. Cada uno posee sus características y usos ideales, veamos cuáles son:

- **Utilizar fuentes del dispositivo:** con esta opción, el archivo de Flash se visualiza con las tipografías que estén instaladas en el equipo cliente. Esta metodología no aumenta el peso del SWF, pero lo obliga a depender de las tipografías del equipo donde se lo abra. Si seleccionamos esta opción, debemos trabajar con fuentes clásicas.
- **Texto de mapa de bits [sin suavizado]:** aunque esta es una opción de suavizado, el suavizado del texto es desactivado. Los bor-



ATENCIÓN SUAVIZADOS

A la hora de trabajar con tipografías y suavizado, debemos tener en cuenta algunas recomendaciones:

- El suavizado de texto estático, dinámico y de introducción de texto será visible solo si se visualiza el archivo con un reproductor Flash versión 7 o superior. En caso contrario, solo se verá en el texto estático.
- Un texto pequeño se lee mejor en tipografías Sans-Serif (Arial, Helvética, Verdana). Los estilos manuscritos, los de cuerpo pesado y aquellos con Serif pueden tener una mala lectura en tamaños chicos. Las tipografías con Serif fueron creadas para leerse en papel; sus terminaciones son muy pequeñas y entran en conflicto con la representación en píxeles de las pantallas.
- Al publicar el archivo para Flash Player 8 o superior y tener elegido el **[Suavizado para legibilidad]** o **[Suavizado personalizado]**, este se activa en alta calidad.

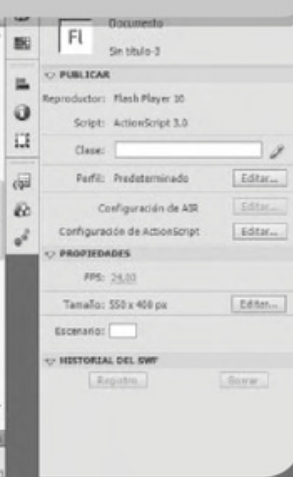
Flash CS5 brinda la posibilidad de usar diferentes tipos de suavizado, como vemos en la figura.

des de las tipografías se visualizan de manera nítida, y las fuentes usadas se incorporan al archivo SWF si no son las web estándar, lo que aumenta el peso del archivo. Esta opción muestra bien el texto si este mantiene el tamaño en el que se exportó, pero pierde calidad cuando se lo cambia de escala.

► **Suavizado para animación:** es el ideal para animar texto de manera fluida, ya que ignora los datos de alineación y ajuste entre los caracteres. En este caso, se incorporan los contornos de las fuentes, lo cual aumenta el peso del archivo SWF.

► **Suavizado para legibilidad:** es muy útil para los textos de tamaño pequeño. Incorpora al SWF los contornos de fuentes.

► **Suavizado personalizado:** esta elección nos permite cambiar dos propiedades de las fuentes: grosor y nitidez. Con la primera controlamos el ancho del suavizado (los caracteres parecen engrosarse), y con la segunda, regulamos si habrá más o menos sfumado en la transición entre los bordes de texto y el fondo de la imagen. ◀



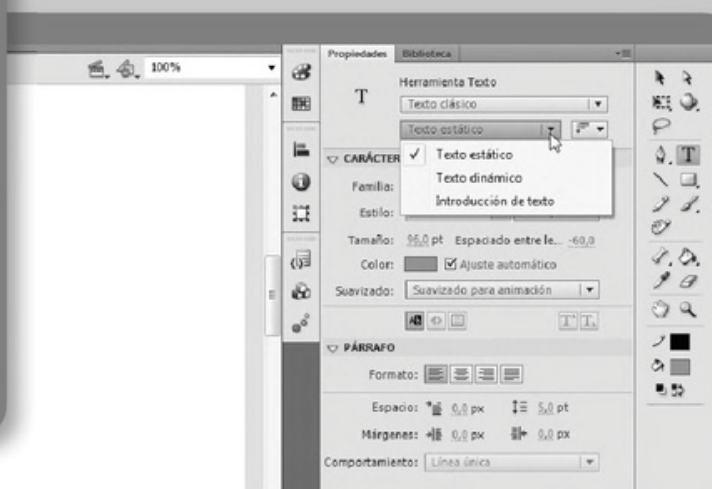
CREAR TEXTO

PASO A PASO

Veamos cómo desarrollar una caja de texto clásico en sus tres variantes mediante las propiedades de los paneles especiales.



01 MOTOR DE TEXTO CLÁSICO



TIPO DE TEXTO 02

Como ya mencionamos, el texto clásico no es la opción predefinida en Flash CS5, sino que hay que seleccionarlo especialmente para hacer uso de él. Con este tipo de texto podemos crear una máscara de recorte de otros objetos o generar una animación con efectos en la línea de tiempo. Además, podemos aplicar filtros de sombra paralela, esfumado y otras alternativas disponibles. ◀



PASO | 01 En la barra de herramientas seleccionamos **[Texto]**, que lleva el icono en forma de letra T. Luego, vamos al panel **[Propiedades]** y, en el desplegable **[Motor de texto]**, elegimos **[Texto clásico]**.

PASO | 02 En el mismo panel, elegimos la opción que corresponda al texto que vamos a utilizar: **[Texto estático]** para el que sea de solo lectura; **[Texto dinámico]** para texto actualizable con base de datos; o **[Introducción de texto]**, para campos de relleno en formularios. En este caso, elegimos la primera.

PASO | 03 Una vez elegido el tipo de texto que utilizaremos, comenzamos a determinar sus características, como fuente tipográfica, estilo, tamaño, color, suavizado que queremos aplicar, alineación que le daremos al párrafo, interlineado y orientación, entre otras posibilidades disponibles.

CLÁSICO

04 | DISPOSICIÓN DEL TEXTO

texto clásico en una sola línea co

texto clásico
con ancho de
caja fijo.

texto clásico en una sola línea continu

texto clásico
editado desde la paleta de
propiedades.

MODIFICAR TEXTO

05

03 | PROPIEDADES DEL TEXTO

PASO | 04 Antes de hacer clic sobre el escenario con la herramienta **[Texto]**, debemos tener en cuenta si queremos hacer una caja en una sola línea (que crezca de manera horizontal o vertical, según su orientación), de texto de ancho fijo (para texto horizontal) o de alto fijo (para texto vertical). Si preferimos, más adelante podemos cambiar la opción elegida.

PASO | 05 Escribimos el texto que queremos ingresar y, si lo deseamos, podemos volver a cambiar sus características desde las opciones del panel **[Propiedades]**. Para hacerlo, solo tenemos que seleccionar la caja de texto con el cursor correspondiente, hacer doble clic sobre el párrafo o un clic con la herramienta **[Texto]**, y seleccionar los caracteres que queremos cambiar. En este caso, elegimos algunas partes del texto y modificamos sus propiedades para diferenciarlo del resto. Como vemos, dentro de una misma caja pueden convivir varias familias y características tipográficas.

PASO | 06 A la caja de texto generada también podemos aplicarle distintos valores de escala, rotación y sesgado, entre otras opciones. Para hacerlo, usamos el panel **[Transformar]**, que activamos desde el menú **[Ventana/Transformar]**, o recurrimos a la herramienta **[Transformación libre]** de la barra.

RESUMEN

El texto clásico es la antigua forma en que Flash lo manejaba y se utilizó hasta la versión CS4 del programa, aunque podemos seguir usándolo sin problemas en la CS5. En la actualidad, la opción predefinida es **Text Layout Framework**, que veremos en las próximas páginas.

TEXT LAYOUT FRAMEWORK

Flash CS5 trae un nuevo motor de edición para marcos de texto. Gracias a él, podemos trabajar el texto como en los programas de maquetación editorial más conocidos, encadenando cajas, y generando columnas y medianiles, entre otras opciones.

Como ya adelantamos, Flash CS5 ofrece un nuevo motor de texto llamado **TLF (Text Layout Format)**. Este es el que se crea por defecto al seleccionar la herramienta **[Texto]**, y tiene una gama más amplia de funciones para el uso de texto enriquecido, y un control mucho más ajustado y variado de las propiedades del texto. Encontramos, entre otras, las siguientes posibilidades:

- ▶ Estilos de caracteres adicionales, como interlineado, ligaduras, subrayado, resaltado de texto, tachado, mayúsculas y minúsculas.
- ▶ Ajustes de párrafos con varias columnas, justificación de la última línea, espaciado entre párrafos, valores de línea de contorno, color de línea y color de relleno.
- ▶ Controles de atributos para textos asiáticos, como Tate Chu Yokoi, Mojikumi, Tipo Kinsoku Shori y la opción de modelo de interlineado.
- ▶ Las cajas de texto tienen la capacidad de enlazarse para hacer que el texto fluya entre ellas. Así, podemos maquetar una página en la que, al incorporar o borrar texto, este se vaya acomodando de manera automática en los distintos marcos.
- ▶ El texto TLF puede ser rotado en 3D, tener efectos de color y modos de mezcla, sin necesidad de convertirlo en un símbolo de clip de película.
- ▶ El motor nuevo tiene la capacidad de crear texto de derecha a izquierda para idiomas como el árabe y el hebreo, por ejemplo.

CONSIDERACIONES Y CARACTERÍSTICAS DE USO

A la hora de utilizar **Text Layout Framework**, debemos tener en cuenta una serie de factores que analizaremos en las próximas líneas:

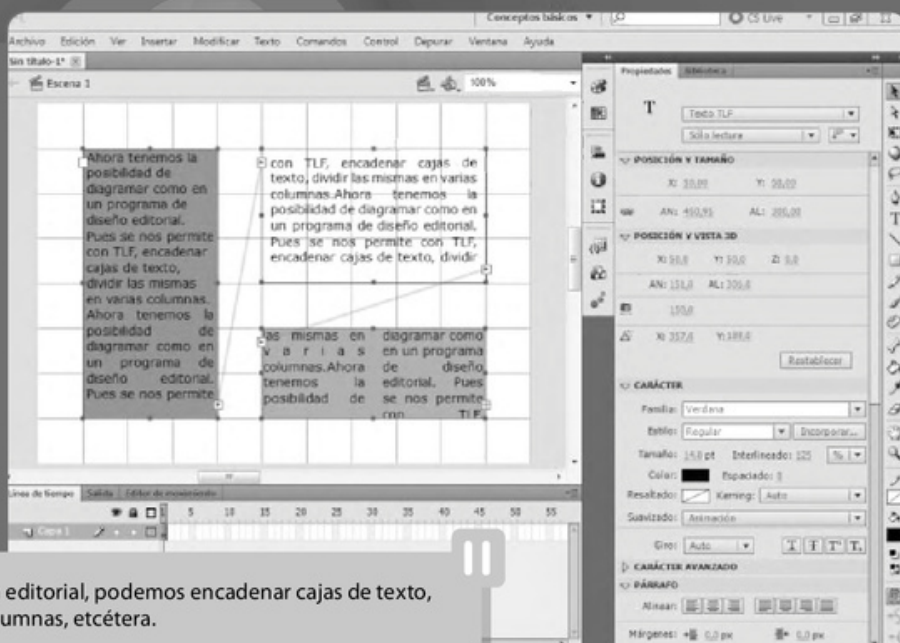
- ▶ Es preciso trabajar en un archivo Flash ActionScript 3 y publicarlo para Flash Player 10 como mínimo. Si trabajáramos en ActionScript 2, tendríamos que actualizarnos para hacer uso de las cualidades de TLF al usar cajas de texto.
- ▶ Existen dos tipos de contenedores de texto: texto de punto y texto de área. Estas opciones se generan al hacer clic en el escenario con la herramienta **[Texto]** (el de punto es el predeterminado y el de área se logra al hacer clic en el escenario y arrastrar el cursor hacia la derecha). En el primero, el tamaño varía según el texto que contenga;

en tanto que en el segundo, ese parámetro no depende del texto incorporado sino del que le asignemos en la sección **[POSICIÓN Y TAMAÑO]**, de la paleta de propiedades.

▶ TLF no admite tipografías Postscript Type 1 (antiguo formato de tipos digitales desarrollado por Adobe, hoy algo en desuso). Si podemos usar True Type y las nuevas OpenType (presentes en nuestro sistema operativo o descargables de manera gratuita desde la Web).

▶ A diferencia del texto clásico, el TLF no puede usarse como máscara.

▶ Al utilizar TLF, en el panel de propiedades de Flash tendremos tres opciones de visualización. En primer lugar, si está seleccionada la herramienta **[Texto]** y no hay ninguna selección en el escenario, se mostrará el **[Modo de Herramienta de texto]**. En cambio, si seleccionamos



Como en los programas de maquetación editorial, podemos encadenar cajas de texto, ponerles color de fondo, dividirlos en columnas, etcétera.



TLF permite aplicarle al texto efectos 3D, modos de mezcla y efectos de color, sin necesidad de convertirlo en un símbolo.

un bloque de texto, veremos el **[Modo de objeto de texto]**. Por último, al estar editando un bloque de texto, aparecerá el **[Modo de edición de texto]**. Estas diferentes alternativas pueden apreciarse en la paleta de propiedades a medida que vamos probando las diversas variantes de selección.

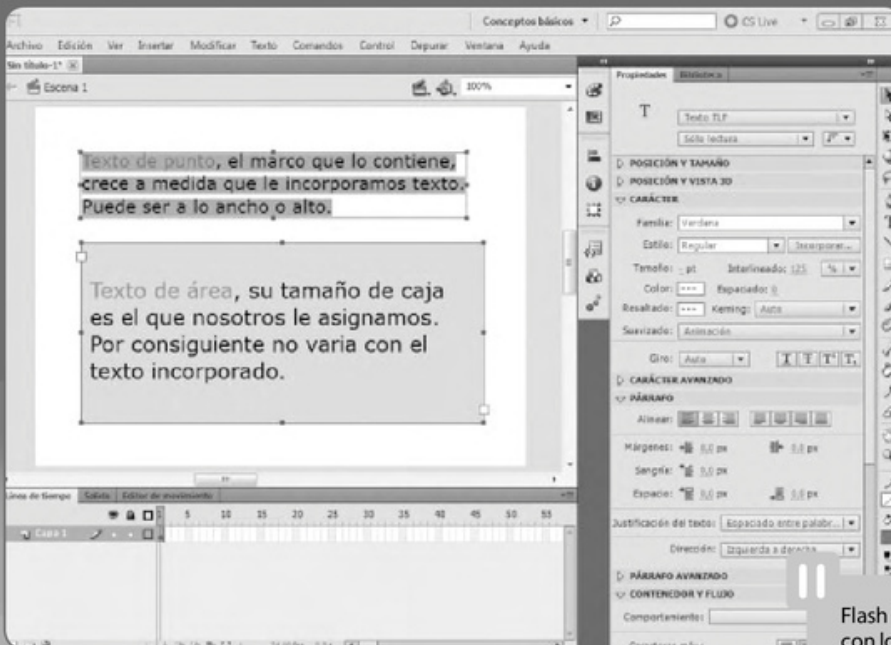
- Esta nueva forma de manejar el texto nos permite trabajar con estilos de caracteres y de párrafos.
- Basándonos en el comportamiento que tendrá el texto en tiempo de ejecución (cuando el usuario final está visualizando el archivo SWF), tenemos tres variantes de edición. Con **[Sólo lectura]**, cuando exportamos el archivo Flash para su visualización, el texto no se puede seleccionar ni editar. Si elegimos **[Seleccionable]**, cuando se publica el archivo SWF, el texto

Aquí podemos ver dos bloques de texto TLF editados de manera diferente por medio de las opciones **[CONTENEDOR Y FLUJO]** de la paleta de propiedades.

PARA USAR TLF, DEBEMOS TRABAJAR EN UN ARCHIVO FLASH ACTIONSCRIPT 3 Y PUBLICARLO PARA FLASH PLAYER 10 COMO MÍNIMO.

se puede seleccionar y copiar en el Portapapeles, pero no, editar (esta es la opción predeterminada de TLF). Con **[Editable]**, al visualizar la película en el reproductor, podemos seleccionar y editar el texto.

- LF necesita que una biblioteca de ActionScript específica esté disponible en Flash Player en el momento en el que se ejecuta. Si el usuario no la posee, el reproductor la descarga de la página de Adobe de manera automática.
- Tal como al trabajar con texto clásico, podemos generar un campo de texto con desplazamiento.
- Existen tres variantes de suavizado. La primera es la de fuentes del dispositivo, que, como en el caso del texto clásico, solo carga sus nombres para luego buscarlas en el sistema del usuario. La segunda es la de legibilidad, que resulta ideal si tenemos texto pequeño que deba ser leído pero no animado, y que requiera incorporar la fuente al archivo Flash para su correcta visualización. Por último, la opción para animación ignora los atributos de alineación y ajuste en las letras, para lograr una mejor fluidez en la animación, y también requiere la incorporación de los archivos tipográficos. ◀

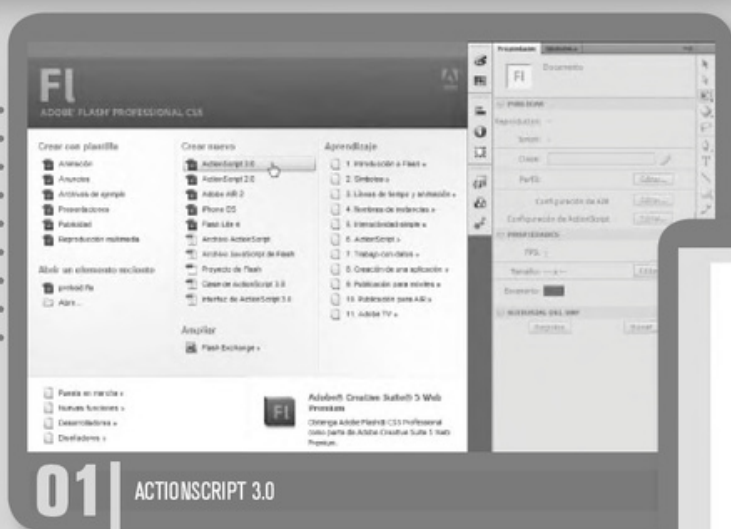


Flash CS5 utiliza dos maneras para trabajar con los bloques de texto en modo TLF: **Texto de punto** y **Texto de área**.

TLF: FLUJO ENTRE

PASO A PASO

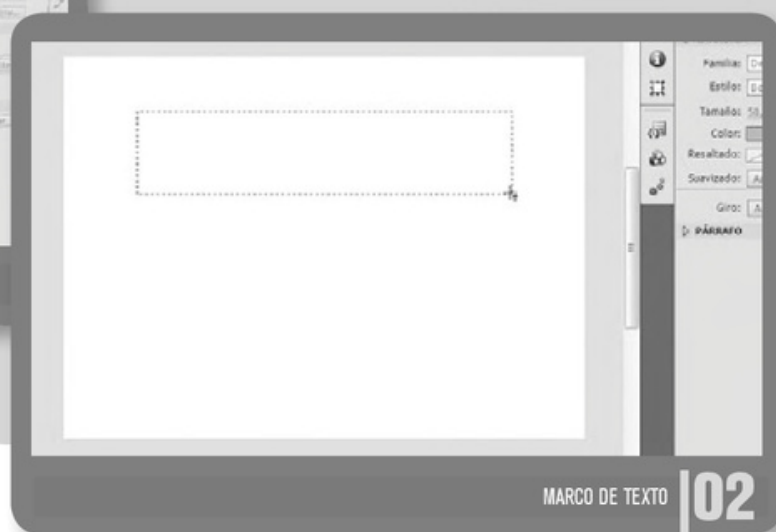
Usaremos una de las novedades de TLF (Text Layout Framework) para hacer recorrer el texto por distintos marcos.



01 | ACTIONSCRIPT 3.0

TLF tiene capacidades que no posee el texto clásico. Una de ellas es la posibilidad de entrelazar distintas cajas de texto y hacer que el texto vaya fluyendo (o continúe) entre ellas. Con este agregado, Flash incorpora características que solo tenían programas más específicos del ambiente editorial, y para usarlo, debemos acordarnos de trabajar en ActionScript 3.0.

PASO | 01 Vamos al menú **[Archivo/Nuevo...]**, seleccionamos **[ActionScript 3.0]** en la ventana de creación de documento y presionamos **[Aceptar]**. Si estábamos trabajando en un archivo ActionScript 2.0, debemos ir al menú **[Archivo/Configuración de Publicación...]** y, luego, a la solapa **[Flash]**. Allí, elegimos la opción **[Flash Player 10]** dentro de **[Reproductor]**, y **[ActionScript 3.0]** dentro de **[Script]**. Ahora podemos continuar, teniendo en cuenta que el cambio puede hacer que no funcionen los scripts en AS2.



MARCO DE TEXTO **02**

PASO | 02 En la barra de herramientas, seleccionamos **[Texto]**, que tiene el icono en forma de letra **T** imprenta mayúscula. Ponemos visible el panel **[Propiedades]** usando la combinación de teclas **<Ctrl + F3>**. Como por defecto aparece seleccionada la opción **[Texto TLF]**, simplemente hacemos clic en el escenario, arrastramos el cursor hasta darle la proporción deseada y lo soltamos.

PASO | 03 Ya tenemos nuestro primer bloque de texto; ahora podemos generar los siguientes de la misma manera. Una vez que terminamos de crear las distintas cajas, buscamos en ellas el **punto de entrada** y el **punto de salida**. Estos son los cuadrados blancos ubicados arriba a la izquierda y abajo a la derecha del marco, que, según la dirección del texto (izquierda a derecha, o viceversa), actúan como entrada o salida de párrafo.

TEXTOS

03 | ENTRADA Y SALIDA

Los párrafos ahora están enlazados, podemos observar las líneas que los unen... También si llenamos los mismos de texto observamos como se trasladan de uno al otro sin problemas. Sólo cuando se nos termina la capacidad de los márcos generados podemos apreciar en el punto de salida final un símbolo + en rojo que

FLUJO DE TEXTO | 05

Los párrafos ahora están enlazados, podemos observar las líneas que los unen... También si llenamos los mismos de texto observamos como se trasladan de uno al otro sin problemas. Sólo cuando se nos termina la capacidad de los márcos generados podemos apreciar en el punto de salida final un símbolo + en rojo que

04 | VINCULAR CAJAS

PASO | 04 Para vincular los distintos párrafos, como primer paso hacemos clic en el punto de salida de la caja inicial.

El puntero del mouse se convertirá en un icono de texto lleno. A continuación, hacemos clic sobre el punto de entrada de la caja siguiente y, de este modo, vinculamos los párrafos. Podemos seguir haciéndolo hasta tener el flujo de texto completo.

PASO | 05 Al empezar a escribir, observaremos que, cuando se llena la primera caja, el texto fluye hacia la segunda, y así hasta la última que esté vinculada. Si cuando terminamos de introducir el párrafo vemos que en el último de los contenedores el punto de salida tiene un símbolo **[+]** en rojo, significará que aún hay texto desbordando de ella. Entonces, tenemos dos opciones: una es cambiar el formato del texto para que encaje y no desborde, y la otra es hacer clic en el icono **[+]** y generar una nueva caja.

PASO | 06 Para desvincular alguno de los contenedores, hacemos doble clic en su punto de entrada o de salida.

Las cajas de texto pueden editarse en cantidad de columnas, relleno (padding), colores de borde y de relleno del contenedor, y dirección de lectura, entre otras posibilidades presentes en el panel **[CONTENEDOR Y FLUJO]**.

RESUMEN

Como pudimos ver, esta nueva herramienta de texto es muy completa en cuanto al manejo de párrafos, la vinculación entre ellos y las distintas posibilidades de formato que brindan los paneles. Además, tenemos la posibilidad de hacer que el texto se cargue de manera externa por medio de ActionScript y XML.

TEXTO EN FORMATO HTML

Aunque la clase `TextFormat` permite aplicar formato a textos cortos fácilmente, en proyectos con mucho texto y varios estilos, es mejor recurrir a otra opción.

En sitios que tienen gran cantidad de texto, lo ideal es utilizar formato **HTML** y **CSS** para aplicarlos globalmente a todo el proyecto. De esta manera, podremos tener el control de los estilos de todos los textos del sitio web en un solo archivo CSS, que contendrá los formatos necesarios para el diseño. Entonces, todo cambio de estilo que debamos realizar podrá hacerse sin inconvenientes y sin siquiera recurrir a Flash, dado que un archivo CSS puede modificarse con cualquier editor de texto.

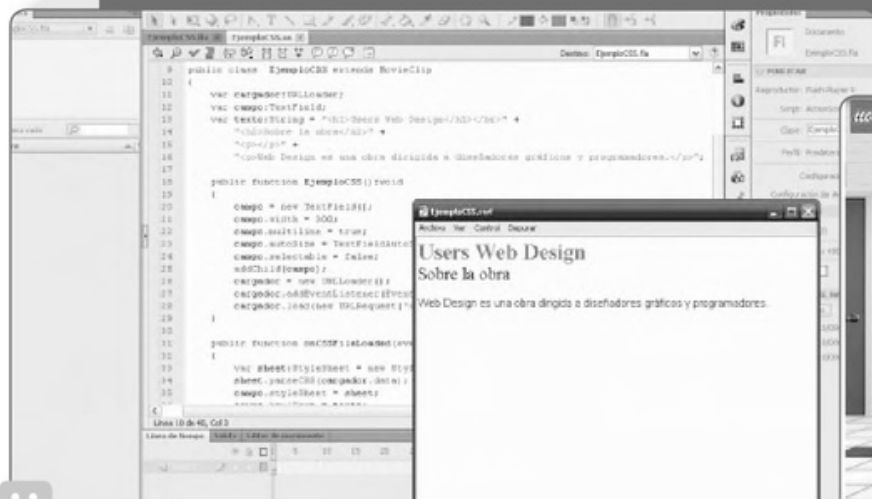
FORMATO LIMITADO

Flash soporta una versión limitada de etiquetas HTML y de propiedades CSS, que podemos ver en las tablas correspondientes a cada una de estas categorías.

CÓDIGO 1

```
var miTxt:TextField = new TextField();
miTxt.x = miTxt.y = 20;
miTxt.width = 500;
miTxt.multiline = true;
miTxt.wordWrap = true;
miTxt.autoSize = TextFieldAutoSize.LEFT;
miTxt.styleSheet = nuevoCSS;
miTxt.htmlText = "<body>";
miTxt.htmlText += "<span class='cabecal'>Como aplicar CSS a textos HTML</span><br>";
miTxt.htmlText += "<span class='autor'>Por RedUSERS</span><br><br>";
miTxt.htmlText += "<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>";
miTxt.htmlText += "</body>";
addChild(miTxt);
```

En www.funkybrownz.co.uk, algunas secciones cargan textos HTML que toman sus estilos a partir de CSS.



Al publicar el documento SWF, podemos ver el resultado de haber formateado el texto HTML con CSS. Esta forma es ideal para textos que requieren muchos estilos.



APLICAR FORMATO HTML

Para aplicar formato HTML en un campo de texto, debemos utilizar la propiedad **[htmlText]** en vez de **[text]**. Por ejemplo, el siguiente código pone en negrita la palabra **[RedUSERS]** de un campo de texto:

```
campoTexto.htmlText =
"<b>RedUSERS</b>";
```

APLICAR FORMATO CON CSS

Para confeccionar una hoja de estilos dentro de Flash, debemos crear una instancia de la clase **[StyleSheet]** y, luego, agregarle un objeto de estilo por cada etiqueta o clase necesaria. Las propiedades CSS se agregan como objetos personalizados para cada etiqueta o clase y, una vez creados, se utiliza el método **[setStyle()]** para añadirlos a la hoja de estilos y asociarlos con su respectiva etiqueta o clase.

A continuación, analizaremos el código para definir una hoja de estilos. Entre las líneas 1 y 23 creamos una nueva instancia de la clase **[StyleSheet]**, para luego definir la etiqueta **[body]**, las clases **[cabezal]** y **[autor]**, y sus respectivas propiedades. Finalmente, agregamos a la hoja de estilos la etiqueta y las clases generadas. Podemos ver un ejemplo de cómo aplicar esto en el código 1. ◀



ETIQUETAS HTML SOPORTADAS POR FLASH

La cantidad de etiquetas HTML que soporta Flash es limitada, pero resulta suficiente para trabajar con texto.

ETIQUETA	DESCRIPCIÓN
▶ 	▶ Soporta los atributos [color] , [face] y [size] .
▶ 	▶ Para que funcione esta etiqueta, debe existir la versión negrita de la fuente elegida.
▶ <i>	▶ Para que funcione esta etiqueta, debe existir la versión itálica de la fuente elegida.
▶ <u>	▶ Subraya el texto.
▶ 	▶ Soporta el atributo [class] .
▶
	▶ El campo de texto debe tener habilitada la opción [Multilínea] para funcionar.
▶ <p>	▶ Requiere habilitar [Multilínea] , soporta los atributos [align] y [class] .
▶ 	▶ Todas las listas tienen viñetas, los códigos de numeración son ignorados.
▶ 	▶ Soporta los atributos: [src] , [width] , [height] , [align] , [hspace] , [vspace] y [id] ; puede incrustar imágenes externas (JPG, GIF, PNG) y archivos SWF.
▶ <a>	▶ Soporta los atributos [href] , [event] y [target] .



PROPIEDADES CSS SOPORTADAS POR FLASH

Las propiedades CSS soportadas por Flash son pocas, pero permiten aplicar una interesante variedad de estilos.

PROPIEDAD CSS	DESCRIPCIÓN
▶ color Font	▶ El formato de color debe ser 0xRRGGBB.
▶ display	▶ Los valores incluyen [none] , [block] e [inline] .
▶ font-family	▶ Nombre de la fuente.
▶ font-size	▶ Tamaño de la fuente en píxeles.
▶ font-style	▶ Estilo de la fuente; incluye los valores [italic] y [normal] .
▶ font-weight	▶ Estilo de la fuente; incluye los valores [bold] y [normal] .
▶ kerling	▶ Activa o desactiva el interletrado; incluye los valores [true] y [false] .
▶ letter-spacing	▶ Interletrado especificado en píxeles.
▶ margin-left	▶ Margen izquierdo especificado en píxeles.
▶ margin-right	▶ Margen derecho especificado en píxeles.
▶ text-align	▶ Incluye los valores [left] , [right] , [center] y [justify] .
▶ text-decoration	▶ Subraya el texto; incluye los valores [underline] y [none] .
▶ text-indent	▶ Indentado de la primera línea del párrafo, especificado en píxeles.



El sitio de Lego (www.legoclick.com) también incorpora texto HTML al que se le dió formato mediante CSS.

APLICAR CSS A UN CAMPO DE TEXTO

PASO A PASO

En este paso a paso, explicaremos cómo aplicar formato CSS a un texto HTML desde una hoja de estilos externa.

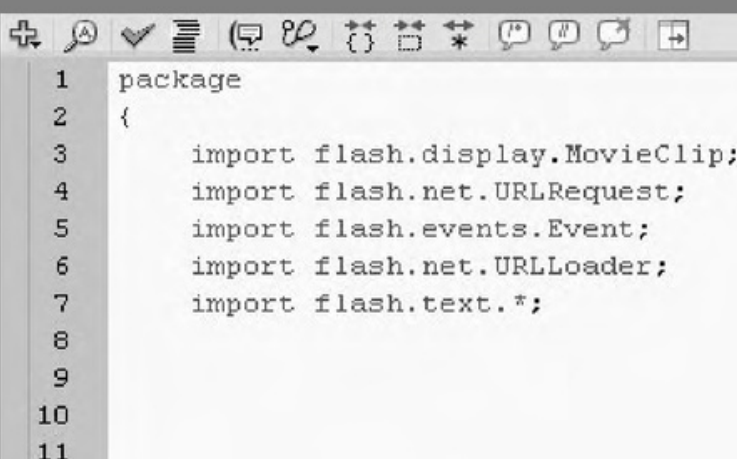


```

1 * CSS Document */
2 h1 {
3     font-family: Times New Roman, Times, _serif;
4     font-size: 30;
5     font-weight: bold;
6     color: #ff0000;
7 }
8 h2 {
9     font-family: Times New Roman, Times, _serif;
10    font-size: 24;
11 }
12 p {
13     font-family: Arial, Helvetica, _sans;
14     font-size: 15;
15 }
16 a {
17     color: #0000ff;
18 }
19 a:visited {
20     color: #ff0000;
21 }

```

01 HOJA DE ESTILOS



```

1 package
2 {
3     import flash.display.MovieClip;
4     import flash.net.URLRequest;
5     import flash.events.Event;
6     import flash.net.URLLoader;
7     import flash.text.*;
8
9
10
11

```

IMPORTAR CLASES | 02

Como mencionamos, en el caso de un sitio con varias secciones, uno de los beneficios de tener el contenido de texto separado de su formato es que podremos modificar rápidamente el estilo de los distintos niveles de encabezados, texto de párrafo y links cuando sea necesario. Para el siguiente ejemplo, necesitaremos, además de Adobe Flash, un editor de código como Adobe Dreamweaver para escribir el código CSS. ◀



PASO | 01 Para empezar, creamos un archivo **ejemplo.css** para la hoja de estilos, con las clases y etiquetas necesarias.

En nuestro ejemplo, vamos a definir propiedades para los encabezados **<h1>** y **<h2>**, para el texto de párrafo **<p>** y para la etiqueta **<a>** de links.

PASO | 02 Creamos en Flash un nuevo archivo ActionScript, donde escribimos todo el código AS para hacer funcionar el ejemplo, y lo grabamos con el nombre **EjemploCSS.as**. En él, lo primero que debemos hacer es escribir el código para importar las clases de Flash que vamos a necesitar en la carga externa del archivo de CSS, la creación del campo de texto y el seteo de sus propiedades, para finalmente mostrarlo en pantalla.

PASO | 03 A continuación, comenzamos a escribir la clase, que debe tener el mismo nombre

DINÁMICO

04 FUNCIÓN CONSTRUCTORA

```

17
18     public function EjemploCSS():void
19     {
20         campoTxt = new TextField();
21         campoTxt.width = 300;
22         campoTxt.multiline = true;
23         campoTxt.autoSize = TextFieldAutoSize.LEFT;
24         campoTxt.selectable = false;
25         addChild(campoTxt);
26         cargadorCSS = new URLLoader();
27         cargadorCSS.addEventListener(Event.COMPLETE, onCSSFileLoaded);
28         cargadorCSS.load(new URLRequest("ejemplo.css"));
29     }
30
31
32
33
34
35
36

```

```

public function onCSSFileLoaded(event:Event):void
{
    var estiloCSS:StyleSheet = new StyleSheet();
    estiloCSS.parseCSS(cargadorCSS.data);
    campoTxt.styleSheet = estiloCSS;
    campoTxt.htmlText = txtEjemplo;
}

```

APLICAR ESTILO

05

03 OBJETOS

```

public class EjemploCSS extends MovieClip
{
    var cargadorCSS:URLLoader;
    var campoTxt:TextField;
    var txtEjemplo:String = "<h1>Users Web Design</h1><br>" +
        "<h2>Sobre la obra</h2><br>" +
        "<p>Web Design es una obra dirigida a diseñadores gráficos y programadores</p>" +
        "<p>Para conocer más puedes visitar <a href='http://redusers.com/cursos/webdesign/'>el sitio</a>.</p>";
}

```

ESTILOS APLICADOS

06

que el archivo, es decir, **[EjemploCSS]**. Primero declaramos los objetos que vamos a utilizar: el cargador para el archivo CSS, el campo de texto dinámico y el texto en sí mismo.

PASO | 04 Definimos la función constructora, cuyo nombre es el mismo que el de la clase. Dentro de ella creamos una instancia del objeto definido para el campo de texto, le establecemos algunas propiedades y lo agregamos a la **display list**. Luego, generamos una instancia del objeto que se ocupa de cargar el archivo de estilos externo.

PASO | 05 En cuanto el archivo **ejemplo.css** ha terminado de cargar, se llama a la función que aplica el estilo y el contenido al campo de texto dinámico. Para que el estilo sea tomado en cuenta, es fundamental aplicarlo al campo de texto dinámico antes de cargar el texto HTML en él.

PASO | 06 Para ver el resultado, a través del menú **[Archivo/Nuevo...]** de Flash creamos un documento ActionScript 3.0 y luego, vamos a **[Archivo/Configuración de publicación...]**. En la solapa **[Flash]**, hacemos clic sobre el botón **[Configuración...]**, ubicado a la derecha del desplegable **[Script]**. En el campo **[Clase de documento]**, escribimos **[EjemploCSS]** y presionamos **[Aceptar]** hasta salir completamente del cuadro de diálogo. Para ver el resultado, publicamos la película desde **[Archivo/Vista previa de publicación/Flash]**.

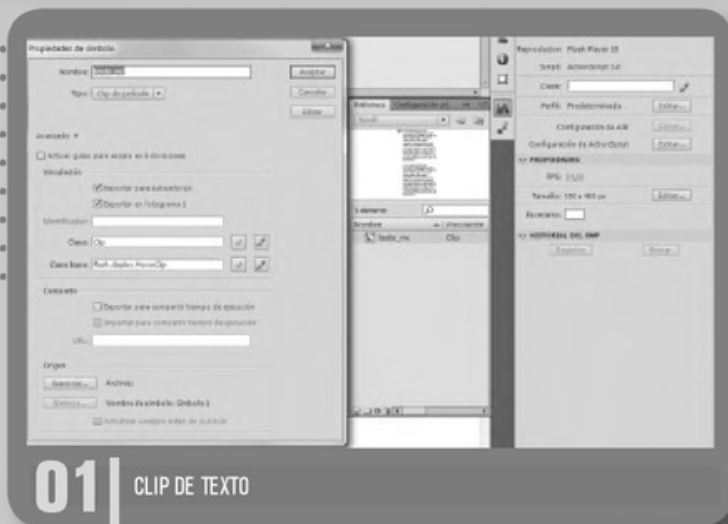
RESUMEN

Una vez que tenemos definido el archivo externo con los estilos y el código ActionScript, ya estamos en condiciones de modificar el aspecto que tendrán los textos del sitio, de manera simple y centralizada.

CREAR UNA SCROLL BAR CON

PASO A PASO

A continuación, explicaremos cómo crear una barra de scroll para textos largos, con efecto de aceleración (easing).



01 CLIP DE TEXTO



IMPORTAR Y DEFINIR CLASES

02

En muchos proyectos web, habrá textos tan largos, que no entrarán en el espacio que tienen destinado.

La única manera de mostrarlos consistirá en enmascararlos para que se vea una cantidad limitada de renglones y, con la ayuda de una **barra de scroll**, dejar que el usuario pueda recorrer el texto completo, oculto por fuera de esa máscara. Para lograr esto, usaremos un clip de película que contenga el texto y código ActionScript para controlar la funcionalidad de scroll. ◀

PASO | 01 Creamos el cuadro de texto estático con su contenido y presionamos **<F8>** para convertirlo en símbolo tipo **[Clip de película]**. Accedemos a la Biblioteca con **<Ctrl + L>**, presionamos el botón derecho sobre el icono del símbolo y elegimos **[Propiedades...]**. Hacemos clic en **[Avanzado]**

para desplegar más opciones y, en la sección **[Vinculación]**, marcamos **[Exportar para ActionScript]**. En el campo **[Clase]** escribimos **[Clip]**, que es el nombre por el cual lo llamaremos desde el código AS, y pulsamos **[Aceptar]**.

PASO | 02 Desde **[Archivo/Nuevo...]**, creamos un archivo de ActionScript y lo grabamos como **Scroll.as**. En él escribimos el código para importar las clases de Flash que necesitaremos para los eventos del mouse, el dibujo de la barra de scroll y mostrar todo en pantalla. Luego, escribimos la clase que extenderá a la clase **Sprite**, declaramos los objetos que vamos a utilizar, las medidas de alto y ancho, los datos de separación y recorrido, y el objeto para el texto en sí.


```
var ancho:Number;
var recorrido:Number;
var clipTexto:MovieClip;

public function Scroll(mc:MovieClip,posx:Number,posy:Number,anc:Number,alt:Number,color1:uint,color2:uint,color3:uint):void {
    clipTexto=mc;
    alto=alt;
    ancho=anc;
    recorrido=7*alto/8;
    x=clipTexto.x-anchos-separacion;
    y=clipTexto.y;
}
```

EASING

04 BARRA DE SCROLL

```
var separacion:Number=10;
var ancho:Number;
var recorrido:Number;
var clipTexto:MovieClip;

public function Scroll(mc:MovieClip,posx:Number,posy:Number,anc:Number,alt:Number,color1:uint,color2:uint,color3:uint):void {
    clipTexto=mc;
    alto=alt;
    ancho=anc;
    recorrido=7*alto/8;
    x=clipTexto.x-anchos-separacion;
    y=clipTexto.y;
    dibujoCuad("baseScroll_mc", ancho, alto, color1, color3);
    dibujoCuad("slider_mc", ancho, alto/8, color2, color3);
    enmascaro(clipTexto);
}
```

SCROLL Y ENMASCARADO

05

03 PERSONALIZAR SCROLL

```
package {
    import flash.display.*;
    import flash.events.*;
    import flash.geom.Rectangle;

    public class Scroll extends Sprite {
        var alto:Number;
        var separacion:Number=10;
        var ancho:Number;
        var recorrido:Number;
        var clipTexto:MovieClip;

        public function Scroll(mc:MovieClip, posX:Number, posY:Number, anc:Number, alt:Number, color1:uint, color2:uint, color3:uint):void {
            clipTexto=mc;
            alto=alt;
            ancho=anc;
            recorrido=7*alto/8;
            x=clipTexto.x-anchos-separacion;
            y=clipTexto.y;
            dibujoCuad("baseScroll_mc", ancho, alto, color1, color3);
            dibujoCuad("slider_mc", ancho, alto/8, color2, color3);
            enmascaro(clipTexto);
            getChildByName("slider_mc").addEventListener(MouseEvent.CLICK, arrastroSlider);
            getChildByName("slider_mc").addEventListener(MouseEvent.CLICK, sueltoSlider);
            getChildByName("slider_mc").addEventListener(MouseEvent.CLICK, sueltoSlider);
            addEventListener(Event.ENTER_FRAME, ubicoSlider);
        }
    }
}
```

LISTENERS

06

```
import flash.display.*;
import flash.events.*;
import flash.geom.Rectangle;

public class Scroll extends Sprite {
    var alto:Number;
    var separacion:Number=10;
    var ancho:Number;
    var recorrido:Number;
    var clipTexto:MovieClip;

    public function Scroll(mc:MovieClip, posX:Number, posY:Number, anc:Number, alt:Number, color1:uint, color2:uint, color3:uint):void {
        clipTexto=mc;
        alto=alt;
        ancho=anc;
        recorrido=7*alto/8;
        x=clipTexto.x-anchos-separacion;
        y=clipTexto.y;
        dibujoCuad("baseScroll_mc", ancho, alto, color1, color3);
        dibujoCuad("slider_mc", ancho, alto/8, color2, color3);
        enmascaro(clipTexto);
        getChildByName("slider_mc").addEventListener(MouseEvent.CLICK, arrastroSlider);
        getChildByName("slider_mc").addEventListener(MouseEvent.CLICK, sueltoSlider);
        getChildByName("slider_mc").addEventListener(MouseEvent.CLICK, sueltoSlider);
        addEventListener(Event.ENTER_FRAME, ubicoSlider);
    }
}
```

PASO 03

A continuación, establecemos la función constructora **[Scroll]**, con los parámetros para definir el desplazamiento: nombre del clip que contiene el texto, posición X e Y del scroll, ancho y alto del scroll (este último será el alto de la máscara), color de la base del scroll, color del slider y color de los bordes.

PASO 04

Lo primero que hace la función **[Scroll]** cuando se crea una instancia de esta clase es asignar, a los objetos creados previamente, los valores recibidos de los parámetros. Es acá donde se definen los valores para el ancho y el alto de la barra y su ubicación.

PASO 05

Luego, llamamos a la función de dibujo para crear la base del scroll y el slider, pasándole los parámetros de ancho, alto y colores que se asignaron al momento de generar la instancia. A continuación, llamamos a la función encargada de enmascarar el clip que contiene al texto.

PASO 06

Le asignamos al slider los listeners que harán las llamadas a las funciones respectivas cuando se presione o se suelte el puntero del mouse (y también en caso de **rollOut**) y el listener que reconocerá el movimiento del slider para reubicar el texto.

PASO A PASO



07 DIBUJAR RECTÁNGULOS

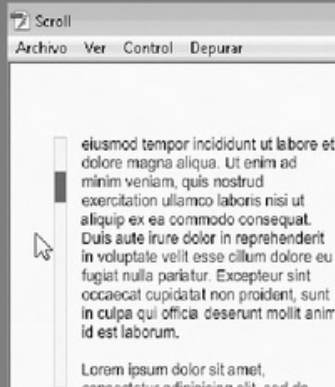


eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do

FUNCIÓN MOUSE_DOWN

08



```
private function dibujoCua(nombre:String, ancho:Number, alto:Number, color:uint, colorborde:uint):void {
    var cuad:Sprite = new Sprite();
    cuad.name = nombre;
    cuad.graphics.beginFill(color);
    cuad.graphics.lineStyle(0.1, colorborde);
    cuad.graphics.drawRect(0, 0, ancho, alto);
    cuad.graphics.endFill();
    addChild(cuad);
}
```

09 FUNCIÓN MOUSE_UP

PASO | 07 Creamos una función llamada **[dibujoCua]**, que se encargará de dibujar la base y el slider del scroll. Los parámetros que debe recibir son el nombre del **Sprite** por crear, el ancho y el alto del rectángulo, el color de relleno y el borde.

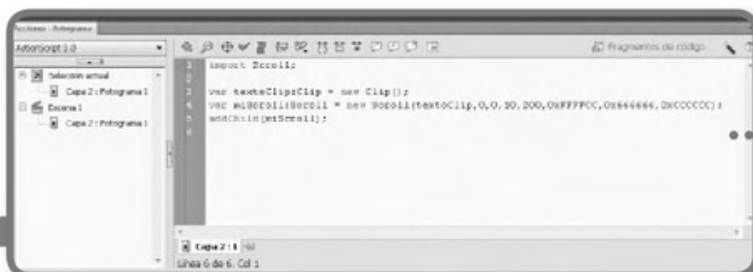
PASO | 08 Para detectar cuándo se arrastra el slider, generamos la función **[arrastradoSlider]**, que crea un objeto **Sprite** igual al clip objetivo del evento que disparó la función, y luego activa la posibilidad de arrastrar el slider.

PASO | 09 Cuando soltamos el slider, invocamos la función **[sueltoSlider]**, que desactiva la posibilidad de arrastrarlo con el puntero del mouse.

PASO | 10 Otra función necesaria es la que se ocupa de enmascarar el texto para dejar a la vista tantos renglones como consideremos necesarios. Ésta se llama **[enmascaro]** y, como parámetro, debemos pasarle el nombre del clip de película al que vamos a aplicar la máscara, es decir, el clip que contiene el texto.

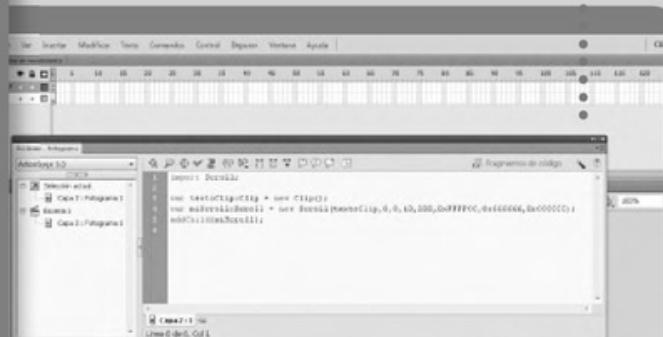
PASO | 11 Para reubicar el texto, usaremos la función **[ubicoSlider]**, que estará atenta al movimiento del slide para calcular cuánto desplazar el texto en función de la distancia que éste fue arrastrado.

PASO | 12 Antes de probar el ejemplo, volvemos al archivo de Flash que contiene el texto y ponemos, en el primer cuadro de la línea de tiempo, el siguiente código:



```
private function enmascaro(mc:MovieClip):void {
    var mascara:Sprite = new Sprite();
    mascara.graphics.beginFill(0x000000);
    mascara.graphics.lineStyle(1, 0x000000);
    mascara.graphics.drawRect(0, 0, mc.width, alto);
    mascara.graphics.endFill();
    mascara.x=separacion+ancho;
    addChild(mc);
    mc.mask=mascara;
    addChild(mascara);
}
```

10 ENMASCARAR TEXTO



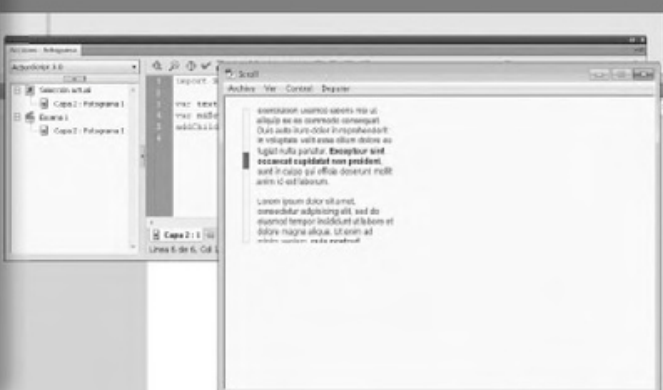
LÍNEA DE TIEMPO 12

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

11 REUBICAR EL TEXTO



PRUEBA BARRA DE SCROLL 13

```
import Scroll;
```

```
var textoClip:Clip = new Clip();
var miScroll:Scroll = new Scroll(textoClip, 0, 0, 10, 200, 0xFFFFCC, 0x666666, 0xCCCCCC);
addChild(miScroll);
```

PASO 13 Para probar el funcionamiento de la barra de scroll, guardamos el archivo Flash como **Scroll.fla** y lo publicamos a través del menú [Archivo/Vista previa de publicación/Flash].

RESUMEN

Como podemos ver, con muy pocos pasos es posible crear una barra de scroll con aceleración que podemos personalizar muy fácilmente, cambiando su ubicación, tamaño y colores.

Como nuestra barra está desarrollada con AS, podemos utilizar todas las instancias que necesitemos con solo crear los objetos [Scroll] correspondientes y pasarle a cada uno distintos clips para mover.

Esta barra no solo nos permite mover texto; si el clip de película que le pasamos por parámetro contiene otro tipo de elemento (una foto, una imagen o una animación), también lo desplazará sin inconvenientes.

ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué es el texto clásico en Flash?

02 ¿Cuál es el formato de texto predefinido en Flash?

03 ¿Para qué sirve el suavizado tipográfico?

04 ¿Qué tipos de suavizado de texto podemos utilizar?

05 ¿Qué es **Text Layout Framework**?

06 ¿Para qué sirve la clase **TextFormat**?

07 ¿Qué etiquetas HTML admite Flash?

08 ¿Qué propiedades CSS son soportadas por Flash?

09 ¿Para qué sirve una scroll bar?

10 ¿Cómo podemos crear una scroll bar con **Easing**?

PRÁCTICAS

01 Trabaje con texto clásico en Flash.

02 Utilice algún suavizado en el texto.

03 Siga las instrucciones del paso a paso para crear un texto.

04 Trabaje con texto en formato HTML.

05 Aplique CSS a un campo de texto dinámico.

Dibujos y loaders

- ▶ EL DIBUJO DE FORMAS
- ▶ DIBUJO EN FLASH
- ▶ LA HERRAMIENTA DECÓ
- ▶ ENTORNOS 3D
- ▶ INTROS EN FLASH
- ▶ LAS ALTERNATIVAS A LAS INTROS
- ▶ CARGADORES PARA SITIOS EN FLASH
- ▶ CREACIÓN DE UN CARGADOR SIMPLE
- ▶ OPTIMIZACIÓN DE SITIOS



EL DIBUJO DE FORMAS

Las herramientas de dibujo permiten crear y modificar líneas y formas. Estos elementos son gráficos vectoriales ligeros, lo que mantiene reducido el tamaño del archivo FLA.

Flash está equipado con una serie de herramientas de dibujo parecidas a las de los principales programas de dibujo vectorial.

Los gráficos vectoriales describen las imágenes mediante líneas y curvas, que incluyen propiedades de color y posición. Permiten realizar gran cantidad de transformaciones y se escalan sin pérdida de calidad. Aquí veremos cómo emplear cada herramienta de dibujo.

HERRAMIENTAS BÁSICAS

La paleta de **[Herramientas]** contiene todos los elementos necesarios para el trabajo. Algunas llevan asociados modificadores que permiten cambiar propiedades y pueden visualizarse al final del panel.

La **[Selección]** (flecha) es la herramienta más utilizada, ya que permite seleccionar objetos cuadrangulares, bordes de objetos

LA HERRAMIENTA [PLUMA] PERMITE REALIZAR TRAZADOS PRECISOS, COMO LÍNEAS RECTAS O CURVAS SUAVES.

(doble clic) y rellenos (clic) a nuestro gusto. Es posible crear formas por medio de trazos a mano alzada con el **[Lápiz]** y el **[Pincel]**, en tanto que con **[Líneas]** podemos trazar rectas, para luego modificarlas. El **[Pincel rociador]** se emplea para "pincelar" un patrón de formas en el escenario de una sola vez. **[Cubo de pintura]** aplica rellenos a los objetos siempre que el borde esté delimitado. El **[Borrador]** funciona como el **[Pincel]**, pero elimina todo por donde va pasando.

HERRAMIENTAS AVANZADAS

La herramienta **[Lazo]** es complementaria a la **[Flecha]**, ya que puede seleccionar cualquier elemento sin importar su forma, aunque a diferencia de ella, no es capaz de seleccionar rellenos u objetos si no es de manera manual. Al elegirla, aparecen la **[Varita Mágica]**, que selecciona según los colores de los objetos; y la selección de polígonos.

La **[Pluma]** permite realizar trazados precisos, como líneas rectas o curvas suaves. Al dibujar con ella, con un clic creamos puntos en los segmentos de las líneas rectas que podemos arrastrar. Es posible ajustar los segmentos de las curvas y las rectas ajustando los puntos de las líneas. Los contornos o filetes pueden ser más o menos gruesos y tener formas determinadas. Así, es posible crear dibujos lineales puros, como los técnicos, o efectuar trabajos más artísticos, con un contorno caligráfico.

El sitio www.MarcEcko.com es un buen ejemplo del uso y las posibilidades que brinda Flash.



Excelente ejemplo de dibujo a mano alzada, explosión de colores y multitud de recursos en Flash.



La herramienta de **[Subselección]** complementa a la **[Pluma]** porque permite mover o ajustar los vértices que componen los objetos creados con ella. El **[Bote de Tinta]** se utiliza para cambiar rápidamente el color de un trazo; se aplica sobre objetos que tienen borde. Por su parte, el **[Cuentagotas]** funciona capturando colores para usar en otro lugar. Algunas herramientas tienen opciones especiales, a las que se accede estando en la línea u objeto dibujado, desde un submenú que indica, por ejemplo, **[Ajustar a Objetos]**, para encajar unos con otros; **[Suavizar]**, para quitar rigidez a trazos rectos; o **[Enderezar]**, para rectificar trazos redondeados.

PANELES DE COLOR Y MUESTRAS

Desde el panel **[Color]** podemos elegir el que más nos agrade: bastará con hacer clic en las pestañas de las herramientas **[Lápiz]** (para el tono de borde) y **[Bote de Pintura]** (para el del relleno). El panel muestra todos los tonos disponibles y permite introducir el código de cada uno. Es posible determinar el tipo de relleno mediante las opciones de sólido, degradado lineal o radial, o mapa de bits, que permite colocar una imagen como relleno. El panel **[Muestras]** se usa para ver de un modo rápido y claro los colores de que disponemos, tanto sólidos (un solo color) como degradados (lineales o radiales). Además, cuando creemos un color, podremos añadirlo a nuestro conjunto de muestras mediante **[Agregar Muestra]**.

TIPS DE DIBUJO

Algunos consejos prácticos a la hora de dibujar en Flash son los siguientes:

CONOCIMIENTOS TÉCNICOS

Recordemos que, para mejorar y potenciar nuestro aprendizaje, desde www.redusers.com/cursos/webdesign podemos descargar los archivos editables de los proyectos que realizamos en esta colección. A partir de ellos podremos obtener ideas para un proyecto propio o, directamente, modificar los archivos para reutilizarlos. En el sitio también encontraremos **guías visuales**, **glosario de términos** y una completa guía de preguntas frecuentes sobre la suite CSS. Además, el libro **Desarrollo y animación en ActionScript 3.0** nos ayudará a comprender los fundamentos de la programación en Flash.

- Realizar un boceto con todos los personajes u objetos nos dará una idea del tamaño y la ubicación de cada pieza.
- Cada objeto debe ir en una capa, con su nombre correspondiente, para poder moverlo y editarlo fácilmente.
- Es conveniente buscar para cada forma los círculos que la componen. Esta antigua técnica ayuda en la creación de trazados complejos.
- Las líneas pueden arrastrarse, para remodelarse y eliminarse con un simple clic. Podemos mover las uniones, añadir líneas extra, doblar y quitar elementos a nuestro antojo.
- Si es necesario copiar un dibujo, conviene tomar cada frame que lo compone, para evitar pérdidas y deformaciones.
- Es una experiencia distinta dibujar líneas con el mouse, y con la práctica, resulta gratificante.
- Utilizar la lupa para localizar pequeñas zonas abiertas en un dibujo permite realizar un buen manejo del bote de pintura sobre esa área.
- Es aconsejable usar la técnica de calcado cuando el dibujo no es nuestro fuerte. Así, copiamos y pegamos el

dibujo, o lo importamos a la Biblioteca. Luego, bloqueamos la capa que lo contiene y, con la herramienta **[Línea]**, vamos calcando y adaptando los trazos a la forma deseada, usando **[Selección]**. ◀



En www.iconique.com vemos la aplicación de degradados profesionales, y un excelente trabajo con trazos y rellenos.

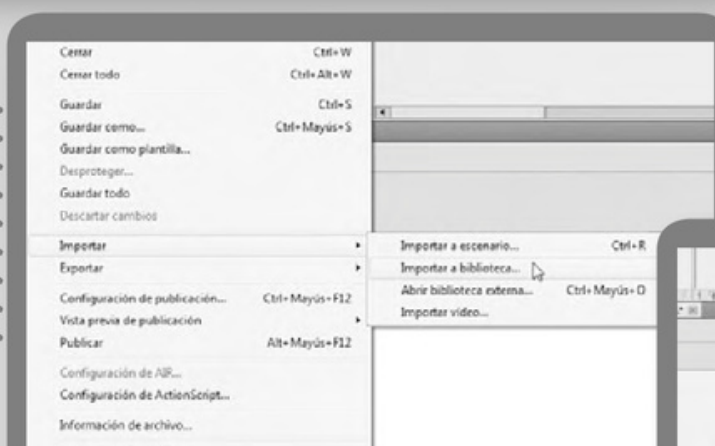


En el sitio www.mariaclaudiacortes.com se puede ver una aplicación de Flash sobre teoría del color y su importancia en la comunicación.

DIBUJO EN

PASO A PASO

En este apartado práctico pondremos en ejercicio las nociones teóricas sobre dibujo en Flash que aprendimos en las páginas anteriores.



01 | ABRIR IMAGEN

En esta sección demostraremos en detalle cómo realizar un dibujo en Flash aplicando la técnica explicada en las páginas precedentes. Para hacerlo, necesitamos la imagen que queramos copiar, la cual podemos descargar de Internet o extraer desde un archivo, para ilustrar digitalmente siguiendo un breve y sencillo procedimiento. ◀

PASO | 01

Podemos utilizar las herramientas básicas de dibujo de Flash para realizar una ilustración digital. Con este fin, buscamos una imagen, que puede ser la digitalización de algún dibujo que hayamos hecho o algún boceto. Cuando tengamos la imagen elegida en la computadora, la importamos a Adobe Flash, para lo cual abrimos la aplicación, creamos un nuevo documento y vamos a **[Archivo/Importar/Importar a Biblioteca]**.



CREAR CAPAS | 02

PASO | 02

En el panel **[Biblioteca]** tendremos la imagen elegida, que debemos arrastrar al escenario para trabajar sobre ella. Cuando esté en ese lugar, estará ubicada dentro de la capa que, por defecto, se encuentra en el panel de la línea de tiempo. Creamos otra capa y la colocamos sobre la capa superior. Podemos nombrar las capas para comenzar a mantener un orden.

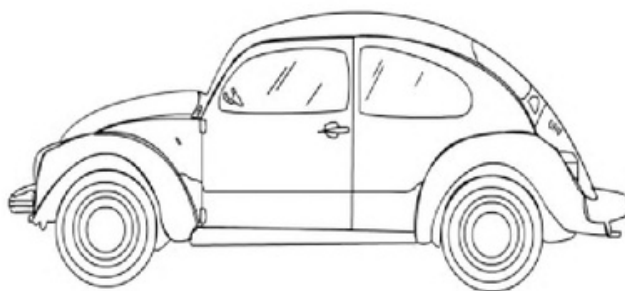
PASO | 03

Seleccionamos la herramienta **[Pluma]** y vamos delimitando alguna forma cerrada que tenga la imagen. En este caso, comenzaremos por calcar el capot del automóvil, pero primero lo haremos redibujándolo con formas rectas. Solo tenemos que hacer un clic y dirigir el cursor hacia el próximo punto, hasta cerrar la figura. Cuando terminemos una figura cerrada, con los trazos de la herramienta **[Pluma]**, pasaremos a modificar los puntos de ancla para amoldar las formas a las figuras.

FLASH



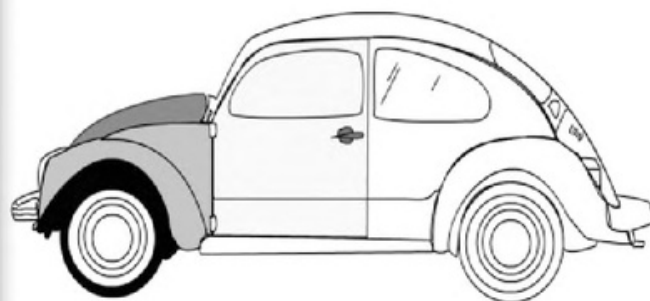
03 | HERRAMIENTA PLUMA



DIBUJAR DETALLES **05**



04 | MODIFICAR PUNTOS DE ANCLA



BOTE DE PINTURA **06**

PASO | 04 Con la herramienta **[Modificar puntos de ancla]** podemos agregar o sacar los que queramos. Hay que tener

presente que las figuras deben agruparse de modo que, si dibujamos una y esta se toca con otra, no deben sustraer partes de sus trazos. Las figuras que se tocan se restan por partes de delimitación. Seleccionamos una figura completa y nos dirigimos a **[Modificar/Agrupar]**.

De este modo, esta figura no cambiará y evitaremos tener que hacer el dibujo en diferentes capas.

PASO | 05 Cuando hayamos terminado de delimitar todas las figuras, pasaremos a dibujar los detalles.

Tenemos que observar cuáles son y agregarlos con las diferentes herramientas que ya aprendimos a manejar. En este caso, le daremos algunos detalles al auto para generar un poco más de complejidad.

PASO | 06 Ahora pasamos a usar la herramienta **[Bote de pintura]** para colorear de manera sólida las formas creadas que integran el dibujo. Como están separadas en grupos, será más fácil darle un color a cada una. Para finalizar, podemos agrupar todo el dibujo seleccionándolo en la sección **[Modificar/Agrupar]**.

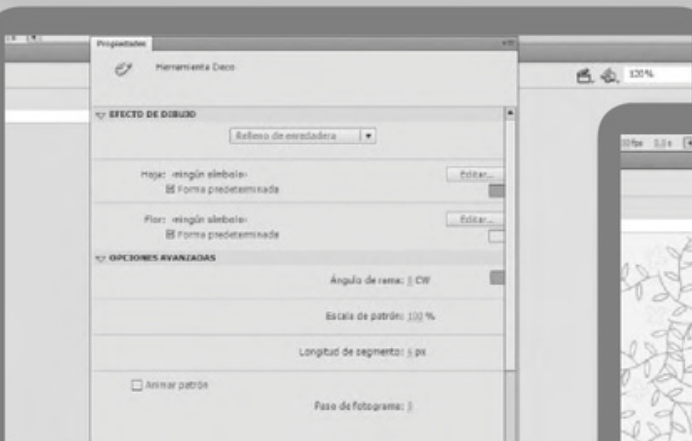
RESUMEN

Hemos aprendido cuáles son los pasos elementales para calcar una ilustración y colorearla con las herramientas básicas de Flash. De este modo, podemos rediseñar imágenes y utilizarlas para otros proyectos.

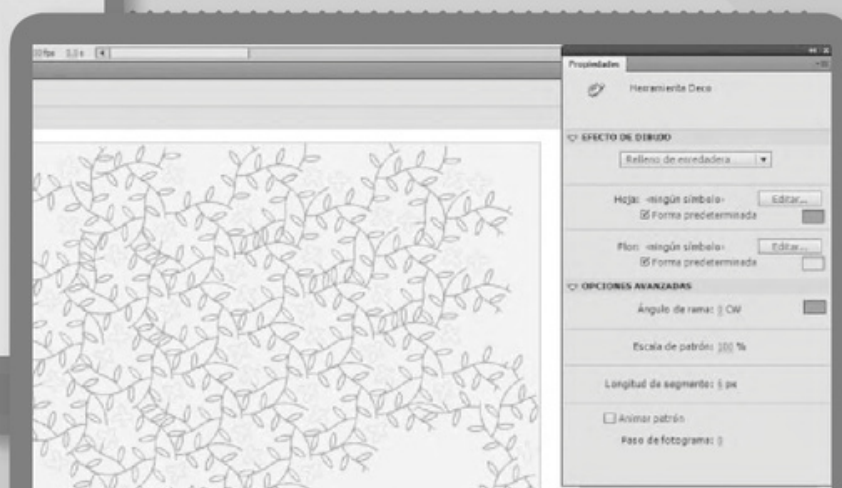
LA HERRAMIENTA

PASO A PASO

Anteriormente, vimos la aplicación de algunas herramientas de dibujo. En este apartado, evaluaremos la utilidad de otra muy importante.



01 CREAR DOCUMENTO



ANIMACIÓN DECORATIVA 02

Esta vez, analizaremos las utilidades de la herramienta **[Decó]** y la implementaremos en un proyecto de Flash. Con ella, podremos crear algunos estilos decorativos prediseñados para utilizar en diferentes ocasiones, y también podremos aprovechar otras de sus características, como la aplicación de movie clips a los trazos. Además, es posible cambiar, animar y asignar código Action Script. ◀

PASO | 01 Abrimos Flash y creamos un nuevo documento. En el escenario dibujamos alguna figura —podemos hacerlo con la herramienta **[Creación de formas]**—, como un rectángulo o un óvalo. Seleccionamos la opción **[Decó]** en el panel de herramientas y abrimos sus propiedades. Encontraremos dos secciones: **[Efecto del dibujo]** y **[Opciones avanzadas]**.

PASO | 02 En la solapa **[Opción de Diseño Decorativo]**, estará seleccionado por defecto el dibujo de enredadera. En las **[opciones avanzadas]** veremos la función **[Animar patrón]**. Si aplicamos la herramienta **[Decó]** en el Escenario, notaremos que el efecto decorativo se animará en la línea de tiempo, creando fotogramas clave para generar una animación en tiempo real. De este modo, obtendremos una película con una animación decorativa producida con esta herramienta.

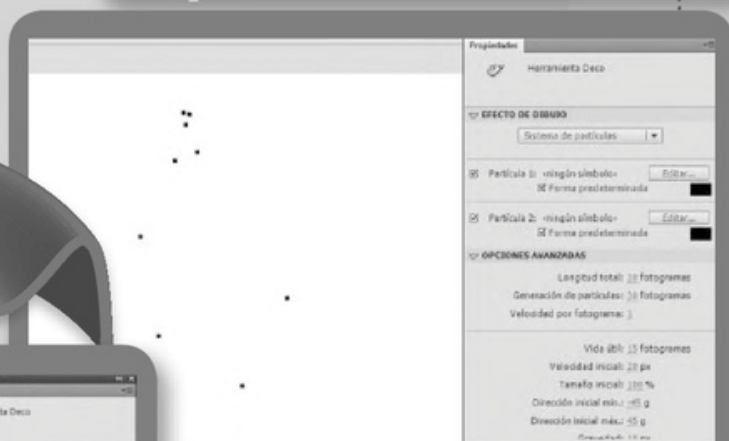
PASO | 03 Con esta opción también podemos realizar un tramado en forma de grilla o mosaico. En este caso, hay símbolos predeterminados que son cubos, pero es posible agregar otras formas, si es que las tenemos configuradas previamente. También tenemos la posibilidad de cambiar los patrones y la aparición de los símbolos de esta opción (ordenada o aleatoria), pero en ambos casos se generará una trama de decoración específica.

[DECÓ]

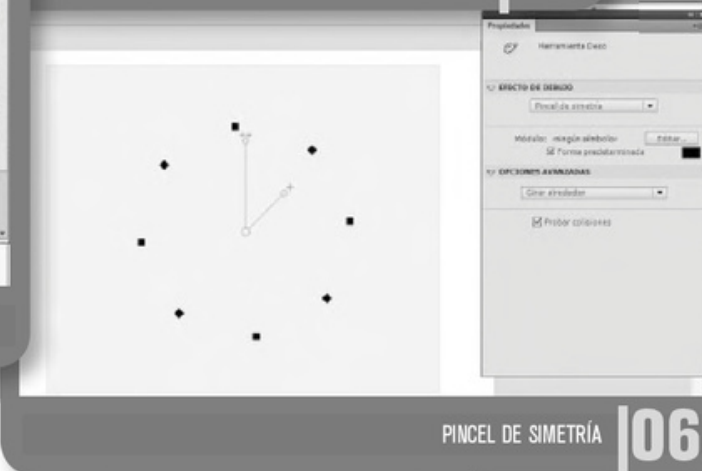
04 | PINCEL DECORATIVO



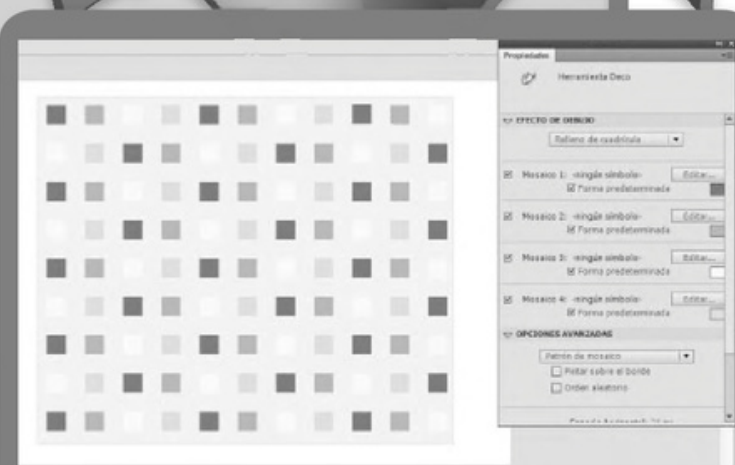
ANIMACIÓN DE PARTÍCULAS | 05



PINCEL DE SIMETRÍA | 06



03 | GRILLA O MOSAICO



PASO | 04

En este paso probaremos el pincel decorativo de la herramienta **[Decó]**.

Este posee 20 opciones de decorado en forma de guirnaldas con distintos diseños: zigzag, cuerdas, triángulos, ondas y hojas, entre otros. Podemos usarlos para decorar trabajos como misceláneas o bien como guardas.

PASO | 05

La opción **[Sistema de partículas]** permite realizar una animación en la línea de tiempo para generar un spray de partículas en movimiento. Existe la posibilidad de cargar un símbolo desde la Biblioteca para que las partículas tengan la forma que queramos. Por defecto, tiene unos cubos pequeños que generan la ilusión de lluvia.

PASO | 06

Para finalizar, podemos probar la opción **[Pincel de simetría]**. A esta aplicación de decoración también le podemos cargar

un símbolo desde la Biblioteca para generar formas simétricas y hacer que estas aparezcan de diferentes maneras: un reflejo a través de una línea, un reflejo a través de un punto específico, en forma circular alrededor, o en forma de grilla o cuadrícula.

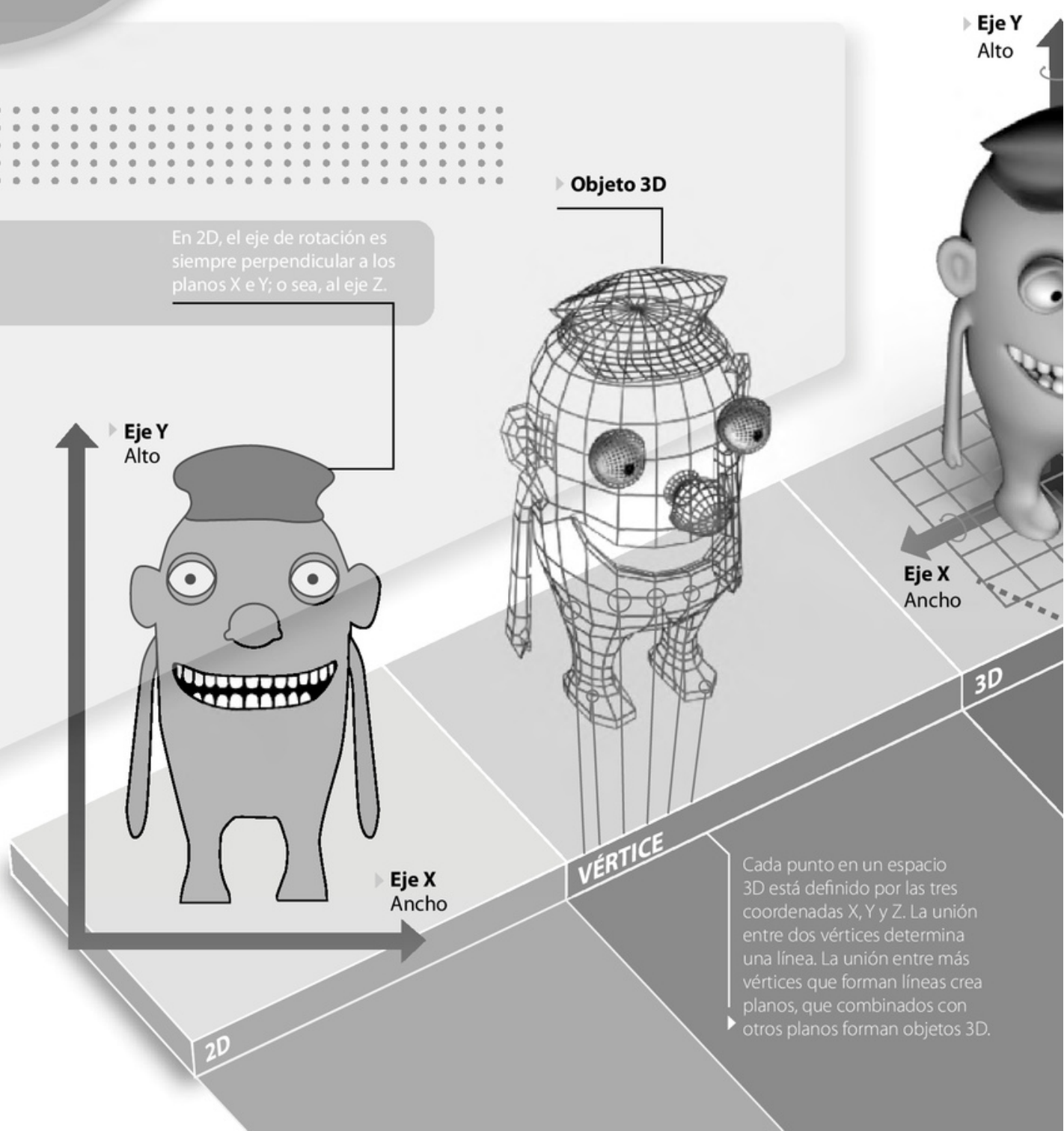
RESUMEN

Como podemos apreciar, las alternativas de la herramienta **[Decó]** nos ofrecen innumerables posibilidades. Es importante aclarar que hay más opciones de las que podríamos detallar en este paso a paso, donde, simplemente, hemos analizado las más relevantes.

ENTORNOS 3D

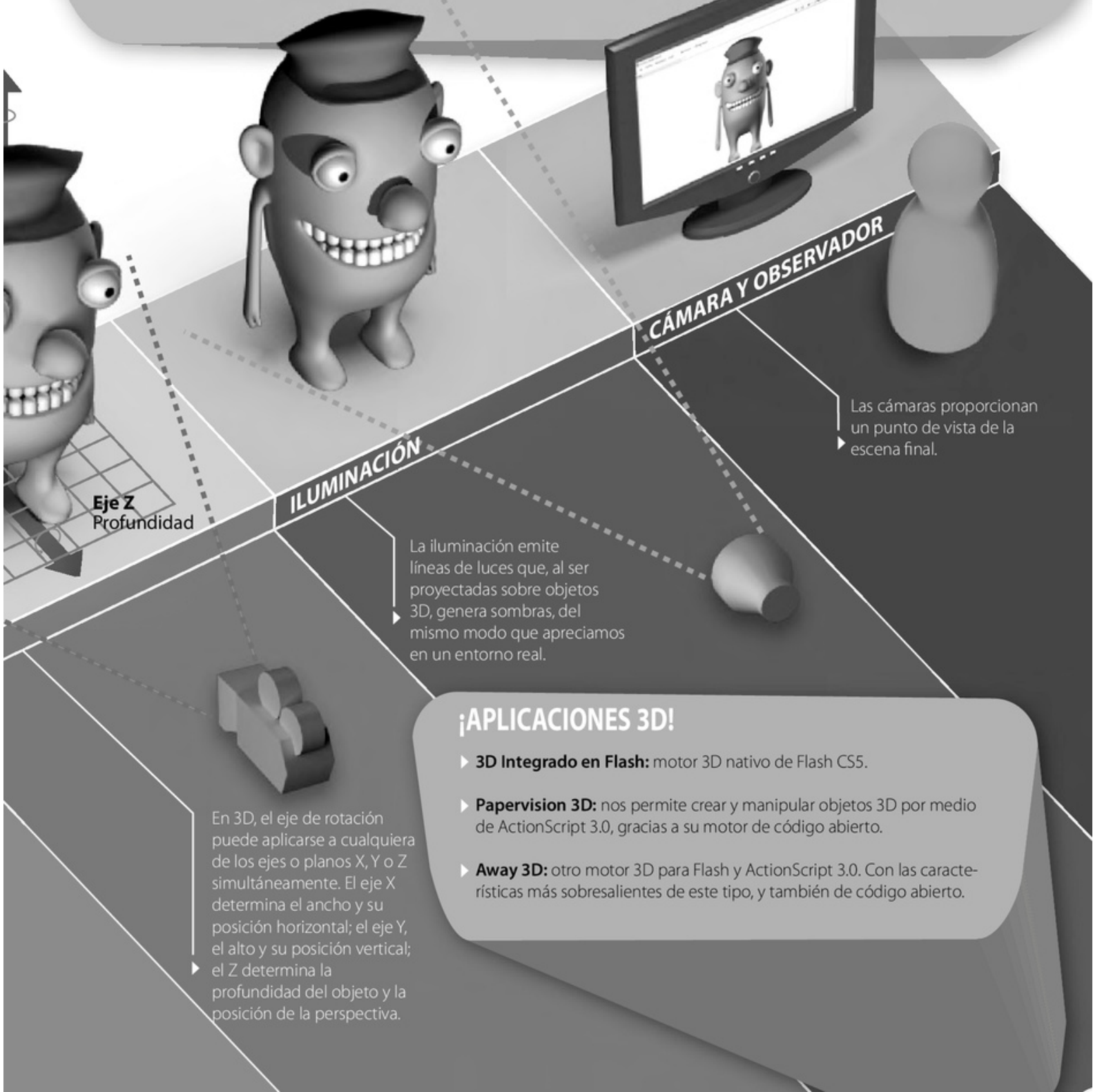
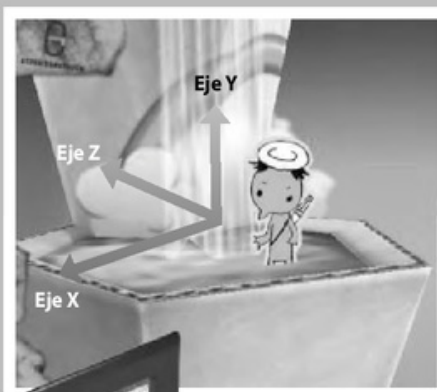
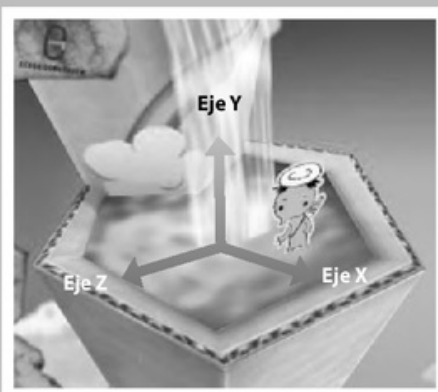
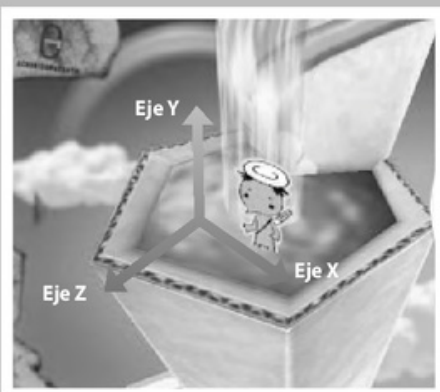
Veremos que el único mundo en 3D es el real, y que solo apreciamos simulaciones en dos dimensiones con un alto y un ancho específicos.

En 2D, el eje de rotación es siempre perpendicular a los planos X e Y; o sea, al eje Z.



DESPLAZAMIENTO Y TRANSFORMACIÓN 3D

Utilizando terminología 3D, nos referimos a **traslación** cuando movemos un objeto en alguno de los tres ejes que componen el espacio 3D (X, Y, Z). Si lo rotamos, escalamos o modificamos, nos referimos a su **transformación**.



CÁMARA Y OBSERVADOR

Las cámaras proporcionan un punto de vista de la escena final.

ILUMINACIÓN

La iluminación emite líneas de luces que, al ser proyectadas sobre objetos 3D, genera sombras, del mismo modo que apreciamos en un entorno real.

Eje Z
Profundidad

¡APLICACIONES 3D!

- ▶ **3D Integrado en Flash:** motor 3D nativo de Flash CS5.
- ▶ **Papervision 3D:** nos permite crear y manipular objetos 3D por medio de ActionScript 3.0, gracias a su motor de código abierto.
- ▶ **Away 3D:** otro motor 3D para Flash y ActionScript 3.0. Con las características más sobresalientes de este tipo, y también de código abierto.

En 3D, el eje de rotación puede aplicarse a cualquiera de los ejes o planos X, Y o Z simultáneamente. El eje X determina el ancho y su posición horizontal; el eje Y, el alto y su posición vertical; el Z determina la profundidad del objeto y la posición de la perspectiva.

INTROS EN FLASH

En estas páginas haremos un análisis en profundidad sobre la evolución de las intros y responderemos a los interrogantes clave. ¿Se siguen utilizando? ¿Cuándo y cómo está bien aplicarlas?

Dada la falta de experiencia y de conocimientos sobre usabilidad de los diseñadores durante el inicio de la Web, era muy común ingresar en un sitio institucional y encontrarnos con una página en blanco, que, luego de una larga espera, se convertía en una animación de elementos vectoriales: textos e imágenes desenmascarándose que, en el mejor de los casos, nos presentaban algún botón para poder saltar la introducción. Si bien en algunos casos las animaciones de las intros podían ser muy impactantes y vistosas, su contenido era generalmente pobre y, lejos de añadir algún valor al diseño de la página, provocaban inconvenientes al usuario final. Debido al gran peso de los archivos y a la baja velocidad de las conexiones a Internet de la época, las intros presentaban un dilema de usabilidad, porque en ocasiones hacían que los usuarios se retiraran del sitio sin siquiera haber visto el contenido que la página tenía para ofrecerles. De esta manera, fracasaba cualquier objetivo comunicacional que se hubiese propuesto la empresa al momento de contratar un servicio de diseño web. La necesidad de solucionar estos problemas, que más adelante explicaremos en detalle, tuvo como resultado la casi extinción de las animaciones de introducción.

LAS INTROS EN LA ACTUALIDAD

Así como Flash comenzó siendo una herramienta de animación vectorial y

evolucionó para convertirse en el entorno de desarrollo más utilizado para crear sitios web interactivos, las intros y la manera en que los diseñadores armaban los sitios también lo hicieron. Con el correr del tiempo, algunos diseñadores comenzaron a tomar conciencia sobre cómo realizar intros para que realmente añadieran un valor al diseño del sitio, y definir criterios para especificar cuando una intro era innecesaria o generaba problemas de usabilidad. Como resultado, podemos ver que las intros dejaron de ser aquellas animaciones que nos impedían llegar a la página

de inicio del sitio, y se transformaron, en muchos casos, en la animación de entrada de los elementos que componen el diseño de la página web que visitemos. Sin embargo, todavía podemos ver sitios que utilizan intros como las de los primeros años de la Web, pero en cuyo caso su uso está perfectamente justificado, ya que el usuario está dispuesto a disfrutar de ellas. Los sitios de películas de Hollywood o series estadounidenses son un excelente ejemplo de un buen uso de intros en Flash y nos demuestran que hay ocasiones en las que todavía podemos aprovecharlas.

VENTAJAS DE LAS ANIMACIONES

Las animaciones captan la atención de los usuarios. Si bien las intros poseen más desventajas que ventajas, no podemos dejar de lado que, bien pensadas, son vistosas, y tanto a diseñadores como a usuarios nos gusta experimentarlas. Sin embargo, el hecho de que llamen la atención de nuestros visitantes no nos

UNA INTRO MAL APLICADA PUEDE HACER QUE EL USUARIO SE VAYA DE LA PÁGINA SIN NAVEGARLA.



REITERACIÓN DE CONTENIDOS

Una de las desventajas más importantes que tienen las intros es la reiteración del contenido. Cualquier información que se brinde dentro de ella puede ser fácilmente reemplazada por una página de inicio bien estructurada con destacados, que ofrezcan ese material. Más adelante veremos qué alternativas existen hoy en día para reemplazar las intros y cómo podemos prescindir de ellas. Sin embargo, por el momento podemos asegurar que estos elementos pueden funcionar muy bien, agregando calidad a la página, pero también pueden alejar al usuario cuando se convierten en un escollo para la navegación.



Una intro obsoleta, en la cual podemos observar el botón **Skip intro** sobre el ángulo inferior derecho.

El sitio <http://portfolio.overprint.com.ar/simpson/indexf.htm> es un excelente ejemplo de una intro moderna.



El sitio <http://watchmenmovie.warnerbros.com> presenta una intro que el usuario no querrá saltar.



DESVENTAJAS DE LAS ANIMACIONES

El factor tiempo es uno de los mayores inconvenientes que tienen las intros. Los usuarios suelen ser impacientes, y no todos están dispuestos a esperar un determinado tiempo de descarga que, fácilmente, podría haberse evitado. Es importante analizar qué tipo de sitio estamos armando y si el target al que apuntamos estará dispuesto a aguardar este tiempo extra que le impedirá llegar a la información de manera rápida y eficaz.

PROBLEMAS DE INDEXACIÓN POR LOS BUSCADORES

Si bien Google realizó un excelente trabajo al momento de generar motores de búsqueda que puedan indexar sitios Flash, es responsabilidad del diseñador pensar en una estructura del sitio capaz de ser indexada. El problema principal que se genera al intentar indexar un sitio que posea una intro en Flash es que, muy probablemente, no utilizemos texto, sino vectores, bitmaps o videos. Por lo tanto, los motores de búsqueda solo leerán un **Saltear intro** o, simplemente, los tags de title y meta de HTML, con lo cual el sitio no será bien posicionado. ◀

da permiso a utilizarlas sin discreción ni criterio alguno. Por eso debemos tener sumo cuidado al realizarlas y evaluar a conciencia si son necesarias para los propósitos del sitio.

LAS INTROS COMO GENERADORAS DE INTERFAZ VISUAL

Como mencionamos anteriormente, la manera de realizar intros cambió por completo. En muchos casos, estos elementos pasaron de ser piezas separadas del contenido, para convertir-

se en animaciones que terminan formando la interfaz que el usuario utilizará posteriormente. Este tipo de intros se caracterizan por durar menos tiempo que las antiguas y no generan inconvenientes de usabilidad para la persona que visita el sitio. Por último, como su diseño está integrado a la interfaz final, la experiencia para el usuario es mucho más enriquecedora que si los elementos aparecieran de manera estática, como en un sitio HTML.

LAS ALTERNATIVAS A LAS INTROS

Ya hicimos un análisis sobre las intros. En estas páginas explicaremos qué podemos hacer para evitarlas y seguir explotando los recursos y posibilidades que nos ofrece Flash CS5.

Teniendo en cuenta el análisis que realizamos anteriormente sobre las ventajas y desventajas al realizar animaciones introductorias, vamos a incursionar en la búsqueda de alternativas a ellas. Como mencionamos anteriormente, una de las desventajas de este recurso es la reiteración de contenidos, y la información que se muestra en ellas puede ser fácilmente reemplazada por una página bien estructurada. A continuación, veremos algunos ejemplos y opciones para destacar información que sea relevante y de interés para el usuario utilizando animaciones en Flash.

ENCABEZADOS ANIMADOS EN FLASH

Los encabezados en Flash son animaciones que permiten brindar al usuario la misma información que daríamos a

conocer dentro de una intro, pero cuando él ya ha ingresado en la página de inicio. La ventaja de esto es que el usuario puede disfrutar de las animaciones o videos que utilicemos para nuestros encabezados y, a la vez, encontrar toda la información y contenidos alternativos que nuestra página tenga para ofrecer, sin necesidad de largas esperas. Si visitamos www.adobe.com/es, veremos un excelente uso de esta técnica. En este sitio, Adobe recurre a los encabezados para incrustar anima-

ciones impactantes realizadas en After Effects, a través del componente de video de Flash. Este tipo de animación fácilmente podría haber sido una página de introducción con un botón de Saltear intro, pero debido a que el contenido que brinda no es relevante, esa hubiese sido una mala decisión de diseño. De esta manera, Adobe muestra lo que se puede lograr con sus herramientas, y el usuario puede acceder a otras secciones que le sean de mayor interés, con facilidad y rapidez.

En www.funkrush.com podemos decidir si ver la animación o visitar directamente la sección de e-commerce.



Encabezado de la página de Adobe publicitando la última versión de sus productos.



SLIDESHOWS ANIMADOS

Una alternativa más para presentar animaciones es el uso de slideshows o galerías que presenten la información o los contenidos más destacados de la página. Esta alternativa, junto con los encabezados animados, son los más utilizados para sitios web híbridos. En ellos se emplean tecnologías de desarrollo estático (**HTML y CSS**) y se destaca la información con animaciones realizadas en Flash cargadas en un **slideshow** que el usuario puede controlar. El beneficio de aplicar estas técnicas radica en que los contenidos secundarios, ya sean animaciones Flash o videos, pueden cargarse una vez que el contenido principal fue presentado. De este modo, el usuario puede recorrer el sitio o visualizar información antes de que se presenten las animaciones desarrolladas para ellos.

DEBEMOS ASEGURARNOS DE QUE **UNA INTRO** NO DURE MÁS DE **CINCO SEGUNDOS**.



Si ingresamos en www.2fish.com.au podemos observar la utilización de intros donde los elementos animados que forman parte de la escena dan forma a la página de inicio.

INTROS MODERNAS EN LA HOME PAGE

Otra opción interesante para sitios completamente desarrollados en Flash es utilizar intros pequeñas, de dos a tres segundos, donde los elementos que se animan finalicen dando forma a la página de inicio. Así podemos añadir un valor agregado al sitio desde la animación y darle al usuario la posibilidad de navegar por el sitio según los estándares de usabilidad. El sitio de la agencia de diseño internacional 2fish (www.2fish.com.au) es un excelente ejemplo de esta alternativa. Allí los elementos que conforman la propuesta de diseño se integran en la página en tan solo dos o tres segundos, y el mismo concepto de animación se mantiene durante toda nuestra estadía en la **home** y se replica para cada sección.

CREAR NUESTRAS PROPIAS ALTERNATIVAS

Si bien estuvimos analizando algunas de las ideas básicas sobre cómo reemplazar una intro, esto no quita que podamos explorar y generar nuestras propias alternativas para hacerlo. En www.thefwa.com y www.dopeawards.com veremos algunos ejemplos de los mejores sitios en Flash, para poder inspirarnos al momento de realizar nuestros trabajos.

RECOMENDACIONES PARA REALIZAR INTROS

Para finalizar con este análisis, dejemos en claro algunas recomendaciones que debemos tener en cuenta al momento de realizar intros o animaciones para sitios Flash.

► **Tiempos de duración:** si vamos a realizar una intro, debemos asegurarnos de que su duración no sea mayor a cinco segundos. Como sabemos, las animaciones más extensas poseen algunas desventajas, por lo que es importante recordar este factor.

► **Saltear intro:** si la animación de introducción es muy extensa, debemos brindarle al usuario la posibilidad de saltarla. Él debe tener la sensación de que puede ser libre para elegir.

► **Añadir cookies para próximas visitas:** este es un factor que muy pocos sitios tienen en cuenta. Si nuestro sitio tiene una animación de introducción muy extensa, podemos añadir una cookie al usuario la primera vez que lo visita. De esta manera, las próximas veces no tendrá que ver la animación. ◀

+ EL TAMAÑO DE LOS ARCHIVOS

El tamaño de los archivos, medido en MB, es uno de los aspectos principales que debemos tener en cuenta al momento de diseñar. Cabe aclarar que el peso es directamente proporcional al tiempo de la carga; es decir, cuanto más pesado es el elemento, mayor será el tiempo de espera por parte del usuario. En otras palabras, el tamaño de los archivos es relevante porque determinará el tiempo de descarga que el usuario tendrá antes de poder visualizar la animación. Si los archivos son muy pesados, pueden llevar a que el usuario se retire antes de verla.

CARGADORES PARA SITIOS EN FLASH

En este apartado de la clase conoceremos en detalle los cargadores (loaders) para sitios desarrollados en Flash CS5 y ofreceremos respuestas a todos los interrogantes sobre su aplicación en la web.

Cuando un usuario ingresa a un sitio Flash, el navegador realiza una petición al servidor web en donde está alojado el archivo **SWF** para comenzar con su descarga y poder mostrarle al usuario el contenido solicitado. Cuanto más pesado sea el archivo, más tiempo tardará en realizarse la descarga y, por ende, se requerirá un mayor tiempo de espera por parte del usuario para poder visualizar nuestro sitio. Es por eso que surgen las precargas. Estas nos permiten notificar al usuario que un archivo esté siendo descargado y que necesitaremos de algunos segundos de su paciencia para que pueda visualizar el contenido.

LA IMPORTANCIA DEL CARGADOR

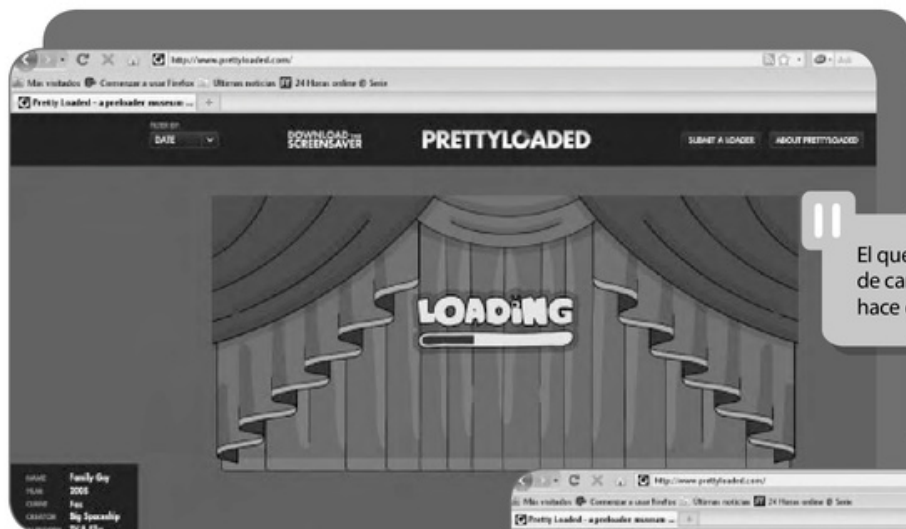
Si nosotros no generáramos una precarga para los archivos de nuestros sitios Flash, los usuarios no sabrían que un archivo está siendo descargado y solamente verían una página en blanco. De esta manera, podrían pensar que nuestro sitio funciona incorrectamente y retirarse de él sin visualizar el contenido. En los inicios de la Web, cuando utilizábamos conexiones de tipo dial-up, era muy común tener que esperar varios segundos y, en algunos casos, hasta minutos para poder descargar alguna película **SWF** de tan solo unos 100 o 200 KB.

+ CONOCIMIENTOS TÉCNICOS

Recordemos que, para mejorar y potenciar nuestro aprendizaje, desde www.redusers.com/cursos/webdesign podemos descargar los archivos editables de los proyectos que realizamos en esta colección. A partir de ellos podremos obtener ideas para un proyecto propio o, directamente, modificar los archivos para reutilizarlos. En el sitio también encontraremos **guías visuales**, **glosario de términos** y una completa guía de preguntas frecuentes sobre la suite CSS. Además, el libro **Desarrollo y animación en ActionScript 3.0** nos ayudará a comprender los fundamentos de la programación en Flash.

Actualmente, en la mayoría de los casos utilizamos conexiones de alta velocidad, por lo que la descarga de una película de menos de 50 KB es casi instantánea. Sin embargo, para archivos **SWF** de más de 50 Kilobytes deberíamos notificar al usuario que hay un archivo que está siendo descargado y que debe esperar unos instantes antes de visualizarlo. Muchos diseñadores informan a los visitantes que hay contenido en proceso de descarga a través de campos de texto con indicadores numéricos del porcentaje realizado. Otros utilizan barras que van creciendo o elementos que simplemente se animan. Los cargadores no sólo se utilizan para notificar al usuario sobre la descarga de contenido, sino también para indicar su subida. Existen muchos sitios que permiten

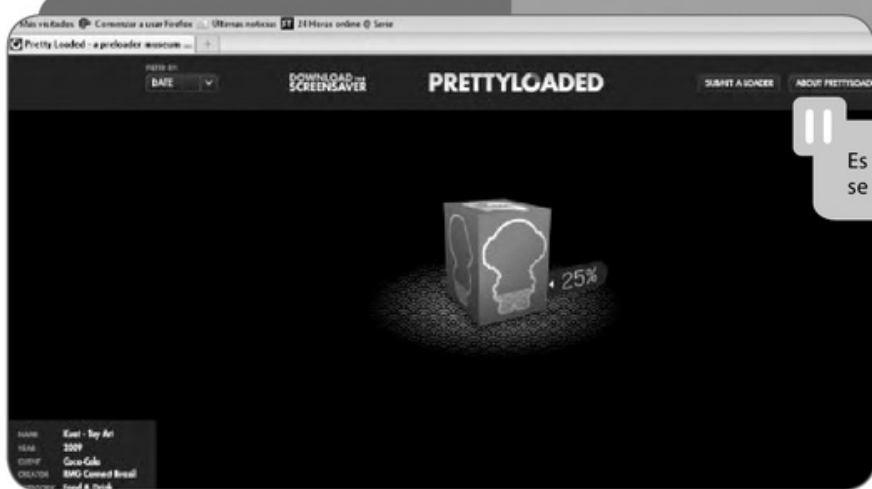




El que vemos en la imagen es un ejemplo de cargador animado. Su simplicidad hace que se cargue más rápido.



Observamos aquí solo uno de los variados ejemplos que podemos encontrar en <http://prettyloaded.com>.



Es este caso, vemos cómo el porcentaje se manifiesta en una solapa lateral.

a los usuarios subir imágenes, videos u otro tipo de archivos a sus servidores. En estos casos, se utilizan cargadores que notifiquen a los usuarios sobre el porcentaje del archivo que ha sido subido al servidor, de manera que él comprenda que no debe retirarse del sitio sin terminar dicha acción o, de lo contrario, el archivo no se subirá por completo. A modo de resumen, podemos decir que los cargadores surgen como una solución a un problema de usabilidad.

TENDENCIAS

Así como las velocidades de las conexiones fueron evolucionando, los diseñadores y la manera en que visualizamos las precargas también. Hoy en día es muy común encontrar con sitios cuyas precargas son un

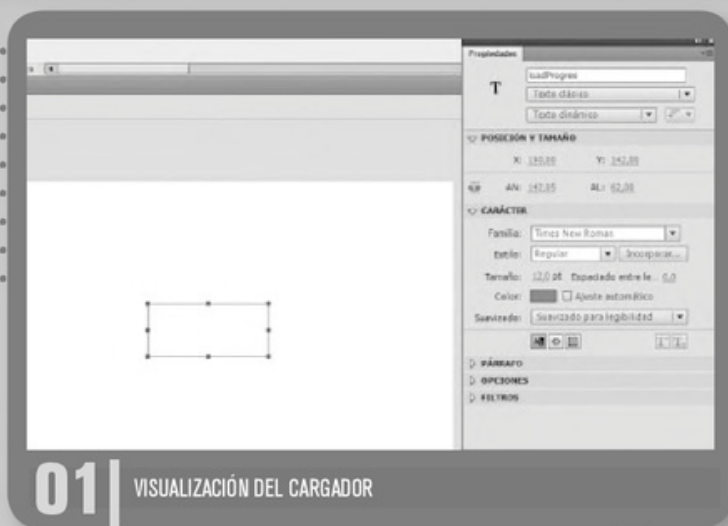
elemento más de identidad del sitio web y que tratan de entretener al usuario de una manera diferente a los ya conocidos campos de texto porcentuales o barras que se van escalando a medida que se descarga o sube un archivo. Es importante comprender que cuanto mayor es el tiempo de descarga de un archivo, existen más probabilidades de que el usuario se retire sin visualizar el contenido. Es por eso que podemos utilizar los cargadores como elementos de entretenimiento para el usuario y como un medio más donde transmitir la identidad de nuestro sitio. Una de las últimas tendencias en la web es aprovechar el tiempo de descarga de los archivos para entretener al visitante. De esta manera, se disminuyen las posibilidades de que este se aburra y retire

del sitio. Para ejemplificar este concepto, basta con visitar <http://www.stonewall.co.za>. Cuando ingresamos al sitio, se presenta una pequeña aplicación que nos permitirá crear nuestras propias melodías, simplemente arrastrando unos puntos y cruces que contienen diferentes notas y tonos musicales. Este es un excelente ejemplo de cómo mantener al usuario entretenido a través del uso de una precarga creativa. Otro ejemplo del buen uso de conceptos de precarga creativos es el sitio del escritor argentino Fernando de Vedia: <http://www.fernandodevedia.com>. En él podemos poner a prueba nuestra memoria a través de un simple juego de estilo "memotest". Para completar los ejemplos creativos de cargadores, la agencia de diseño de Nueva York, Big Spaceship, una de las más importantes y reconocidas a nivel mundial, desarrolló una galería de cargadores que podemos encontrar en www.prettyloaded.com. Allí podremos ver los casos más interesantes y creativos sobre cómo entretener a nuestros usuarios durante los tiempos de descarga y hasta tomar algunas ideas para futuras creaciones. ◀

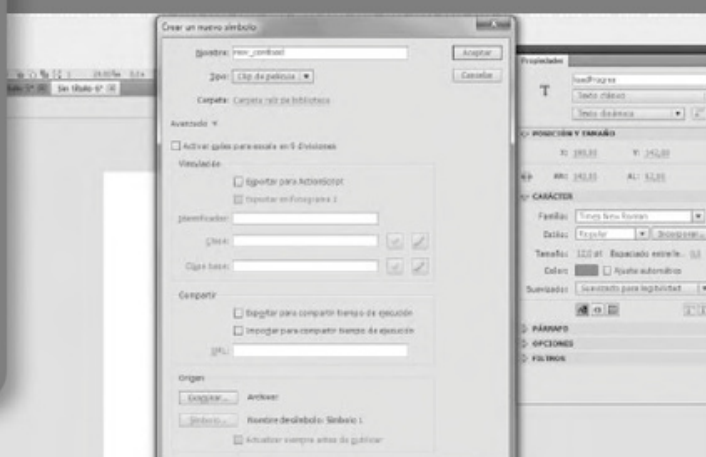
CREACIÓN DE UN CARGADOR

PASO A PASO

Veremos cómo realizar un cargador simple en Flash, de manera que los usuarios puedan visualizar el proceso de descarga de nuestro sitio web.



01 | VISUALIZACIÓN DEL CARGADOR



NOMBRE DE INSTANCIA | 02

Al finalizar los pasos de este proceso, veremos qué fácil resulta hacer un cargador con Flash CS5 mediante su lenguaje de programación ActionScript 3. Además, aprenderemos a reutilizar el código para generar cargadores con animaciones impactantes que se destaquen y deslumbren a los usuarios que utilicen nuestros futuros proyectos web. ◀

PASO | 01

PASO | 01

Para crear la visualización del cargador, tomamos la herramienta de texto presionando **<T>**, generamos un campo de texto dinámico, y definimos su tipografía, tamaño y color. Debemos asegurarnos de que sea un campo **dinámico**, para lo cual vamos al panel de **[Propiedades]** y seleccionamos la opción de ese nombre. Por último, asignamos un nombre de instancia, que para este caso será **"loadProgress"**.

PASO | 02

PASO | 02 Luego de crear el campo de texto, lo convertimos en **movie clip** (<Ctrl + F8>) y le asignamos un nombre de instancia, que, para nuestro ejemplo, será **"preLoad"**. Estos nombres sirven para llamar a los elementos gráficos desde el código que escribiremos más adelante.

PASO | 03

PASO | 03

El tercer paso es crear un layer nuevo y posicionarnos en el primer frame de la película. Una vez allí, abrimos el panel de acciones con **<F9>** y redactamos las líneas de código que veremos en la pantalla 03 de la página siguiente. La primera línea le indica a la película Flash que este frame debe detenerse para que el resto de los contenidos no sean descargados. En la segunda línea, le asignamos el valor inicial al campo de texto, haciendo, luego, el que creamos en el primer paso. Esto se logra a través del método **“.text”**, que nos permite asignar un valor de texto. En este caso, será **“0%”**, el


```
stop();
preLoad.loadProgress.text = "0%";
preLoad.x = stage.stageWidth / 2 - preLoad.width / 2;
preLoad.y = stage.stageHeight / 2 - preLoad.height / 2;

this.loaderInfo.addEventListener(ProgressEvent.PROGRESS, onLoadProgress);
this.loaderInfo.addEventListener(Event.COMPLETE, onLoadComplete);
```

SIMPLE

04 | AÑADIR LOS LISTENERS

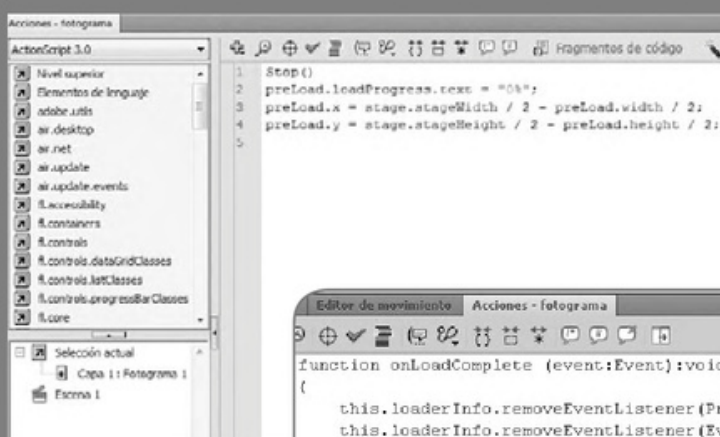
```
Stop();
preLoad.loadProgress.text = "0%";
preLoad.x = stage.stageWidth / 2 - preLoad.width / 2;
preLoad.y = stage.stageHeight / 2 - preLoad.height / 2;

this.loaderInfo.addEventListener(ProgressEvent.PROGRESS, onLoadProgress);
this.loaderInfo.addEventListener(Event.COMPLETE, onLoadComplete);

function onLoadProgress (event:ProgressEvent):void
{
    var cargado:int = event.bytesLoaded;
    var total:int = event.bytesTotal;
    var porcentaje:int = cargado/total*100;
    preLoad.loadProgress.text = String(porcentaje)+"%";
}
```

MANEJAR LA FUNCIÓN DEL PROGRESO DE CARGA.

05



03 | CREAR UN LAYER

```
function onLoadComplete (event:Event):void
{
    this.loaderInfo.removeEventListener(ProgressEvent.PROGRESS, onLoadProgress);
    this.loaderInfo.removeEventListener(Event.COMPLETE, onLoadComplete);

    nextFrame();
}
```

REMOVER LOS LISTENERS

06

porcentaje inicial de descarga de nuestra película. Por último, posicionamos el movie clip "preLoad" en el centro del stage.

PASO | 04 En Flash, todos los objetos visuales poseen un **loaderInfo** que nos permite acceder a información sobre su carga.

Asignándole un **ProgressEvent** al **LoaderInfo** de la película, realizamos un seguimiento de la carga. Por último, le asignamos un **listener** que dispare un evento en el momento en que se finaliza la descarga del resto de la película.

PASO | 05 Creamos una variable que contenga la información de los bytes cargados de la película. Luego, nos fijamos cuál es el total de bytes que deben ser descargados para, posteriormente, realizar un cálculo que almacenamos en la variable **porcentaje**. Esto se hace dividiendo los bytes cargados por el total a descargar, y multiplicándolo por 100. Una vez realizada esta cuenta,

actualizamos el campo de texto del cargador con el porcentaje obtenido, como vemos en la captura.

PASO | 06 En el último paso nos ocupamos de remover los **Listeners** añadidos para manejar la descarga de la película. Una vez que lo hicimos, le indicamos a la película Flash que vaya al siguiente **frame**, donde se encuentra el código que le presentará el sitio al usuario. En la imagen vemos el momento de la carga.

RESUMEN

En estos pasos vimos cómo crear un preloader actualizando un campo de texto con el porcentaje de la película descargada. Si bien este es un ejemplo simple, nos permite comprender los conceptos principales para poder realizar cualquier preloader.

OPTIMIZACIÓN DE SITIOS



Teniendo en cuenta que no todos los usuarios contarán con conexión de banda ancha rápida, queda en nosotros reducir el peso que el sitio tendrá a fin de minimizar los tiempos de descarga.

Aunque Flash efectúa automáticamente una optimización en los archivos **SWF** en el momento de su publicación, existen varias estrategias para reducir aún más ese tamaño. Primero, veremos algunos pasos que podemos llevar a cabo antes de exportar un documento para optimizarlo y, así, disminuir su peso y mejorar la velocidad de reproducción. Luego, descubriremos algunas herramientas para monitorear la optimización y analizaremos qué características deben tener los objetos que crearemos o los archivos que importemos al programa para que el documento final tenga un tamaño reducido.

USO DE SÍMBOLOS

Es muy común que nuestro diseño tenga elementos que se utilicen repetidas veces, y que la única diferencia entre ellos sea un cambio de tamaño, de color o de opacidad, por ejemplo. Esto suele ocurrir cuando utilizamos iconos, botones o, incluso, animaciones completas. Por cada nuevo elemento que creamos en el escenario se sumarán más bytes a la película final, lo que incrementará su peso. La solución para no aumentar excesivamente este parámetro es crear clips de película en la Biblioteca y colocar en el escenario una instancia distinta por cada uno que necesitemos. Luego, a través de **ActionScript** podremos alterar muchas de sus propiedades, tales como posición, tamaño, transparencia y color. Incluso, podremos aplicarles filtros como sombra, iluminado o bisel.

ANIMACIONES

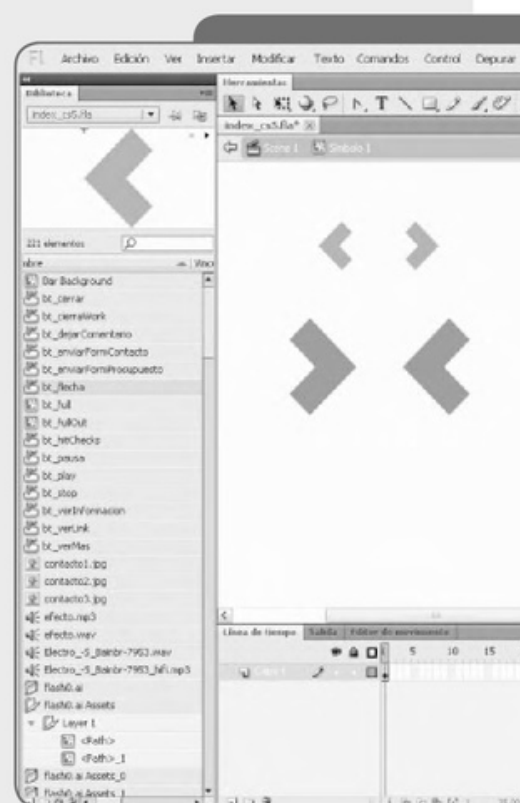
Uno de los puntos fuertes de Flash es la creación de animaciones. Una opción que tenemos es utilizar la técnica de **fotograma por fotograma**, imitando a los ilustradores tradicionales. Sin embargo, esto puede aumentar de manera considerable el tamaño del documento publicado. Para evitar este inconveniente, el programa también nos da la posibilidad de hacer animaciones por interpolación de movimiento, usando **ActionScript**, lo que nos garantiza el uso de menos recursos y, a su vez, mayor libertad de acción. Lo único que tenemos que hacer es recurrir a la clase **Tween** sobre un movie clip para modificar en el tiempo algunas de sus propiedades, por ejemplo:

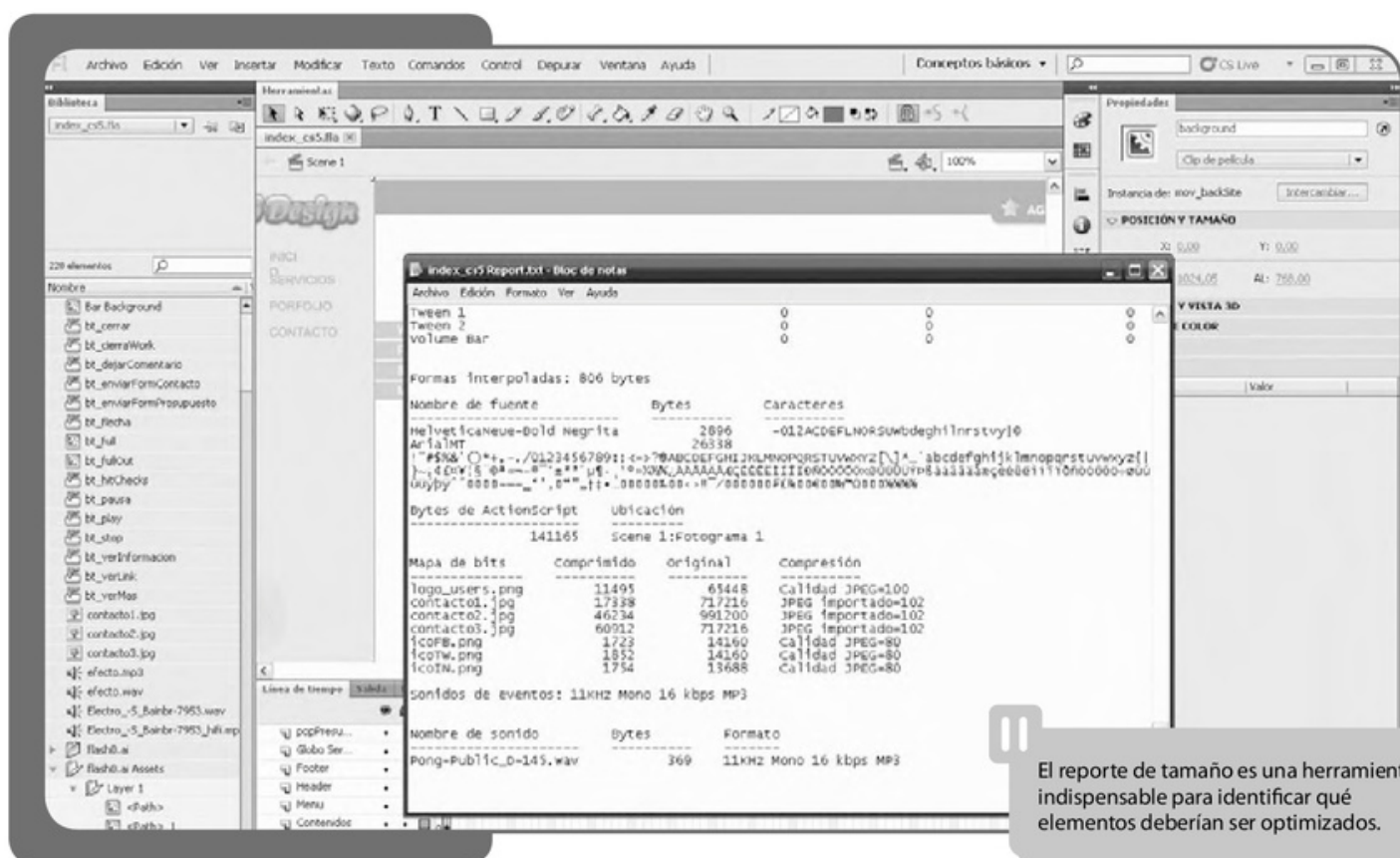
```
import fl.transitions.Tween;
import fl.transitions.
easing.*; var
nuevoTween:Tween = new
Tween(objeto_mc, "x",
Elastic.easeOut, 0, 300, 3,
true);nuevoTween.start();
```

IMÁGENES

Por tener una mayor calidad y generar un menor peso, siempre es recomendable utilizar elementos vectoriales en Flash, ya que este programa fue pensado para trabajar, principalmente, con ellos. Sin embargo, también podemos usar imágenes de mapas de bits, aunque si no las

optimizamos de modo adecuado, es posible que aumenten demasiado el peso del documento **SWF**. Aunque Flash tiene opciones para reducir el peso de las imágenes en el momento de publicarlas disminuyendo su calidad, lo ideal es optimizarlas antes de importarlas al programa. Por ejemplo, Photoshop cuenta con la función **[Exportar para Web]**, que nos permite configurar distintos valores para lograr un equilibrio entre peso y calidad al guardar la imagen. En este sentido, **PIXresizer** es un





programa freeware específicamente creado para reducir el tamaño de las fotos. Nos da la posibilidad de elegir entre distintos tamaños de imagen, convertir a otros formatos y disminuir varias al mismo tiempo automáticamente.

Otro consejo para evitar subir el peso del **SWF** es importar las imágenes a Flash con el tamaño final al que se las utilizará. Por ejemplo, si una fotografía utilizada en el escenario tiene la mitad de tamaño que la versión originalmente importada a

la Biblioteca, su peso, al momento de publicarla, será el mismo, como si no hubiese sido reducida. El formato **GIF** resulta ideal para imágenes con menor cantidad de colores y áreas bien definidas, como puede ser un logotipo o una trama geométrica, aunque su desventaja es que está limitado a imágenes de 256 colores. En cambio, el **JPG** logra la mejor relación calidad/peso en imágenes con amplias paletas de colores y degradés, como son las fotográficas. Por su parte, el formato **PNG** combina un poco de cada uno de los dos anteriores: permite un reducido peso en imágenes fotográficas o de colores planos y, además, utiliza transparencia.

SOBRE LA TIPOGRAFÍA

Cuando utilizamos campos de **texto dinámico con tipografías** que no son las de sistema (Arial, Verdana, Times New Roman, etc.), debemos embeberlas para que cualquiera pueda verlas en su computadora. Dado que esto suma el peso de la tipografía al del documento, debemos ser cuidadosos. Por un lado, conviene limitar la cantidad de fuentes distintas que usemos y, por el otro, sólo incrustemos los caracteres que vayamos a emplear. Es decir, no tiene sentido importar los **255 símbolos** de una fuente tipográfica si sólo vamos a usar las mayúsculas. ◀



ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué herramientas básicas de dibujo encontramos en Flash?

02 ¿Cuáles son las herramientas avanzadas de dibujo que ofrece Flash?

03 ¿Para qué sirve la herramienta **Pluma**?

04 ¿Qué es la herramienta **Decó**?

05 Mencione las características de un entorno 3D.

06 ¿Qué aplicaciones en 3D podemos lograr en Flash?

07 ¿Cómo podemos crear una intro en Flash?

08 ¿Qué alternativas existen a las intros en Flash?

09 Mencione algunas recomendaciones para crear una intro exitosa.

10 ¿Qué es un cargador para sitios?

PRÁCTICAS

01 Utilice las herramientas de dibujo incorporadas en Flash.

02 Aplique la herramienta **Decó**.

03 Cree una intro en Flash.

04 Cree un cargador para sitios en Flash.

05 Optimice su sitio web.

Animaciones básicas

- ▶ AXIOMAS DE ANIMACIÓN
- ▶ CLASSIC TWEEN
- ▶ CLASSIC TWEEN Y GUÍAS DE MOVIMIENTO
- ▶ MOTION TWEEN
- ▶ MOTION TWEEN Y MOTION PRESET
- ▶ MOTION EDITOR
- ▶ ANIMACIONES CON MOTION EDITOR
- ▶ CINEMÁTICA INVERSA
- ▶ CÓMO ANIMAR HUESOS CON MUELLES



AXIOMAS DE

ANIMACIÓN

Veremos de qué modo crear y manipular novedosas animaciones para agregar a nuestros proyectos, aprovechando al máximo las propiedades de Flash.

Flash es una herramienta creada, originalmente, para la animación de gráficos vectoriales. Hasta el día de hoy, ha ido incorporando diversos elementos que hacen que sea un completo recurso para la animación y creación de interactividad. El principio de animación y de todo lo que creemos en Flash –así sean animaciones o ámbitos interactivos– está basado en el uso de líneas de tiempo, divididas en fotogramas. Estas, a su vez, están ordenadas en capas desde las que se maneja la prioridad de visualización e interacción de los elementos. La línea de tiempo es un elemento que encontraremos tanto en el escenario principal, como dentro de cualquier **movie clip**, cada uno de los cuales tiene su propio timeline. Esto hace que tengamos una gran versatilidad para generar toda clase de animaciones y movimientos. Usando la línea de tiempo, podremos determinar cómo es un elemento en el principio de ella (forma, tamaño, ubicación, color, etc.) y definir cómo será al final. Flash se ocupará de generar todos los estados intermedios para dar forma a la animación.



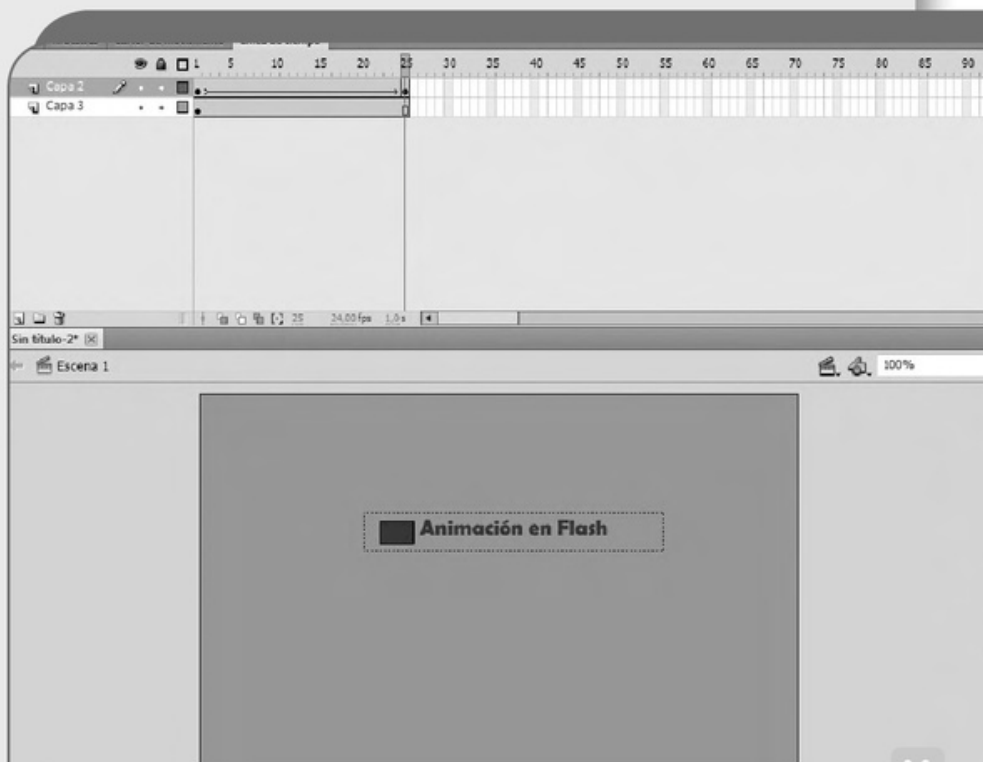
ORDEN DE LOS FOTOGRAMAS

El orden en que se muestran todos los fotogramas a lo largo de una línea de tiempo determina una secuencialidad en la reproducción. Por eso, es conveniente organizarlos contemplando la secuencia de eventos en la animación. Cabe recordar que todas las opciones para aplicar (insertar, borrar, agregar fotograma clave, copiar, pegar, etc.) pueden encontrarse haciendo clic con el botón derecho del mouse sobre el fotograma en cuestión, que está en la línea de tiempo.

LAS LÍNEAS DE TIEMPO

Las líneas de tiempo organizan el contenido del documento en fotogramas y capas. También determinan el orden de reproducción, es decir, el momento en que los elementos que estamos manipulando van apareciendo o desapareciendo del escenario. Al igual que en la mayoría de los programas de edición de video, **el tiempo está dividido en fotogramas**. Podemos relacionar los de Flash con los de un rollo de película, en el que, sobre cada uno, hay una imagen distinta. Claro que, al verlos reproduciéndose uno tras otro, percibimos el movimiento entre esas imágenes que, en un principio, eran estáticas. Lo mismo sucede en Flash, donde una animación está compuesta

por varios fotogramas en los que podemos ver la animación dividida en 20 o 30 pasos. La reproducción de estos elementos se realiza con cierta velocidad, representada por un valor denominado **fotogramas por segundo**. Este es un parámetro único para toda la película y establece cuántos fotogramas se verán en el transcurso de un segundo. El valor predefinido en Flash es de 24 fotogramas por segundo; esto quiere decir que para hacer una animación que dure 1 segundo, esta deberá estar compuesta por 24 fotogramas. Las capas, en cambio, representan varias bandas sobre la película y establecen el orden de visualización de los elementos. En cada una podremos tener uno o más de ellos interactuando entre sí.



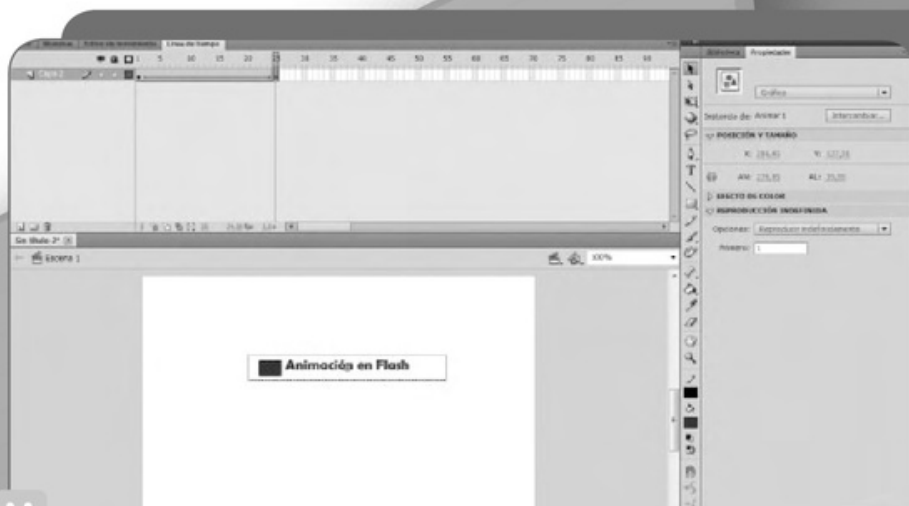
En este caso, observamos dos elementos superpuestos, cuyo orden de visualización está determinado por la posición de las capas en la línea de tiempo.

LAS CAPAS

Para entender claramente el funcionamiento de las capas en Flash, podemos trazar un paralelo con el área del diseño gráfico. Entonces, diríamos que las capas son hojas de papel calco superpuestas unas sobre otras sobre un papel en blanco, donde en cada hoja podemos representar uno o varios elementos. De tal modo, habrá elementos que queden ubicados por encima de otros, con lo cual se establecerá un orden o prioridad de visualización. También podemos reunir los elementos que componen las distintas capas aplicando el comando **[Agrupar]**, ubicado en el menú **[Modificar]**.

LOS FOTOGRAMAS CLAVE

Como comentamos anteriormente, las líneas de tiempo están divididas en fotogramas. Cuando generamos una animación, tenemos, por lo menos, un fotograma clave, aunque podemos tener tantos como nos hagan falta. Sabiendo esto, podemos definir un fotograma clave como **aquel en el que se puede determinar un cambio en la animación, o bien, aquel en el que es posible agregar una o varias acciones de ActionScript**. Estos elementos se representan en la línea de tiempo con un punto circular dentro del recuadro del fotograma.



Animación compuesta por 24 fotogramas, con una configuración de 24 fotogramas por segundo.

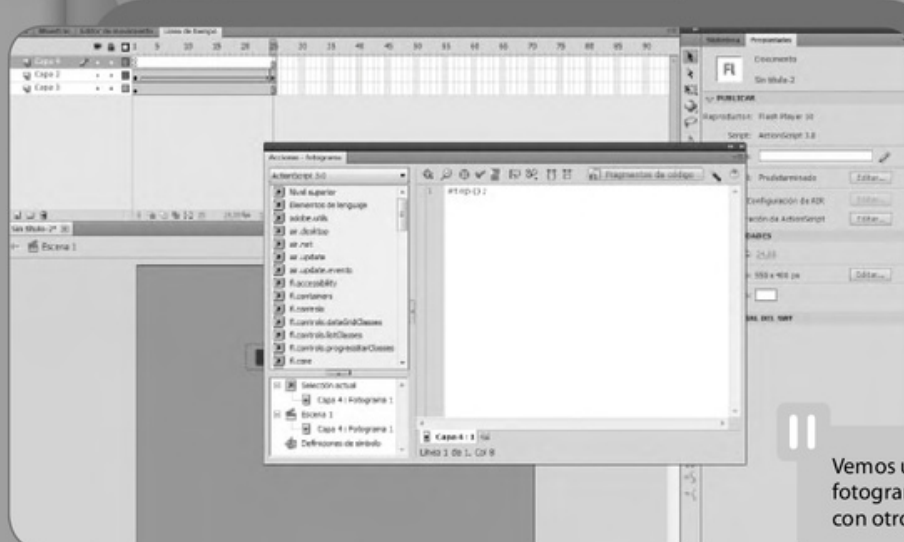
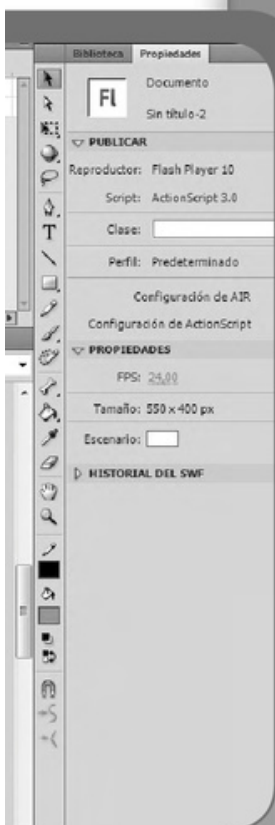
EL PRINCIPIO DE ANIMACIÓN ESTÁ BASADO EN EL USO DE LÍNEAS DE TIEMPO, DIVIDIDAS EN FOTOGRAMAS.

¿QUÉ SE PUEDE ANIMAR CON FLASH?

Todos los elementos que incorporemos al escenario de Flash pueden ser animados de una manera u otra. Dependiendo del elemento que estemos manipulando, podemos modificar su forma, tamaño, color o intensidad del filtro que tenga aplicado. Una combinación de muchos factores es lo que genera animaciones más atractivas. Por ejemplo, si queremos animar la caída de una pelota, primero pensaríamos en un único movimiento vertical de

arriba hacia abajo. Analizando un poco mejor el desplazamiento, nos damos cuenta de otros detalles:

- La velocidad del movimiento no debiera ser igual todo el tiempo, sino que tendría que ir acelerándose a medida que se acercara a su destino.
- Durante el trayecto del movimiento, al igual que sucede cuando vemos cualquier objeto que se desplaza a gran velocidad, no veríamos los contornos de la figura totalmente definidos, sino que percibiríamos algo así como cuando observamos una fotografía que está "movida".
- Por último, cuando el elemento llega a destino, cuando cae, posiblemente observaríamos que rebota, es decir, que se levanta y vuelve a caer algunas veces. Esta sería una mínima secuencia de pasos que demuestran rápidamente las potentes prestaciones de Flash en cuanto a animación.



Vemos una animación con dos fotogramas clave, y una capa superior con otro, con una acción de stop.



CLASSIC TWEEN

Con este método tradicional de Flash, empezaremos a comprender cómo es posible crear distintas animaciones y efectos con muchas de las herramientas de esta aplicación.

Qué elementos intervienen en este tipo de animación? Classic Tween, básicamente, se realiza entre **dos fotogramas clave** a los que llamamos **inicial y final**. En esta técnica interviene un mismo símbolo **movie clip** del que es posible modificar sus propiedades, como ubicación en X e Y, ancho, alto, transparencia, rotación, color y los filtros que tenga aplicados. Si tenemos un movie clip en el escenario en el fotograma 1, y agregamos un fotograma clave en esa misma capa, pero en el fotograma 24, es posible que generemos una animación. Para lograrlo, tenemos que modificar por lo menos una de las propiedades del movie clip en el fotograma 24. Supongamos que allí cambiamos la ubicación y el tamaño, ya sea desde el panel de **[Propiedades]** o con las herramien-

+

¿SE PUEDE USAR MÁS DE UNA GUÍA?

Esta es una de las preguntas que no podemos dejar de responder, ya que se ha planteado en muchos de los estudiantes de diseño. Por supuesto, podemos agregar tantas capas guía como animaciones distintas necesitemos hacer en un mismo documento. Lo que no es posible hacer, es crear dos capas guía e intentar vincularlas con una sola capa guiada. Tampoco podemos vincular las guías de movimiento con máscaras.

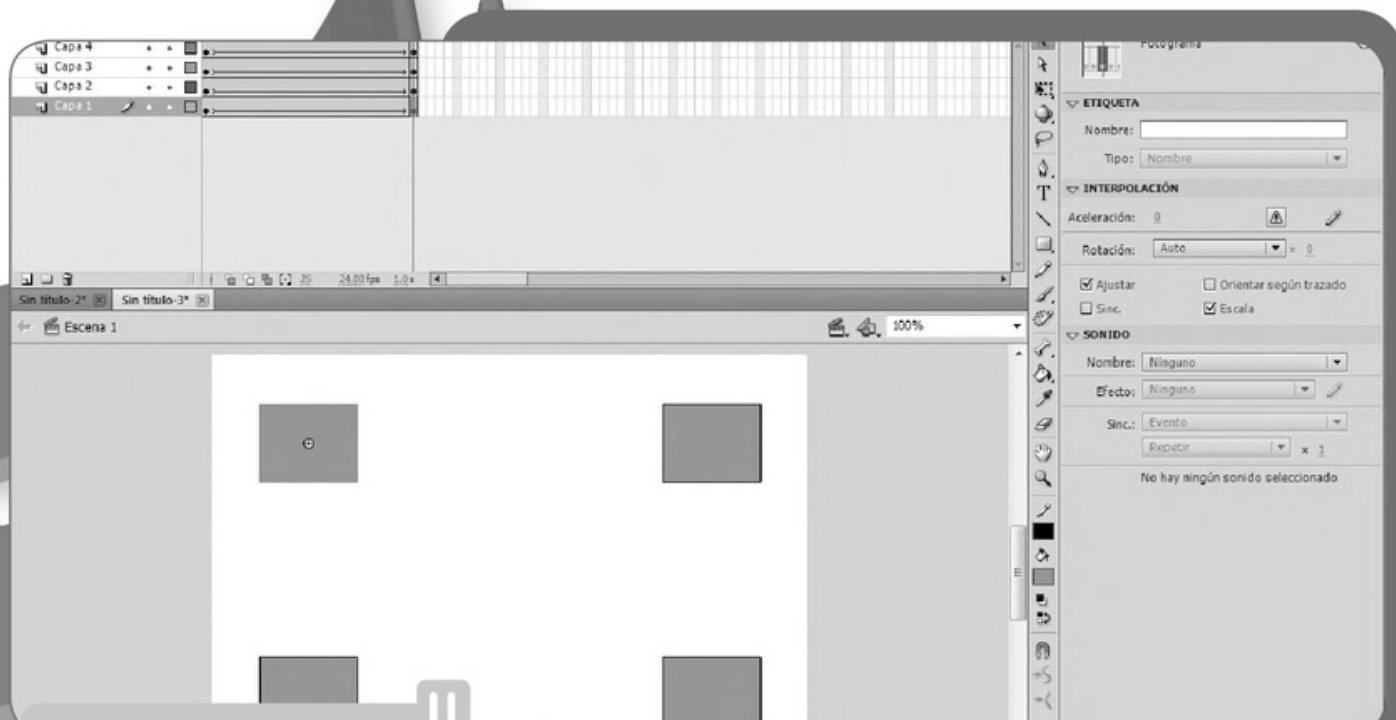
tas de la caja. Una vez hecho esto, solo tendremos dos puntos del movimiento: en uno está el movie clip con su tamaño y ubicación originales, y en el otro, el mismo elemento con las propiedades modificadas. Lo que necesitaremos entonces es que Flash recree o genere los pasos intermedios que deberíamos ver entre un punto y otro de la animación que estamos creando. Esta tarea es la que lleva

a cabo **Classic Tween**, herramienta a la cual accedemos haciendo clic derecho del mouse sobre la línea de tiempo en cualquiera de los fotogramas del 1 al 23 y, allí, seleccionando **[Classic Tween]**. Hecho esto, veremos que toda la línea de tiempo entre un fotograma clave y otro queda pintada de celeste, y aparece una flecha que parte del primer fotograma y termina en el 24. Esto indica que Flash creará los pasos intermedios para generar una animación que comience con el movie clip en el estado en que se encuentra en el **fotograma 1** y termine en el estado en que está en el **fotograma 24**. Es muy importante aclarar que solo es posible aplicar un Classic Tween sobre una capa que contiene un único elemento, y que si queremos realizar animaciones sobre varios de ellos, estos tendrán que estar ubicados en capas distintas.

RESPECTO A LOS NOMBRES DE INSTANCIA

Siempre que queramos aplicar código ActionScript a un elemento (por ejemplo, a un botón), y que este, a su vez, esté animado, es muy recomendable agregarle el nombre de instancia **antes** de animarlo y de añadir los fotogramas clave.





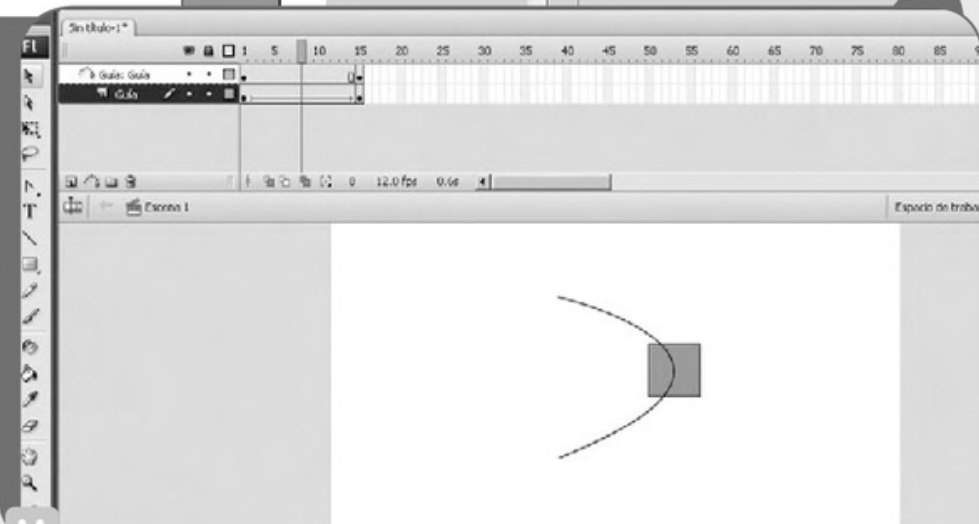
Aquí observamos cuatro elementos animados con el método Classic Tween, cada uno ubicado solo en su propia capa.

Si lo hacemos luego de realizar la animación, deberemos agregarle al movie clip el mismo nombre de instancia en cada fotograma clave que tenga la animación.

GUÍAS DE MOVIMIENTO

Usando las guías de movimiento, es posible hacer que las animaciones realicen recorridos curvos, siguiendo el trazado que dibujemos. Podemos usar como guía cualquier trazado hecho con la herramienta [Pluma]. Este tiene que ser una única línea que no se corte en ningún lado, pero que puede describir la cantidad de curvas y giros que sean necesarios. La guía de movimiento no es simplemente un trazado: para que funcione como tal, debe contemplar algunas cuestiones:

- ▶ Tiene que estar ubicado en una capa y no debe haber ningún otro elemento en ella.
- ▶ La capa donde está el trazado debe ser una **capa guía**. Esto se logra haciendo clic derecho del mouse y seleccionando la opción [Agregar guía de movimiento clásica]. Luego de esto, se agregará automáticamente una capa que será la guiada, y sobre ella debemos generar la animación que queramos controlar mediante la guía de movimiento. Sobre la capa guiada dibujamos un trazado que será el recorrido seguido por la animación.



Animación del tipo Classic Tween con la intervención de una guía de movimiento.

▶ El tipo de animación que se aplique a la capa guiada debe ser siempre del tipo Classic Tween. Cabe aclarar que las guías de movimiento no funcionan para animaciones Motion Tween ni Shape Tween.

TRABAJAR CON LA GUÍA

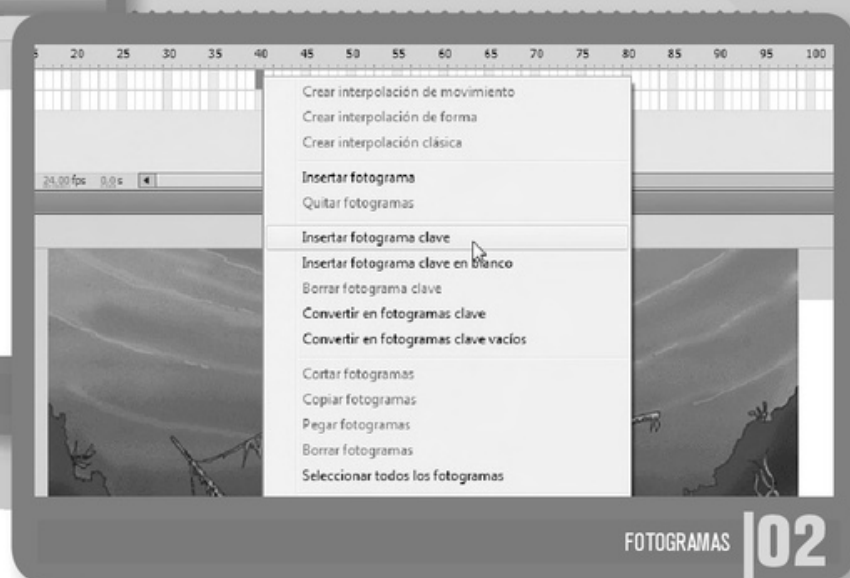
Una vez creados la capa guía y el trazado correspondiente (este puede ser editado y modificado en cualquier momento), debemos generar una animación Classic Tween sobre la capa guiada, por ejemplo, cambiando la posición del movie clip en el último fotograma o extendiendo la capa guía tantos fotogramas como dure la animación recién hecha. Si publicamos este archivo tal como

está, todavía veríamos desplazarse el movie clip de un punto a otro, sin seguir ninguna forma del trazado. Volviendo al escenario de Flash, en el primer fotograma de la animación arrastramos el movie clip hasta que el centro de rotación se pegue a la línea guía en alguno de sus extremos. Luego, en el último fotograma de la animación, repetimos el mismo procedimiento, es decir, pegamos el movie clip en el otro extremo del trazado. Si necesitamos que otro clip siga el mismo recorrido, podemos añadir un nuevo layer guiado por la misma guía. Simplemente, agregamos una nueva capa y la ubicamos inmediatamente debajo de la guía, luego de lo cual realizamos los mismos procedimientos anteriores. ◀

CLASSIC TWEEN Y GUÍAS DE

PASO A PASO

Luego de aprender las nociones básicas sobre animación, veremos cómo aplicarlas con Classic Tween y las guías de movimiento.



Una de las características principales de Flash es su gran potencial de animación, por lo que comenzaremos a crear una sencilla, empleando Classic Tween y, luego, aplicaremos las guías básicas de movimiento. ◀

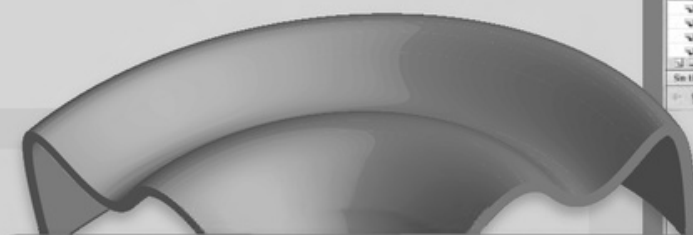
PASO | 01 Creamos un nuevo documento en Flash e importamos a la biblioteca los elementos que vamos a utilizar (seleccionando las opciones desde la ventana) como mapa de bits, para conservar el aspecto o mantener editables todos los trazados. Arrastramos cada imagen al escenario y, con la tecla <F8>, creamos los símbolos necesarios, cada uno en una capa.

PASO | 02 Posicionados en el **fotograma 1** de la capa que queremos animar, generamos una interpolación clásica, desde **[Insertar/Interpolación clásica]**. Ubicamos el objeto por animar donde deseamos que aparezca su primer movimiento. Seleccionamos el **fotograma 40** y creamos un fotograma clave, yendo a **[Insertar/Línea del Tiempo/Fotograma Clave]**. Arrastramos el objeto hacia donde se mostrará en la escena. Probamos la película.

PASO | 03 Para ajustar los cambios, nos ubicamos en el **fotograma 1** y, desde el panel de **[Propiedades]**, calculamos la aceleración que refleje mejor el movimiento, la rotación, etcétera, y volvemos a probarla. Damos la opción al símbolo de reproducir por única vez, para evitar loops o bucles indefinidos. ¡Ya podemos ver nuestro proyecto en acción!

MOVIMIENTO

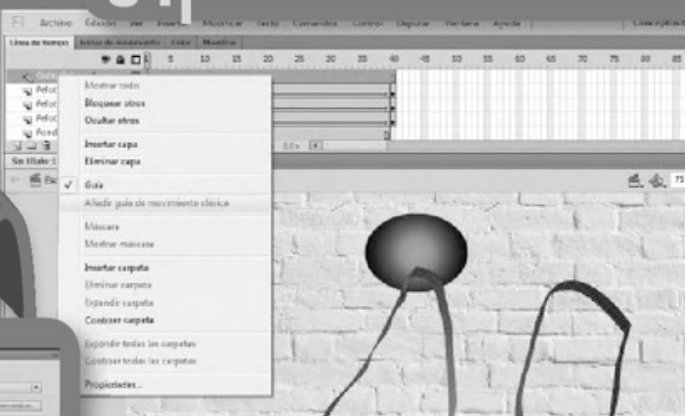
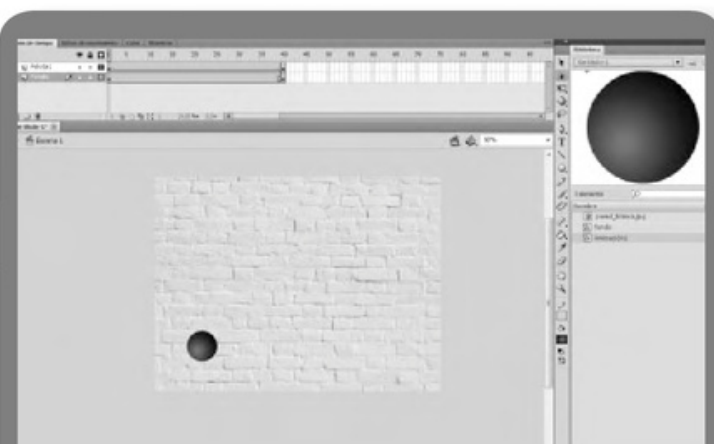
04 | GUÍAS DE MOVIMIENTO



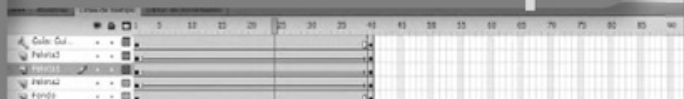
03 | AJUSTES

PASO | 04 Para aplicar las guías de movimiento, veremos un ejemplo con pelotas que saltan. Creamos un nuevo documento e importamos el fondo deseado a la biblioteca. Dibujamos una pelota con la herramienta **[Óvalo]** y la pintamos. La transformamos en **[Símbolo/Gráfico]** y arrastramos los elementos al escenario, cada uno en una capa, cuidando el orden de aparición de los personajes en la escena. A la capa **"Pelota"** le insertamos una **interpolación clásica**.

PASO | 05 Generamos una nueva capa sobre la de las pelotas y, en ella, con el botón derecho del mouse, insertamos una **Guía de movimiento clásica**. En el fotograma 1 de la guía, con la herramienta **[Pincel]**, **[Lápiz]** o **[Pluma]**, trazamos una línea que refleje el recorrido de las pelotas. También podemos modificarla, una vez concluida la animación. Ubicamos cada pelota sobre la línea.



05 | CREAR CAPA



06 | ACCELERACIÓN

PASO | 06 En el panel **[Propiedades]**, ajustamos la aceleración de cada pelota con distintos tiempos, rotaciones y cantidad de vueltas, para lograr un efecto más realista. Todas las capas que se ubiquen bajo la capa **"Guía de movimiento"** seguirán su trazado y, para desvincularlas, debemos arrastrarlas sobre la capa guía o desde **[Modificar/Línea del Tiempo/Propiedades/Normal]**.

RESUMEN

En este apartado aprendimos que el proceso de animación resulta muy sencillo, tanto con Classic Tween como mediante la aplicación de guías de movimiento.

MOTION TWEEN

Desde la versión CS4, Flash incorporó este novedoso sistema que soluciona la mayoría de los problemas que siempre tuvo este tipo de animación en las ediciones anteriores.

Para animar con el nuevo **Motion Tween**, no es necesario agregar un fotograma clave, que supone realizar un corte en la película y que todos los cambios que hagamos sean totalmente independientes entre un fotograma y otro. Ahora existe un nuevo tipo de fotograma clave llamado de **Propiedad**. Para hacer una animación, simplemente debemos extender la línea de tiempo hasta obtener la duración deseada. Esto se logra haciendo clic con el botón derecho sobre un fotograma y eligiendo **[Insertar fotograma]** o presionando **<F5>**. Luego hay que agregar el tipo de animación Motion Tween y, en el último fotograma (o en cualquiera después del primero), hacer alguna modificación en las propiedades del movie clip que queremos animar (ancho, alto, ubicación, etc.). Veremos que ese fotograma, donde modificamos la propiedad del movie clip, queda marcado con un **rombo** que hace referencia al **fotograma clave de Propiedad**.

¿POR QUÉ USAR MOTION TWEEN?

Al usar **Motion Tween**, podremos hacer animaciones en 3D utilizando las herra-

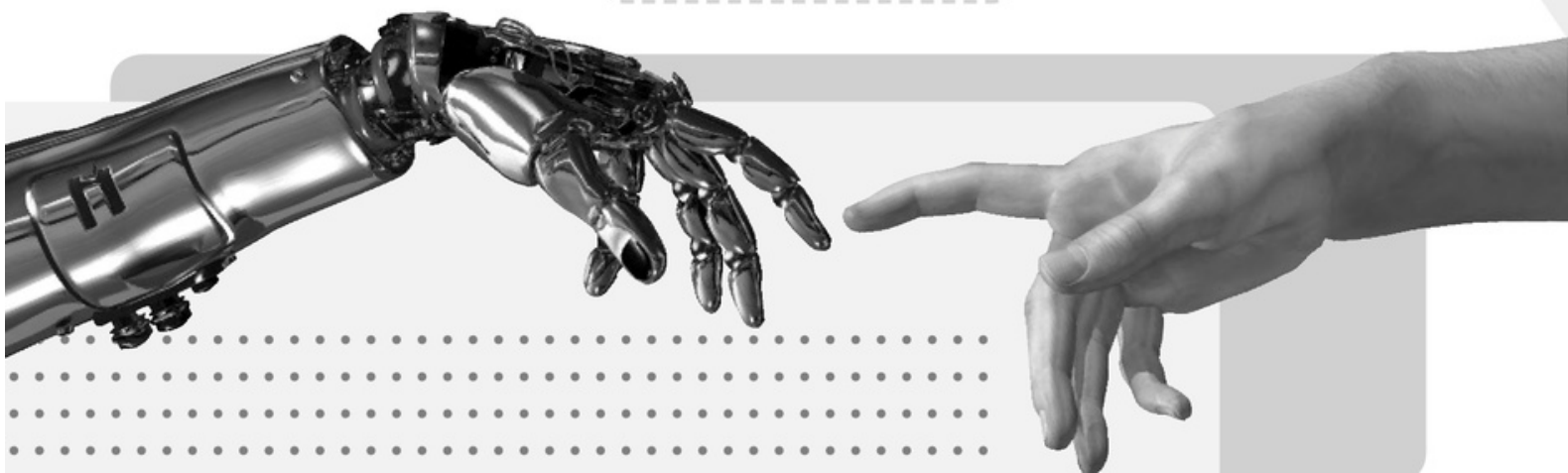
mientas que ofrece Flash para efectuar ese tipo de modificaciones. Si variamos la ubicación de un movie clip, aparecerá dibujada sobre el escenario una línea vectorial que marca la trayectoria del movimiento, y que podemos modificar usando las herramientas de dibujo. Esto permite que dicha trayectoria no sea en línea recta, sino que recorra el trazo tal como lo cambiamos. Ahora bien, puede suceder que la animación haya quedado tal como queríamos, pero que no se esté haciendo en el lugar

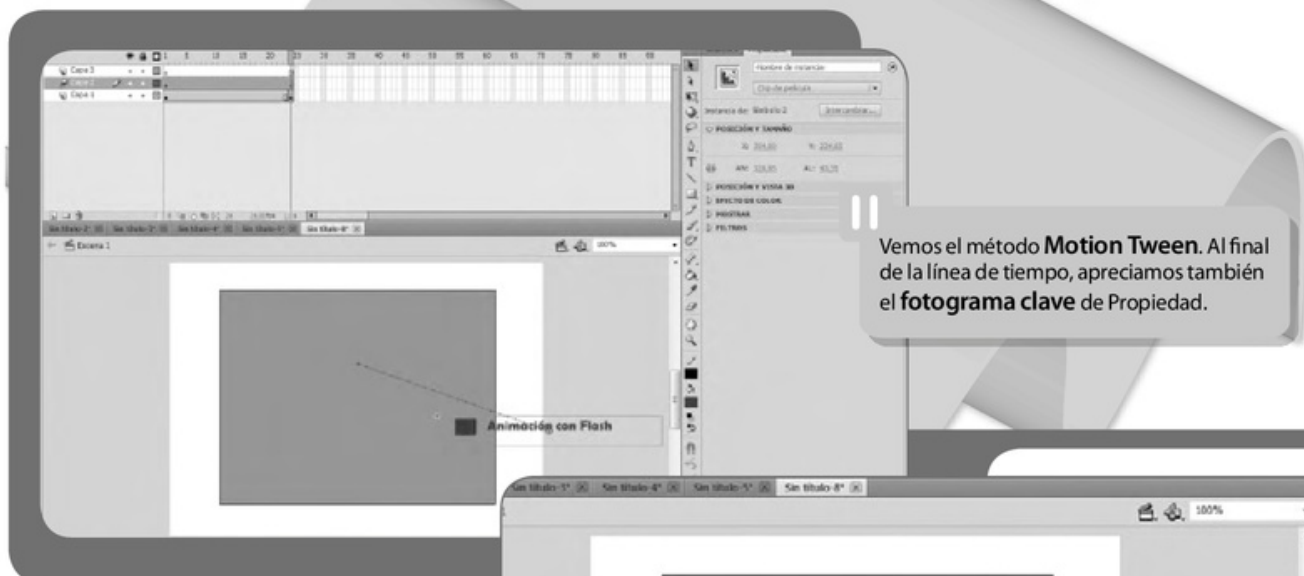
+ UNO POR VEZ

Luego de comprender cada uno de los conceptos detallados en este apartado, es importante destacar que podemos aplicar solo un Motion Preset a cada elemento que haya en el escenario. Si intentamos aplicar dos a la vez, veremos un mensaje de advertencia diciendo que el nuevo Motion Preset que se está queriendo aplicar reemplazará al presente actualmente. Este es un dato que siempre debemos tener en cuenta.

SOLO ES POSIBLE APLICAR UN MOTION PRESET A CADA ELEMENTO QUE HAYA EN EL ESCENARIO.

del escenario correcto. En la actualidad, no hay ningún problema en este caso, ya que al existir un único fotograma clave, podemos cambiar de lugar el movie clip y la línea de trayectoria, seleccionando todo junto en el escenario con la herramienta de **[Selección]**. De esta manera, no tendremos que modificar nada más. Al utilizar este tipo de animación, si se modifica más de una propiedad del mismo movie clip entre un fotograma clave y uno de Propiedad, es posible variar las aceleraciones de modo independiente para cada propiedad en la misma animación. Por último, una vez que logramos configurar la animación que estamos armando, podemos guardarla como una animación preseteada, usando la opción **[Salvar selección como preset]**, dentro de la ventana **[Motion Presets]**.





Vemos el método **Motion Tween**. Al final de la línea de tiempo, apreciamos también el **fotograma clave** de Propiedad.

MOTION PRESET

Flash incorpora una gran librería de animaciones predefinidas que aplican varios efectos a la vez para crear los movimientos más reales y con solo unos pocos clics del mouse.

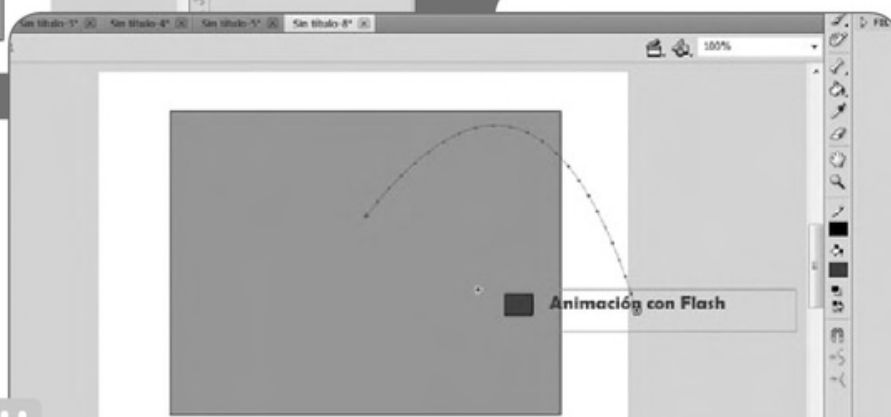
Si animamos un movie clip modificando su posición en el escenario, comenzando el movimiento en la parte superior y terminando en la inferior, y luego publicamos el archivo y vemos la animación resultante, podríamos decir que el elemento se está cayendo, pero: ¿esto es realmente así? Cuando un elemento se cae, ¿este solo cambia su ubicación en el eje vertical desde un punto superior hasta otro inferior? Si analizamos este tema, nos daremos cuenta de que no es tan simple. Por ejemplo, cuando un elemento cae, además de un cambio en su posición en el **eje Y**, también se produce un fenómeno de **aceleración**. Es decir, el elemento en cuestión va incrementando su velocidad a medida que cae.

¿Qué debería suceder en el trayecto? Por lo general, veríamos que ese elemento se desenfoca visualmente al estilo de una fotografía movida. Por último, podemos pensar qué sucede cuando el elemento hace contacto con la parte inferior del escenario. Lo natural sería que rebotara y, eventualmente, que con el mismo contacto se deformara un poco hacia los lados. Tal como analizamos, un simple movi-

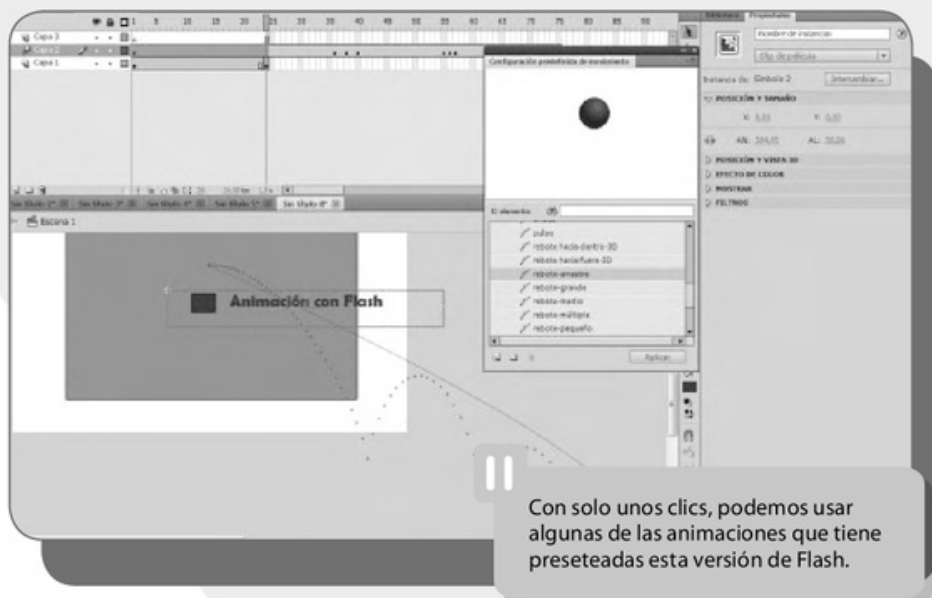
miento de caída está conformado, a su vez, por movimientos y efectos complementarios, tales como: aceleración, desenfoco, ensanchamiento y animación en sentido contrario. Justamente, este tipo de animación recibe el nombre de **animación compleja**. Este sería un ejemplo elemental de lo que **Motion Preset** puede hacer.

ANIMAR CON MOTION PRESET

En el menú **[Ventana]** encontraremos la opción **[Motion Preset]**. Al hacer clic sobre ella, se abrirá una ventana en la que veremos distintos tipos de efectos de animación para aplicar sobre cualquier movie clip. Para hacerlo, creamos un clip de película en el primer fotograma. Si seleccionamos primero el fotograma y, luego, el tipo de animación desde la ventana **[Motion Preset]**, y después hacemos clic en **[Aplicar]**, veremos que sobre la línea de tiempo en la que teníamos un único fotograma, se generará automáticamente un **Motion Tween** con el efecto elegido. Este, al igual que si lo hubiéramos creado a mano, es susceptible de ser modificado con cualquiera de las herramientas de **[Dibujo]** y **[Escala]**. ◀



La línea guía que se genera automáticamente con este método de animación, luego de modificarla con el cursor de selección para que la animación describa una curva.



Con solo unos clics, podemos usar algunas de las animaciones que tiene preseleccionadas esta versión de Flash.

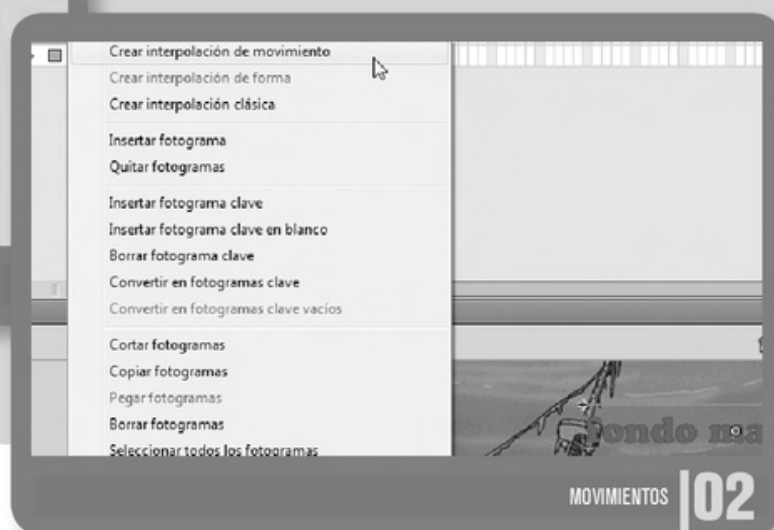
MOTION TWEEN Y MOTION

PASO A PASO

Ya conocimos el uso y la aplicación de estas herramientas. Pongamos manos a la obra para aplicar todos los conceptos en esta práctica.



01 CAPAS



MOVIMIENTOS **02**

Con Motion Tween podremos crear y modificar el trayecto de cualquier símbolo generado en la línea de tiempo. Por su parte, las configuraciones predefinidas de movimiento son interpolaciones que pueden aplicarse a un objeto del escenario. Estas son dos formas rápidas de aprender lo básico sobre las animaciones. ◀

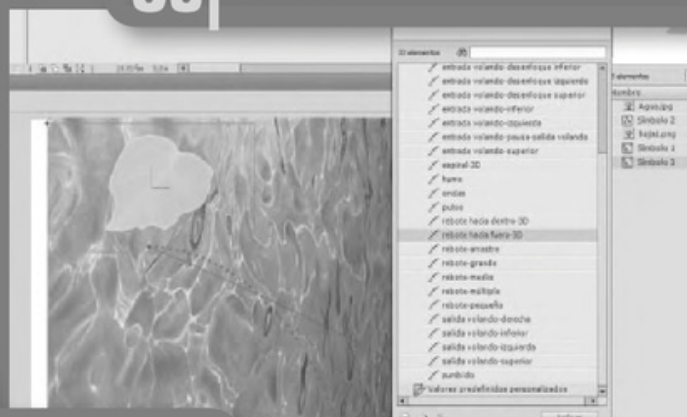
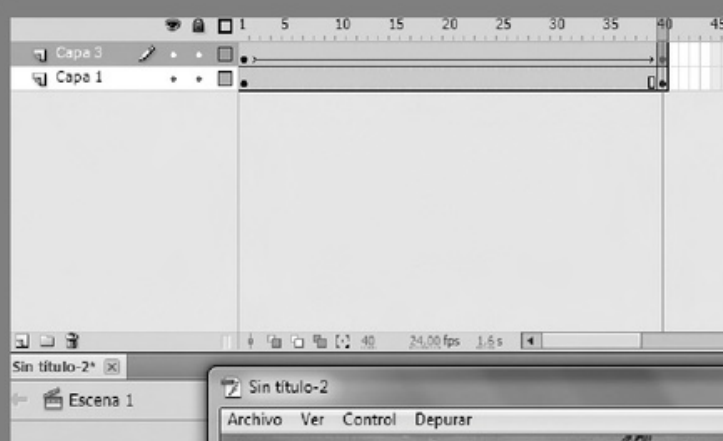
PASO | 01 Abrimos un documento nuevo y llevamos a la Biblioteca el fondo elegido para esta ocasión. Creamos un texto, desde el panel de **[Propiedades]** modificamos su tamaño, color, espaciado, etcétera, y lo convertimos en **[Símbolo/Clip de Película]**. Hacemos una capa "Fondo" y otra "Texto" para comenzar la animación, y colocamos el fondo por detrás del símbolo.

PASO | 02 Posicionándonos en el fotograma 1 de la capa "Texto", insertamos una interpolación de movimiento desde el botón contextual o desde **[Crear/Interpolación de Movimiento]**. Veremos que, automáticamente, se crean en la línea de tiempo los fotogramas necesarios. Estiramos arrastrando la capa "Fondo", para que acompañe a toda la animación.

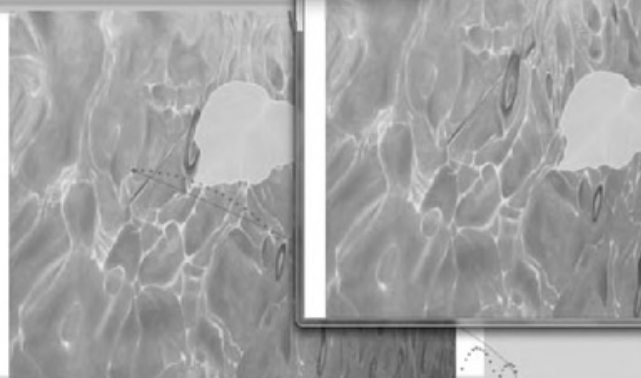
PASO | 03 En el último fotograma de la capa animada, llevamos el símbolo a la posición deseada. Aparecerá una línea para indicar el trayecto que recorre el texto. Recorriendo la línea, es posible mejorar y cambiar la ubicación de los puntos para dar otros efectos, modificar las propiedades y los colores con el editor de movimientos, y realizar otros cambios que nos permitan lograr una película más vistosa. A continuación, utilizaremos Motion Preset (interpolaciones de movimiento preconfiguradas).

PRESET

03 ANIMACIÓN

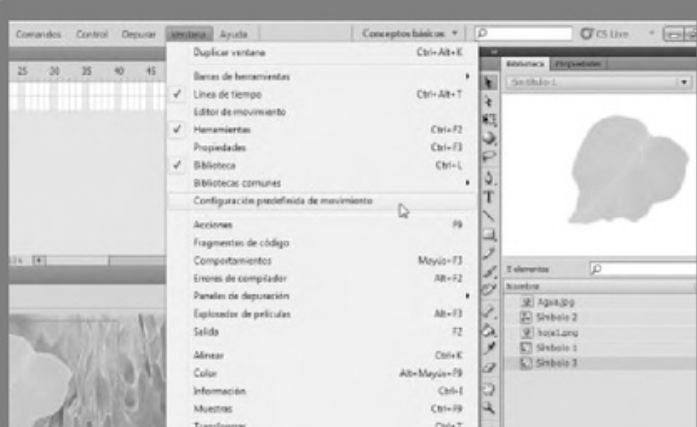


MOVIMIENTO 05



TRAYECTORIA 06

04 MOTION PRESET



PASO | 04 Haremos un efecto de hoja cayendo sobre el agua. Para esto, importamos las imágenes de fondo y hoja a la Biblioteca. Creamos las capas necesarias para contener estos elementos e incorporamos las imágenes a la escena, convirtiéndolas en movie clip. Seleccionamos el objeto y hacemos clic en el botón **[Aplicar]**, de **[Ventana/Configuración predefinida de movimiento]**. Veremos dos carpetas: **predefinidos**, con todas las animaciones disponibles; y **personalizados**, para futuras animaciones.

PASO | 05 Probamos libremente cada animación prediseñada haciendo un clic sobre su nombre en la carpeta de configuración predefinida de movimiento. En este ejemplo, utilizaremos la opción **rebote hacia afuera 3D**, que dará efecto de movimiento a la hoja. Vemos que se crean el símbolo y todos los fotogramas animados en la línea del tiempo. Estiramos la capa "Fondo" hasta el final.

PASO | 06 La línea punteada informa el trayecto que tendrá la hoja al caer; podemos corregir su dirección sobre el escenario, luego de ver la acción. Por último, probamos la película dando los ajustes finales a la animación. Cada objeto puede llevar solo una animación personalizada, y las animaciones con efectos 3D pueden ser solo clips de película.

RESUMEN

Hemos aprendido el método para crear y modificar la trayectoria de un objeto. También vimos cómo las interpolaciones de movimiento preconfiguradas pueden aplicarse a un objeto del escenario.

MOTION EDITOR

Desde el panel Motion Editor, podremos controlar todas las características de cualquier animación Motion Tween en forma precisa y puntual. Veamos cómo hacerlo.

Realizar una animación de tipo **Motion Tween** es muy sencillo. Manipulando adecuadamente el panel Motion Editor, lograremos resultados de gran calidad y podremos ajustar cada uno de los parámetros que intervengan. Para comenzar, debemos crear una animación sencilla de tipo Motion Tween. Si seleccionamos cualquier fotograma de ella y abrimos Motion Editor, yendo al menú **[Ventana]**, nos encontraremos con las siguientes opciones:

- **[Movimiento básico]:** desde aquí se puede modificar la posición en X e Y. También, la rotación en los ejes X, Y, Z.
- **[Transformación]:** aquí se encuentran los controles de sesgar en X, Y; escala en X, Y; efectos de color y filtros.

► **[Suavizado]:** desde esta opción controlaremos la aceleración o desaceleración y distintos tipos de efectos de animación.

Veamos ahora en detalle qué podemos hacer en cada caso.

MOVIMIENTO BÁSICO

Desde esta sección del panel podremos controlar la posición del movie clip ubicado en el escenario a lo largo de toda la animación, con el fin de modificar sus coordenadas X, Y, Z. Si movemos el clip en el escenario, notaremos que, en esta parte del panel, empiezan a modificarse las nuevas coordenadas que va ocupando el elemento. También podemos variar cualquiera de las coordenadas directamente con estos controles.

TRANSFORMACIÓN

Como su nombre lo indica, con esta opción podremos controlar todo lo referido a la apariencia visual del movie clip que estemos animando. Podemos escalarlo o sesgarlo, cambiarle el color, o aplicarle uno o varios filtros. Al igual que en el caso anterior, dichas modificaciones pueden hacerse desde el mismo panel o por medio del menú **[Herramientas]** de Flash.

SUAVIZADO

Desde aquí podremos controlar una de las opciones más potentes de esta herramienta. No solo tenemos la posibilidad de agregar algún efecto de aceleración sobre el tiempo total de la animación, sino que, si hacemos clic sobre el botón **[Agregar]**, también podremos incorporar diversos efectos de animación predefinidos y aplicarlos en forma general. Además, el efecto que agreguemos se habilitará para

usar en los paneles **[Movimiento Básico]** y **[Transformación]**. A su vez, podremos aplicarlo solo en el movimiento del eje X o en cualquiera de las propiedades que se estén modificando con la animación de manera individual.

EDITAR LAS CURVAS DE PROPIEDAD

Sobre el panel de Motion Editor veremos representados los fotogramas de la línea de tiempo en la que se generó la animación y, sobre ellos, distintas líneas y curvas que grafican las direcciones e intensidades del movimiento creado o de los valores de las propiedades modi-



CONOCIMIENTOS TÉCNICOS

Recordemos que, para mejorar y potenciar nuestro aprendizaje, desde www.redusers.com/cursos/webdesign podemos descargar los archivos editables de los proyectos desarrollados en esta colección. A partir de ellos, obtendremos ideas para uno propio o, directamente, podremos modificar los archivos para reutilizarlos. En el sitio también encontraremos **guías visuales**, un **glosario de términos** y una completa guía de preguntas frecuentes sobre la suite CSS. Además, el libro **Desarrollo y animación en ActionScript 3.0** nos ayudará a comprender los fundamentos de la programación en Flash.

ficadas. Cabe destacar que este método de control de animación es igual al que usan programas como After Effects y 3D Studio Max.

Estas curvas pueden cambiarse de manera muy similar a cuando dibujamos con la herramienta **[Pluma]**. Aclaremos que las curvas correspondientes al movimiento básico (las propiedades X, Y, Z) solo pueden ser editadas al mismo tiempo; no ocurre lo mismo en el caso de las de transformación, ya que podremos editar cada curva de propiedad por separado. Una vez definido un movimiento básico, podemos empezar a manipular las curvas, haciendo clic con el botón derecho del mouse en cualquiera de las rectas que lo grafiquen. Al hacerlo, aparecerá la opción **[Agregar fotograma clave]**; si la



elegimos, veremos que en la línea aparece un nuevo nodo que podremos mover con el puntero del mouse para producir distintos desplazamientos. Si volvemos a apretar el botón derecho del mouse sobre cualquiera de los nodos agregados, podremos quitarlo o generar un suavizado en el movimiento de las propiedades. Cuando añadimos nodos sobre las rectas que grafican las transformaciones, veremos que, además de poder agregarlos o quitarlos, y suavizarlos, también podremos dibujar las curvas que se van generando.

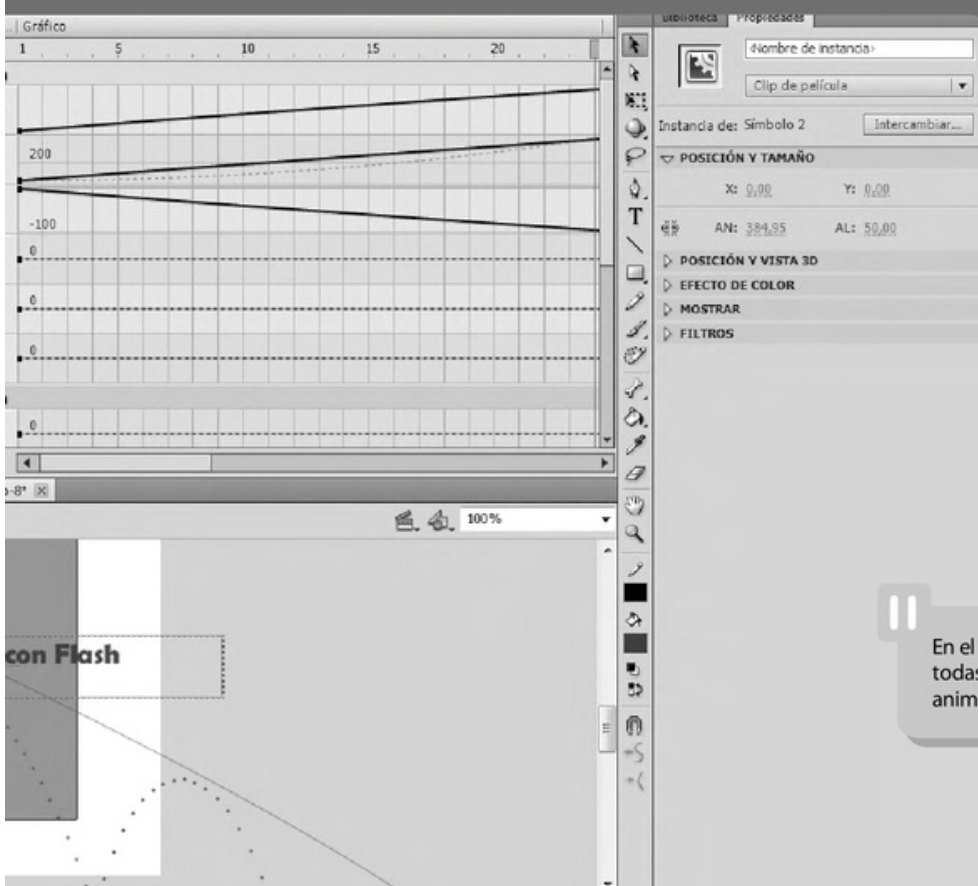
REMOVER Y SUAVIZAR

Si hacemos clic con el botón derecho del mouse sobre el nuevo nodo, en la recta de transformación veremos las siguientes opciones:

- **[Remover]**: permite quitar el fotograma clave.
- **[Suavizar hacia la derecha o hacia la izquierda]**: en ambos casos, desde el nodo se generará una pequeña curva en el sentido seleccionado y aparecerá un control de dirección e intensidad de dicha curva, similar al que se presenta cuando editamos el nodo de un trazado creado con la **[Pluma]**.
- **[Suavizar punto]**: esta opción es una combinación de las dos anteriores, ya que genera una curva completa hacia la derecha o la izquierda, con sus respectivos controles para dar dirección y aumentar o disminuir la intensidad con la que se aplicará este efecto.

Como pudimos ver, Motion Editor es un complemento muy versátil para efectuar animaciones. ◀

CON **MOTION EDITOR**
LOGRAREMOS EFECTOS
AVANZADOS DE ANIMACIÓN
CON **NIVEL PROFESIONAL.**

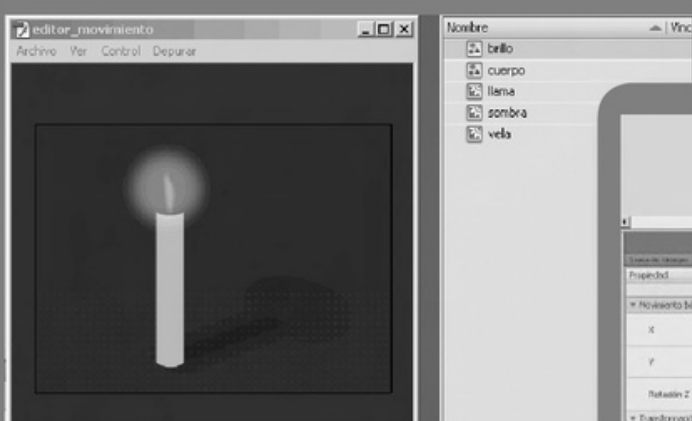


En el panel **Motion Editor** encontraremos todas las opciones para generar animaciones con gran detalle y control.

ANIMACIONES CON MOTION

PASO A PASO

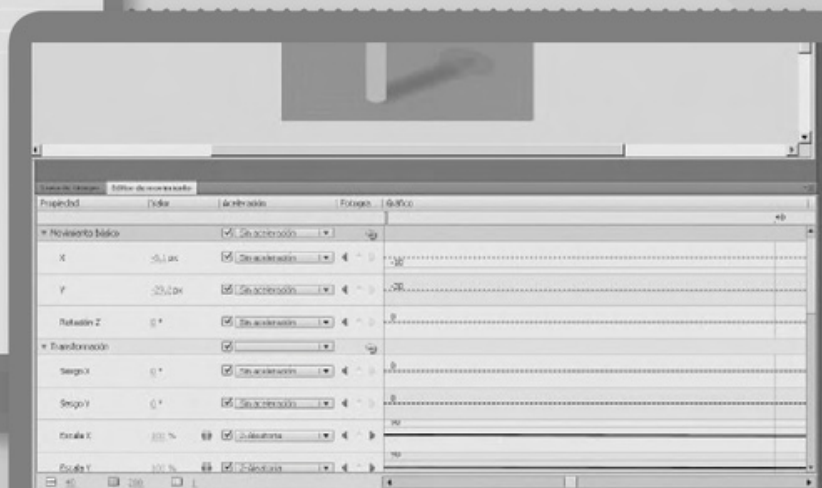
Con el panel Motion Editor lograremos animaciones de gran calidad y podremos ajustar cada uno de los parámetros que intervengan. Veamos cómo hacerlo.



01 | INTERPOLACIÓN

El editor de movimiento es una valiosa ayuda a la hora de visualizar todas las propiedades de un grupo de interpolación en la línea de tiempo, de un objeto interpolado o del trazado de movimiento en el escenario. Permite añadir precisión y detalle a partir de las curvas de propiedades y controlar el trabajo de diferentes maneras. ◀

PASO | 01 Tomaremos un ejemplo sencillo de interpolación de movimiento; en este caso, una vela que arde, y refleja sus movimientos con brillo y sombra. Tenemos creados los símbolos en la Biblioteca, el fondo y la animación, y esta se encuentra volcada en la línea del tiempo. Consta de las capas "cuerpo de la vela", "sombra", "brillo" y "llama", lo que da una imagen realista, con todos sus efectos.



PROPIEDADES | 02

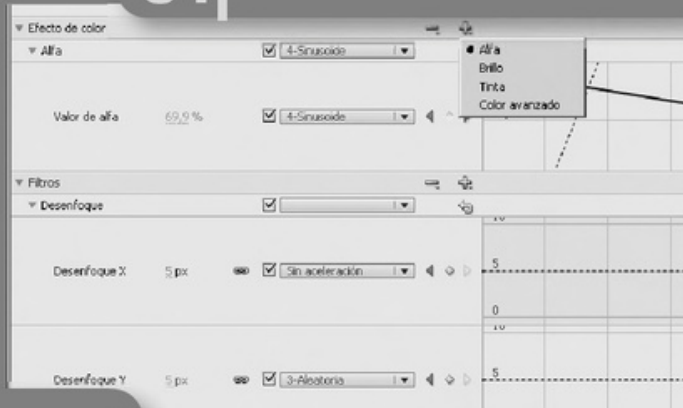
PASO | 02 Vemos la línea de tiempo de las animaciones y, para inspeccionar, seleccionamos la capa "llama".

La pestaña contigua a la línea de tiempo es el **[Editor de movimiento]**; nos dirigimos a ella para ver y ajustar todas sus propiedades. Podemos seguir la propiedad, el valor, la aceleración, el fotograma clave anterior y siguiente, quitar el fotograma clave o restablecer valores.

PASO | 03 A la derecha del editor de movimiento se muestran los valores en un gráfico bidimensional para cada propiedad. Cada gráfico representa el tiempo (horizontalmente) y la variación de los valores de la propiedad (verticalmente). En este caso, se ve fácilmente la propiedad de los suavizados. La curva de línea discontinua indica el efecto de la aceleración en los valores de propiedad.

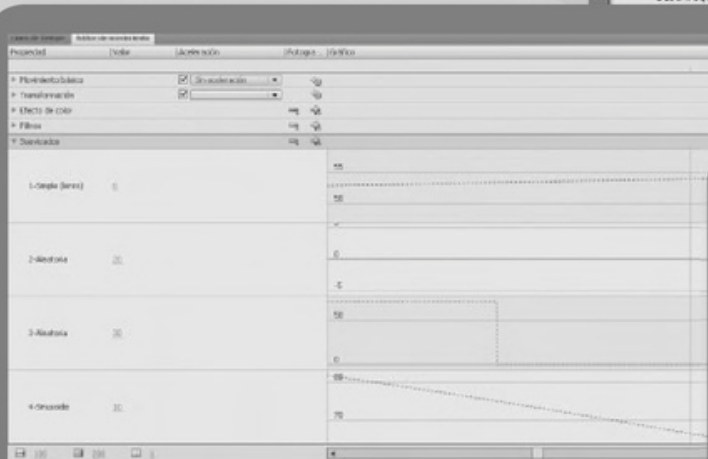
EDITOR

04 | AJUSTE



CAMPOS Y EFECTOS | 05

03 | SUAVIZADOS



PASO | 04

Podemos ajustar las propiedades en el editor situándonos en el triángulo ubicado junto a la solapa **[Propiedades]**.

Para controlar el número de fotogramas de interpolación que se muestran en el editor de movimiento, especificamos la cantidad en la solapa **[Fotogramas visibles]** de la parte inferior del editor.

PASO | 05

Es posible ajustar el tamaño de las vistas contraída y expandida en las solapas **[Tamaño de gráfico]** y **[Tamaño de gráfico expandido]**, en la parte inferior. Para añadir un nuevo efecto de color, filtro o aceleración, nos posicionamos en el botón **[Añadir]** de la fila de la categoría y seleccionamos el elemento **[Añadir]**. Este aparecerá inmediatamente en el editor.

PASO | 06

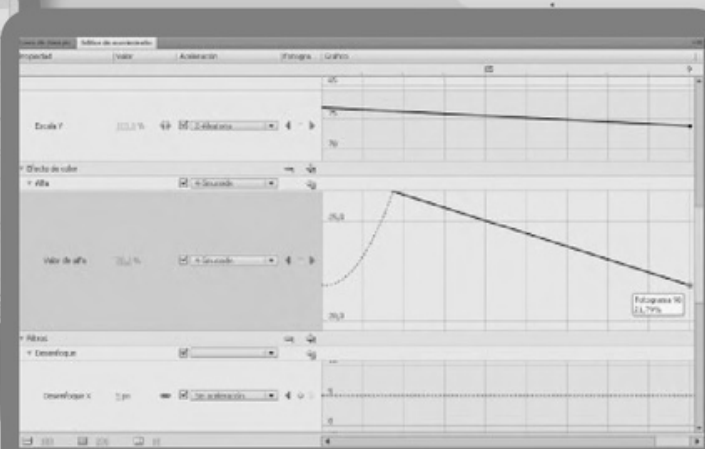
Con el editor de movimiento es posible controlar con precisión la forma de todas las curvas de propiedad de una

interpolación. Editamos la curva de cada gráfico con ayuda de los controles estándar de Bezier, al igual que hacemos con los trazos de la herramienta **[Selección]** o **[Pluma]**: hacia arriba aumenta el valor de la propiedad, y hacia abajo disminuye.

RESUMEN

Como vimos, Motion Editor es el complemento ideal para realizar animaciones. Si logramos familiarizarnos con él, podremos obtener resultados avanzados.

EDITAR CURVAS | 06



CINEMÁTICA INVERSA

Con cinemática inversa (IK, *inverse kinematic*) animaremos objetos en forma individual o en conjunto, si los relacionados entre sí son huesos.

IK en Flash

El uso de IK en Flash abre muchas posibilidades para agilizar el trabajo en animaciones o juegos. Además de tener librerías nativas de ActionScript, podemos encontrar varias desarrolladas por otros autores con propiedades interesantes como la incorporación de efectos físicos y partículas, como www.fisixengine.com.

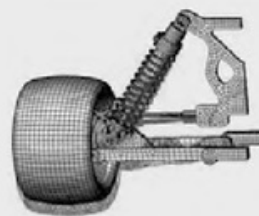
¿Dónde se aplica IK?

► Videojuegos

Una aplicación frecuente de IK es en videojuegos para animar y controlar personajes. Como podemos ver en el famoso juego de la viborita también podríamos aplicar este método en forma sencilla. www.markfennell.com/flash/snake

► Simulación mecánica

Recreación de simulaciones sobre el comportamiento mecánico de objetos, tales como máquinas, brazos mecánicos, y otros.



► Sitios web

Muchas veces podremos encontrarnos con sitios web en donde se aplica este recurso, ya sea en sus interfaces de navegación, como así también en sus elementos de diseño. www.fisixengine.com/demos/mona1.html

01 HUESO

Cada hueso está relacionado en forma jerárquica con el hueso anterior de la cadena cinemática, de modo que se generan relaciones "padre/hijo" entre ellos.

02 CADENA CINEMÁTICA

La cadena cinemática está compuesta por huesos vinculados entre sí por medio de sus respectivos pivotes o puntos de unión.

03 ÁNGULO DE ARTICULACIÓN

Cada articulación dispone de parámetros para limitar el ángulo de rotación del hueso en su pivote.

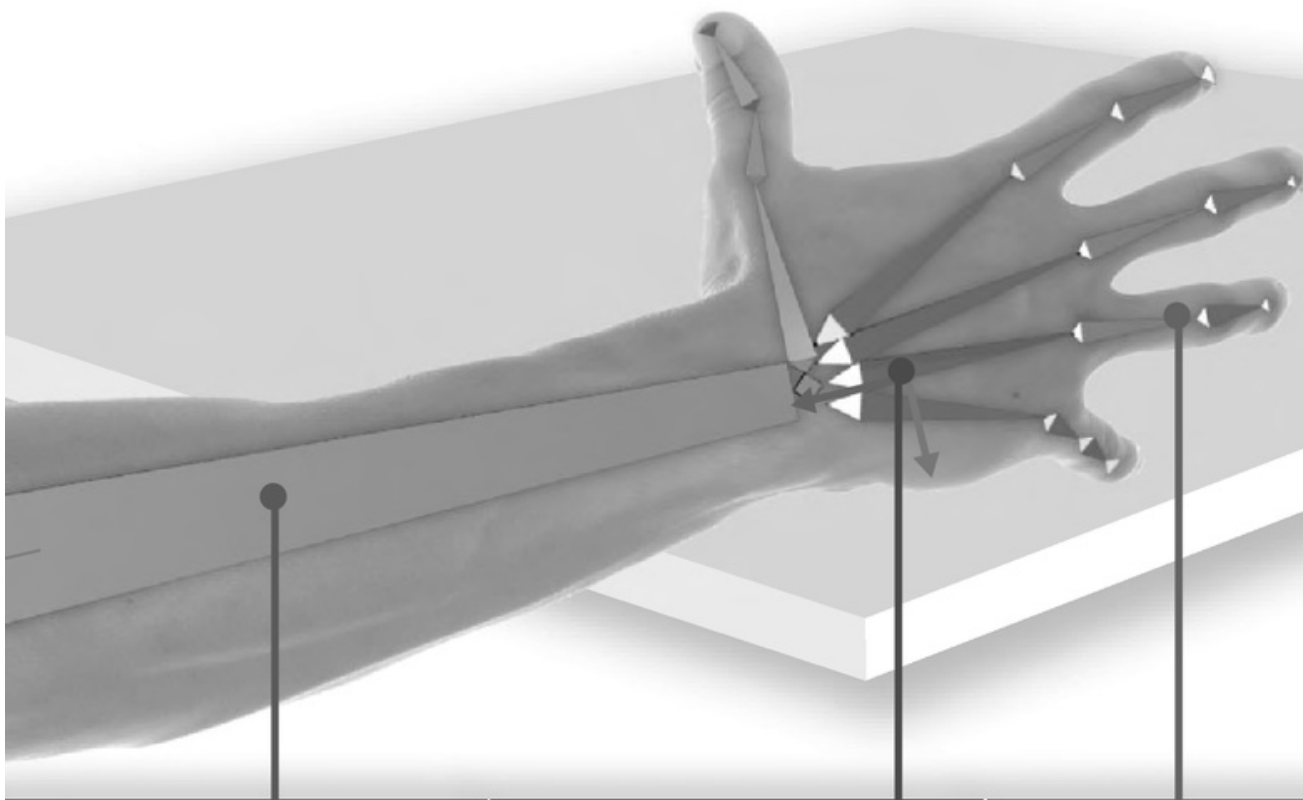
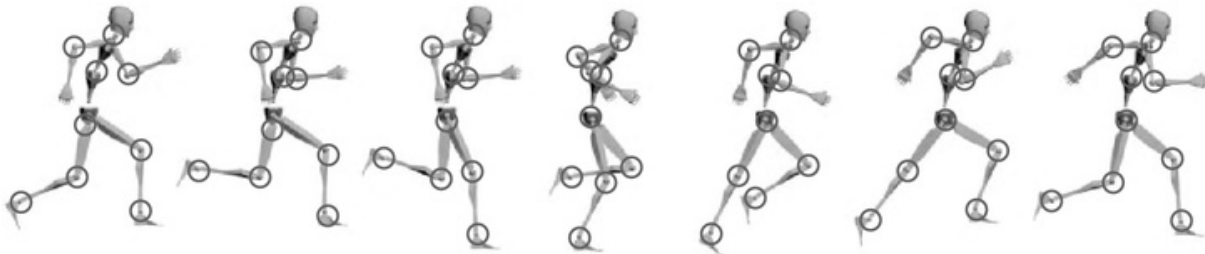


Cinemática inversa

Utiliza efectores de IK o pivotes para mover y rotar a la vez toda una cadena jerárquica. Para hacerlo, calcula la distancia con el origen de la cadena y rota automáticamente los huesos intermedios cuando se modifica la distancia de dicho origen.

► Poses

Se llama de esta manera a los diferentes estados en la animación con cinemática inversa, que serían los equivalentes a los keyframes de las interpolaciones.



04 ESQUELETO IK

El esqueleto IK es el conjunto de huesos y cadenas cinemáticas que forman la estructura.

05 ARTICULACIONES

Cada movimiento de IK está definido por los pivotes que unen los huesos.

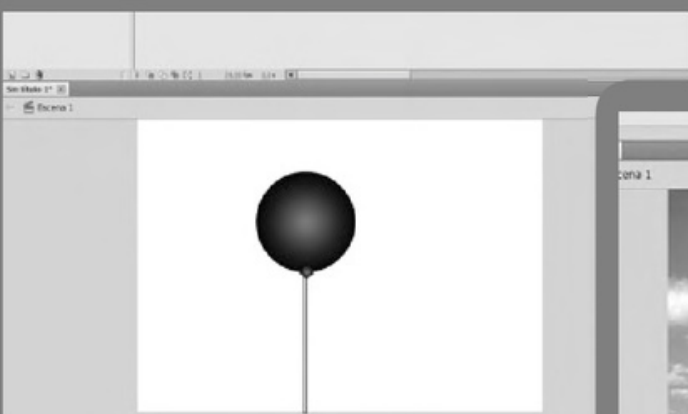
06 PIVOTE

Por medio de los pivotes determinamos el lugar en donde se genera el movimiento y las uniones.

COMO ANIMAR HUESOS CON

PASO A PASO

Después de haber aprendido la noción de cinemática inversa, aplicaremos estos conocimientos en una práctica que consiste en animar huesos con muelles.



01 | INTERPOLACIÓN



PROPIEDADES | 02

Vamos a crear una forma con relleno en el escenario para darle movilidad en tiempo de creación. En este caso, haremos que un globo cobre vida y se mueva con el viento, ayudados por la herramienta **[Huesos]** y mejorándolo con opciones de la propiedad **[Muelle]**. ◀

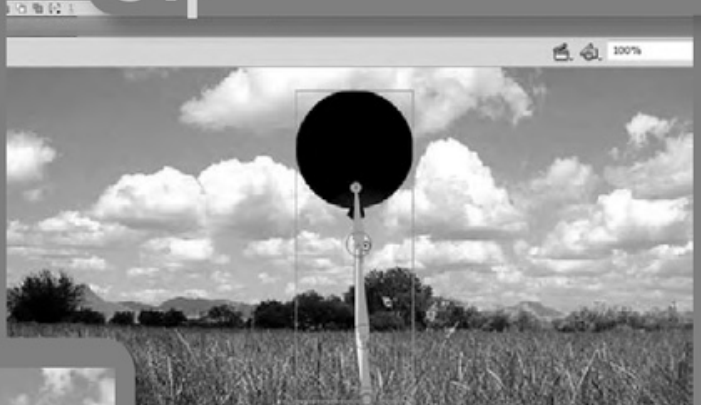
PASO | 01 Creamos un nuevo documento, dibujamos un globo y lo pintamos con la herramienta **[Óvalo]**, le agregamos un hilo con **[Rectángulo]** y lo coloreamos. Importamos a la Biblioteca el fondo deseado, en este caso, un cielo. Lo insertamos en una nueva capa y seleccionamos toda la forma teniendo cuidado de incluir tanto los colores como los trazos. Escalamos toda la forma según el tamaño deseado.

PASO | 02 Con la herramienta **[Huesos]**, creamos el esqueleto de la animación. Al trabajar con una forma, podemos darle tantos huesos como queramos para lograr mejores efectos. Para esto, arrastramos y soltamos el hueso a lo largo del objeto por animar, con lo cual conseguimos varios huesos en la forma. Flash la convierte en un objeto de **forma IK** y lo mueve a una nueva capa de la línea de tiempo llamada **"pose"**.

PASO | 03 Cada capa de pose puede contener únicamente un esqueleto. El primer hueso de un esqueleto es la raíz. Nos ubicamos en la capa "pose" y creamos, en el fotograma 45, un fotograma pose. Extendemos también el fondo y, ubicados al final de la animación, desplazamos el esqueleto para lograr el movimiento deseado. Aplicamos **[Muelle]** y probamos la película. Ahora trabajaremos en tiempo de ejecución. Sobre el mismo diseño de globo, animaremos los huesos con ActionScript 3.0.

MUELLES

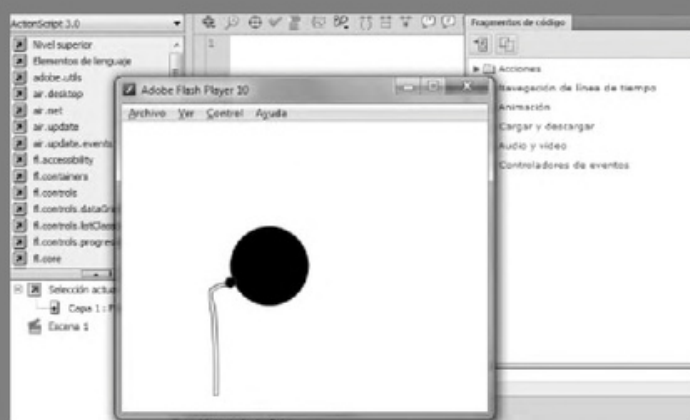
04 | AJUSTE



CAMPOS Y EFECTOS | 05



03 | SUAVIZADOS



EDITAR CURVAS | 06

PASO | 04 Los esqueletos IK conectados a formas o a instancias de clips de película se pueden controlar con ActionScript 3.0, no así aquellos conectados a gráficos o a instancias de símbolos de botón. Solo es posible controlar con ActionScript 3.0 los esqueletos con una sola pose. Los que tienen más de una únicamente pueden controlarse en la línea de tiempo.

PASO | 05 Con la herramienta **[Selección]**, tomamos un fotograma de una capa de pose que contenga un esqueleto. El inspector de propiedades muestra las correspondientes al esqueleto. En el inspector de propiedades, seleccionamos **[Tiempo de ejecución]**, dentro del menú **[Tipo]**. Ahora podemos manipular la jerarquía con ActionScript 3.0 en tiempo de ejecución.

PASO | 06 De manera predeterminada, el nombre del esqueleto en el inspector de propiedades es el mismo que el de la capa de pose. Utilizamos esta denominación para hacer referencia al esqueleto en ActionScript. En nuestro caso, trabajaremos con el nombre "globo". Probamos la película y vemos que, si hacemos clic y arrastramos en los sectores donde están las uniones de los huesos, moveremos el globo.

RESUMEN

En este paso a paso hemos aprendido a crear una forma con relleno en el escenario, y a otorgarle movilidad en tiempo de creación y en tiempo de ejecución. Sobre el mismo diseño, animamos los huesos con ActionScript.

ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué son los axiomas de animación?

02 ¿Cuáles son las ventajas de **Classic Tween**?

03 Mencione algunas de las nociones básicas sobre animación.

04 ¿Para qué sirve **Motion Tween**?

05 ¿Qué es un **Motion Preset**?

06 Describa qué podemos encontrar en el panel **Motion Editor**.

07 ¿Qué tipo de efectos podemos lograr con **Motion Editor**?

08 ¿Qué es la cinemática inversa?

09 ¿Dónde se aplica la cinemática inversa?

10 Describa cómo podemos animar huesos en Flash.

PRÁCTICAS

01 Utilice **Classic Tween** y las guías de movimiento.

02 Utilice **Motion Preset**.

03 Genere una animación con **Motion Editor**.

04 Aplique la cinemática inversa en Flash.

05 Anime huesos con muelles.



Animaciones avanzadas

- ▶ EFECTOS FÍSICOS
- ▶ LIBRERÍAS DE ANIMACIÓN
- ▶ TWEENMAX Y TWEENLITE
- ▶ CÓMO APLICAR EFECTO A UNA IMAGEN
- ▶ SISTEMA DE PARTÍCULAS
- ▶ CÓMO ANIMAR EL LAYOUT DE NUESTRO SITIO



EFFECTOS FÍSICOS

En este apartado conoceremos cuáles son los principios elementales de la física que debemos tener en cuenta para realizar excelentes animaciones mediante el uso del código.

Al mover objetos mediante código, el siguiente paso es realizar un efecto de física, como rebote, elasticidad o gravedad, entre otros. Para estos casos, no hay más opción que recurrir a las matemáticas y transcribir aquellas operaciones a funciones que simulen dicha realidad en un campo virtual. Los efectos físicos sobre un objeto son innumerables, como el movimiento lineal u ondulatorio, ya sea con aceleración o desaceleración en cualquiera de sus etapas inicial o final. Tampoco podemos dejar de lado todas las variantes que se pueden conseguir sumando gravedad y fricción. Es por eso que analizaremos los casos más utilizados de la física, que serán los más recurrentes a la hora de realizar animaciones.

ACELERACIÓN

La aceleración de un objeto puede ser lineal y proporcional a la velocidad a la que se realiza el efecto. Un claro ejemplo de aceleración lineal es el que se muestra en el **Código 2**. Para este ejemplo, se re-

CÓDIGO 2

```

meli_mc.addEventListener(Event.ENTER_FRAME, EventEnterFrame);
function EventEnterFrame(e:Event):void {    meli_mc.x += 5;
}

```

quiere un clip de película en el escenario con el nombre de instancia "meli_mc". Aquí ya tenemos una simple interpolación, donde el objeto se mueve linealmente y con una velocidad proporcional por cada pasada del cabezal. Pero seguramente que la idea es lograr una aceleración y una desaceleración del objeto, ya sea al principio, al final o en ambas etapas.

Por ejemplo, si queremos que un objeto se desplace a un punto indicado inicialmente rápido, y se detenga en forma gradual, tendríamos que calcular la distancia entre ambos puntos, dividirla por un factor y, a ese resultado, sumar la posición actual del objeto. Luego, deberíamos dividir ese segmento otra vez por el mismo valor y sumar la posición actual del objeto. Este proceso debe realizarse

sucesivamente hasta que no quede más distancia. El ojo percibirá un efecto de **desaceleración** al final de la interpolación (**Código 3**). En el **Código 3** agregamos dos variables nuevas: una es el punto de destino final, que es el ancho del escenario menos el del objeto; otro es el factor de velocidad: cuanto mayor sea el número, más demostrará el efecto en completarse.

ELASTICIDAD

Este tipo de interpolación se destaca por el **rebote** en la animación, que puede ser al inicio, al final o en ambos extremos. El rebote está dado por la multiplicación de los valores relativos de posición por un factor asignado. Luego, por cada ciclo se repiten los pasos de aceleración.

GRAVEDAD

Este efecto está dado por varios factores que deben igualar la realidad de la vida cotidiana. Uno es el factor de fuerza y peso del objeto. También interviene un valor que representa el rebote del objeto, para darnos una mejor experiencia sobre el efecto buscado.

Por lo tanto, en este caso tendremos que declarar los valores iniciales, como fuerza y peso; los finales de tope o piso; y también, el factor de rebote. Sin dudas que, si lo analizamos, veremos cómo, desde la

CÓDIGO 3

```

var destinoX:uint = stage.stageWidth - meli_mc.width;
var velocidad:uint = 5;
meli_mc.addEventListener(Event.ENTER_FRAME, EventEnterFrame);
function EventEnterFrame(e:Event):void
{
    meli_mc.x += (destinoX - meli_mc.x)/velocidad;
}

```


animación simple lineal del primer ejemplo, fuimos agregando efectos físicos con resultados muy buenos, sin incrementar la cantidad de frames de la película.

Como puede observarse en el

Código 4, se declaran nuevas variables, como velocidad, rebote y gravedad.

La **velocidad** se va incrementando gradualmente mientras la animación se completa.

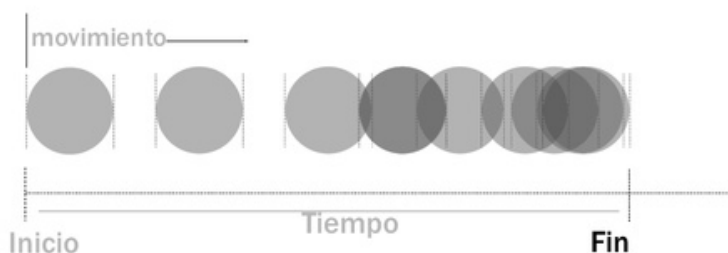
El **rebote** es el factor que se resta de la distancia por cada ciclo de movimiento. Debe ser menor a 1 para que el objeto dé la sensación de pérdida de fuerza. Si dejamos el valor en 1, el objeto no dejará de rebotar.

En cambio, si lo incrementamos, rebotará cada vez más lejos del límite o piso indicado. La variable **gravedad** se encarga de manejar la velocidad con la cual el objeto es atraído al punto final. Estos tres principios básicos de la física son los que más se requieren a la hora de realizar animación por código; es decir, son los más recurrentes para lograr otros efectos. Mezclando cualquiera de ellos o modificándolos levemente, es posible obtener entornos muy realistas de simulación. No hay límites a la hora de simular fórmulas físicas en ActionScript, solo se requiere un poco de matemáticas y lógica. ◀

CÓDIGO 4

```
var destinoY:uint = stage.stageHeight - peli_mc.height;
var velocidad:Number = 0;
var rebote:Number = 0.9;
var gravedad:Number = 5;
peli_mc.addEventListener(Event.ENTER_FRAME, EventEnterFrame);
function EventEnterFrame(e:Event):void
{
    velocidad = velocidad + gravedad ;
    peli_mc.y += velocidad/5;
    if (peli_mc.y > destinoY) {
        peli_mc.y = destinoY ;
        velocidad *= -rebote ;
    }
}
```

La velocidad se reduce cuanto más nos acercamos al destino final



Quando el objeto está más próximo al destino final, su velocidad se reduce proporcionalmente.

```
function EventEnterFrame(e:Event):void
{
    Peli_mc.x+= (destinoX- Peli_mc.x) / velocidad;
}
```

Aunque contemos con un solo fotograma, el cabezal pasa por él cuantas veces indiquemos en los frames

Línea de tiempo con 1 frame

ENTER_FRAME
En el mismo frame recorre y rebobina.
Por cada pasada dispara un evento



Fotograma activo

Fotogramas

El movimiento lineal, cualquiera sea su dirección, es proporcional e igual a su velocidad de desplazamiento.

Se mantiene velocidad constante

movimiento



```
Peli_mc.addEventListener(Event.ENTER_FRAME, EventEnterFrame);
function EventEnterFrame(e:Event):void
{
    Peli_mc.x+=5;
}
```

LIBRERÍAS DE ANIMACIÓN

En este apartado aprenderemos a implementar los paquetes para aplicar el movimiento y manipular los eventos sobre los objetos que son animados desde el código. Veamos qué son las librerías y cómo utilizarlas.

La animación mediante interpolación de movimiento es una tarea de tipo manual, que presenta cierta dificultad si es que queremos animar objetos que son cargados dinámicamente o cuyas propiedades son relativas. En estos casos, se implementan paquetes para manejar movimiento, eventos y física sobre objetos desde el código.

SOBRE TWEEN

Entre las clases nativas de Adobe Flash, contamos con una llamada Tween, que pertenece al paquete "fl.transitions". Dicha clase hereda

de **[EventDispatcher/ Object]**.

Es importante aclarar que "EventDispatcher" se ocupa de que los elementos cargados en la lista de visualización puedan manejar eventos. La clase Tween da la posibilidad de animar objetos y editar sus propiedades, como nivel de opacidad (alpha) o escala (scaleX), utilizando código. También permite manipular el tipo de física para un objeto determinado, como aceleración, desaceleración o rebote.

IMPLEMENTACIÓN BÁSICA

Para realizar un ejemplo de la implementación básica, se requiere tener en el es-

cenario un movie clip con el nombre de instancia **pel_i_mc**. Tenemos que hacer clic en el primer fotograma y colocar el siguiente código:

```
import fl.transitions.  
Tween; import  
fl.transitions.easing.  
Elastic;  
var miTween:Tween  
= new Tween(peli_  
mc,"y",Elastic.  
easeOut,0,100,1,true)
```

IMPORTAR LA CLASE TWEEN

Para importar con la clase Tween, debemos escribir el siguiente código:

```
import fl.transitions.  
Tween;
```

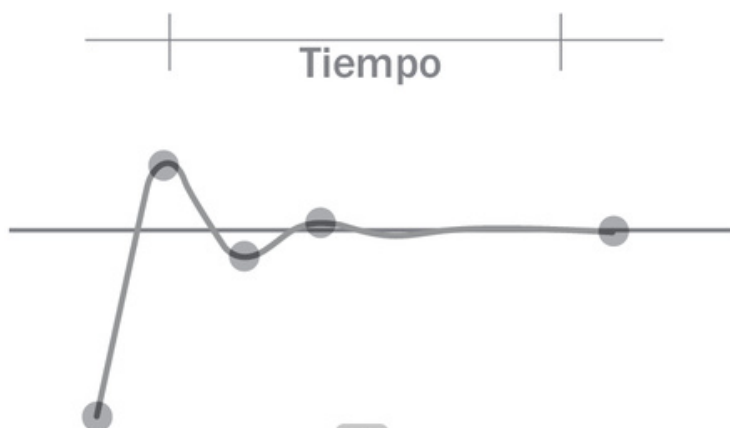
Esta es la clase que nos da todas las posibilidades de mover y editar propiedades de un objeto, como un movie clip.

```
import fl.transitions.  
easing.Elastic;
```

Importamos la clase de suavizado de movimiento. Cuando movamos un objeto con ella, podremos indicar com-

Movimiento Elastics

```
vas mi tween:tween=new Tween(peli_mc,Elastic.easeInOut, 0, 100, 1, true)
```



Gráfica de simulación de movimiento al inicio utilizando la clase Tween.

portamientos físicos como aceleración, rebote o desaceleración, tanto al inicio como al final de la animación.

```
var miTween:Tween
= new Tween(peli_
mc,"y",Elastic.
easeOut,0,100,1,true)
```

Finalmente, creamos el objeto pasando como parámetros los valores requeridos. En este caso, el objeto se llama **miTween** y, al llamar el constructor (new), pasamos los siguientes parámetros:

► **pel_mc** (objeto): nombre de instancia de variable del objeto cuya propiedad vamos a modificar. Es importante recordar que solo se puede afectar a objetos que hereden de la clase InteractiveObject, como MovieClip, Sprite, TextField, Stage, Loader y SimpleButton.

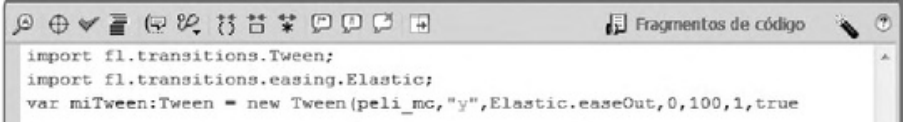
► **"y" (String)**: cadena de texto que representa la propiedad por modificar; en este caso, la posición en el eje Y.

► **Elastic.easeOut (Function)**: es el tipo de suavizado del movimiento. Si queremos que el movimiento se comporte como un elástico y rebote al final de la animación, esta es la mejor opción, o en su defecto, podemos usar las siguientes: Back, Bounce, Elastic, None, Regular y Strong, que debemos importar si necesitamos disponer de ellas. Es importante aclarar que al final de la línea de código se agregó un asterisco (*) para indicar que se deben cargar todos los paquetes de ese directorio.

CÓDIGO 5

```
import fl.transitions.Tween;
import fl.transitions.easing.Elastic;

var miTween:Tween = new Tween(peli_mc,"y",Elastic.
easeOut,0,100,2,true);
pel_mc.addEventListener(MouseEvent.CLICK,onClick);
function onClick(e:MouseEvent):void
{
    miTween.start();
}
```



Para el ejemplo de implementación básica necesitamos un movie clip en el escenario.

```
import fl.transitions.
easing.*;
```

A estas propiedades de aceleración les sumamos el suavizado que queramos: **easelIn**, **easeOut** y **easelInOut**.

- **0 (Number)**: valor inicial o comienzo de la propiedad "y".
- **100 (Number)**: valor final/máximo de la propiedad "y".
- **1 (Number)**: duración de la transición.

► **True** (Boolean, por defecto, false): es el indicador adecuado si vamos a sincronizar la transición con segundos como unidad de tiempo, o con los fotogramas. **True** permite utilizar una medida de tiempo en segundos. **False** se usa para utilizar una medida de tiempo expresada en fotogramas.

En el ejemplo anterior podemos ver que, inmediatamente, cuando compilamos el SWF se ejecuta la animación del movie clip en el eje Y. Si quisiéramos controlar dicho movimiento desde un evento del mouse, tendríamos que utilizar el método **start()**; para dar comienzo a la interpolación cuando hacemos clic en el movie clip (**Código 5**). Ahora, al hacer clic en el movie clip, ejecutamos la interpolación mediante el método **start()**. En este sentido, los métodos más comunes que podemos utilizar son los siguientes:

- **start()**: inicializa la interpolación.
- **stop()**: detiene la interpolación.
- **resume**: reanuda la interpolación donde se detuvo.
- **forward**: adelanta la interpolación hasta el final.
- **rewind**: retrocede la interpolación al valor inicial.

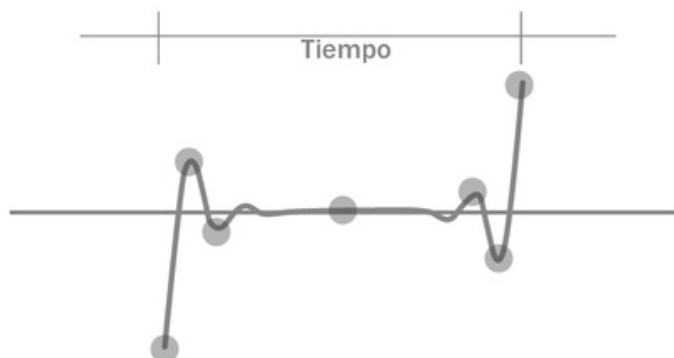
También es importante recordar que la clase Tween cuenta con detectores de eventos que nos facilitarán la tarea de animar objetos y desencadenar acciones, según los estados de estos. ◀

Movimiento lineal

Se mantiene velocidad constante

Suavizado easelInlut

Movimiento Elastics
var mi tween:tween=new Tween(peli_mc,Elastic.easeInOut, 0, 100, 1, true)



Gráfica de simulación de movimiento al inicio y al final empleando la clase Tween.

TWEENMAX Y TWEENLITE

Desde que Flash es un lenguaje de programación orientado a objetos, se incrementó sustancialmente la disponibilidad de clases personalizadas. En este caso, contamos con la clase Tween. Veamos cómo utilizarla.

Actualmente, también podemos contar con clases personalizadas como **TweenMax** o **TweenLite**. Estas no solo tienen más propiedades, suavizados y efectos, sino que también son ideales para controlar objetos diferentes de los heredados por **InteractiveObject**, como sonido o fotogramas. Una vez que descargamos la clase y la importamos a **ActionScript**, podremos implementar su uso de la siguiente manera:

```
import com.greensock.*;
import com.greensock.
easing.*; TweenLite.
to(peli_mc, 1, {y:225,
ease:Elastic.easeOut});
```

Como puede observarse, básicamente, es lo mismo que la clase Tween de Adobe, pero con la particularidad de que solo indicamos la posición final del objeto utilizando el método **"to"**. La estructura de uso es mediante una clase estática **"TweenLite"** o **"TweenMax"**; utilizamos el método **"to"** para indicar el movimiento y las propiedades. En los paréntesis indicamos, primero, el objeto

por mover y el tiempo en segundos (por defecto), excepto que entre las llaves pongamos **useFrames: true**, en cuyo caso la interpolación se completará teniendo en cuenta la cantidad de fotogramas. Con las llaves (**{}**) señalamos las propiedades y los métodos que se van a utilizar, separando los grupos con coma (,).

```
TweenLite.to( objeto,
tiempo {métodos} )
```

Los **easing** también son similares a la clase Tween de Adobe, con la diferencia de que contamos con una cantidad superior de movimientos. En este caso, utilizamos un movimiento elástico al final de la animación. Al importar todos los easings, tendremos disponibles los siguientes movimientos: Back, Bounce, Circ, Cubic, CustomEase, Elastic, FastEase, Linear, Quad, Quart, Quint y Strong, con sus respectivos comportamientos (easeOut, easeIn, easeInOut). Recordemos que el movimiento va indicado dentro de las llaves. La clase utiliza esta nomenclatura para indicar el método y el valor. Si quisiéramos determinar una demora antes de iniciar la animación, sería así: **{delay:1}**

```
TweenLite.from(peli_mc,
1, {y:225, ease:Elastic.
easeOut,delay:1});
```

Podemos agregar cuantos métodos queramos, separados por coma (,); por ejemplo: {y:10,x:45,delay:10,onComplete:listen}. Esta clase también cuenta con una

opción novedosa para indicar la posición final del objeto utilizando su ubicación actual, a diferencia del ejemplo anterior. En vez de utilizar el método **"to"**, empleamos **"from"**. El valor 225 del eje Y ahora no es la posición final, sino la inicial.

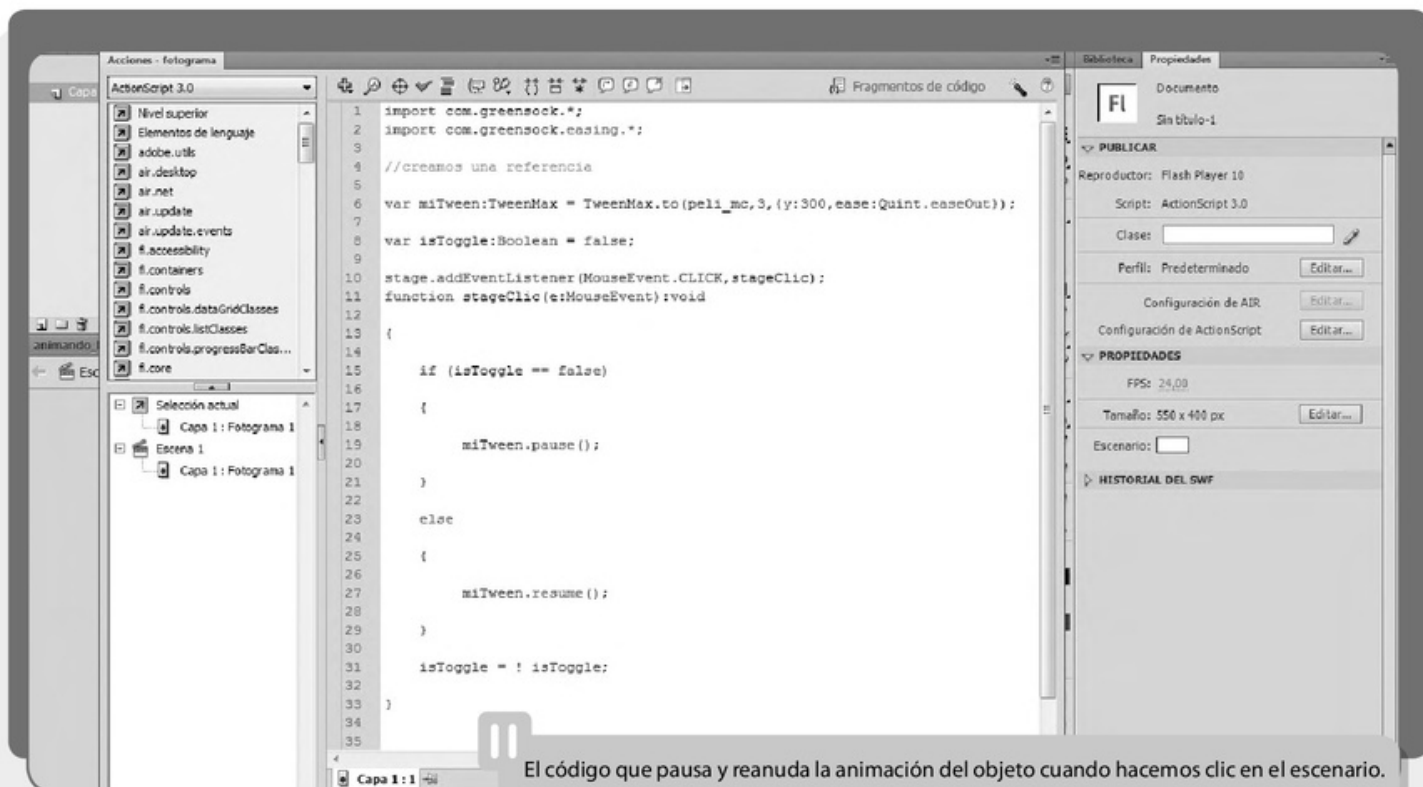
```
import com.greensock.*;
import com.greensock.
easing.*; TweenLite.
from(peli_mc, 1, {y:225,
ease:Elastic.easeOut});
```

CONTROLES

TweenMax, o su versión Lite, dispone de métodos para controlar el estado del movimiento mientras este se realiza. Veamos cuáles son:

- **pause()** y **resume()**: pausan o reanudan la interpolación donde se encuentre actualmente.
- **Reverse (Boolean)**: indica que la interpolación debe ir en reversa. Si usamos **True**, significa que se aplicará el mismo tiempo para reversa que el tiempo inicial.
- **UseFrames (Boolean)**: indicamos **True** para utilizar frames como unidad de tiempo, y no, segundos.
- **Delay(Number)**: segundos de demora antes de realizar la animación.

Veamos ahora cómo utilizar los método pause y resume. Para este tipo de interpolación es importante crear una variable con la referencia al tween. Este ejemplo pausa y reanuda la animación del objeto cada vez que hacemos clic en el escenario.



CONTROLADORES DE EVENTOS

Los eventos están relacionados con una función, siguiendo internamente los principios de la clase **Event** y sus **listeners** correspondientes. Asimismo, la nomenclatura es muy fácil e intuitiva. En el ejemplo del siguiente código se agrega un escuchador para cuando se complete la interpolación mediante el método **incomplete**.

```
TweenMax.to(peli_mc, 1,
{y:300, ease:Quint.easeOut, onComplete:seCompleto});
function
seCompleto():void
{
    trace("Interpolación completa");
}
```

animación, envía parámetros a la función relacionada.

- **onComplete**: ejecuta un evento cuando la animación finalizó.
- **onCompleteParams**: cuando finaliza la animación, envía parámetros a la función relacionada.
- **onRepeat**: ejecuta un evento cada vez que la animación inicia un ciclo.
- **onReverseComplete**: ejecuta un evento cuando el movimiento en reversa llega a su fin.

La clase cuenta con muchos más detectores de eventos, y estos, en su mayoría, pueden recibir parámetros como el ejemplo de **"onCompleteParams"**. Cabe destacar que no es posible usar **onComplete** y **onCompleteParams** al mismo tiempo para detectar y enviar parámetros. Simplemente, debemos recurrir a **onCompleteParams** para detectar el final de la animación y, entre paréntesis, enviar los valores como parámetros, como se muestra el código:

```
import com.greensock.*;
import com.greensock.
easing.*; TweenMax.
to(peli_mc, 1, {y:300,
ease:Quint.easeOut, o
nCompleteParams:seCom
pleto (4)}); function
seCompleto(id:uint):void
{
    trace("parametro
pasado: "id);
}
```

En conclusión, la clase Tween de Adobe resulta ideal para interpolaciones básicas o iniciales. En contrapartida, podremos explotar mejor el código, tanto con interpolaciones y efectos con paquetes personalizados, con TweenMax. El uso al momento de programar determinará cuál será el más adecuado. ◀

EVENTOS QUE PODEMOS UTILIZAR

A continuación, detallamos una lista de los eventos más importantes que podemos aplicar:

- **onStart**: ejecuta un evento cuando comienza la animación.
- **onStartParams**: cuando comienza la

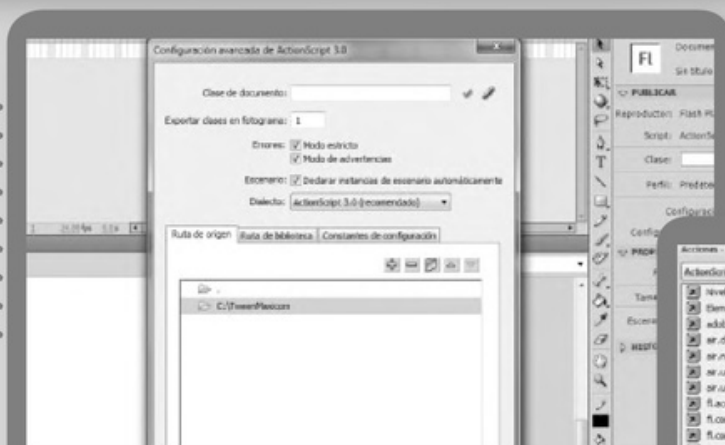
NO ES POSIBLE USAR
ONCOMPLETE Y ONCOMPLETEPARAMS
AL MISMO TIEMPO PARA
DETECTAR Y ENVIAR PARÁMETROS.



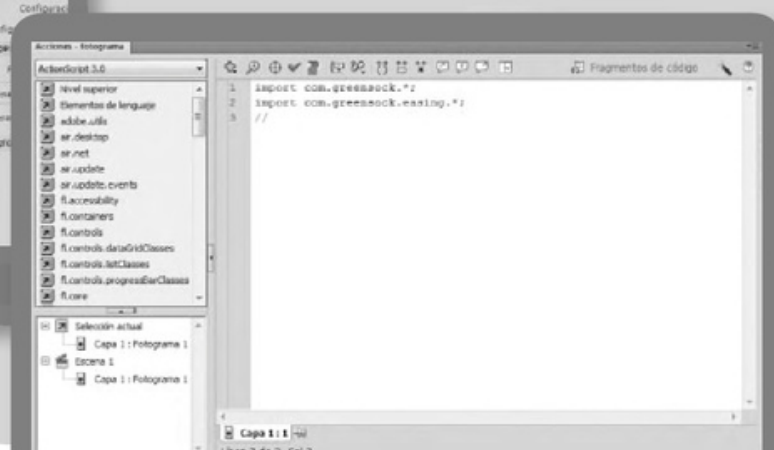
CÓMO APLICAR EFECTO A UNA

PASO A PASO

En este apartado aprenderemos a cargar una imagen y veremos el método que nos permitirá interpolar un efecto desde ActionScript 3.



01 DESCOMPRIMIR



IMPORTAR **02**

Uno de los requerimientos más necesarios se presenta cuando cargamos imágenes externas y queremos darles algún tipo de atractivo a la hora de visualizarlas. Aquí usaremos la clase TweenMax, junto a sus eventos, para resolver esta situación. Primero, debemos descargar las clases que queremos; en este caso, el paquete completo TweenMax, desde: www.greensock.com/as/greensock-as3.zip.

PASO | 01 Luego de descargar y descomprimir los archivos, abrimos Flash y relacionamos la librería con nuestro proyecto. Hacemos clic en **[Archivo/Configuración de ActionScript]**. Abrimos la solapa **[Ruta de origen]** y, luego, pulsamos en el signo "+" para agregar una nueva ruta. Finalmente, elegimos la carpeta para abrir la ventana de diálogo.

PASO | 02 Importamos las clases necesarias para disponer de interpolación:

```
import com.greensock.*;
import com.greensock.easing.*;
```

PASO | 03 Si se produce un error o no se puede importar la clase, es porque la ruta en el

IMAGEN

```
// Clases importadas de TweenMax
import com.greensock.*;
import com.greensock.easing.*;
//Clases nativas de Flash requeridas para este ejemplo
import flash.display.Loader;
import flash.net.URLRequest;
import flash.events.Event;
//Loader para cargar un jpg externo
var loaderImg: Loader = new Loader ();
//Eventos para la carga externa
loaderImg.load (new URLRequest ("img_ejemplo.jpg"));
loaderImg.contentLoaderInfo.addEventListener(Event.COMPLETE, loadComplete);
//Funciones de estado de la carga
function loadComplete (e:Event):void
{
    trace ("complete ok")
}
function loadInit (e:Event) :void
{
    trace ("Init ok")
}
```

03 | AGREGAR EVENTOS

```
import com.greensock.easing.*;
//Clases nativas de Flash requeridas para este ejemplo
import flash.display.Loader;
import flash.net.URLRequest;
import flash.events.Event;
//Loader para cargar un jpg externo
var loaderImg: Loader = new Loader ();
//Eventos para la carga externa
loaderImg.load (new URLRequest ("img_ejemplo.jpg"));
loaderImg.contentLoaderInfo.addEventListener(Event.COMPLETE, loadComplete);
//Funciones de estado de la carga
function loadComplete (e:Event):void
{
    this.addChild(loaderImg);
    TweenMax.to (e.target.content, 1, {alpha:1, blurFilter:(blurX: 20, blurY: 20)});
    TweenMax.to (e.target.content, 2, {ease: Elastic.easeOut, scaleX:0.4, scaleY: 0.4, delay:1});
    TweenMax.to (e.target.content, 2, {ease: Expo.easeOut blurFilter: (blurX:0, blurY:0), delay:2});
}
function loadInit (e:Event) :void
{
    e.target.content.alpha = 0;
}
```

EFFECTOS | 05

```
import flash.events.Event;
//Loader para cargar un jpg externo
var loaderImg: Loader = new Loader ();
//Eventos para la carga externa
loaderImg.load (new URLRequest ("img_ejemplo.jpg"));
loaderImg.contentLoaderInfo.addEventListener(Event.COMPLETE, loadComplete);
//Funciones de estado de la carga
function loadComplete (e:Event):void
{
    this.addChild(loaderImg);
    //efecto in para el alpha, movimiento lineal
    TweenMax.to (e.target.content, 1, {alpha:1});
    //escalar el objeto con efecto elastico al final. Se ejecuta 1 segundo mas tarde
    TweenMax.to (e.target.content, 2, {ease: Elastic.easeOut, scaleX:0.4, scaleY: 0.4, delay:1});
}
function loadInit (e:Event) :void
{
    e.target.content.alpha = 0;
}
```

04 | COMPORTAMIENTO

```
import com.greensock.*;
import com.greensock.easing.*;
import flash.display.Loader;
import flash.net.URLRequest;
import flash.events.Event;
var loaderImg: Loader = new Loader ();
loaderImg.load (new URLRequest ("img_ejemplo.jpg"));
loaderImg.contentLoaderInfo.addEventListener(Event.COMPLETE, loadComplete);
function loadComplete (e:Event):void
{
    this.addChild(loaderImg);
    TweenMax.to (e.target.content, 1, {alpha:1, blurFilter:(blurX: 20, blurY: 20)});
    TweenMax.to (e.target.content, 2, {ease: Elastic.easeOut, scaleX:0.4, scaleY: 0.4, delay:1});
    TweenMax.to (e.target.content, 2, {ease: Expo.easeOut blurFilter: (blurX:0, blurY:0), delay:2, onComplete:(funTween)});
}
function loadInit (e:Event) :void
{
    e.target.content.alpha = 0;
}
function funTween () :void
{
    trace ("fin de la interpolación completa")
}
```

EJECUTAR LA FUNCIÓN | 06

paso anterior está incorrecta; deberemos verificarla. En esta instancia creamos el comportamiento para cargar una imagen y agregarle un listener para los eventos de carga (COMPLETE e INIT).

Esta es una de las características más útiles, ya que con dichos métodos podemos iniciar otras interpolaciones o acciones sucesivas.

PASO | 04 Generamos un comportamiento Tween para que, cuando finalice la carga, se muestre un fade de la imagen, se escale y se centre al medio del escenario. Este último comportamiento se realiza al finalizar el alpha, mediante el uso del método **delay()**.

PASO | 05 TweenMax cuenta con varios efectos, como Blur, para sacar de foco un objeto. Aplicaremos el efecto después de la escala.

PASO | 06 Finalmente, podemos ejecutar una nueva función, si es que la interpolación ha finalizado.

RESUMEN

En este apartado práctico hemos aprendido a aplicar un efecto a una imagen para hacerla más atractiva en términos estéticos. El proceso se realizó mediante el uso de la clase TweenMax junto a eventos para resolver esta situación.

SISTEMA

DE PARTÍCULAS

Conoceremos el sistema que nos permitirá realizar animaciones más reales que si las hiciéramos con movie clips o frame por frame. Veamos de qué se trata y cómo aplicar este efecto.



SOBRE EL SISTEMA DE PARTÍCULAS

Sabemos que un sistema de partículas nos permite realizar efectos bien reales en nuestro proyectos. En líneas generales, podemos decir que se trata de una colección muy variada de partículas diminutas que, funcionando en conjunto, representan un objeto difuso. A lo largo de un intervalo de tiempo, se generan partículas nuevas dentro del sistema y, luego, se movilizan, cambian y mueren dentro del propio sistema. Este es el modo de funcionamiento del sistema de partículas, que nos ofrece una sensación de realidad en cada movimiento.

Durante años, el ser humano no solo intentó clonarse a sí mismo por medio de la tecnología, sino que también quiso imitar el mundo que lo rodea. Desde las tecnologías que asisten a las personas, hasta los sistemas que podrían reproducir el Big Bang, el hombre va detrás de esos descubrimientos. Si bien con Flash no podríamos realizar tales epopeyas, sí seremos capaces de realizar simulaciones que nos den la sensación de que algo real está sucediendo. Esta es la función, en principio, de los sistemas de partículas. Cuando hablamos de sistemas de partículas no hacemos referencia a una parte oculta de Flash, en donde tenemos que utilizar movie clips subatómicos. Tan solo se trata de un poco de programación, una pizca de matemática y algo de física. En otras palabras, con un sistema de partículas podríamos simular efectos que serían complicados de implementar con otras herramientas de Flash, como animando movie clips o frame por frame.

Con estos métodos, solo obtendríamos animaciones toscas y, en algún momento, repetitivas. Es por eso que se recurre a la programación de partículas, para que las animaciones resulten más reales.

¿DÓNDE APLICAR EL SISTEMA DE PARTÍCULAS?

La respuesta a esta pregunta es simple: en juegos. Sin embargo, con el correr del tiempo, la creatividad y la idea de dar valor agregado a las páginas web para despegar a Flash de los formatos convencionales comenzó a aplicarse en

CON UN SISTEMA DE PARTÍCULAS PODEMOS SIMULAR EFECTOS QUE SERÍAN COMPLICADOS DE IMPLEMENTAR CON OTRAS HERRAMIENTAS.

diversas partes. Esto es así siempre y cuando la idea lo requiera y no nos haga cruzar esa delgada línea que separa el buen gusto del mal gusto. Un ejemplo de buen gusto puede verse en: <http://old.pluginmedia.net>. Esta empresa se dedica al desarrollo de juegos en Flash. Es por eso que, por medio de su página web, debe transmitir lo mejor que sabe hacer. De esa idea surgió la página bajo el concepto de juego: en el sitio mencionado, una nave extraterrestre dispara un rayo láser verde que elimina humanos, mientras que un menú chorrea una viscosidad verde. Todo lo mencionado hasta el momento es interesante, entretenido y bien aplicado. Si bien estamos distrayendo al navegante de la idea principal de todo comercio (la venta), lo estamos atrapando indirectamente con nuestro trabajo, que es lo más importante.





En <http://old.pluginmedia.net>, la selección de los botones se hace disparando el rayo láser.

APLICACIÓN EN JUEGOS

La aplicación de partículas en juegos es, quizás, el ambiente más conocido para este tipo de efectos. Justamente, es allí donde podemos darle cierto realismo a lo que estamos haciendo por medio de efectos tales como nieve, lluvia, fuego, polvo, bruma y rayos, entre otros. Siempre tendremos que hacer uso de la razón para aplicar la medida justa en cada proyecto que llevemos adelante. Veamos algunos ejemplos de partículas aplicadas a juegos. En <http://ipachristmascard.co.uk> encontraremos un juego navideño en el cual personificamos a un reno que debe ir saltando de techo en techo para llegar a la meta. Se trata de un entretenimiento simple, y por ende, no es necesario agregar muchos efectos más que la nieve, es decir, nieve que cae desde posiciones aleatorias, con tamaños aleatorios (obviamente, especificando un máximo), y con una ondulación que varía en apertura y velocidad. Todo esto es algo simple y completa el juego.

LIBRERÍAS DE CÓDIGO ABIERTO

En todos estos casos, no siempre es necesario sentarnos durante largas horas a programar funciones o clases para generar efectos de lluvia, fuego o rayos. Existen librerías escritas por otras personas o por grupos, que son de código abierto para todo público, de modo que podemos utilizarlas en nuestros proyectos libremente. Es bueno conocerlas, descargarlas y, mucho mejor aún, probarlas. Es importante que los



Aquí, el menú izquierdo derrama un líquido. Todos se realizó mediante el sistema de partículas.

principiantes hagan proyectos que, si bien hoy tal vez no tengan sentido, seguramente el día de mañana los harán sentir agradecidos de haber usado esas horas en aprender a manipular este tipo de elementos. Veamos algunas de ellas:

► **Stardust particle engine:** <http://code.google.com/p/stardust-particle-engine>: en el paquete de descarga encontraremos bastante información, un explicativo de 46 páginas, muchísimos ejemplos y formas de aplicar esta librería. En el siguiente link podremos ver cómo se aplica **Stardust Particles**: <http://active.tutsplus.com/tutorials/effects/shoot-out-stars-with-the-stardust-particle-engine>

► **Emitte:** <http://code.google.com/p/emitter>: es de los creadores de STARDUST; un poco más simple, sin soporte para 3D, pero con buena documentación y ejemplos adecuados. Dependiendo del proyecto, quizá sea conveniente recurrir a esta versión para no agregar peso extra o complejidad a lo simple que tengamos que hacer.

► **Flint:** <http://flintparticles.org>: con buena documentación, las tareas que permite desarrollar son similares a las de otras librerías. En este caso, tenemos soporte vía blog, foro y tutoriales para aplicaciones con Papervision 3D, etc. Una librería interesante para para investigar un poco y aprovechar al máximo. ◀

SISTEMA DE PARTÍCULAS

Es un conjunto de elementos que representan a un determinado objeto más complejo, como puede ser nieve, fuego, agua, cabello o humo, entre otros.

Aplicaciones



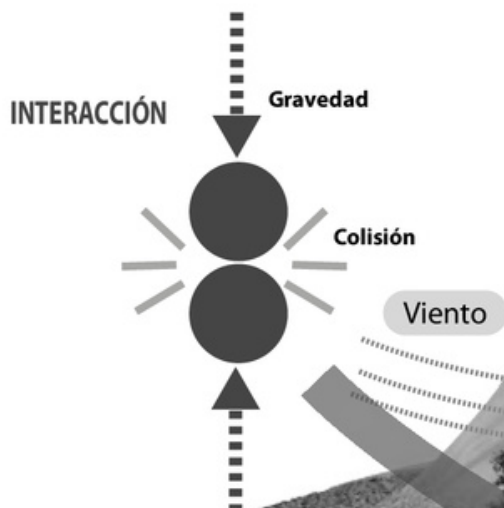
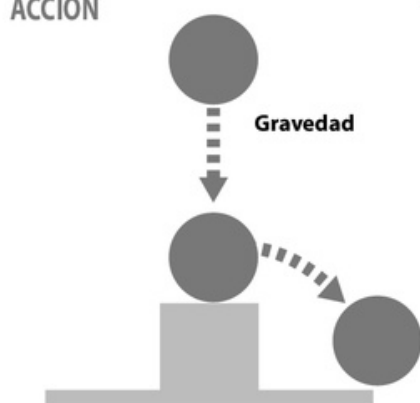
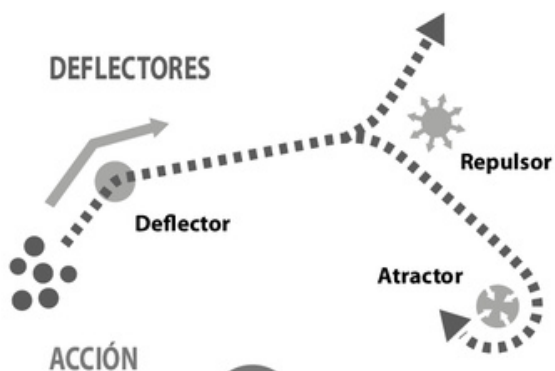
▶ Videojuegos

Flow, desarrollado por Jenova Chen para su tesis sobre videojuegos, es un claro ejemplo de utilización de sistemas de partículas y simulación. Es totalmente adictivo y simple.
<http://thatgamecompany.com/games/flow>



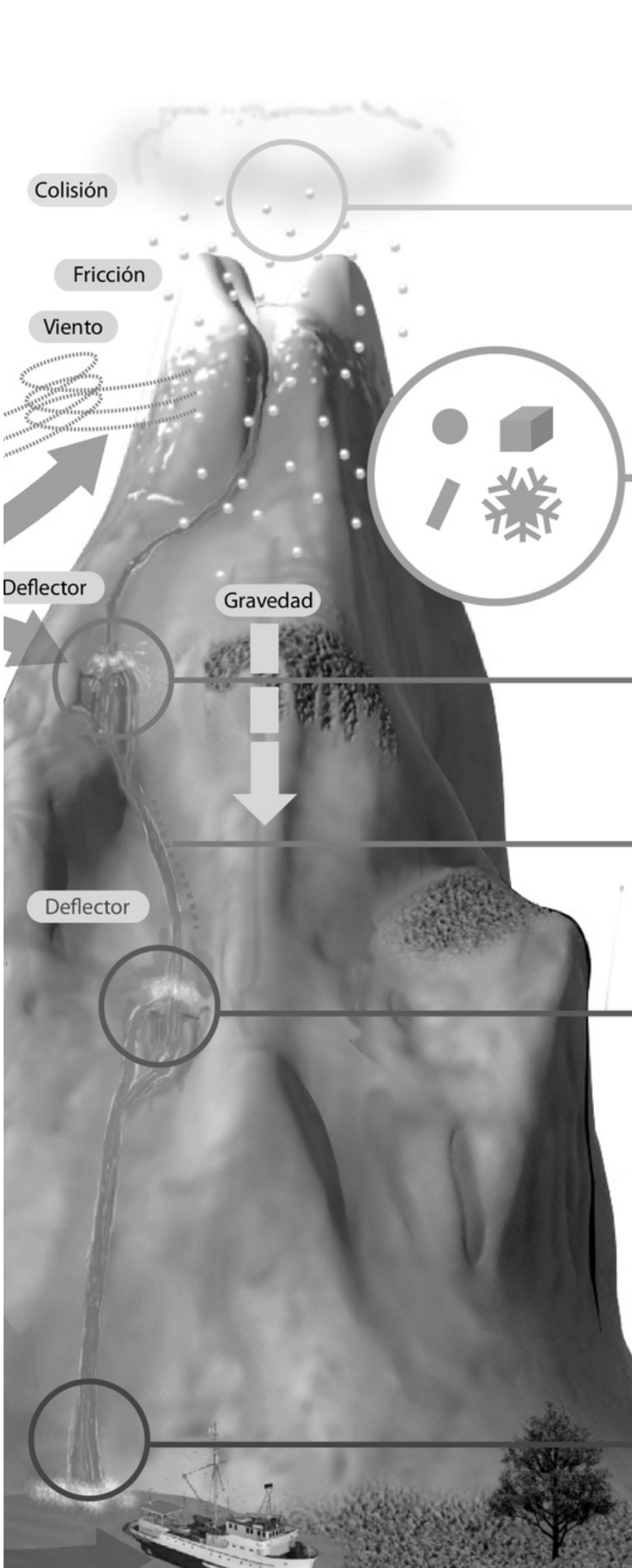
▶ Interfaces y diseño web

Gracias al desarrollo de lenguajes de programación potentes como ActionScript 3, se ha logrado integrar efectos visuales de muchísima calidad, en donde los usuarios pueden interactuar con sistemas de partículas, y efectos físicos y dinámicos que antes eran casi imposibles de simular.
www.liquidjourney.com/flash/lq3.html



Viento

Hojas que caen



01

Creación

Por medio de la combinación y aplicación de fórmulas matemáticas, físicas y programación. (Física + Matemática + Programación)

02

Emisor

Emisor del sistema de partículas, desde donde estas son emitidas.

03

Partículas

Pueden ser elementos geométricos, desde un simple punto o línea, hasta cuerpos más complejos en 3D.

04

Acción

Los movimientos de las partículas están determinados por las fuerzas interiores y exteriores que actúan sobre ellas.

05

Zona / recorrido

Las partículas tienen zonas definidas para actuar.

06

Deflectores

Una partícula puede ser afectada en su trayectoria y comportamiento al pasar por elementos que generen ciertas fuerzas sobre ella, tales como deflexión, repulsión o atracción.

07

Duración

Las partículas tienen una duración. Se puede indicar el tiempo de su "existencia".

08

Interacción

Cada partícula puede ejercer una fuerza determinada que también puede afectar a otras partículas en su entorno.

CÓMO ANIMAR EL LAYOUT DE NUESTRO

PASO A PASO

En este apartado conoceremos los métodos más simples para realizar animaciones sobre el layout de nuestro sitio.

Acciones - fotograma

Fragmentos de código

```
1 import flash.events.Event;
2
3 // Importo las clases de Tweening
4 import com.greensock.*;
5 import com.greensock.easing.*;
6
7 stage.align = StageAlign.TOP_LEFT;
8 stage.scaleMode = StageScaleMode.NO_SCALE;
9
10 function handlerResize(e:Event):void{
11     background.x = 0;
12     background.y = 0;
13     background.width = stage.stageWidth;
14     background.height = stage.stageHeight;
15
16     header.back.width = stage.stageWidth;
17     header.x = 0;
18     header.y = 0;
19     header.menuHeader.x = stage.stageWidth - header.menuHeader.width;
20 }
```

```
16 header.back.width = stage.stageWidth;
17 header.x = 0;
18 header.y = 0;
19 header.menuHeader.x = stage.stageWidth - header.menuHeader.width;
20
21 footer.back.width = stage.stageWidth;
22 footer.x = 0;
23 footer.y = stage.stageHeight - footer.back.height;
24
25 footer.socials.x = stage.stageWidth / 2 - footer.socials.width /
26 footer.copyright.x = stage.stageWidth - 10;
27 footer.fulls.x = 10;
28 }
29
30 stage.addEventListener(Event.RESIZE, handlerResize);
31 stage.dispatchEvent(new Event(Event.RESIZE));
32
33 TweenMax.to(background, 3, {alpha:1, delay:1, ease:Strong.easeOut});
```

01 | IMPORTAR

Es importante aclarar que podemos realizar las animaciones por medio de la línea de tiempo o utilizando librerías de animación. En este caso, vamos a recurrir a ambas opciones, básicamente, para entender que ambos métodos pueden convivir, y es posible aplicar uno u otro dependiendo de la situación. La librería que vamos a utilizar es TweenMax.

PASO | 01 Abrimos el archivo **FLA** principal con nuestra diagramación líquida. Seleccionamos el fotograma clave en donde tenemos el código ActionScript y presionamos <F9> para abrir el panel de acciones. Allí escribimos las líneas que nos permitirán importar la clase TweenMax:

```
import com.greensock.*;
import com.greensock.easing.*;
```

ANIMAR EL MOVIE CLIP

02

PASO | 02

Vamos a animar el movie clip background con la librería TweenMax. En el panel **[Acciones]**, al final de todo el código actual, redactamos:

```
TweenMax.to(background, 1, {alpha:3,
delay:1, ease:Strong.easeOut});
```

Para que funcione, tenemos que poner nuestro background en alpha 0 (transparente).

PASO | 03

Hacemos clic sobre el movie clip background y vamos a **[Ventana/Propiedades]**, donde elegimos **[Efecto de Color/Estilo]**. Seleccionamos la opción **[Alpha]** y colocamos su valor en 0 (cero). Para verificar nuestro movie clip, presionamos

SITIO

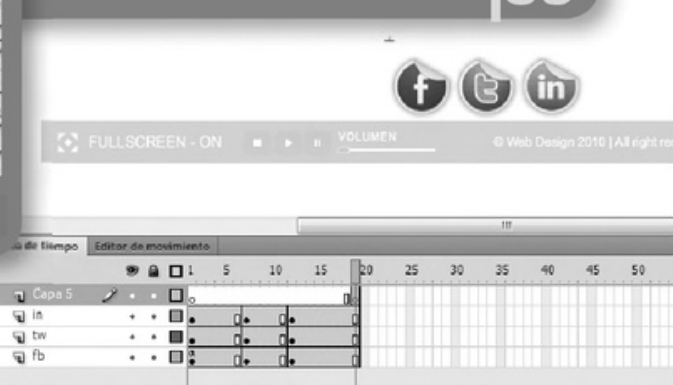
04 | ICONOS DE REDES SOCIALES



03 | TRANSPARENCIA



INICIO Y DETENCIÓN | 05



CAPAS Y FOTOGRAMAS | 06

<Ctrl + Enter>. Veremos que el background aparecerá por medio de una transparencia, con un delay de 1 segundo.

PASO | 04 Es necesario que dentro del movie clip footer creamos otro y le pongamos como nombre de instancia **socials**. Dentro de él agregamos los tres iconos de las redes sociales. Con cada uno de ellos hacemos un movie clip con los siguientes nombres de instancia: Facebook **[fb]**, Twitter **[tw]** y LinkedIn **[lin]**. Quedará una estructura como la siguiente:

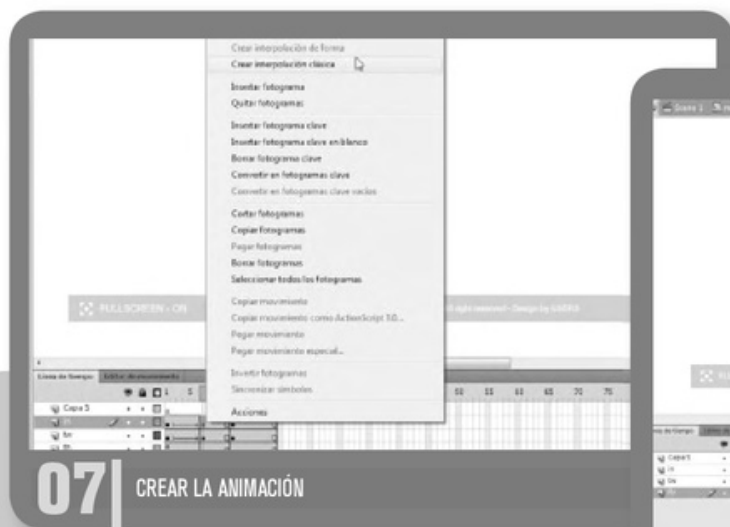
[footer] está en el escenario principal
[socials] está dentro de footer
[fb] está dentro de socials
[tw] está dentro de socials
[lin] está dentro de socials
moviesAnidados.ai (moviesAnidados.jpg)

PASO | 05 Vamos a poner cada icono en una capa distinta. Por sobre estas tres capas, crearemos una más y extenderemos las cuatro hasta el fotograma 19. Para hacerlo, presionamos en el fotograma 19 de cada una de las capas y pulsamos <F5>. Creamos un fotograma clave en el fotograma 19 de nuestra capa vacía y le agregamos un **[stop();]**. La animación se iniciará en el fotograma 1 y frenará en el 19.

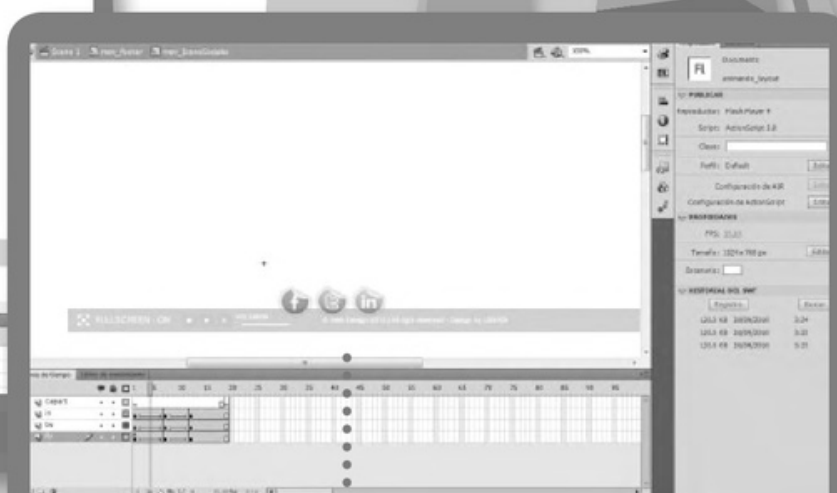
PASO | 06 En las tres capas en donde se encuentran los iconos de las redes sociales vamos a crear un fotograma clave en 8 y 12.

CÓMO ANIMAR EL LAYOUT DE NUESTRO SITIO

PASO A PASO



07 CREAR LA ANIMACIÓN



VARIAR LA FORMA **08**



09 AGREGAR REBOTE



PASO | 07 Una vez creados los fotogramas clave, presionamos el botón derecho del mouse entre el fotograma 1 y el 8, y seleccionamos la opción **[Created Motion Tween]**. También podemos elegir un fotograma entre el 1 y el 8, y dirigimos a **[Insert/Classic Tween]**. Estos pasos deben repetirse en las tres capas y, también, en el rango de fotogramas que va desde el 8 hasta el 12.

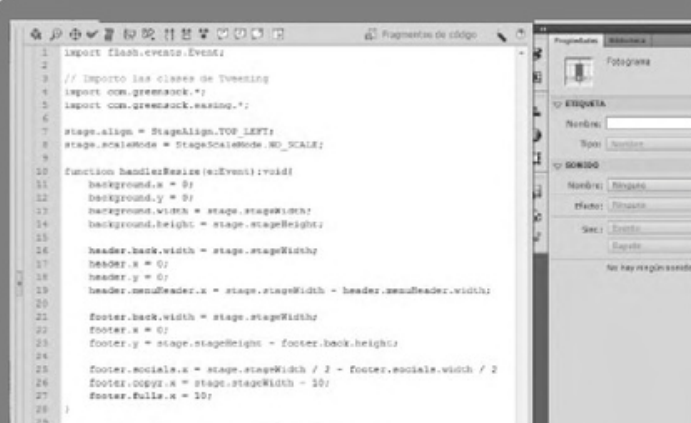
PASO | 08 Solo nos queda variar la morfología de los movie clips en cada fotograma clave. Hacemos clic en el primer fotograma de la capa, que corresponde al icono de Facebook. Tomamos la herramienta **[Transformación libre]** presionando la letra **<q>**. Mientras pulsamos **<Shift>**, disminuimos el tamaño del icono. A nuestro movie clip le restamos al menos 20 píxeles a su posición en el **eje Y**, para que quede más abajo y, luego, le damos un alpha de 0 para que desaparezca. Esto mismo lo repetimos con los otros dos iconos.

PASO | 09 Para que el movimiento no sea tan tosco, vamos a agregar un rebote que le dará un poco más de estilo. Seleccionamos el fotograma 8 de la capa de Facebook y le restamos 10 píxeles en su **eje Y**. Esto lo repetimos con el resto de los clips de película.

PASO | 10 Ahora vamos a darle un orden de aparición a cada uno de los iconos. Esto



10 | ORDEN DE APARICIÓN



APLICAR LO APRENDIDO

12



11 | REFERENCIAS

puede hacerse seleccionando los fotogramas animados (del 1 al 12) de cada capa. Supongamos que el orden es [fb], [tw] y [lin]. Por ser el primero, [fb] se quedará en su posición actual, así que seleccionamos los fotogramas del 1 al 12 de la capa del icono [tw]. Arrastramos los cuatro fotogramas hacia delante; de esta forma, la animación del icono de Twitter se iniciará 4 fotogramas después que la de Facebook. Hacemos lo mismo con la capa de LinkedIn, pero en vez de 4 fotogramas, en este caso serán 8. De este modo, se va a iniciar 4 fotogramas después que Twitter.

PASO | 11 Por una razón de referencias, será necesario copiar el primer fotograma animado de Twitter y de LinkedIn en el primer fotograma de su respectiva capa. De este modo, no cambia la estética de la animación y no perdemos la referencia a ese movie clip.

PASO | 12 Ya tenemos una base para animar nuestro layout. Podemos probar con el resto de los objetos que nos quedan en pantalla, aprovechando lo que ya aprendimos.

PASO | 13 Una vez animados todos los objetos de nuestro interés, estamos listos para verificar la versión final. Presionamos <F12> o vamos a [File/Publish] para ver el resultado en el navegador.

RESUMEN

En este apartado hemos aprendido a animar el layout por medio de la línea de tiempo y con las librerías de animación. Para este último caso, utilizamos la clase TweenMax.

VERIFICACIÓN

13

ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué son los efectos físicos?

02 ¿Para qué sirven las librerías de animación?

03 Mencione algunas librerías de animación que podemos utilizar.

04 ¿Qué es **TweenMax**?

05 ¿Qué es **TweenLite**?

06 Describa la forma de aplicar efectos a una imagen.

07 ¿Qué es un sistema de partículas?

08 ¿Para qué sirve un sistema de partículas?

09 ¿Qué ventajas proporciona un sistema de partículas para un juego?

10 Mencione algunas librerías de código abierto para generar diversos efectos.

PRÁCTICAS

01 Experimente con las aplicación de efectos físicos.

02 Utilice algún suavizado en el texto.

03 Agregue efectos a una imagen.

04 Aplique un sistema de partículas.

05 Descargue y pruebe una librería de código abierto.

Imágenes y galerías

- ▶ INCORPORACIÓN DE IMÁGENES
- ▶ OPTIMIZACIÓN DE IMÁGENES
- ▶ FILTROS DINÁMICOS
- ▶ BITMAP Y BITMAPDATA
- ▶ GALERÍA DE FOTOS
- ▶ COMPONENTES
- ▶ CREAR UN SLIDESHOW DINÁMICO



INCORPORACIÓN

DE IMÁGENES

Trabajar con imágenes puede resultar una labor simple o compleja. Esto dependerá del sistema que utilicemos y de la planificación, pero además, de la selección de las herramientas adecuadas de Flash.

Un elemento fundamental de cualquier diseño web son las imágenes. Veamos de qué forma podemos incorporarlas a nuestro trabajo para sacarles el mayor provecho. A la hora de planificar nuestra pieza, ya sea un sitio completo o un banner, las imágenes ocupan un lugar destacado en cualquier proyecto para la Web. Desde un simple encabezado de secciones, hasta la ilustración de productos y notas, el objetivo de asegurar una integración fluida con el resto del contenido debe ser primordial para obtener un acabado profesional.

Para incorporar imágenes en Flash tenemos dos opciones: incluirlas en la biblioteca de la película o cargarlas externamente. En general, ambos enfoques se usan a la vez, dependiendo de qué tipo de material se quiera mostrar. Pero es importante tener bien en claro las diferencias, y cuáles son las ventajas y desventajas de cada una de estas alternativas.

IMÁGENES VECTORIALES Y DE MAPA DE BITS

Cuando trabajamos con imágenes, es importante tener en mente la distinción entre **vectoriales** y de **mapa de bits**. Las primeras pueden describirse mediante fórmulas matemáticas y ocupan menos espacio en la memoria, ya que el mismo sistema de **renderización** se encarga de desplegarlas. Típicamente, este tipo de imágenes son las que generamos desde programas de diseño

como **Adobe Illustrator** y que permanecen editables dentro de Flash. Tienen extensiones AI o EPS.

Las imágenes de mapa de bits, en cambio, se describen píxel a píxel. Son siempre más grandes en cuanto a tamaño de archivo, como las de extensión BMP o las que incorporan compresión, como **JPG**.

IMÁGENES EN LA LIBRERÍA

Esta es la manera más simple de incluir imágenes en una película de Flash. Para importarla, debemos seguir la ruta [**Archivo/Importar/Importar a la librería**]. Una vez allí, la imagen puede ser utilizada directamente, es decir, arrastrándola desde la biblioteca hasta el escenario.

VENTAJAS Y DESVENTAJAS

¿Cuáles son las ventajas y desventajas de este método? Como principal ventaja, podemos mencionar la facilidad de implementación. Las imágenes se importan a la biblioteca y, desde allí, son arrastradas al escenario; nada más intuitivo. La desventaja principal es el peso que cada imagen agrega al archivo final. De todas formas, hay que aclarar que si se usa un **bitmap** no comprimido (por ejemplo, se importa un BMP), este luego es comprimido según se haya estipulado en las opciones de publicación: [**Menú Archivo/Opciones de Publicación/**

Flash], en donde se puede elegir la cantidad de compresión **JPG** que se aplicará. En cualquier caso, el aumento de peso que implica incorporar imágenes a la librería es importante y no es aconsejable excederse, excepto para la inclusión de iconos u otros elementos significativos que componen la estructura. De lo contrario, en general siempre optaremos por cargar imágenes externas.

IMÁGENES EXTERNAS

Las imágenes externas se cargan mediante la **clase Loader** y son agregadas al **DisplayList** para formar parte del diseño en forma dinámica. En su mínima expresión (es decir, sin precarga) sería del siguiente modo:

```
var ldr:Loader = new
Loader();ldr.load(new
URLRequest("imagen.jpg"));
addChild(ldr);
```

En este caso, incorporamos a nuestro escenario el archivo **imagen.jpg**. Si, por ejemplo, pesa 100 KB, ése será el peso que habremos quitado de nuestro **SWF** final. Pero la carga externa de imágenes es un poco más compleja e implica un mayor planeamiento. Por ejemplo: ¿qué



IMÁGENES EN LA ANIMACIÓN

Cuando cargamos imágenes, también debemos tener en cuenta si participan en animaciones y de qué tipo son estas últimas; por ejemplo, bitmaps muy grandes que se mueven, rotan y ocupan gran cantidad de memoria, o si aplicamos animaciones de alpha. En líneas generales, los traslados lineales (sobre el eje X o Y) resultan lo más económico para el procesador, siendo el alpha lo que más recursos consume. Siempre es aconsejable disponer de una máquina con poco poder de procesamiento para testear las animaciones.

pasaría si nuestra película debiera cargar 20 imágenes? ¿Las cargaríamos todas juntas para tenerlas disponibles o las descargaríamos a medida que fueran necesarias?

No hay una respuesta contundente para decidir la estrategia, ya que esta dependerá, sobre todo, del flujo de navegación que proponga el sitio. Sin embargo, hay algo indiscutible: cuanto más rápido, mejor.

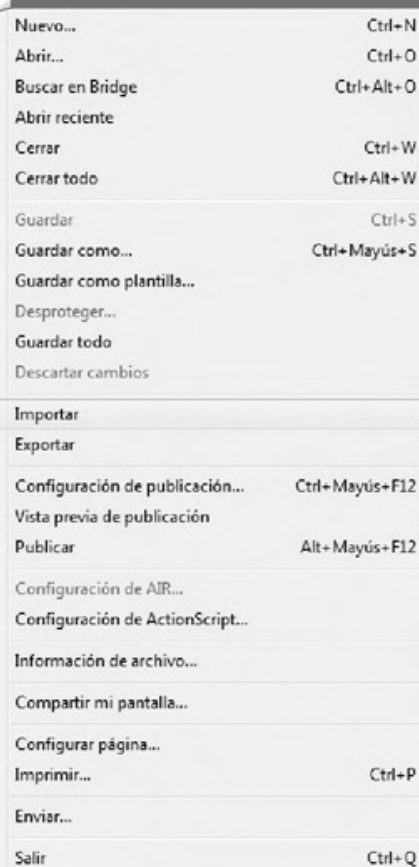
CARGA CON PRELOAD

Siempre que carguemos imágenes externamente, necesitaremos agregar un preload, a menos que el tamaño de estas sea muy pequeño. Esto es así porque, hasta que la imagen no termine de cargar, el usuario no sabrá lo que está sucediendo y, simplemente, verá un espacio vacío. ◀

```

1 var texto:TextField = new TextField();
2 var marco:Shape = new Shape();
3 var barra:Shape = new Shape();
4 function precarga(quien:String):void{
5     var ldr:Loader = new Loader();
6     //dibujo la barra
7     marco.graphics.lineStyle(1, 0x000000);
8     marco.graphics.drawRoundRect(0, 20, 70, 5, 0);
9     barra.graphics.beginFill(0x000000);
10    barra.graphics.drawRect(0, 20, 70, 5);
11    barra.graphics.endFill();
12    //agrego elementos al escenario
13    addChild(texto);
14    addChild(marco);
15    addChild(barra);
16    //posiciono elementos
17    texto.x = marco.x = barra.x = stage.stageWidth / 2 - marco.width / 2;
18    texto.y = stage.stageHeight / 2 - texto.height / 2;
19    marco.y = barra.y = texto.y + 5;
20    //agrego listeners al loader y finalmente cargo
21    ldr.loaderInfo.addEventListener(ProgressEvent.
22    PROGRESS, onLoadProgress);
23    ldr.load(new URLRequest(quien));
24    addChild(ldr);
25 }
26 function onLoadProgress (event:ProgressEvent):void{
27     var cargado:int = event.bytesLoaded;
28     var total:int = event.bytesTotal;
29     var porcentaje:int = cargado/total*100;
30     texto.text = "Cargado: "+String(porcentaje)+"%";
31     barra.width = porcentaje*marco.width/100;
32 }
33 function onLoadComplete (event:Event):void{
34     removeChild(texto);
35     removeChild(marco);
36     removeChild(barra);
37 }
38 precarga("Fondo.jpg");
  
```

Aquí vemos un ejemplo de código reutilizable de preload. ¡Con Flash CSS, todo es posible!



Podemos observar la ruta que debemos seguir para importar a la biblioteca que deseamos para nuestro proyecto.



Opciones disponibles en la ventana de [Configuración de publicación]. Debemos configurarlas según nuestra necesidad.



OPTIMIZACIÓN DE IMÁGENES

Cuando trabajamos con imágenes en una película, debemos tener presente el tamaño máximo en el cual estas serán mostradas, ya que si los originales tienen menores dimensiones, obtendremos malos resultados al intentar visualizarlas.

Una imagen de 100 x 100 píxeles, llevada a 300 píxeles, indefectiblemente se va a deformar (pixelar), y habrá que utilizar otra imagen con el primer tamaño para mantener la calidad adecuada. Cuando hacemos lo contrario, no hay problemas: si importamos una foto para utilizar con un tamaño menor, la calidad se mantendrá, aunque estaremos agregando peso al archivo innecesariamente. Con respecto a este tema, es importante considerar que, si tenemos una imagen grande y la achicamos con la herramienta **[Escalado]**, solo la veremos más pequeña, pero su peso será exactamente el mismo del original. Aunque Flash automáticamente comprime las imágenes para ser publicadas, si las optimizamos en un programa externo

EL SUAVIZADO INTERPOLA PÍXELES PARA MEJORAR LA IMAGEN TODO LO POSIBLE.

para tener una calidad óptima, lograremos mayor control sobre el producto. Esto es así porque Flash no distingue entre las imágenes, y aplica una compresión uniforme a todo el material que no esté ya comprimido. En líneas generales, usaremos **JPG** para fotografías (de mapa de bits), y preferiremos **PNG** o **GIF** cuando queramos convertir imágenes vectoriales a bitmap. De este modo, podremos beneficiarnos de la paleta indexada de colores que manejan estos formatos. Es importante tener en cuenta que al usar un programa

externo para crear la imagen, como Adobe Photoshop, si le hacemos cambios en cualquier momento, luego podremos ir a la imagen en la biblioteca, hacer clic derecho sobre ella, seleccionar **[Propiedades]** y actualizarla en Flash desde el botón **[Actualizar]**. Si el archivo original fue borrado, Flash de todos modos guardará una copia, pero ya no podremos editarlo.

SUAVIZADO DE IMÁGENES

Cuando queremos moderar el efecto de pixelado que se genera al agrandar imágenes más allá de su tamaño original, o cuando aplicamos rotaciones en una animación, es aconsejable utilizar la opción de **suavizado** para disimular la pérdida de calidad. Esta herramienta interpola píxeles con el objetivo de mejorar la imagen.

Para aplicar el suavizado tenemos que importar una imagen a la biblioteca, hacer clic con el botón derecho del mouse en ella y elegir **[Propiedades]**. Luego, marcamos la casilla **[Permitir suavizado]**, para que el efecto se aplique sobre la imagen en el escenario.

FILTROS Y EFECTOS

Flash ofrece una serie de filtros de imágenes que son fáciles de aplicar, tanto desde la interfaz del programa como mediante el uso de código. Veamos cuáles son las opciones y cuál utilizar de acuerdo con nuestro proyecto. Si colocamos una imagen en el escenario y la convertimos en movie clip, podemos seleccionarla y aplicarle una serie de



Podemos observar cómo quedan algunos de los filtros aplicados a una imagen.

```

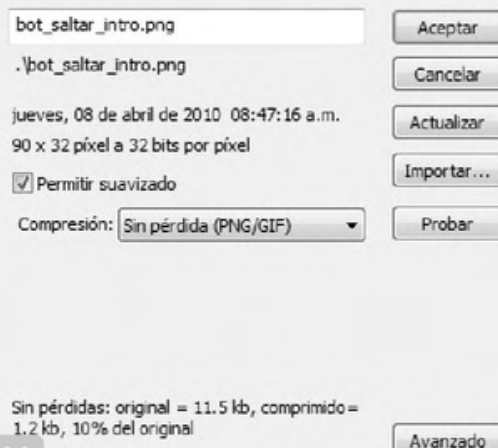
1
2 function onLoadComplete (event:Event):void{
3     var bm:Bitmap = Bitmap(e.currentTarget.loader.content);
4     bm.smoothing = true;
5     removeChild(texto);
6     removeChild(marco);
7     removeChild(barra);
8 }

```

En muchas ocasiones deberemos utilizar el suavizado por código, para lo cual recurrimos a la clase Bitmap.

filtros desde el panel de propiedades, de manera simple e intuitiva. Solo debemos desplegar la opción **[Filtros]** y pulsar el botón **[Añadir filtros]**, que es el primero en el ángulo inferior izquierdo. Los filtros disponibles son: sombra, desenfoque, iluminar, bisel, iluminado degradado, bisel degradado y ajustar color. A ellos se los denomina **básicos**, porque no necesitamos hacer transformaciones complejas o matrices para aplicarlos. Más adelante, veremos otros casos. Los filtros se aplican en orden; esto quiere decir que, si a un objeto le aplicamos una sombra y, luego, un desenfoque, este último afectará al elemento con la sombra, ya que esta fue aplicada en un paso anterior. En este caso, el orden de los factores sí altera el producto.

Saltar intro



Observamos las propiedades disponibles en el cuadro de diálogo de Mapa de bits.

BLUR DE MOVIMIENTO

Un efecto bastante utilizado es el filtro Blur, para simular movimiento en cualquier dirección, algo a lo que todos estamos muy acostumbrados gracias a los dibujos animados. En fotografía, un desenfoque de movimiento se genera cuando el objeto se mueve demasiado rápido para la velocidad de la película que lo captura. El resultado de esto es una imagen borrosa que connota movimiento. En la animación, podemos crear la ilusión de la velocidad mediante la adición de desenfoque de movimiento a una imagen. Simplemente, se trata de utilizar la interpolación y el filtro de desenfoque de movimiento.

APLICAR SOMBRAS DINÁMICAMENTE

Podemos aplicar sombras de forma dinámica sobre un texto alternativamente, es decir que desde un botón ponemos y quitamos la sombra de un texto dinámico, de un modo bastante simple y con sólo unas pocas líneas de código. Para abreviar el código, colocaremos manualmente en el escenario un campo de texto dinámico con nombre de instancia **muestra_txt** y un componente **button** al que le pondremos como nombre de instancia **cambiar_btn** y, como label, **Sombra**.

DESENFQUE DE IMÁGENES EN EL ROLL OVER

Otro efecto bastante utilizado es el de imágenes desenfocadas que se enfocan al pasar el mouse sobre ellas. Supongamos que tenemos una imagen en el escenario llamada "imagen_mc". Con el código que se muestra a continuación, armamos el efecto de enfoque/desenfoque, al que, además, agregaremos un filtro de flow al enfocar.

```

1 import flash.events.MouseEvent;
2 import flash.filters.BlurFilter;
3 import flash.filters.GlowFilter;
4
5 imagen_mc.addEventListener(MouseEvent.CLICK, enfoca);
6 imagen_mc.addEventListener(MouseEvent.CLICK, desenfoca);
7 var filter1:BlurFilter = new BlurFilter(8, 8, 1);
8 var filter2:GlowFilter = new GlowFilter()
9
10 imagen_mc.filters = [filter1]
11
12 function desenfoca(evt:MouseEvent):void{
13     imagen_mc.filters = [filter1]
14 }
15 function enfoca(evt:MouseEvent){
16     imagen_mc.filters = [filter2]
17 }
18
19

```

El código va en el primer y único fotograma. Usamos el filtro de sombra con los parámetros por defecto, para simplificar.

FILTROS DINÁMICOS

Además de los filtros detallados en las páginas anteriores, en Flash encontramos otros complementarios, denominados dinámicos. Veamos cómo aplicarlos de acuerdo con las exigencias de cada proyecto.

Aplicar los filtros en forma dinámica resulta bastante simple: solo tenemos que armar un **array** de filtros y asignarlo a la propiedad **Filters** del elemento que queremos afectar. Todos los filtros pueden ser inicializados con sus valores por defecto, pero debemos revisar la documentación si queremos manejar los valores específicamente. Los filtros se suman; es decir, podemos darle a un mismo elemento varios filtros, sumándolos al array asignado. Por ejemplo, en el siguiente caso se aplican tres filtros:

```
imagen_mc.filters =
[new DropShadow(), new
BlurFilter(),
new GlowFilter()]
```

Para quitarlos simplemente asignamos null

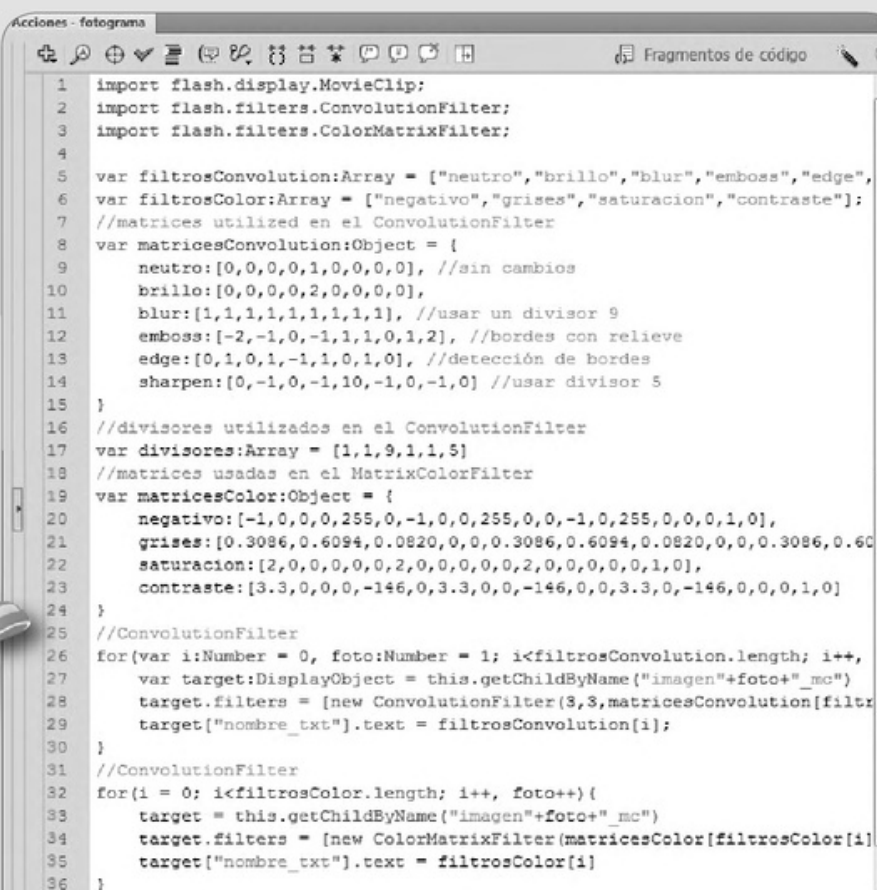
```
imagen_mc.filters = null;
```

CAMBIAR DE COLOR

Para cambiar de color usando AS3, debemos utilizar la clase **flash.geom.ColorTransform** y aplicarla a la propiedad **transform.colorTransform** del objeto. Con este fin, especificamos el color que queremos darle al movie clip con nombre de instancia **cuadrado_mc**. También podemos aplicar un offset por cada canal modificando las propiedades **redOffset**, **greenOffset**

y **blueOffset**; además del alpha a través de la propiedad **alphaOffset**, siempre en un **rango** que va de 0 a 255.

```
var miColor:ColorTransform
= cuadrado_mc.transform.
colorTransform
miColor.color = 0xFF0000
cuadrado_mc.transform.
colorTransform = miColor
```



```
Acciones - fotograma
Fragmentos de código
1 import flash.display.MovieClip;
2 import flash.filters.ConvolutionFilter;
3 import flash.filters.ColorMatrixFilter;
4
5 var filtrosConvolution:Array = ["neutro","brillo","blur","emboss","edge",
6 var filtrosColor:Array = ["negativo","grises","saturacion","contraste"];
7 //matrices utilizadas en el ConvolutionFilter
8 var matricesConvolution:Object = {
9     neutro:[0,0,0,0,1,0,0,0,0], //sin cambios
10    brillo:[0,0,0,0,2,0,0,0,0],
11    blur:[1,1,1,1,1,1,1,1,1], //usar un divisor 9
12    emboss:[-2,-1,0,-1,1,1,0,1,2], //bordes con relieve
13    edge:[0,1,0,1,-1,1,0,1,0], //detección de bordes
14    sharpen:[0,-1,0,-1,10,-1,0,-1,0] //usar divisor 5
15 }
16 //divisores utilizados en el ConvolutionFilter
17 var divisores:Array = [1,1,9,1,1,5]
18 //matrices usadas en el MatrixColorFilter
19 var matricesColor:Object = {
20     negativo:[-1,0,0,0,255,0,-1,0,0,255,0,0,-1,0,255,0,0,0,1,0],
21     grises:[0.3086,0.6094,0.0820,0,0,0.3086,0.6094,0.0820,0,0,0.3086,0.6094,0.0820,0,0,0.3086,0.6094,0.0820,0,0],
22     saturacion:[2,0,0,0,0,2,0,0,0,0,2,0,0,0,0,2,0,0,0,0],
23     contraste:[3.3,0,0,0,-146,0,3.3,0,0,-146,0,3.3,0,0,-146,0,3.3,0,-146,0]
24 }
25 //ConvolutionFilter
26 for(var i:Number = 0, foto:Number = 1; i<filtrosConvolution.length; i++){
27     var target:DisplayObject = this.getChildByName("imagen"+foto+"_mc")
28     target.filters = [new ConvolutionFilter(3,3,matricesConvolution[filtrosConvolution[i]],
29     target["nombre_txt"].text = filtrosConvolution[i];
30 }
31 //ConvolutionFilter
32 for(i = 0; i<filtrosColor.length; i++, foto++){
33     target = this.getChildByName("imagen"+foto+"_mc")
34     target.filters = [new ColorMatrixFilter(matricesColor[filtrosColor[i]],
35     target["nombre_txt"].text = filtrosColor[i]
36 }
```

Aplicamos los efectos **ConvolutionFilter** y **ColorMatrixFilter** sobre la misma foto. Pusimos las matrices en objetos cuyas propiedades identifican al filtro al que se refiere cada matriz.

Aplicamos matrices con **ConvolutionFilter**, más otras con **ColorMatrixFilter**, del que hablaremos en breve.

ENTINTAR UN MOVIE CLIP

Para entintar un clip (tint) podemos jugar con las propiedades **redMultiplier**, **greenMultiplier** y **blueMultiplier**, de **ColorTransform**. El rango debe estar entre 0 y 1. Si, por ejemplo, queremos llevar el color al verde, podríamos hacer lo siguiente:

```
var
miColor:ColorTransform =
cuadrado_mc.transform.
colorTransform
miColor.redMultiplier
= 0;miColor.
lueMultiplier =
0;cuadrado_mc.transform.
colorTransform = miColor
```

Este ejemplo va hacia el verde porque el valor predefinido de cada canal es 1, es decir, pleno. Por lo tanto, al dejar el rojo y el azul en 0, el verde tiene la hegemonía. Para ir hacia colores que no son específicamente ninguno de los tres componentes, debemos jugar con los porcentajes asignados de cada uno.

+ CONOCIMIENTOS TÉCNICOS

Recordemos que, para mejorar y potenciar nuestro aprendizaje, desde www.redusers.com/cursos/webdesign podemos descargar los archivos editables de los proyectos que realizamos en esta colección. A partir de ellos, obtendremos ideas para un desarrollo propio o, directamente, podremos modificar los archivos para reutilizarlos. En el sitio también encontraremos **guías visuales**, un **glosario de términos** y una completa guía de preguntas frecuentes sobre la suite CSS. Además, el libro **Desarrollo y animación en ActionScript 3.0** nos ayudará a comprender los fundamentos de la programación en Flash.



FILTROS AVANZADOS: CONVOLUTIONFILTER

Los filtros avanzados se aplican sobre la base del **ConvolutionFilter**, una clase que, a través del uso de una matriz multidimensional, permite la aplicación de efectos como relieve, desenfoque, brillo, detección de bordes y contraste, entre otros. La matriz se aplica para mapear cada píxel a una nueva configuración de bitmap combinando los píxeles adyacentes, pero la matemática compleja que se utiliza para esta operación escapa al alcance de esta sección. Simplemente, tengamos en cuenta que, según el tipo de matriz que apliquemos, iremos logrando diferentes efectos.

La clase **ConvolutionFilter** recibe tres parámetros: los dos primeros son las dimensiones de la matriz, y el tercero, un array con la matriz misma. Hay un cuarto parámetro opcional; se trata de un divisor que se aplica de forma homogénea a todos los elementos de la matriz. En general, para la mayoría de los efectos usaremos una matriz de 3 x 3. Veamos un ejemplo de instancia de un **ConvolutionFilter**, donde tenemos un **número uno (1)** rodeado de ceros (0), lo que suele tener un efecto neutral, es decir, no percibimos cambios en la imagen aplicada:

```
imagen_mc.filters =
[ConvolutionFilter(3, 3,
[0,0,0,0,1,0,0,0,0])]
```

PODEMOS DAR MUCHOS FILTROS A UN MISMO ELEMENTO, SUMÁNDOLOS AL ARRAY ASIGNADO.

En este caso, no habrá cambios visibles en el movie clip, pero en función de cómo vayamos manipulando la matriz, lograremos distintos efectos, como brillo, desenfoque, emboss (distorsión de bordes), edge (detección de bordes) y sharpen (marcado de bordes).

FILTROS AVANZADOS: COLORMATRIXFILTER

Este es otro de los filtros avanzados que podemos utilizar para aplicar efectos. En este caso, recibe un solo parámetro, que es la matriz por aplicar. Las matrices que se usan con **ColorMatrixFilter** suelen ser más grandes, por ejemplo, de 4 x 4. Los efectos que se pueden lograr son: negativo, escala de grises, saturación y contraste, entre otros. ◀

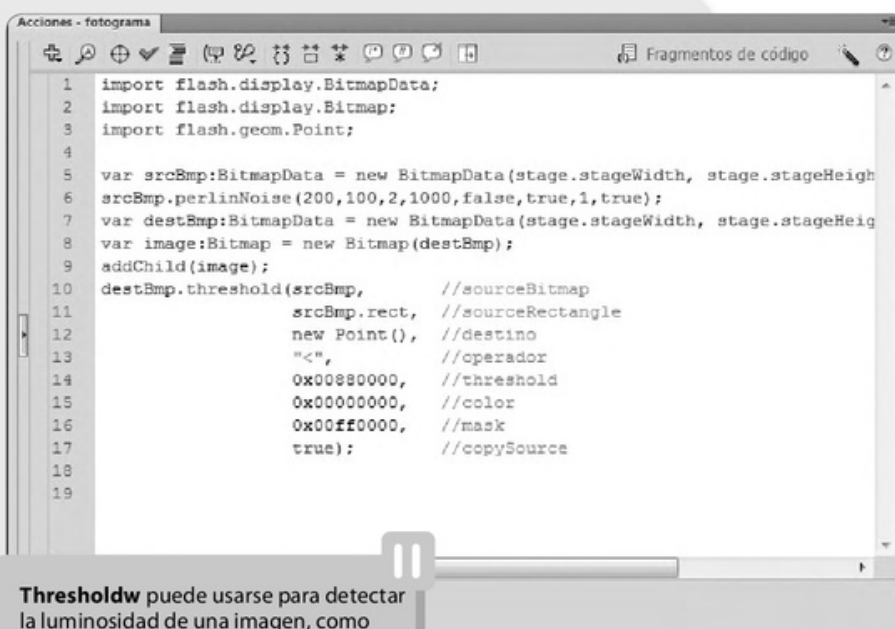
BITMAP Y BITMAPDATA

La clase `BitmapData` permite trabajar con los datos (píxeles) de un objeto `bitmap`. Podemos utilizar sus métodos para crear imágenes de mapa de bits opacas o transparentes de diversos tamaños, y manipularlas de varias formas.

Esta clase permite separar las operaciones de representación de mapa de bits de las rutinas internas de actualización, de la visualización de Flash Player. Con la manipulación directa de un objeto `BitmapData` es posible crear imágenes complejas sin recargar cada fotograma al redibujar constantemente el contenido de datos vectoriales. Si bien es bastante simple aplicar filtros básicos, los avanzados necesitan mayor cantidad de datos para realizar las operaciones. Trabajando a nivel de bitmaps, tenemos más posibilidades de manipular las imágenes, aunque muchas veces debemos tratarlas a nivel de píxel, lo que agrega complejidad.

OBJETOS

Un objeto **BitmapData** es un elemento simple que contiene, básicamente, una información de píxeles. En Flash, podemos crear un bitmap desde cero, asignándole un tamaño y un color base a la propiedad `BitmapData`. Esto es útil para generar imágenes desde cero, pero también podemos usar el método **clone** para copiar el contenido de cualquier imagen que hayamos cargado. Las operaciones sobre píxeles suelen manipular una gran cantidad de información: imaginemos que una simple imagen de 200 x 100 píxeles tiene 20.000 píxeles, cada uno con un color que puede ser distinto. Una de las características más interesantes es la manipulación de píxeles, para lo cual disponemos de cuatro métodos: `getPixel`, `getPixel32`, `setPixel` y `setPixel32`. Todos operan de manera similar, sólo que uno mantiene la información de transparencia mientras que el otro no lo hace (24 o 32 bits).



Thresholdw puede usarse para detectar la luminosidad de una imagen, como en este ejemplo.

OPERACIONES CON BITMAPS

Cuando trabajamos con bitmaps, podemos realizar distintas operaciones, como las que describiremos a continuación:

► **Crear rellenos:** usamos el método **fillRect** con dos parámetros, una instancia de `Rectangle` que define un área rectangular y un color para utilizar.

```

bmp.fillRect(new Rectangle(0,0,50,100),
  0xFFFF0000)

```

► **Copiar píxeles entre distintos bitmaps:** por medio del método **copyPixels**, para tomar píxeles desde

un bitmap a otro, usando también una instancia de `Rectangle` y de `Point`.

```

bmp.copyPixels(sourceBmp,
  new Rectangle(0,0,50,100),
  new Point(10, 10))

```

► **Copiar canales de color:** usamos el método **copyChannel**, que es bastante parecido a **copyPixels**, pero tiene dos parámetros adicionales: el canal por copiar y el de destino.

```
bmp.noise(1000, 0, 255,
BitmapDataChannel.RED |
BitmapDataChannel.GREEN |
BitmapDataChannel.BLUE,
false)
```

► **Generar ruido de píxeles con Noise:** utilizamos el método `noise` con cinco parámetros: **seed**, para generar la secuencia aleatoria; **low** y **high**, que determinan los valores máximos y mínimos para cada píxel, entre 0 y 255; el **canal del bitmap** al que será aplicado el ruido y, finalmente, un valor booleano que determina si el ruido va a ser aplicado a los tres

```
bmp.
copyChannels(sourceBmp,
sourceBmp.rect, new Point,
BitmapDataChannel.RED,
BitmapDataChannel.RED)
```

canales por igual.
► **Generar efectos orgánicos con PerlinNoise:** genera un ruido orgánico a través del filtro `perlinNoise`. Veremos un ejemplo en combinación con los umbrales de color.
► **Usar umbrales de color:** el método **Threshold** utiliza dos bitmaps, uno es el que será alterado, y el otro se usa como parámetro para los cálculos. Esta

operación es más compleja y lleva seis parámetros. Básicamente, su operación es comparar cada píxel del bitmap con un valor específico. Si la comparación falla, el píxel del bitmap destino es pasado a un color determinado; de lo contrario, no cambia y es copiado.

APLICAR FILTROS A LOS BITMAPS

Los filtros que ya hemos visto y detallado en las páginas anteriores también se aplican a los bitmaps. Por ejemplo, podemos hacer un efecto de mezclado (`dissolve`) entre dos bitmaps, a través del método `pixelDissolve`, que usa la información de un bitmap de inicio hacia un bitmap de destino en el que mezcla de a poco los píxeles. ◀

+ SOBRE LA CLASE BITMAPDATA

Los métodos de la clase `BitmapData` admiten efectos que no están disponibles a través de los filtros utilizados para los objetos de visualización que no sean mapas de bits. Un objeto `BitmapData` contiene una matriz de datos de píxeles que pueden representar un mapa de bits completamente opaco o uno transparente que contiene datos del canal alpha. Ambos tipos de objetos se almacenan como búfer de enteros de 32 bits. Cada entero de 32 bits es una combinación de cuatro valores de canal de 8 bits (de 0 a 255), que describen los valores de transparencia alpha, y rojo, verde y azul (ARGB) del píxel.

```
es - fotograma
import flash.display.Bitmap;
import flash.display.BitmapData;
import flash.events.Event;

var bmp:BitmapData = new BitmapData(stage.stageWidth, stage.stageHeight, false);
var image:Bitmap = new Bitmap(bmp);
addChild(image);
var seed:Number = Math.random() * 10000;
var pixelCount:Number = 0;
addEventListener(Event.ENTER_FRAME, onEnterFrame);
var numPixels:Number = bmp.width * bmp.height / 100;

function onEnterFrame(evt:Event):void
{
    seed = bmp.pixelDissolve(bmp, bmp.rect, new Point(), seed, numPixels,
    pixelCount += numPixels;
    if (pixelCount > (bmp.width * bmp.height))
    {
        removeEventListener(Event.ENTER_FRAME, onEnterFrame);
    }
}
```

Cada iteración copia 1/100 de la imagen. A 30 frames por segundo, este dissolve tardaría unos 3 segundos.

GALERÍA DE FOTOS

En algún momento habremos tenido la necesidad de montar una galería de imágenes particular. Vamos a analizar cómo armar una y de qué modo utilizarla.

Tarde o temprano, en la medida en que vayamos avanzando en nuestra carrera, seguramente tengamos que construir una galería de fotos. No importa si hacemos sitios sociales, catálogos o banners, siempre necesitaremos incluir alguna galería de un tipo u otro. Las galerías pueden tener unas pocas fotos, lo cual no presentaría una gran dificultad, pero también pueden tener que mostrar cientos de ellas, y esto es algo que no siempre podremos resolver con exactitud. Por ejemplo, supongamos que en un sitio social los usuarios pueden subir fotos de una fiesta en la que han participado. Es difícil saber cuántas imágenes subirán, y por eso, lo mejor es plantear una galería que pueda crecer en forma dinámica. Entonces, vamos a definir los elementos de la estructura de una galería de fotos típica, para luego analizarlos con mayor precisión.

FUENTE DE DATOS

Puede ser un archivo de texto, un XML, una base de datos o un directorio del servidor indexado a través de algún script en PHP, Java, etc. La fuente de datos debe proveer de una ruta y un nombre de archivo para que la foto pueda ser cargada, aunque es posible también que brinde información adicional, como título de la foto, comentario, fecha y tema.

PREVISUALIZACIÓN EN PEQUEÑO

Este punto no es obligatorio, pero la mayoría de las galerías tienen alguna forma de previsualización del contenido mediante fotos de menor tamaño, para que el usuario tenga idea del contenido antes

de descargar el material. Esto dependerá, por supuesto, del espacio disponible y de cómo se instale la mecánica de visualización. Pero si vamos a mostrar una gran cantidad de fotos, es recomendable agregar la posibilidad de hacer una previsualización.

SIEMPRE ES RECOMENDABLE INCLUIR ALGÚN TIPO DE PRECARGA QUE NOS INDIQUE EL PROGRESO DE LA ACCIÓN.

En general, para ver las imágenes en pequeño tamaño se utilizan tomas distintas de las originales, ya que estas suelen ser bastante pesadas. Esto obliga a duplicar el contenido, con lo que tendremos, incluso, una carpeta con las imágenes pequeñas y otra con las originales, y así no necesitaremos cargar tantos archivos. Para este tipo de trabajos, sobre todo si la cantidad de imágenes es grande, recomendamos utilizar alguna clase de automatismo de Photoshop o, incluso, del mismo Fireworks. Ambas aplicaciones brindan la posibilidad de generar thumbnails desde una carpeta específica. Debemos evitar hacer el tratamiento en el servidor con alguna librería de manipulación de imágenes, porque por lo general, este tipo de operaciones

Acciones - fotograma

```
1 //Recibe el ancho y alto de la foto como parámetros
2 //devuelve un objeto con el nuevo ancho y alto
3 var maxWidth:Number = 400;
4 var maxHeight:Number = 300;
5 private function checkSize(w:Number, h:Number):Object{
6     if(h<=maxHeight && w<=maxWidth) {
7         return {h:h, w:w}
8     } else { //supera máximos
9         var orientation = (w>h)?"w" : "h"
10        //horizontal
11        if(orientation=="h"){
12            var per = int((maxHeight*100)/h)
13            var newWidth = int((w*per)/100)
14            return {h:maxHeight, w:newWidth}
15        } else { //vertical
16            var per = int((maxWidth*100)/w)
17            var newHeight = int((h*per)/100)
18            return {h:newHeight, w:maxWidth}
19        }
20    }
21 }
```

Rutina de reescalado que recibe el ancho y el alto de la foto actual, y modifica esos parámetros proporcionalmente para mantenerse dentro de un máximo.

demandan muchos recursos de memoria y procesador, y pueden dejar al servidor sin disponibilidad para los requerimientos del sitio.

IMÁGENES EN GRANDE Y PRECARGA

Las imágenes en grande son la pieza central de la galería, porque, al fin y al cabo, es lo que el usuario quiere ver cuando hace una selección. En este punto suele plantearse una discusión bastante recurrente, sobre todo, si lo que hacemos es un catálogo de productos, en donde el cliente quiere que todo se vea de la mejor manera posible. Los originales quizá pesen unos cuantos MB por foto. Entonces, el interrogante que surge es: ¿cuál es el peso adecuado? La respuesta tiene varias implicancias, por ejemplo, la expectativa del usuario de observar el material. Pero si nos ubicamos en una situación promedio, tal vez tengamos que mejorar todo lo posible la experiencia para que los visitantes permanezcan en el sitio.

De este modo, será necesario tener en cuenta la relación entre peso y velocidad de navegación. Por ejemplo, una conexión ADSL promedio puede bajar entre 20/30 KB por segundo, con lo cual una imagen de unos 100 KB tardará un poco más de 3 segundos en descargarse. Éste es el límite de lo tolerable. A medida que nos vamos alejando de estos valores, entramos en un territorio peligroso, en donde

PREVISUALIZACIÓN Y PRECARGA

Con respecto al tamaño de las imágenes, cuando utilizamos una imagen más chica para previsualizar, debemos saber que el original de la foto grande necesita tener un tamaño lo más cercano posible a las dimensiones en que se va a ver. Si el tipo de foto no es uniforme, es probable que de todos modos tengamos que hacer algún tipo de reescalado para mostrarla. Esto nos obliga a tener la precarga realizada, ya que no podemos reescalar una foto si no está totalmente cargada.

seguramente estaremos jugando con la paciencia del usuario. Otro aspecto importante, sobre todo si las fotos son de gran tamaño, es incluir algún tipo de precarga que indique el progreso. De lo contrario, generaremos muchos espacios en blanco, ya que durante gran parte del tiempo en que vemos una galería de fotos, se estará cargando lo que vamos a ver después, y es imperativo que el usuario sepa lo que está ocurriendo.

TRANSICIONES Y EFECTOS

Existen numerosas galerías de fotos en la Web, con efectos sofisticados y perspectivas 3D. Los efectos que se pueden

aplicar son muy variados, tanto en la presentación de las miniaturas como en la transición entre una y otra foto. Quizás el más utilizado sea el fade entre imágenes, pero trabajando con bitmaps y filtros, es posible lograr resultados mucho más interesantes. Por ejemplo, usar blur, saturar colores, hacer dissolve y estallar las fotos en píxeles, entre otras alternativas. La clase TransitionManager es muy útil para construir transiciones rápidamente. Las opciones, además del fade, son: Blinds, Fly, Iris, Photo, PixelDissolve, Rotate, Squeeze, Wipe y Zoom. Cada una tiene parámetros específicos que se pueden consultar en la ayuda de Flash. ◀

Ejemplo de galería con dos columnas de thumbnails y la foto grande en el medio.

Acciones - fotograma

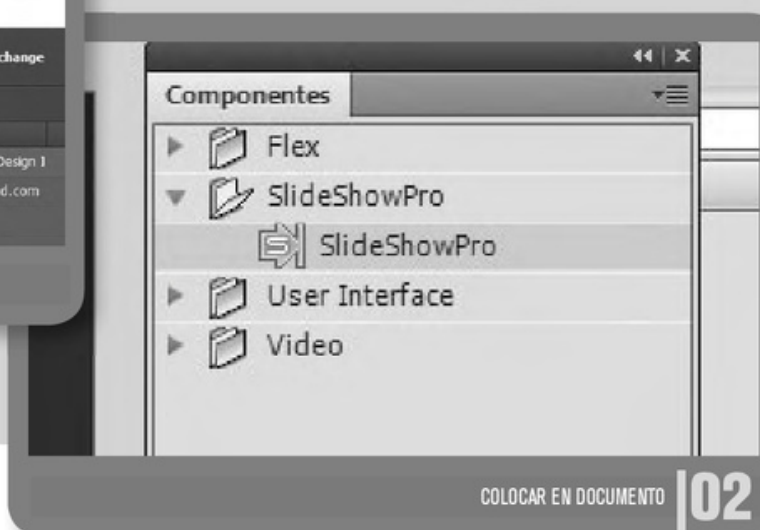
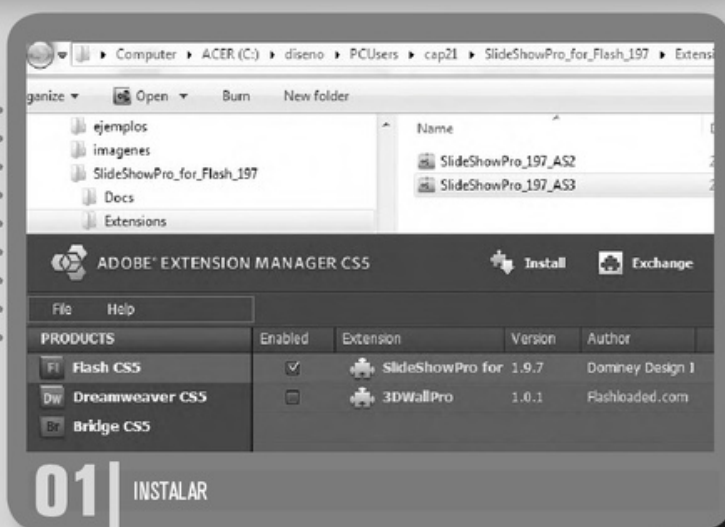
```
1 import fl.transitions.*;
2 import fl.transitions.easing.*;
3 import flash.display.MovieClip;
4
5 //fade entre foto "desde" y "hasta"
6 function fade(desde:MovieClip, hasta:MovieClip){
7     TransitionManager.start(desde, {type:Fade, direction:Transition.OUT,
8     TransitionManager.start(hasta, {type:Fade, direction:Transition.IN, c
9 }
10 //ejemplo de uso
11 fade(foto1, foto2)
12
13
```

Una transición de fade entre dos imágenes, en donde el uso de la clase TransitionManager simplifica mucho la tarea.

COMPONENTES

PASO A PASO

Slideshow Pro es un sistema para publicar fotos en la Web a través de Flash, de manera simple y eficiente. Veamos cómo utilizarlo paso a paso.



Slideshow Pro es un componente que permite crear álbumes de imágenes de manera rápida y eficiente, con la posibilidad de que los usuarios más avanzados personalicen el componente a través de código. Veamos en detalle cómo utilizar y aplicar este sistema. ◀

PASO | 01 Para instalar Slideshow Pro, primero debemos descargar el componente desde <http://slideshowpro.net>. Luego, abrimos el archivo comprimido, y dentro de la carpeta **Extensions**, ejecutamos **SlideShowPro_197_AS3.mxp**. Esto instalará el componente desde el **[Extension Manager]**.

PASO | 02 Una vez instalado el componente, lo encontraremos en el **[Panel de Componentes]** abriendo el nodo **SlideShowPro**. Lo arrastramos al escenario y lo colocamos en las coordenadas 0,0, para que nuestro slideshow ocupe toda la película. Luego de guardarlo en nuestro disco rígido, debemos elegir desde dónde queremos cargar el contenido.

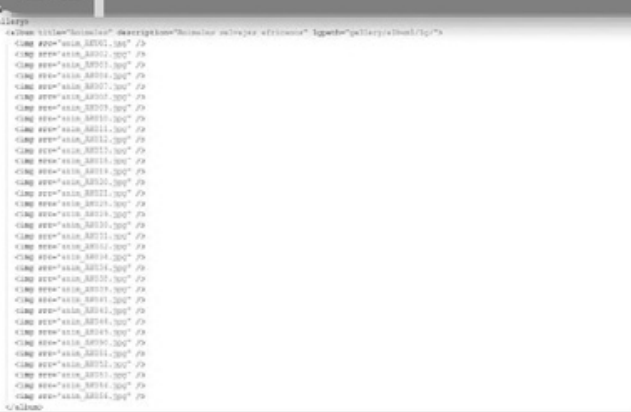
PASO | 03 El componente permite elegir entre tres tipos diferentes de fuentes: contenido creado manualmente, contenido extraído de RSS feeds y álbumes/galerías generadas desde **SlideshowProDirector** (un Content Management System de los que permiten generar y manejar fotos localmente).

PASO | 04 Vamos a explicar cómo trabajar manualmente con las fotos. En el directorio donde hayamos creado el **FLA** del slideshow,



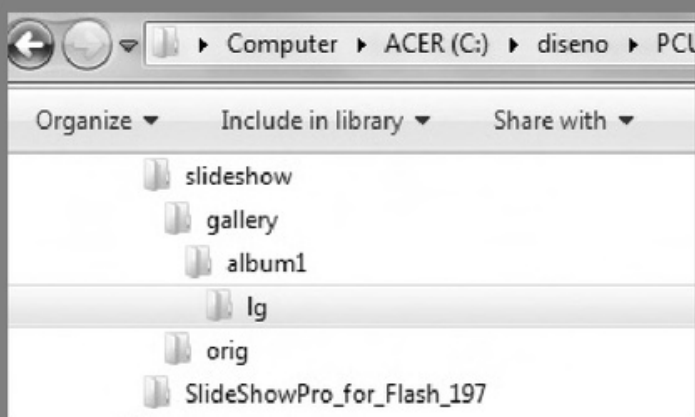
03

SELECCIONAR FUENTE DE DATOS



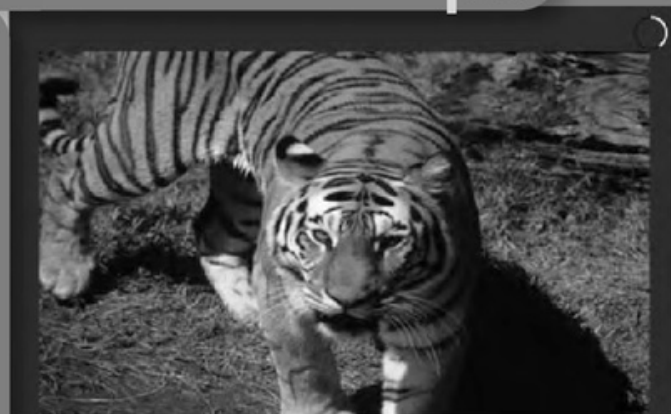
CREAR DOCUMENTO XML

05



04

CREAR DIRECTORIO DE FOTOS



PUBLICAR

06

generamos un directorio llamado **gallery**, y dentro de él, creamos un nuevo directorio, denominado **album1**. Este tendrá todas las imágenes del primer álbum. Dentro de este último, abrimos un nuevo directorio llamado **lg**, en donde vamos a colocar todas las fotos que queramos mostrar.

PASO | 05

Escribimos lo que muestra la pantalla.

Este es el esquema básico del documento. Lo guardamos en el

mismo directorio que el FLA y lo llamamos **images.xml**.

A continuación, llenamos los datos en el XML: ponemos el título, la descripción en la propiedad y la ruta a la carpeta **lg** (desde el FLA). Tendremos un nodo **img** por cada foto, donde pondremos el nombre del archivo en la propiedad **src**, sin indicar la ruta, que ya está especificada en la propiedad **lgpath**.

PASO | 06

Al oprimir **<Ctrl + Enter>**, publicamos el slideshow. Si vamos a las opciones de

publicación, en la solapa **[Flash]**, debemos asegurarnos de estar publicando para Flash Player 9 o superior, AS3. Para ver el resultado en el navegador, cargamos el HTML generado. Si queremos colocarlo en nuestro servidor, subimos el SWF y todas las carpetas creadas para **album1** (**gallery/album1/lg/**) junto a las fotos.

RESUMEN

Aprendimos a instalar el componente Slideshow Pro para crear álbumes de imágenes. También vimos cómo colocar textos, y cómo subir todo al servidor. Si bien éste es un ejemplo, el proceso se puede utilizar para proyectos generales.

CREAR UN SLIDESHOW DINÁMICO

PASO A PASO

En este apartado práctico conoceremos en detalle una técnica óptima para destacar información gráfica y textual en un sitio web.

```

1 import flash.events.Event;
2
3 // Importo las clases de Tweening
4 import com.greensock.*;
5 import com.greensock.easing.*;
6
7 stage.align = StageAlign.TOP_LEFT;
8 stage.scaleMode = StageScaleMode.NO_SCALE;
9
10
11 function handleResize(e:Event):void {
12     background.x = 0;
13     background.y = 0;
14     background.width = stage.stageWidth;
15     background.height = stage.stageHeight;
16
17     header.back.width = stage.stageWidth;
18     header.x = 0;
19     header.y = 0;
20     header.menuHeader.x = stage.stageWidth - header.menuHeader.width;
21
22     footer.back.width = stage.stageWidth;
23     footer.x = 0;
24     footer.y = stage.stageHeight - footer.back.height;
25
26     footer.scroll.x = stage.stageWidth / 2 - footer.scroll.width / 2;
27     footer.copyright.x = stage.stageWidth - 10;
28     footer.full.x = 10;
29 }
30
31 stage.addEventListener(Event.RESIZE, handleResize);
32 stage.dispatchEvent(new Event(Event.RESIZE));
  
```

01 CREAR XML

Línea de tiempo Salida Editor de movimiento

- Stage Align
- Footer
- Header
- Menu
- Backgrou...

CREAR CAPAS

02

La aplicación de un slideshow dinámico con imágenes animadas nos permite crear enlaces diferentes para una imagen. Esta técnica se utiliza para captar la atención del usuario mediante un slide que aparece en la zona inferior de la fotografía, en donde se puede mostrar texto personalizado para cada caso. Las transiciones entre imágenes son progresivas. ►

PASO | 01 Creamos el archivo XML con la estructura de datos para nuestro slideshow. La idea es mostrar los trabajos en donde el atributo **destacado** sea igual a 1. El bloque **<categoria>** puede repetirse en todas las categorías que tengamos. Lo mismo sucede con **<trabajo>** y, dentro de esta última, agregamos los tags **<titulo>**, **<descripcion_corta>** e **<imagen>**, que contiene la ruta a la fotografía.

PASO | 02 Creamos cuatro capas, en las cuales pondremos el código **ActionScript** de manera separada, para ubicarlo mejor al realizar modificaciones. Los nombres de los layers serán: **Variables / Clases**, **Stage Align**, **XML** y **Functions**. Abrimos el panel de acciones en Variables / Clases y declaramos las variables que vamos a utilizar en el slideshow. Luego, importamos la librería TweenMax.



```

13 }
14 function onComplete (event : Event) : void
15 {
16     myXML=XML(loadUrl.data);
17     nNodos=new XMLList(myXML);
18     for(var i:uint = 0; i<nNodos.categoria.length(); i++){
19         for(var n:uint = 0; n<nNodos.categoria[i].trabajo.length(); n++){
20             // Guardo en el Array todas las imagenes que van a estar en el array
21             if(nNodos.categoria[i].trabajo[n].destacado == "1"){
22                 objHome.push({
23                     titulo:nNodos.categoria[i].trabajo[n].titulo,
24                     descripcion:nNodos.categoria[i].trabajo[n].descripcion,
25                     imagen:nNodos.categoria[i].trabajo[n].imagen
26                 });
27             }
28         }
29     }
30     loadUrl.removeEventListener (Event.COMPLETE, onComplete);
31     loadUrl.removeEventListener (ProgressEvent.PROGRESS, progressHandler);
32
33     xml_loaded = true;
34     startData();
35 }

```

04

GUARDAR DATOS

```

mov_btPresupuesto
mov_btsFull
mov_contenedor
mov_contenedorImagenes
mov_contentSlideTexto
mov_contImagenHome mcImagen
mov_contLoad
mov_flechaSlide
mov_footer
mov_footerCopy
mov_header

```

CREAR EL MOVIE CLIP

05

```

function startData(){
    switch (seccionActiva){
        case "home":
            for(var i:uint = 0; i<objHome.length; i++){
                var imgSlide:mcImagen = new mcImagen();
                imgSlide.x = (imgSlide.width + 10) * i;
                imgSlide.y = 10;
                imgSlide.name = "mc" + i;
                imgSlide.id = i;
                imgSlide.info.alpha = 0;
                imgSlide.contLoad.alpha = 0;
                imgSlide.info.titulo.text = objHome[i].titulo;
                imgSlide.info.texto.text = objHome[i].descripcion;
                mcAr.push(imgSlide);
                contenedor.seccion.cont.addChild(imgSlide);
            }
        }
    }

```

DUPLICAR EL MOVIE CLIP

06

03

REALIZAR LA CARGA

PASO | 03

En este paso debemos realizar la carga del **XML**. Una vez completado el proceso, convertimos la información obtenida en un **XMLList**.

PASO | 04

En esta instancia vamos a guardar los datos necesarios para el slideshow, dentro de un **array de objetos**, como muestra la imagen. La variable **xml_loaded** permite saber si el **XML** fue cargado o no.

PASO | 05

Creamos el movie clip contenedor para nuestros trabajos. Este debería de incluir un clip para la carga de la imagen, otro para mostrar el cargador y otro más que contenga el bloque de información, que consiste en un texto para el título y otro para la descripción, junto al pequeño fondo en degradé detrás del texto. Recordemos agregarle un nombre de clase, como **mcImagen**.

PASO | 06

Vamos a duplicar el movie clip sobre la base de la cantidad de información que haya en nuestro array de objetos. Por medio de **addChild**, vamos agregando al clip **contenedor.seccion.cont** (previamente creado) cada una de las instancias de **mcImagen**.

CREAR UN SLIDESHOW DINÁMICO

PASO A PASO

07 AGREGAR INFORMACIÓN

MISIÓN PROYECTO
movieClip: mc1 | Título: Curso Visual y Diseño Web Design | Descripción: Curso de Photoshop para aprender y experimentar con la herramienta digital y digital.
movieClip: mc2 | Título: Curso WEB | Descripción: Revista de noticias online con tipo, diseño, experimentación, y mucho más.
movieClip: mc3 | Título: Curso WEB | Descripción: Revista de noticias online con tipo, diseño, experimentación, y mucho más.
movieClip: mc4 | Título: Diseño | Descripción: Diseño de interfaces de usuario.
movieClip: mc5 | Título: Diseño | Descripción: Diseño de interfaces de usuario.
movieClip: mc6 | Título: Diseño | Descripción: Diseño de interfaces de usuario.
movieClip: mc7 | Título: Diseño | Descripción: Diseño de interfaces de usuario.
movieClip: mc8 | Título: Diseño | Descripción: Diseño de interfaces de usuario.
movieClip: mc9 | Título: Diseño | Descripción: Diseño de interfaces de usuario.

TEXTO E IMÁGENES

08

```
OUTPUT
Carga Imagen: imagenes/portfolio/print/img2.jpg | en el movieClip:imagen
Carga Imagen: imagenes/portfolio/print/img1.jpg | en el movieClip:imagen
Carga Imagen: imagenes/portfolio/print/img3.jpg | en el movieClip:imagen
Carga Imagen: imagenes/portfolio/print/img4.jpg | en el movieClip:imagen
Carga Imagen: imagenes/portfolio/web/img4.jpg | en el movieClip:imagen
Carga Imagen: imagenes/portfolio/illustracion/img1.jpg | en el movieClip:imagen
Carga Imagen: imagenes/portfolio/illustracion/img2.jpg | en el movieClip:imagen
Carga Imagen: imagenes/portfolio/illustracion/img3.jpg | en el movieClip:imagen
Carga Imagen: imagenes/portfolio/multimedia/img1.jpg | en el movieClip:imagen
Carga Imagen: imagenes/portfolio/multimedia/img2.jpg | en el movieClip:imagen
```

```
function overIMGSlide(e:Event):void{}
function outIMGSlide(e:Event):void{}

contenedor.section.cont.getChildByName("mc"+i).addEventListener(MouseEvent.CLICK, overIMGSlide);
contenedor.section.cont.getChildByName("mc"+i).addEventListener(MouseEvent.ROLL_OUT, outIMGSlide);
```

09 CARGAR Y MOSTRAR

PASO | 07 Ahora, añadimos información extra a estos movie clips: una variable id, un name, y una posición en X, Y. Volveremos invisibles el panel de información y el cargador. También agregaremos el texto de título y el correspondiente a la descripción. Guardamos una referencia del movie clip creado dentro del array **mcAr**.

PASO | 08 En este paso nos ocuparemos del texto y de las imágenes. Creamos las funciones **overIMGSlide** y **outIMGSlide**, que, por ahora, dejamos vacías. Con este agregado, lograremos que el texto aparezca y desaparezca por medio de las acciones de **MouseEvent.ROLL_OVER** y **MouseEvent.ROLL_OUT**.

PASO | 09 Debemos crear una función que se ocupe de cargar y mostrar la imagen dentro

del movie clip que nosotros le indiquemos. La llamamos **function loadThumbs()**. Esta función esperará dos parámetros: **placeholder:MovieClip**, para recibir la referencia del clip a donde se cargará la imagen; y **path:String**, para recibir como referencia la ruta en donde está la imagen.

PASO | 10 En la función asociada al método **ProgressEvent.PROGRESS** del cargador de imágenes, agregamos el código para representar el porcentaje de carga de cada una. Las funciones **overIMGSlide** y **outIMGSlide** tendrán el mismo código, salvo por el efecto de blur. En la acción **over**, el blur tendrá un valor de 8 y, en la acción **out**, será de 0. Cuando se active el evento **MouseEvent.ROLL_OVER**, se van a **desenfocar** todas las imágenes menos la seleccionada, en tanto que **MouseEvent.ROLL_OUT** devolverá el foco a todas ellas.

```

344 }
345 TweenLite.to(contenedor.seccion.cont,1,{x:(xToGo * -1) + (stage.stageWidth/2 - mcAr[id].width/2),ease:Strong.easeOut});
346 }
347
348 function loadThumb(placeHolder:MovieClip,path:String) void {
349     var loader:Loader;
350     loader = new Loader();
351     loader.load(new URLRequest(path));
352
353     var mcContent = placeHolder.parent.getChildByName("content") as MovieClip;
354     mcContent.loadProgress.text = "0%";
355     TweenMax.to(mcContent,1,{alpha:1,ease:Strong.easeOut});
356
357     loader.contentLoaderInfo.addEventListener(Event.COMPLETE,handleComplete);
358     loader.contentLoaderInfo.addEventListener(ProgressEvent.PROGRESS,handleProgress);
359
360     function handleComplete(e:Event) void {
361         TweenMax.to(mcContent,1,{alpha:1,ease:Strong.easeOut});
362         placeHolder.addChild(loader.content);
363         loader.contentLoaderInfo.removeEventListener(Event.COMPLETE,handleComplete);
364         loader.contentLoaderInfo.removeEventListener(ProgressEvent.PROGRESS,handleProgress);
365         loader = null;
366     }
367
368     function handleProgress(event:ProgressEvent) void {
369         var porcentaje:Number = Math.round(event.bytesLoaded/event.bytesTotal * 100);
370         mcContent.loadProgress.text = porcentaje.toString() + "%";
371     }
372 }
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

10 ENFOQUE



ANTERIOR Y SIGUIENTE

12

```

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

11 DECLARAR FUNCIONES



DESGLIZAR

13

PASO | 11 Vamos a declarar las funciones que se ocuparán de habilitar y deshabilitar las flechas de navegación del slideshow. Después de realizar esto, les agregamos el contenido. Veamos cómo sería para una flecha, y luego, cada uno completará la versión para la segunda.

PASO | 12 Una vez replicada la función para el botón anterior, vamos a la función **startData()** y, debajo del bucle for, agregamos la llamada a dos funciones:

```

activoSiguienteSlide();
desactivoAnteriorSlide()

```

PASO | 13 Nos falta dar de alta una función más, redactando el siguiente código:

```

function navigate(id:int,resize:Boolean):void {
    var xToGo:Number = 0;
    for (var i:int = 0; i < mcAr.length; i++) {
        if (i < id) {
            xToGo += (mcAr[i].width + spacing);
        }
        TweenLite.to(contenedor.seccion.cont,1,{x:(xToGo * -1) + (stage.stageWidth/2 - mcAr[id].width/2),ease:Strong.easeOut});
    }
}

```

RESUMEN

Hemos aprendido a destacar información gráfica en un proyecto web. Es importante mencionar que cada uno de estos pasos es arbitrario, pero puede aplicarse a cualquiera de nuestros proyectos. Las diferencias tienen que ver con la cantidad de textos y objetos que seleccionemos.

ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Cómo podemos incorporar imágenes a nuestro proyecto?

02 ¿Por qué es necesario optimizar las imágenes?

03 ¿Qué son los filtros dinámicos?

04 ¿Qué es bitmap?

05 Defina bitmapdata.

06 ¿Qué ventajas nos entrega una galería de fotos?

07 ¿Qué son los componentes?

08 ¿Para qué sirve **Slideshow Pro**?

09 ¿Qué ventaja ofrece un slideshow dinámico?

10 Mencione las herramientas necesarias para crear un slideshow dinámico.

PRÁCTICAS

01 Incorpore algunas imágenes en un proyecto de Flash.

02 Optimice las imágenes.

03 Cree una galería de fotos para su proyecto.

04 Utilice **Slideshow Pro**.

05 Cree un slideshow dinámico.

Formularios

- ▶ FORMULARIOS EN FLASH
- ▶ CÓMO CREAR UN FORMULARIO
- ▶ TABULACIÓN Y VALIDACIÓN DE FORMULARIOS
- ▶ TEXTO PREDICTIVO
- ▶ ENVÍO DE FORMULARIOS
- ▶ CONTENT MANAGEMENT SYSTEM
- ▶ GESTORES DE CONTENIDO
- ▶ FLASHBLOCKS



FORMULARIOS EN FLASH

Flash no solo permite crear páginas web animadas, sino que, además, nos da la posibilidad de realizar las tareas necesarias para llevar adelante un sitio; una de ellas es la creación de formularios.

Un formulario, básicamente, está compuesto por dos partes: la primera es la programación del lado de nuestro SWF (navegador), es decir, el código ActionScript; la segunda tiene que ver con el servidor del código PHP. En esta instancia, veremos los recursos que Flash pone a nuestra disposición para programar en este lenguaje. La herramienta principal para la creación de formularios son los campos de texto de tipo **input**, destinados a que el usuario ingrese datos. A su vez y, en caso de ser necesario, podemos asignarle al mismo input la opción de password. De este modo, todo lo que se coloque en el mismo campo se mostrará como asteriscos o círculos (sin revelar los caracteres).

LA PRINCIPAL HERRAMIENTA PARA LA CREACIÓN DE FORMULARIOS SON LOS CAMPOS DE TEXTO DE TIPO INPUT.

Otra de las opciones para estos campos es que sean single line o multiline. Esto se determina al armar el formulario, y sirve para darle la posibilidad al usuario de que ingrese una o más líneas de texto dentro de ellos. Flash, desde el panel de **[Componentes]**, brinda una amplia variedad de opciones para los formularios, como CheckBox, RadioButtons y

ComboBox, entre otras. Sin embargo, también existe la posibilidad de que nosotros mismos armemos nuestros "componentes". Esto dependerá totalmente de las ganas que tengamos de seguir afianzando nuestro conocimiento.

RESTRICCIONES

Al tener que manipular campos de texto, una de las propiedades más interesantes que nos ofrece Flash es la llamada **restrict**, cuya función, básicamente, es restringir los caracteres que vayan a ingresarse; por ejemplo:

```
campo_de_texto.  
restrict = "A-Z 0-9";
```

En este caso, lo que estamos haciendo es limitar al campo de texto para que puedan ingresarse solo caracteres en mayúsculas, espacios y números. También, si lo que queremos es excluir un grupo de caracteres, debemos anteponer el símbolo "^"; por ejemplo:

```
campo_de_texto.  
restrict = "^a-z";
```

De este modo, estaremos excluyendo todos los caracteres de la A hasta la Z en minúscula. Esta es una de las propiedades más interesantes que podemos encontrar a la hora de prevenir el ingreso



de caracteres indeseados dentro de los campos, a la vez que nos facilitará, más adelante, su validación.

CONTROLADORES DE EVENTOS

Los controladores de eventos son una de las mejoras que ofrece ActionScript 3 con respecto a su predecesor.

Se utilizan para realizar un seguimiento luego de enviar los datos al servidor; los más habituales son:

- **complete:** se ejecuta una vez que se completa la carga de los datos de respuesta, y que estos se guardan en el objeto "data" para su posterior manipulación.
- **httpStatus:** devuelve el código de estado de la solicitud actual.
- **ioError:** se ejecuta cuando el error producido obliga a que se corte la conexión con el archivo en cuestión.
- **progress:** se ejecuta a medida que se van descargando los paquetes de

información. De esta manera, y por medio de este método, podemos crear una animación de carga o mostrar su porcentaje de progreso.

- **securityError:** este evento es llamado cuando Flash detecta que la carga de datos está infringiendo los niveles de seguridad del player.

PROPIEDAD URLREQUEST

Los controladores o eventos mencionados previamente surgen de la clase **URLLoader**. Esta, junto a **URLRequest**, se ocupan de capturar toda la información de ida y vuelta de una solicitud http. Comencemos viendo las propiedades de **URLRequest**:

- **ContentType:** permite definir el tipo de contenido MIME.
- **Data:** trabaja directamente con la propiedad **method**, de modo que los datos que definamos en ella serán

enviados como se declaró en esta última.

- **Method:** define el método de envío de los datos (**URLRequestMethod.GET** o **URLRequestMethod.POST**).
- **RequestHeaders:** por medio de una matriz (array), podremos enviarle una cabecera distinta.
- **Url:** es la dirección que se va a solicitar, la cual debe encontrarse en el mismo servidor que el SWF. En caso contrario, tendremos que utilizar un archivo de política de seguridad, conocido como **crossdomain**.

PROPIEDAD URLLOADER

Nos queda por describir **URLLoader**, que, junto a los eventos ya explicados, contiene las siguientes propiedades:

- **bytesLoaded:** proporciona información sobre los bytes cargados al momento de hacer la consulta.
- **bytesTotal:** indica el total de bytes por cargar.
- **data:** es el contenedor de los datos recibidos. Una vez finalizada la carga, estos datos se recibirán en el formato seleccionado en **dataFormat**.
- **dataFormat:** determina el tipo de dato que se va a recibir (TEXT, BINARY, VARIABLES).

De este modo, hemos listado los controladores o eventos de las clases **URLLoader** y **URLRequest**. ◀



SOBRE CROSSDOMAIN

Se trata de un mecanismo (archivo) de seguridad de las comunicaciones en navegadores. Evita que un script, como XMLHttpRequest de AJAX o una aplicación de Flash o Silverlight, acceda a un servidor web diferente de aquel en el que reside. Un ejemplo de este tipo de archivo **crossdomain** puede observarse en los servidores de Flickr. Cuando Flash hace una carga de un archivo alojado en otro dominio, busca en este dominio el **crossdomain.xml** y, si no lo encuentra, no permite la carga.

CÓMO CREAR UN

PASO A PASO

Aprovechando las herramientas que nos proporciona Flash, vamos a realizar nuestro primer formulario con ActionScript 3 y PHP.

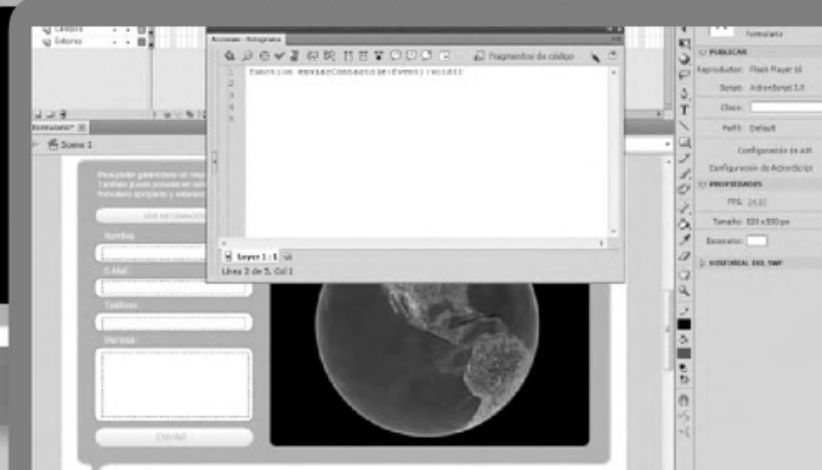
Para poder garantizarle un mejor servicio, por favor utilice nuestro formulario de contacto y normalmente le contestaremos en un par de días. También puede ponerse en contacto con nuestros socios directamente. Si está interesado en recibir una propuesta por favor complete el formulario adjunto y nos enviaremos de inmediato.

VER INFORMACIÓN

Nombre:
E-Mail:
Teléfono:
Mensaje:

ENVIAR

01 | ESTÉTICA DEL FORMULARIO



PROGRAMACIÓN | 02

Existen muchas maneras de crear formularios para aplicar en nuestros sitios web. Podemos utilizar JavaScript y HTML, entre otras opciones, pero en este apartado práctico, explicaremos en detalle cómo lograrlo con ActionScript 3.0 y PHP. Recordemos que los formularios son una parte muy importante de un sitio, sobre todo, en lo que respecta a la nominalización de la base de datos de usuarios. ▶

PASO | 01 Utilizando las herramientas básicas de Flash, armaremos nuestro formulario de contacto. El proceso puede verse en los archivos fuente que acompañan a la colección, junto con los nombres de instancia y demás elementos.

PASO | 02 Una vez armado el formulario a nivel estético, pasaremos a la programación correspondiente, para lo cual creamos nuestra primera función: enviarContacto, tal como muestra el código:

```
function enviarContacto(e:Event):void{}
```

PASO | 03 Después de declarar la función del paso anterior, le agregamos un Event Listener al botón que iniciará el envío

FORMULARIO

04 | ENVIAR CONTACTO

```

1 sendContacto.addEventListener(MouseEvent.CLICK, enviarContacto)
2
3 function enviarContacto(e:Event):void{
4     var url:String = "contacto.php";
5     var variables:URLVariables = new URLVariables();
6     var request:URLRequest = new URLRequest(url);
7     var loader:URLLoader = new URLLoader();
8
9     variables.dataFormat = URLLoaderDataFormat.TEXT;
10    variables.nombre = e.target.parent.nombre.text;
11    variables.mail = e.target.parent.mail.text;
12    variables.telefono = e.target.parent.telefono.text;
13    variables.mensaje = e.target.parent.mensaje.text;
14
15    request.method = URLRequestMethod.POST;
16    request.data = variables;
17
18    loader.dataFormat = URLLoaderDataFormat.VARIABLES;
19    loader.addEventListener(Event.COMPLETE, handleCompleteContacto);
20    loader.addEventListener(IOErrorEvent.IO_ERROR, onIOError);
21    loader.load(request);
22
23

```

```

sendContacto.addEventListener(MouseEvent.CLICK, enviarContacto)

function enviarContacto(e:Event):void{
    var url:String = "contacto.php";
    var variables:URLVariables = new URLVariables();
    var request:URLRequest = new URLRequest(url);
    var loader:URLLoader = new URLLoader();

}

```

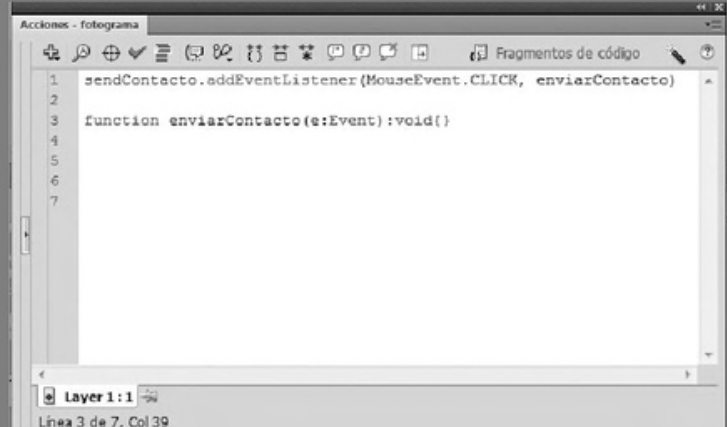
INSTANCIAS | 05

```

4 var url:String = "contacto.php";
5 var variables:URLVariables = new URLVariables();
6 var request:URLRequest = new URLRequest(url);
7 var loader:URLLoader = new URLLoader();
8
9 variables.dataFormat = URLLoaderDataFormat.TEXT;
10 variables.nombre = e.target.parent.nombre.text;
11 variables.mail = e.target.parent.mail.text;
12 variables.telefono = e.target.parent.telefono.text;
13 variables.mensaje = e.target.parent.mensaje.text;
14
15 request.method = URLRequestMethod.POST;
16 request.data = variables;
17
18 loader.dataFormat = URLLoaderDataFormat.VARIABLES;
19 loader.addEventListener(Event.COMPLETE, handleCompleteContacto);
20 loader.addEventListener(IOErrorEvent.IO_ERROR, onIOError);
21 loader.load(request);

```

GUARDAR INFORMACIÓN | 06



03 | AGREGAR EVENTO

del formulario. En este caso, su nombre de instancia es `sendContacto`:

```
sendContacto.addEventListener(
    MouseEvent.CLICK, enviarContacto)
```

PASO | 04 Luego de agregar el evento del paso anterior y de haber colocado el nombre de instancia, veremos con qué código se completará la función de envío. Básicamente, toda la programación que se vea en los pasos siguientes estará dentro de la función **enviarContacto**.

PASO | 05 El primer paso es colocar en instancias las clases que serán necesarias para mandar la información:

También vamos a guardar en una variable de tipo string la ruta al PHP que se ocupará de procesar la información recibida y de realizar el envío del e-mail:

```
function enviarContacto(e:Event):void{}
```

PASO | 06 En este paso, vamos a guardar la información de los campos dentro de la instancia de la clase `URLVariables`, que, en nuestro ejemplo, se denomina "variables". Definimos la propiedad `dataFormat` con el valor `URLLoaderDataFormat.TEXT`. De esta manera, lo que hacemos es indicarle a la instancia "variables" que los datos por recibir serán de formato texto.

CÓMO CREAR UN FORMULARIO

PASO A PASO

```

6  var request:URLRequest = new URLRequest(uri);
7  var loader:URLLoader = new URLLoader();

9  variables.dataFormat = URLLoaderDataFormat.TEXT;
10 variables.nombre = e.target.parent.nombre.text;
11 variables.mail = e.target.parent.mail.text;
12 variables.telefono = e.target.parent.telefono.text;
13 variables.mensaje = e.target.parent.mensaje.text;

15 request.method = URLRequestMethod.POST;
16 request.data = variables;

18 loader.dataFormat = URLLoaderDataFormat.VARIABLES;
19 loader.addEventListener(Event.COMPLETE, handleCompleteContacto);
20 loader.addEventListener(IOErrorEvent.IO_ERROR, onIOError);
21 loader.load(request);
22
23

```

Layer 1:1
Línea 15 de 27, Col 1

07 MÉTODO DE ENVÍO

```

16 request.data = variables;

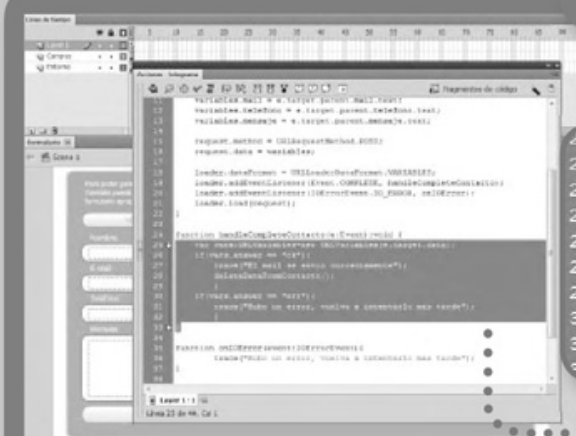
18 loader.dataFormat = URLLoaderDataFormat.VARIABLES;
19 loader.addEventListener(Event.COMPLETE, handle
20 loader.addEventListener(IOErrorEvent.IO_ERROR,
21 loader.load(request);
22
23
24 function handleCompleteContacto(e:Event):void {}
25 function onIOError(event:IOErrorEvent){}
26
27

```

Layer 1:1
Línea 24 de 27, Col 1

CONTROLADORES

08



```

23
24 function handleCompleteContacto(e:Event):void {
25     var vars:URLVariables=new URLVariables(e.target.data);
26     if(vars.answer == "ok"){
27         trace("El mail se envió correctamente");
28         deleteDataFormContacto();
29     }
30     if(vars.answer == "err"){
31         trace("Hubo un error, vuelva a intentarlo mas tarde");
32     }
33 }

```

09 EJECUTAR

PASO | 07 Vamos a definir el método para el envío de la información; es decir, si será por medio de GET o POST, la información que será enviada, y el formato en el que queremos recibir la información de retorno y los controladores para estas solicitudes.

PASO | 08 Los controladores que nos interesa utilizar en este momento son: Event.COMPLETE, que ejecutará la función; handleCompleteContacto; y IOErrorEvent.IO_ERROR, que ejecutará la función onIOError. Esto nos permite saber en el momento en que se completó el envío y, en caso de haber algún problema con el archivo, también conocer su estado.

PASO | 09 La función handleCompleteContacto se ejecuta una vez que finaliza el proceso de envío y recepción de datos. De esta forma, podemos mostrar distintos resultados, dependiendo de lo que nos devuelva

nuestro PHP. Por ejemplo, si el mail fue enviado, devolverá el string "ok" y, en caso contrario, "err". Esta respuesta estará dentro de la variable answer.

PASO | 10 Si el envío fue correcto, vamos a mostrar un mensaje que lo indique, y a limpiar los campos del formulario. De no ser así, mostraremos un mensaje para indicar que no pudo mandarse. Es importante destacar que la función onIOError se ejecutará en caso de que exista un problema con el archivo en sí:

```

function onIOError(event:IOErrorEvent)
{
    trace("Hubo un error,
    vuelva a intentarlo mas tarde");
}

```

```

35 function onIOError(event:IOErrorEvent){
36     trace("Hubo un error, vuelva a intentarlo mas tarde");

```

```

22 }
23
24 function handleCompleteContacto(e:Event):void {
25     var vars:URLVariables=new URLVariables(e.target.data);
26     if(vars.answer == "ok"){
27         trace("El mail se envio correctamente");
28         deleteDataFormContacto();
29     }
30     if(vars.answer == "err"){
31         trace("Hubo un error, vuelva a intentarlo mas tarde");
32     }
33 }
34
35 function onIOError(event:IOErrorEvent){
36     trace("Hubo un error, vuelva a intentarlo mas tarde");
37 }
38

```

Layer 1:1
línea 35 de 44, Col 1

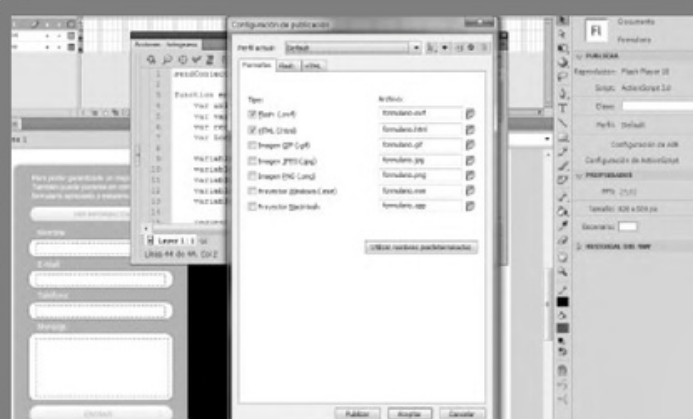
10 SOLUCIÓN DE ERROR

```

1 <?php
2
3 $mensaje = "Nombre: " . $_POST["nombre"] . "\n";
4 $mensaje .= "Telefono: " . $_POST["telefono"] . "\n";
5 $mensaje .= "Mail: " . $_POST["mail"] . "\n";
6 $mensaje .= "Mensaje: " . $_POST["mensaje"] . "\n";
7
8 $cabecera = "From: " . $_POST["nombre"] . "<".$_POST["mail"].">";
9 $cabecera .= "Reply-To: " . $_POST["nombre"] . "<".$_POST["mail"].">";
10
11 if(mail("fabian@fabianhr.com", "Formulario de Prueba", $mensaje, $cabecera)){
12     echo "answer-ok";
13 }else{
14     echo "answer-err";
15 }
16
17 ?>
18

```

11 ARCHIVO PHP



EXPORTACIÓN 12



VERIFICACIÓN DEL PROYECTO 13

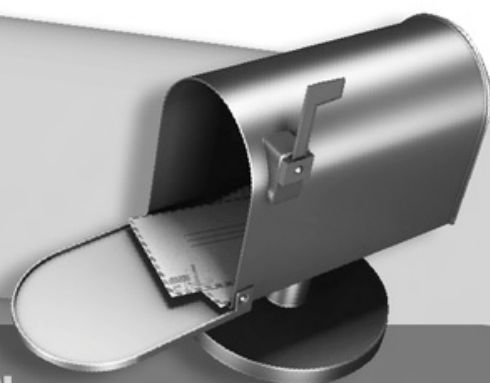
PASO 11 Una vez que completamos los pasos previos, vamos a escribir nuestro archivo PHP. Recordemos que el método de envío de datos que usaremos será POST. En la imagen se muestra el código que debemos redactar.

PASO 12 Ahora vamos a exportar el proyecto; pero antes, es necesario recordar que el PHP tiene que estar junto al SWF. En caso de que esté en otro lugar, tendremos que especificarlo en el código.

PASO 13 Después de realizar cada uno de los pasos detallados anteriormente, solo nos resta efectuar la verificación del proceso, para asegurarnos de no haber cometido ningún error.

RESUMEN

En este paso a paso, hemos aprendido a realizar nuestro primer formulario utilizando ActionScript 3 y PHP. Recordemos que estos ejemplos son simples para privilegiar la comprensión, pero pueden aplicarse a procedimientos más complejos.



TABULACIÓN Y VALIDACIÓN DE FORMULARIOS

Conoceremos cuáles son los aspectos que debemos tener en cuenta para mejorar la performance de nuestros formularios. Además, abordaremos las nociones de metacaracteres y metasecuencias.

Un formulario en Flash no se tabula automáticamente, tal como sucede con uno del tipo HTML. En nuestro caso debemos indicarle a Flash cuál es el orden que deseamos para esta tabulación y cómo debe comportarse con ciertos objetos. Básicamente, esto se lleva a cabo para facilitar la navegación de los campos que completará el usuario. Es habitual que aquellos usuarios que navegan asiduamente por páginas de Internet y completan formularios recurran a la tecla de tabulación para acceder rápidamente a los distintos campos. Por eso, es una muy buena práctica tabular cada uno de los formularios que hagamos. Este proceso se realiza por medio de la propiedad **tabIndex**, que se le asigna a cada campo aumentando su valor en 1.

```
nombre.tabIndex = 1
apellido.tabIndex = 2
direccion.tabIndex = 3
```

De este modo, cuando presionemos la tecla de tabulación, primero se pasará al campo **nombre**; luego, a **apellido**; y por último, a **dirección**. Si volviésemos a presionar la tecla <Tab>, teniendo el cursor sobre el campo dirección, automáticamente iríamos a **nombre**. Un aspecto importante que debemos tener en cuenta es que, una vez declarado un circuito de **tabIndex**, la tecla <Tab> solo cambiará la posición entre los objetos a los que les fue asignado un orden de tabulación.

VALIDACIÓN DE UN FORMULARIO

Son muchas las maneras y los métodos para validar los campos de un formulario,

y la elección dependerá, directamente, de nosotros como desarrolladores y de la comodidad que sintamos con un método u otro. Por ejemplo, tenemos la opción de utilizar **indexOf** en caso de querer validar la existencia de algún carácter en particular. Este método suele aplicarse a la validación de un mail, por ejemplo, y se implementaría de la siguiente manera:

```
if(mail.text.indexOf("@")
== -1){ trace("FALTA @ EN
EL CAMPO MAIL") }
```

Con esto, estaremos verificando si en el campo **mail** existe el símbolo @. En caso de que no exista, mostramos un "error"; de lo contrario, la validación se hace efectiva y el código sigue su curso. Además, hay un método mucho más amplio y, tal vez, un poco más complejo, pero más perfeccionado y efectivo. Nos referimos al uso de expresiones regulares. Como es mucha la información disponi-

ble al respecto, en este caso vamos a realizar una introducción a este mundo. En principio, podemos decir que las **expresiones regulares** son patrones para comprobar la existencia de ciertas cadenas de caracteres dentro de otras. Estos patrones pueden formarse por caracteres, metacaracteres, metasecuencias e indicadores. Cada una de estas opciones o elementos genera patrones de diferentes comportamientos, como la búsqueda de un simple string (el nombre "Juan") o la repetición del mismo string a lo largo de todas las líneas de un texto. Por su parte, los **indicadores** nos ayudan a refinar los resultados. Como ejemplos, citaremos solo algunos de ellos:

- ▶ **"i" (ignorecase)**: evitar que haya diferencia entre minúsculas y mayúsculas.
- ▶ **"x" (extended)**: permite utilizar caracteres especiales extendidos, como acentos y eñes, entre otros.
- ▶ **"g" (global)**: permite capturar todos los resultados obtenidos.
- ▶ **"m" (multiline)**: nos habilita para utilizar los metacaracteres "^" y "\$".



LAS EXPRESIONES REGULARES SON PATRONES PARA COMPROBAR LA EXISTENCIA DE CIERTAS CADENAS DE CARACTERES DENTRO DE OTRAS.

Además, existen los metacaracteres como **"^"**. En este caso, al incluirlo en un patrón junto al indicador **"m"**, le estamos indicando que busque al principio de la línea, mientras que con el signo **"\$"** deberá buscar al final de cada una. Es importante destacar que también están las metasecuencias, en donde tenemos la posibilidad de detectar secuencias de repeticiones definiendo la cantidad o frecuencia de estas.

METASECUENCIAS

Las metasecuencias son secuencias de caracteres que tienen un significado especial en un patrón de una expresión regular. Por ejemplo, la metasecuencia **\d** permite buscar cualquier dígito de 0 a 9 en una cadena. En otras palabras, las metasecuencias están formadas por más de un carácter. A continuación, describiremos algunas de ellas:

- ▶ **{n}**: especifica un cuantificador numérico o un rango de cuantificadores para el elemento anterior.
- ▶ **\b**: detecta la posición entre un carácter de palabra y uno de otro tipo. Si el primero o el último carácter de la cadena es una palabra, también detecta el principio o el final de la cadena.
- ▶ **\s**: detecta cualquier carácter de espacio en blanco (un carácter de espacio, tabulación, nueva línea o retorno).
- ▶ **\w**: detecta un carácter de palabra (AZ-, az-, 0-9 o _). Hay que tener en cuenta que no detecta caracteres que no sean los del idioma inglés, como é, ñ o ç.
- ▶ **\xnn**: detecta el carácter con el valor ASCII especificado, definido por el número hexadecimal nn.

Es importante destacar que solo hemos detallado algunas de las cadenas de caracteres que conforman una metasecuencia. ◀



LOS METACARACTERES Y SU APLICACIÓN

Recomendamos profundizar el conocimiento de estos significados y sus aplicaciones.

METACARÁCTER	APLICACIÓN
▶ ^ (intercalación)	▶ La intercalación después del corchete de apertura sirve para excluir caracteres de la búsqueda.
▶ \$ (signo dólar)	▶ Detecta el final de la cadena que se especifica antes del metacarácter.
▶ \ (barra diagonal inversa)	▶ Omite el significado especial de los metacaracteres.
▶ * (asterisco)	▶ Detecta cero o más ocurrencias del elemento anterior.
▶ + (signo más)	▶ Detecta una o más ocurrencias del elemento anterior.
▶ . (punto)	▶ Detecta cualquier carácter individual excepto la metasecuencia \n (nueva línea).
▶ () Paréntesis	▶ Define grupos de caracteres dentro del patrón.
▶ [] Corchetes	▶ Define clases de caracteres. Permite especificar posibles caracteres para una determinada posición de la cadena.
▶ ? (cierre interrogación)	▶ Detecta una o cero ocurrencias del elemento anterior.
▶ (barra vertical)	▶ Detecta el elemento que está a la izquierda o a la derecha. Es un metacarácter de alternancia.
▶ - (guión)	▶ Sirve para especificar rangos de caracteres.

TEXTO PREDICTIVO

En estas páginas profundizaremos en las aplicaciones del texto predictivo, también conocido como autocompletar, para saber cuándo y cómo utilizar este sistema en nuestros proyectos web.

El texto predictivo –también conocido como “autocompletar”– es otra de las opciones que suman posibilidades a la experiencia del usuario. Si bien no es la práctica más utilizada en el diseño web, es interesante tenerla en cuenta para integrarla en nuestros proyectos o conocer cómo funciona. Hoy en día, el sistema de autocompletar o texto predictivo se emplea con múltiples propósitos. Se utiliza tanto en campos de texto, para sugerir al usuario que complete lo que comenzó a escribir o, simplemente, para presentarle un listado de opciones para completar los caracteres que hasta ese momento el usuario haya ingresado en el campo correspondiente.

APLICACIÓN

El texto predictivo también puede encontrarse en los IDEs de programación o en los programas que utilizamos a diario para armar una página con Flash, para programar en PHP, HTML o CSS. En estos casos, las sugerencias van enfocadas, directamente, a ciertas opciones asociadas a los caracteres que el usuario escribe, como cuando nosotros, en Flash CS5, ingresamos:

```
Import flash.
```

Una vez que soltamos la tecla de “punto”, emergerá un listado de opciones asociadas a la librería Flash. Si continuáramos, entonces, pondríamos lo siguiente:

```
Import flash.events.
```

EL TEXTO PREDICTIVO INTENTA ASISTIR AL VISITANTE EN SU BÚSQUEDA, AUNQUE NO SIEMPRE ES BIEN ACEPTADO POR LOS USUARIOS.

Una vez más, el código mostraría un detalle de todos los eventos que podrían aplicarse. En nuestro caso, este sistema puede implementarse para facilitar el trabajo a los visitantes de una página. De esta manera, les estaríamos proporcionando atajos a ciertos procesos que podrían ser molestos. Es importante aclarar que no a todas las personas les gusta buscar el nombre de su país cada vez que tiene que seleccionarlo de un listado, o escribir una palabra completa. En caso de tener un listado muy extenso, con palabras

Un claro ejemplo de textos predictivo está presente en los diccionarios de cualquier teléfono móvil de la actualidad.





Yahoo.com, como parte del Astra Flash Components, nos brinda una solución muy práctica a la hora de utilizar este sistema.

variadas y ordenadas alfabéticamente, el usuario podría focalizar la búsqueda dependiendo de las letras con las cuales quisiera que comenzara el término. De este modo, no tendría que desplegar el listado y moverlo hasta encontrar las palabras que empezaran, por ejemplo, con la letra "a".

OTRAS VARIANTES

También es posible utilizar solo un campo de texto que la sugerencia complete según la palabra que se esté escribiendo. De este modo, si quisiéramos escribir "interactivo", en el campo pondríamos "interac" y, automáticamente, el sistema lo completaría con la primera palabra que lograra relacionar a los caracteres ingresados. En <http://activeden.net/item/predictive-text-field/89110> hay un buen ejemplo de este tema que venimos tratando. En este caso, podemos ingresar la letra "a", luego la "b" y, a medida que agreguemos caracteres, veremos cómo se va acotando el resultado de las sugerencias. Si queremos vaciar el campo, basta con presionar en la "X". Yahoo.com, como parte del Astra Flash Components, nos ofrece una solución muy práctica a la hora de utilizar este sistema. En la URL es posible ver la implementación de este componente de Flash, pero a primera vista, nos daremos cuenta de que es bastante simple agregar contenidos para autocompletar:

```
var airports:Array
= [{label:"SFO"},
{label:"XNA"},
{label:"SJC"}];
```

Si bien no vamos a explicar en detalle cómo armar este tipo de sistemas, sería



+ CONOCIMIENTOS TÉCNICOS

Recordemos que, para mejorar y potenciar nuestro aprendizaje, desde www.redusers.com/cursos/webdesign podemos descargar los archivos editables de los proyectos realizados a lo largo de esta colección. A partir de ellos, podremos obtener ideas para un emprendimiento propio o, directamente, modificar los archivos para reutilizarlos. En el sitio también encontraremos **guías visuales, glosario de términos** y una completa guía de preguntas frecuentes sobre la suite CS5. Además, el libro **Desarrollo y animación en ActionScript 3.0** nos ayudará a comprender los fundamentos de la programación en Flash.

aconsejable profundizar en esta información y animarse a realizar uno. Como datos significativos, debemos tener presente que los textos se generan de alguna forma física; es decir, tienen que estar alojados dentro de una base de datos, en archivos XML en el mismo Flash. Sea cual fuera el método que elijamos, es

conveniente saber que la información o los resultados de cada búsqueda serán más rápidos si el texto se encuentra cargado dentro del SWF. Es por eso que, si el diccionario de palabras está en una base de datos, lo mejor será cargar todos los términos y guardarlos en arrays o arreglos, ordenando alfabéticamente la primera letra de cada uno. De este modo, el resultado del autocompletado será instantáneo. También es importante comprender que la búsqueda se hará carácter por carácter, y por eso es necesario conocer el total de caracteres que llevamos escribiendo. Lo que seguramente precisaremos es un evento que esté escuchando todo el tiempo a nuestro campo de texto y que, cada vez que haya un cambio en él, automáticamente revise el array de palabras: si existe alguna coincidencia, y a medida que esto vaya ocurriendo, se irán descartando opciones que no sirvan. Este es el camino que tendríamos que seguir, pero realmente sería interesante investigar más sobre estos métodos; casi sin darnos cuenta, terminaremos aprendiendo en paralelo algunos métodos para manipular strings. ◀

ENVÍO DE

FORMULARIOS

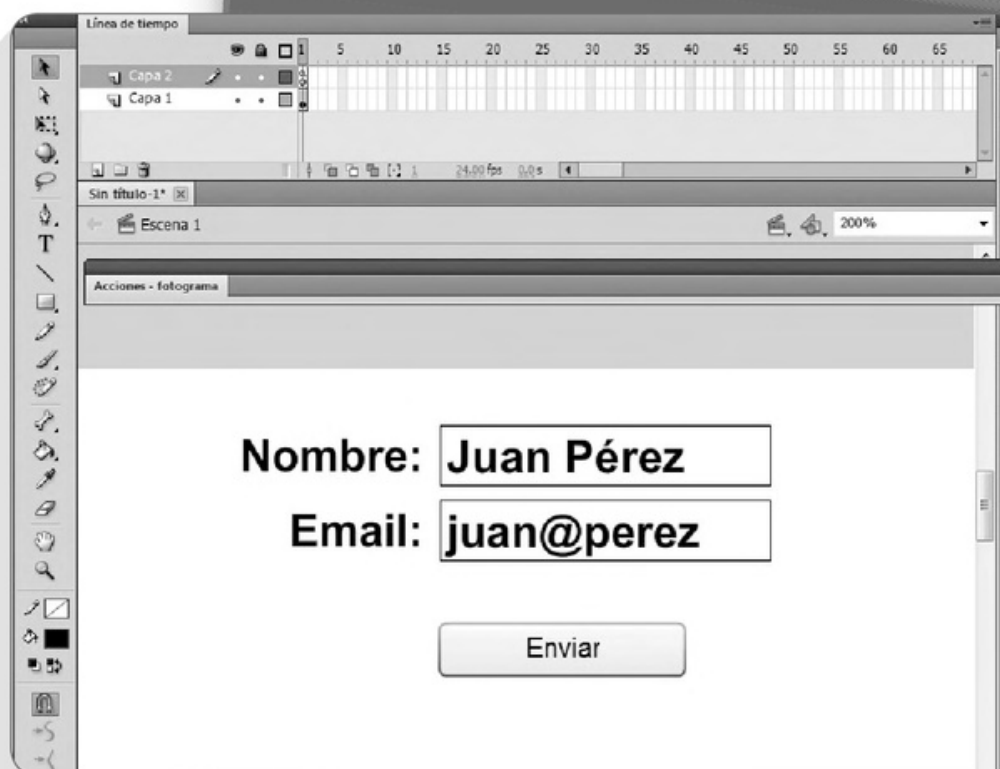
Flash puede conectarse con cualquier tipo de back-end para enviar información por e-mail. En este apartado, conoceremos cuáles son las claves que debemos seguir para mandar formularios sin problemas.

Cuando planificamos cualquier tipo de formulario para nuestra web en Flash, el problema que surge de inmediato es cómo enviar esa información que el usuario está completando. La opción habitual es hacerlo por e-mail a una dirección del administrador del sitio o del gestor de la empresa. Pero ¿de qué forma realizamos este proceso?

Al igual que para la carga dinámica de datos, la inserción en base de datos o tantas otras operaciones comunes en los sitios modernos, el envío de e-mails con información de los formularios es fundamental para la operativa. Flash, por sí mismo, no realiza este paso, sino que delega la tarea en un **lenguaje de servidor**, como PHP, .NET o Java, con los cuales se comunica a través de la invocación de rutinas en el servidor.

CLIENTE-SERVIDOR

Muchas veces nos hemos preguntado por qué ActionScript no puede realizar por sí mismo tareas como el envío de e-mails y el registro en bases de datos. Esto obedece a una lógica y a una estrategia de la firma Adobe: ActionScript es un lenguaje que se ejecuta en el **cliente** y, por lo tanto, no realiza tareas que deben correrse en el **servidor**. Esta división garantiza, por ejemplo, que nuestra película en Flash se localizará en la interfaz de usuario, y las tareas de servidor quedarán para un lenguaje específico. Incluso, las invocaciones de rutinas que podemos hacer en el servidor están limitadas por el sandbox de Flash Player, que impide llamar a aquellas que están en dominios diferentes del de nuestra película, a menos que un documento específico, llamado **crossdomain.xml**, autorice su uso desde el servidor invocado.



INVOCAR SCRIPTS EN EL SERVIDOR

Para enviar e-mails de la forma más simple, nuestra película Flash utiliza un **URLLoader** que se ocupa de invocar el script adecuado. Esta no es la única forma de hacerlo, pero por su simplicidad y robustez, es, sin duda, la más utilizada. El funcionamiento es como el de cualquier otro **RPC (Remote Procedure Call, o llamada de procedimiento remoto)**, en el cual se invoca un script y se define una función que será llamada cuando la acción se ejecute (se utiliza el término inglés **callback** para definir esta llamada). La llamada puede contener parámetros o no, al igual que la respuesta. Dado que estamos intentando pasar información por medio de un e-mail, nuestra llamada

contendrá, seguramente, uno o varios parámetros, y la respuesta, al menos uno: si el envío fue exitoso o no.

Para ilustrar el proceso, tomemos algunos datos muy simples del nombre y el correo de la persona que completa el formulario de contacto. También vamos a demostrar cómo enviarlo por e-mail. Para crearlo, será suficiente con un par de input text con nombres de instancia **email_txt** y **nombre_txt**, además de un componente button que traeremos del panel de componentes para realizar el envío.

GESTIÓN DEL ENVÍO

A diferencia de su predecesor, AS2, en AS3 hay tres clases involucradas en el

```

enviar_btn.addEventListener(MouseEvent.CLICK, enviar)
//enviamos la información por mail
function enviar(evt:MouseEvent):void{
    var myData:URLRequest = new URLRequest("enviaMail.php");
    myData.method = URLRequestMethod.POST;
    var variables:URLVariables = new URLVariables();
    variables.nombre = nombre_txt.text;
    variables.email = email_txt.text;
    myData.data = variables;
    var loader:URLLoader = new URLLoader();
    loader.dataFormat = URLLoaderDataFormat.VARIABLES;
    loader.addEventListener(Event.COMPLETE, dataOnLoad);
    loader.load(myData);
}
//recibimos la respuesta acerca del envío
function dataOnLoad(evt:Event) { //ok o error según se haya envi
    if (evt.target.data.resultado == "OK"){
        gotoAndStop("enviado");
    } else {
        gotoAndStop("error");
    }
}
}

```

Este es el código que vamos a usar para enviar el formulario. Una función manda el e-mail, y otra recibe la respuesta.

Vemos un ejemplo de cómo queda el layout del proceso de llamada a un procedimiento remoto.

envío de información: URLVariables, URLRequest y URLLoader, para encapsular la información que se envía, el pedido y, finalmente, el objeto que realiza la acción, respectivamente. Si bien esto es más largo de escribir, tiene ventajas en cuanto a fiabilidad. Veamos ahora el archivo PHP que gestionará el envío. No hemos incluido la validación para simplificar el código, pero antes de mandarlo, habría que verificar, al menos, que el mail y el nombre no estuvieran vacíos, tal como se comprueba en el **Código 1**. Este script toma los valores pasados por POST desde Flash, los manda por e-mail e imprime el resultado de la operación para que Flash pueda leerlo. Por simplici-

+ FRONT-END Y EL BACK-END

En diseño de software, el front-end es la parte del programa que interactúa con los usuarios, mientras que el back-end es la que procesa la entrada desde el front-end. La separación del sistema entre estos dos conceptos es un tipo de abstracción que ayuda a mantener las diferentes partes aisladas. La idea es que el front-end sea el responsable de recolectar los datos de entrada del usuario y de procesarlos conforme a la especificación que el back-end pueda utilizar. Es importante destacar que la conexión del front-end y el back-end es un tipo de interfaz.

FLASH NO REALIZA EL ENVÍO DE FORMULARIOS, SINO QUE DELEGA ESA TAREA EN UN LENGUAJE DE SERVIDOR, COMO PHP, .NET O JAVA.

dad, omitimos los headers del comando mail, que puede ser útil si queremos agregar un campo from o mandar el texto como HTML en vez de uno plano.

SINTAXIS AS3

Cuando escribimos código AS3, muchas veces nos preguntamos por qué es tan extenso, considerando que en AS2 la sintaxis era más breve. Hay varias razones

para esto, pero veremos las tres más importantes:

- Tipos seguros (type safe) permite que los errores de tipo de datos sean interceptados en el momento de la compilación.
- El uso de funciones declaradas, por sobre las declaraciones inline o de funciones anónimas, le da legibilidad al código.
- El uso de modificadores de clase y método, además de la corrección del problema del alcance de los callbacks dentro de clase, hace de AS3 un lenguaje mucho más completo en cuanto a los requerimientos POO (programación orientada a objetos). ◀

CÓDIGO 1

```
<?php
```

```

$nombre = $_POST['nombre'];
$email = $_POST['email'];
$mensaje = "Info de contacto\n\nNombre: $nombre\nEmail: $email";
if(mail("info@misitio.com", "Info desde mi sitio", $mensaje)){
    echo "resultado=ok";
} else {
    echo "resultado=error";
}

```

```
?>
```


CONTENT MANAGEMENT SYSTEM

También conocidos por su sigla CMS, estos administradores permiten que los sitios web sean manejados en forma simple e inteligente. Veamos de qué se trata y cómo utilizarlos.

Posiblemente, muchas veces hayamos oído hablar de los CMS (*Content Management System*, o sistemas de manejo de contenido), aunque podríamos decir que han cobrado relevancia recién en los últimos años. Si nos preguntamos a qué obedece la existencia de los CMS, en principio, podemos decir que, a medida que los sitios se fueron haciendo más dinámicos en cuanto a la manera de mostrar su contenido, cada vez resultaba más trabajoso mantenerlos actualizados. Pensemos que, en su origen, los sitios en Internet eran poco más que archivos de texto (HTML) que eran reemplazados con cierta periodicidad. En cambio, hoy en

día, cualquier sitio dinámico utiliza algún lenguaje de generación dinámica de contenido (PHP, .NET, Java) y alguna base de datos de donde toma el material que va a mostrar. El manejo de ese flujo de información no es una tarea trivial, y de allí surge la necesidad de utilizar gestores de contenido.

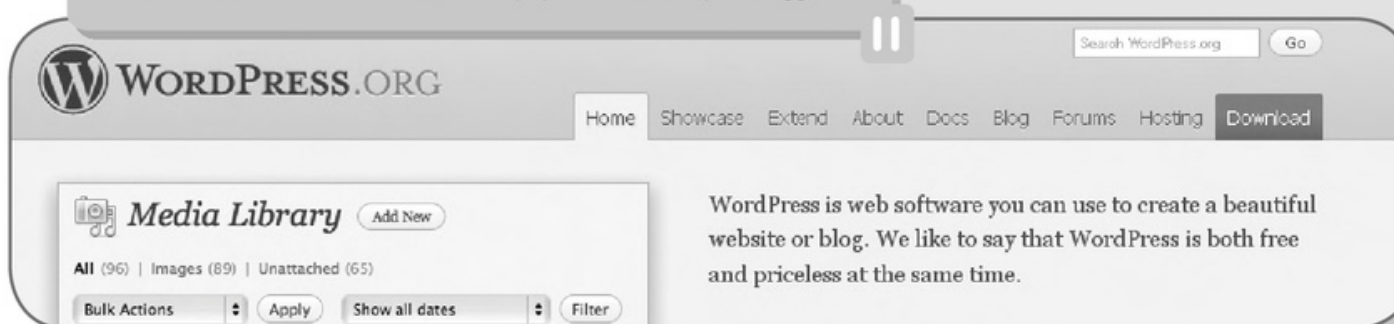
TIPOS DE GESTORES

Los gestores de contenido constan de un administrador que, a través de formularios, permite organizar el material publicado, editarlo y borrarlo. También cuentan con un sistema de usuarios que ofrece distintos derechos a quienes lo utilizan. Hay gestores de contenidos

sumamente complejos y costosos, pero existen muchos que son libres y Open Source. Veremos, a continuación, un listado de los más conocidos:

- **Foros:** sitios que permiten realizar discusiones en línea, en los cuales los usuarios pueden reunirse y discutir temas sobre los que están interesados.
- **Blogs:** publicación de noticias o artículos en orden cronológico, con espacio para comentarios y discusión.
- **Wikis:** sitios web donde todos los usuarios pueden colaborar con artículos, aportando información o rescribiéndola. También ofrecen un espacio para discusiones.

Observamos el home de WordPress, el más popular de los CMS para bloggers.



¿TE RESULTA ÚTIL?

Lo que estás leyendo es el fruto del trabajo de cientos de personas que ponen todo de sí para lograr un mejor producto.

Utilizar versiones "pirata" desalienta la inversión y da lugar a publicaciones de menor calidad.

NO ATENTES CONTRA LA LECTURA. NO ATENTES CONTRA TI. COMPRA SÓLO PRODUCTOS ORIGINALES.

Nuestras publicaciones se comercializan en kioscos o puestos de voceadores; librerías; locales cerrados; supermercados e internet (usershop.redusers.com). Si tienes alguna duda, comentario o quieres saber más, puedes contactarnos por medio de usershop@redusers.com

- **eCommerce:** sitios destinados al comercio electrónico.
- **E-learning:** sitios orientados a brindar información online.
- **Sitios web:** espacios virtuales con contenido y funcionalidades diversas, que sirven como fuente de información o como soporte a una comunidad.
- **Galería:** permite administrar y generar automáticamente un portal que muestra contenido audiovisual, en general, imágenes.

MECANISMO BÁSICO DE LOS GESTORES

Todos los gestores, más allá del tipo de sitio al que estén orientados, se basan en operaciones **ABM** (Altas-Bajas-Modificaciones) de una base de datos. Estas operaciones son transparentes para el usuario; es decir, no es necesario saber sobre programación ni bases de datos para realizarlas. Para escribir y mostrar texto formateado en un navegador, deberíamos usar HTML. Sin embargo, los CMS tienen editores visuales (la mayoría, basada en JavaScript) que permiten generar HTML a partir de controles visuales. Por ejemplo, si seleccionamos texto y presionamos el botón de **[Negrita]**, el editor agregará el tag `<b></b>`, que es el modo en que HTML reconoce el formato **bold**. La organización jerárquica del material es el aspecto en el que más difieren los CMS, y ese es el lugar en el cual se transparenta su objetivo. Por ejemplo, un CMS

LOS CMS CONSTAN DE UN ADMINISTRADOR QUE, A TRAVÉS DE FORMULARIOS, PERMITE ORGANIZAR EL MATERIAL PUBLICADO.

para blogs se organizará sobre la base de categorías, subcategorías y artículos, mientras que uno orientado a **manejar un foro** se basará en las entradas de los usuarios en forma de hilos, donde alguien formula una pregunta (u opinión) y otros responden. En un CMS destinado al **e-commerce**, la "estrella" serán los productos, mientras que en otro **educativo**, la clave serán las clases divididas en temas y un repositorio de archivos para descargar. Vale destacar que no es obligatorio usar gestores de contenidos, pero estos sistemas simplifican en gran medida el desarrollo web, y permiten realizar una gestión ágil y eficiente.

FLASH Y CMS

Hemos nombrado una gran cantidad de CMS, pero ninguno de ellos utiliza Flash como base, sino algún lenguaje de servidor para producir una salida

HTML. De hecho, casi todos los que son de código abierto y gratuitos se basan en **PHP + MySQL**.

Hay diversos motivos por lo cuales los CMS basados en Flash no han surgido con la misma fuerza que aquellos basados en salida HTML, pero existen unos cuantos como **Flashblocks** (del que hablaremos en breve) o **Motocms**. Otra posibilidad, no menos importante, es consumir la salida de cualquier CMS para mostrar la información en Flash. Por ejemplo, podemos mostrar una lista de artículos basándonos en los feeds (especie de resumen publicado automáticamente) de WordPress. Consumir directamente el artículo puede traer problemas, ya que no todos los tags HTML utilizados para el formateo son soportados por Flash, además de que muchos CMS emplean tags propios que no son compatibles con los estándares. Sin embargo, muchos CMS permiten salidas para Flash, sobre todo en XML e, incluso, en formato binario (AMF, **ActionScript Message Format**). Podríamos nombrar como ejemplo el módulo de Remoting para Drupal, que le permite a Flash consumir directamente en formato binario cualquier contenido ingresado en este popular sistema de gestión. Otro de los aspectos interesantes para Flash en relación a los CMS es la gran cantidad de plugins basados en él de que se dispone, sobre todo, para banners, noticias rotativas, videos y menús. ◀



Home | About Joomla! | Community | Forum | Extensions | Resources | Documentation | Developer | Shop



Try Joomla!

Try the online demo and learn the basics.



Learn and Extend Joomla!

Find help, extensions, and other users.



Download Joomla!

Open source content management system.

Google Custom Search

Search

Joomla! Announcements



Monday, 01 November 2010
Joomla 1.6 Beta 13 Now Available
The Joomla! Project is proud to announce the immediate availability of Joomla 1.6 beta 13.

joomla/create-and-share.html

Community News

Get Involved!
Here are a few places to get started.

The forums are where to meet other



DOWNLOAD JOOMLA!
GET THE LATEST VERSION

En www.joomla.org encontramos un CMS genérico para sitios web.

GESTORES DE CONTENIDO

Los CMS son muy útiles para la gestión de contenidos. Podremos utilizarlos en cualquier sitio de carga dinámica, ya sea Flash o HTML.

Content Management System (CMS)

Un Sistema de Gestión de Contenidos es una aplicación que se utiliza para facilitar el manejo y la administración de sitios web, ya sea en Internet o en una intranet. A través de él, podremos cambiar el contenido de un sitio, tanto sus textos e imágenes, como links y otra información que necesite actualizarse.

En cuanto a los sitios dinámicos en Flash, lo más habitual es encontrarnos con Sistemas de Gestión de Contenidos especialmente desarrollados para cada uno, debido a que cada sitio en Flash difiere de otro en cuanto a estructura y organización de la información. Algunos CMS comerciales son:

- FLASHBLOCKS
- Flash CMS 2
- MotoCMS
- XML2CSM

Login

Generalmente, los CMS están alojados en el mismo servidor del sitio, y el usuario administrador puede ingresar a este por medio de un login.

Carga de contenidos

Por medio del CMS subimos imágenes, texto y demás contenido para el sitio.





Website / ActionScript 3.0 - www.andymurray.com

Actualización del sitio

Un sitio puede buscar información para cargar su contenido de manera dinámica, la cual puede estar alojada en una base de datos o en archivos XML.

Actualización de contenidos

El CMS actualiza la información en la base de datos o en los archivos XML respectivos al sitio. Están generalmente desarrollados en PHP.

PHP

```
if($numRows == 0) {
    return 0;
} else {
    $row =
    mysql_fetch_object($
    result);
    return
    $row->quantity;
}
```



FLASHBLOCKS

Comenzó como un esfuerzo individual para facilitar el desarrollo web, y terminó por convertirse en una activa empresa que comercializa un completo sistema montado en Flash.

La primera versión de Flashblocks fue escrita en AS1 hace unos 10 años y, luego, fue evolucionando hacia AS2, siendo AS3 la última versión disponible. Cualquiera de ellas puede descargarse desde **www.flashblocks.com**.

Flashblocks es un sistema **drag & drop** de contenido que puede editarse directamente en Flash. La información que se modifica se guarda en archivos XML que irán alimentando la instancia dinámica de un sitio. Para que funcione correctamente, necesitamos el soporte de PHP en nuestro servidor, ya que es el lenguaje de servidor que complementa el sistema. Luego de bajar e instalar Flashblocks a través del **Administrador de extensiones** de Adobe, tendremos una carpeta de componentes de Flashblocks en nuestra librería. Para comenzar a trabajar en una película nueva, debemos arrastrar el componente Flashblocks al escenario y, allí, completar los campos que expone el inspector de propiedades:

- **licence:** aquí ingresamos el ID que recibimos por mail cuando descargamos Flashblocks.
- **dataFile:** es la ruta hacia el archivo de datos que será cargado en la instancia del componente. Se cargará, por defecto, el archivo `flashblocks/data/dataFile.xml`.
- **dataFileUserLevel:** es el mínimo nivel de usuario autenticado requerido para acceder a la edición de datos. Si este usuario tiene un nivel 3, y el `dataFileUserLevel` requerido es 7, entonces el usuario no podrá editar ninguna información del archivo de datos del sitio.
- **configFile:** es la ruta hacia el archivo de configuración de Flashblocks. Contiene la

ubicación de todos los archivos necesarios para funcionar, incluyendo el archivo con datos predefinidos que se usará en la primera carga. La ruta por defecto es `flashblocks/flashblocks_config.xml`.

► **flashblocksID:** cada instancia del componente Flashblocks debe tener un único FlashblocksID. Puede ser cualquier identificador que el usuario determine; por

ejemplo, `archivoNoticias`, `archivoDatos1`, etc. Por defecto, el nombre es `fb0`.

► **autoLoad:** con un valor **true** por defecto, Flashblocks cargará el archivo de datos específicos no bien sea instanciado en el escenario. Si este valor está en **false**, será necesario

Desde <http://flashblocks.com> podemos descargar el programa y acceder a tutoriales muy interesantes.

cargar manualmente la información usando el método `loadData()`, `flashblocksInstance.loadData()`;

CREACIÓN DE ELEMENTOS USANDO FLASHBLOCKS

Si hemos seguido los pasos anteriores, ya tenemos una instancia del componente Flashblocks en el escenario con los datos básicos completos. Para construir un ejemplo simple, vamos a arrastrar un componente Button del panel de componentes de Flash al escenario; luego, desplazamos un componente **Editblocks** desde la carpeta **flashblocks** en el panel de componentes, arriba del botón. Veremos que en el ángulo superior izquierdo del botón aparece el icono del editor. Repetimos el proceso con un campo de texto; es decir, dibujamos un campo de texto dinámico, como lo hacemos siempre, y arrastramos el componente **Editblocks** a él. Veremos el icono otra vez. Este es el principio; cualquier componente que tengamos en escena puede ser editable si le agregamos el componente Editblocks.



¿ES ADOBE FLEX/FLASHBUILDER UN CMS?

Para quienes están acostumbrados a usar Flex, posiblemente la pregunta les parezca extraña, pero los que no conocen Adobe Flashbuilder pueden pensar que es realmente un editor de contenido o que está emparentado de alguna manera con un CMS. Sin embargo, Flex es un framework que permite el desarrollo rápido de RIA (**Rich Internet Applications**), es decir, aplicaciones web con Flash como formato de salida. Para este fin, ofrece una amplia gama de componentes que permiten desarrollar interfaces en muy poco tiempo, aprovechando también todas las características de animación y diseño que destacan a Flash, pero enfocado en el desarrollo rápido.

SUBIR AL SERVIDOR Y EDITAR

Una vez que hemos armado y publicado nuestra película, debemos subir todo al servidor para que realmente podamos editar el contenido. Vamos a subir a un directorio la película Flash, el archivo `include.php` y la carpeta **Flashblocks**. Luego, daremos permiso de escritura a la carpeta `flashblocks/data`. Luego de realizar este paso, estaremos en condiciones de editar contenido directa-

mente desde el navegador. Para hacerlo, cargamos en él el SWF que subimos antes, hacemos clic en la película, presionamos **<Ctrl + Shift>** o **<Command + Shift>** y seleccionamos **[Login]** desde el menú contextual. A continuación, pulsamos **<Enter>** dejando el campo de usuario y el password en blanco (a menos que ya hayamos entrado datos en el archivo `flashblocks_login.php`). Nos encontraremos con el botón y el campo de texto; al mover el mouse sobre ellos, notaremos que el campo de texto está vacío. Ahora, seleccionamos **Label Text** y, luego, presionamos **<Ctrl + Shift>** o **<Command + Shift>**. Aparecerá un editor de texto simple, en donde entramos el texto "Ver ahora" y presionamos **[Insert Text]**. El editor desaparecerá y la etiqueta del botón se actualizará. Del mismo modo que explicamos en el párrafo anterior, actualizamos el contenido del campo de texto. Para guardar el nuevo contenido, presionamos de forma simultánea **<Ctrl + Shift>** y seleccionamos **[Save]**, del menú contextual. Este es el principio sobre el que se basa Flashblocks: componentes editables que vamos modelando según nuestra necesidad. Como vemos, la operativa es muy distinta de los CMS habituales, en donde nos basamos en formularios para ingresar la información, pero en este caso, podemos aprovechar la riqueza de un entorno rico como ofrece Flash. ◀

El sitio <http://soulconstruction.com> es uno de los que mejor utilizan Flashblocks.

SOUL CONSTRUCTION INC. HOME ABOUT TESTIMONIALS PROJECTS NEWS CONTACT

+ GIVING BACK
Giving back to the community

+ FEATURED PROJECT
The Parker House

+ BIOGRAPHIES
Soul Construction Team Biographies

+ COMMUNITY ASSOCIATIONS
Associations that we belong to.

Welcome to Soul Construction.com

Welcome to Soul Construction.com where you can find out all about what Pacific Business News recognized as one of the 50 fastest growing companies in Hawaii. We are committed to raising the bar in commercial construction by combining outstanding, top-tier craftsmanship with a level of organization and integrity rarely found in the industry. Please take a look around! I think you'll like what you see.

Aloha,
Josh Copeland
Owner

ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué son los formularios?

02 ¿Para qué sirve un formulario?

03 ¿Qué herramientas necesitamos para crear un formulario?

04 Mencione la importancia de la tabulación de formularios.

05 ¿Qué es el texto predictivo?

06 ¿Para qué necesitamos enviar un formulario?

07 ¿Qué es Content Management System?

08 ¿Cuál es la utilidad de los gestores de contenidos?

09 Mencione algunos de los CMS existentes.

10 ¿Qué es FlashBlocks?

PRÁCTICAS

01 Cree un formulario en Flash.

02 Valide un formulario.

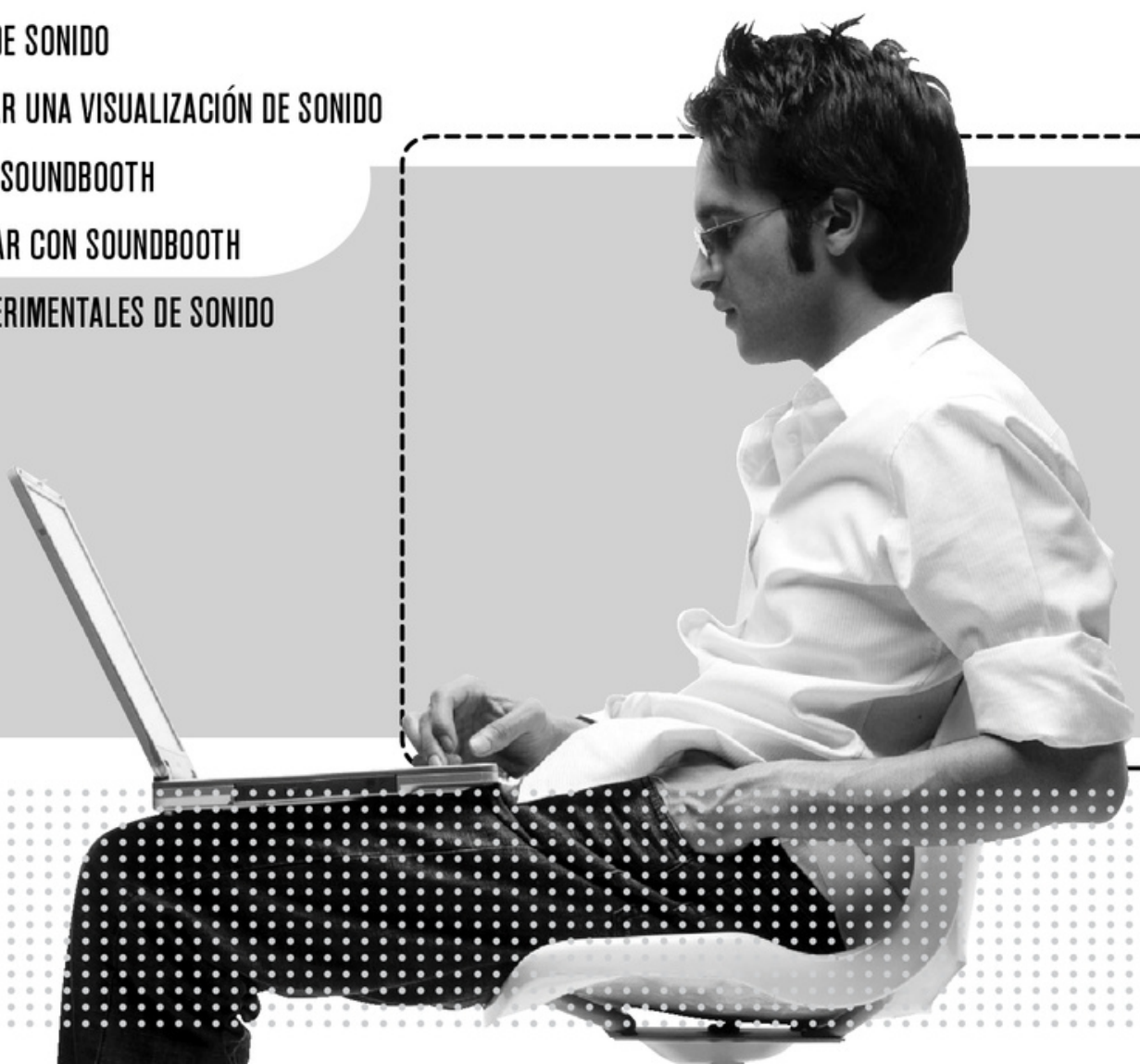
03 Envíe un formulario.

04 Implemente un texto predictivo.

05 Experimente con un CMS.

Sonido

- ▶ USO DE SONIDO
- ▶ SINCRONIZACIÓN DE SONIDO
- ▶ SONIDO Y ACTIONSSCRIPT 3
- ▶ LIBRERÍAS DE SONIDO
- ▶ CÓMO CREAR UNA VISUALIZACIÓN DE SONIDO
- ▶ ADOBE CS5 SOUNDBOOTH
- ▶ CÓMO EDITAR CON SOUNDBOOTH
- ▶ SITIOS EXPERIMENTALES DE SONIDO



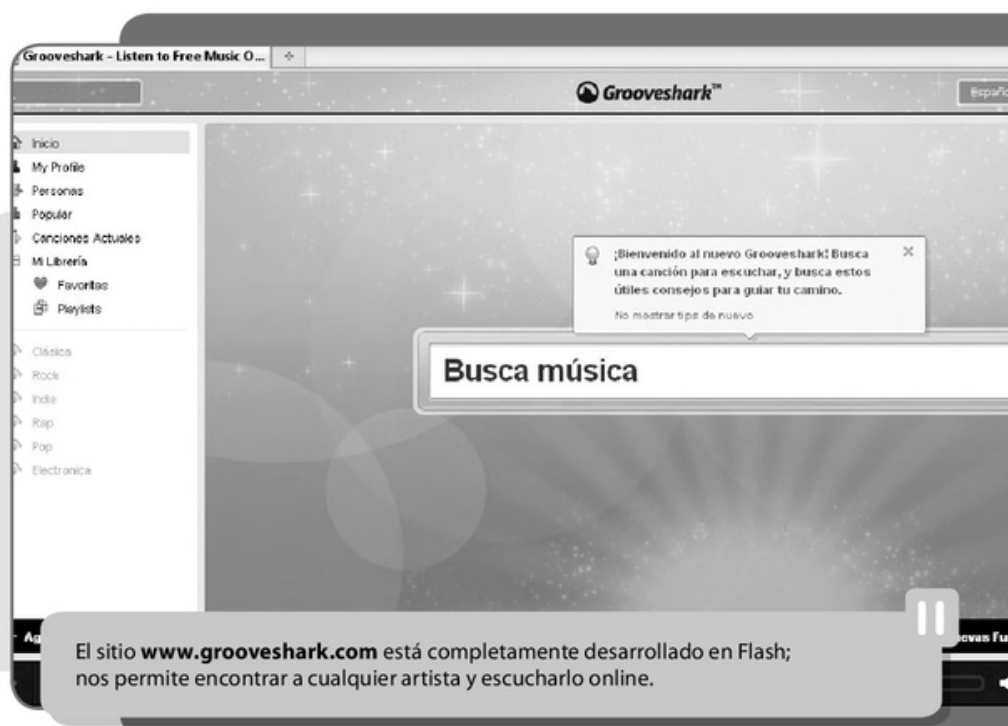
USO DE SONIDO

En este apartado abordaremos los conceptos de implementación de audio en los desarrollos web y veremos algunos ejemplos sobre cómo utilizar el sonido teniendo en cuenta los objetivos de nuestro trabajo.

El sonido es un elemento esencial en el esquema de comunicación de los desarrollos web.

Básicamente, podemos utilizarlo de diferentes maneras, según los objetivos que tenga nuestro proyecto. Por un lado, podemos recurrir a sonidos como canciones o bandas sonoras que acompañen al usuario durante su estadía en nuestro sitio, mientras que, por otro lado, podemos emplearlos como efectos para notificar al usuario que está realizando determinada acción, como detenerse sobre un botón o efectuar un clic en cierta sección. Por último, Flash nos permite utilizar los archivos de sonido de manera que podamos sincronizarlos con nuestras animaciones, para así darles mayor profesionalismo. Las posibilidades son amplias, y todo dependerá de cuáles sean los objetivos comunicacionales de nuestro trabajo y, principalmente, de el público al que esté dirigido. Es importante tener en cuenta estas variables, ya que la decisión de incorporar sonidos dentro de un desarrollo web puede tener algunas consecuencias para el usuario que visite el sitio. Dependiendo de la manera en que incorporemos los archivos de audio, estos pueden incrementar el tamaño de las películas SWF, con lo cual el tiempo de descarga para el usuario será mucho

FLASH NOS PERMITE UTILIZAR LOS ARCHIVOS DE SONIDO DE MANERA QUE PODAMOS SINCRONIZARLOS CON LAS ANIMACIONES.



mayor. Desde el punto de vista de la usabilidad, los sonidos pueden ser molestos para el navegante, si es que no le ofrecemos la posibilidad de desactivarlos. Por esta última razón, debemos evaluar si realmente es necesario incorporarlos en nuestros proyectos y cuál es la técnica para hacerlo que mejor se adecua a las necesidades de cada emprendimiento.

EMPLEO DE AUDIO EN FLASH CS5

Flash es una de las herramientas líderes en el área de desarrollo web para el uso de audio. Con la versión CS5, podemos importar los tipos de archivo más comunes, como WAV, MP3 y AIFF, y también, proyectos de Adobe Soundbooth (ASND). Esto nos permite incorporar en nuestros desarrollos archivos de sonido que, mediante la aplicación de las herramientas de la IDE de Flash CS5 y ActionScript 3.0, podremos manipular, en

lo que se refiere tanto a la reproducción como a la visualización gráfica.

Al igual que sucede con las imágenes y los videos, los archivos de audio pueden ser tratados de manera interna o externa, y cada método tiene sus propias ventajas y desventajas, las cuales explicaremos a continuación.

SONIDOS INTERNOS

Como mencionamos anteriormente, Flash nos permite utilizar sonidos internos y externos. Los primeros, como su nombre lo indica, son aquellos que se ubicarán dentro de un archivo FLA. Esta modalidad puede resultar útil en caso de que necesitemos solo un archivo de sonido o algunos pocos, preferentemente, de reducido tamaño. Existen dos ventajas al momento de tener los archivos de audio dentro del FLA. La primera es que nos brinda automática disponibilidad del



www.thedigitalinvaders es el sitio de una escuela de diseño mexicana, en donde se demuestra el excelente uso de las técnicas de manipulación de sonido en Flash.

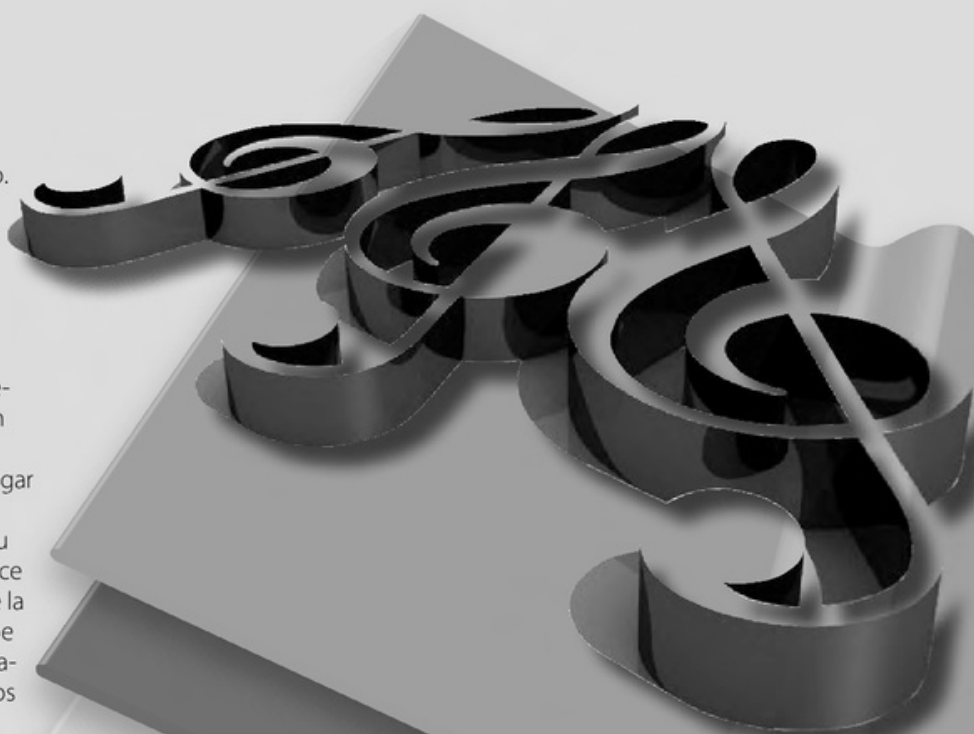
sonidos, tendremos que editar el archivo FLA y, por ende, habrá que exportar una nueva película. En cuanto al segundo punto, como bien sabemos, cada archivo que introduzcamos dentro del FLA –ya sea imagen, sonido o video– aumentará el tamaño de la película. Por lo tanto, todo el contenido tardará más tiempo en cargarse, que si utilizáramos los sonidos de manera externa.

SONIDOS EXTERNOS

Flash nos permite utilizar sonidos que estén fuera de los archivos FLA. En la mayoría de los casos, es recomendable recurrir a este método, exceptuando aquellos en que necesitemos realizar una

sincronización entre un sonido y una animación en la línea de tiempo. Las ventajas que implica cargar los sonidos externos son las mismas desventajas que tiene usar los internos. Es decir que, al momento de realizar cambios en los archivos de sonido, no tendremos que abrir los FLA ni volver a exportar la película. Por otro lado, podemos destacar como principal ventaja el menor peso de los archivos resultantes. Esta técnica suele ser la más utilizada para el desarrollo de reproductores de audio o, simplemente, para la manipulación de sonidos pesados que puedan incrementar considerablemente el tamaño de las piezas finales. ◀

material una vez que la película se ha cargado por completo; esto significa que podremos reproducirlos o utilizarlos sin tener que esperar a que se descargue el archivo de sonido. La segunda ventaja tiene que ver con las posibilidades que brinda Flash al momento de sincronizar los archivos de sonido con nuestras animaciones en la línea de tiempo. Más adelante, veremos este último tema con más detalle para saber de qué modo podemos generar una mejor sincronización de las animaciones con los sonidos. Entre las desventajas que implica agregar los sonidos dentro del archivo FLA, podemos mencionar que se impide su fácil y rápida actualización, y se produce un aumento considerable del peso de la película. En cuanto al primer caso, cabe decir que, si por algún motivo necesitamos actualizar o realizar cambios en los



SINCRONIZACIÓN

DE SONIDO

En este apartado aprenderemos a utilizar los sonidos de manera interna y a optimizarlos, y veremos qué herramientas nos brinda Flash para sincronizarlos con nuestras animaciones en la línea de tiempo.

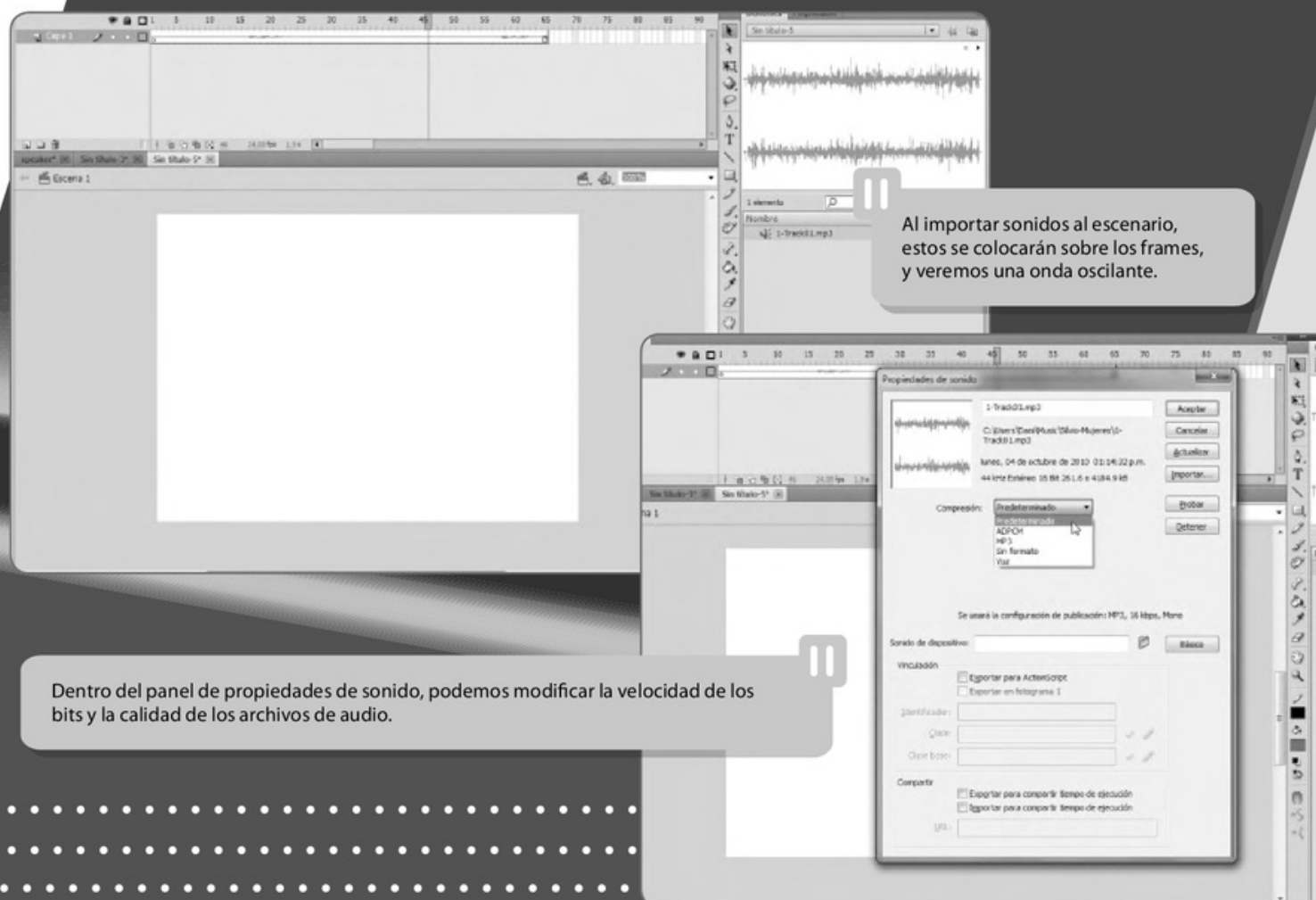
Como mencionamos anteriormente, para utilizar los sonidos de manera interna debemos importar los archivos de audio dentro de nuestro FLA. El proceso que involucra la importación de sonidos en Flash resulta muy simple y es exactamente igual al que se realiza al trabajar con imágenes o videos. Lo primero que debemos hacer es ir a **[Archivo]**, seleccionar del menú la

opción **[Importar]** y, luego, elegir entre **[Importar a escenario]** o **[Importar a biblioteca]**. Si nos decidimos por el primer caso, a diferencia de las imágenes que se posicionan en la escena, el audio se ubicará en el primer fotograma de la película. Si agregamos fotogramas en la línea de tiempo con **<F5>**, podremos visualizar la información del audio. Es importante destacar que si no colocamos un **stop()** al final de la línea, el sonido

será reproducido cada vez que el cabezal del timeline se ubique en el primer fotograma.

OPTIMIZAR SONIDOS

Flash nos da la posibilidad de optimizar los archivos de sonido de la misma manera que los hacemos con las imágenes. En este caso, debemos abrir la biblioteca con **<Ctrl + L>** y presionar el botón derecho del mouse sobre el archivo



de sonido que hayamos importado. Accederemos al panel de **[Propiedades]**, en donde podremos seleccionar entre diferentes opciones dentro de la solapa **[Compresión]**. Principalmente, debemos centrar nuestro análisis en las opciones **[Por defecto]** y en **[Compresión mp3]**. La primera deja el archivo sin ningún tipo de compresión, y la segunda es la que mayor compresión brinda. Es recomendable no comprimir archivos de sonido que tengan poco tamaño, porque pueden alterarse de manera negativa. Para los archivos de música o sonidos más pesados, se aconseja aplicar la compresión de MP3. Dentro del mismo panel, podremos modificar la velocidad de bits del sonido y la calidad. En cuanto a este último parámetro, lo mejor es dejarlo en **[Rápido]**, ya que este método produce el mayor balance entre calidad y tamaño.

SINCRONIZACIÓN DE SONIDOS CON BOTONES

Una de las aplicaciones de sonido más difundidas dentro del diseño web es para notificar al usuario que está realizando determinada acción; por ejemplo, avisarle que se está ubicando sobre un botón o que acaba de seleccionarlo. Realizar esta tarea desde la IDE de Flash es muy simple. Debemos entrar en la instancia del botón, crear un layer e insertar un **key frame** en la instancia del **Over**, donde colocaremos el sonido arrastrándolo desde la biblioteca hasta la escena. De esta manera, cada vez que



AMPLITUD DE LOS CANALES

La amplitud de los canales es la información a la que podemos acceder y manipular de manera más simple. Mediante las propiedades **leftPeak** y **rightPeak** de una vía de sonido es posible conocer la amplitud de cada canal estéreo que se esté reproduciendo. Estas propiedades nos darán datos entre 0 y 1, siendo 0 la no reproducción de sonido y 1 la amplitud máxima.

el usuario apoye el mouse sobre uno de los botones, recibirá la notificación correspondiente.

SINCRONIZAR SONIDOS CON ANIMACIONES

Muchas veces tenemos la necesidad de colocar efectos a nuestras animaciones a través del sonido. Con las herramientas de Flash y todas las técnicas que hemos visto a lo largo de este curso, resultará muy fácil sincronizar los sonidos. Para demostrarlo, detallaremos un ejemplo.

ANIMACIÓN EN LA LÍNEA DE TIEMPO

Debemos tener preparado en el archivo FLA un movie clip que posea una imagen

o un dibujo, como el de una pelota. Lo primero que debemos hacer para realizar la sincronización es armar una animación. Para los objetivos que planteamos en este ejemplo, vamos a situarnos en el layer donde tenemos el movie clip "Pelota" y a generar tres keyframes en los cuadros 1, 5 y 10, respectivamente. En el que está ubicado en el frame 5 situamos la pelota, como si estuviese realizando contacto con el suelo; así, la animación simulará una pelota rebotando.

IMPORTAR EL ARCHIVO DE AUDIO

Para sincronizar las animaciones con los sonidos, debemos utilizar los archivos de audio de manera interna. Por lo tanto, buscamos un archivo de audio que simule el sonido realizado por una pelota al picar en el suelo. Una vez que importamos el archivo, creamos un layer y arrastramos el sonido desde la librería hacia la escena.

SINCRONIZACIÓN DE AUDIO

En el panel **[Propiedades]** hay una opción llamada **[Sincronización]**, en donde debemos elegir **[Stream]**. De este modo, escucharemos el sonido a medida que movamos el cabezal de la línea de tiempo, para así oírlo mientras realizamos la animación.

FINALIZAR LA SINCRONIZACIÓN

Gracias a las propiedades **Stream**, podemos previsualizar el sonido y ajustar la posición de los frames del layer. Para concluir con la sincronización, vamos a acortar o extender los frames dentro del layer del sonido. Este proceso puede variar según el sonido que estemos utilizando y la manera en que esté realizada la animación.

AÑADIR EFECTOS

Flash trae consigo algunos efectos propios que podemos utilizar en las animaciones. Para aplicarlos, debemos dirigirnos al panel **[Propiedades]** y seleccionar uno del menú desplegable. ◀



SONIDO Y ACTIONSCRIPT 3

Al momento de manejar sonido con ActionScript, debemos saber qué es lo que podemos hacer y cuáles son las clases o métodos que debemos utilizar para alcanzar los objetivos propuestos.

ActionScript posee algunas clases que se utilizan para manipular los archivos de audio.

A continuación, realizaremos un breve análisis sobre las funcionalidades que nos brindan:

- **Sound:** se usa, principalmente, para descargar sonidos externos, reproducirlos y manejar sus propiedades básicas.
- **SoundChannel:** permite crear un canal separado para cada sonido nuevo que vayamos a reproducir. De esta manera podemos manipular cada sonido que se esté reproduciendo en forma individual.
- **SoundMixer:** permite utilizar un objeto que combine todos los canales de nuestra película, de modo que podamos manipular todos por completo. Con esta clase, por ejemplo, podremos dejar de reproducir todos los sonidos que se estén ejecutando en una aplica-

ción de manera simultánea y con una sola línea de código.

► **SoundLoaderContext:** en conjunto con el método `load()` de la clase `Sound`, permite especificar un buffer para que el sonido sea cargado antes de comenzar con su reproducción.

► **SoundTransform:** se emplea para controlar el volumen y el panning entre los canales izquierdo y derecho de un sonido estéreo.

Si bien para los objetivos de este curso esta información puede resultar un poco abrumadora, nos sirve para entender en detalle las funcionalidades con que cuenta ActionScript 3.0 para el control básico de sonido.

CARGAR SONIDOS UTILIZANDO ACTIONSCRIPT 3.0

Para realizar la carga de sonidos externos, debemos crear una instancia de la clase

URLRequest, a la cual le asignaremos la ruta donde se encuentra el archivo en cuestión:

```
var miRequest:URLRequest = new URLRequest();
miRequest.url = "mp3/cancion.mp3";
```

Luego, creamos un objeto de la clase **Sound** al que llamaremos "**miSonido**". Por medio del método `load()`, le indicamos el archivo por cargar:

```
var miSonido:Sound = new Sound();
miSonido.load(miRequest);
```

Antes de importar un pista a nuestro desarrollo podremos grabarla en un estudio profesional si es necesario.



En esta instancia, el archivo de sonido ya estará cargado dentro de la película. Sin embargo, si la previsualizamos, notaremos que no podremos escuchar el sonido debido a que todavía no le hemos dicho a la película que éste debe reproducirse.

REPRODUCIR Y MANIPULAR SONIDOS

Si bien la clase **Sound** posee el método **.play()** que nos permite reproducir el sonido cargado de manera simple, existe un paso previo para tener un mayor control sobre el audio agregado. Este paso involucra la creación de una instancia de la clase **SoundChannel**, donde almacenaremos la información del sonido para controlarlo de manera individual:

```
Var miCanal:SoundChannel;
miCanal = miSonido.play()
```

Asignándole la función **play()** al canal que acabamos de crear, comenzaremos con la reproducción del sonido. En este momento, si tratamos de previsualizar la película, notaremos que el sonido comienza a reproducirse de manera instantánea. El método **play** de la clase **Sound** posee tres parámetros que podemos definir al momento de invocarlos. El primero es el milisegundo en el que queremos que

comience su reproducción. El segundo sirve para determinar la cantidad de veces que queremos que se reproduzca ese sonido. El tercero es el volumen con el que deseamos que se escuche al momento de la reproducción.

APLICAR STOP

Utilizando el método **stop()** de la clase **SoundChannel**, podemos detener la reproducción del sonido almacenado en dicho canal. Si generáramos dos botones que dieran **stop** y **play** al sonido almacenado en el canal, notaríamos que cada vez que se asignara el método **stop()**, este llevaría el tiempo de reproducción a 0 segundos. Esto puede parecer un problema, pero gracias a que antes asignamos el sonido a una instancia de la clase **canal**, podemos comenzar a reproducir el sonido desde el momento en que lo paramos. Para hacerlo, almacenamos en una variable su posición antes de llamar al método **stop()**, de la siguiente manera:

```
Var
posicionAlPausar:Number =
miCanal.position;
miCanal.stop();
```

Para reproducir el sonido desde esa instancia, basta con pasar la variable como el primer parámetro del método **play()**:

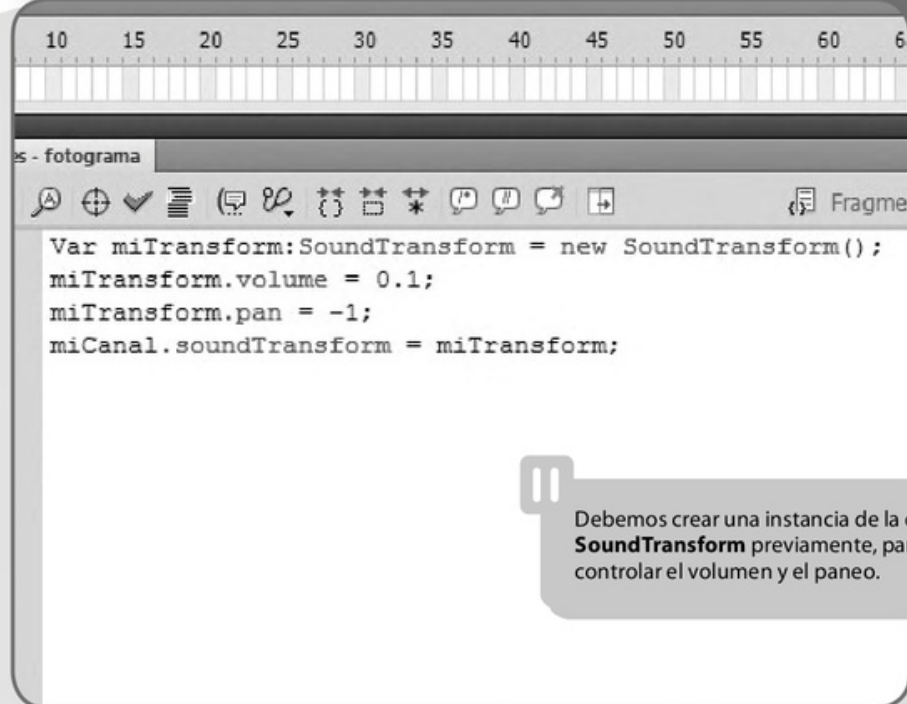
```
miCanal = miSonido.
play(posicionAlPausar);
```

Así concluimos con la manipulación del sonido en ActionScript 3.0. Ahora, sólo nos resta manejar el volumen de los canales, tanto por separado como de manera simultánea.

VOLUMEN Y PANEÓ DEL SONIDO

Modificar el volumen es muy simple una vez que tenemos el sonido almacenado dentro de un canal. Sin embargo, antes de manipularlo, debemos crear una instancia de la clase **SoundTransform**, que se encarga de controlar el volumen y el paneo de los sonidos: la propiedad de volumen de la clase **SoundTransform** permite pasarle como parámetro un número (entre 0 y 1) que se asignará al volumen. En cuanto al paneo, podemos pasarle parámetros entre -1 y 1, siendo -1 la asignación del sonido al altavoz izquierdo, y 1, la correspondiente al derecho. ◀

Debemos crear una instancia de la clase **SoundTransform** previamente, para controlar el volumen y el paneo.



+ CONOCIMIENTOS TÉCNICOS

Recordemos que, para mejorar y potenciar nuestro aprendizaje, desde **www.redusers.com/cursos/webdesign** podemos descargar los archivos editables de los proyectos realizados en esta colección. A partir de ellos, podremos obtener ideas para un emprendimiento propio o, directamente, modificar los archivos para reutilizarlos. En el sitio también encontraremos **guías visuales**, un **glosario de términos** y una completa guía de preguntas frecuentes sobre la suite CSS. Además, el libro **Desarrollo y animación en ActionScript 3.0** nos ayudará a comprender los fundamentos de la programación en Flash.

LIBRERÍAS DE SONIDO

El lenguaje de programación ActionScript 3.0 trajo una gran variedad de librerías y clases que nos permiten trabajar con el sonido, no solo desde su reproducción y manipulación, sino también de manera gráfica.

Si bien estuvimos realizando un análisis y nos sumergimos en el uso de las clases más relevantes que incluye ActionScript 3.0 para el control de sonido, existen algunas más que no mencionamos anteriormente, pero que son importantes para dar mayor profesionalismo a nuestros proyectos. Debido a los objetivos que propone este curso, en este apartado no abordaremos ejemplos prácticos referidos al uso de librerías o clases, ya sean propias de Flash o realizadas por grupos de desarrollo de código libre y gratuito (Open Source); sino que trataremos las extensas posibilidades que brinda este programa para la experimentación y la manipulación del sonido.

ETIQUETAS ID3 Y LA CLASE ID3INFO

Al codificar un archivo MP3, los encoders permiten incluir metadatos dentro de él. Estos contienen información sobre el archivo de audio, y suelen llamarse **etiquetas ID3**. Por medio de la clase ID3Info de ActionScript 3.0, podemos acceder a los metadatos almacenados dentro de los archivos MP3. Existen siete tipos de metadatos que podemos leer por medio de las propiedades; entre ellos, título del tema, autor, género, álbum al que pertenece y año en que fue creado, entre otros. Las etiquetas ID3 suelen utilizarse para realizar reproductores de archivos MP3 que requieran la posibilidad de brindar información sobre ellos al usuario.

CLASE MICROPHONE

Hasta el momento, nos referimos al uso de archivos de sonido, tanto internos como externos. ¿Pero cómo podemos capturar audio generado por los usuarios? Para este tipo de situaciones, Flash ofrece la **clase Microphone**, con la que podemos tomar audio desde un micrófono que se encuentre conectado a la computadora, a fin de reproducir la entrada de sonido por los altavoces o enviarlo a un servidor remoto para grabarlo.

REPRESENTACIONES VISUALES DE SONIDO EN FLASH

Si bien antes hicimos referencia a la clase **SoundMixer**, no hemos destacado una de las grandes funcionalidades que posee. Esta cuenta con un método llamado **computeSpectrum()**, a través de la cual podemos acceder a los datos de las formas de onda de sonido que está reproduciendo Flash. Dicha información nos permite crear representaciones visuales de los sonidos en ejecución, empleando su amplitud y frecuencia. Este tema es

muy importante debido a que existen varias librerías de código de tipo Open Source que nos permitirán generar nuestras propias visualizaciones de sonido experimentales de manera muy sencilla. Más adelante, a lo largo de este manual, veremos cómo crear nuestras propias visualizaciones de sonido que reflejen su amplitud.

HYPE FRAMEWORK

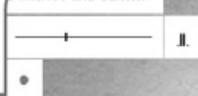
Desde www.hypeframework.org es posible descargar un framework de ActionScript 3.0 destinado a realizar animaciones experimentales de manera muy simple. Posee una funcionalidad muy interesante al momento de realizar experimentos con las visualizaciones de sonido. Branden Hall y Joshua Davis, sus creadores, realizaron un amplio estudio sobre las propiedades de amplitud y frecuencia de los sonidos. El resultado es una herramienta que, a través de la implementación de unas simples líneas de código, nos permite entrelazar propiedades de un movie clip o sprite a



FLASH MEDIA SOUNDMIXER

Un dato relevante que siempre debemos tener presente es que la clase **SoundMixer** controla la reproducción y las propiedades de seguridad inherentes a todos los sonidos de una aplicación determinada. De hecho, se mezclan varios canales de sonido mediante un objeto **SoundMixer** común, de manera que los valores de propiedad afectarán a todos los objetos **SoundChannel** que se reproducen actualmente.

Pintando una Canción



El sitio <http://soydeaire.labuat.com> es un excelente ejemplo de lo que podemos lograr utilizando el método `computeSpectrum` de la clase `SoundMixer`.

En <http://blog.soulwire.co.uk> encontraremos un excelente ejemplo de lo que puede lograrse con `Tonfall`.



una octava del sonido que esté siendo reproducido. Esta característica del framework nos da la posibilidad de crear animaciones de alto impacto que tengan en cuenta no solo el volumen del sonido, sino también los tonos graves y agudos. Si hablamos de audio en Flash, es imposible no mencionar a Andre Michelle (www.andre-michelle.com), quien probablemente sea el desarrollador en Flash más experimentado en lo que respecta a sonido. En octubre de 2010, Andre Michelle lanzó `Tonfall`, un framework de audio hecho en `ActionScript 3.0` para quienes desean introducirse en la programación y generación de sonidos en este programa. Esta herramienta permite que los programadores sin mucha experiencia creen sonidos sintéticos en Flash y realicen asombrosos experimentos con audio. Si bien el framework es muy novedoso, todavía no existe mucha documentación ni información sobre los alcances del proyecto, pero podemos descargarlos desde <http://blog.andre-michelle.com/2010/tonfall-open-source-audio-framework>.

FLASH NOS PERMITE UTILIZAR LA CLASE `MICROPHONE` PARA CAPTURAR AUDIO DESDE UN MICRÓFONO.

COMPUTESPECTRUM

Este método nos permite obtener los datos de sonido como un objeto `ByteArray` que almacene 512 datos, cada uno de ellos con un valor entre -1 y 1. Estos valores son los que representan la amplitud de los puntos de la onda de sonido de una reproducción. Los datos que manipulemos y el modo en que lo hagamos definirán qué tipo de gráfico generaremos al realizar nuestra visualización. Si deseamos crear una clásica visualización en forma de onda, deberíamos utilizar el método `computeSpectrum`. Este nos permite generar una representación gráfica de los puntos de la onda de un sonido. ◀



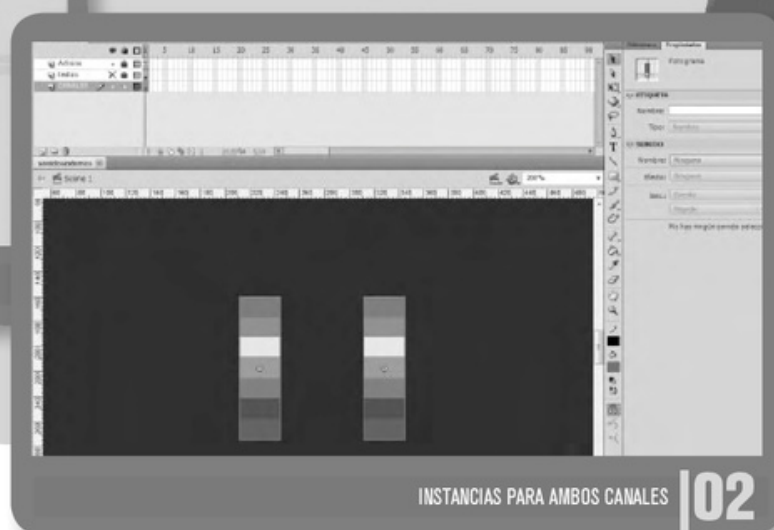
CÓMO CREAR UNA VISUALIZACIÓN

PASO A PASO

Ya explicamos las opciones que ofrece Flash para la manipulación de sonido. Ahora, aprendamos a realizar una visualización.



01 CREAR ELEMENTOS GRÁFICOS



INSTANCIAS PARA AMBOS CANALES

02

A lo largo de este apartado práctico, vamos a realizar una representación de la amplitud del sonido, tomando los valores que posee cada canal. Este es el modo más simple para comenzar a trabajar con el audio de manera gráfica. El resultado serán dos barras que nos mostrarán la amplitud del canal izquierdo y el derecho de los altavoces, respectivamente.

PASO | 01 Vamos a crear un movie clip con los elementos que, posteriormente, animaremos. Para este ejemplo, necesitaremos un fondo del color que deseemos, al cual le aplicaremos un **movie clip** de máscara. A este le asignamos como nombre de instancia "**miMascara**", para, luego, llamarlo desde el código.

PASO | 02 Tomamos el **movie clip** que contiene tanto la máscara como el fondo, y generamos dos instancias de él en el escenario, a las que denominamos **amplitudIzquierda** y **amplitudDerecha**, respectivamente.

PASO | 03 Ya debemos estar familiarizados con este paso, que es un requisito para crear la visualización. Primero, creamos la **URLRequest** y le asignamos la dirección donde se encuentra el **MP3**. Luego,



DE SONIDO

```
1 import flash.net.URLRequest;
2 import flash.media.Sound;
3 import flash.media.SoundChannel;
4 import flash.events.Event;
5
6 var miRequest:URLRequest = new URLRequest();
7 miRequest.url = "mp3/cancion.mp3";
8
9 var miSonido:Sound = new Sound();
10 miSonido.load(miRequest);
11
12
```

03 | CARGAR EL SONIDO EXTERNO

```
1 import flash.net.URLRequest;
2 import flash.media.Sound;
3 import flash.media.SoundChannel;
4 import flash.events.Event;
5
6 var miRequest:URLRequest = new URLRequest();
7 miRequest.url = "mp3/cancion.mp3";
8
9 var miSonido:Sound = new Sound();
10 miSonido.load(miRequest);
11
12 var miCanal:SoundChannel = new SoundChannel();
13 miCanal = miSonido.play();
14 addEventListener(Event.ENTER_FRAME, animarAmplitudes);
15
16 function animarAmplitudes(e:Event):void{
17     amplitudIzquierda.miMascara.scaleY = miCanal.leftPeak;
18     amplitudDerecha.miMascara.scaleY = miCanal.rightPeak;
19 }
```

```
5
6 var miRequest:URLRequest = new URLRequest();
7 miRequest.url = "mp3/cancion.mp3";
8
9 var miSonido:Sound = new Sound();
10 miSonido.load(miRequest);
11
12 var miCanal:SoundChannel = new SoundChannel();
13 miCanal = miSonido.play();
14 addEventListener(Event.ENTER_FRAME, animarAmplitudes);
15
16
```

04 | CREAR EL CANAL Y ASIGNAR EVENTOS

05 | CONTROLAR LA AMPLITUD



06 | PROBAR LA MÁSCARA

generamos una nueva instancia de la clase **Sound** y le pedimos que descargue el sonido externo a través del método **load()**.

PASO | 04 En este paso, vamos a crear el canal que contendrá el sonido anteriormente cargado, y a comenzar la reproducción a través del método **play()**. Por último, añadimos un **listener** que maneje un evento de tipo **Enter Frame**, donde controlaremos la animación de los **movie clips**.

PASO | 05 Ahora solo nos resta controlar el tamaño de la máscara de los **movie clips**. Para hacerlo, le asignamos la propiedad **scaleY**, igual a **leftPeak** y **rightPeak** del canal que reproduce el sonido. Estas propiedades del canal nos brindan valores entre 0 y 1 sobre la muestra de la amplitud de un canal específico, de modo que, asignándole estas propiedades a la máscara como valor, podremos escalarlas para mostrar el fondo de los **movie clips**.

PASO | 06 Si seguimos todos los pasos de manera correcta, las máscaras se escalarán y mostrarán el fondo del clip, tal como se ve en la imagen. Si bien este es un ejemplo simple, sirve para entender de manera didáctica el modo de acceder a los valores de amplitud de sonido de la clase **SoundChannel**.

RESUMEN

En este proceso hemos aprendido a realizar una representación de la amplitud del sonido, tomando los valores de cada canal. Como resultado, observaremos dos barras que nos mostrarán la amplitud del canal izquierdo y del derecho de los altavoces.

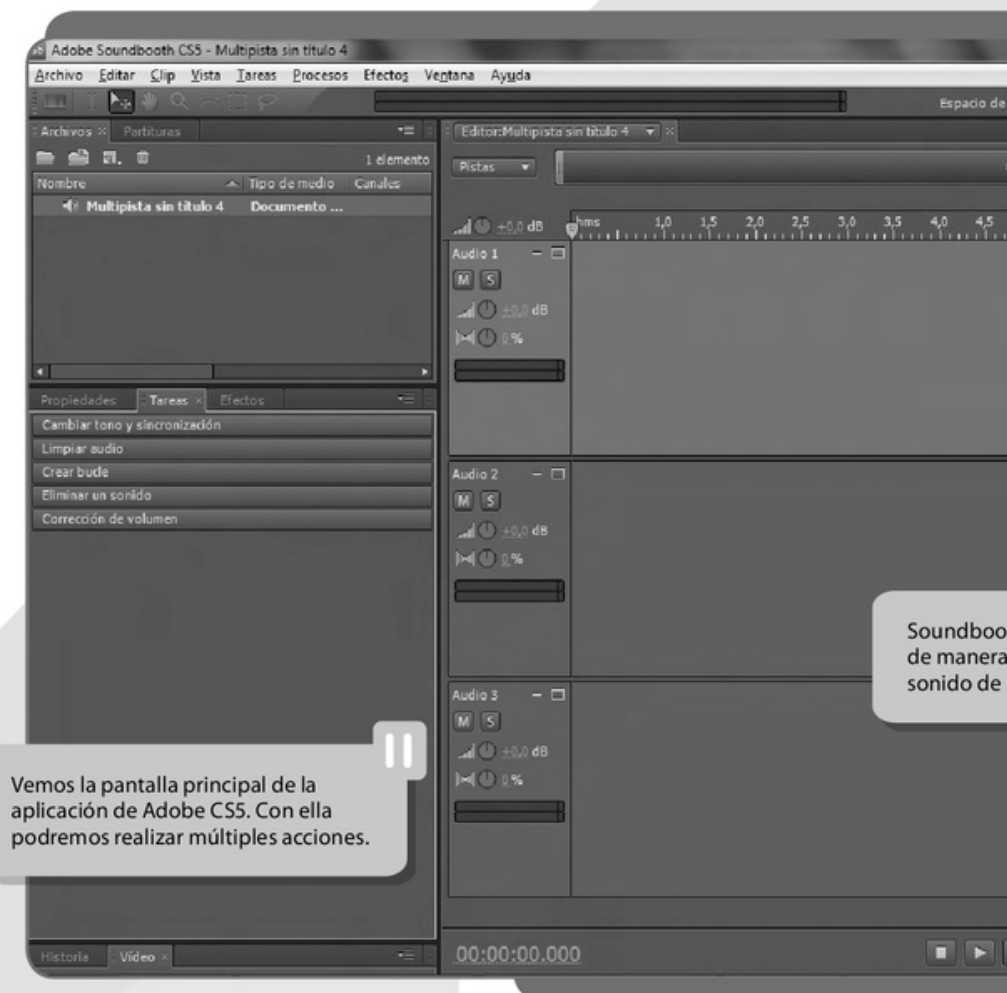
ADOBE CS5

SOUNDBOOTH

En este apartado nos dedicaremos a explorar las herramientas que nos brinda Adobe Soundbooth CS5 para editar archivos de audio, añadir efectos y exportarlos, de manera de poder utilizarlos en nuestros proyectos.

Anteriormente, hemos hablado sobre cómo incorporar sonidos y archivos de audio en nuestros proyectos para brindar un mayor realismo a las animaciones. Si bien vimos cómo realizar y sincronizar sonidos en la línea de tiempo, es importante tener en cuenta que, a medida que las animaciones sean más complejas, necesitaremos tener un mayor control sobre los archivos de sonido si queremos obtener un resultado más preciso. Para lograrlo, existen algunos programas de edición de sonido que permiten trabajar con archivos de audio de manera simple, rápida y eficaz. Si nunca hemos utilizado un software para este fin, es probable que, al entrar por primera vez en Soundbooth, nos resulte un poco confuso. En principio, podemos decir que en este programa para edición de audio, nos encontramos con lo que parece ser una línea de tiempo que posee diferentes capas, donde podremos insertar sonidos de manera simultánea. Esto se debe a que Soundbooth es un editor de sonido cuyos proyectos son multicapa, es decir que podemos incorporar diversos sonidos en capas distintas para que se reproduzcan a la vez. En la parte izquierda de la interfaz veremos una barra dividida en tres secciones principales:

- ▶ En la parte superior está el **panel de archivos**, donde se muestran todos los archivos de sonido que importemos al proyecto, y un buscador que nos permitirá encontrar otros gratuitos desde una megalibrería de Adobe que podemos descargar e incorporar en nuestro trabajo.
- ▶ En segunda instancia, hay cuatro pane-



Vemos la pantalla principal de la aplicación de Adobe CS5. Con ella podremos realizar múltiples acciones.

Soundbooth de manera sonido de l

les cuyas funciones se activan al seleccionar un archivo de audio específico. Desde allí podremos añadir efectos especiales a los sonidos, agregar marcadores o ver las propiedades del archivo seleccionado.

- ▶ Por último, en la parte inferior de la barra hay un panel que contiene las últimas acciones que hemos realizado, por lo que si deseamos revertir algún efecto aplicado, podemos hacerlo desde allí de manera simple e intuitiva.

MULTITRACKING

Como mencionamos anteriormente, la funcionalidad más importante de Soundbooth radica en que es un software destinado a incorporar múltiples pistas de audio dentro de un mismo archivo. Esto nos permite manejar el material tal como lo hacemos al trabajar con capas en Flash. Podemos generar cuantas capas deseemos, incorporar un sonido distinto en cada una y aplicarles efectos o

CON SOUNDBOOTH PODEMOS EDITAR SONIDOS EN CAPAS DISTINTAS DE MANERA SIMULTÁNEA.

transiciones según sea necesario. Soundbooth muestra la onda del sonido en cada una de las capas, para que podamos manipular el audio y visualizarlo todo en forma simultánea. A su vez, debajo de estas capas, encontramos los botones de reproducción, que nos

permiten escuchar los sonidos al mismo tiempo en que los editamos. Esta opción resulta de gran utilidad, porque podemos percibir los cambios aplicados de manera instantánea.

LIBRERÍA DE SONIDOS GRATUITA

Muchas veces necesitamos incorporar en nuestros proyectos archivos de audio específicos, como pasos o disparos, por ejemplo. Con este fin, Soundbooth nos brinda la posibilidad de buscar archivos de sonido de la librería de Adobe y utilizarlos de forma gratuita. Simplemente, debemos ir a la sección **[Score]**, ubicada en el panel superior izquierdo, y seleccionar entre las categorías que se presentan o efectuar una búsqueda.

EFFECTOS DE SONIDO

Aplicar efectos de sonido a los archivos es muy simple con Soundbooth CS5.

Podemos utilizar el panel de efectos, ubicado en la sección izquierda del programa, y seleccionar entre las diversas opciones que se ofrecen por defecto. Más adelante, veremos paso a paso cómo editar un archivo MP3 para añadirle un efecto de "llamada telefónica" o seleccionar entre una inmensa cantidad de efectos que nos permitirán obtener los resultados que estemos buscando, en cuestión de segundos.

PISTAS DE AUDIO DE VIDEO

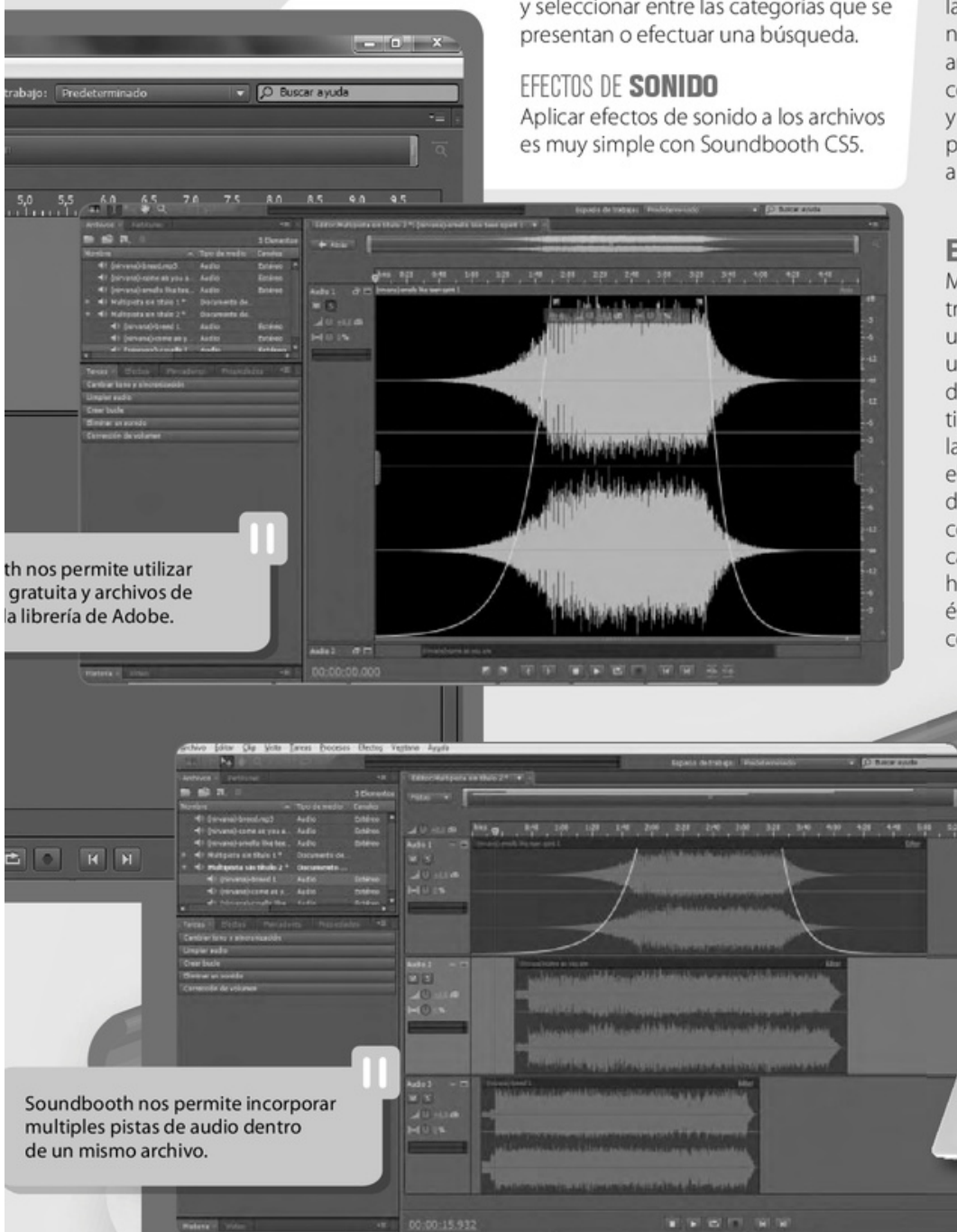
Otra funcionalidad muy interesante de Adobe Soundbooth es la capacidad de incorporar archivos de video, con la opción de tomar las pistas de sonido necesarias de él y así crear un nuevo archivo de audio. Esta posibilidad, junto con el uso de los efectos de sonido y de la librería online de Adobe, nos permitirá llevar nuestros proyectos a un nivel mucho más profesional.

EDICIÓN A TRAVÉS DE LOS CANALES

Muchas veces, necesitamos hacer una transición entre dos sonidos, efectuar un efecto de entrada gradual o recortar una canción. Utilizando las herramientas de Soundbooth, podremos realizar este tipo de operaciones directamente sobre la capa de trabajo, donde se encuentra el sonido. Para hacerlo, Soundbooth nos da la posibilidad de usar los puntos de control dentro de la línea de tiempo de cada canal. De esta manera, podemos hacer un paneo del sonido si es que éste es estéreo, o aplicar efectos simples, como de entrada o salida gradual. ◀

th nos permite utilizar
gratuita y archivos de
la librería de Adobe.

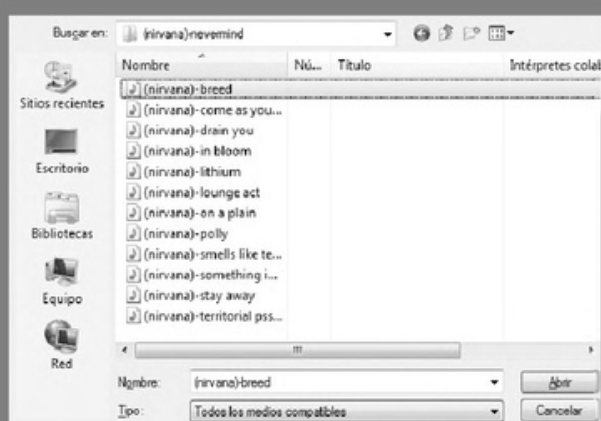
Soundbooth nos permite incorporar
múltiples pistas de audio dentro
de un mismo archivo.



CÓMO EDITAR CON SOUNDBOOTH

PASO A PASO

A lo largo de este apartado, veremos cómo introducir varios sonidos en un archivo de Soundbooth para su posterior manipulación.



01 | AGREGAR ARCHIVOS A LA LIBRERÍA



ARCHIVOS EN LAS PISTAS | 02

En este procedimiento, aprenderemos a añadir efectos desde Soundbooth y a realizar transiciones de manera simple entre dos sonidos distintos. Al finalizar, entenderemos en profundidad el proceso de edición de sonido en Adobe Soundbooth CS5, para así darles mayor profesionalismo a nuestros proyectos en Flash. ◀

PASO | 01 Para importar los archivos, ya sean de sonido o de video, vamos al panel de archivos y hacemos clic en el icono representado por una carpeta abierta; también podemos ir a **[Archivo/Importar/Archivos]**. Una vez que se abre el cuadro de diálogo, seleccionamos todos los que necesitamos y hacemos clic en **[Abrir]**.

PASO | 02 Luego de importar los archivos, nos dirigimos a la librería y hacemos clic sobre nuestro proyecto de Soundbooth para, posteriormente, arrastrar los archivos de audio sobre la línea de tiempo. Es recomendable que cada uno vaya a una capa distinta, de manera que tendremos tantas capas como archivos de sonido haya.

PASO | 03 Si seleccionamos una de las capas, veremos que en los extremos de las ondas de sonido, aparecerán dos pequeños cuadrados que funcionan a modo de puntos de control. Al arrastrarlos sobre la capa, podremos generar el efecto deseado, que se verá representado a través de las líneas amarillas. Cuanto más acerquemos los extremos hacia el interior, veremos que la onda de sonido se modificará en su amplitud.

04 | MODIFICAR EL VOLUMEN



AGREGAR EFECTOS | 05

03 | EFECTOS DE ENTRADA Y SALIDA



EXPORTAR ARCHIVO | 06

PASO | 04 Para variar el volumen de un clip, hacemos clic sobre él, para que Soundbooth muestre un pequeño panel sobre la línea de tiempo. Allí aparecerán algunas herramientas de control en forma de círculos; al modificar su rotación, incrementamos o disminuimos la amplitud del sonido.

PASO | 05 Para añadir un efecto a un sonido, lo seleccionamos en la línea de tiempo y, luego, nos dirigimos al panel correspondiente. Una vez allí, en el menú desplegable elegimos los distintos efectos disponibles. Para nuestro ejemplo, utilizaremos el efecto **[Reverberación de circunvolución]**, que imitará una llamada telefónica.

PASO | 06 Una vez que aplicamos todos los efectos deseados en nuestro archivo de Soundbooth, solo resta exportarlo en

formato MP3. Para hacerlo, nos dirigimos a **[Archivo/Exportar/Multitrack Mixdown]**. Al hacer clic, aparecerá un pequeño panel que mostrará los tipos de archivos disponibles para exportarlo. Cuando aceptemos, habremos finalizado con el trabajo.

RESUMEN

En este apartado práctico hemos aprendido a implementar las aplicaciones elementales para la edición de un archivo de sonido y su exportación en formato MP3.

SITIOS EXPERIMENTALES DE SONIDO

En este apartado analizaremos algunos ejemplos novedosos de sitios que experimentan con Flash a nivel de sonido, con el objetivo de inspirarnos en la creación de nuestros propios proyectos.

A lo largo de estas páginas, pudimos ver las posibilidades que nos brinda Flash para implementar archivos de audio en nuestros proyectos, ya sea a través de la línea de tiempo, para generar animaciones que se sincronicen con nuestros sonidos; o utilizando ActionScript 3.0, para realizar cargas externas o complejas manipulaciones de audio. Si bien hemos incursionado en técnicas simples y complicadas, es importante destacar que lo que hemos aprendido hasta el momento son requisitos mínimos para la realización de proyectos profesionales. Pero si nuestro objetivo es impactar a los usuarios, debemos llevar la complejidad de nuestros proyectos un nivel más allá de la simple manipulación de archivos. Descubriremos que, en lo que respecta a Flash, los límites los ponemos nosotros (los diseñadores y desarrolladores), y utilizando algunas técnicas aprendidas y otras que nos quedan por ver, sorprenderemos a nuestros usuarios de maneras jamás pensadas.

EXPERIMENTOS SOBRE VISUALIZACIÓN DE SONIDO

Como vimos anteriormente, una de las novedades de ActionScript 3 fue la incorporación de clases que permiten generar visualizaciones de sonido reales, que utilizan valores como la amplitud o la frecuencia de un archivo de audio. Si bien hemos incursionado en el proceso de creación de una visualización de sonido que tome la amplitud de este como variable para realizar una animación, a continuación analizaremos algunos ejemplos con mayor nivel de dificultad, que nos permitirán entender cómo unificar varias técnicas de

ActionScript 3.0 en un mismo trabajo. Una de las posibilidades para efectuar visualizaciones de sonido es el empleo de técnicas de animación que difieran de la simple tarea de escalar los elementos a nivel de volumen. Si utilizamos lo

aprendido sobre la API de dibujo de ActionScript 3.0, junto con el uso de la clase `computeSpectrum`, podremos generar situaciones visualmente muy ricas, como las que vemos en <http://blog.the-flashblog.com/?p=197> <http://blog.the-flashblog.com/?p=197>.

En este sitio hay ejemplos que utilizan librerías 3D, como Papervision o Away3D;

AUDIOTOOL ES UNA HERRAMIENTA DESARROLLADA EN FLASH QUE NOS PERMITE CREAR MÚSICA, ALMACENARLA Y COMPARTIRLA ONLINE.

www.stonewall.co.za es un sitio experimental en donde podemos crear música con nuestra creatividad.



Con Audiotool podemos editar sonido en línea y compartirlo con más personas.

sobre todo, veremos que los casos más interesantes hacen un intenso uso de técnicas de animación, como los sistemas de partículas y, también, el manejo de filtros, como Blur o Glow. Es importante hacer hincapié en la relevancia de usar el método **ComputeSpectrum** de la clase **SoundMixer**, ya que esta nos permite tener un mayor control sobre las animaciones, al brindarnos más cantidad de valores y variables para implementar.

AUDIOTOOL

Si hablamos de sitios experimentales, no podemos pasar por alto **www.audiotool.com**. Audiotool es una herramienta desarrollada íntegramente en Flash, que no solo permite crear música, almacenarla y compartirla con el resto de la comunidad, sino que ofrece hacerlo de manera online y por medio de nuestro navegador. Sin embargo, lo más importante es que los sonidos generados con la aplicación están desarrollados, en su mayoría, en ActionScript 3.0. Esto significa que la aplicación no realiza descargas de archivos MP3, sino que los



VISUALIZACIÓN DE METADATOS

Podemos seleccionar qué campos u opciones del panel Metadatos deseamos visualizar, en caso de que no sea necesario utilizar todos. Para seleccionar estos campos, debemos hacer clic en el icono de [Visualización de Metadatos] en el ángulo superior izquierdo del panel. Allí seleccionaremos los campos que deseamos utilizar y lo guardaremos como nuestro ajuste personalizado de visualización. Podemos generar tantas configuraciones de ajustes como sean necesarias.

sonidos son generados en tiempo real por librerías de código desarrolladas por el equipo de Audiotool. Como explicamos anteriormente, en la sección dedicada a las librerías de sonido, Audiotool hace uso de las funciones de ActionScript 3.0 que permiten generar sonidos, y utiliza métodos con alto nivel de complejidad para aplicar efectos que simulan a las herramientas disponibles en un estudio de grabación. Si bien este tipo de desarrollos son llevados a cabo por personas altamente experimentadas, podemos incursionar en la

creación de sonidos sintéticos utilizando la librería **Tonfall**, de la cual hablamos en la sección sobre librerías de sonido. Sin duda, este es uno de los mejores ejemplos que podemos encontrar en la Web, para darnos cuenta de que Flash es una herramienta que no sólo se utiliza para realizar sitios web, sino que también permite generar aplicaciones con alto nivel de interactividad, que brinden a sus usuarios experiencias ricas e inmersivas.

UTILIZAR EL SONIDO DE MANERA INTELIGENTE

No hay mejor manera de entretener a nuestros usuarios que brindándoles interactividad. Existen numerosos sitios que utilizan el sonido como elemento envolvente para transmitir su identidad; sin embargo, **stonewall.co.za** nos permite crear nuestra propia música mientras el sitio es descargado. Esta es una de las formas más inteligentes de utilizar el sonido que podemos encontrar en la Web, ya que permite al usuario interactuar de manera lúdica con los elementos del sitio mientras se realiza una tarea que, de otro modo, podría resultar aburrida o engorrosa. En la mayoría de los sitios, el sonido se usa como un elemento que acompaña o molesta al usuario durante su visita. Sin embargo, en este apartado vimos algunos ejemplos de cómo brindar un sentido diferente, que nos permita dar al usuario elementos con los cuales interactuar o transmitirle un mensaje comunicacional. Ahora sólo nos resta utilizar las herramientas vistas durante este curso para implementar el sonido en nuestros proyectos de forma creativa e inteligente. ◀



ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué ventajas nos entrega utilizar sonido en nuestro proyecto?

02 ¿Cómo podemos incorporar sonido en una animación?

03 ¿Qué son los sonidos internos?

04 ¿Qué son los sonidos externos?

05 ¿Para qué sirve la sincronización de sonido?

06 ¿Cómo podemos importar un archivo de audio?

07 Mencione las ventajas de ActionScript 3 para manejar sonido.

08 ¿Qué clases ofrece ActionScript 3 para el sonido?

09 ¿Qué son las librerías de sonido?

10 ¿De qué forma podemos crear una visualización?

PRÁCTICAS

01 Utilice sonidos internos y externos en su proyecto.

02 Sincronice sonido con botones.

03 Sincronice sonido con animaciones.

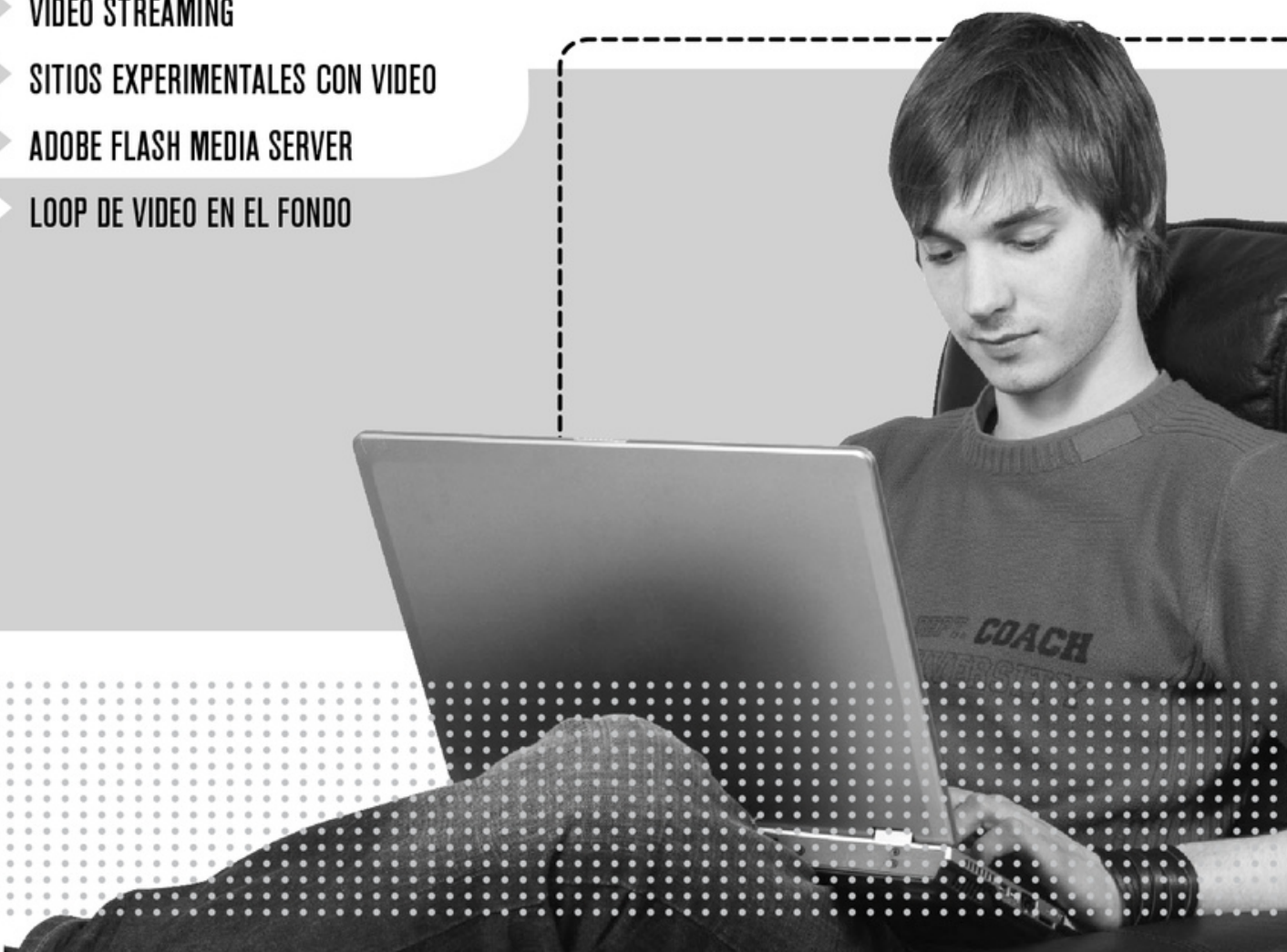
04 Utilice las clases de ActionScript 3 para trabajar con sonido.

05 Capture audio desde un micrófono utilizando la clase adecuada.



Video

- ▶ VIDEO EN LA WEB
- ▶ FORMATOS DE VIDEO
- ▶ ADOBE MEDIA ENCODER
- ▶ INCORPORAR VIDEO A NUESTRAS PELÍCULAS
- ▶ VIDEO STREAMING
- ▶ SITIOS EXPERIMENTALES CON VIDEO
- ▶ ADOBE FLASH MEDIA SERVER
- ▶ LOOP DE VIDEO EN EL FONDO



VIDEO EN LA WEB

El avance en programas compresores y en el ancho de banda disponible revolucionó las aplicaciones y el uso de video en la Web. Sitios como YouTube ofrecen una mejor experiencia multimedia.

El uso de video, desde sus principios, implicaba un gran problema de implementación y reproducción. Este recurso requería no solo de plugins o complementos especiales instalados en el sistema, sino que también consumía un gran ancho de banda con respecto a las tasas de transferencia disponibles hace unos años. En la actualidad, no es que el peso de los videos se haya reducido; por el contrario, con respecto a nuevas tecnologías como el HD (alta definición), se incrementó. Lo que ha ocurrido es que se amplió de manera considerable la velocidad de descarga, con lo cual los videos se volvieron más accesibles para la mayoría de los usuarios.

¿QUÉ SE NECESITA PARA VER VIDEOS?

Los requerimientos actuales para reproducir video desde un navegador no están lejos de los ofrecidos por las computadoras más básicas. Pero es importante saber que una buena cantidad de memoria RAM y una placa de video de media calidad son condiciones básicas. También se requiere una conexión a Internet de, por lo menos, 512 KB/s para ver video sin cortes a una calidad adecuada. En este momento, existen videos en formatos de mayor calidad, como HD 240p, 360p, 720p y 1080p. Claro que, cuanto mejor es la calidad, mayor es el peso del material, por lo que se requiere de mayor velocidad de descarga si no queremos percibir cortes. Otro aspecto importante es el software: se necesitan programas conocidos como **códecs** o **decoders** para visualizar los videos. En la actualidad, contamos con diferentes formatos de video, algunos más populares que otros. Los códecs son los encargados de descomprimir o interpretar el video y mostrarlo visualmente de manera correcta. Si



LOS CÓDECS SON LOS ENCARGADOS DE INTERPRETAR EL VIDEO Y MOSTRARLO DE MANERA CORRECTA.

no disponemos de los adecuados, tal vez no podamos abrir el archivo, o solo escuchemos el audio pero no veamos ninguna imagen. Los códecs más utilizados vienen inclui-

dos en el sistema operativo, aunque hay formatos, como el **H264**, que requieren, por ejemplo, instalar Flash Player para verse desde la Web. También existen los formatos **MOV** o **MP4**, que pueden ser preproducidos por Player, de Flash, o QuickTime, de Apple.

¿POR QUÉ TANTOS FORMATOS?

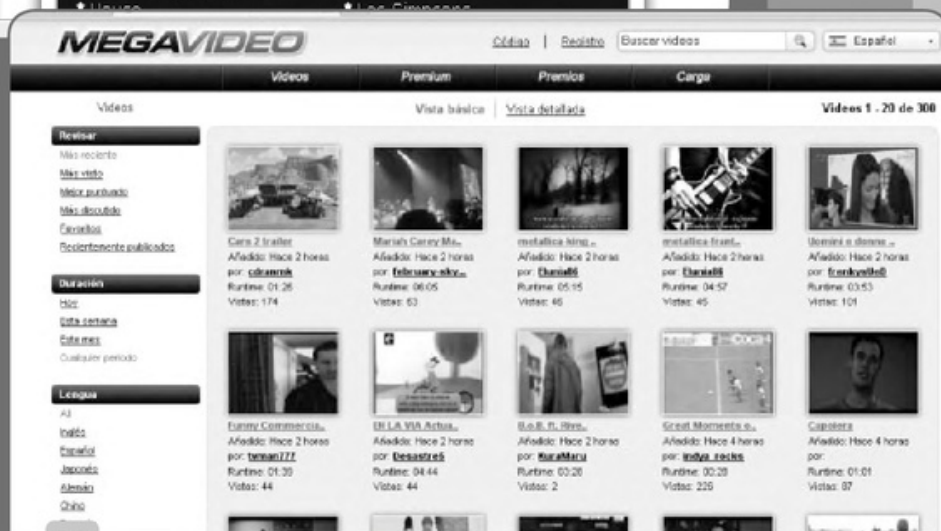
Los videos están compuestos por una secuencia de imágenes a cierta cantidad de cuadros por segundo. Si, por analogía, usamos el mismo criterio que con las imágenes, podemos llegar a la conclusión de que, cuanto mayor es la calidad de cada imagen que lo compone, mayor será el peso total; lo mismo sucederá si la imagen tiene muchos

colores o utiliza más datos por píxel. Otro factor importante es el tamaño con respecto a su alto y ancho: cuanto más grande es la área del video, más bytes ocupará. Con este criterio, podríamos decir que los formatos dependerán no solo del medio en el que se digitalizó el material, sino también del medio de transporte. Si este es un disco DVD, el formato puede ser **BOV** a una calidad HD sin ningún tipo de compresión. En cambio, si el medio

es la Web, necesitamos un formato adecuado que asegure una buena relación calidad/peso, como el **MP4** o el **FLV**. La compresión se refiere a los bytes quitados, aquellos que no son necesarios para almacenar un dato completo, ya que dicha información puede ser reconstruida al momento de descomprimir el material, tomando otros datos similares o contiguos. Cuanto más alto es el factor de compresión, mayor es la cantidad de datos eliminados del archivo original.

Es importante recordar que una compresión excesiva lleva a la pérdida de datos y, por ende, produce una menor calidad del video. Según el método de compresión o códec que se utilice, lograremos diferentes resultados. Los formatos de video más conocidos son **AVI** y **MOV**, y cada uno de ellos cuenta con sus compresores, como **DivX** para **AVI** y **H264** para **MOV**. En conclusión, podemos asegurar que, a la hora de elegir un formato, no hay nada que asegure cuál será mejor; simplemente, el más adecuado variará según las circunstancias y el resultado que se quiera obtener. Lo ideal es lograr menor peso y mayor calidad; si satisfacemos esa ecuación, cualquier formato servirá. No olvidemos que **H264**, **MOV**, **MP4** y **QT** tienden a producir excelente calidad con menor peso. En la actualidad, Adobe Flash soporta video en formato HD, es decir, el nativo **FLV** (Flash Video). Además, con cualquier versión superior a Flash Player 10, podremos visualizar videos en formato **MOV**, **MP4**, **FLV**, **F4V** o **H264**. Es importante aclarar que Flash Player no reproduce todos estos formatos por sí mismo, por lo que se precisa la conversión en el caso de la reproducción online. Como dijimos anteriormente, el formato adecuado variará según el uso que queramos darle al material. Si lo que queremos es mostrar un video en alta calidad a un tamaño razonable y con un peso bajo, recurriremos a **MP4** o **H264**. En cambio, para un video de fondo de menor calidad y bajo peso, optaremos por el **FLV**. Esto no quiere decir que los videos **FLV** sean siempre de baja calidad, porque ambos parámetros (calidad y peso) los definiremos desde Flash Encoder cuando preparemos la película. Más adelante, veremos paso a paso el proceso de compresión. Uno de los factores más importantes de los archivos **FLV** es que soportan un canal alpha (transparencia) igual que los **MOV** y, también, incorporan un sistema de marcadores mediante metadatos llamado **cuepoint**. Estos son puntos de anclaje donde el video ejecuta un evento cuando la cabeza lectora pasa por allí. Es una función que resulta útil para subtítulos, interactividad o sincronismo de eventos. ◀

www.seriesyonkis.com es un excelente sitio en donde podemos ver series y películas con alta calidad de streaming.



www.megavideo.com es otra de las referencias de streaming en vivo. También ofrece la posibilidad de descarga de archivos.

FORMATOS DE VIDEO

Conocer los formatos de video más utilizados en la Web nos será de suma utilidad, sobre todo, cuando necesitemos establecer una relación adecuada entre la calidad del material y su peso. Veamos en detalle este tema.

Sabemos que existen diferentes formatos de imágenes, tales como JPEG, TIF y GIF, entre otros; y que también hay algunos específicos para los videos. Comprender cómo funciona cada uno de ellos nos ayudará considerablemente a lograr mejores resultados con nuestros desarrollos web.

MPEG

Esta sigla proviene de *Moving Picture Experts Group*, un grupo de trabajo encargado de impulsar estándares con respecto al uso de imágenes. Los videos que se encuentran en este formato se caracterizan por una compresión que los hace más livianos al momento de transportar. El sistema que se aplica para lograr este resultado consiste en la fragmentación de los datos y el almacenamiento de aquellos que son diferentes. Los que se repiten son quitados para disminuir el peso total del material. Hace algunos años, el MPEG fue, sin dudas, uno de los formatos más adecuados y prometedores para la Web. Actualmente, ha sido desplazado por otros, como H264 y OGG Theodora, que ofrecen mejor relación calidad/rendimiento.

AVI

El formato AVI (sigla de *Audio Video Interleave*) es el propuesto por Microsoft desde 1992 como contenedor de videos y audio. Esto significa que el video o el audio no necesariamente tienen que ser AVI, ya que este es un contenedor.



SOBRE LOS CONTENEDORES

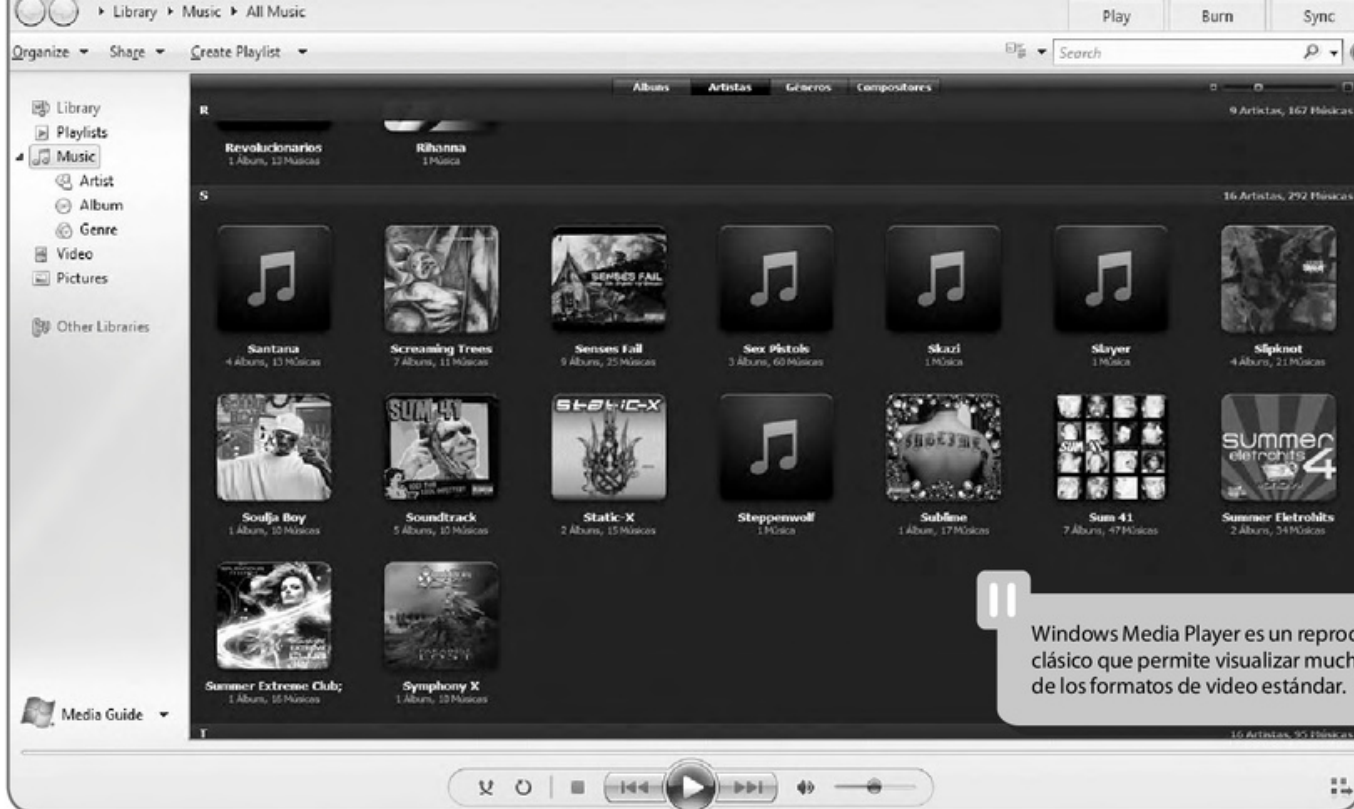
Gran parte de los archivos multimedia contienen datos de audio y de video. Además, también incorporan referencias que permiten la sincronización de ambos elementos. Cada uno de estos datos puede ser manipulado por medio de programas, procesos o hardware diferentes. Sin embargo, para que estos streams sean útiles para almacenar o transmitir, deben ser encapsulados juntos. Esta función es realizada por un formato de archivo de video denominado contenedor, como MPG, AVI, MOV, MP4, RM, OGG, MKV o TTA. Algunos de estos formatos están limitados a contener streams que se reducen a un pequeño juego de códecs, mientras que otros son usados para tareas más generales.

El códec que interpreta el flujo puede reproducir otros formatos, como DivX, MP3, AC3, etc. La propiedad más importante de este formato es, sin dudas, su capacidad de contener varios flujos de información en el mismo fragmento de datos. Esta particularidad se denomina **interleave**, y da la posibilidad de que, cuando vemos una película, el audio cambie de un idioma a otro. Esto quiere decir que el cuadro (imagen) y el sonido están vinculados uno a otro. Este formato aún sigue siendo muy popular, dado que la distribución de películas hogareñas –en su mayoría, en CD-ROM– contó con versiones en AVI DivX, que permitió lograr muy buena calidad sin requerir de un DVD-R. En la actualidad, es muy raro ver la implementación de AVI en la Web.

FLV

Flash Video es un formato contenedor similar a AVI, es decir, diseñado para transmitir video en la Web. Fue creado por la empresa Macromedia y continuado por Adobe después de que esta adquiriera a la primera.





Flash Video puede ser reproducido por Flash Player principalmente, aunque si se instalan los códecs de ffdshow, también se puede ejecutar en otros reproductores, como Windows Media. Actualmente, Flash Player se encuentra en casi todos los equipos, lo que hizo que sitios como YouTube, Yahoo Video y Google Video decidieran utilizarlo como reproductor estándar. La calidad del video que ofrece este formato, en general, es buena. Después de la versión 8, se introdujo un nuevo códec de compresión, **On2 TrueMotion**, que incrementó la calidad de manera considerable. Las últimas versiones de FVL ya trabajan con soporte para H264, de modo que brindan calidad en alta definición (HD).

ENTONCES, ¿CUÁL USAR?

Claro está que, para la Web, lo ideal es usar Flash Video (FLV), debido a sus prestaciones en cuanto a la relación peso/calidad y a la nueva posibilidad de soportar video en HD. Pero no solo estas cualidades son las que lo destacan del resto: también tenemos que considerar que Flash Video puede interactuar con el lenguaje de programación que interpreta Flash Player (ActionScript). Esto abre un gran abanico de posibilidades, desde la manipulación del tiempo y los eventos, hasta la interactividad necesaria para el usuario. En este caso, el video deja de ser una secuencia lineal, y pasa a convertirse en un medio interactivo. Flash Video también ofrece la posibilidad de manejar canales alpha para conseguir montajes sobre fondos croma y transmitir video en modo streaming (vivo) a numerosos usuarios al mismo

QUICKTIME BRINDA LOS CÓDECS SECUNDARIOS QUE REQUIERE FLASH VIDEO ENCODER.

tiempo. En este momento, es el formato más difundido, con mayor penetración y compatibilidad. Se lo usa en películas, programas en vivo, y hasta en salas de chat con audio y video.

EL REPRODUCTOR DE FLV

Flash Video puede ser visto en la mayoría de los sistemas operativos por medio de la aplicación Adobe Flash Player, el plugin disponible para navegadores web. También puede reproducirse con otros programas de terceros, como MPlayer, VLC Media Player, o cualquier otro reproductor que use filtros DirectShow, como Media Player

Classic, Windows Media Player y Windows Media Center, siempre que el filtro ffdshow esté instalado.

CONVERSIÓN DE ARCHIVOS AVI Y MPEG EN FLV

Cuando se instala el programa de desarrollo Adobe Flash, este incluye una aplicación llamada Flash Video Encoder, encargada de convertir formatos como AVI, MPEG, MOV, MP4, MP3 y otros, en formato FLV. También existen aplicaciones de conversión de terceros, aunque lo más recomendable es usar la de Adobe si queremos conseguir buenos resultados. Es importante destacar que para convertir formatos de video a Flash Video, es necesario tener instalado QuickTime, ya que este nos brindará los códecs secundarios que requiere Flash Video Encoder. ◀



ADOBE

MEDIA ENCODER

Es un software con un mecanismo de codificación que incluye numerosos ajustes asociados a determinados formatos de exportación, como MPEG-2, Adobe Flash Video y H.264. Veamos de qué se trata y cómo utilizarlo.

Adobe Flash cuenta con su propio encoder o códec de video para su formato FLV. Para este fin, dispone de una herramienta que codifica en dicho formato de manera rápida y simple, por lo que resulta la más adecuada para crear archivos FLV. Existen otras aplicaciones de otros proveedores que logran alta calidad, pero que ofrecen pocas opciones de salida o ajustes con respecto a esta.

NOCIONES BÁSICAS

Adobe Media Encoder se instala junto con el paquete Adobe Collection o cuando instalamos Adobe Flash profesional. De acuerdo con la versión de Flash que tengamos, dispondremos de una u otra versión del encoder. Aunque la versión de Adobe instale y ejecute el Media Encoder correctamente, no podremos crear archivos si no tenemos instalado el programa **QuickTime** en nuestra computadora. Es importante recordar que, en las últimas versiones de Flash, existen mejoras en el codificador, como el soporte para codificar archivos **H264**. También, la última versión cuenta con un control de edición de salida mucho más eficaz que las anteriores.

ENCODER AUDIO Y VIDEO

Con respecto al audio, este se codifica en el archivo en formato **MPEG II** intercalado con el video, motivo por el cual se define el formato FLV como un contenedor. Desde esta herramienta, también podremos controlar la salida de video en cuanto a características referidas a la calidad de la imagen. En este caso, se aplica la misma regla que detallamos anteriormente: mayor

calidad implica mayor peso; mayor tamaño en ancho y alto del video determina un archivo más pesado.

Ya sabemos que el tamaño del video en su salida define su peso; es decir, si la salida es de **800 píxeles** de alto por **600 píxeles** de ancho, el archivo será mucho más pesado que si tuviera **240 x 180**. Normalmente, los videos en Flash suelen tener medidas inferiores a **480 píxeles**. No sucede lo mismo con el video en HD: por ejemplo, si este fuese en **720 píxeles**, la medida del video sería de **1280 píxeles** de ancho por **720 píxeles** de alto, lo cual incrementaría sustancialmente el peso del archivo. Entonces, hay que saber que formatos como el HD no sólo aumentan su peso por el tamaño del video, sino que también se agregan otros aspectos, como la velocidad en Kbps. Esta unidad es uno de los factores más



SOBRE MEDIA ENCODER

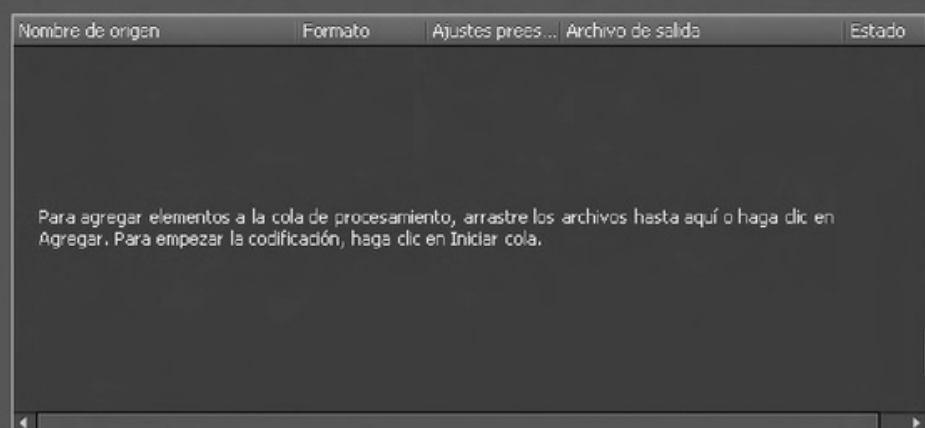
Cuando exportamos un archivo de película para medios de comunicación con distintas características, como pantalla completa o diferente velocidad de fotogramas, por lo general, tenemos que desentrelazar los cuadros, recortar la imagen o aplicar determinados filtros. Adobe Media Encoder permite realizar estas tareas como opciones de **precodificado**. También admite especificar acciones posteriores a la codificación, entre las que se incluyen la generación de un archivo de registro o la descarga automática del archivo exportado en un servidor especificado.

UNA CARACTERÍSTICA QUE TIENE EL FORMATO FLV ES LA CAPACIDAD DE ALMACENAR METADATOS.

importantes en el muestreo de video, y se refiere a la cantidad de información que puede viajar en un flujo de datos. Cuanto mayor sea este valor, más alta será la calidad de cada fotograma, porque se dispondrá de más información de muestreo. Flash Media Encoder, en su versión más reciente, utiliza, por defecto, un valor de 1600 Kbps, lo cual asegura una calidad muy buena, pero un peso considerable si lo que se quiere es mostrar un simple video. Entonces, una alternativa es que, cuanto más pequeño sea el video, más podremos aumentar los Kbps. Es importante recordar que si bajamos demasiado los valores, se producirá un efecto de pixelado sobre el video a causa de la pérdida excesiva de datos. Como siempre, lo mejor es ir probando según la necesidad y buscar el mejor criterio entre peso y calidad.

ENCODER Y LOS FPS

Otro factor que manipula el encoder son los fotogramas por segundo (FPS). Por defecto, este parámetro es de **30 FPS**, lo que resulta adecuado para la mayoría de los videos. No es necesario cambiarlo, excepto que el editor del material (quien creó el video original) así lo requiera. Tampoco esto afectará al sonido, ya que este medio va codificado en la misma línea de tiempo del video, para asegurar una

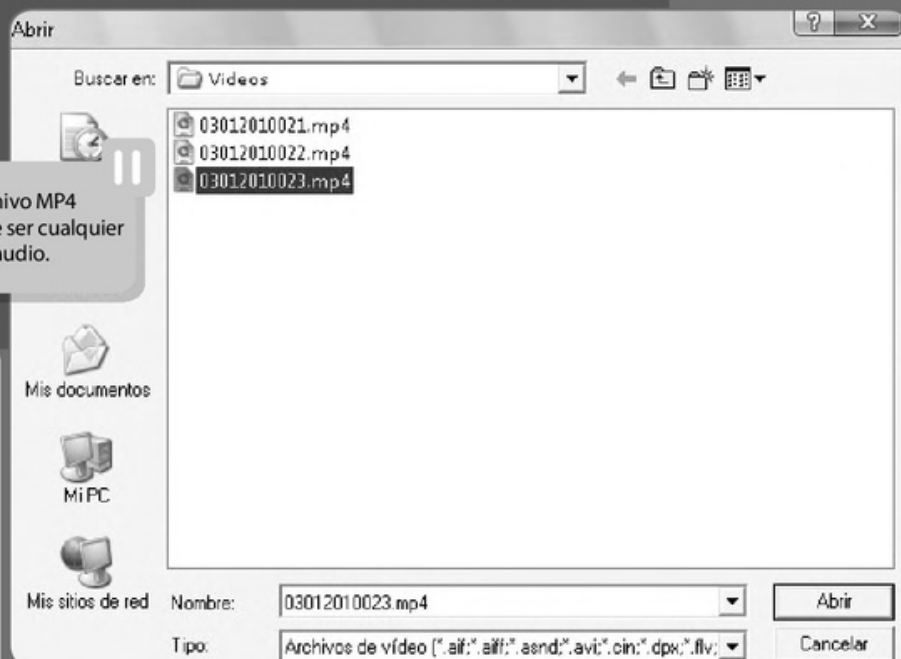


Para añadir un archivo en Adobe Media Encoder podemos arrastrarlo o hacer clic en el botón **[Añadir]**.

Mensaje:
Video:
Audio:
Velocidad de bits:

En este caso cargamos un archivo MP4 para convertirlo en FLV; puede ser cualquier formato admitido de video o audio.

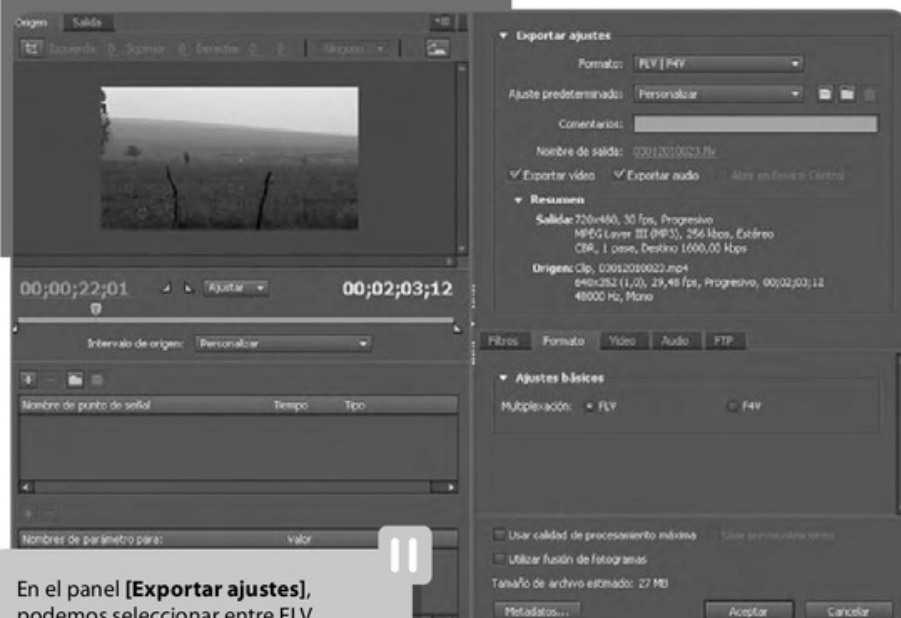
Tiempo de cola transcurrido: 00:00:00



correcta sincronización entre ambos. Con respecto al audio, este se incorpora al contenedor FLV intercalado con el video; todo compone un solo flujo de datos, no como en el caso del formato AVI, que podía incluir varios. Por ese motivo, los archivos FLV no pueden incluir varios idiomas en el mismo video. El formato en el que se guarda el flujo dentro del FLV es **MPEG II**. En él podemos ajustar la calidad de Kbps, lo cual también influirá en el peso final del archivo. Es decir que, del mismo modo que en el video, cuantos más Kbps tenga, más pesado será el archivo, pero mejor resultará la calidad del audio. También podemos optar entre un flujo de audio mono o estéreo.

CARACTERÍSTICAS FLV

Una de las características más importantes de los videos FLV (y que ningún otro formato posee) es la capacidad de almacenar metadatos, como los cuepoints. Ya mencionamos que estos son puntos de señal incrustados en el video a cierta cantidad de tiempo, los cuales, mientras el material se reproduce, emiten un evento para indicar valores como tiempo, tipo de señal, sincronismo, y otros. Estos elementos son de suma utilidad cuando queremos sincronizar subtítulos con el video, o incluir eventos en otros objetos en Flash según la posición del video. ◀



En el panel **[Exportar ajustes]**, podemos seleccionar entre FLV o H264; si es para Web debe ser FLV.

INCORPORAR VIDEO A NUESTRAS PELÍCULAS

Adobe Flash cuenta con varios métodos para la incorporación de videos en la línea de tiempo, teniendo en cuenta si este es un archivo local o un streaming.

La incorporación de video sobre la línea de tiempo es uno de los procesos más simples a la hora de manipular este material en formato FLV. Con solo importar o arrastrar el video sobre ese elemento, el programa muestra una ventana de diálogo y, mediante unos simples clics, ya dispondremos del video listo para ver. Cuando incorporamos un video, Flash muestra una serie de opciones para llevar a cabo la acción. Lo primero que debemos hacer es indicar dónde se encuentra el archivo de video, es decir, si está en el equipo (computadora local) o en la Web (Internet). Esta diferenciación es muy importante, ya que si el archivo está en la computadora, se lo colocará en el archivo **FLA**, incrustándolo o vinculándolo por medio de la ruta; con unos mínimos ajustes, ya podremos reproducirlo. Por lo contrario, si el archivo está en la Web, deberíamos considerar otros factores, como los permisos de acceso o ActionScript extra para reproducirlo.

ARCHIVOS DENTRO Y FUERA DEL EQUIPO

Existen dos aspectos que debemos destacar en este tema: por un lado, la referencia de **video en el equipo**, y por el otro, la noción de **video streaming**. Cuando hablamos de lugar "local", nos referimos a una ubicación que el archivo tiene dentro de nuestra computadora. Este archivo no cambia su estado cuando subimos el SWF y el video a la Web. Por su parte, es importante comprender el sentido estricto de la noción de video

streaming, que se refiere a un proceso generado por fuera de la PC, o sea, en Internet. El streaming lo realiza un servidor, es decir, una computadora que está en Internet. Como ejemplo de esto, podemos mencionar el caso de una videoconferencia, durante la cual vemos el video en vivo a medida que nos conectamos. El otro caso, que no es video streaming, puede ser la descarga progresiva de un archivo que está (previamente) completo, y que cada usuario que se conecte irá

viendo desde el principio. En este apartado nos centraremos en el caso del video **"en el equipo"**.

INCORPORACIÓN DE VIDEO

Para incorporar un video, nos dirigimos al menú **[Archivo/Importar]** o presionamos el atajo de teclado **<Ctrl + R>**. También podemos arrastrarlo directamente a la línea de tiempo. Luego, tenemos que cargar el video externo con componente de reproducción. Esta opción adjunta un componente de



Una vez agregado a la lista de videos por codificar, podemos hacer clic en **[Ajustes]** para lograr la mejor configuración de salida.

reproductor de video con sus respectivos controladores, como Play, Pause, Progress y Stop, entre otros. El elemento incorporado puede ser personalizado más adelante desde los diversos tipos de controles, así como también desde los colores. Esta opción vincula el archivo externo al reproductor mediante una **URL** que se encuentra dentro de la carpeta local. Es recomendable que el video esté dentro de la misma estructura raíz de la Web. El peso en esta opción es relativamente bajo, ya que sólo el SWF incorpora el player de video, y el player llama al video externamente.

INCORPORAR FLV EN SWF

Otro proceso que debemos realizar es incorporar el FLV al SWF y reproducirlo en la línea de tiempo. Con este método de inserción, logramos colocar el material en algún contenedor, como en la línea de tiempo principal. En este caso, se utilizarán cuantos fotogramas sean necesarios para que entre el video completo. También podemos hacerlo dentro de un contenedor que puede ser un sprite o un movie clip. Esta última opción no cuenta con los controles de reproducción, como la anterior; por lo tanto, si queremos controlar el flujo de video, tendremos que realizarlo mediante el uso de ActionScript. Cuando incorporamos un video de este modo, éste se incrusta en el SWF, lo cual genera un peso adicional considerable con respecto al método anterior. La utilidad más común de este sistema es cuando manejamos videos muy pequeños para efectos o cuando trabajamos para otros medios de soporte, como un CD multimedia.

La ventana de ajustes cuenta con una vista previa de entrada y salida, con su respectiva línea de tiempo compuesta.



SOBRE STREAMING

La tecnología de streaming permite visualizar material en Internet sin necesidad de descargarlo a la computadora del usuario. La palabra streaming se refiere a la acción de flujo continuo de datos sin interrupción. Esto significa que el usuario puede escuchar un audio o ver un video en tiempo real. Este tipo de tecnología trabaja almacenando en un búfer lo que se va escuchando o viendo.

IMPORTAR VIDEO

Otro proceso implica importar un video como material de dispositivo móvil para SWF. Este método es muy parecido al anterior, con la diferencia de que el SWF está compilado en **Flash Lite**. Es el mecanismo adecuado cuando queremos mostrar video en un dispositivo móvil, como un celular o un smartphone. Al igual que el ejemplo anterior, se considera que el peso final del SWF es superior en comparación a la carga de videos por URL, ya que el material es incrustado en el archivo. El objetivo de nuestro proyecto determinará cuál de los tres métodos será el adecuado. En general, es bueno separar el video del SWF, en archivos diferentes, por lo que la primera opción es la más recomendada. Esta práctica nos dará independencia sobre los archivos y nos permitirá modificar

el ActionScript, el SWF o el video sin afectar a los demás. La ventana de diálogo correspondiente también cuenta con la opción de iniciar Adobe Media Encoder para hacer ajustes de bitrate y tamaño, y hasta para cortar la película. En algunos casos, es conveniente efectuar los ajustes necesarios para lograr un buen peso y una calidad acorde al proyecto que estamos desarrollando. Es normal que el proveedor de video (es decir, el editor) nos envíe un formato diferente de FLV y que, en la mayoría de los casos, este sea mucho más pesado de lo requerido, en el sentido de que la calidad sea superior. Una calidad superior a la óptima no suele ser visible, y afecta directamente a la velocidad de descarga del material. ◀

El botón **[Metadatos]** puede editar la información incrustada en el video y, luego, procesarla desde ActionScript.

Opciones de exportación: Incrustar en el archivo de salida

Metadatos de origen

Reglas de conservación: Conservar todo

☒ Exportar pista maestra de voz y marcadores de secuencia

Metadatos de archivo de salida

Exportar plantilla: Todos los metadatos

Metadatos predefinidos que se pueden añadir al

- ▶ Dublin Core
- ▶ Básico
- ▶ Gestión de derechos
- ▶ Gestión de medios
- ▶ Paged-Text
- ▶ Medios dinámicos
- ▶ PDF de Adobe
- ▶ Photoshop
- ▶ RAW de cámara
- ▶ Esquema EXIF para las ...
- ▶ Esquema EXIF para las ...
- ▶ Esquema EXIF para las ...
- ▶ Guión
- ▶ Extensión de audio BW...
- ▶ DICOM
- ▶ SWF móvil

Aceptar

Cancelar

VIDEO

STREAMING

Años atrás, un video no podía ser visto hasta haberlo descargado por completo. Las tendencias y los recursos crearon un nuevo sistema para ver este material, llamado streaming "sin interrupción".

Cuando utilizamos sitios como YouTube, es habitual ver un video, adelantar y volver o cambiar a otro sin necesidad de esperar a que el archivo se descargue por completo. Esta capacidad de que el usuario pueda manipular del video se conoce como streaming.

El streaming puede ser presentado en dos maneras concretas:

► **Streaming vivo:** es una transmisión en vivo y en directo de un flujo de datos, durante la cual el usuario no puede adelantarse y, en la mayoría de los casos, tampoco retroceder. Es utilizada en canales de videoconferencias, como el caso de la herramienta Adobe Connect Now, o en nuevas tecnologías, como la televisión en vivo por Internet GoogleTV. Este tipo de streaming se realiza tomando la señal, codificándola y enviándola a un servidor con Adobe Media Server. Los clientes se conectan a él, y el servidor distribuye la señal en vivo y sincronizada

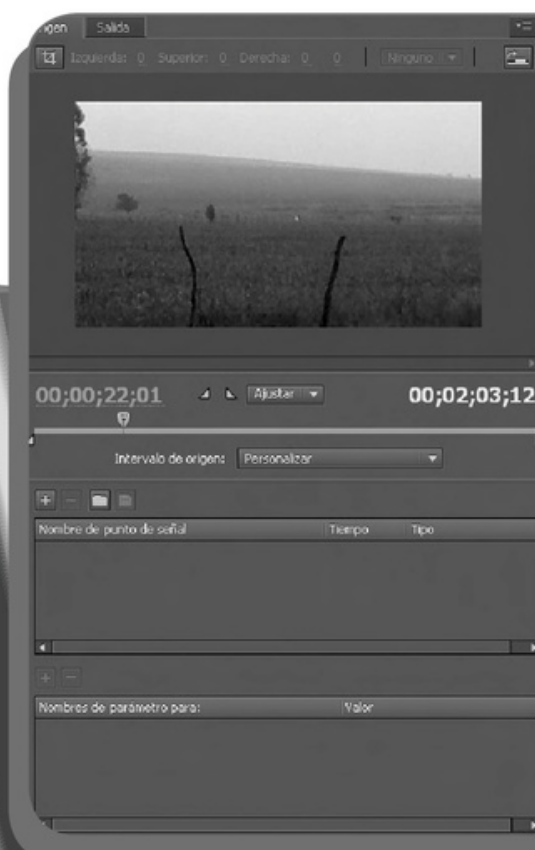
EL STREAMING PUEDE PRESENTARSE EN DOS MANERAS CONCRETAS: EN VIVO O EN DIFERIDO.

para cada uno de ellos. Este método implica, por el lado del servidor, un uso de ancho de banda considerable si la cantidad de usuarios es alta. Tengamos en cuenta que este tipo de streaming es bidireccional; por lo tanto, se pueden crear aplicaciones, como salas de chat, utilizando Adobe Media Server u otros servidores de streaming Open Source, como red5. Para este tipo de servicio se requiere un hosting donde alojar los programas del servidor o contratar un servicio de streaming, que es la opción más recomendada.

► **Streaming diferido o progresivo:** es importante destacar que muchas personas se refieren a este método con el nombre de "local", lo que, de alguna manera, resulta correcto, porque el archivo FLV es previamente vinculado localmente en una computadora y, luego, subido a un hosting junto a su SWF correspondiente. Esto quiere decir que el archivo de video no se crea en vivo, sino que está previamente codificado y disponible en su totalidad. Esta categoría no escapa a los términos de streaming, dado que la descarga se realiza gradualmente. Parte del video se almacena en un buffer o memoria, y luego el flujo de descarga puede ser

CÓDIGO 1

```
var miNC:NetConnection = new NetConnection();
miNC.connect(null);
var miNS:NetStream = new NetStream(miNC);
var miVideoPlayer:Video = miVideoPlayer;
miNS.addEventListener(AsyncErrorEvent.ASYNC_
ERROR,errorEvent);
function errorEvent(e:AsyncErrorEvent):void
{
}
//
miNS.play("videoclip.flv");
miVideoPlayer.attachNetStream(miNS);
```



reproducido, adelantado y hasta detenido. En definitiva, como no se requiere descargar todo el contenedor de un video, estamos hablando de streaming.

TRABAJAR CON VIDEOS

Teniendo en cuenta que no utilizamos un componente de video, si queremos manipular un video que está fuera de nuestro SWF de manera progresiva, tendremos que seguir el proceso detallado a continuación.

El material que vamos a vincular con nuestro SWF requiere un contenedor de tipo video inicialmente. Éste se crea desde la biblioteca, haciendo clic derecho del mouse sobre el panel y seleccionando **[Nuevo video]**. Se abrirá una ventana de diálogo con dos opciones principales:

[Incorporar video en línea de tiempo] o **[Video controlado por ActionScript]**.

En este caso, vamos a elegir la segunda opción y a hacer clic en **[Aceptar]**.

Veremos que nuestra librería tiene un elemento de video con el icono de una filmadora; lo arrastramos al escenario y, luego, le damos un nombre de instancia, como "miVideoPlayer". Guardamos el archivo FLA y, en esa misma ubicación, colocamos el video por reproducir. Es importante recordar que, para reproducir un video utilizando el método de

PRECARGA O BUFFER

Se trata de un componente clave para el streaming. En muchas ocasiones, el flujo de datos que va desde el servidor hacia la computadora del usuario puede estar interrumpido por el lag o retraso. Este fenómeno es ocasionado cuando los datos escasean debido a interrupciones en la conexión o a sobrecarga en el ancho de banda. Para solucionar este problema, los reproductores multimedia precargan o almacenan en el buffer la información que van recibiendo, para disponer de una reserva de datos y así evitar que la reproducción se interrumpa.

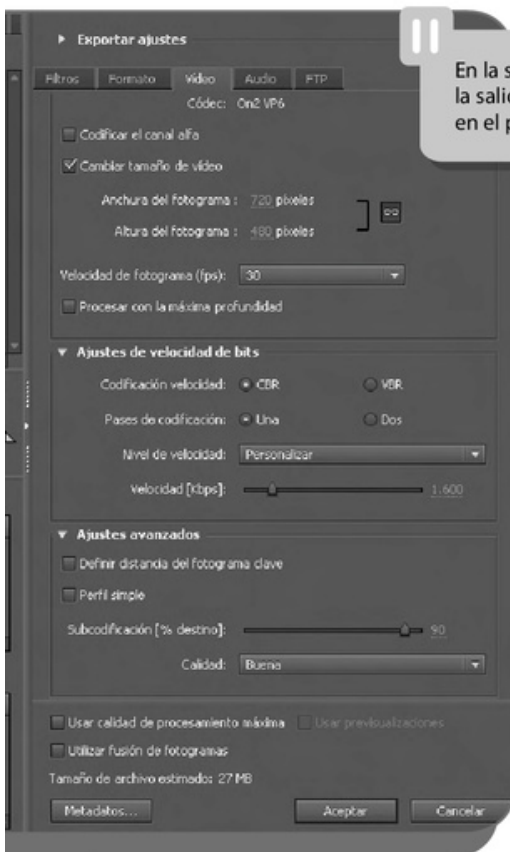
ActionScript, Flash requiere de varias clases que trabajen en forma conjunta. Por ejemplo, **NetStream** se encargará de cargar el video y controlar la reproducción; **Video** lo renderizará visualmente (en este caso, cuando creamos un objeto Video en la librería, se hereda de la clase **Flash.Media.Video**); mientras que **NetConnection** gestiona el tipo de origen del material (en él indicamos si el flujo de datos es progresivo o en vivo).

En el fotograma principal escribimos el **Código 1**.

Este código reproduce un video en formato FLV, pero también podría ser un MP4 o MOV. Se establece un objeto **NetConnection** (miNC); y, luego, se le indica null al método **connect** para informar que el video no es desde un flujo en vivo como **Flash Media Server**, sino de descarga progresiva. Después se crea un objeto **NetStream** y se lo vincula con el flujo de datos. Observemos que el objeto **Video** se genera pasando el nombre de instancia del contenedor que pusimos en el stage. Además, se agregó un escuchador a **NetStream** (**AsyncErrorEvent**) para evitar mensajes emergentes cuando haya errores de datos asíncronicos.

Al método **play** de **NetStream** se le pasa un string o cadena de texto con el nombre del video que vamos a reproducir (**videoclip.flv**). Finalmente, se relaciona el flujo de datos con el visualizador **miVideoPlayer**. El resto de las funcionalidades –como pausa, stop y reanudar– se controlan utilizando métodos de la clase **NetStream**. Claramente, se nota que cada clase controla una tarea particular de forma independiente. La clase **Video** es reservada para la parte visual y decorativa. ◀

En la solapa **[Video]** podemos ajustar la salida y Kbps. Estos valores influyen en el peso final del video.



SITIOS EXPERIMENTALES

CON VIDEO

En la actualidad, las redes sociales han adoptado la tecnología de streaming para publicar video. Esta posibilidad se debe al avance tecnológico aplicado al ancho de banda de Internet.

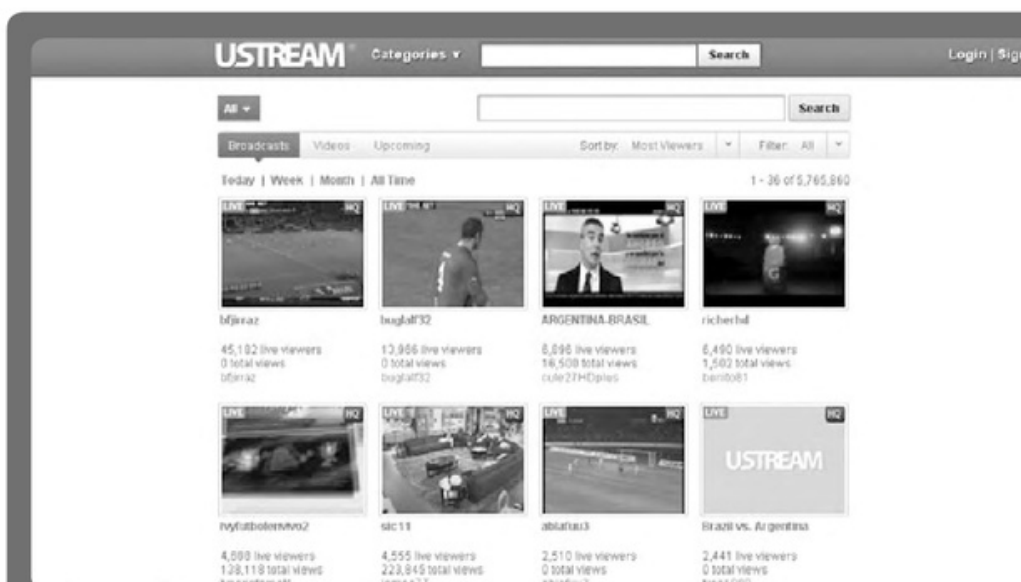
El video es uno de los medios más versátiles para transmitir datos, mensajes y sensaciones. Actualmente, contamos con un ancho de banda de alta calidad, lo cual incrementa en forma sustancial la aplicación de proyectos, pruebas y experimentaciones en video en la Web. A continuación, detallaremos algunos sitios experimentales muy interesantes.

WWW.GETTHEGLASS.COM

En este sitio se muestra un juego interactivo donde la secuencia es lineal, mediante videos en formato FLV. Podemos observar que los ambientes están generados en películas separadas, y son llamadas según la situación que se genere. El video se presenta en descarga progresiva, es decir, a medida que vamos avanzando o lo solicitamos. La programación y la sincronización del material están cuidadosamente organizadas mediante eventos simples, como escuchadores, cuando el video finaliza y ejecuta la siguiente acción. Un detalle importante que se puede apreciar es el uso de diferentes calidades de video según la información que se esté mostrando. La presentación del ambiente en general cuenta con un detalle mucho más grande que los efectos de transición entre una escena y otra. Este tipo de interactividad no es muy novedosa, pero es rara de ver, ya que requiere un trabajo arduo en la preparación de todas las escenas posibles que puedan existir.

WWW.USTREAM.TV

Este sitio ofrece video streaming en vivo a diferentes usuarios en simultáneo, utilizando una plataforma propia. Esta técnica es muy popular para programas o videoconferencias donde se requiere de un servicio rápido y con buena calidad.



LA TECNOLOGÍA STREAMING ES UNA DE LAS GRANDES PROTAGONISTAS DE LA EXPANSIÓN DE INTERNET COMO FENÓMENO.

WWW.HBO.COM

En la Web de la popular señal de TV HBO se utiliza video on demand para decorar el diseño. Podemos observar que, al ser un canal de televisión, se ha explotado el video como primer recurso de diseño, y a partir de él se muestra el resto de los contenidos. En este caso, el video cobra un protagonismo importante y, por eso, el peso es mayor que en otras situaciones.

WATERLIFE.NFB.CA

Este es un excelente sitio, en el cual se muestra claramente cómo un video puede enriquecer el diseño y acentuar el mensaje transmitido por las palabras y las imágenes. Cuenta con otros recursos creativos, como el uso de librerías de PaperVision 3D. También podremos apreciar que el audio no acompaña a los videos, pero sí hay un sonido ambiente general para crear la atmósfera adecuada.

MOODSTREAM. GETTYIMAGES.COM

Sitio experimental donde se cargan distintos elementos multimedia en forma aleatoria, según un criterio. La característica más destacable son los videos en descarga progresiva; asimismo, notaremos que el buffer de carga no se hace presente, ya que los videos están

+ DE FLASH A HTML 5

Si bien Flash sigue siendo la opción más utilizada, el volumen de los videos colgados en las principales redes sociales disponibles en HTML5 se ha duplicado en los últimos cinco meses. Este avance radical estaría vinculado a las aplicaciones de los dispositivos móviles. Por su parte, los editores del propio contenido en la Web han contribuido a inclinar la balanza a favor de HTML5. Es importante destacar que, si bien Flash sigue siendo el más usado, muchos sitios ofrecen la posibilidad de cambiar la plataforma de reproducción del dispositivo desde el que accedan los usuarios.

www.gettheglass.com muestra un juego interactivo donde la secuencia es lineal, mediante videos en formato FLV.

www.ustream.tv ofrece video streaming en vivo a diferentes usuarios en simultáneo.

En **www.moodstream.gettyimages.com** se cargan elementos multimedia en forma aleatoria.

Player, que tanto uso y provecho sacó. Los usuarios que tienen navegadores compatibles con HTML5 ya pueden ver videos en este formato.

VIMEO.COM

Los usuarios básicos tienen permiso para subir hasta 500 MB a este sitio. La interfaz de Vimeo simplifica la navegación, y da la posibilidad de crear y moderar grupos por categorías. Admite video en alta definición, carga de material de más de 1 GB para las cuentas Premium, foros y una base de usuarios/artistas. Dado que Vimeo se usa casi siempre para subir archivos de video de gran tamaño, la conversión de uno solo de ellos puede demandar bastante tiempo.

HULU.COM

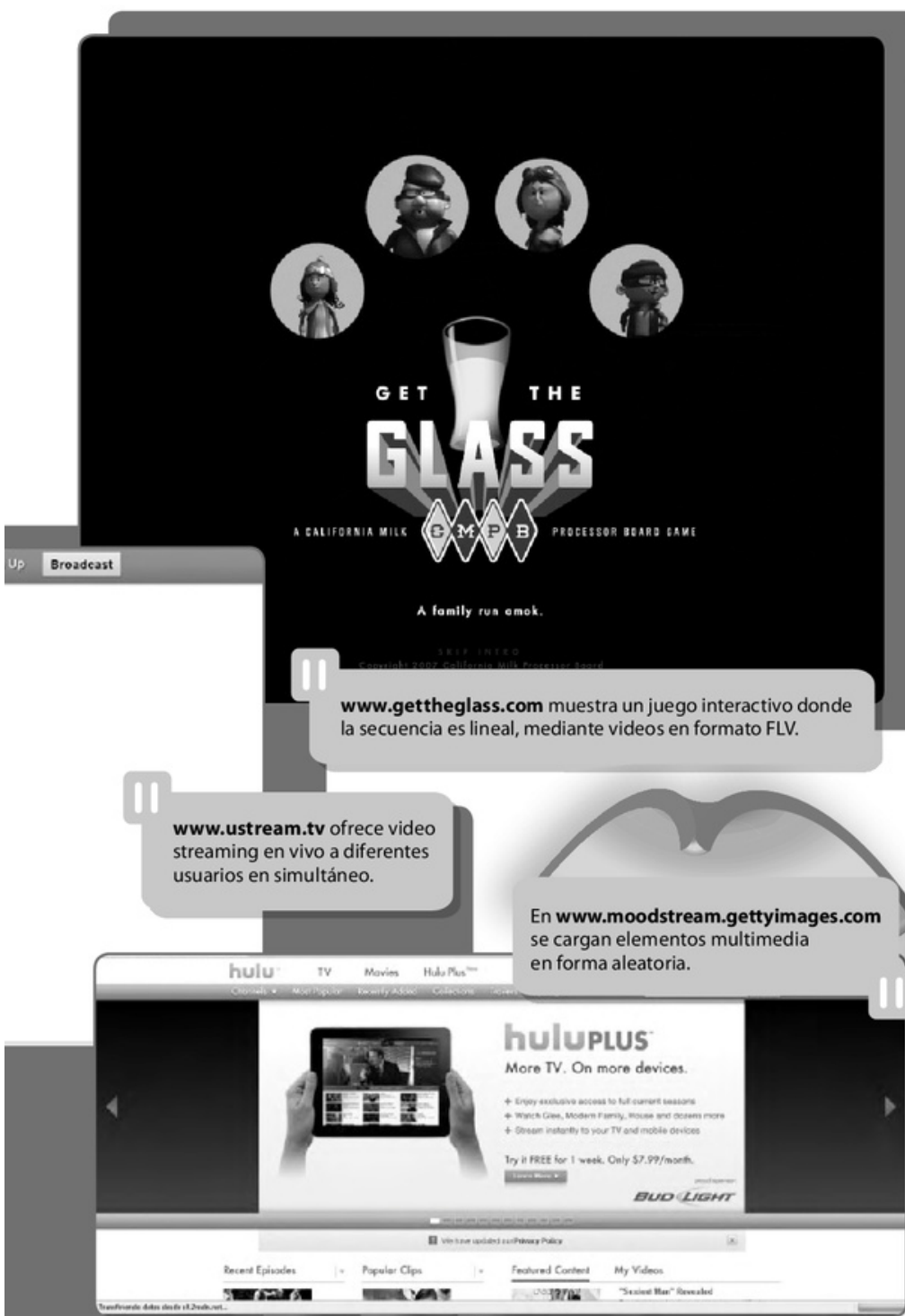
Hulu utiliza videos en 360 p para conexiones lentas a Internet. Permite la visualización de programas de TV, series y películas. El sitio es gratuito, y los videos pueden descargarse. Vale aclarar que los videos no pueden incrustarse en blogs o sitios web, y la mayoría de los que están en streaming no se pueden ver en alta definición. Es importante mencionar que los sitios experimentales vistos en este apartado son solo algunos de los más interesantes. En Internet encontraremos un abanico de posibilidades para experimentar sobre el video en términos de streaming. ◀

exhaustivamente optimizados para generar el mejor rendimiento, aunque se van descargando por adelantado de modo que, cuando se visualicen, estén disponibles.

YOUTUBE.COM

No podemos dejar de lado el líder en videos en la Web. La influencia que ha tenido el uso masivo de este sitio impulsó a desarrollar nuevas tecnologías para que los videos sean más rápidos de descargar, con mejor calidad y alta disponibi-

lidad. Desde sus pasos iniciales, este sitio optó por Flash Video, lo cual hizo popular a Flash Player, de Adobe, al punto de convertirlo en un estándar para todos los navegadores. La incorporación de video en alta resolución impulsó el uso de códecs como h264; y, en la actualidad, contamos con videos de hasta 1080 p Full HD. La carrera de YouTube nunca cesó. En la actualidad, hay experimentaciones con video 3D, HD 4K (1080 x 4), y la gran sorpresa es querer dejar de lado Flash



ADOBE

FLASH MEDIA SERVER

En este apartado veremos qué sucede del lado del servidor con nuestros videos, para lo cual presentaremos Adobe Flash Media Server como solución para lograr un flujo de video y comunicación en tiempo real.

A lo largo de esta clase especialmente dedicada al uso y el trabajo con video en Flash y por medio de ActionScript 3, hemos visto que existen diferentes formas de manejar videos de acuerdo con las necesidades que tengamos. Además, conocimos los diversos formatos, sus características y de qué modo optimizarlos para conseguir una visualización más adecuada. Es por eso que en estas páginas nos dedicaremos al funcionamiento de los sitios web, pero desde el punto de vista del servidor en relación con nuestros videos. Para esto, detallaremos las características más importantes de Adobe Flash Media Server como solución para lograr un flujo óptimo de video y comunicación.

¿QUÉ ES Y PARA QUÉ SIRVE?

Adobe Flash Media Server es una familia de tres tipos de servidores web de contenidos multimedia que nos permiten realizar streaming en alta calidad de "contenidos en vivo", como puede ser la transmisión de una webcam, una cámara digital especial o TV digital, y también la transmisión de "contenidos ya grabados" y alojados en cualquiera de las versiones de estos servidores. A diferencia de otros servidores que utilizamos habitualmente –como Apache–, Adobe Flash Media Server nos proporciona una verdadera opción de streaming, ya que otros servidores, cuando alojamos archivos de video, no utilizan esta técnica, sino que la información se descarga en forma pro-

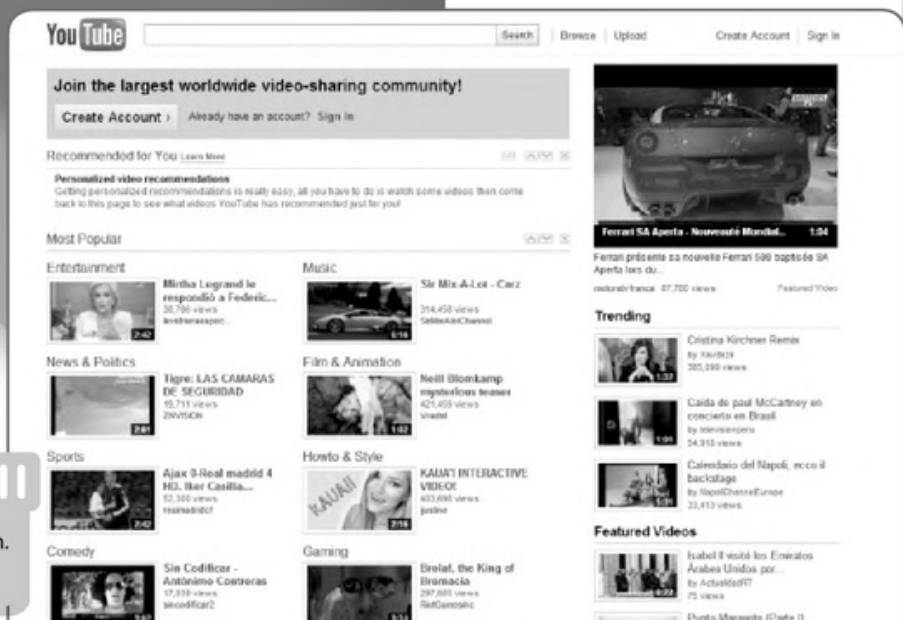
gresiva. Cada una de las versiones de esta familia tiene sus características y aplicaciones.

En primer lugar, está **Adobe Flash Media Streaming Server**, pensado para proyectos que requieran transmisiones por demanda de baja capacidad. Es ideal, por ejemplo, para sitios en los que se transmita video en vivo para una cantidad de usuarios conectados no muy grande, seminarios

online o, también, cámaras de seguridad. La siguiente versión, denominada **Adobe Flash Media Interactive Server**, ofrece funcionalidades más avanzadas, como multidifusión IP, para ahorrar ancho de banda; y HTTP Dynamic Streaming, que permite lograr más opciones de reproducción. También, con esta versión tendremos mayores posibilidades de acceso a otros dispositivos, como iPhones y móviles. Es perfecta para proyectos en los cuales será necesaria una óptima experiencia multiusuario. Por último, nos encontramos con la versión más completa de la familia, **Adobe Flash Media Enterprise Server**, que proporciona excelentes ventajas y mayores características que las anteriores. Está especialmente pensada para proyectos realmente grandes y ambiciosos, en los cuales puedan intervenir usuarios en forma masiva, como ocurre con las redes sociales.

ADOBE FLASH MEDIA SERVER
ES UNA FAMILIA DE
SERVIDORES WEB
QUE PERMITE REALIZAR
STREAMING EN ALTA CALIDAD.

YouTube, el sitio de video más importante, utiliza una plataforma Flash. Debemos tomarlo como referencia.



CARACTERÍSTICAS PRINCIPALES

Adobe Flash Media Server cuenta con un gran número de características; veamos las más importantes. Es compatible con la arquitectura de 64 bits, lo que nos permitirá instalarlo en una amplia variedad de plataformas, incluidas las de 64 bits. De este modo, podremos aprovechar más recursos. Cuenta con **Multidifusión IP**, que permite el envío simultáneo de información a diferentes destinos, determinando cuál es la estrategia más eficiente para cada enlace de la red que solicite contenido. En otras palabras, controla el flujo de transmisión de cada usuario que esté requiriendo la recepción de contenido multimedia, ya sea audio o video. Por otra parte, **HTTP Dynamic Streaming** proporciona la ventaja de poder realizar una transmisión de flujo HTTP de contenido multimedia integrado con Flash Player. De la misma manera en que recibimos, por ejemplo, un video por medio de HTTP, HTTP Dynamic Streaming utiliza idéntica tecnología, pero agrega las siguientes cualidades:

► **Compatible con varias velocidades de bits:** permite cambiar la calidad del video, dependiendo del ancho

de banda y de la capacidad de hardware de la computadora.

► **Se puede buscar en el video:**

esta es una de las particularidades más destacadas, porque nos da la posibilidad de ver un video en una posición determinada sin necesidad de descargar todo el contenido. Podemos ver un ejemplo en www.megavideo.com.

► **Control de ancho de banda:** permite variar el ancho de banda de descarga, de modo que no se sobrecarguen otras aplicaciones.

► **Compatible con transmisiones en directo:**

no hace falta explicar esta característica, porque, como su nombre lo indica, implica la posibilidad de crear transmisiones de contenido multimedia en vivo y en directo. Podemos encontrar muchos sitios que la aplican, por ejemplo, para hacer TV o radio digital. Un ejemplo es el sitio argentino www.tvpts.tv.

Es importante destacar que Adobe Flash Media Server es utilizado en algunos de los sitios de visualización de video más importantes de la Red, como www.youtube.com y www.immigrantarchiveproject.com, entre muchos otros. ◀

En este sitio podemos buscar alguna parte específica del video sin necesidad de descargar todo el material.

Site que permite ver y subir videos a inmigrantes, contando sus experiencias. (www.immigrantarchiveproject.com).

Este es un sitio que emite en directo utilizando Flash Media Server (www.tvpts.tv).



LOOP DE VIDEO EN EL

PASO A PASO

En esta práctica aprenderemos una técnica de alto impacto para dejar una impronta interesante en nuestras páginas web.



01 CREAR HTML

```

8 <body>
9 <body>
10 <div id="flashContent">
11 <object classid="clsid:d27c2b6e-ae6d-11cf-96b8-444553540000" width="100%" id="loopvideo" align="center">
12 <param name="movie" value="loopvideo.swf"/>
13 <param name="quality" value="high"/>
14 <param name="bgcolor" value="#ffffff"/>
15 <param name="play" value="true"/>
16 <param name="loop" value="true"/>
17 <param name="wmode" value="window"/>
18 <param name="scale" value="noscale"/>
19 <param name="menu" value="false"/>
20 <param name="devicefont" value="false"/>
21 <param name="align" value="center"/>
22 <param name="allowScriptAccess" value="sameDomain"/>
23 <param name="allowFullScreen" value="true"/>
24 <!--[if IE] -->
25 <object type="application/x-shockwave-flash" data="loopvideo.swf" width="100%" height="100%">
26 <param name="movie" value="loopvideo.swf"/>
27 <param name="quality" value="high"/>
28 <param name="bgcolor" value="#ffffff"/>
29 <param name="play" value="true"/>
30 <param name="loop" value="true"/>
31 <param name="wmode" value="window"/>
32 <param name="scale" value="noscale"/>
33 <param name="menu" value="false"/>

```

EDITAR HTML

02

Si queremos llevar adelante esta solución más que interesante para nuestra web, es preciso tener en claro que hay dos aspectos por considerar. El primero es la configuración del archivo HTML que embebe nuestro SWF; y el segundo, obviamente, el SWF. Teniendo estos dos aspectos bien presentes, veamos en detalle cómo realizar este proceso. ▶

PASO | 01 Lo primero que debemos hacer es generar el archivo HTML desde el panel de publicación. Es importante recordar que los valores de alto y ancho deben estar al 100%, y que la opción **[Escala]** tiene que estar seleccionada.

PASO | 02 Luego de haber configurado el HTML con los valores de ancho y de alto adecuados, debemos continuar con la segunda acción de este proceso. Es decir, editamos el HTML y agregamos esta línea al final de los dos **object** creados por Flash:

```
<param name="allowFullScreen" value="true" />
```

PASO | 03 El tercer paso en este proceso tiene que ver con agregar en Flash las dos líneas habituales para una Web full screen. El código adecuado se muestra a continuación:

FONDO

04 FUNCIONES

```

1 stage.scaleMode = StageScaleMode.NO_SCALE;
2 stage.align = StageAlign.TOP_LEFT;
3
4 var _netStr:NetStream = null;
5 var _netCon:NetConnection = null;
6 var _video:Video = null;
7
8 function SetupNetConnection()
9 {
10 }
11 function SetupNetStream()
12 {
13 }
14 function onMetaData(info:Object):void
15 {
16 }
17 function NetStatus(e:NetStatusEvent)
18 {
19 }
20 function Resize(e:Event):void
21 {
22 }
23

```

Acciones - fotograma

```

1 stage.scaleMode = StageScaleMode.NO_SCALE;
2 stage.align = StageAlign.TOP_LEFT;
3
4 var _netStr:NetStream = null;
5 var _netCon:NetConnection = null;
6 var _video:Video = null;
7
8

```

03 FULL SCREEN

AGREGAR BOTÓN | 05

```

2 stage.align = StageAlign.TOP_LEFT;
3
4 var _netStr:NetStream = null;
5 var _netCon:NetConnection = null;
6 var _video:Video = null;
7
8 function SetupNetConnection()
9 {
10 }
11 function SetupNetStream()
12 {
13 }
14 function onMetaData(info:Object):void
15 {
16 }
17 function NetStatus(e:NetStatusEvent)
18 {
19 }
20 function Resize(e:Event):void
21 {
22     FullScreenButton.x = 10;
23     FullScreenButton.y = stage.stageHeight - (FullScreenButton.height + 10);
24 }
25
26 stage.addEventListener(Event.RESIZE, Resize, false, 0, true);
27 SetupNetConnection();
28
29 function FullScreenButtonHandler(event:MouseEvent)
30 {
31     stage.stageDisplayState = StageDisplayState.FULL_SCREEN;
32 }
33 FullScreenButton.addEventListener(MouseEvent.CLICK, FullScreenButtonHandler);
34 addChild(FullScreenButton);
35
36 Resize(null);
37 SetupNetStream();
38

```

```

3
4 var _netStr:NetStream = null;
5 var _netCon:NetConnection = null;
6 var _video:Video = null;
7
8 function SetupNetConnection()
9 {
10     _netCon = new NetConnection();
11     _netCon.addEventListener(NetStatusEvent.NET_STATUS, NetStatus, false, 0, true);
12     _netCon.connect(null);
13     trace(_netCon);
14 }
15
16 function SetupNetStream()
17 {
18 }
19 function onMetaData(info:Object):void
20 {
21 }
22 function NetStatus(e:NetStatusEvent)
23 {
24     trace(e.info.code);
25 }
26
27 function Resize(e:Event):void
28 {
29     FullScreenButton.x = 10;
30 }
31

```

INICIALIZAR CLASE | 06

```

stage.scaleMode = StageScaleMode.NO_SCALE;
stage.align = StageAlign.TOP_LEFT;
seguido declaramos nuestras primeras
variables. var _netStr:NetStream = null;
var _netCon:NetConnection = null;
var _video:Video = null;

```

PASO | 04 Ahora vamos a declarar las funciones que contendrá nuestro proyecto por medio del siguiente código:

```

function SetupNetConnection(){}
function SetupNetStream(){}
function onMetaData(info:Object):void {}
function NetStatus(e:NetStatusEvent){}
function Resize(e:Event):void {}

```

PASO | 05 En este quinto paso, agregamos un botón que se ocupará de la opción de full screen. En nuestro layout encontraremos un ejemplo de esto.

PASO | 06 El próximo paso es crear e inicializar la clase NetConnection para lograr la reproducción del archivo de video. La función quedaría como se muestra a continuación:

```

function SetupNetConnection()
{
    _netCon = new NetConnection();
    _netCon.
    addEventListener(NetStatusEvent.NET_STATUS,
    NetStatus, false, 0, true);
    _netCon.connect(null);
}

```


LOOP DE VIDEO EN EL FONDO

PASO A PASO

```

5 var _netCon:NetConnection = null;
6 var _video:Video = null;
7
8 function SetupNetConnection() {
9     _netCon = new NetConnection();
10    _netCon.addEventListener(NetStatusEvent.NET_STATUS, NetStatus, false, 0);
11    trace(_netCon);
12    _netCon.connect(null);
13    _netCon.addEventListener(NetStatusEvent.NET_STATUS, NetStatus, false, 0);
14}
15
16 function SetupNetStream() {
17    var customClient:Object = new Object();
18    customClient.onMetaData = onMetaData;
19
20    _netStr = new NetStream(_netCon);
21    _netStr.client = customClient;
22    _netStr.bufferTime = 3;
23    _netStr.addEventListener(NetStatusEvent.NET_STATUS, NetStatus, false, 0);
24}
25
26 function onMetaData(info:Object):void {}
27 function NetStatus(e:NetStatusEvent) {
28    switch (e.info.code) {
29        case "NetConnection.Connect.Success":
30            SetupNetStream();
31            break;
32    }
33}

```

07 AGREGAR LLAMADA



AGREGAR SWITCH | 08

```

12 trace(_netCon);
13 }
14 function SetupNetStream() {
15     var customClient:Object = new Object();
16     customClient.onMetaData = onMetaData;
17
18     _netStr = new NetStream(_netCon);
19     _netStr.client = customClient;
20     _netStr.bufferTime = 3;
21     _netStr.addEventListener(NetStatusEvent.NET_STATUS, NetStatus, false, 0);
22 }
23
24 _video = new Video();
25 _video.attachNetStream(_netStr);
26 _video.smoothing = true;
27 addChild(_video);
28 _netStr.play("video.flv");
29 }
30
31 function onMetaData(info:Object):void {}

```

09 CARGA DEL FLV

```

17 function onMetaData(info:Object):void {}
18 function NetStatus(e:NetStatusEvent) {
19     switch (e.info.code) {
20     {
21         case "NetConnection.Connect.Success":
22             SetupNetStream();
23             break;
24         case "NetStream.Play.Stop":
25             _netStr.seek(0);
26     }
27 }

```

PASO | 07 Ahora vamos a agregar una llamada a la misma clase desde el final del código. Si el proceso se realizó correctamente,

la función **NetStatus** debería devolver algo como **NetConnection.Connect.Success**. Esto puede probarse poniendo en la función un **trace**.

PASO | 08 Una vez que recibimos la confirmación de la conexión, vamos a agregar un switch en la función **NetStatus**. Básicamente, la idea es que, luego de realizar la conexión, llamemos a la función **SetupNetStream** para que esta se ocupe de cargar el archivo de video. Por lo tanto, reemplazamos el actual **trace** por el código que se ve en la imagen. En este caso, si el código recibido es del tipo **Success**, ejecutará la función **SetupNetStream**; de lo contrario, iniciará el video.

PASO | 09 Este paso es el más importante, porque en él realizaremos la carga del archivo FLV. Dentro de la función

SetupNetStream ingresaremos el código que se ve en la imagen. Básicamente, lo que estamos haciendo es inicializar una instancia de la clase **NetStream**, la cual tendrá como parámetro la instancia de la conexión que ya habíamos logrado (**_netCon**). En otras palabras, **NetStream** utilizará a **NetConnection** para solicitar el archivo de video y mostrarlo a medida que se vaya cargando. Cuando configuramos un **bufferTime** de 3, lo que hacemos es solicitar que se almacenen 3 segundos de video en el buffer y, de esta manera, evitar que el video se vea cortado.

PASO | 10 Hasta ahora, no logramos ver nada, y es porque nos falta algo muy importante: inicializar y volcar el componente de video en el escenario. Esto se realiza debajo de la línea **_netStr.addEventListener**. Creamos el componente de video, le

```

13     }
14     function SetupNetStream(){
15         var customClient:Object = new Object();
16         customClient.onMetaData = onMetaData;
17
18         _netStr = new NetStream(_netCon);
19         _netStr.client = customClient;
20         _netStr.bufferTime = 3;
21         _netStr.addEventListener(NetStatusEvent.NET_STA
22
23         _video = new Video();
24         _video.attachNetStream(_netStr);
25         _video.smoothing = true;
26         addChild(_video);
27         _netStr.play("video.flv");
28     }

```

10 | VOLCAR COMPONENTE DE VIDEO

```

38     }
39 }
40 function Resize(e:Event):void
41 {
42     var proportion:Number;
43     if (stage.stageHeight / stage.stageWidth > _video.height / _video.width)
44     {
45         proportion = _video.width / _video.height;
46         _video.height = stage.stageHeight;
47         _video.width = stage.stageHeight * proportion;
48     } else {
49         proportion = _video.height / _video.width;
50         _video.width = stage.stageWidth;
51         _video.height = stage.stageWidth * proportion;
52     }
53     fullScreenBoton.x = 10;
54     fullScreenBoton.y = stage.stageHeight - (fullScreenBoton.height + 10);
55 }
56 stage.addEventListener(Event.RESIZE, Resize, false, 0, true);
57 SetupNetConnection();
58
59 function fullScreenBotonHandler(event:MouseEvent)

```

11 | PROPORCIONES



VERIFICACIÓN DEL PROYECTO

12



MISMA TÉCNICA, OTROS EJEMPLOS

13

asignamos el stream con el cual va a trabajar, le pedimos que le dé un suavizado al video, lo volcamos al escenario y le damos **[Play]**.

PASO | 11 Ya conseguimos ver el video, pero está pequeño. Lo que tenemos que hacer ahora es agrandarlo proporcionalmente para que cubra la pantalla por completo. Entonces, dentro de la función **[Resize]** agregamos en el código de la imagen.

PASO | 12 Si hemos realizado correctamente los procesos anteriores y redactado cada uno de los códigos del modo adecuado, solo nos quedaría, para finalizar con este apartado, ejecutar el archivo HTML y verificar el proyecto completo.

PASO | 13 Luego de verificar que todo funcione correctamente, podemos observar algunos de los ejemplos de lo que

podemos hacer con esta técnica en:

www.armanisamsungmobile.com. En este caso, cuando entremos en el interior de algunos de los modelos, podremos avanzar y retroceder un poco el video de fondo para ver cómo la cámara rota unos metros. ◀

RESUMEN

Es interesante descubrir este método para dar un valor agregado a nuestras páginas web y, especialmente, a su background. De esta forma, algo que siempre fue estático empieza a cobrar protagonismo en nuestro diseño.

ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué ventajas ofrece utilizar video en la web?

02 ¿Qué formatos de video podemos utilizar?

03 Mencione algunas ventajas y desventajas de diversos formatos de video.

04 ¿Para qué sirve Adobe Media Encoder?

05 ¿De qué forma debemos incorporar video a una película de Flash?

06 ¿Qué es video streaming?

07 Mencione las ventajas del video streaming.

08 ¿Qué tipos de streaming podemos utilizar?

09 ¿Qué es Adobe Flash Media Server?

10 ¿Qué es un loop de video?

PRÁCTICAS

01 Incorpore un video en su proyecto.

02 Utilice Adobe Media Encoder.

03 Experimente con video en streaming.

04 Visite algunos sitios web que ofrezcan video.

05 Aplique un loop de video en el fondo.

APIs externas y complementos

- ▶ ¿QUÉ SON LAS APIs?
- ▶ YOUTUBE API
- ▶ GOOGLE MAPS
- ▶ CÓMO CARGAR COORDENADAS
- ▶ PAPERVISION 3D
- ▶ FACEBOOK API
- ▶ ADOBE CS5 FLASH CATALYST
- ▶ UTILIZAR FLASH CATALYST
- ▶ SWFOBJECT
- ▶ PERSONALIZAR MENÚ CONTEXTUAL
- ▶ CÓMO PROTEGER FLASH
- ▶ AGREGAR ELEMENTOS INTERACTIVOS
- ▶ ARCHIVOS TEMPORALES
- ▶ CÓMO APLICAR EFECTOS DE PARTÍCULAS
- ▶ GOOGLE ANALYTICS PARA FLASH



¿QUÉ SON LAS APIs?

Para definir una API, debemos comprender que el acceso a un sistema se determina por las opciones que este ofrece en virtud de otros lenguajes de programación y sistemas.

Quienes trabajan frecuentemente en desarrollo web suelen hablar acerca de la API para uno u otro servicio, pero rara vez se explica claramente el significado de este término, que pasa a ser un "sobrentendido" que no siempre es tal.

API (*Application Programming Interface*), que podríamos traducir como interfaz de programación entre aplicaciones, hace referencia a ciertos servicios y librerías que permiten acceder a procesos entre componentes de software. Dicho de otra forma, es una manera de abstracción que ofrece acceso a un conjunto de funciones de uso general, para así evitar que haya que volver a programar todo desde cero cada vez que se utiliza.

En términos de aplicaciones web (como Facebook o Google Maps), cuando nos referimos a una API, por lo general estamos hablando de bibliotecas que ofrecen funcionalidades de acceso a los diferentes servicios. Por ejemplo, existe

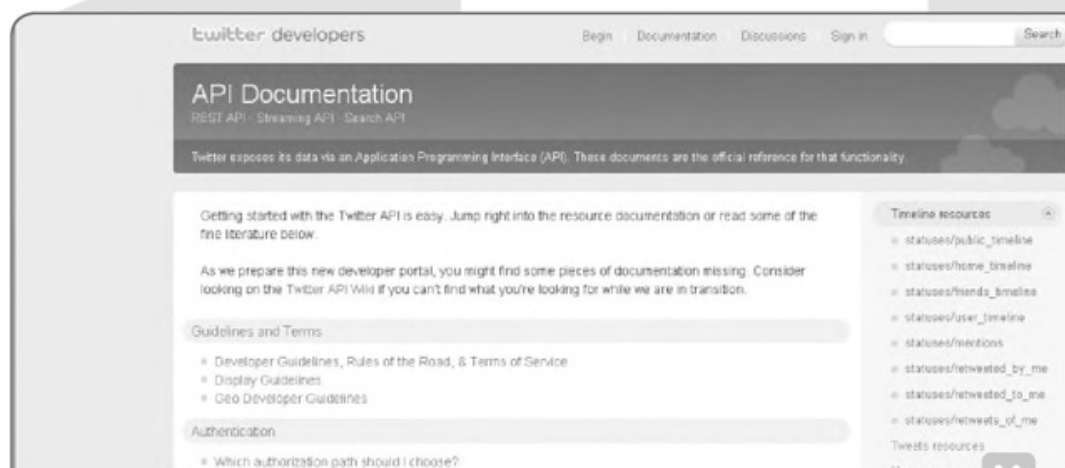
CÓDIGO 1

```
import com.greensock.*;
import com.greensock.easing.*;
//movemos el clip mc 300 px a la derecha en 3 segundos
TweenLite.to(mc, 3, {x:mc.x+300, ease:Quad.easeInOut,
onComplete:reverse});
function reverse(){
    TweenLite.to(mc, 3, {x:mc.x-300, ease:Quad.easeInOut});
}
```

una API de Facebook destinada a realizar operaciones, como publicar en el muro o enviar un mensaje directamente desde Flash; y otra en Google para manipular un mapa, moverlo y agrandarlo sin necesidad de conocer a fondo cómo funciona el motor de Google Maps.

En general, todas las aplicaciones web que lograron cierto éxito han publicado alguna API para operar con su servicio.

Muchas de estas librerías están orientadas a cierto lenguaje de programación. En la **Tabla 1** podemos ver un listado de las APIs dedicadas a ActionScript. La mayoría de los sitios web comenzaron con APIs simples, llamadas HTTP, con mensajes cortos en XML. Esta forma de propósito general es menos pesada que usar Web Services, basados en el



En Twitter Developer se concentran todas las tecnologías que pueden usar la API de Twitter.



protocolo SOAP (*Simple Object Access Protocol*), el cual es una variante del XML y se utiliza en sistemas de gran complejidad. De este modo, con una simple llamada a una URL, recibimos un mensaje con la respuesta, que es sencillo de entender y utilizar. A medida que la oferta de funcionalidad se fue agrandando, las librerías específicas de cada lenguaje aparecieron para simplificar la tarea de estudiar cada URL y el tipo de mensaje. Si revisamos librerías de uso interno en Flash (es decir, aquellas no relacionadas con sitios web), podemos nombrar muchas otras:

- ▶ **as3corelib:** utilidades varias para exportar a JPG o criptografía (<http://github.com/mikechambers/as3corelib>).
- ▶ **APE (Actionscript Physics Engine):** motor de física (www.cove.org/apel).
- ▶ **as3soundeditorlib:** librería para edición de sonido en ActionScript 3.0 (<http://code.google.com/p/as3soundeditorlib>).
- ▶ **as3ds:** estructuras de datos para desarrolladores de juegos (<http://code.google.com/p/as3ds>).
- ▶ **As3Crypto:** librería de criptografía ActionScript 3 (<http://crypto.hurlant.com>).
- ▶ **asmotion:** librería de animación para AS3 (<http://code.google.com/p/asmotion>).
- ▶ **Away3d:** librería de 3D (<http://code.google.com/p/away3d>).
- ▶ **FZip:** librería de ActionScript 3 Class para el tratamiento de archivos ZIP (<http://codeazur.com.br/lab/fzip>).
- ▶ **uicomponents-as3:** set de componentes de UI livianos (<http://code.google.com/p/uicomponents-as3>).
- ▶ **TweenMax/TweenLite:** librería de animación (www.greensock.com).



APIs

Tabla 1: Encontramos aquí un amplio listado con todas las APIs dedicadas a ActionScript.

▶ Facebook	▶ http://code.google.com/p/facebook-actionscript-api
▶ Google Maps	▶ www.afcomponents.com/components/g_map_as3
▶ Ebay http	▶ http://code.google.com/p/as3ebaylib
▶ mapQuest	▶ http://company.mapquest.com/mqbs/4a.html
▶ Yahoo	▶ http://developer.yahoo.com/flash/as3_api_libraries.html
▶ Amazon S3	▶ http://code.google.com/p/as3aws3lib
▶ YouTube	▶ http://code.google.com/p/as3youtubelib
▶ MySpace	▶ http://code.google.com/p/opensocial-actionscript-client

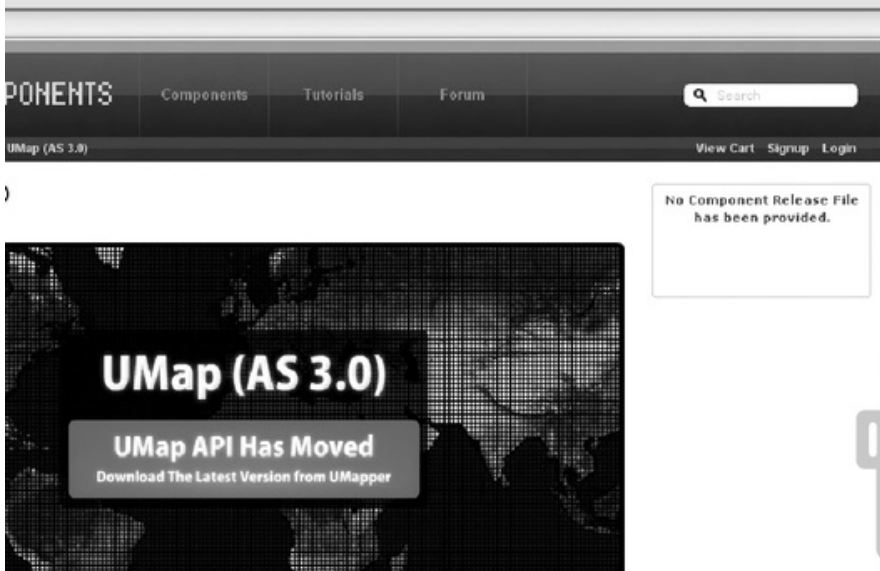
El uso de librerías para acceder a distintas funcionalidades es fundamental si queremos acelerar el proceso de trabajo, debido a que si cada desarrollador que programara funcionalidades debiera empezar desde cero, los procesos de producción serían incompatibles con un ciclo de innovación a la medida de la exigencia del mercado.

USO DE UNA LIBRERÍA INTERNA

En las próximas páginas de esta clase, veremos las librerías de acceso a servicios como YouTube o GoogleMaps; pero antes daremos un ejemplo de uso de librería interna, en donde, simplemente, aprovechamos las clases que se brindan para realizar una tarea común. Pongamos por caso que queremos

animar un movie clip que vaya de izquierda a derecha de la pantalla y que, cuando termine, haga el camino inverso. Usando una forma tradicional (modificar la posición **x** en un **onEnterFrame**), deberíamos verificar constantemente la posición para invertir la velocidad, y seguir haciéndolo para comprobar si ha regresado al lugar de origen. Gracias a la librería **Tweenlite**, podemos realizar este proceso con muy pocas líneas. En una película nueva, creamos un movie clip (un cuadrado, por ejemplo, estará bien) y le ponemos nombre de instancia **mc**; luego, usamos el **Código 1** en el primer y único frame.

Sin necesidad de recurrir a ningún **if** ni declarar un **onEnterFrame** para manejar la animación, logramos que el movie clip vaya de un lado a otro del escenario. En este código, podemos destacar la declaración de un callback (función que será llamada cuando la animación termine) que nos permite iniciar una segunda animación. Siguiendo este mismo principio, podríamos encadenar cuantas animaciones quisiéramos. De hecho, otra de las bibliotecas que podemos bajar del mismo sitio, llamada **TimelineLite**, da la posibilidad de agrupar secuencias de animación hechas con **TweenLite** para encadenar conjuntos de animaciones y, así, armar un todo más complejo. ◀



www.afcomponents.com. En este sitio las librerías están orientadas al lenguaje de programación Actionscript.

YOUTUBE API

A partir de la compra de YouTube por parte de Google, este popular portal o red social de video multiplicó su oferta de servicios. Tanto es así, que tuvo que adecuar sus capacidades técnicas a las nuevas demandas.

Seguramente, todos hemos oído hablar de YouTube, el servicio de video más popular en la actualidad. Se dice que en él es posible encontrar prácticamente cualquier material audiovisual que se busque, y tal vez no estemos muy alejados de la verdad, sobre todo, considerando la gran participación de los usuarios que han subido todo tipo de contenido durante años. Esto también ha planteado problemas de legalidad, porque no todo lo que se sube está libre de derechos, y muchas empresas de producción han entablado juicios para impedir la difusión de material con copyright. Pero a pesar de los conflictos legales, YouTube ha crecido sin cesar y constituye el servicio online de material en video más grande, disponible en gran cantidad de idiomas. La inmensa variedad de contenido acumulado le permite, incluso, crear canales temáticos y servicios por país, alimentados por los millones de usuarios que suben incesantemente sus videos. Esta gran disponibilidad planteó el interés de los desarrolladores por acceder a ese material y mostrar videos desde el servicio de YouTube, pero incrustados dentro de los propios sitios web. Casi desde el mismo inicio del servicio, existió la posibilidad de incluir el reproductor en cualquier HTML, con solo copiar y pegar un simple código, aunque la posibilidad de personalización de esta opción es reducida. Con el incremento del ancho de banda y la competencia de otros portales de video, la presión por incorporar nuevas funcionalidades fue aumentando. Sobre todo, los reclamos se localizaron en

Google code

por ejemplo, "guía java" o "reproductor sin bordes"

Herramientas y API de YouTube
Página principal
Documentación
Preguntas

¿Qué son las herramientas y el API de YouTube?



YouTube
Broadcast Yourself™

YouTube en cualquier pantalla, en cualquier momento. Las herramientas y API de YouTube permiten integrar las funciones y los videos de YouTube en tu sitio web, en tu aplicación o software, o en tu dispositivo.



API de datos

El API de datos permite incorporar las funciones de YouTube a tu propio sitio web o aplicación. Entre otras muchas cosas, podrás realizar búsquedas, subir videos o crear listas de reproducción.



API del reproductor

Las API del reproductor te permiten controlar la reproducción de los videos de YouTube en tu sitio web. Configura opciones básicas, controla la interfaz del reproductor e incluso crea tus propios controles del reproductor.



Reproductor personalizado

El reproductor personalizado te permite realizar más funciones además de pegar un video en tu sitio. Cualquiera podrá configurar fácilmente el reproductor personalizado para ver listas de reproducción, videos favoritos o sus propios videos.



Widgets

Los widgets son simples elementos de página que puedes insertar en tu sitio web para incorporar funciones de YouTube. Añade una tira de videos o permite a tus usuarios realizar búsquedas de videos con solo añadir un poco de código JavaScript a tu página.

poder utilizar un player que no estuviera tan claramente identificado con el portal, además del acceso a datos sobre videos. Fue así que YouTube (comprado por Google) ofreció el player **chromeless** con una API disponible para que los usuarios creen su propio set de controles.

UN PLAYER PERSONALIZABLE

Siempre existieron muchos trucos para cargar directamente contenido de YouTube sin usar el player; incluso, para cargar el player mismo dentro de otra película. Pero en tanto aparecían maneras de hacerlo, YouTube cambiaba el sistema para impedirlo. ¿Por qué evitar la difusión de material que, sin embargo, no tiene

copyright? En realidad, hay una mezcla de razones, y el factor económico tiene, sin duda, un gran peso.

La descarga de material audiovisual cuesta mucho dinero como para que cualquiera haga uso del video sin ningún beneficio para el que brinda el servicio. Así surgió el player chromeless, es decir, sin ningún tipo de controles, que puede ser cargado en otra película de Flash y usado con controles personalizados. Una lista de las funcionalidades que podemos invocar en el player son:

- ▶ **player.cueVideoById:** carga la miniatura del video especificado.
- ▶ **player.loadVideoById:** reproduce

YOUTUBE OFRECE EL PLAYER CHROMELESS CON UNA API DISPONIBLE PARA QUE LOS USUARIOS GENEREN SU SET DE CONTROLES.

Vemos cómo, mediante las APIs, otros sitios pueden incrustar videos referenciados en YouTube.

Las herramientas y las APIs de YouTube permiten integrar videos en los sitios web.

analogía de la receta de cocina para comparar los dos modelos de software (libre y propietario), nos enseñan qué es el Software Libre.

Con este video educativo, Daniel Bañobre Dopico y Francisco Alvarellos Maceira se toman tan solo 2 minutos y 30 segundos para contarnos sobre el Software Libre. **A disfrutarlo y compartirlo!**



Subscribe al RSS Guardarlo en Del.icio.us Compartirlo en Twitter Compartirlo en Facebook

educativo, Software libre, software propietario, Video

Dejá tu comentario (5)

as frecuentes Artículos Blog Grupo Condiciones

¿Cómo empiezo?

1. Lee la [Guía de introducción](#).
2. Examina la [Galería de aplicaciones](#) y observa el [código de ejemplo](#).
3. Pide ayuda a la comunidad. Únete a nuestra comunidad y participa en el grupo de debate.
4. ¿Estás listo para empezar a programar? Lee la documentación del desarrollador para el [API de datos](#), las [API del reproductor](#) o el [reproductor sin bordes](#).

Vídeo de aspectos generales de las API de YouTube



el video especificado.

- ▶ **player.cueVideoByUrl:** carga la miniatura del video indicado en la URL.
- ▶ **player.loadVideoByUrl:** reproduce el video de la URL indicada.

En cuanto a funcionalidades del reproductor, podemos detallar las siguientes:

- ▶ **player.playVideo():** reproduce el video en cola en ese momento.
- ▶ **player.pauseVideo():** pausa el video que se está reproduciendo.
- ▶ **player.stopVideo():** detiene el video actual.
- ▶ **player.seekTo(seconds):** busca el momento especificado del video.

También hay funcionalidades que se refieren al volumen del reproductor, tamaño, estado, calidad, información del video y eventos especiales que podemos utilizar para distintas situaciones.

EL ACCESO A LOS DATOS

Es posible acceder a datos a través de llamadas a distintas URLs, que los devuelven en los formatos de sindicación **Atom 1.0** y **RSS 2.0**. Estas llamadas acceden a las siguientes funcionalidades: autenticación, obtención de datos y feeds de video, subir videos, actualiza-

ción y eliminación de videos, funciones de la comunidad (comentar, votar), almacenamiento y recopilación (favoritos, listas de reproducción, suscripción a canales) y manejo de perfil de usuario. Esta última API no es para ActionScript, y no existe una biblioteca específica, ya que si bien es bastante frecuente que se use Flash para cargar el player, no suele ser así cuando se recuperan datos más generales, existiendo bibliotecas específicas para Java, .NET, PHP y Python.

¿POR QUÉ ES CARO EL STREAMING DE VIDEO?

El almacenamiento y la difusión de material audiovisual implica un gran uso de espacio en disco y de ancho de banda (transferencia de datos). Un simple video de menos de 2 minutos puede pesar cientos de MB en un formato no comprimido. Incluso, en los comprimidos, el material que supera una hora puede tener varios GB. Estos archivos deben ser no solo almacenados, sino también convertidos desde el formato original y transferidos al usuario cada vez que los solicita. Para que la descarga no resulte lenta, hay que tener replicado el material en una serie de servidores que atiendan las peticiones por cercanía. Todo esto conforma un complejo CDN (**Content Delivery Network**, o red de distribución de contenido) con una infraestructura y un costo que se multiplican con la cantidad de material. Todavía hoy, aun a pesar de que Google introduce publicidad en los videos, los costos están lejos de cerrar. ◀



CONOCIMIENTOS TÉCNICOS

Recordemos que, para mejorar y potenciar nuestro aprendizaje, desde www.redusers.com/cursos/webdesign podemos descargar los archivos editables de los proyectos que realizamos en esta colección. A partir de ellos, podremos obtener ideas para un emprendimiento propio o, directamente, modificar los archivos para reutilizarlos. En el sitio también hay **guías visuales**, un **glosario de términos** y una completa guía de preguntas frecuentes sobre la suite CS5. Además, el libro **Desarrollo y animación en ActionScript 3.0** nos ayudará a comprender los fundamentos de la programación en Flash.

GOOGLE MAPS

Google ofrece un componente que permite incluir mapas en nuestras películas de Flash, además de una API destinada a manipularlos para nuestra comodidad. Veamos cómo utilizarlo.

Los servicios de mapas se han popularizado mucho y han surgido competidores muy fuertes, entre ellos, Google y Yahoo. Los mapas no solo pueden ser consultados en forma estática (como lo haríamos con uno de papel), sino que a través de ellos, podemos obtener rutas desde un punto a otro para planear nuestro viaje. También nos permiten conocer la ubicación de negocios, hospitales y estaciones de servicio; e, incluso, ver fotos satelitales de la zona en la que estemos interesados.

GOOGLE EARTH

Es un software que se instala localmente y que carga imágenes satelitales de cualquier parte del mundo. Con él, podemos mirar nuestra propia casa desde el cielo, no importa en donde estemos (aunque, claro, el detalle variará según nuestra ubicación geográfica). Algunas posibilidades que nos ofrece son:

GOOGLE STREETS ES UN SERVICIO DE GOOGLE QUE PERMITE RECORRER UNA CIUDAD VIRTUALMENTE, DESPLAZÁNDOSE POR SUS CALLES.

- ▶ Incluir mapas en nuestra propia película, en vez de recurrir a un link externo.
- ▶ Hacer consultas a los mapas sin salirnos de nuestro sitio.
- ▶ Agregar marcadores, quitarlos y relacionar cualquier información con ellos.
- ▶ Hacer seguimientos de rutas entre dos puntos.
- ▶ Arrastrar el mapa para ver zonas no visualizadas originalmente.

Todas estas posibilidades de manipulación, sumadas a la inclusión de imágenes

satelitales, brindan el acceso a una completa localización geográfica que puede ser utilizada, por ejemplo, desde en una simple invitación para una fiesta, hasta para dar la ubicación de locales en una campaña de marketing.

HERRAMIENTAS BÁSICAS

Lo primero que debemos tener para poder interactuar con el API de Google Maps es una clave que nos permita acceder a sus servicios. Esta clave se obtiene completando un formulario que hay en el sitio, en donde debemos indicar en qué dominio queremos usar los servicios (este debe coincidir con el dominio desde donde se descarga el SWF). Una vez hecho esto, descargamos la librería, que viene en dos versiones: para Flash y Flex (en ambos casos, como archivo SWC). Para utilizarla, copiamos el SWC obtenido a la carpeta de componentes de Flash. Cuando ya lo tenemos en la librería, solo nos resta arrastrarlo a escena (en realidad, es solo para incluir las librerías, no es algo inmediatamente visual), y ya contaremos con el entorno listo para funcionar. Más adelante explicaremos en detalle de qué manera configurar el entorno y cargar un mapa. Ahora, comentaremos los principales elementos que distinguen a esta API.

LATITUDES Y LONGITUDES

Se inicializa mediante un objeto **LatLng**. Este objeto ofrece el método básico para hacer referencia a ubicaciones en el mapa. Se crea mediante el traspaso de sus parámetros en orden { latitud, longitud }, tal como se hace tradicionalmente en cartografía. Para definir latitud y longitud, basta con ingresar una dirección que Google pueda reconocer a través del proceso de codificación geográfica, que transforma direcciones (como "1600 Amphitheatre Parkway, Mountain View, CA") en coordenadas geográficas (como





Google Maps es una aplicación web, es decir que no se instala en nuestra computadora.

37.423021 de latitud y -122.083739 de longitud). Estas pueden utilizarse para colocar marcadores o situar el mapa. Para obtener dicha información, se usa un objeto **ClientGeocoder**, que escucha a una serie de eventos (`GeocodingEvent.GEOCODING_SUCCESS` y `GeocodingEvent.GEOCODING_FAILURE`) en respuesta a la llamada **ClientGeocoder.geocode()**. Para definir los límites del mapa, se emplea un objeto **LatLngBounds**, que define una región rectangular mediante dos objetos **LatLng** que representan las esquinas sudoeste y noreste del cuadro de límites, respectivamente.

INTERACCIONES CON EL MAPA

De manera predeterminada, los objetos del mapa tienden a reaccionar ante la actividad del usuario del mismo modo que en <http://maps.google.com>. No

+ MARCADORES

Los marcadores son pequeños iconos que pueden agregarse en un mapa y que representan un lugar que queremos destacar por algún motivo. Un marcador puede indicar cualquier referencia: la dirección buscada, una tienda, un hotel o, simplemente, un punto de interés. Además de agregar un marcador, podemos asociar a él cualquier información que precisemos, con lo cual se convertirá en una poderosa herramienta de marketing.

obstante, es posible modificar este comportamiento a través de distintos métodos. Por ejemplo, `Map.disableDragging()` desactiva la capacidad de hacer clic y arrastrar el mapa a una nueva ubicación. También se puede interactuar con el mapa mediante programación. La interfaz Map proporciona métodos que permiten alterar el estado del mapa directamente. Por ejemplo, los métodos **setCenter()**, **panTo()**, **zoomIn()** y **zoomOut()** operan sobre el mapa, en vez de sobre la interacción del usuario. Con esta simple línea de código, generamos un movimiento del mapa:

```
map.panTo(new
  LatLng(37.4569,
    -122.1569));
```

En Google Streets podemos apreciar el Big Ben. Es posible ver cómo las fotos se encadenan entre sí, y sobre el borde del puente, se borraron algunas personas.

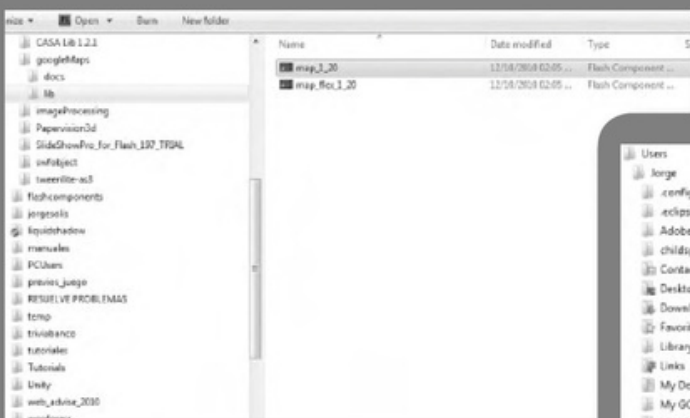
GOOGLE STREETS

Es un servicio de Google que permite recorrer una ciudad virtualmente, desplazándose por sus calles; incluso, permite mirar la fachada de nuestra casa. Está disponible para las principales ciudades de Europa, Estados Unidos, Canadá, Australia y Japón, aunque de a poco, otras más se van agregando a la lista. Las fotografías se toman desde un vehículo especial que recorre las calles con cámaras en el techo; así, se captan imágenes 360 que luego se utilizan para ensamblar las vistas. Hubo inconvenientes con personas que no querían que su casa apareciera en esta guía, por lo cual Google habilitó formularios especiales en donde es posible solicitar la eliminación de algunos lugares del recorrido. ◀

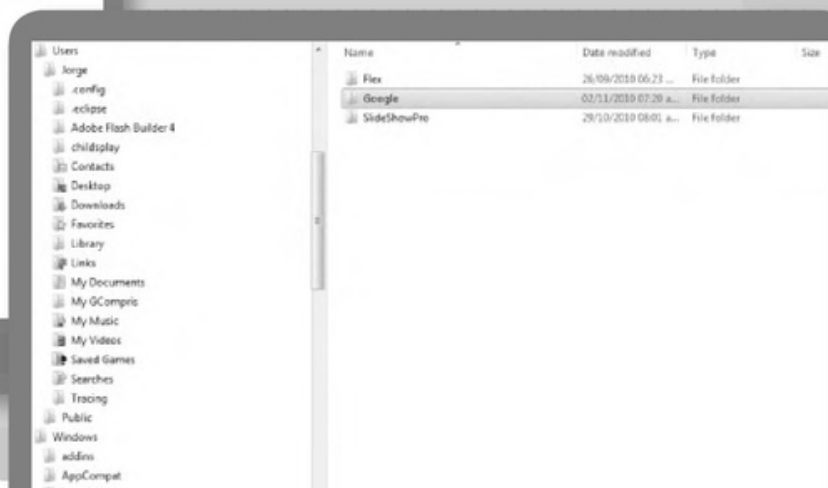
CÓMO CARGAR COORDENADAS

PASO A PASO

Google Maps ofrece una API muy simple de usar, con la que podemos incluir mapas rápidamente en un sitio web.



01 | DESCARGAR EL SDK



CREAR CARPETA

02

La posibilidad de hacer zoom y paneos, utilizar marcadores e incluir comentarios convierte a este servicio en una eficaz herramienta de marketing. En este ejemplo, veremos la mecánica básica de carga y posicionamiento por coordenadas. Vamos a utilizar el SDK que ofrece Google en ActionScript 3.0 para cargar un mapa y situarlo con centro en una coordenada concreta de la ciudad de Buenos Aires. Luego, colocaremos diez marcadores al azar. ◀

PASO | 01 Para empezar a trabajar con la librería, tenemos que descargarla desde el sitio de Google. Obtenemos el SDK desde: <http://maps.googleapis.com/maps/flash/release/sdk.zip> y lo descomprimos en nuestro disco rígido.

PASO | 02 Creamos una carpeta con el nombre de Google dentro de la carpeta de compo-

nentes de Flash, que variará según el sistema operativo. En Windows 7 es algo así:

C:\Users\{Usuario}\AppData\Local\Adobe\Flash CS5\es_ES\Configuration\Components

PASO | 03 Reiniciamos Flash para que el componente agregado en el paso anterior aparezca en la carpeta correspondiente. Vamos al menú **[Archivo/Nuevo/ActionScript 3.0]** y, de este modo, generamos el documento en el que vamos a trabajar.

PASO | 04 Ahora tenemos que incluir el componente que hemos copiado en el paso 2, para poder hacer uso de la biblioteca. Abrimos el panel de componentes desde el menú **[Ventana/Componentes]**, abri-

04 INCLUIR COMPONENTE

```

1 import com.google.maps.LatLng;
2 import com.google.maps.Map;
3 import com.google.maps.MapEvent;
4 import com.google.maps.MapType;
5
6 var mapa:Map = new Map();
7 mapa.key = "tu_api_key";
8 mapa.setSize(new Point(stage.stageWidth, stage.stageHeight));
9 mapa.addEventListener(MapEvent.MAP_READY, onMapReady);
10 this.addChild(mapa);
11
12 function onMapReady(event:Event):void {
13     mapa.setCenter(new LatLng(40.736072, -73.992062), 14, MapType.NORMAL_MAP_TYPE);
14 }

```

General

Plantillas

Tipo:

- ActionScript 3.0
- ActionScript 2.0
- Adobe AIR 2
- iPhone OS
- Flash Lite 4
- Adobe Device Central
- Clase de ActionScript 3.0
- Interfaz de ActionScript 3.0
- Archivo ActionScript
- Archivo comunic. ActionScript
- Archivo JavaScript de Flash
- Proyecto de Flash

Descripción:

Cree un nuevo archivo FLA en la ventana de documento de Flash. Se ajusta la configuración de publicación de AS 3.0. Utilice archivos FLA para configurar el contenido multimedia y la estructura de los archivos SWF publicados para Adobe Flash Player.

03 REINICIAR FLASH

INCLUIR MAPA 05

AGREGAR MARCADORES 06

mos la carpeta Google y arrastramos el componente Google-MapsLibrary al escenario. Se verá solo un recuadro azul, pero no importa, porque es una representación visual que nos permitirá recordar que tenemos incluidas las librerías.

PASO | 05 Para poder agregar el mapa en nuestra película, debemos utilizar código ActionScript 3.0, ya que no es posible hacerlo sin él. Entonces, seleccionamos el único fotograma de la línea de tiempo y oprimimos <F9> para abrir el panel de acciones, donde escribimos el código que se ve en la imagen.

PASO | 06 Vamos a agregar algunos marcadores al azar en nuestro mapa, para demostrar su funcionalidad. Volvemos a seleccionar el único fotograma de la película y abrimos el panel de acciones con <F9>. Allí modificamos la función **onMapReady** y agregamos dos imports debajo de los que ya tenemos:

```

import com.google.maps.MapType;
import com.google.maps.overlays.Marker;

```

RESUMEN

Con estos sencillos pasos, hemos demostrado de qué manera cargar coordenadas en Google Maps y, además, cómo establecer referencias por medio de marcadores.

PAPERVISION 3D

Flash se orienta a la animación, utilizando elementos vectoriales para minimizar el tamaño de los archivos. Las librerías de 3D compiten por ser el estándar; veamos una de las más importantes.

El uso de bitmaps y la transición de la animación plana hacia la ilusión de las tres dimensiones fueron llegando con el tiempo. Quienes se hayan asomado al tema habrán podido jugar con extensos listados de código que hacían girar un cubo en el espacio dibujado con su esqueleto. A la hora de integrar 3D de objetos reales, lo más común era importar una secuencia desde 3D Studio Max, Maya o similar, en forma de video o secuencia de PNG que generaban archivos de un peso muy importante. Luego, los recorridos eran, simplemente, desplazarse hacia adelante o hacia atrás en la animación. De hecho, esta sigue siendo la opción más recomendable para pequeñas animaciones integradas en banners, ya que cada tarea se mantiene en la órbita del especialista y, así, se simplifica la programación.

DESAFIAR EL RENDIMIENTO

Papervision 3D surgió como tantas otras librerías, jugando con el escalado y la profundidad de los elementos para simular el **eje Z** y montando los algoritmos de cálculo de proyección en el espacio tridimensional (cámara, luz, recorridos), para tratar de minimizar el uso de recursos. Sin prestar atención especial al rendimiento, es posible bloquear la memoria y la CPU de las máquinas corriendo Flash Player, que de ninguna manera es un programa orientado a ejecutar animaciones en 3D. La primera versión beta del programa, allá por diciembre de 2005, dejó asombrados a quienes lo probaron, porque los objetos con textura rotaban sin ser una secuencia prearmada. Los que participaron en los primeros experimentos pueden dar fe de que el sistema estaba en el borde mismo del rendimiento, por lo que el uso de los efectos resultaba bastante dificultoso fuera del campo experimental. Con la evolución de Flash Player –y, sobre



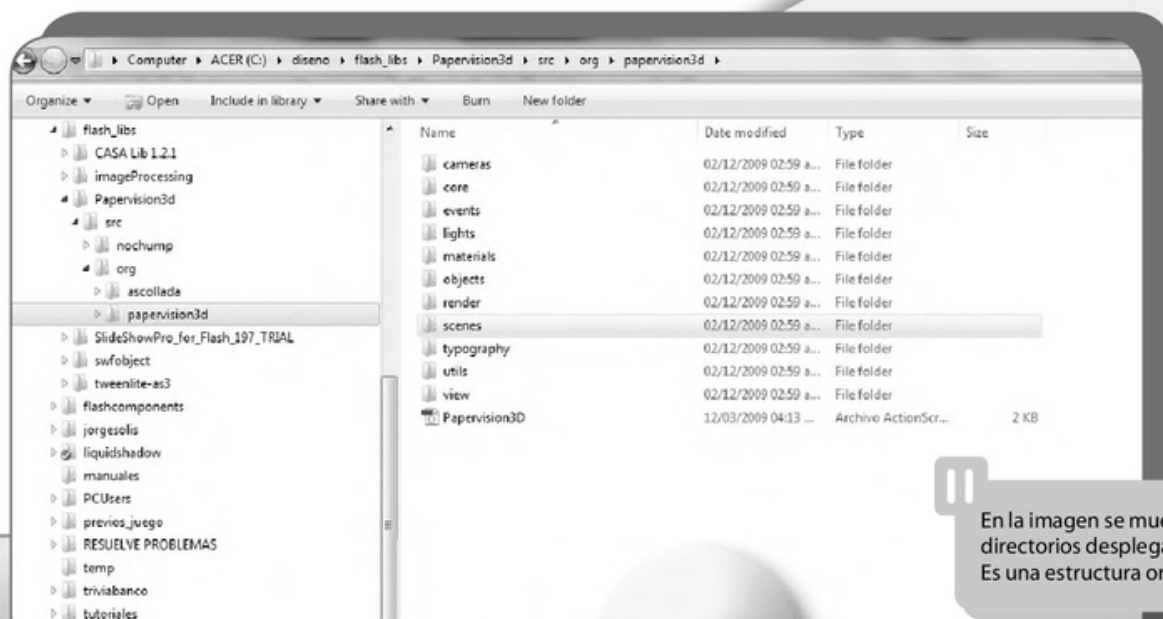
¿MODELAR EN PAPERVISION?

Es preciso tener en claro que Papervision 3D es una librería que permite renderizar objetos en tres dimensiones, ya sea a través de formas básicas o con archivos externos DAE, que son modelos descritos en un esquema XML. No tiene utilidades de modelado más allá de algunas herramientas básicas, y si necesitamos crear modelos complejos en 3D, la opción más adecuada es recurrir a programas específicos, como 3D Studio Max o Maya. Papervision se enfoca en renderizar, no en modelar.

todo, con la aparición de AS3 y una nueva máquina virtual–, habían surgido las condiciones para un uso menos demandante y, por lo tanto, menos peligroso en el sentido del rendimiento. De allí en más, no solo aparecieron por doquier simples rotaciones de objetos en 3D, sino que también empezaron a verse complejas escenas y juegos en entornos tridimensionales.

EJEMPLOS SENCILLOS

Lo primero que debemos hacer para empezar a utilizar la librería de Papervision es descargar la última versión disponible desde <http://code.google.com/papervision3d/downloads/list>. Si queremos curiosar o hacer algún tipo de modificación en el código fuente, podemos bajar el archivo ZIP y expandir



En la imagen se muestra el árbol de directorios desplegado de Papervision. Es una estructura organizada en carpetas.

Entorno marítimo virtual de la homepage de www.papervision3d.org, en el cual el usuario asume el rol de buceador.



todo su contenido a un directorio. En caso de que queramos hacer uso de su funcionalidad, podemos obtener el SWC, archivo comprimido que Flash utiliza como biblioteca. Otra ventaja de estos archivos es que son fácilmente portables entre proyectos, ya que, en vez de copiar una librería completa con gran cantidad de carpetas (o de configurar un **classpath** que se puede perder al cambiar de máquina), simplemente movemos un solo archivo que podemos pasar a otro desarrollador si fuera necesario.

Al descargar la librería, veremos que hay una gran cantidad de elementos, como cámaras, vistas, materiales y luces, y otros propios del lenguaje 3D, más la clase Papervision3D, que contiene una serie de parámetros generales necesarios. Si bien en las primeras versiones era preciso

PAPERVISION 3D SURTIÓ COMO OTRAS LIBRERÍAS, JUGANDO CON LOS ELEMENTOS PARA SIMULAR EL EJE Z.

crear algún tipo de base sobre la cual empezar, el proceso se ha simplificado bastante con la inclusión de la clase base **View**, porque con ella podemos tomar como base para instanciar de forma simple lo mínimo necesario.

IMPORTACIÓN DE MODELOS

Si bien el uso de objetos primitivos puede servir para componer elementos simples, en realidad, lo más interesante se logra importando objetos previamente

modelados en programas como 3D Studio Max o Maya. La importación de modelos en 3D fue posible, en un principio, gracias al formato de Collada (**Collaborative Design Activity**, archivos con extensión DAE). Se trata de un formato intermedio al que muchos programas de modelado pueden exportar los objetos. Constituye un esquema XML que define una serie de características que todos los programas adheridos se comprometen a soportar.

POSIBILIDADES DE PAPERVISION 3D

Papervision 3D, junto a otras librerías de 3D que han ido surgiendo en Flash, como Away3D, fue utilizado para introducir juegos en entornos tridimensionales o para elaboradas interfaces. Pero siempre con el foco puesto en el rendimiento, debido a que Flash no tiene acceso directo al subsistema gráfico de la computadora. Esto hace que el abuso de efectos o el mal planeamiento de recursos haga que la CPU y la memoria queden sobrecargados, situación que debemos evitar cuando trabajamos con este tipo de librerías. Para esto, es importante que, siempre que publiquemos, verifiquemos el rendimiento del sistema y comprobemos si la reproducción es fluida, pensando que también el usuario puede llegar a tener una máquina más lenta. Para próximas versiones de Flash Player se anuncia un sistema de acceso directo a la placa gráfica a través de OpenGL en Mac y Linux, y DirectX en Windows. Esto permitirá mejorar muchísimo el rendimiento de rendering en 3D, aunque, por ahora, el sistema estará disponible en beta hacia el primer trimestre de 2011. ◀

FACEBOOK API

Facebook se ha convertido en una de las redes sociales más importantes de Internet, y su API fue una de las claves de su éxito. Veamos qué función cumple y de qué modo podemos aprovecharla.

Facebook es, quizás, uno de los sitios más importantes de la Red contemporánea, emblema de lo que se denomina, genéricamente, "redes sociales". No solo tiene millones de usuarios, sino que ha marcado la manera en que la gente se relaciona en la Red, haciendo populares términos como "el muro" o "taggear". El API de Facebook contribuyó en gran parte a lograr ese éxito, porque miles de aplicaciones pudieron interactuar con el entorno del sitio. El beneficio fue mutuo: a la vez que agregaban valor al portal, conseguían público. De hecho, hay empresas que se dedican a desarrollar campañas de marketing y juegos específicamente para Facebook, al tener un conocimiento profundo no solo de cuestiones técnicas, sino también de la misma dinámica de los usuarios y del uso que hacen del sitio.

DESCRIPCIÓN DEL API

Facebook está en un proceso de cambio, en el que conviven una API basada en simples llamadas HTTP y otra más nueva orientada a objetos. Hay una librería de Flash que usa la primera, basada en simples llamadas HTTP, denominada **REST** (*Representational State Transfer*, o transferencia de estado representacional). Podemos encontrarla en <http://code.google.com/p/facebook-as3>. La API más nueva, **Graph API**, se orienta a objetos para crearlos y relacionarlos a través de llamadas **POST** a **URLs** en donde pasamos la información relevante.

Los principales objetos de la nueva Graph API son:

- ▶ **Group:** grupo de Facebook.
- ▶ **Album:** álbum de fotos.
- ▶ **Link:** link compartido.

- ▶ **Post:** entrada individual dentro de las noticias de un perfil.
- ▶ **Insights:** estadísticas acerca de una aplicación, página o dominio.
- ▶ **Checkin:** check-in realizado a través de algún sitio de Facebook.
- ▶ **Event:** evento de Facebook.
- ▶ **Application:** aplicación individual registrada en la plataforma.
- ▶ **Note:** nota de Facebook.
- ▶ **Photo:** foto individual.
- ▶ **User:** perfil de un usuario.
- ▶ **Status message:** mensaje en el muro del usuario.
- ▶ **Page:** página de Facebook; este objeto soporta updates realtime para todas sus propiedades, excepto aquellas marcadas con un asterisco.
- ▶ **Subscription:** suscripción individual de una aplicación para obtener actualizaciones realtime de cierto tipo.
- ▶ **Video:** video individual.

Para dar un ejemplo, una llamada para poner un mensaje en el muro de Alicia sería así:

```
curl -F 'access_token=xxxxxxxxx' \
      -F 'message=Hola
      Alicia, me gusta este
      nuevo API.' \
      https://graph.
      Facebook.com/alicia/feed
```

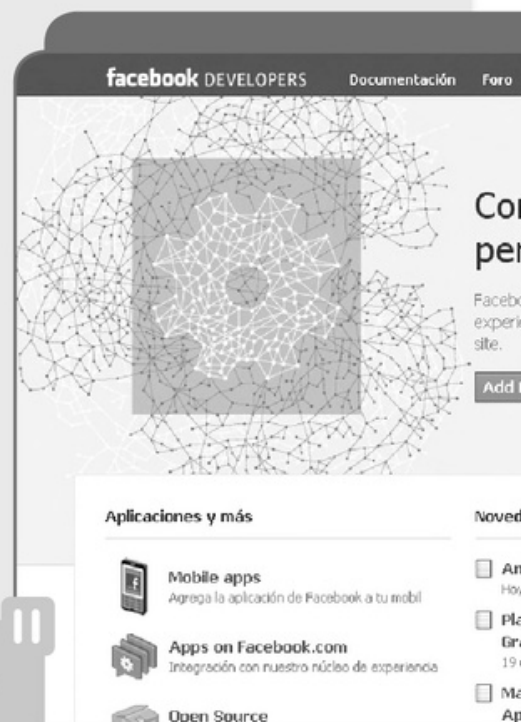
En este caso, se usa curl, una librería que permite simular operaciones web.

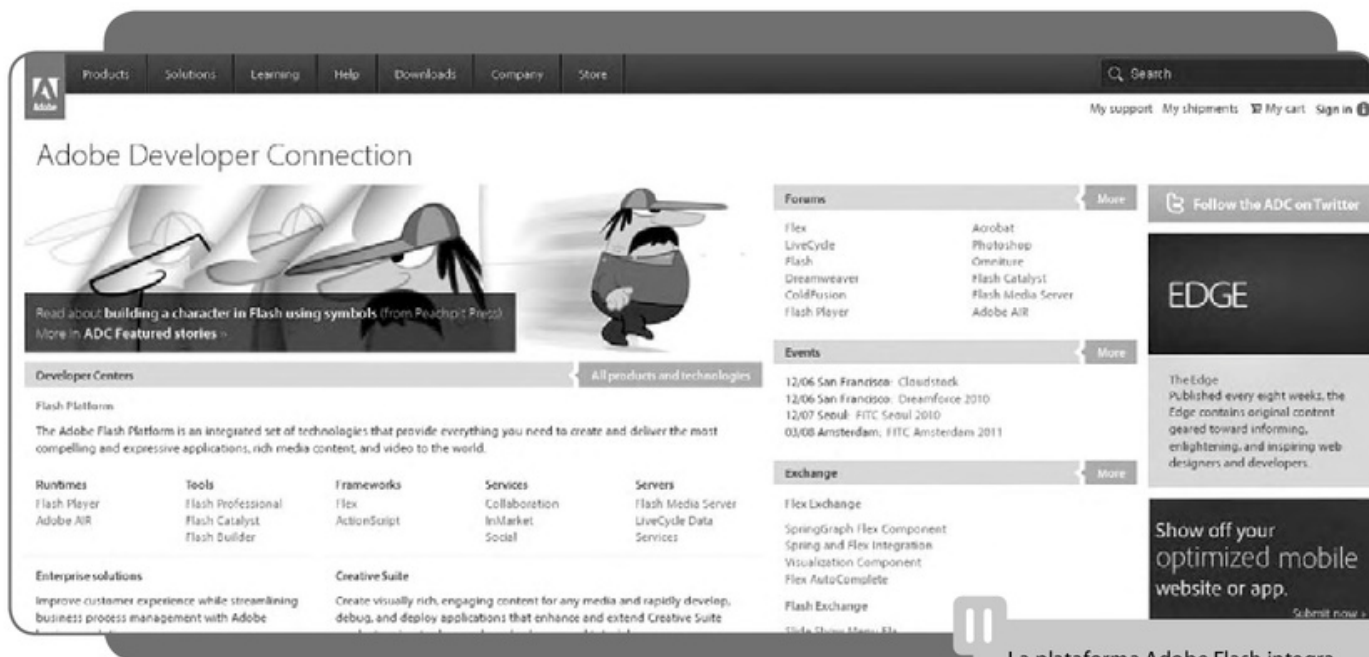
Las API de Facebook permiten crear experiencias sociales para impulsar el crecimiento en nuestros sitios web.

ACCESO MEDIANTE LA LIBRERÍA DE ACTIONSCRIPT

La biblioteca de ActionScript brinda acceso a muchas funcionalidades de la API, aunque no todas las operaciones están permitidas. Para operar con Facebook, lo primero es conseguir un **API Key** en el mismo sitio, que luego emplearemos

EL API DE FACEBOOK PERMITIÓ QUE MILES DE APLICACIONES INTERACTUARAN CON EL ENTORNO DEL SITIO.





La plataforma Adobe Flash integra tecnologías que permiten crear y entregar contenido multimedia.

para construir el cliente desde Flash, usando la librería que podemos descargar desde <http://code.google.com/p/facebook-actionscript-api>.

Con esta biblioteca, es posible construir tres tipos de aplicaciones:

► Aplicaciones en Facebook:

son aplicaciones puestas en servidores propios, pero a las que se accede desde Facebook.

► **Aplicaciones externas:** el usuario accede directamente en otro dominio, pero utilizan la plataforma de Facebook para

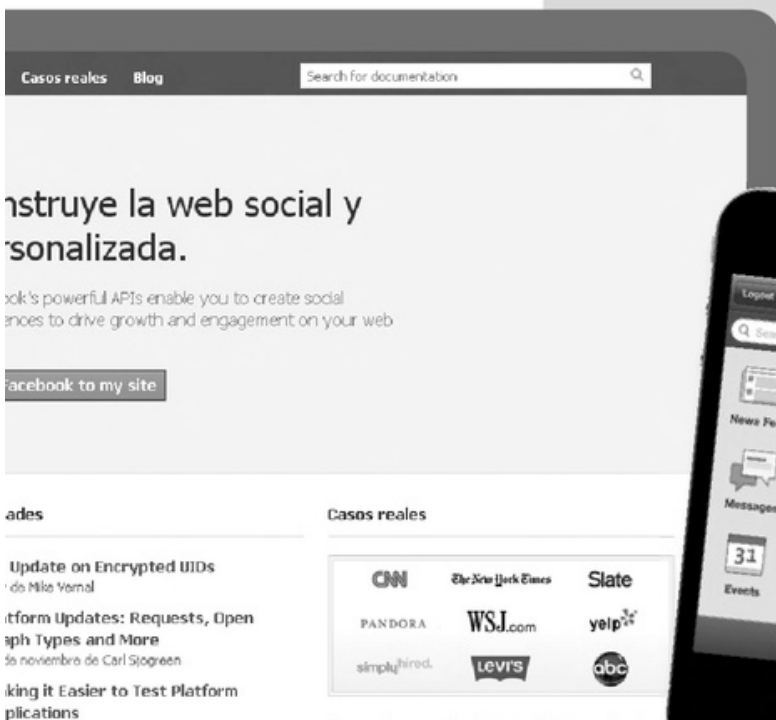
socializar los resultados y ganar adeptos.

► **Aplicaciones de escritorio:** desarrolladas con Adobe AIR o como proyectores, que al igual que las aplicaciones externas, acceden a Facebook para beneficiarse de la socialización de los resultados. Adobe le dedicó una sección completa a Facebook (www.adobe.com/devnet/facebook.html), en donde no solo publicó la librería, sino que también recopiló todo tipo de artículos para ayudar a la comunidad de desarrolladores a interactuar con el API. Estos van desde el análisis básico de la arquitectura hasta el showcast de distintos ejemplos,

poniendo especial énfasis en aquellos que generan una ganancia económica, como los juegos.

¿QUÉ CONSTITUYE LOS SOCIAL MEDIA?

Posiblemente hayamos oído hablar de los **social media**, sitios que generan comunidades a partir de ideas compartidas o de grupos de pertenencia. De los muchos sitios que constituyen hoy el llamado espacio social de la Web, dos se destacan claramente: Facebook y Twitter. Facebook nació con la idea de conectar antiguos compañeros de secundaria, pero hoy permite encontrar amigos de todo tipo, compartiendo a través de los muros toda clase de pensamientos y mensajes, además de promocionando eventos. Por su parte, Twitter permite comunicar qué se está haciendo en un momento determinado a través de un mensaje limitado a 140 caracteres. Esto generó una gran cantidad de usuarios interesados en intercambiar mensajes cortos, que pronto inundaron la Red, a tal punto que, muchas veces, las figuras públicas son analizadas a partir de lo que publican en sus cuentas, que son frecuentemente citadas por los medios periodísticos. ◀



ADOBE CS5

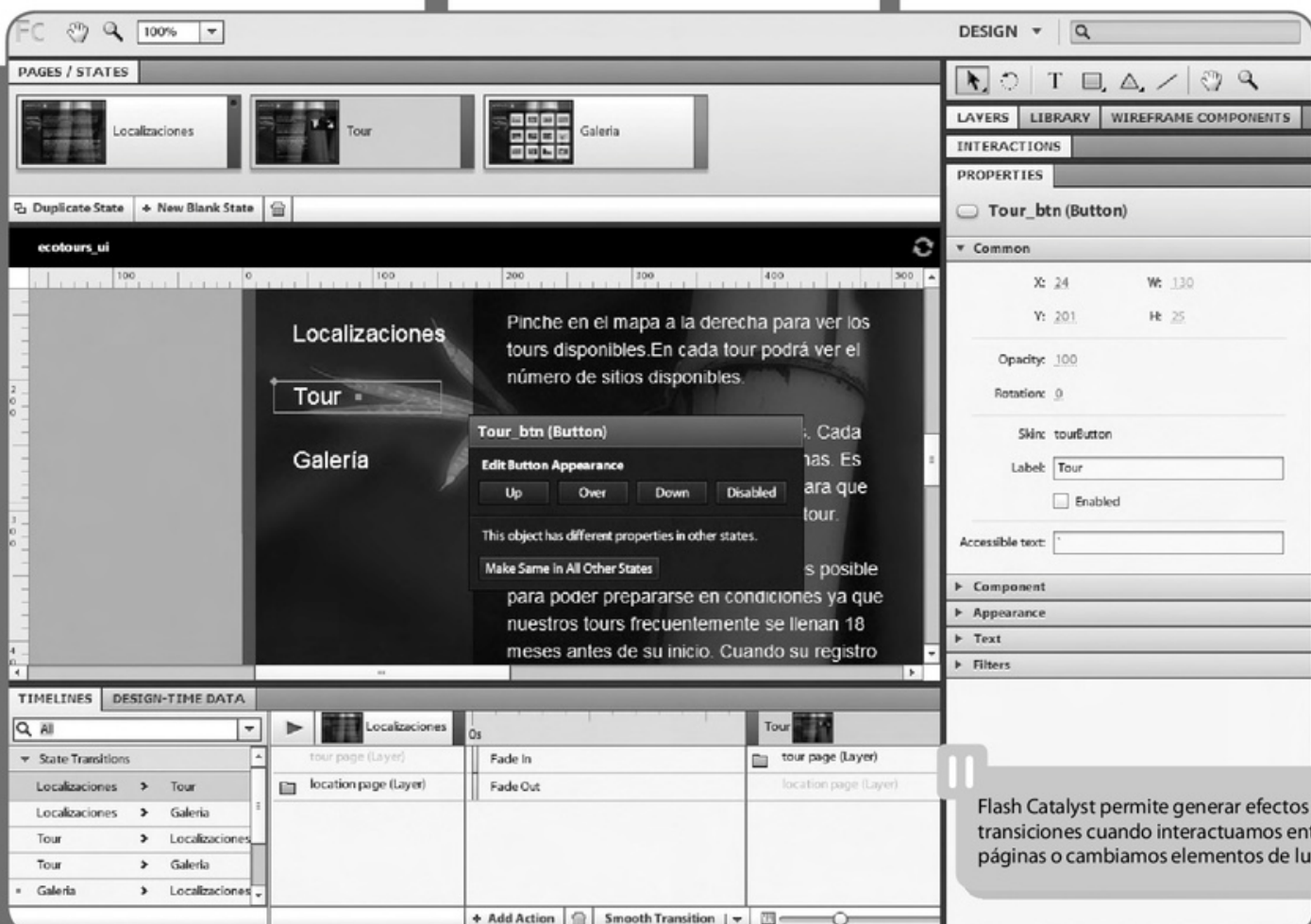
FLASH CATALYST

Es una herramienta que apunta a reducir los problemas de integración de arte en aplicaciones web desarrolladas con Flex. Veamos cuáles son sus ventajas y qué aportes puede brindarnos para nuestros proyectos.

Catalyst es una herramienta orientada, sobre todo, a los diseñadores que trabajan sobre la interfaz de aplicaciones en Flex, ahora Flash Builder, ya que resuelve un problema bastante recurrente en el ciclo de trabajo: el diseñador debe tocar el mismo código fuente que el desarrollador. Esto provoca inconvenientes de integración, porque el diseñador puede romper la lógica de programación sin querer, o viceversa.

Para entender mejor la dificultad, el lenguaje de etiquetas de Flex (MXML) contiene tanto la lógica como el layout de la aplicación. Esto fuerza a que el diseño y la programación estén en el mismo archivo MXML. Catalyst permite importar archivos de Photoshop o Illustrator como punto de partida de la aplicación web. Ofrece una completa paleta de componentes, los mismos que tenemos en Flash Builder, con lo cual la construcción de la interfaz

de usuario logra permanecer funcional e intacta cuando es pasada a la parte de programación. También la integración con datos es muy simple, porque, a través de una serie de ayudantes, podemos introducir datos rápidamente desde distintas fuentes y hacer, por ejemplo, una lista que se actualice según esa información. Todo esto puede llevarse a cabo sin necesidad de tener profundos conocimientos sobre programación. Otra funcionalidad importante de



Flash Catalyst permite generar efectos y transiciones cuando interactuamos entre páginas o cambiamos elementos de lugar.



Adobe Catalyst permite importar archivos de Photoshop e Illustrator como punto de partida de la aplicación web.

mos elementos de lugar. Por defecto, incluye una transición que no tiene duración, pero que cobra efecto si incrementamos el tiempo entre los dos estados en el timeline. La animación predeterminada es **fade in/fade out**, pero hay otras que podemos utilizar. Las siguientes transiciones se agregan automáticamente cuando cambiamos las propiedades de un objeto entre estados:

- Posición
- Rotación
- Reescalamiento

Hay otras propiedades en donde no se agrega ningún efecto automáticamente:

- Cambio de relleno o borde
- Opacidad
- Corners
- Modos de blend
- Filtros
- Cambio de formateo de texto
- Cambio de texto

Es importante destacar que, al crear transiciones avanzadas, podemos modificar el movimiento a través de distintas fórmulas de **easing** para que sea más realista. Esto se hace desde el panel de propiedades, donde contamos con las siguientes opciones:

- **Default:** opción que cambia la velocidad constante.
- **Linear:** pequeña aceleración al comienzo y desaceleración al final, movimiento constante.
- **Sine:** acelera hasta la mitad del movimiento, desacelera hasta el final.
- **Power:** similar a sine, pero se puede definir el exponente que incrementa la aceleración/desaceleración.
- **Elastic:** genera un pequeño movimiento hacia atrás antes de llegar a destino.
- **Bounce:** hace un rebote al llegar a destino.

Para agregar mayor riqueza a nuestras transiciones, podemos combinarlas. En ese caso, abrimos el timeline, seleccionamos **[State Transitions]**, luego, un objeto, presionamos el botón **[Add Action]** y elegimos un efecto de popup. ◀

Catalyst es la exportación/importación de los gráficos en un nuevo formato XML de intercambio gráfico denominado FXG (**Flash XML Graphics**). Se trata de un lenguaje declarativo de gráficos que da la posibilidad de incluir la descripción del material gráfico dentro del mismo MXML. Posiblemente, esta no sea una ventaja tan evidente, pero el hecho de usar un lenguaje declarativo da la opción de incluir el material gráfico en los sistemas de control de versiones más populares, que requieren archivos de texto plano para rastrear las versiones y compararlas.

GENERACIÓN DE CÓDIGO

Usando Flash Catalyst, nos movemos en un entorno enteramente visual. Todos los elementos se ven en modo diseño, y accedemos a ellos a través de los distintos paneles, tanto para agregar interactividad como para cambiar el comportamiento entre los diferentes estados. Esto ofrece un entorno amigable para quienes están acostumbrados a usar programas orientados a lo visual. El ejemplo más

cercano es el editor de HTML de Adobe, Dreamweaver, en donde también podemos trabajar visualmente para generar código HTML.

El flujo de trabajo en Catalyst casi siempre comienza con un archivo de Illustrator o Photoshop, que se usa de base para la composición. Si bien podríamos construir la interfaz desde cero (ya sea vectorialmente o importando arte), lo cierto es que Catalyst no es una herramienta de diseño, sino de maquetación. En la imagen se muestra el código generado por Catalyst para el diseño del paso a paso que veremos en las próximas páginas. El aspecto es parecido a un HTML o a los mismos documentos de Flex, ya que, en realidad, se trata de eso: un documento MXML que podemos llevar a nuestro proyecto Flex para continuar trabajando directamente en él.

TRANSICIONES Y EFECTOS

Flash Catalyst permite generar efectos y transiciones con gran facilidad cuando nos movemos entre páginas o cambia-



¿FLEX O FLASH BUILDER?

A partir de la versión 4, Adobe decidió cambiar el nombre a la herramienta Flex, que pasó a llamarse FlashBuilder. La idea era emparentar a Flex con su tecnología madre, Flash, porque, finalmente, el archivo que genera Flex es un SWF a partir de una serie de MXML. Con esta modificación, se intentó devolver Flex a su contexto, es decir, resaltar que se trata de un framework de Flash para construir aplicaciones (que incluye su IDE, Flex Builder), y no, de una herramienta totalmente separada.

Con la inclusión de Catalyst en el ciclo de desarrollo, la necesidad de generar un concepto alrededor de Flash maduró la estrategia y condujo al cambio de nombre.

UTILIZAR FLASH

PASO A PASO

Flash Catalyst ofrece muchísimas posibilidades a los diseñadores para generar archivos MXML sin necesidad de escribir código.



01 | IMPORTAR



CONFIGURAR **02**

En este ejercicio, vamos a crear un pequeño sitio de turismo con dos opciones de menú simples que mostrarán información basada en texto y en un mapa interactivo, desde donde se accede a algunos detalles. Como punto de partida, utilizaremos un archivo de Photoshop que ya contiene la base del diseño del sitio, de modo de enfocarnos en los aspectos del armado.

PASO | 01 Vamos a importar el archivo PSD, para lo cual abrimos Flash Catalyst y, en la pantalla de bienvenida, seleccionamos **[Create new Project from PSD file]**. En el cuadro de diálogo de importación dejamos las opciones predefinidas, a excepción del color de fondo, que cambiamos a negro.

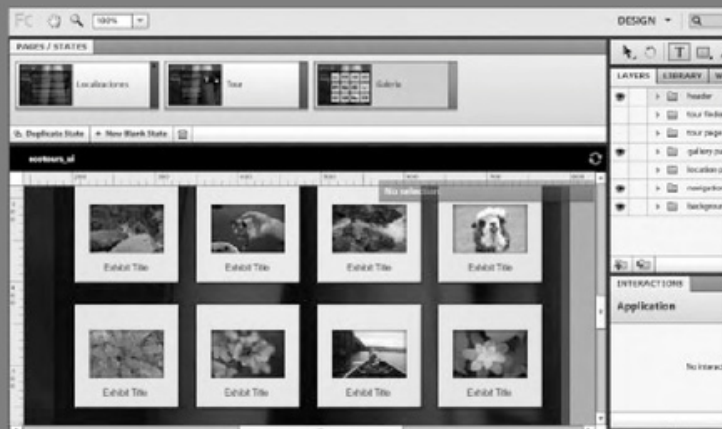
PASO | 02 Pasaremos a agregar las tres páginas de nuestro ejemplo. Hacemos clic derecho

en la de arriba y seleccionamos **[Duplicate state]** dos veces, para tener tres páginas. Luego, editamos los nombres de cada una de modo que se llamen Localizaciones, Tour y Galería, como cada una de las opciones que tenemos en el menú.

PASO | 03 Para especificar cuál será el material visible en cada estado, debemos seleccionarlo desde el panel superior. Seleccionamos las capas que queremos que estén visibles desde el panel de la derecha, en donde veremos que hay una para cada página. Debemos tener esto en cuenta para organizar nuestro material en Photoshop.

PASO | 04 Vamos a convertir los ítem del menú en un menú navegable. Seleccionamos el texto y el fondo del botón del medio (que está en estado over). Desde la barra de herramientas contextual, elegimos **[Convert Artwork to Component/**

CATALYST



03 SELECCIONAR



05 DUPLICAR BOTÓN



MENÚ INTERACTIVO 06

04 MENÚ DESPLEGABLE

Button. En el panel de la derecha, renombramos ambas capas (label y over), ya que reutilizaremos el botón para los otros dos estados del menú. Finalmente, ocultamos la capa **over** para el estado **up** y **disabled**.

PASO | 05 Duplicamos el botón copiándolo y pegándolo dos veces, y reemplazando los dos textos del menú restantes (Localizaciones y Galería) con las nuevas instancias. En el panel de capas, abrimos la carpeta **Navigation** y los renombramos a **Tour_btn**, **Localizaciones_btn** y **Galería_btn**. Seleccionamos cada uno de los botones haciendo clic derecho del mouse y elegimos **[Share to State/All States]**. De esta manera, nos aseguramos de que estén visibles en las tres páginas de nuestra web.

PASO | 06 Vamos a darle interactividad al menú de navegación. Hacemos clic derecho

en el botón **[Localizaciones/Interactions/Add Interaction]**. Seleccionamos la acción **[onClick]** del primer desplegable, **[Play Transition to State]** del segundo desplegable y el estado **[Localizaciones]** en el tercero. Desmarcamos el **checkbox enabled** en el panel de **[Properties]** del botón. Repetimos el mismo procedimiento para los otros dos botones.

RESUMEN

En este sencillo apartado gráfico, hemos aprendido a crear un sitio con dos opciones de menú, las cuales nos mostrarán información basada en texto y en un mapa interactivo, para acceder a algunos detalles.

SWFObject

Para mostrar cualquier contenido de Flash en una página web, debemos insertarlo en el HTML base. Es por eso que SWFObject se ha convertido en la librería alternativa más importante para realizar esta tarea.

Para mostrar contenido Flash en una página web, tradicionalmente se lo montaba en **HTML** sobre **tags objects** (si se trata de Internet Explorer) o en forma **embed** (para el resto de los navegadores basados en Mozilla). A partir de un conflicto por patentes con la empresa Eolas Technologies, Internet Explorer debió deshabilitar las películas de Flash, que debían ser previamente cliqueadas para estar activas. En otras palabras, hasta no hacer clic sobre la película, no se podía interactuar con ella. Si bien este problema solo afectaba a Internet Explorer, en ese momento este era el principal navegador en cuanto a cantidad de usuarios, por lo que la empresa desarrolladora de Flash debió tomar cartas en el asunto. Entonces, para poder mostrar el contenido Flash en una página HTML de Internet Explorer, ya no era suficiente con usar los tags, sino que era preciso escribirlos dinámicamente con JavaScript. La plantilla estándar que utilizaban Flash y Dreamweaver incluía un **archivo externo en JavaScript** que escribía dinámicamente los tags necesarios; de este modo, se evitaba tener que hacer clic sobre la película para activarla en el navegador. Sin embargo, surgió el siguiente inconveniente: muchas personas se olvidaban de subir el archivo JavaScript que contenía la rutina necesaria, y entonces la película Flash no aparecía en absoluto. Luego de la primera versión, Adobe (que compró Macromedia ese mismo año) sacó otra que ya no requería incluir un archivo externo, para evitar los problemas que surgían al olvidarse de subirlo, pero la opción de SWFObject ya era bastante popular.

CÓDIGO 1

```
<object classid="clsid:D27CDB6E-AE6D-11cf-
96B8-444553540000" width="780" height="420">
  <param name="movie" value="myContent.swf" />
  <!--[if !IE]>-->
    <object type="application/x-shockwave-flash"
data="miContenido.swf" width="780" height="420">
      <!--<![endif]>-->
      <p>Contenido Alternativo</p>
    <!--[if !IE]>-->
  </object>
  <!--<![endif]>-->
</object>
```

Desde <http://code.google.com/p/swfobject> podemos descargar la versión 2.2 de SWFObject.



swfobject

SWFObject is an easy-to-use and standards-friendly method to embed Flash content, which utilizes one small JavaScript file

Project Home Downloads Wiki Issues Source
Summary | Updates | People

<swfobject>

Getting started

Welcome! You'll probably want to start with:

- [SWFObject documentation](#)
- [Detecting Flash Player versions and embedding SWF files with SWFObject 2](#) (beginners level tutorial)
- [Providing alternative content for SWF files](#) (beginners level tutorial)

Please **DO NOT** link directly to the JavaScript files in our SVN repository. Either download swfobject_2_x.zip from the [Downloads](#) section and place a copy of the SWFObject 2 script on your own web server or link to a hosted version of the script provided by the [Google AJAX Libraries API](#).

News

SWFObject 2.2 is available for download! It's good, go [get it](#)! You can read more about it in [SWFObject 2.2 - What's new?](#)

Questions, bugs

IMPLEMENTACIÓN DE SWFOBJECT

Esta biblioteca ha pasado por varias versiones; la última disponible desde <http://code.google.com/p/swfobject> es la 2.2. Es importante aclarar que su uso ha cambiado bastante desde la popular 1.5, sobre todo, en cuanto a la sintaxis que se emplea. SWFObject se utiliza no sólo para películas generadas desde Flash, sino que también existen versiones orientadas a insertar contenido en HTML5 y Flex. Desde la página de Google, donde está el proyecto, podemos acceder a una amplia documentación al respecto. Los objetivos de utilizar JavaScript para poder visualizar el SWF son los siguientes:

- Detectar la versión de Flash Player para evitar que otras antiguas del plugin estropeen el contenido Flash, y determinar si el contenido alternativo se debe mostrar o no.
- Ofrecer funcionalidad para volver al contenido alternativo en caso de que la versión del plugin no esté actualizada.
- Ofrecer la opción de utilizar la instalación express de Adobe para descargar la última versión de Flash Player.
- Resolver el problema producido por previos Webkits que ignoraban los elementos param anidados dentro del elemento embed.
- Dar la posibilidad de publicar contenido Flash con JavaScript y, en consecuencia, evitar posibles mecanismos que requieran hacer clic para activar.
- Ofrecer una elaborada API JavaScript que permite efectuar tareas comunes con Flash Player y el contenido Flash.

La biblioteca brinda dos métodos de publicación: dinámico y estático. El primero



PARÁMETROS SWFOBJECT

Para utilizar SWFObject, el comando que se vaya a incluir acepta cinco parámetros obligatorios y cuatro opcionales.

COMANDO	DETALLE
► swfUrl (String, requerido)	► Especifica la URL del SWF.
► Id (String, requerido)	► (String, requerido) Especifica el id del elemento HTML que incluye el contenido alternativo y que será reemplazado por el archivo Flash.
► width (String, requerido)	► Especifica la anchura del SWF.
► Height (String, requerido)	► Especifica la altura del SWF.
► Versión (String, requerido)	► Especifica la versión de Flash Player para la cual fue publicado el SWF (el formato es: "major.minor.release").
► expressInstallSwfurl (String, opcional)	► Especifica la URL del SWF para la instalación automática con Express Install.
► Express install	► Sólo se ejecuta la primera vez que es invocado. Únicamente lo soportan Flash Player 6.0.65 o versiones posteriores.
► Flashvars (String, opcional)	► Especifica las variables flash (flashvars) con parejas name:value.
► Params (String, opcional)	► Especifica los elementos param anidados. Bajo el elemento object con parejas name:value.
► Attributes (String, opcional)	► Especifica los atributos del objeto con parejas name:value.

utiliza etiquetas estándar, y permite que dispositivos sin JavaScript o navegadores que lo tengan deshabilitado muestren igualmente el contenido Flash. En el **Código 1** vemos un ejemplo del método estático. En él se incluye contenido alternativo por si el navegador no es capaz de mostrar el objeto porque carece de Flash Player. Además de los parámetros especificados, es posible recurrir a otros adicionales:

- **play**: si se inicia la reproducción automáticamente, por defecto, true.
- **loop**: si se repite la película, por defecto, true.
- **menu**: si se muestra el menú contextual, por defecto, true.
- **quality**: calidad de reproducción: low, high, autolow, autohigh, best.
- **scale**: posibles valores, showall (por defecto), noborder, exactfit, noscale.
- **align**: alineación, posibles valores l, t, r, tl, tr.
- **wmode**: fondo opaco o transparente.
- **bgcolor**: color de fondo.
- **base**: directorio para resolver

rutas relativas a archivos.

- **swliveconnect**: si se puede conectar con otros SWF en el mismo HTML a través de localconnection.
- **flashvars**: variables definidas por el usuario.
- **allowscriptaccess**: si el SWF puede invocar funciones de JavaScript en el HTML contenedor.
- **allowfullscreen**: si se permite pantalla completa.

El método dinámico es en el que, realmente, la librería se pone en juego, utilizando JavaScript para escribir el contenido según las necesidades. Para aplicarlo, debemos incluir la librería que se encuentra en un archivo JS externo y, luego, registrar el contenido Flash con la librería y decirle qué hacer con él. Para insertar contenido en forma dinámica, hay que realizar los siguientes pasos: en primer lugar, crear contenido alternativo utilizando etiquetas estándar; luego, incluir la librería en el HEAD del HTML; y finalmente, incrustar el SWF. ◀

Search projects

Activity: High

Code license:
MIT License

Featured downloads:

swfobject_2_2.zip
swfobject_generator_1_2_air.zip
swfobject_generator_1_2_html.zip
Show all »

Featured wiki pages:

api
documentation
faq
hosted_library
Show all »

Feeds:

Project feeds

Groups:

SWFObject discussion group

PERSONALIZAR MENÚ CONTEXTUAL

La posibilidad de modificar el menú contextual permite darle una fuerte personalización a nuestro sitio o herramienta. Veamos cuáles son las opciones que debemos aplicar de acuerdo con las necesidades del proyecto.

La opción de personalizar el menú contextual fue siempre un reclamo de la comunidad hacia Macromedia, demanda que se materializó con la clase **ContextMenu** en AS2, que finalmente se mantuvo en AS3. La clase **ContextMenu** puede ser invocada sobre cualquier elemento que forme parte del **DisplayList**, incluyendo el objeto **Stage**, desde el cual se crean todos los demás objetos visibles. Incluso, es posible determinar distintos menús contextuales para diferentes objetos que tengamos sobre el escenario. Esto nos da una pauta acerca del grado de personalización que podemos lograr. También es importante cuando estamos creando alguna aplicación en la cual queremos que las opciones de interacción se elijan desde el menú contextual, ya que podemos ir construyendo las



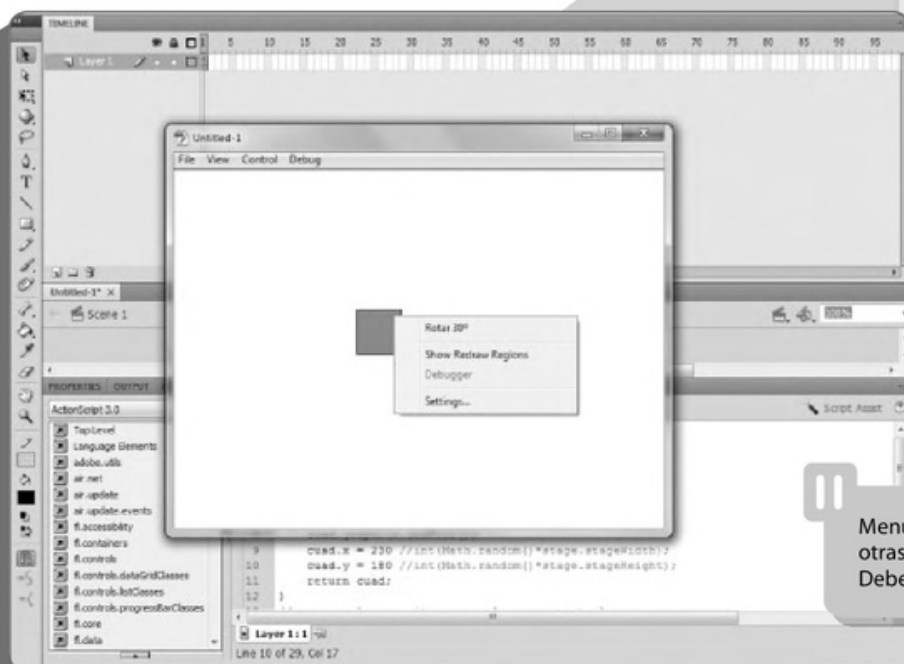
MENÚ CONTEXTUAL OCULTO

Uno de los temas más frecuentes en los foros de discusión es cómo hacer para ocultar todos los elementos del menú contextual. Si hacemos la prueba, veremos que Settings y About Flash Player son elementos que no se ocultan, incluso, con `hideBuiltInItems()`. Esto es así porque Adobe no quiere "hacer desaparecer" las opciones para personalizar Flash Player, y quiere dejar su copyright en "Acerca de ...". A partir de herramientas de terceros, es posible evitar esto en proyectores (archivos ejecutables), pero en cualquier película que subamos a Internet, será imposible ocultar todos los elementos del menú contextual.

distintas alternativas en la medida en que el contexto y la selección cambien.

AGREGAR UNA ACCIÓN AL MENÚ

Veamos un pequeño ejemplo de personalización, agregando una acción a un menú contextual. Lo primero que debemos tener es algún elemento sobre el cual hacer clic derecho. Usaremos el **código de la Figura 1** para crear un simple cuadrado en el escenario. Esta función dibuja un cuadrado celeste y lo pone en el escenario en una posición al azar. Reutilizando una función que crea el elemento (que para el ejemplo no es algo significativo), podremos crear un menú contextual para el elemento A y otro para el elemento B, muy fácilmente. Solo tenemos que crear una instancia del menú contextual, agregarle los elementos que queremos y asignársela al elemento en que lo necesitemos. Veamos en el **código de la Figura 2** cómo añadir interacción a un elemento recién creado con la función que hemos listado. En este **código**, lo primero que hacemos es instanciar el objeto **ContextMenu**, donde definiremos los elementos que queremos agregar. En este caso, se trata de una rutina para girar el elemento 30 grados, al cual definimos en una instancia de **ContextMenuItem** y asociamos a la función **rotar** a través de su evento **ContextMenuEvent.MENU_ITEM_SELECT**. De este modo, para agregar cualquier elemento, tenemos que realizar



Menú contextual de rotación, entre otras opciones de configuración. Debemos rotar el elemento 30 grados.

```

1 //creamos el objeto y lo agregamos al escenario
2 function creaCuadrado():Sprite{
3     var cuad:Sprite = new Sprite();
4     stage.addChild(cuad);
5     cuad.graphics.lineStyle(1,0);
6     cuad.graphics.beginFill(0x0099CC);
7     cuad.graphics.drawRect(-25,-25,50,50);
8     cuad.graphics.endFill();
9     cuad.x = int(Math.random()*stage.stageWidth);
10    cuad.y = int(Math.random()*stage.stageHeight);
11    return cuad;
12 }

```

FIGURA 1

```

13 //Primer cuadrado.
14 var menuCuad:ContextMenu=new ContextMenu;
15 var itemRotar:ContextMenu=new ContextMenuItem("Rotar 30°");
16 menuCuad.addEventListener(ContextMenuEvent.MENU_ITEM_SELECT,rotar);
17 menuCuad.hideBuiltInItems();
18 var cuadrado:Sprite = creaCuadrado();
19 cuadrado.contextMenu=menuCuad;
20 //Segundo cuadrado
21 var menuCuad2:ContextMenu=new ContextMenu();
22 var copyright:ContextMenu=new ContextMenuItem("Creado por Juan Perez");
23 menuCuad2.hideBuiltInItems();
24 menuCuad2.customItems.push(itemRotar);
25 menuCuad2.customItems.push(copyright);
26 var cuadrado2:Sprite = creaCuadrado();
27 cuadrado2.contextMenu=menuCuad2;

```

FIGURA 2

```

28 //Segundo cuadrado
29 var menuCuad2:ContextMenu=new ContextMenu();
30 var copyright:ContextMenu=new ContextMenuItem("Creado por Juan Perez");
31 menuCuad2.hideBuiltInItems();
32 menuCuad2.customItems.push(itemRotar);
33 menuCuad2.customItems.push(copyright);
34 var cuadrado2:Sprite = creaCuadrado();
35 cuadrado2.contextMenu=menuCuad2;
36
37 function rotar(event:ContextMenuEvent):void
38 {
39     event.contextMenuOwner.rotation += 30;
40 }

```

FIGURA 3

```

41 item = new ContextMenuItem("Imagen con Copyright");
42 break;
43 }
44 cm.customItems.push(item);
45 cm.hideBuiltInItems();
46 return cm;
47 }
48 foto0.contextMenu = setContext(fotos[0]);
49 foto1.contextMenu = setContext(fotos[1]);
50 foto2.contextMenu = setContext(fotos[2]);
51
52 function downloadImage(evt:Event){
53     trace("Download imagen");
54 }
55
56 function contactAuthor(evt:Event){
57     trace("Contactar autor");
58 }

```

FIGURA 4

tres pasos: instanciar ContextMenuItem, asociarlo a alguna función que se ejecute al seleccionarlo y agregarlo al array de CustomItems del ContextMenu creado. En la función rotar hemos usado la referencia que nos pasa el evento, de modo

que si queremos reutilizarla para otro menú, no habrá problemas por tener referencias estáticas hacia algún elemento. Como paso final, declaramos la propiedad contextMenu del elemento creado como el menú contextual que hemos

EL MENÚ CONTEXTUAL OFRECE ALTERNATIVAS FUNDAMENTALES PARA LA NAVEGACIÓN DEL USUARIO.

instanciado. Ahora, vamos a generar un segundo elemento y otro ContextMenu, el cual, además del método del primero, tiene un Copyright agregado (Figura 3). En este ContextMenu hemos agregado la opción de girar del menú anterior. Los ContextMenuItem se pueden reutilizar siempre y cuando las funciones asociadas a ellos sean reutilizables. Además, añadimos un ítem nuevo en el cual mostramos el copyright. Esto resulta bastante útil cuando distribuimos componentes personalizados, donde, incluso, podemos agregar un link a nuestra web.

CONDICIONAR EL MENÚ CONTEXTUAL

El menú contextual se puede manipular como cualquier otro objeto que utilicemos en el código. Muchas veces queremos mostrar una cosa u otra, dependiendo de determinadas condiciones. Por lo general, lo haríamos utilizando un **if...else**, y en el caso del menú contextual, no hay mucha diferencia. ¿Qué sucedería, por ejemplo, si tuviéramos una serie de fotos, algunas libres, otra con derechos negociables y otras con derechos exclusivos? Supongamos que el cliente nos encarga hacer un álbum con esas imágenes y que el usuario debe poder descargarlas bajo los siguientes supuestos:

- Si es una foto libre, se permite bajarla.
- Si es una foto con derechos negociables, hay que contactar al autor.
- Si es una foto con derechos reservados, es necesario mostrar el copyright.

Para realizar estas acciones, tendríamos que hacer un menú contextual por cada imagen, de manera que, dependiendo del tipo seleccionado, apareciera una u otra opción. En el ejemplo de la Figura 4, las funciones asignadas al menú para **downloadImage** y **contactAuthor** permiten ilustrar el ejemplo, pero en un caso real, basta con implementarlas. En el caso del copyright, no le ponemos ninguna llamada a función porque solo queremos que aparezca el texto al hacer clic derecho. ◀

CÓMO PROTEGER FLASH

Desarrollar un sitio en Flash no nos protege automáticamente de los problemas de seguridad en Internet; es preciso tener en cuenta algunas precauciones especiales.

Cuando hablamos de seguridad en Flash, debemos partir de la base de que el formato en sí no es seguro. El archivo SWF, aun siendo binario, es fácil de descompilar para acceder a su código fuente. Esto se puede hacer a través de los descompiladores o herramientas que permiten tomar un SWF y extraer todos los elementos o rearmar el archivo origen FLA. Esta ingeniería inversa es posible debido a que SWF es un formato abierto, es decir que cualquiera puede acceder a sus especificaciones; por lo tanto, es relativamente simple construir una herramienta que identifique los elementos y recomponga el original. Entonces, partiendo de la base de que nuestro código fuente puede ser leído si se descompila el SWF, siempre debemos tomar algunos recaudos:

- ▶ No incluir en el SWF nombres de usuarios o contraseñas.
- ▶ Evitar direcciones web a las que hagamos peticiones.
- ▶ No exponer algoritmos de generación de información.
- ▶ Si asignamos datos sensibles a alguna variable, evitar nombres obvios, como, **tarjDeCreditoNr**.
- ▶ Si utilizamos funciones que manejan información sensible, no usar nombres simples, como **enviarTarjeta**.

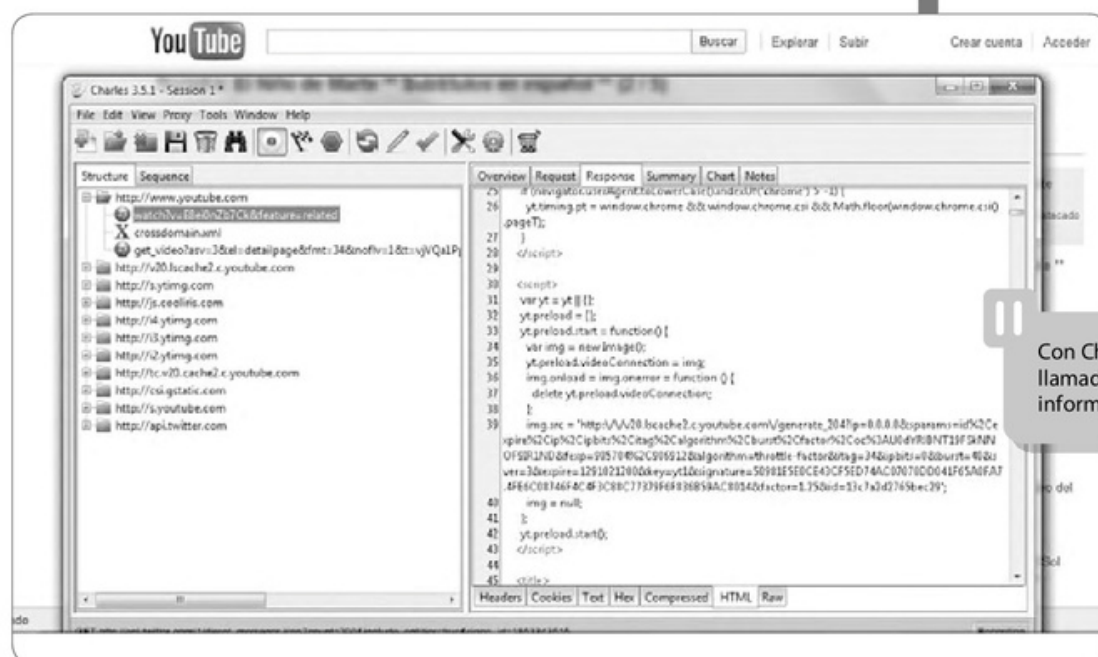
DATOS EN LA RED

El acceso al código fuente es solo una de las formas en que nuestro código está en riesgo. Otra manera de exposición de datos sensibles se da cuando la información

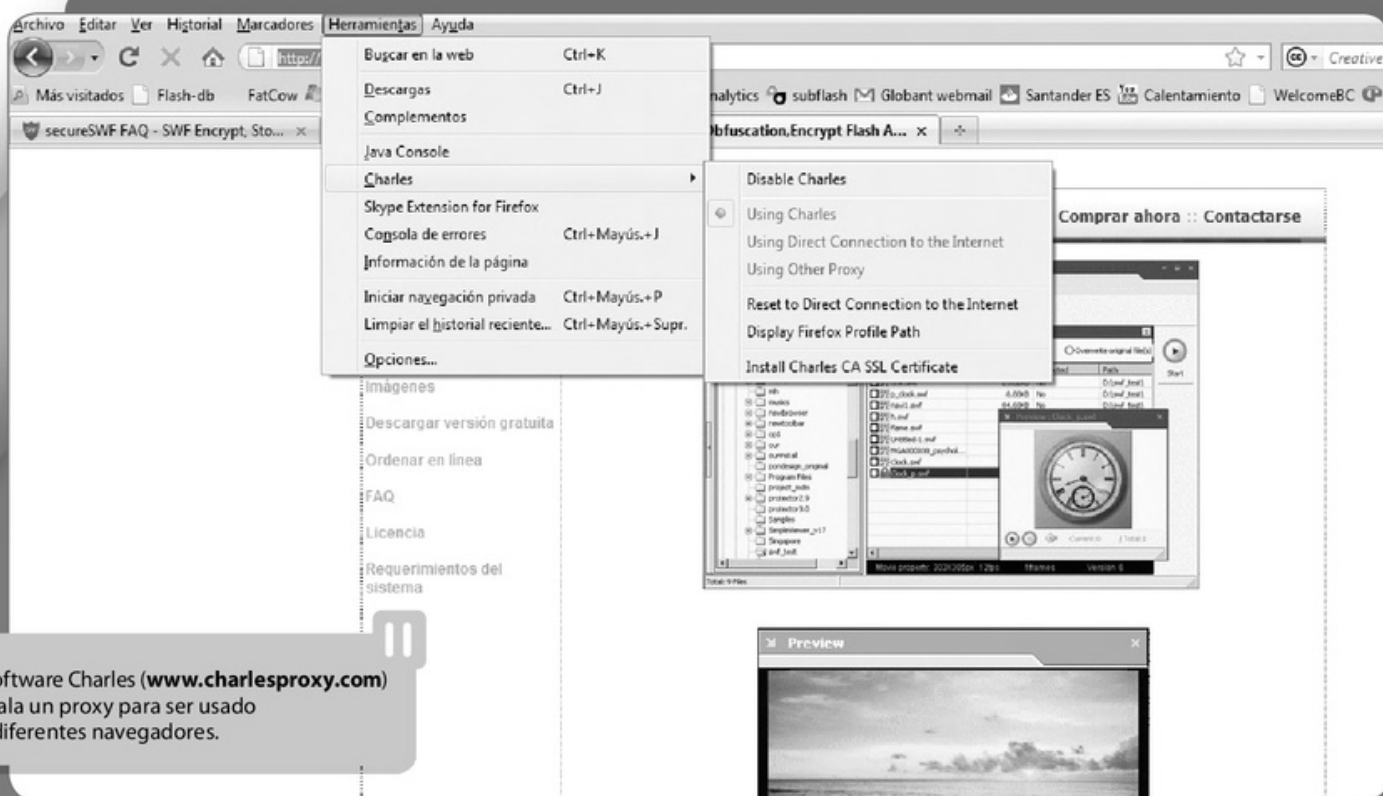
es enviada o recibida desde el servidor. Dado que la comunicación se realiza por el mismo protocolo/puerto (HTTP puerto 80), resulta bastante fácil analizar los paquetes que la película Flash manda y recibe.

Con un simple sniffer (herramienta de análisis de paquetes HTTP), es posible inspeccionar todos los datos que circulan. Para evitar problemas, entonces, debemos tener en cuenta algunas medidas preventivas:

- ▶ Nunca enviar información sensible al servidor como texto plano.
- ▶ Evitar logins de usuario y contraseñas sin encriptar.
- ▶ Nunca mandar información de puntaje o similar sin un sistema de ticket o token.



Con Charles, vemos una serie de eventos llamados al back-end, que sirven la información del player.



Otro problema con los archivos SWF es que cualquiera puede descargarlos y colgarlos en su propio sitio. Para evitar esta situación, hay que revisar desde el mismo SWF en qué dominio se está mostrando y tomar las acciones correspondientes si no es uno autorizado. Tenemos que prestar atención a la forma en que es cargado, porque si se incluye el SWF directamente desde el servidor, el chequeo que se hace desde Flash falla (a través de la propiedad loaderInfo.url), porque reporta el sitio desde donde es cargado. En este caso, el chequeo se complica y hay que agregar un script de servidor al que Flash llame para certificar el dominio.

TÉCNICAS DE OFUSCACIÓN

Dado que es imposible evitar que un archivo SWF se descompile, existen herramientas para ensuciar el código, con lo cual se vuelve ilegible si se lo descompila. Es importante entender que ofuscación no es lo mismo que encriptación. Si el código estuviera totalmente encriptado, sería completamente ilegible, pero hay un problema importante común a toda información encriptada: debe dejar la clave de encriptación junto con toda la información encriptada. Esto hace que crear una herramienta que desencrpte todo el código sea una tarea relativamente simple, dejando otra vez todo el código como texto plano. Por lo general, las herramientas de

EL ARCHIVO SWF, AUN SIENDO BINARIO, ES FÁCIL DE DESCOMPILAR PARA ACCEDER A SU CÓDIGO FUENTE.

ofuscación cambian los nombres de variables y métodos, hacen indirectas las llamadas, y modifican el indexado y los espacios entre líneas. Para ofuscar código fuente, contamos con una decena de herramientas que nos permiten hacerlo, algunas de las cuales son:

- SWFEncrypt (www.amayeta.com)
- Flashincrypt (www.flashincrypt.com)
- SecureSWF (www.kindisoft.com)

Casi todas tienen una interfaz para seleccionar el archivo SWF que se quiere tratar y diferentes opciones de configuración.

TÉCNICAS DE SNIFFING

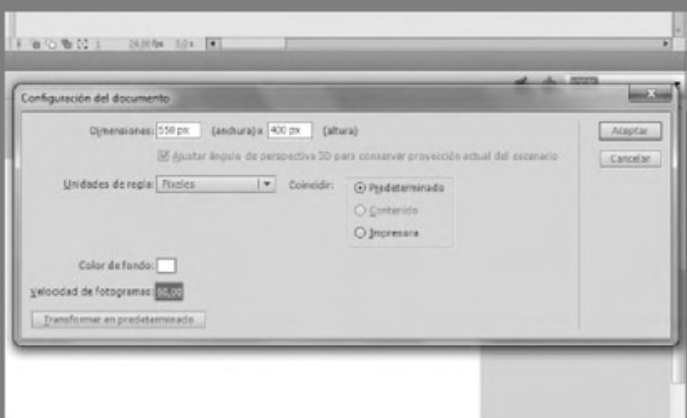
En el caso del denominado sniffing (espíar datos), que es el análisis de paquetes vía HTTP, la posibilidad de encriptación reside a nivel de la capa de transporte, usando el protocolo HTTPS seguro, que incluye encriptación de 32 bits a nivel del navegador. Si quisiéramos intercambiar cualquier tipo

de información sensible con el servidor, deberíamos usar HTTPS e, incluso, si quisiéramos que cualquier comunicación se hiciera con seguridad, siempre deberíamos emplear HTTPS. Pero este protocolo no viene sin costo, dado que cualquier comunicación de ida y vuelta tiene que ser encriptada/desencriptada, y esto agrega tarea al procesador. ¿Cómo se puede observar el tráfico? Es muy simple; veamos un ejemplo con Charles, que se puede descargar desde www.charlesproxy.com. Una vez descargado e instalado, lo iniciamos y podremos apreciar su tradicional icono de jarrón en el menú de programas. Charles instala un proxy para ser usado en diferentes navegadores. Una vez que estamos seguros de que Charles está corriendo y que lo hemos habilitado en el navegador, podemos ir a cualquier página con contenido Flash y comenzar a espiar las comunicaciones. Tomemos como ejemplo a YouTube. Si accedemos a esta red social y cargamos cualquier video, podremos ver una serie de eventos de llamados al **back-end**, que sirven la información requerida por el player para funcionar. ¿Qué pasaría si analizáramos cada una de las llamadas y empezáramos a simularlas por nuestra cuenta? Posiblemente, podríamos simular que somos el player y conseguir cantidad de información sobre los videos almacenados. ◀

AGREGAR ELEMENTOS INTERACTIVOS

PASO A PASO

Una opción interesante para entretener a los visitantes y darle un nuevo valor agregado a nuestra página web es incorporar elementos interactivos.



01 | ENTORNO DE TRABAJO



CREAR MOVIE CLIP | 02

Si bien nuestra web contará con interactividad desde los botones hasta las transiciones (si carece de interactividad es porque estamos haciendo algo mal), es conveniente incluir estos pequeños detalles que darán al usuario una experiencia aún más entretenida. ▶

PASO | 01 Lo que vamos a hacer es crear nuestro nuevo proyecto, en este caso, con un detalle distinto de los demás. El framerate de la película será de 60 cuadros por segundo, lo que le dará mayor fluidez a la animación. Cuando creamos el documento nuevo, en la parte inferior de la ventana veremos la opción **[Frame Rate]**, donde cambiamos el valor a 60.

PASO | 02 Vamos a crear el movie clip que haremos rebotar en breve. Conseguimos la imagen de una pelota y la importamos al proyecto. La convertimos en movie clip con el nombre "pelota" y alineamos la imagen al centro del clip para que sea nuestro punto de referencia.

PASO | 03 Ahora vamos a ocuparnos de las variables. **VelocidadMaxima** nos servirá solo



```

1 import com.greensock.*;
2 import com.greensock.easing.*;
3
4 var velocidadMaxima:Number = 10;
5 var radio:Number = 50;
6 var vx:Number;
7 var vy:Number;
8 var pelota:Sprite;
9 var anteriorMouseX:Number=0;
10 var anteriorMouseY:Number=0;
11 var mouseActualX:Number;
12 var mouseActualY:Number;
13 var liberoX:Number;
14 var liberoY:Number;
15

```

```

10 var anteriorMouseY:Number=0;
11 var mouseActualX:Number;
12 var mouseActualY:Number;
13 var liberoX:Number;
14 var liberoY:Number;
15
16 function juegapelota() {
17 }
18 function onClipEnterFrame(e:Event):void {
19 }
20 function pelotaDownHandler(e:MouseEvent):void{
21 }
22 function pelotaUpHandler(e:MouseEvent):void{
23 }
24

```

04 | PRIMERAS FUNCIONES

```

3 var radio:Number = 50;
4 var vx:Number;
5 var vy:Number;
6 var pelota:Sprite;
7 var anteriorMouseX:Number=0;
8 var anteriorMouseY:Number=0;
9 var mouseActualX:Number;
10 var mouseActualY:Number;
11 var liberoX:Number;
12 var liberoY:Number;
13
14 function juegapelota() {
15     addEventListener(Event.ENTER_FRAME, onClipEnterFrame, false, 0, true);
16 }
17
18 function onClipEnterFrame(e:Event):void {
19     trace("SE EJECUTA EL LOOP >> onClipEnterFrame");
20 }
21
22 function pelotaDownHandler(e:MouseEvent):void{
23 }
24
25 function pelotaUpHandler(e:MouseEvent):void{
26 }
27
28 juegapelota();

```

EVENT ENTER FRAME | 05

```

1 import com.greensock.*;
2 import com.greensock.easing.*;
3
4 var velocidadMaxima:Number = 10;
5 var radio:Number = 50;
6 var vx:Number;
7 var vy:Number;
8 var pelota:Sprite;
9 var anteriorMouseX:Number=0;
10 var anteriorMouseY:Number=0;
11 var mouseActualX:Number;
12 var mouseActualY:Number;
13 var liberoX:Number;
14 var liberoY:Number;
15
16 function juegapelota() {
17     pelota.buttonMode=true;
18     pelota.addEventListener(MouseEvent.CLICK, pelotaDownHandler);
19     pelota.addEventListener(MouseEvent.CLICK, pelotaUpHandler);
20
21     addEventListener(Event.ENTER_FRAME, onClipEnterFrame, false, 0, true);
22 }

```

PELOTA Y EVENTOS | 06

en el inicio y representa la velocidad del primer rebote. Nosotros vamos a usar un valor de 10 para que ingrese con suavidad. Cuanto más alto sea este número, más fuerte será el primer rebote.

PASO | 04 El siguiente paso es declarar las funciones. Tratemos de pensar cuáles serán necesarias durante el proyecto; luego podremos agregar más, pero esta será una buena práctica para agilizar nuestra mente.

PASO | 05 Para que el proyecto funcione como corresponde, debemos tener un evento que esté permanentemente chequeando la posición de la pelota y la acción que hagamos sobre ella. Es por eso que dentro de la función "juegapelota", agregamos la línea de código que vemos en la pantalla.

PASO | 06 Lo primero que vamos a hacer es programar lo necesario para poder desplazar la pelota por la pantalla, que se consigue gracias al método Drag and Drop (arrastrar y soltar). En la función "juegapelota" agregamos los eventos requeridos que se muestran en pantalla.

AGREGAR ELEMENTOS INTERACTIVOS

PASO A PASO

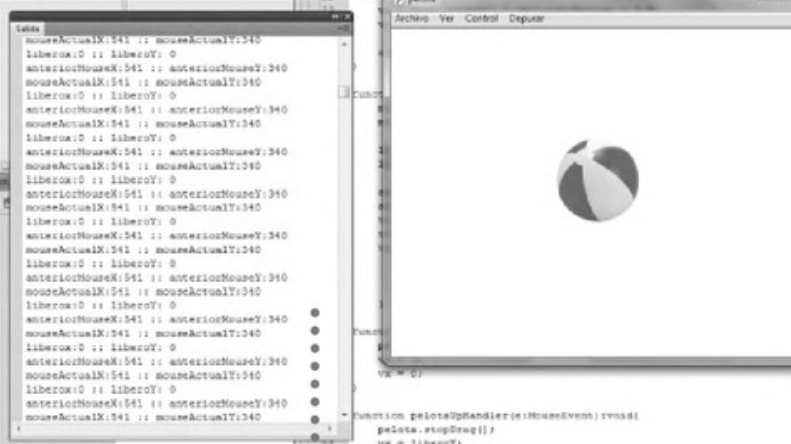
```

20
21     addEventListener( Event.ENTER_FRAME, onClipEnterFrame );
22 }
23 function onClipEnterFrame(e:Event):void {
24
25 }
26 function pelotaDownHandler(e:MouseEvent):void{
27     pelota.startDrag(false);
28 }
29 function pelotaUpHandler(e:MouseEvent):void{
30     pelota.stopDrag();
31 }
32 juegapelota ()
33

```

Capa 1:1
línea 26 de 33, Col 1

07 | ARRASTRAR PELOTA



REFERENCIAS | 08

```

mouseActualX = this.mouseX;
mouseActualY = this.mouseY;
liberoX = mouseActualX - anteriorMouseX;
liberoY = mouseActualY - anteriorMouseY;
anteriorMouseX = mouseActualX;
anteriorMouseY = mouseActualY;

```

```

51     pelota.stopDrag();
52     vx = liberoX;
53     vy = liberoY - 0.6;
54 }
55
56 juegapelota()

```

Layer 1:1
Línea 42 de 56, Col 1

09 | ANIMACIÓN

PASO | 07 El próximo paso es completar el código de las funciones `pelotaDownHandler` y `pelotaUpHandler`.

Lo que lograremos con esto es tomar la pelota presionando el botón del mouse para moverla y dejarla en su nueva posición cuando lo soltemos.

PASO | 08 Vamos a agregar dentro de la función `onClipEnterFrame` el código que vemos en la imagen. En pantalla vemos la salida.

Con este bloque de código, tomamos una referencia de la posición actual del mouse (ejes X e Y). La resta de la variable `mouseActualX - anteriorMouseX` nos dará como resultado un valor negativo o positivo. Así, sabremos la dirección de lanzamiento de la pelota. Guardamos un historial de la posición actual del mouse; de esta manera, el contenido de la variable "anterior..." será la última referencia que se tomó del mouse.

PASO | 09 Haremos que la pelota se mueva por cuenta propia. Dentro de la función `pelotaDownHandler` agregamos lo siguiente:

vy = 0; vx = 0; en `pelotaUpHandler` incorporamos: **vx = liberoX; vy = liberoY - 0.6;** en la función `OnClipEventFrame` vamos a poner: **pelota.x += vx; pelota.y += vy;** nos daremos cuenta de que la pelota no solo no rebota, sino que tampoco desacelera ni desaparece tras los límites del SWF. Los movimientos de rebote, aceleración y desaparición detrás de los límites del SWF se verán en los próximos pasos.

PASO | 10 Vamos a decirle a la pelota cuáles son los límites en donde debería de rebotar, y en el momento de rebotar, qué tendría que hacer. Para esto, agregamos las líneas que vemos en pantalla a la función `onClipEventFrame`. Lo que estamos haciendo es marcar los límites de stage con los dos primeros condicionales (if).

```

36 anteriorMouseY = mouseYActualY;
37
38 pelota.x += vx;
39 pelota.y += vy;
40
41
42
43 if ( ( pelota.x <= radio && vx < 0 ) || ( pelota.x >= stage.stageWidth - radio &&
44 vx = -vx * 0.6;
45 }
46 if ( ( pelota.y <= radio && vy < 0 ) || ( pelota.y >= stage.stageHeight - radio &&
47 vy = -vy * 0.6;
48 }
49 if ( Math.abs( vx ) < 0.01 ){
50 vx = 0;
51 }
52 if ( Math.abs( vy ) < 0.1 && pelota.y >= stage.stageHeight - radio - 2 ){
53 vy = 0;
54 pelota.y = stage.stageHeight - radio;
55 }
56
57
58
59
60
61 function pelotaDownHandler(e:MouseEvent):void{
62

```

10 | REBOTE

```

36 anteriorMouseY = mouseYActualY;
37
38 pelota.x += vx;
39 pelota.y += vy;
40
41 if ( vy != 0 ) vy += 0.5;
42 if ( vx != 0 ) vx += (0-vx)*0.015;
43
44 if ( ( pelota.x <= radio && vx < 0
45 vx = -vx * 0.6;
46 }
47 if ( ( pelota.y <= radio && vy < 0

```

11 | GRAVEDAD

Con las otras dos condiciones, le quitamos movimiento a la pelota hasta lograr que deje de moverse sobre sus ejes.

PASO | 11

Si lo que queremos es entretener, aún nos faltan dos características: gravedad y rotación.

En la función `onClipEventFrame`, luego de la línea `pelota.y += vy`; vamos a agregar las que mostramos en pantalla. Verificamos el código y notaremos que es un poco más divertido.

PASO | 12

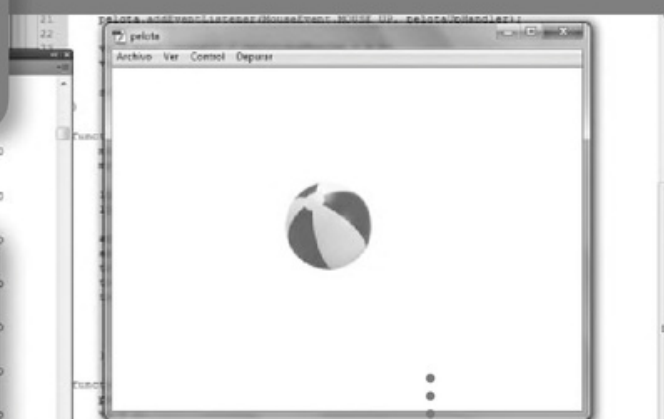
Para hacer rotar la pelota sobre su propio eje, tenemos que colocar el código en la función `pelotaUpHandler`; será sólo una línea. Lo que vamos a hacer con esta línea es rotar la pelota 20 grados y aprovechar el valor aleatorio que nos da "vy" para variar un poco esos grados de giro. Veamos el detalle en la captura de muestra.

```

67
68 function pelotaUpHandler(e:MouseEvent):void{
69     pelota.stopDrag();
70     vx = liberoX;
71     vy = liberoY - 0.6;
72     TweenMax.to(pelota, 2, {rotation:20*vy});
73 }
74
75 juegapelota()

```

ROTACIÓN | 12



INICIA | 13

```

pelota.alpha = 0;
TweenMax.to(pelota, 0.5, {alpha:1});

```

PASO | 13

Ahora sí parece una pelota real, y como no queremos que tenga una entrada tosca, vamos a agregar este código en la función `juegapelota`, para que entre con una transparencia.

RESUMEN

En este proceso, hemos logrado lanzar una pelota, hacer que rebote contra los límites del stage, que acelere y desacelere, que grave y rote. Todas estas acciones se consiguen desde las líneas de código.

ARCHIVOS TEMPORALES

Aunque parezca un contrasentido, los archivos temporales que guarda el navegador son, por un lado, beneficiosos, pero por el otro, resultan contraproducentes para la película Flash. Veamos por qué esto es así.

Los archivos temporales de los navegadores funcionan como un mecanismo creado con el fin de minimizar el tiempo de respuesta de la navegación por Internet. El sistema es simple: cuando cargamos una página web en el navegador, esta se guarda por cierto tiempo en el disco rígido de la computadora y, durante el tiempo que dure allí, no será descargada de Internet en la próxima visita.

LA CACHÉ Y FLASH

Por defecto, Flash utiliza la caché del navegador para mejorar una futura respuesta, sobre todo, teniendo en cuenta que este mecanismo surgió cuando los archivos SWF eran más grandes, y el ancho de banda, más pequeño. Cuando el contenido es persistente, genera algunos problemas, como los siguientes:

- Aun subiendo nuevas versiones, siempre vemos lo mismo.
- Las llamadas a scripts externos se

quedan en la caché y siempre nos devuelven los mismos datos.

► El contenido multimedia cargado externamente no puede evolucionar, ya que las versiones más antiguas son cargadas desde la caché.

Para evitar estos conflictos, existen varias opciones. Una de las más habituales es usar datos aleatorios en las llamadas; esto es, no repetirnos cuando llamamos a scripts. ¿Cómo lo hacemos? Agregando un número al azar. Así, por ejemplo, si tenemos que llamar a **www.misitio.com/miscript.php**, podemos hacer **www.misitio.com/miscript.php?15432340937483**.

En este caso, hemos agregado una variable en la URL, que añade un número aleatorio que no se repite. Entonces, los scripts funcionan como si siempre se tratara de una llamada nueva. De este modo, la caché no persiste y podemos acceder a nuevo contenido sin problemas. Si quisiéramos hacer lo contrario, es decir,

cargar lo que teníamos porque sabemos que no hay contenido nuevo, la llamada podría permanecer como está, pues cargar desde la caché es el comportamiento predefinido de Flash.

En el caso de scripts, otra forma de evitar la caché es hacer que los scripts manden los encabezados adecuados (por ejemplo, NO-CACHE), pero la implementación dependerá del lenguaje elegido. Es muy importante tener esto en cuenta a la hora del desarrollo, porque frecuentemente nos sucederá que subimos contenido nuevo y no logramos verlo (o peor aún, que nuestro cliente no lo ve). Es por eso que siempre debemos tener presente el tema de la caché e, incluso, recurrir a lo más simple de todo: cambiar la dirección o el nombre de página cada vez que subimos una nueva versión.

CARGAR MUCHAS VECES LO MISMO

Una duda bastante habitual es qué ocurre si mandamos a cargar muchas veces el mismo contenido. Esto puede suceder, por ejemplo, si en una galería de fotos paginadas vamos hacia adelante y hacia atrás. ¿El contenido carga de la caché o se vuelve a cargar al repetir frecuentemente un pedido? A partir de lo que hemos expuesto en estas páginas, podemos deducir que el contenido vendrá de la caché, no importa cuántas veces intentemos la carga, ya que el comportamiento por defecto de Flash es ese, a menos que por algún motivo, establezcamos otro. Otro aspecto que es necesario considerar en caso de querer usar la caché es que

+ CONOCIMIENTOS TÉCNICOS

Recordemos que, para mejorar y potenciar nuestro aprendizaje, desde **www.redusers.com/cursos/webdesign** podemos descargar los archivos editables de los proyectos que realizamos en esta colección. A partir de ellos podremos obtener ideas para un emprendimiento propio o, directamente, modificar los archivos para reutilizarlos.

En el sitio también encontraremos **guías visuales**, un **glosario de términos** y una completa guía de preguntas frecuentes sobre la suite CS5. Además, el libro **Desarrollo y animación en ActionScript 3.0** nos ayudará a comprender los fundamentos de la programación en Flash.

muchas veces hay un servidor proxy que espeja contenidos. Es decir, aunque borremos la caché de nuestro navegador, el contenido está copiado en el servidor, y no solo a nivel local, lo cual impide que un simple borrado de la caché del navegador solucione el tema. En ese caso, usar datos aleatorios para cambiar la URL solicitada es la respuesta.

HERRAMIENTAS DE TESTEO

Además de las herramientas que el mismo programa Flash ofrece, como el debugger, existen numerosos programas externos de los que nos podemos beneficiar a la hora del desarrollo.

XRAY

Es una aplicación que permite tomar una "foto instantánea" del estado de una aplicación sin impactar negativamente en la performance o el tamaño del archivo de la aplicación. La naturaleza de Xray es analizar en profundidad la aplicación y refutar los mitos que dicen que es imposible analizar el SWF en profundidad. El programa tiene dos partes que trabajan juntas: la interfaz (aplicación standalone) y el conector (que debe ser incluido en el SWF).

LOG4F

Con esta herramienta es posible habilitar el login mientras el SWF corre, sin necesidad de usar Flash. Los comandos de logging son habilitados o deshabilitados desde un archivo de configuración en XML. Para utilizar este recurso, tenemos que poner el archivo Log4f.as en nuestro classpath y, luego, agregar los comandos con esta estructura: Log4f(typ, header, body).

ALCON

Es una herramienta liviana para desarrolladores de ActionScript, que provee de una serie de métodos simples y rápidos para inspeccionar cualquier aplicación hecha en ActionScript 2 o ActionScript 3, desde el navegador, un proyector o una aplicación AIR. Brinda una manera de obtener información desde cualquier sitio, no solo cuando se corre en el programa Flash o Flex. Trae un Monitor de Aplicación que puede monitorear la velocidad de frames y el consumo de memoria, y un Inspector de Objetos para analizar cualquier objeto, array o clase, y sacarlos a un archivo de log.

FLASH TRACER

Es un plugin del popular navegador Firefox, que permite ver en una pestaña del navegador la salida producida por el

comando trace de Flash. Para utilizarlo es necesario tener Flash Player Debug. Hay que tener en cuenta que, según las versiones de Flash Player, la ruta del debug puede cambiar, pero para evitar problemas, se puede instruir a la extensión para que use una ruta específica. ◀

LOS ARCHIVOS TEMPORALES MINIMIZAN EL TIEMPO DE RESPUESTA DE LA NAVEGACIÓN POR INTERNET.

FlashTracer (plugin de Firefox) muestra la salida del comando trace en una pestaña.

Alcon provee de métodos simples para inspeccionar aplicaciones hechas en ActionScript 2 o ActionScript 3.

Con esta herramienta, habilitamos el login mientras el SWF corre, sin necesidad de usar Flash.

"Log4f". Tools for flash developers.

With "Log4f" it is possible to enable logging at runtime without IDE. Log4f statements can remain in shipped code and may be controlled at runtime from configuration file. Take a look at Log4f demo here.

Time	ID	Type	Header	Body
01:10:40 21 Jan	7157	DEBUG	IO RESPONSE	
01:10:40 21 Jan	7155	INFO	IO RESPONSE	
01:10:39 21 Jan	7154	INFO	IO RESPONSE	
01:10:39 21 Jan	7153	DEBUG	IO RESPONSE	
01:10:39 21 Jan	7152	INFO	IO RESPONSE	
01:10:39 21 Jan	7151	INFO	IO RESPONSE	
01:10:39 21 Jan	7150	INFO	IO RESPONSE	
01:10:39 21 Jan	7149	INFO	IO RESPONSE	
01:10:39 21 Jan	7147	INFO	IO RESPONSE	
01:10:39 21 Jan	7146	INFO	IO RESPONSE	
01:10:39 21 Jan	7144	INFO	IO RESPONSE	
01:10:39 21 Jan	7143	INFO	IO RESPONSE	
01:10:35 21 Jan	7142	INFO	IO RESPONSE	
01:10:35 21 Jan	7139	INFO	IO RESPONSE	

CÓMO APLICAR EFECTOS DE PARTÍCULAS

PASO A PASO

En este apartado aprenderemos a aplicar efectos de partículas con el seguimiento del mouse.



01 SELECCIÓN DEL FOTOGRAMA



VALORES DE ANIMACIÓN

02

Los efectos siguiendo el mouse son muy populares y de gran utilidad para adornar una interfaz de manera fácil y rápida. Hay mucha variedad, pero todos se basan en la posición del mouse como punto de partida. En este paso a paso, crearemos un efecto de partículas que se generan desde el puntero, para producir una ilusión de "lluvia de confeti" cada vez que hacemos clic en cualquier parte del escenario. ◀

PASO | 01

Vamos a generar el ejemplo de partículas que ilustra la imagen directamente desde código. Para lograrlo, creamos una película nueva en Flash. Vamos a **[Archivo/Nuevo/ActionScript 3.0]**, seleccionamos el único fotograma de la línea de tiempo y oprimimos **<F9>** para abrir el panel de acciones.

PASO | 02

Escribimos la función principal de la película para crear la partícula base de la animación. En las primeras 5 líneas generamos un color al azar; luego, se lo aplicamos al relleno de un círculo que situamos en las coordenadas del mouse. Finalmente, configuramos dos valores que serán utilizados en la animación: **degreeX**, basado en una dirección positiva o negativa al azar; y **eje Y**, que es el recorrido que hará la partícula antes de desaparecer.

PASO | 03

Ya tenemos la función principal que crea la partícula y le da los valores iniciales. Ahora tenemos que invocarla y hacerla mover, porque la función sólo genera partículas estáticas, pero asigna el evento **ENTER_FRAME** a una función llamada **Mueve**. Esta función, que se repite frame a frame, modifica las propiedades X e Y para darle movimiento al elemento, a la vez que decrece el alpha para que se vaya desvaneciendo. Una

```

19 addChild(bola)
20 }
21 function mueve (evt:Event):void{
22     var t: MovieClip = evt.target as MovieClip;
23     t.ejeY -= 1;
24     t.y -= t.ejeY;
25     t.x -= .04;
26     if (t.alpha <= 0)
27     {
28         t.removeEventListener (Event.ENTER_FRAME, mueve);
29         removeChild (t);
30     }
31 }
32

```

04 | DISPARAR PARTÍCULAS

```

3
4 var amplitud:Number =5;
5 var altura:Number =5;
6 var desaparición:Number = 0.01;
7
8 function creaBola (evt:Event):void{
9     var bola:MovieClip = new MovieCl
10     var red:uint = int (Math.random()
11     var green:uint = int (Math.random
12     var blue:uint = int (Math.random(
13     var color:Number = (red << 16| gr
14     bola.graphics.beginFill (color,1)

```

PARÁMETROS | 05

```

1 import flash.display.MovieClip;
2 import flash.events.Event;
3
4 var amplitud:Number =5; // desplazamiento de las bolas en el eje X desde su base
5 var altura:Number =5; // punto en el eje Y hasta el cual suben las bolas, valores negativos impiden que suba
6 var desaparición:Number = 0.01; // desde cero y uno, cuanto tarda en desaparecer
7
8 function creaBola (evt:Event):void{
9     var bola:MovieClip = new MovieClip();
10     var red:uint = int (Math.random()*255);
11     var green:uint = int (Math.random()*255);
12     var blue:uint = int (Math.random()*255);
13     var color:Number = (red << 16| green << 8 | blue);
14     bola.graphics.beginFill (color,1);
15     bola.graphics.drawCircle (0,0,5);
16     bola.graphics.modPath();
17     bola.x = mouseX-25;
18     bola.y = mouseY-25;
19     bola.rotation = Math.random()*2;
20     bola.rotation = (bola.rotation + 0.1*Math.random()*4) + 1*Math.random()*4;
21     bola.alpha = 1*Math.random()*100;
22     bola.addEventListener (Event.ENTER_FRAME, mueve);
23     addChild (bola);
24 }
25
26 function mueve (evt:Event):void{
27     var t: MovieClip = evt.target as MovieClip;
28     t.ejeY -= 1;
29     t.y -= t.ejeY;
30     t.x -= .04;
31     if (t.alpha <= 0)
32     {
33         t.removeEventListener (Event.ENTER_FRAME, mueve);
34         removeChild (t);
35     }
36 }

```

LISTADO COMPLETO | 06

vez que desaparece por completo, la función borra el ENTER_FRAME asociado a la partícula y, luego, la partícula misma.

PASO | 04 Agregamos entonces los eventos y las funciones auxiliares que utilizarán las partículas. El handler para el MOUSE_DOWN asocia la función creaBola a un ENTER_FRAME, es decir, se crearán bolas a la misma velocidad en FPS que tengamos para la película hasta que soltemos el mouse, ya que el evento MOUSE_UP borra el ENTER_FRAME.

PASO | 05 Si bien hemos completado el efecto, podemos colocar algunos datos para poder variarlo fácilmente, sin tener que tocar el código de manera interna. Los datos que podemos cambiar son: **amplitud**, que es el desplazamiento de las bolas en el eje X antes de caer; **altura**, el punto en el eje Y hasta el cual sube antes de bajar (los números negativos impiden que suba); y

desaparición, que toma valores entre 0 y 1, y determina cuánto tarda en desaparecer.

PASO | 06 Podemos experimentar cambiando los valores que definimos en el paso anterior. Para finalizar, observemos en pantalla cómo quedó el listado que incluye las variables nuevas.

RESUMEN

En este apartado práctico hemos aprendido a aplicar el efecto de partículas que aparecen cuando presionamos el botón del mouse y desaparecen al soltarlo. Es un excelente efecto para darles dinamismo a nuestros sitios.

GOOGLE ANALYTICS

PARA FLASH

Adobe, en colaboración con Google Analytics, desarrolló un componente especial para utilizar en Flash. Veamos de qué se trata esta aplicación y en qué condiciones podemos utilizarla para mejorar nuestros proyectos.

Adobe ha desarrollado un componente que genera todas las llamadas desde Flash que suelen hacerse a través de la biblioteca de JavaScript incluida en ga.js. De esta forma, Flash puede ejecutar el método de Analytics apropiado, como **trackPageview()** o **trackEvent()**. Para que el seguimiento pueda llevarse a cabo correctamente, es importante saber si Flash puede acceder vía JavaScript al DOM (*Document Object Model*) del navegador, porque de otra forma, fallaría por completo. Para esto, es fundamental entender cómo se maneja la clase **ExternalInterface** en AS3 y qué hacer en caso de que no se disponga de ella. El componente simplifica en gran medida la tarea y abarca los casos más habituales:

- ▶ Cuando se dispone de un **widget Flash** incrustado en una página HTML.
- ▶ Para las aplicaciones Flex independientes o los sitios alojados en páginas HTML que solo incluyen contenido Flash.
- ▶ En los juegos o programas comercializados con contenido Flex/Flash en los que el programador no puede controlar la ubicación del widget.

Es importante destacar que el componente no funciona con Adobe AIR o desde el Flash IDE.

CÓMO USARLO

Lo primero que debemos hacer es descargar el componente desde <http://code.google.com/p/gaforflash>. En la carpeta **lib** encontraremos el archivo **analytics_flash.swc**, la biblioteca que tenemos que

utilizar en Flash. Para crear un componente que aparezca en la librería y podamos arrastrar fácilmente al escenario, generamos una carpeta **Google** dentro de la de **Componentes** de Flash, que variará según el sistema operativo. En ella copiamos el archivo SWC, cerramos Flash si lo teníamos abierto y volvemos a abrirlo. Abrimos la librería de componentes y ya podremos arrastrar al escenario el de **Analytics**, que, en realidad, no tiene ninguna entidad visual, sino que solo incluye todas las clases necesarias. Si seleccionamos el componente y abrimos el panel desde el menú **[Ventana]**, podremos acceder a las configuraciones que este expone:

- ▶ **ID de propiedad web:** también se conoce como el número UA de nuestro código de seguimiento. Tiene el formato UA-xxxxx-yy, siendo x y los números de nuestra cuenta y la información de perfil para el objeto de seguimiento.
- ▶ **Modo de seguimiento:** selecciona el modo Bridge o AS3. Determina cómo se comunica el seguimiento con los servidores de Google Analytics.
- ▶ **Modo de depuración:** independientemente del entorno o del modo de seguimiento utilizado, podemos activar la depuración para validar y probar el seguimiento.

Google Analytics



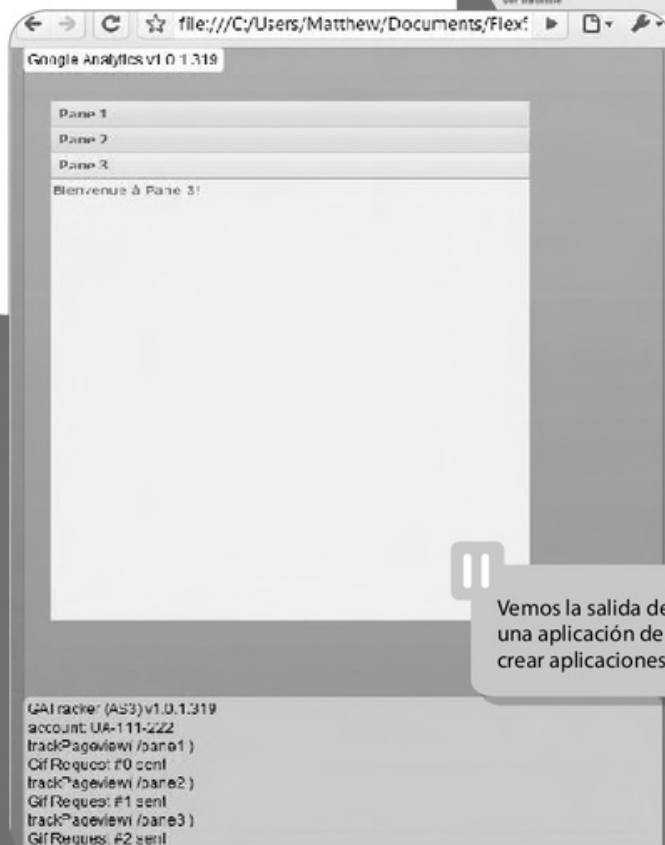
En la imagen vemos un típico reporte de Analytics. Este tipo de resultado lo podemos verificar en cualquier momento y desde cualquier lugar.

MODOS BRIDGE Y AS3

Según cómo se distribuya el contenido Flash, el componente Analytics se comunica con los servidores de Analytics tendiendo un puente de comunicación entre el contenido Flash en una instalación existente de seguimiento de Analytics o comunicándose directamente con los servidores de este. El modo **Bridge** es útil cuando controlamos la página HTML y el contenido Flash, y ya tenemos seguimiento con JavaScript, con lo cual solo necesitamos agregar el seguimiento en Flash. Por su parte, el **modo AS3** se emplea cuando hay control del código pero no de la página donde se aloja el SWF. Esto ocurre cuando utilizamos un componente distribuido, es decir, que podrá ser usado en muchos sitios, pero no sabemos a priori en cuáles. Este modo es in-



Vemos cuán detallado es un informe, con gráficos, usuarios, país, sitios de referencia y las páginas vistas.



Vemos la salida del debug habilitada en una aplicación de Flex. Flex nos permite crear aplicaciones orientadas a la Web.

dependiente del código de seguimiento **ga.js** y contiene toda la funcionalidad de seguimiento de Analytics, sin necesidad de una instalación de **ga.js** separada.

ACTIVAR DEBUG

El componente de Google Analytics tiene un modo de depuración que permite

GOOGLE ANALYTICS ES UN COMPONENTE QUE GENERA LAS LLAMADAS DESDE FLASH HECHAS A TRAVÉS DE LA BIBLIOTECA DE JAVASCRIPT.

validar y simplificar la solución de problemas. Si se activa, todos los datos de seguimiento se interceptan y se dirigen a una pantalla de un cuadro de texto en vez de a los servidores de Analytics. En este modo, podemos ver en tiempo real los datos que, de lo contrario, recogería el servidor. Esta función también ayuda a mantener los datos de evaluación separados de los de producción. Podemos habilitar la función de solución de problemas definiendo la opción **visualDebug** en **true** en el inspector de componentes.

PAGEVIEWS VS. EVENTOS

La API del componente soporta dos tipos de eventos: **PageViews** y **Eventos** (en un futuro cercano, soportará un tercer tipo, e-commerce transactions). En un sitio dinámico en Flash o en una aplicación, seguramente nos veríamos tentados a enviar siempre eventos, pero hay buenas razones para usar ambos. El **pageview** es la base sobre la cual se hacen los reportes en Google Analytics y es, sin duda, una de las unidades semánticas más importantes. Dentro de una aplicación en Flash, el nombre de PageView es un poco engañoso, porque los distintos estados se crean en forma dinámica sin recargar el navegador. Pero si extrapolamos este concepto de HTML y lo llevamos a las secciones de nuestra web o estados de nuestra aplicación, podremos crear una especie de mapa que tenga sentido. ◀

ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué son las **APIs**?

02 ¿Para qué sirve Youtube API?

03 ¿Qué es Google Maps?

04 ¿De qué forma debemos cargar coordenadas?

05 ¿Qué es Papervision 3D?

06 ¿Cómo podemos usar Flash API?

07 ¿Qué es Flash Catalyst?

08 ¿Qué es Swobject?

09 ¿De qué forma debemos agregar elementos interactivos?

10 ¿Cómo podemos aplicar efectos de partículas?

PRÁCTICAS

01 Utilice Youtube API.

02 Experimente con Facebook API.

03 Utilice Flash Catalyst.

04 Agregue elementos interactivos.

05 Aplique efectos de partículas.



Servidores y tecnología web

- ▶ CLIENTES Y SERVIDORES
- ▶ PROTOCOLOS
- ▶ DNS: SISTEMA DE NOMBRES DE DOMINIOS
- ▶ URL Y TIPOS DE DOMINIOS
- ▶ EL SISTEMA WEB Y SU FUNCIONAMIENTO
- ▶ ELEGIR UN NOMBRE DE DOMINIO
- ▶ ¿CÓMO REGISTRAR UN DOMINIO?
- ▶ SERVICIOS DE HOSTING
- ▶ SUBIR NUESTRA WEB A INTERNET
- ▶ SUBIR UN SITIO AL SERVIDOR



CLIENTES Y SERVIDORES

Para visitar un sitio, este debe encontrarse alojado en un servidor. Este es el equipo que nos muestra las páginas a medida que las vamos solicitando con el navegador.

Los servidores web son software que actúa como alojador de información. Se trata de un programa que está en continua ejecución esperando peticiones hechas por un cliente o un usuario de Internet, para luego procesarlas y enviar la respuesta adecuada, que consiste en la página solicitada o la información relacionada con la búsqueda hecha por el visitante.

CLIENTE: EL REMITENTE

El cliente emite una petición, espera y recibe la respuesta del servidor, que corresponde a la información solicitada. Tiene la capacidad de conectarse a varios servidores a la vez; esto quiere decir que los usuarios de Internet pueden conectarse a múltiples servidores web al mismo tiempo, los cuales estarán disponibles a la espera de dichas solicitudes.

SERVIDOR: EL RECEPTOR

Los servidores esperan las peticiones de los clientes y, al recibirlas, las procesan y envían las respuestas correspondientes. Un servidor web puede brindar servicios a incontables clientes, es decir, es capaz de mostrar las páginas a muchísimos visitantes del sitio.

EL MODELO CLIENTE/SERVIDOR

Cuando realizamos una búsqueda en Internet, estamos haciendo una **petición HTTP**, y el servidor nos enviará las respuestas a través del código HTML.

El lenguaje HTML es recibido por el cliente, que se ocupa de decodificar ese código para que pueda ser visto en la pantalla por el usuario, dentro del navegador.

Los servidores web se encuentran instalados en computadoras y hospedan la información de los sitios web por los que navegaremos. Cuando el usuario visita una página, recibe información desde el servidor web y, así, ocurre una interacción entre el cliente y el servidor. Ambos requieren aplicaciones específicas para ejecutar sus acciones.

Dichas aplicaciones interpretarán los lenguajes necesarios que se ejecutarán en el momento en que el usuario y el cliente actúen.

Las aplicaciones del lado del cliente interpretan los códigos llamados **scripts**. En general, los navegadores ejecutan estos códigos en los lenguajes Java, como JavaScript, y es posible instalar plugins si son necesarios para decodificar los scripts. En el caso de las aplicaciones del lado del servidor, es este el que procesa el código y luego envía la información.

El servidor responde a las peticiones con respuestas en código HTML, que luego serán decodificadas por los navegadores de los clientes.

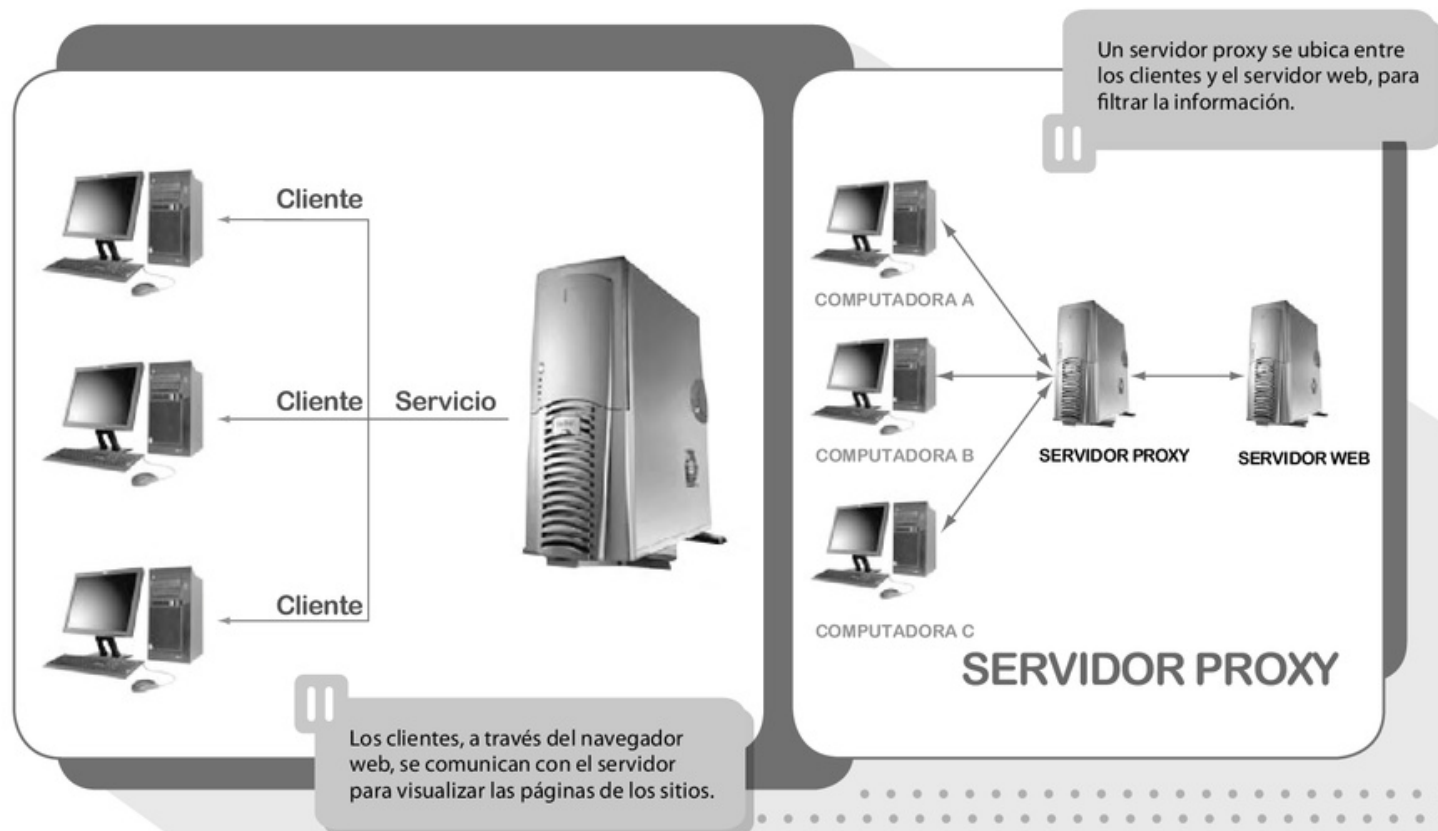
TIPOS DE SERVIDORES

Existen diferentes tipos de servidores, cada uno con distintas capacidades y utilidades, que detallamos a continuación:

LOS SERVIDORES WEB SE ENCUENTRAN INSTALADOS EN COMPUTADORAS Y HOSPEDAN LA INFORMACIÓN DE LOS SITIOS QUE NAVEGAMOS.



El **cliente** es el equipo que solicita la información. El **servidor** es el encargado de dar respuesta a esa petición.



- **Servidores de plataformas:** funcionan con el hardware y el software que se encuentra en el sistema; es decir, son los elementos que hacen funcionar al servidor. En general, se trata de sistemas operativos especiales para servidor.
- **Servidores de aplicaciones:** funcionan a través de software que enlaza dos aplicaciones. Generalmente, actúan entre las bases de datos y los usuarios.
- **Servidores de audio/video:** trabajan con aplicaciones multimedia en la Web, admitiendo la circulación de videos y de audio continuamente.
- **Servidores de chat:** admiten el intercambio de información en tiempo real entre un gran número de usuarios a la vez.
- **Servidores de fax:** actúan en el intercambio de información desde un dispositivo de fax a otro, con la posibilidad de enviar documentos impresos.
- **Servidores FTP:** *File Transfer Protocol* (protocolo de transferencia de archivos) es un tipo de servidor que actúa como soporte de archivos, y da la posibilidad de mover uno o más de ellos a la vez. Los servidores que mencionamos son convenientes para el traslado constante de documentos en Internet.
- **Servidores Groupware:** es un tipo de software que permite la interacción de usuarios en un contexto formal, en el que pueden actuar grupos de trabajo a través de Internet o en una intranet.

- **Servidores IRC:** es un tipo de servidor que funciona para que los usuarios puedan conectarse entre sí a través de servidores y redes diferentes, llamadas IRC (*Internet Relay Chat*).
- **Servidores de listas:** permiten optimizar el orden de las listas de correo electrónico para mejorar el flujo de información en el uso de este elemento en foros, publicidad, anuncios, noticias, etcétera.
- **Servidores de correo:** funcionan con redes corporativas (LAN y WAN) e Internet. Actúan en el transporte y el almacenamiento de correo electrónico, ofreciendo un flujo de información con hospedaje privado.
- **Servidores de noticias:** ejercen la función de entregar y asignar todo tipo de información noticiosa a una gran cantidad de grupos de noticias públicos, disponibles en Internet.

- **Servidores proxy:** son intermediarios entre el cliente y un servidor web. Cumplen la función de servidores, pero actúan, básicamente, en el filtrado de información, la mejora del funcionamiento y compartiendo conexiones.
- **Servidores Telnet:** permiten a un usuario ejecutar tareas en un equipo desde otro, trabajando en forma remota. De esta manera, es como si se estuviese trabajando desde el equipo al que se está conectando.
- **Servidores Web:** sirven contenidos a través de la Red, mostrando a los usuarios de Internet los sitios que tienen alojados. ◀



En general, los servidores son equipos especialmente diseñados para soportar actividad intensa y continua.

CLIENTE SERVIDOR

¿De dónde vienen los textos e imágenes cuando navegamos? De la World Wide Web. Y todo esto se basa en la interacción entre un cliente y un servidor.

Si el documento es HTML puro, se entrega al cliente.



SERVIDOR

Servidor DNS local.
Proveedor de Servicios
de Internet (ISP).



04 PEDIDO AL SERVIDOR

Se define como el proceso que provee el servicio requerido por el cliente. Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. La URL de un documento corresponde a su dirección en la Red: cada archivo o computadora presente en la red puede rastrearse a través de su URL.

03 CONSULTA ICANN

Si el ISP no dispone de la IP de la URL consultada, realiza una consulta al ICANN, que devuelve la consulta al IP correspondiente.

02 CONSULTA

Entra en contacto con una red de servidores DNS. Se realiza una consulta al ISP sobre la URL ingresada. Al igual que las guías telefónicas, se muestra una lista de millones de páginas.

Si el ISP dispone de la IP de la URL consultada, se la envía al cliente.

01 INGRESO

El usuario ingresa la URL del site en el navegador. Por ejemplo: www.google.com.ar.



HTML

El protocolo que se utiliza en las comunicaciones entre un cliente y un servidor es el famoso HTTP (*Hyper Text Transfer Protocol*) y los documentos que se suban a la Red deben tener un formato especial llamado HTML (*Hyper Text Markup Language*)

Método en el que viajan los datos

0001	0011
0101	0110
0001	0011
0101	0110
0001	0011
0101	0110

0001	0011
0101	0110
0001	0011
0101	0110
0001	0011
0101	0110

0001	0011
0101	0110
0001	0011
0101	0110
0001	0011
0101	0110

< ! ?
PHP

Se encuentra la ubicación de los archivos dentro del servidor.

05 CONSULTA BASE DE DATOS

Si el documento contiene bloques de programación en lenguajes tales como PHP o si se realizan consultas a base de datos o XML, el servidor primero resuelve estos bloques.

06 ENTREGA DE HTML

El servidor entrega una página HTML, para que el navegador web la recepcione.

07 NAVEGADORES

El navegador recibe los datos enviados por el servidor, los interpreta y los decodifica para que los usuarios los puedan visualizar en pantalla. Todo este proceso es independiente del navegador que utilizemos, pero es importante que los sitios estén bien programados para no tener complicaciones en la interpretación de las páginas.

CLIENTE

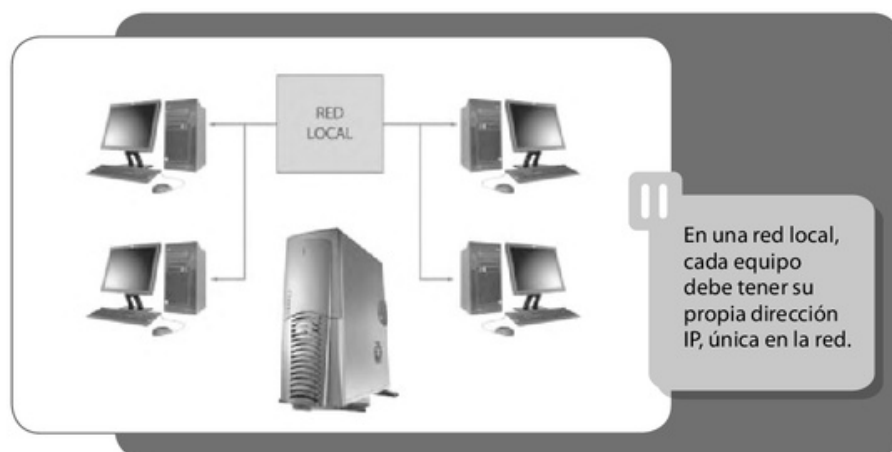


INTERNET

PROTOCOLOS

Seguramente hemos oído hablar sobre los diferentes protocolos que usan las computadoras para comunicarse entre sí por Internet, pero ¿sabemos realmente qué son y para qué sirven? Hagamos una breve introducción.

El término **protocolo** engloba un conjunto de conductas, reglas y normas establecidas para que las computadoras se entiendan entre sí. La familia de protocolos de comunicación en Internet se conoce como **TCP/IP** y tiene dos representantes fundamentales. El protocolo **TCP** (*Transmission Control Protocol*, o protocolo de control de transmisión) garantiza que los datos emitidos por una computadora sean entregados en su destino sin errores y con el mismo orden en el que fueron producidos. El **IP** (*Internet Protocol*, o protocolo de Internet) utiliza paquetes de datos para establecer la comunicación entre dos computadoras de una red. Por sí solo, IP no puede garantizar la integridad ni la recepción de un paquete, por eso se lo emplea en conjunto con TCP. En Internet existen diferentes tipos de protocolos, y su uso es específico para cada tarea que hagamos en ella. Los más conocidos son **HTTP** (*HyperText Transfer Protocol*), **FTP** (*File Transfer Protocol*),



SMTP (Simple Mail Transfer Protocol) y POP (Post Office Protocol).

Para saber el origen y el destino de los paquetes de datos, el protocolo IP utiliza las denominadas **direcciones IP**, un número único que identifica inequívocamente a cada computadora conectada a una red. Ese número está compuesto por cuatro bloques de cifras que van de 0 a 255,

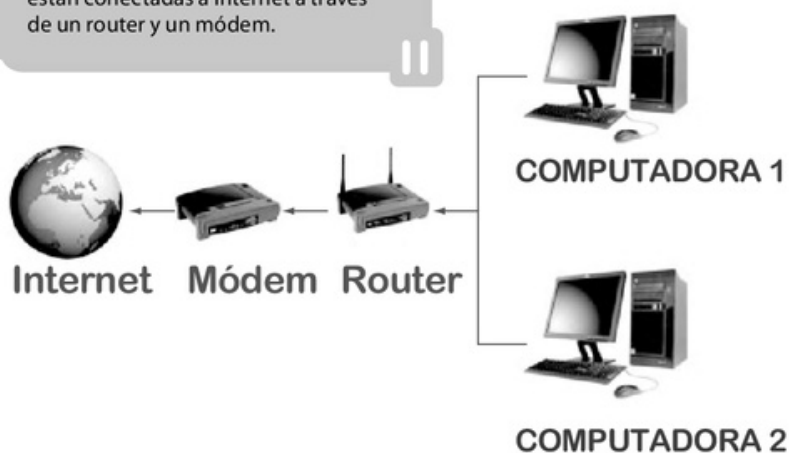
separadas por un punto (por ejemplo, 64.233.163.104). Podríamos decir que si Internet fuese una autopista, las direcciones IP serían las patentes de los autos que circulan por ella. Es posible distinguir entre diferentes tipos de direcciones IP:

► **IP local:** es la que identifica a nuestra computadora dentro de nuestra red hogareña o de trabajo (por ejemplo, 192.168.1.46).

► **IP pública:** es la que identifica a nuestra computadora por fuera de nuestra red hogareña o de trabajo. Si nuestro equipo se encuentra conectado directamente a Internet (sin usar un router de por medio), la IP es la que nos identificará como dispositivo conectado a la Web.

► **IP dinámica y estática:** los dos tipos de direcciones IP que vimos anteriormente pueden ser dinámicas o estáticas. Esto quiere decir que nuestra IP puede variar cada vez que nos conectemos a la red, o bien quedar establecida y ser siempre la misma. Si deseamos montar un pequeño servidor de pruebas en nuestro lugar de trabajo, seguramente nos convendrá tener una IP fija, para así poder apuntar un dominio a ella. ◀

Las dos computadoras de esta red están conectadas a Internet a través de un router y un módem.



DNS: SISTEMA DE NOMBRES DE DOMINIO

Gracias a la existencia de los DNS, podemos escribir una dirección web fácil de recordar y no, una serie de números. A continuación, conoceremos cómo funciona este sistema.

El sistema de nombres de dominio es una gran base de datos cuya principal función es traducir (o **resolver**, en la jerga) un nombre a una serie de números que son utilizados por los equipos conectados. El uso más común que se le da a este sistema es traducir un **dominio** (por ejemplo, **www.google.com**) a una dirección IP, como 64.233.163.104. Lógicamente, es mucho más fácil recordar un nombre que memorizar ese conjunto de números separados por puntos.

¿CÓMO ESTÁ FORMADO UN DOMINIO?

Los nombres de dominio están formados por dos o más partes (formalmente denominadas **etiquetas**), separadas entre sí por un punto.

La **jerarquía** de los dominios se establece de derecha a izquierda, comenzando por la etiqueta del extremo derecho,

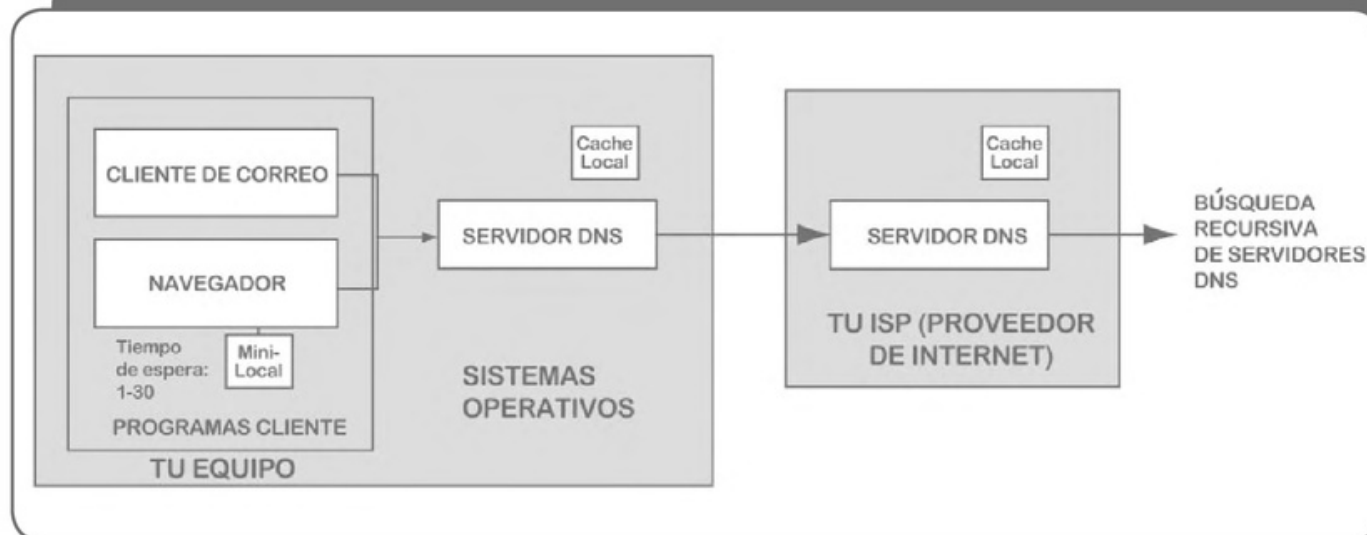
llamada **Top level domain** (dominio de nivel superior). Esto corresponde al **.com** en **www.google.com**, o al **.org** en **www.wikipedia.org**. Cada etiqueta hacia la izquierda se conoce como **subdominio**. Cada uno de ellos debe tener al menos un carácter y un máximo de 63, solo alfanuméricos, a excepción del símbolo - (guión medio). Por ejemplo **"es"** es un subdominio de **wikipedia.org** en **es.wikipedia.org**.

LOS DNS TRADUCEN DIRECCIONES URL A DIRECCIONES IP PARA QUE SEA MÁS FÁCIL RECORDAR LA UBICACIÓN DE LOS SITIOS.

En total, un nombre de dominio completo puede tener, como máximo, 255 caracteres.

FUNCIONAMIENTO

Ahora bien, ¿qué sucede cuando escribimos una dirección web en el navegador de Internet? En primera instancia, el sistema operativo busca de forma local, en la caché de la máquina, si tiene almacenado ese nombre de dominio. Si no lo tiene, inicia la búsqueda en los DNS a los que el equipo esté conectado. Por lo general, el más cercano es el de nuestro proveedor de Internet. Si el DNS de nuestro proveedor puede resolver el dominio, se encargará de enviar hacia nuestra computadora el contenido web que encontrará en la IP de destino indicada por el DNS. Por el contrario, si no puede resolverlo, iniciará una búsqueda recursiva hasta que lo logre. ◀



URL Y TIPOS

DE DOMINIO

A continuación, aprenderemos qué es una URL y qué tipos de dominios podemos encontrar a la hora de registrar un sitio web propietario o para un tercero en Internet.

La sigla **URL** proviene de *Uniform Resource Locator*, es decir, indicador de la localización de un objeto en la Red Global Mundial (WWW o *World Wide Web*). Esta es la manera de organizar y encontrar la información en la Web. Se denomina **uniforme** porque en todo el mundo se utiliza la misma forma para ubicar los archivos o documentos en Internet. Para entender este concepto, debemos comprender que todos los archivos en Internet tiene una dirección específica y única, y lo que hace una URL es especificar aquella en la que ese documento está ubicado.

¿CÓMO LO LEEMOS?

Como vimos al referirnos a DNS, una dirección real es un conjunto de cuatro números separados por puntos. Como estos números son difíciles de recordar para los seres humanos, las direcciones se representan en forma alfanumérica, de modo que sean más descriptivas y fáciles de recordar; luego, son traducidas por el sistema de nombres de dominio de Internet. Como ya sabemos, una URL está compuesta como describimos a continuación: **Protocolo://dirección del sitio/ruta/archivo**.

EXISTEN DISTINTOS TIPOS DE DOMINIOS SEGÚN LA TEMÁTICA, LA UBICACIÓN Y LA FINALIDAD DE UN SITIO.

TIPOS DE DOMINIOS

Un dominio identifica a un sitio y es único en Internet, ya que, como veremos, los dominios son otorgados por entidades especiales que administran su delegación. Existen distintos tipos de dominios según la temática, la ubicación y la finalidad que queramos darle a nuestra página. Entre ellos, encontramos los **genéricos**, que podemos ver detallados en la tabla; los **estatales** (relativos a cada país); y otros que se han incorporado recientemente a Internet, como **.biz**, **.tv** y **.mobi**. ◀



DOMINIOS GENÉRICOS

Los dominios genéricos pueden ser registrados por personas físicas o jurídicas. Cada uno de ellos es ideal para un tipo de sitio o territorio.

TIPO DE DOMINIO

CARACTERÍSTICAS

▶ .com

▶ Los dominios comerciales son los más adecuados para las empresas u organizaciones con fines de lucro.

▶ .com.ar

▶ Este es un ejemplo de dominio territorial, que se utiliza para sitios relacionados con la República Argentina.

▶ .org

▶ Este tipo de dominio es usado por organizaciones sin fines de lucro.

▶ .net

▶ Es utilizado por empresas relacionadas con Internet y tecnología; la extensión proviene de network.

▶ .edu

▶ Este es el dominio empleado por organismos educativos.

▶ .gov

▶ Este dominio es utilizado por los gobiernos para sus sitios oficiales.

▶ .mil

▶ Dominio de uso exclusivo para todo tipo de organizaciones militares.



EL SISTEMA WEB Y SU FUNCIONAMIENTO



Es ideal saber qué sucede cuando visitamos un sitio web. Analizaremos cómo funciona Internet y de qué modo viajan los archivos hasta que son mostrados en la pantalla.



ICANN

ICANN es la organización (sin fines de lucro) responsable de crear y administrar, entre otras cosas, una política de asignación de nombres de Internet, de modo tal que los usuarios puedan encontrar todas las direcciones válidas de las páginas que buscan en la Web. También se ocupa de delegar los nombres de dominio, como .com y .net. Todo este trabajo tiene como fin lograr una estabilidad operacional de Internet para un buen funcionamiento del sistema.

Para conectarse a Internet es necesario contar con una computadora, un módem y una conexión de teléfono o red de cable. Actualmente, también es posible lograrlo por medio de una red de telefonía móvil. Básicamente, el funcionamiento de la Web se basa en la interacción entre un cliente y un servidor.

Para empezar, lo primero que hacemos es ingresar la dirección de una página web en el navegador, por ejemplo, **www.google.com.ar**. La convención www de a poco se está dejando de usar, pero todavía está vigente.

Al ingresar la URL, el navegador entra en contacto con una red de **servidores de DNS** que, como las guías telefónicas, albergan una lista de millones de páginas web. Una vez que encuentra el sitio solicitado por el usuario, localiza al servidor que contiene los datos correspondientes. Si el DNS del proveedor de Internet del usuario dispone de la IP para la URL consultada, se la envía al cliente con los datos de la página. De lo contrario, realiza una consulta al ICANN, organización responsable de asignar espacio de direcciones numéricas de protocolo de Internet (IP), y esta devuelve la IP correspondiente.

Con la dirección IP obtenida, el navegador se conecta al servidor y realiza una petición, solicitando la página. En el servidor hay archivos en diferentes formatos, como HTML, CSS, JS, PHP y ASP. Si el archivo que encuentra es un HTML, lo envía directamente al usuario, pero si el documento localizado contiene bloques de información o archivos con otro formato, el servidor

primero resuelve estos bloques y los reemplaza por sus resultados en HTML, según sea necesario. El último paso de esta ruta ocurre cuando el navegador recibe los paquetes de

información, que interpreta y ensambla para que puedan ser visualizados en la computadora del usuario. Como resultado de este proceso, se muestra la página en la pantalla. ◀



Para visualizar el sitio de Google, debemos ingresar la URL. Luego aparece la página en el navegador.

ELEGIR UN NOMBRE DE DOMINIO

Llegó el momento de elegir un nombre para nuestro sitio. Aunque esta no es una tarea realmente difícil, es conveniente tener en cuenta algunos consejos, como los que vemos a continuación, para elegir uno que sea exitoso.

La selección del nombre del dominio es fundamental a la hora de obtener el éxito o el fracaso en el, cada vez más grande, mundo de Internet. Es una tarea delicada y creativa a la que se le debe dedicar tiempo. A continuación encontraremos consejos útiles que nos ayudarán a elegir un nombre de dominio eficaz para nuestro posicionamiento en los buscadores y, por qué no, para mejorar el de nuestra marca.

- Es conveniente **registrar** el nombre de dominio antes de comenzar el desarrollo o el diseño del sitio. Si el dominio que queremos utilizar ya está tomado, debemos pensar alternativas o ponernos en contacto con el poseedor actual para ver si está dispuesto a venderlo.
- La primera opción debería de ser utilizar el nombre de la **marca**. El nombre del dominio debe tener **relación** con el servicio o producto brindado, o con el nombre de la empresa. El objetivo es facilitarles la tarea a los usuarios para que puedan saber qué contiene el sitio con solo leer su nombre.
- El nombre de dominio debe **distinguirse** lo más posible. Si bien los dominios son únicos e irrepetibles, conviene elegir un nombre que no sea la versión plural o singular de un dominio que ya esté ocupado.
- Debemos buscar un dominio que pueda pronunciarse de forma sencilla. Si resulta muy complejo para ser escrito, pocos usuarios podrán recordarlo. La **simplicidad** ayudará a que localicen más rápido nuestro sitio.
- Si nuestro dominio es similar a otros o difícil de escribir, es conveniente registrar los nombres de dominio relacionados con los **errores** típicos que se cometerían. De esta forma, nos aseguramos de

The screenshot shows the 000DOMAINS.COM website interface. At the top, there's a navigation bar with links like DOMAINS, TRANSFERS, RENEWALS, WEB HOSTING, EMAIL, FAQs, SUPPORT, WHOIS SEARCH, and VIEW CART. Below this, a search bar contains 'disenoweb'. The results show various domain extensions: .COM (Taken), .CO (Taken), .NET (Taken), .ORG (Taken), .ME (available for \$24.99/yr), .TV (available for \$25.99/yr), .US (Taken), .MOBI (Taken), and .ES (Taken). Below the search bar, there's a 'Search Again' section with a dropdown menu set to '.COM' and a 'Bulk Search' button. The main content area is divided into 'Domain Ideas' and 'Extended Results'. The 'Domain Ideas' section lists several domain suggestions with their prices, such as 'disenowebonline.net' for \$15.49/yr and 'disenonet.net' for \$15.49/yr. The 'Extended Results' section shows more suggestions, including 'mydisenoweb.net' for \$15.49/yr and 'thedisenoweb.com' for \$15.49/yr. A 'My Selected Domains' section is also visible on the right side of the results.

Cuando buscamos un dominio en **www.000domains.com**, veremos las extensiones usadas y las disponibles.

UN BUEN DOMINIO DEBE SER FÁCIL DE RECORDAR Y TENER RELACIÓN CON LO QUE OFRECE LA EMPRESA O CON LA MARCA.

que los usuarios lleguen de todos modos a nuestro sitio.

- Lo ideal es que el dominio tenga un **significado**. Por ejemplo, aunque muchos no lo sepan, **Google** es una unidad de medida, y no, una palabra inventada.
- El nombre de dominio debe ser **corto**. Es aconsejable que tenga menos de 7 caracteres, ya que será más fácil de recordar y más rápido de escribir. Si usamos nombres largos, como frases compuestas,

deben resultar fáciles de memorizar, como **www.entremujeres.com**, por ejemplo.

► El nombre no debe ser un eslogan.

Dominios como **lamejorweb.com** son tomados como poco confiables y no muy transparentes.

► No es recomendable utilizar guiones. Los símbolos hacen que el nombre de dominio sea más difícil de recordar, de escribir y de transmitir entre los usuarios. Es mejor acudir a la imaginación hasta encontrar otra opción antes de agregar un guión.

► Tampoco es conveniente utilizar números en el nombre, ya que al decirlo oralmente, es probable que haya que especificar si debe escribirse el número o la palabra. Por ejemplo, **www.puerta15.com**, ¿será **www.puerta15.com** o **www.puertaquince.com**?

► Debemos tener cuidado con los nombres que contengan la **letra x**, ya que algunos servidores proxy y firewalls pueden confundir el sitio con un portal con contenido exclusivo para adultos.

► Es aconsejable incluir los **keywords** o palabras clave dentro del nombre de dominio. Por ejemplo, si nos dedicamos a la industria cosmética, podemos pensar los nombres creando combinaciones o agregando otras palabras al término "cosmética".

► Conocer a nuestro público es fundamental. Debemos saber a quién nos dirigimos y qué servicio ofrecemos, para tenerlo en cuenta a la hora de pensar el dominio.

► No debemos **cambiar** de dominio con frecuencia, ya que los buscadores asocian nuestra página con un ranking, y es muy probable que, al cambiar de nombre, estos datos se pierdan.

► Los dominios tienen una **fecha de expiración** que debemos tener presente para hacer la renovación a tiempo. Existen empresas especializadas en la compra y venta de nombres de dominio que están próximos a vencer. Es mejor ser precavidos y hacer la reserva por períodos largos.

► Aunque puede ser difícil encontrar el dominio .com que deseamos, esta debería de ser la primera elección. Los motores de búsqueda los prefieren, y los usuarios, por lo general, están más acostumbrados al uso de este tipo de páginas. Si es posible, conviene registrar todas las **variantes** adecuadas, como .com.ar, .net, .org, .biz, .net, etcétera. Aunque esto es costoso, evita futuros

+ DOMINIOS REGIONALES

En muchos casos, se opta por elegir solo el dominio del país propio (por ejemplo, **.com.ar**) para localizar más las búsquedas. Sin embargo, este es un error, ya que Google y otros buscadores identificarán que ese dominio es argentino y lo incluirán dentro de las búsquedas de sus resultados de páginas de dicho país. En cambio, si también compramos el dominio **.com**, existen factores que le indican al buscador que el sitio es una página de la Argentina sin necesidad de restringir la búsqueda solo a esa nación. Como vemos, lo más aconsejable es adquirir los dos dominios.

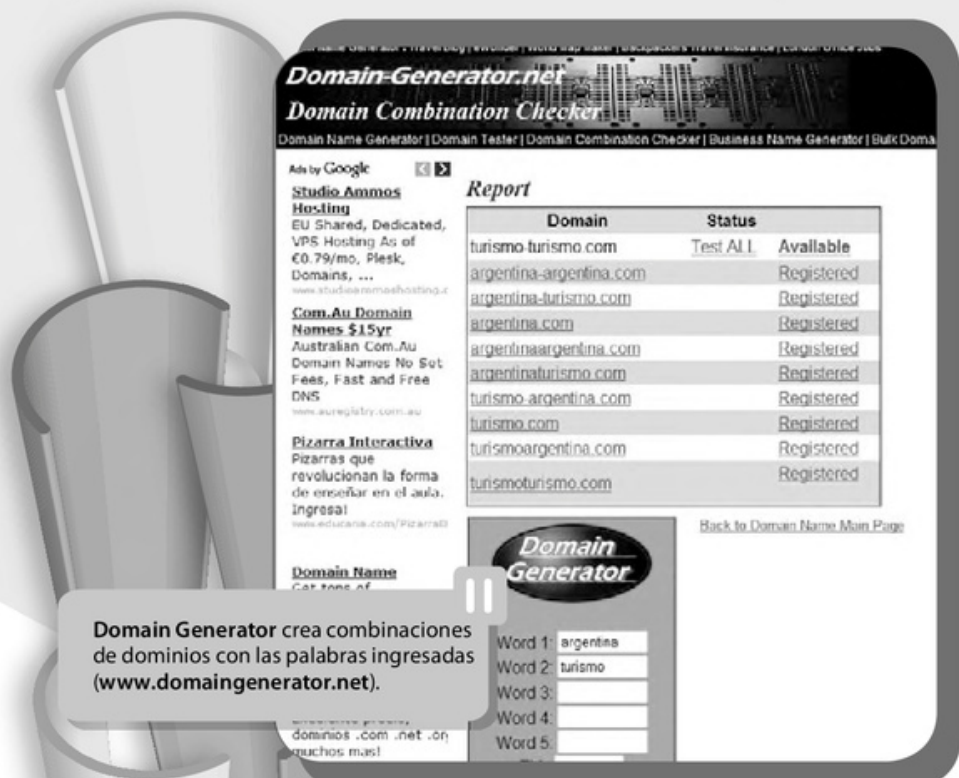
conflictos con otros sitios.

► Además del dominio principal, podemos crear **subdominios**. Por ejemplo, aparte de **www**, podemos configurar en forma gratuita todos los demás que necesitemos, como **http://ar.weather.yahoo.com**.

► Pedir la opinión de conocidos es una buena opción para contar con más posibilidades a la hora de elegir. De hecho, para evaluar si un dominio es fácil de recordar, podemos repetirlo el nombre un par de veces a un conocido y, luego, preguntárselo para ver si lo recuerda.



► Existen herramientas especializadas para el análisis de dominios. Por ejemplo, **Domain Typo Generator** (**www.domaintools.com**) muestra los errores más habituales en la escritura de un dominio. **Domain Name Generator** (**www.namestation.com**) sugiere nombres, e **Internet Archive** (**www.archive.org**) permite saber qué había anteriormente si se trata de un dominio que ya ha expirado. ► Siempre debemos registrar el dominio a nuestro nombre para evitar complicaciones legales. Algunas empresas ofrecen los dominios a un menor precio, pero a cambio de registrarlo a nombre de otro. Con estos consejos en mente, deberemos activar la creatividad y buscar el mejor nombre posible. No todos los buenos dominios están registrados. ◀



¿CÓMO REGISTRAR UN DOMINIO?

Con el dominio elegido, debemos registrarlo para poder utilizarlo a la hora de publicar el sitio. Conoceremos los aspectos principales a tener en cuenta para hacer este trámite.

La registración de un dominio establece que una persona obtiene el control y la administración sobre ese nombre. Este proceso se realiza entre la persona que quiere registrar el dominio y un registrador o una entidad registrante. El organismo que se encarga de regular el registro de los dominios es NIC, ente que tiene delegaciones en diferentes países, como **www.nic.ar** para la República Argentina o **www.nic.cl** para Chile. El proceso de registración es muy similar en todos, ya que está regido por una serie de pasos semejantes.

REGISTRO DE UN DOMINIO

Varias empresas se dedican a la registración de dominios, y entre ellas es recomendable elegir una de nuestro país. Debemos analizar cuál nos ofrece el servicio y el precio, ya que muchas

LA REGISTRACIÓN DE UN DOMINIO ESTABLECE QUE EL REGISTRANTE OBTIENE EL CONTROL Y LA ADMINISTRACIÓN SOBRE ESE NOMBRE.

veces, estas firmas también brindan servicio de **hosting**.

En primer lugar, para registrar un dominio debemos comprobar si el nombre elegido está disponible. Es conveniente investigar que no sea el nombre o parte del nombre de una marca registrada, dado que así evitaremos una posible demanda por la propiedad del dominio. Luego, necesitamos completar un formulario online con los siguientes datos:

- ▶ **Propietario del dominio:** titular (persona o entidad) que aparece como propietario del dominio durante el período que dure el registro.
- ▶ **Contacto administrativo:** persona o entidad designada por el propietario para administrar los datos del dominio.
- ▶ **Contacto técnico:** persona o entidad encargada de mantener actualizados los números DNS para que el dominio funcione correctamente.
- ▶ **Contacto de facturación:** persona o entidad encargada de pagar las renovaciones del dominio.
- ▶ **DNS (Domain Name Servers):** direcciones de los DNS principales que contienen información sobre la IP del dominio.



Cada país tiene su propio organismo para llevar a cabo el registro de dominios regionales. En este caso, vemos **www.nic.mx**, de México; **www.nic.br**, para Brasil; y **www.cointernet.com.co**, de Colombia.

REGISTRAR UN DOMINIO PASO A PASO

Tener un dominio en Internet es importante porque otorga identidad y ubicación a un proyecto o negocio. Veamos cómo registrar el que hemos elegido.



01 | DISPONIBILIDAD



Consulta de Dominios

El dominio userswebdesign.com.ar se encuentra disponible

Para solicitar el registro de este dominio, haga click **aquí**.

Última actualización 2010-09-10 10:08:08

INICIAR REGISTRO 02

Como mencionamos, el procedimiento para registrar un dominio **.com** es bastante similar en cualquiera de los organismos de delegación de dominios. En este caso, realizaremos el registro de un dominio **.com.ar**, trámite muy sencillo y gratuito que podemos llevar a cabo en **NIC Argentina** (www.nic.ar). ▶

PASO | 01 Accedemos a la dirección **www.nic.ar**. Allí, dentro del recuadro **[Consulta de dominios]**, ingresamos el nombre de dominio que queremos registrar y, luego, en el menú desplegable, elegimos la extensión correspondiente, en este caso, **.com.ar**. Presionamos el botón **[Buscar]** para ver si el dominio está disponible. Si ya está ocupado, aparecerán los datos del registrante y deberemos buscar otro nombre.

PASO | 02 Si el nombre que elegimos está disponible, se abre una ventana donde se nos ofrece la posibilidad de comenzar el trámite de registro. Allí, hacemos clic en la opción indicada para iniciar el proceso, en este caso, llamada **[aquí]**.



REGISTRAR UN DOMINIO

PASO A PASO

Proceso guiado para el Registro de un nombre de Dominio

Paso

1 2 3 4 5 6 7

Ingrese el nombre del dominio (máximo 19 caracteres) y luego seleccione el tipo de subdominio

userswebdesign .com.ar

Indique el e-mail del Solicitante:

velezp@gmail.com

continuar

Solicitante puede ser el propio Registrante o una persona física o jurídica ante del Registrante. Completar esta solicitud electrónica para registrar un nombre de dominio, el tante manifiesta conocer y aceptar las reglas, procedimientos e instrucciones.

03 | ELEGIR DOMINIO

Consulta de dominios
Registrar dominio
Renovar dominio
Transferir dominio
Trámites vía web
Dominios .edu.ar
Guía del solicitante
Normativa vigente
Listado de Dominios IDN
Preguntas frecuentes
Glosario de términos
Guías interactivas
Vías de contacto

NIC Argentina
Esmeralda 1212, C1007ABR
Buenos Aires - Argentina
Tel. +54 (11) 4815-7631
Fax. +54 (11) 4815-7630
e-mail: info@nic.ar

Ministerio de Relaciones Exteriores,
Comercio Internacional y Culto

Registro del nombre de dominio:
userswebdesign.com.ar

Paso

1 2 3 4 5 6 7

Alta de una nueva Entidad Registrante

Nombre Paula Velez

Calle Rivadavia

Domicilio

Nro. 321 Piso Depto.

Ciudad/Localidad Capital Federal CP 1414

Provincia Ciudad de Buenos Aires

Teléfono 4382-4512 Fax 4382-4512

E-Mail vlezp@gmail.com

Actividad Diseño de Páginas Web

Seleccione el tipo de entidad que corresponda

☒ Registrante, persona física y residente en Argentina

☐ Registrante, persona jurídica y residente en Argentina

☐ Registrante y residente en el extranjero

continuar

DAMOS DE ALTA | 04

NIC Argentina

Registro de dominio

Registrar dominio

Renovar dominio

Transferir dominio

Trámites vía web

Dominios .edu.ar

Guía del solicitante

Normativa vigente

Listado de Dominios IDN

Preguntas frecuentes

Glosario de términos

Guías interactivas

Vías de contacto

NIC Argentina

Esmeralda 1212, C1007ABR

Buenos Aires - Argentina

Tel. +54 (11) 4815-7631

Fax. +54 (11) 4815-7630

e-mail: info@nic.ar

Ministerio de Relaciones Exteriores,
Comercio Internacional y Culto

Registro de dominio

Registrar dominio

Renovar dominio

Transferir dominio

Trámites vía web

Dominios .edu.ar

Guía del solicitante

Normativa vigente

Listado de Dominios IDN

Preguntas frecuentes

Glosario de términos

Guías interactivas

Vías de contacto

NIC Argentina

Esmeralda 1212, C1007ABR

Buenos Aires - Argentina

Tel. +54 (11) 4815-7631

Fax. +54 (11) 4815-7630

e-mail: info@nic.ar

Ministerio de Relaciones Exteriores,
Comercio Internacional y Culto

Registro de dominio

Registrar dominio

Renovar dominio

Transferir dominio

Trámites vía web

Dominios .edu.ar

Guía del solicitante

Normativa vigente

Listado de Dominios IDN

Preguntas frecuentes

Glosario de términos

Guías interactivas

Vías de contacto

NIC Argentina

Esmeralda 1212, C1007ABR

Buenos Aires - Argentina

Tel. +54 (11) 4815-7631

Fax. +54 (11) 4815-7630

e-mail: info@nic.ar

Ministerio de Relaciones Exteriores,
Comercio Internacional y Culto

05 | PERSONA RESPONSABLE

Nombre y Apellido Lara Velez

Calle 9 de Julio

Domicilio

Nro. 555 Piso Depto.

Ciudad/Localidad Capital Federal CP 1425

Provincia Ciudad de Buenos Aires

de Residencia Argentina

Teléfono 4381-4523 Fax 4381-4523

Número telefónico 9 a 17

e-Mail larav@gmail.com

continuar

PASO | 03

Volvemos a ingresar el nombre de dominio que queremos registrar, elegimos la extensión y escribimos la dirección de correo electrónico que usaremos para el intercambio de mensajes con NIC. Es importante que esta dirección sea verdadera, ya que deberemos responder los e-mails para que el proceso de registro se lleve a cabo. Luego de completar los datos, presionamos **[continuar]**.

PASO | 04

Si no habíamos registrado ningún dominio con anterioridad, presionamos **[nueva entidad]** para darnos de alta como registrantes. Completamos el formulario con nuestros datos verdaderos y sin errores, ya que de eso dependerá que podamos realizar el registro o cambios posteriores. Cuando terminamos de ingresar los datos, hacemos clic en **[continuar]**.

PASO | 05

Ingresamos CUIT y DNI propios en la pantalla siguiente y presionamos **[continuar]**. Luego, elegimos a la persona responsable si esta ya está registrada o la damos de alta tal como hicimos con la entidad registrante.

PASO | 06

Ahora debemos indicar si tenemos o no un proveedor de servicios ISP. Lo importante es saber si contamos o no con las direcciones IP de los servidores DNS. Si no disponemos de estos datos, podemos hacer una **reserva** del dominio para asegurarnos su titularidad. Si tenemos las dos direcciones DNS necesarias, podemos realizar el registro del dominio, presionando **[delegar]**. Completamos las direcciones y URLs de los DNS y volvemos a hacer clic en **[delegar]** para continuar.

Registro del nombre de dominio:
userswebdesign.com.ar

Paso
1 2 3 4 5 6 7

Delegación del Dominio

Servidores de Nombre Primario:
Dirección IP: 24.232.0.18 Nombre: dns2.ovni.com.ar

Servidor de Nombre Secundario:
Dirección IP: 24.232.0.17 Nombre: dns1.ovni.com.ar

Datos Opcionales

Servidor de Nombre Alternativo:
Dirección IP: Nombre:

Servidor de Nombre Alternativo:
Dirección IP: Nombre:

Servidor de Nombre Alternativo:
Dirección IP: Nombre:

delegar

06 | RESERVAR O DELEGAR

Ingrese los caracteres que ve en esta figura.

Verificación de texto: TET55ton

NOTA: debe ingresar los caracteres que se ven en la imagen respetando mayúsculas y minúsculas. Si luego de ingresarlos obtiene "ERROR: El texto Verificador ingresado no es correcto", intente nuevamente.

Si luego de varios intentos sigue obteniendo el mismo error, vuelva a la página anterior y presione Reservar (o Continuar) para obtener una nueva imagen con una cadena diferente de caracteres e intente nuevamente.

Para completar la solicitud debe aceptar los términos de la Normativa vigente para el registro de nombres de dominio Internet en Argentina.

☒ He leído y acepto la Normativa vigente para el registro de nombres de dominio Internet en Argentina.

confirmar

FINALIZAR EL REGISTRO | 08

Registro del nombre de dominio:
userswebdesign.com.ar

Paso
1 2 3 4 5 6 7

Seleccione un nombre de ISP

ISP detectados:

- Cablevision S.A.
- Canal 2 Cablevision S.A.
- Grupo CABLEVISION S.A.

continuar

Si no encuentra el ISP deseado reformule su búsqueda

reformular

07 | PROVEEDOR ISP

Registro y Delegación de userswebdesign.com.ar Recibido: [X]

☆ asigdom@nic.ar para usuario mostrar detalles: 14:49 (7 minutos antes)

.....

IMPORTANTE: NO RESPONDA ESTE E-MAIL SIN LEERLO CUIDADOSAMENTE.

.....

Al responder este e-mail Ud. iniciará un trámite ante NIC-Argentina.

Verifique que los datos correspondan al formulario que Ud. llenó en forma interactiva en el Web de NIC-Argentina en la fecha y hora del envío de este e-mail.

Si Ud. no llenó este formulario en el Web del NIC, o los datos del formulario no corresponden a un trámite que Ud. desea iniciar en esta oportunidad ante el NIC, NO RESPONDA ESTE E-MAIL.

Si Ud. desea iniciar el trámite, responda, con cuidado de no modificar en nada su contenido.

CORREO DE CONFIRMACIÓN | 09

PASO | 07 Escribimos el nombre del proveedor ISP y presionamos **[buscar]**. Luego, seleccionamos la opción correcta entre los resultados que se muestran y pulsamos **[continuar]**. Aparecerá el nombre del contacto técnico, que es una persona relacionada con el ISP elegido. Pulsamos **[buscar]**, elegimos la opción adecuada entre los resultados de la búsqueda (podemos confirmar este dato con el ISP) y presionamos **[continuar]**.

PASO | 08 Se presentará un resumen con los datos ingresados a lo largo de los pasos anteriores. En la parte inferior de la página encontramos un recuadro donde debemos indicar los caracteres de la imagen (esto es para evitar registraciones automatizadas). Luego, marcamos la casilla de aceptación de la normativa vigente y hacemos clic en **[confirmar]**.

PASO | 09 Después de confirmar el registro, recibiremos un mensaje en la cuenta de correo indicada en el formulario de registro. El mail provendrá de **asigdom@nic.ar** y deberemos leerlo atentamente para verificar que todos los datos sean correctos. En caso afirmativo, lo reenviamos **sin modificarlo** a la dirección del remitente. Recibiremos otro correo como respuesta, confirmando que hemos registrado el dominio.

RESUMEN

Una vez que hayamos registrado el dominio, la URL de nuestro sitio será reconocida en Internet al cabo de unos días u horas (el tiempo que tardan en actualizarse los servidores DNS con la dirección IP del dominio). Luego de ese lapso, el sitio será accesible desde cualquier equipo.

SERVICIOS

DE HOSTING

Existen distintas maneras de alojar una página en la Web. Aquí haremos un pequeño listado de las diferentes vías para hospedar un sitio en diversos servidores online.

Webhosting es un servicio proporcionado por empresas (Web Host) que proveen espacio online para cualquier tipo de archivo que pueda ser visto desde un navegador web.

Al querer alojar nuestra página o sitio web, lo primero que debemos tener en cuenta es si deseamos que este servicio sea gratuito o pago. Además del costo, hay otras consideraciones y diferencias que debemos conocer:

- El alojamiento gratuito, al no cobrarnos ningún canon por el servicio que nos brinda, no suele hacerse responsable de que la visualización del sitio sea constante ni de resolver inconvenientes técnicos que pudiesen afectar al sitio. Por el contrario, el hosting pago garantiza el funcionamiento del servicio y soluciona los inconvenientes que surgen, para que nuestro sitio siga online todo el tiempo.
- En general, los hosts gratuitos colocan en las páginas que alojan todo tipo de **publicidad**, realizada mediante banners, pop-ups, etcétera. Nosotros no podemos controlar este tipo de elemento, por lo que el sitio se cargará

de manera más lenta que lo normal.

- El espacio provisto en los servidores para la subida de archivos que acompañan al sitio es mucho menor en los servidores gratuitos que en los pagos. Por otro lado, la mayoría de los servicios gratuitos no poseen un servicio de **backup** de archivos en caso de que estos se borren por accidente.

- En los servicios gratuitos, suelen aparecer **restricciones**. Puede limitarse el ancho de banda, es decir, el tráfico que puede tener el sitio web (por ejemplo, no debe exceder una determinada cantidad de MB por mes) o el uso de imágenes

alojadas en el servidor. También, pueden permitirse archivos de hasta un cierto tamaño en KB, o admitirse algunos tipos de archivos y no otros.

- En los hostings gratuitos, las funcionalidades extra (como servicios de PHP, ASP o CGI) deben pagarse. En cambio, estas funcionalidades son provistas dentro del paquete de servicios de los hosts pagos.

- En la mayoría de los servidores de alojamiento pago, los trámites de registración de dominio se incluyen en la tarifa de hosting o, si se realiza el trámite de un dominio pago, puede incluirse como extra el alojamiento durante un año.

A LA HORA DE ELEGIR UN
SERVICIO DE HOSTING,
DEBEMOS CONOCER LAS
TECNOLOGÍAS Y EL
ESPACIO QUE NECESITAMOS.

LA ELECCIÓN

Subir nuestro sitio a un servidor gratuito o a uno pago dependerá del uso que vayamos a darle.

Si recién comenzamos con el diseño web, o la página es de tipo personal y básica, podemos elegir uno gratuito. Por el contrario, si necesitamos funciones adicionales, que haya publicidades, contar con un buen ancho de banda y soporte técnico especializado, tendremos que recurrir a los servicios pagos.

CONSIDERACIONES
AL CONTRATAR UN HOSTING PAGO

Lo primero que debemos considerar es si contrataremos un **servidor Linux** o **Windows**. Los planes basados en Linux son más accesibles que los de Windows. Ambos son bastante confiables en cuanto a seguridad, tienen buena velocidad de respuesta y vienen con paneles de control fáciles de comprender. Entre las diferencias, podemos mencionar que Linux se utiliza con proyectos basados en PHP, Perl o MySQL; mientras



OTRA VARIANTE

Existe una alternativa a las opciones de hosting pago o gratuito que ya explicamos. Es conocida como **reseller** (revendedor), y es una de las posibilidades que ofrecen las empresas de web hosting pago. Esta variante también depende de un pago mensual, pero es especial porque, además de abonar el costo propio, permite proveer el servicio a terceros y cobrar por ello. De esta manera, es posible obtener una ganancia por el pago realizado o, al menos, no desembolsar dinero propio para pagar el abono. Básicamente, el paquete ofrece bastante espacio en disco, varios dominios, cuentas de correo ilimitadas, extensiones, etcétera. Es una interesante opción que debemos considerar.

Zymic (www.zymic.com) es una empresa de hosting gratuito con buenas prestaciones. Permite subir archivos por FTP y no agrega banners publicitarios.

ToWebs (www.towebs.com) brinda hosting pago con distintas variantes y opciones, y permite elegir si el servidor tendrá sistema **Linux** o **Windows**.

	Personal Web	Personal Plus	Ecommerce Basic	Ecommerce Plus	Corporation Basic	Corporation Plus
	LINUX	WINDOWS	LINUX	WINDOWS	LINUX	WINDOWS
	\$ 30	\$ 30	\$ 30	\$ 30	\$ 50	\$ 70
	\$ 14.95	\$ 24.95	\$ 24	\$ 35	\$ 99	\$ 145
	5T	5T	5T	5T	5T	5T
/año	\$ 30*	\$ 30*	\$ 30*	\$ 30*	\$ 30*	\$ 30*
	1Gb	1Gb	2 Gb	2 Gb	10 Gb	10 Gb
	1	1	ilimitados	10	ilimitados	ilimitados
	5	\$5.-/m	ilimitados	5	ilimitados	ilimitados
	5T	5T	5T	5T	5T	5T

En esta imagen podemos ver la interfaz del panel de control de Solunet (**www.solunet.com.ar**), que nos permitirá configurar las diferentes opciones del sitio web.

Luego, debemos evaluar el **espacio de alojamiento**, que es el lugar donde colocaremos los archivos. Si nuestro sitio es pequeño, nos alcanzará con un hosting básico que nos provea de unos pocos MB. Pero si es grande o contiene muchos archivos, es conveniente ver cuánto pesa su carpeta en nuestra PC y, según ese parámetro, contratar un servicio a medida. También tenemos que conocer la posibi-

pueden servirnos en el desarrollo del sitio, como PHP, ASP, Perl, CGI, extensiones FrontPage y Flash/Shockwave. Algunas permiten el funcionamiento de scripts en nuestra página, otras tienen que ver con el uso de formularios desde la Web, y otras, con el manejo de bases de datos. La **administración** de los archivos del sitio es otro factor importante. Puede ser vía FTP o mediante transmisión manual (con comandos desde un navegador de Internet). Finalmente, evaluaremos si hay **panel de control**, una interfaz que permite administrar los recursos de nuestros sitios web de manera sencilla. ◀

SUBIR NUESTRA WEB A INTERNET

Existen varios caminos que podemos utilizar para poner nuestro sitio y sus archivos en Internet. Conozcamos cuáles son las alternativas más comunes.

Cuando hablamos de **subir** una página web, nos referimos al acto de colocar en el servidor (ya sea propio o de un hosting gratuito o pago) todos los archivos que componen nuestro sitio web. Esto incluye los HTML, CSS, PHP, JPG, GIF, PNG, etcétera, que se encuentren relacionados o vinculados para descargar. A la hora de ordenarlos, lo más adecuado es agruparlos en carpetas para cada tipo de archivo (imágenes, scripts, videos, etcétera), que serán parte de una carpeta o directorio principal. En la actualidad, para subir los archivos a los servidores en la Web, podemos hacer uso de dos técnicas: vía FTP y a través de una interfaz web (explorador de archivos). A continuación, conoceremos las características de cada una.

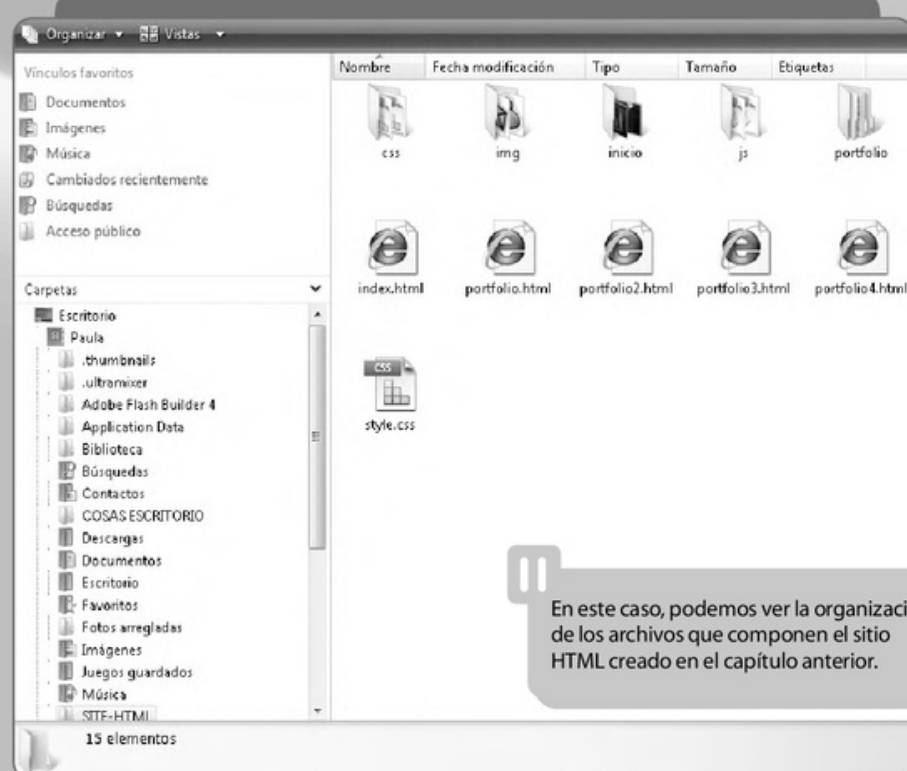
FTP (FILE TRANSFER PROTOCOL)

FTP es la manera tradicional de subir archivos. Es un servicio que ofrecen los hostings pagos y, también, algunos gratuitos, y se utiliza para transferir los archivos desde el equipo del usuario hasta el servidor. Para usar este tipo de transferencia debemos tener un programa **FTP Client** (cliente FTP), que realiza la conexión entre el equipo y el servidor seleccionado. Al tener que elegir un cliente FTP, disponemos de programas pagos y gratuitos. En la primera categoría, la aplicación más conocida es **CuteFTP** (www.cuteftp.com), que debe su fama a su facilidad de uso y a que fue uno de los primeros clientes multiplataforma (Windows y Macintosh). Entre las opciones gratuitas, también llamadas de distribución libre, el de uso más extendido es **FileZilla** (www.filezilla-project.org), que reúne

muchas de las mejores condiciones de un programa de FTP, como la facilidad de uso, la posibilidad de realizar conexiones a través de servidores proxy y firewalls, y la opción de guardar las direcciones más usadas. Podemos encontrarlo en versiones para Windows, Macintosh y Linux. Además de las aplicaciones específicas, muchos de los programas de diseño web incorporan una herramienta por medio de la cual, sin salir del software, podemos subir nuestros archivos al sitio elegido. Por ejemplo, Adobe Dreamweaver brinda esta utilidad luego de configurar el sitio. Básicamente, todos los clientes FTP tienen una interfaz similar, que consiste en

una ventana dividida en dos partes. En una de ellas podemos ver nuestro disco duro, con sus distintas unidades y carpetas; y en la otra, el sistema de archivos del servidor, con sus correspondientes carpetas. Como podremos notar en las próximas páginas, transferir archivos es tan sencillo como arrastrarlos de un lado hacia el otro.

Al comenzar a utilizar estos programas, quizá resulte un tanto complicada la tarea de configurar la cuenta de FTP, ya que debemos ingresar algunos datos que tal vez no conozcamos. Sin embargo, estos los provee la empresa de hosting, que seguramente nos ayudará en el proceso.



En este caso, podemos ver la organización de los archivos que componen el sitio HTML creado en el capítulo anterior.



FTP CLIENT O INTERFAZ WEB

El modo FTP Client es efectivo para transferir un gran volumen de archivos. En cambio, la interfaz web es recomendable para pocos archivos, no para sitios complejos.

	FTP CLIENT	INTERFAZ WEB
USO IDEAL	▶ Grandes volúmenes de archivos.	▶ Pocos archivos por sitio.
PESO DE ARCHIVOS	▶ Archivos de pesos variados.	▶ Hasta un limitado peso en KB.
DIRECTORIOS	▶ Visualización simultánea de los directorios.	▶ Visualización de un directorio por vez.
TRANSFERENCIA	▶ Posibilidad de transferir los archivos de a uno o varios a la vez.	▶ Depende del sistema con el que cuente el proveedor de hosting.

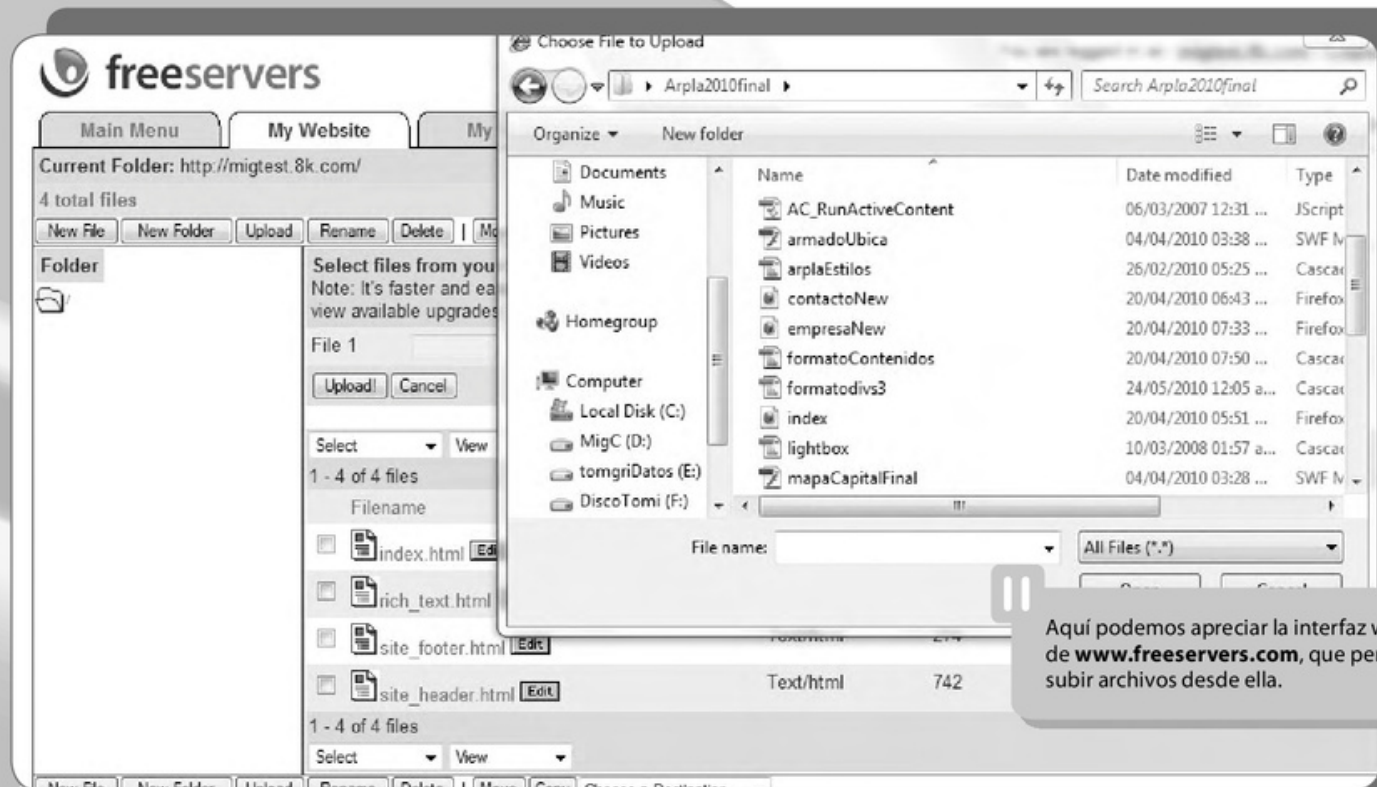


INTERFAZ WEB (EXPLORADOR DE ARCHIVOS)

Este método es el más usado por los proveedores de host gratuito. El procedimiento es muy parecido al de **adjuntar (attach)** un archivo a un e-mail en cuentas de webmail, como Hotmail, Yahoo o Gmail. La interfaz está constituida por un formulario al que accedemos luego de ingresar como usuarios, y nos provee una herramienta con la cual vamos subiendo, de manera individual, cada archivo a nuestro espacio en la Web. En algunos casos, si bien se seleccionan los archivos uno por uno, al momento de subirlos el proceso ocurre grupalmente.

La mecánica de este método es muy sencilla: presionamos un botón denominado **[Examinar...]** (o similar), buscamos el archivo en el equipo en el cuadro de diálogo que se abre, lo seleccionamos al encontrarlo y, para terminar, presionamos el botón que, en general, se llama **[Subir]**. La desventaja de esta variante es que los hosts gratuitos suelen imponer un límite de peso de archivo en KB que no podemos exceder. Si nuestro sitio tiene imágenes grandes o archivos de video o música que superen esos límites, no podremos subirlos. Además, si el sitio posee muchos archivos, el proceso resultará bastante tedioso. ◀

EXISTEN **CLIENTES FTP GRATUITOS Y PAGOS**, Y TAMBIÉN **UTILIDADES INCLUIDAS EN ALGUNOS PROGRAMAS DE DESARROLLO WEB.**



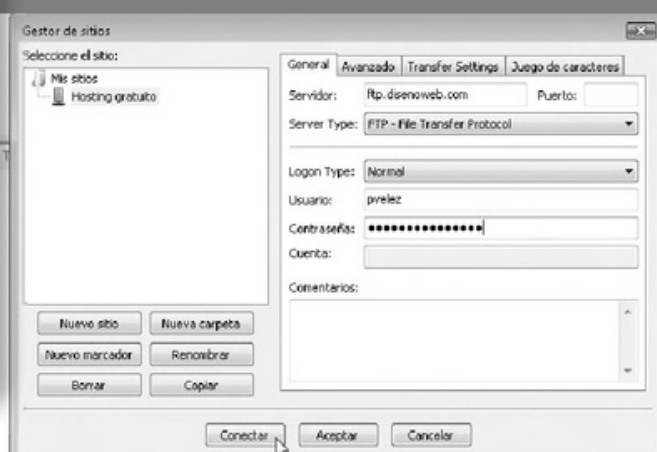
SUBIR UN SITIO AL

PASO A PASO

Llegó el momento de subir los archivos a Internet para que los usuarios puedan verlos.



01 DESCARGA Y DATOS



CONFIGURAR CUENTA

02

Ya mencionamos que FileZilla es un cliente FTP gratuito de gran nivel; por eso, en este tutorial aprenderemos a utilizarlo para subir nuestro sitio al servidor.

Gracias a la practicidad y simpleza de este programa, podemos crear una cuenta, conectarnos y subir los archivos en muy pocos pasos. ◀

PASO | 01

Lo primero que debemos hacer es **descargar e instalar** FileZilla en nuestro equipo. Luego, le solicitamos

los datos de conexión a nuestro proveedor, ya que los necesitaremos para configurar el programa y acceder a nuestro espacio. Estos incluyen nombre del host, usuario, contraseña y nombre de dominio.

PASO | 02

Abrimos el programa, vamos a **[Archivo/ Gestor de sitios...]** y, en la ventana que se abre, hacemos clic en **[Nuevo sitio]**

y escribimos el nombre que le daremos a la conexión, como **[Hosting gratuito]**. En los campos ubicados a la derecha de la ventana, ingresamos los datos suministrados por el proveedor del servicio de hosting. Al finalizar, presionamos **[Conectar]**.

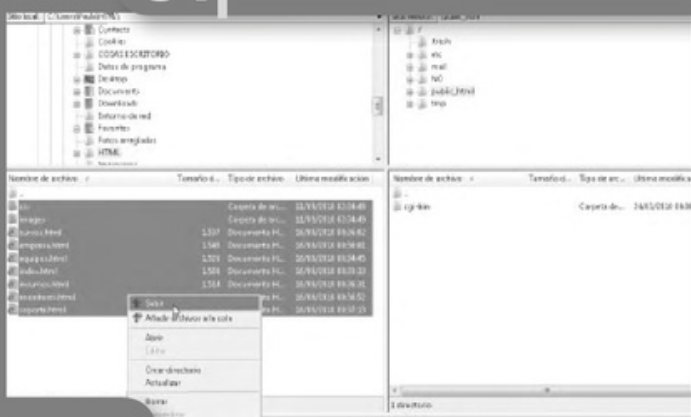
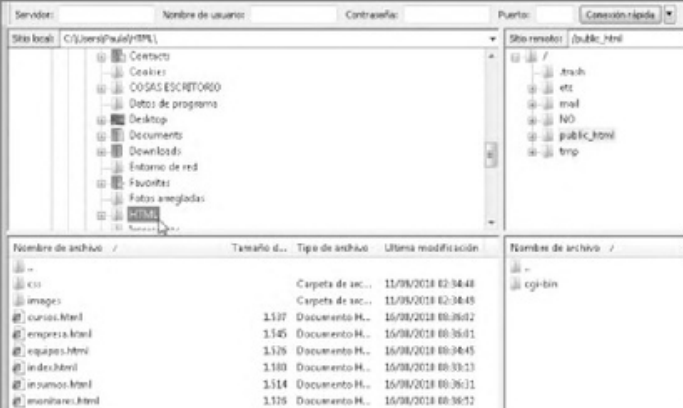
PASO | 03

Al conectarnos, la pantalla aparece dividida en dos partes. La lista del lado

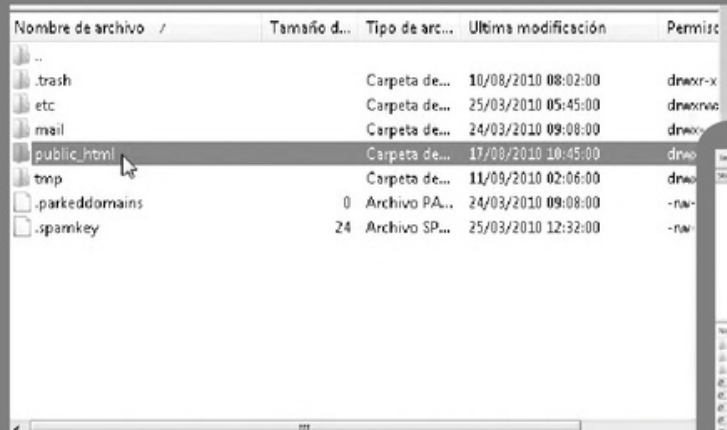


SERVIDOR

04 CARPETA DESTINO



SUBIR ARCHIVOS 05



03 CONEXIÓN

izquierdo corresponde a los archivos de nuestro equipo, y la ubicada del lado derecho, a nuestro espacio online. Si bien puede haber diferencias en los nombres de las carpetas, es común encontrar en el servidor una llamada **[public_html]**, directorio en el que deberemos ubicar todos los archivos que conforman nuestro sitio.

PASO 04 Hacemos doble clic en la carpeta **[public_html]** y, en el área izquierda, recorremos nuestro disco hasta encontrar la ubicación de los archivos que vamos a subir. Recordemos que el archivo de la página principal debe denominarse **index.html**, **index.htm** o **index.php**, entre otras opciones, porque esa será la página que aparecerá por defecto cuando los visitantes accedan al sitio.

PASO 05 Para subir los archivos, primero debemos seleccionarlos en el lado izquierdo de la

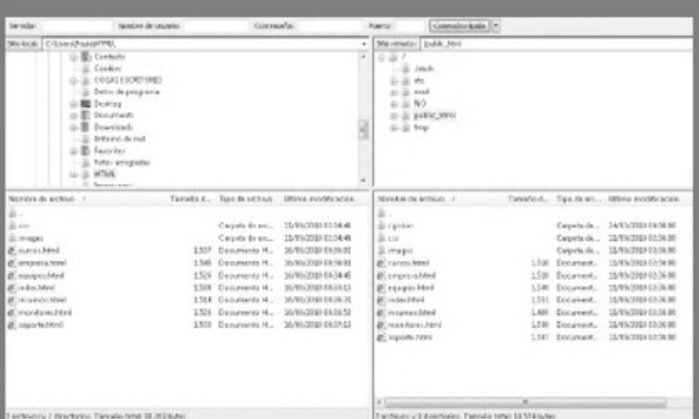
aplicación. A continuación, podemos arrastrarlos con el mouse hasta el lado derecho, o hacer clic con el botón derecho del mouse y elegir **[Subir]**. Si lo deseamos, podemos realizar estas mismas operaciones sobre archivos individuales o hacer doble clic sobre un archivo por vez para subirlo.

PASO 06 Una vez que termina la transferencia, habremos subido los archivos al servidor FTP. Para verificar si todo subió correctamente, bastará con visitar el sitio ingresando su dirección en el navegador.

RESUMEN

Para descargar archivos, los arrastramos, o hacemos clic derecho y elegimos la opción **[Descargar]**. Aunque existen otras aplicaciones similares, FileZilla cuenta con un comparador de archivos, que verifica si todo subió correctamente.

FINALIZAR Y VERIFICAR 06



ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué es un cliente y un servidor?

02 ¿Qué son los protocolos?

03 ¿Qué significa DNS?

04 Caracterice una URL.

05 ¿Cómo funciona el sistema web?

06 ¿Cómo debemos elegir un nombre de dominio?

07 ¿Qué necesitamos para registrar un dominio?

08 ¿Para qué sirve un servidor de hosting?

09 ¿Cómo debemos subir una web a Internet?

10 ¿De qué forma debemos subir un sitio al servidor?

PRÁCTICAS

01 Identifique los tipos de dominio que puede utilizar.

02 Elija un nombre de dominio para su web.

03 Registre un dominio.

04 Seleccione un servidor de hosting.

05 Suba un sitio web al servidor.

Vida Laboral

- ▶ LA SALIDA LABORAL
- ▶ WORKFLOW DE TRABAJO
- ▶ ¿EMPREENDEDOR, EMPLEADO O FREELANCE?
- ▶ FREELANCE: ¿CÓMO INICIARSE?
- ▶ ADMINISTRAR EL TIEMPO
- ▶ CONSEGUIR CLIENTES ONLINE
- ▶ EMPREENDEDOR. ¿YO?
- ▶ PLANIFICAR UN PORTFOLIO EFECTIVO
- ▶ OPORTUNIDAD EN LAS IDEAS
- ▶ DETERMINAR EL PRECIO DEL TRABAJO
- ▶ MODELO DE NEGOCIO
- ▶ FUNCIONAMIENTO DE UNA EMPRESA
- ▶ PLANIFICAR PRESENTACIONES EFECTIVAS
- ▶ LAS ACTITUDES NOCIVAS DEL DISEÑADOR
- ▶ GESTIÓN FINANCIERA
- ▶ NEGOCIACIONES EFECTIVAS
- ▶ CLIENTES MANIPULADORES
- ▶ GENERAR RELACIONES CONTRACTUALES
- ▶ PRIMER AÑO DEL EMPRENDIMIENTO



LA SALIDA LABORAL

Antes de comenzar a estudiar, es común que tengamos dudas acerca de las posibilidades que nos brindará esta capacitación y sobre los requisitos que deberíamos cumplir para que nuestro trabajo fuera rentable y nos permitiera crecer. Conozcamos todas las posibilidades que tenemos.

Las posibilidades de generar ingresos y trabajo sobre la base de las necesidades que plantea Internet son cada vez más grandes, y adquieren un carácter más exigente y competitivo. Para ampliar el abanico de alternativas laborales que pueda tener cualquier diseñador, programador o autodidacta aficionado a Internet, se debe contar con las herramientas necesarias e imprescindibles de trabajo.

Si bien es posible utilizar aplicaciones casi automáticas para la creación de sitios web, logos, galerías de fotos y otros elementos para incorporar a las páginas, lo cierto es que podemos ganar mucho más dinero sabiendo hacer nuestros propios sitios y diseños sin las limitaciones que estas aplicaciones nos imponen. De esta forma, ingresaremos en un mercado donde encontraremos muchas opciones laborales, ya sea trabajando para una empresa dedicada al diseño y el desarrollo web o haciéndolo de modo independiente como profesionales **freelance**.

Esto último será posible desde nuestra casa u oficina con muy poca inversión económica, una opción ideal si queremos iniciar nuestra propia empresa. En ambos casos, es importante **tener una metodología de trabajo efectiva** y estar al tanto de las **últimas tendencias** presentes en el mundo de los sitios web, tanto estéticas como técnicas.

ÁREAS DE TRABAJO

Son muchas las formas en las que podremos monetizar nuestro trabajo como diseñadores y desarrolladores web. En este sentido, cabe mencionar que aun-

que podemos convertirnos en profesionales muy completos, no necesariamente tenemos que abarcar todas las áreas del desarrollo de un proyecto web, ya que podemos tercerizar aquellas partes del trabajo que no deseemos o podamos realizar. Gracias a esto, podemos profundizar nuestros conocimientos sobre una determinada tarea y convertirnos en verdaderos expertos del área.

Por ejemplo, muchos diseñadores se especializan, precisamente, en armar toda la estructura visual de un sitio (**layout**), creando la **interfaz gráfica** y los **banners** (animaciones en donde se promocionan productos, eventos, o publicidad de servicios y otras ofertas). Por otro lado, hay algunos diseñadores que se centran en la **diagramación**, la **usabilidad** y otros aspectos más conceptuales, como podría ser la connotación o la denotación del tema que se va a transmitir o el concepto creativo por medio del cual el sitio crea interés y se diferencia del resto. También podemos

Un profesional debe estar al tanto de las necesidades actuales de sus clientes.





UN PROFESIONAL COMPLETO

A pesar de que cada etapa del trabajo requiera conocimientos específicos, es una realidad que los requisitos que las empresas exigen a sus colaboradores o profesionales son cada vez mayores. Debido a esto, en casi todos los puestos de trabajo es preciso conocer los principios básicos de HTML. Un profesional debe contar con la información y los recursos técnicos necesarios para dar solución a las demandas de sus clientes. Además, deberá estar en constante actualización con respecto a las novedades del mercado, tendencias y software de aplicación. Esto se debe, principalmente, al crecimiento y a la importancia que tomó Internet como mercado de consumo y de oportunidades económicas. Allí las empresas invierten muchísimo dinero, y dichas inversiones deben estar avaladas por profesionales calificados.



encontrar diseñadores que, además de efectuar las tareas que mencionamos, se encargan de realizar el armado en HTML y CSS, es decir, hacen el **maquetado** de los sitios. En muchas ocasiones, también son responsables de la **integración** y la **programación** de sitios más complejos.

El hecho de que no todos los diseñadores sepan maquetar o programar un sitio y de que no todos los programadores sepan diseñarlo o maquetarlo genera muchas oportunidades de **especialización** para quienes estén interesados en ingresar en este espacio laboral que ofrece tantas alternativas. Lo que sí es claro es que quien esté involucrado en un proyecto web tiene que manejar algunos conceptos compartidos por cada uno de los integrantes de un equipo de desarrollo. En este sentido, resultarán de gran ayuda la investigación de nuevas tendencias, el estudio constante para mantenerse actualizado y la interacción con otros profesionales de diversas áreas, tales como marketing y generación de contenidos, por ejemplo.

Las posibilidades de especialización están relacionadas con todas las etapas que involucra un proyecto de desarrollo web. Este puede dividirse en diferentes instancias, que van desde la creación conceptual hasta el diseño del sitio, pasando por fases de maqueting, de integración, de programación y hasta de control de calidad. Cada una de estas etapas implica tareas para diferentes perfiles y especialistas, por lo que podemos encontrarnos con maquettadores, integradores, diseñadores, programadores, animadores y redactores, entre otras funciones. ▶

ES IMPORTANTE TENER UNA METODOLOGÍA DE TRABAJO EFECTIVA Y ESTAR AL TANTO DE LAS ÚLTIMAS TENDENCIAS DE LOS SITIOS WEB.

La notebook se transformó en una herramienta indispensable para el trabajador freelance.





DEPENDENCIA VS. FREELANCE

Ambas modalidades de trabajo tienen ventajas y desventajas que conviene analizar en detalle antes de decidir cuál elegir.

FREELANCE

- ▶ Libertad de horarios.
- ▶ Tenemos que organizarnos nosotros mismos.
- ▶ Ponemos el precio a nuestro trabajo.
- ▶ Tenemos que promocionarnos y buscar clientes.
- ▶ Debemos estar al tanto de cuestiones administrativas, presupuestos y facturación, entre otras tareas.
- ▶ Existen gastos fijos en equipamiento, telefonía, luz y otros servicios.

EMPLEADO

- ▶ Horarios establecidos de trabajo.
- ▶ Muchas veces la organización está a cargo de otra persona.
- ▶ Tenemos ingresos definidos por otro, pero son seguros.
- ▶ La empresa busca sus clientes.
- ▶ Todas estas tareas son llevadas a cabo por la empresa.
- ▶ El equipamiento es provisto por la empresa, que también abona los consumos que implica la actividad laboral.

DIVERSAS FORMAS DE TRABAJO

Según nuestra personalidad y nuestros objetivos particulares (y muchas veces también según nuestras necesidades), elegiremos alguna de las formas de trabajo que nuestra profesión nos brinda, y hasta podremos hacer un mix entre ellas. Hay quienes prefieren la tranquilidad de trabajar por su propia cuenta y así poder organizar sus horarios y proyectos, brindando sus servicios a clientes tanto nacionales como internacionales. También están quienes optan por tener la tranquilidad de

un ingreso fijo y seguro, y entonces trabajan para alguna empresa en relación de dependencia o de otro tipo, pero en forma efectiva o fija. Ambas modalidades de trabajo tienen sus ventajas y sus desventajas. Si trabajamos de manera independiente, tendremos que ser constantes y responsables con nuestros clientes. No solo deberemos preocuparnos por diseñar o desarrollar un sitio web, sino que también tendremos que encargarnos de temas administrativos, como cobros, pagos y facturación. Y por último, lo más importante:

TRABAJAR PARA EMPRESAS YA ESTABLECIDAS EN EL DESARROLLO WEB POSIBILITA LA PARTICIPACIÓN EN PROYECTOS PARA GRANDES MARCAS.



NUESTRO EQUIPAMIENTO COMO DISEÑADORES FREELANCE

Para trabajar de forma independiente, necesitaremos contar con un buen equipamiento y una oficina de reuniones, aunque esto no significa que debamos invertir mucho dinero. Además de una conexión a Internet estable con bastante velocidad, precisaremos una computadora potente: puede ser una PC de escritorio, una notebook o un equipo Mac. Aunque este último es un poco más costoso que las primeras opciones, muchos usuarios afirman que es más estable y eficaz. El equipo debería de tener la siguiente configuración:

- ▶ **Un procesador Core 2 Duo o un AMD Phenom II son suficientes, aunque un CoreQuad es ideal.**
- ▶ **La cantidad de memoria RAM es importante al momento de trabajar con aplicaciones de diseño y manipular eficientemente imágenes, videos y vectores. En la actualidad, el mínimo recomendable es de 4 GB, aunque podemos trabajar cómodamente con un equipo de 2 GB.**
- ▶ **Un disco rígido de 320 GB o de 500 GB nos alcanzará para organizar y guardar todos nuestros proyectos.**
- ▶ **Un monitor con definición de 1280 x 1024 como mínimo, y con un tamaño de 17" en notebooks o de 19" o más en monitores de escritorio para trabajar cómodamente con varias aplicaciones a la vez.**
- ▶ **Micrófono y cámara web para realizar reuniones virtuales por medio de Skype, MSN o Gtalk.**
- ▶ **Como adicionales, es recomendable tener una tableta gráfica (las mejores son las Wacom, aunque son las más caras), pendrives o discos extraíbles para hacer backups de nuestros proyectos, cámara digital, impresora y escáner.**



¡BANNERS!

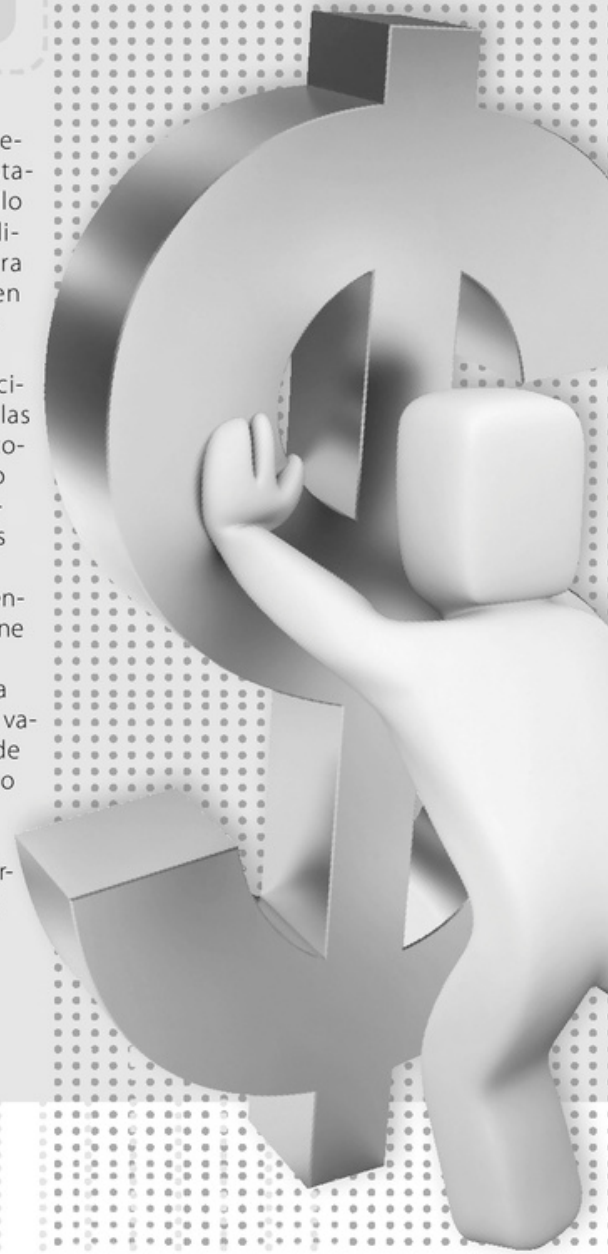
No todos los diseñadores apuntan a desarrollar sitios íntegros y complejos: hay quienes prefieren trabajar con elementos puntuales. Este es el caso de los banners, que son animaciones promocionales para publicidad y ofertas. Esta tarea es bastante entretenida y no necesita de toda la dedicación que requiere crear un sitio desde cero. Con la lectura de este libro, el profesional que trabaje con banners tendrá tareas puntuales y constantes, que podrá resolver de forma rápida y con jugosas retribuciones.

preocuparnos por encontrar nuestros futuros clientes, lo que significa que tendremos que ofrecer nuestros servicios, pactar reuniones, confeccionar resúmenes de información (briefs) y presupuestar, entre otras actividades. Uno de los mayores beneficios que tiene esta modalidad de trabajo es que seremos nosotros mismos (en relación al estándar del mercado) los que pondremos el precio de nuestra tarea, y también decidiremos la cantidad de tiempo que vamos a dedicarle o en donde la haremos.

Por otro lado, encontramos a quienes trabajan en estudios de diseño o empresas de desarrollo web haciendo labores relacionadas con alguna de las etapas de los proyectos. Entre las virtudes de esta modalidad podemos mencionar el trabajo en equipo, la cooperación en el momento de vincular nuestras tareas a las de los otros integrantes, y también el hecho de poder enfocarnos en desarrollar y especializarnos en determinados aspectos de nuestro trabajo al no tener que concentrarnos en otros temas. Y no nos olvidemos de la tranquilidad que nos brinda disponer de un ingreso fijo y seguro todos los meses.

Generalmente, trabajar para empresas especializadas que ya están establecidas en el mundo del desarrollo web hace que tengamos la posibilidad de participar en proyectos para grandes marcas. Como estos suelen ser de alta importancia y complejidad, nuestros conocimientos y nuestra experiencia se ven favorecidos, aunque también sucede que las empresas nos dan trabajos monótonos en donde siempre hacemos lo mismo y es poco probable experimentar o salirse de los parámetros establecidos.

Luego de este análisis, debemos pensar cuál de las opciones nos conviene y nos resulta más agradable para desarrollar nuestra profesión. Con la ayuda de esta obra y a medida que vayamos avanzado en la adquisición de conocimientos, aprenderemos cómo dar nuestros primeros pasos sin fallar en el intento, y qué cosas son importantes para lograr convertirnos en verdaderos profesionales en esta fascinante tarea. ◀



WORK FLOW

Veamos en esta infografía cuáles serán las diferentes etapas de nuestro trabajo.

Como diseñadores y desarrolladores web debemos conocer los procedimientos que tenemos que seguir al momento de entrevistarnos con nuestros clientes. Esto nos ayudará a comprender sus necesidades, elaborar una propuesta que se adapte a ellas, y hacer una correcta evaluación del nivel de diseño y la complejidad de desarrollo que demandarán esos sitios. De esta manera, podremos presupuestar adecuadamente nuestro trabajo. Veamos a continuación el desarrollo completo de estos pasos a seguir. ◀

CONSEJOS PRÁCTICOS

Un modelo de procedimiento ideal para el desarrollo web ayudaría al diseñador (webmaster) a superar la complejidad o fracaso del sitio, a minimizar el riesgo de fallo del proyecto, a encargarse de los cambios que, con toda certeza, surgirán durante su realización y a entregar el sitio con rapidez, proporcionando la información necesaria para poder realizar toda la gestión durante el proceso.

06

ENTREGA

Tras la presentación al cliente de la versión definitiva del sitio, con su correspondiente aprobación y testeo, se procede a alojarlo en su servidor definitivo. Se vuelve a verificar su funcionamiento y configuración y, si todo está perfecto, se considera finalizado el proyecto.

TESTEO

05

Debemos tener presente la asignación de un tiempo de control de calidad, para testear el buen funcionamiento del sitio, y verificar que se hayan respetado los lineamientos y objetivos planteados al inicio. En este caso, se suele disponer de una persona encargada específicamente de la tarea.

DINÁMICA DE TRABAJO DE LOS DISEÑADORES



Comenzaremos a construir el sitio, para lo cual deberemos tener en cuenta detalles técnicos. Empezaremos con el maquetado en HTML y, luego, pasaremos a la programación en otros lenguajes, como PHP o JavaScript, si es necesario. En el caso de Flash, tendremos que armar la estructura del sitio, la programación de la interfaz, y las interacciones y animaciones requeridas.

DESARROLLO

04

01

DEFINICIÓN DEL PROYECTO

El paso más importante en el proceso de diseño web es la definición precisa del proyecto, que incluye un resumen y un plan de él. La reunión con el cliente sirve para recopilar toda esta información de manera que podamos establecer el objetivo y las necesidades del proyecto.



02

ESTRUCTURA DEL SITIO

El desarrollo de un buen plan de trabajo, y por consiguiente, la concreción de un buen sitio web requieren de la confección en detalle de la estructura y sus contenidos; siempre basándonos en los briefs conceptuales y de información que creamos previamente.

ANALIZADORES Y DESARROLLADORES WEB



03

DISEÑO

Este es el momento de ser creativos. Sobre la base de toda información relativa al sitio, su objetivo, target, estructura y demás factores, debemos diseñar y dar rienda suelta a nuestra imaginación, siempre teniendo en cuenta las referencias técnicas y comunicacionales involucradas en el proyecto.



MANTENIMIENTO

El mantenimiento postventa es fundamental para proveer a nuestros clientes de soluciones rápidas y contenidos actualizados. Ante los imprevistos que pueden surgir por diferentes motivos, es importante reaccionar de inmediato para que ellos no se vean perjudicados en sus negocios.

¿EMPREENDEDOR, EMPLEADO O FREELANCE?



Antes de aplicar los conocimientos técnicos revelados en esta obra, es importante entender las diferencias, ventajas y desventajas de cada una de las formas de práctica profesional.

Como ya hemos mencionado, podemos encarar nuestra profesión de diseñadores web de diferentes maneras. En cada una de ellas desarrollaremos distintas tareas, además del diseño propiamente dicho. Para elegir con criterio, lo ideal es conocer bien las características de cada una de estas posibilidades a la hora de desempeñarnos.

EL EMPLEADO

El empleado es una persona que intercambia su tiempo y esfuerzo por un salario justo, no menor al mínimo legal estipulado por la ley. En general, las ventajas de esta modalidad están relacionadas con los derechos que se adquieren por ser trabajador asalariado. Por ejemplo, en el caso de la República Argentina, estar registrado como



TRABAJO EFECTIVO

Otro aspecto importante del régimen laboral argentino está relacionado con el período de prueba de los trabajadores. Los empleados deben ser registrados como tales desde el primer día de trabajo, y el período de prueba no puede exceder los tres meses, ya que un día después de ese plazo se considerará que la persona está efectiva en su puesto. De hecho, está penada la conducta de contratar sucesivamente a diferentes trabajadores por tres meses con el objetivo de evitar la efectivización en un puesto permanente.

empleado da derecho a recibir aportes jubilatorios, cobertura médica para sí y la familia a cargo, vacaciones pagas, y licencia con goce de sueldo por maternidad, nacimiento o enfermedad. Además, se recibe un sueldo anual extra (aguinaldo) y compensaciones en caso de lesiones o enfermedades derivadas del trabajo. Por otro lado, y aunque no siempre se

cumple, la ley del mismo país protege al empleado de ciertos excesos, como jornadas laborales de más de ocho horas diarias o 48 horas semanales (exceptuando casos específicos, como las tareas agrícolas o el trabajo en pequeñas empresas familiares). Las horas adicionales se consideran como suplementarias, y se recibe por ellas su valor más el 50% en días normales, y el doble en días feriados o no laborables.

Además de los derechos, el empleado está unido a su empleador también a través de obligaciones, como las de cumplir un horario y los objetivos que le sean impuestos, prestar fidelidad, reconocer la autoridad del empleador y aceptar órdenes, entre otras.

Como vemos, la vida del trabajador en relación de dependencia dista mucho de

Si trabajamos para una empresa, el empleador será quien nos provea las herramientas de trabajo. Si somos freelance, deberemos adquirir nuestros propios equipos.



navegar en un río de tranquilidad. Para evitar posibles atropellos, ante cualquier duda sobre una propuesta de trabajo, es conveniente consultar la legislación laboral vigente en el país o recurrir a algún asesor legal o abogado.

FREELANCE

Es una persona que también intercambia su trabajo por dinero, pero solo lo hace para una tarea puntual, derivada de su oficio o profesión, en un período determinado de tiempo. Si bien iremos exponiendo las particularidades de esta modalidad (junto con las del emprendedor) a medida que avancemos en esta sección, repasaremos algunas de ellas como introducción. La diferencia decisiva con el modelo anterior reside en que todo lo relacionado con el servicio queda bajo la responsabilidad del trabajador freelance, que debe responder ante los clientes, los proveedores y el Estado. Por consiguiente, además de un asesor legal, quien trabaja por cuenta propia necesita contar con un asesor de finanzas, ya que seguramente deberá registrarse como trabajador independiente en organismos tributarios gubernamentales. En algunos países, existen regímenes de impuestos especiales y simplificados para estos pequeños contribuyentes, lo que evita recurrir con frecuencia a los profesionales de la contabilidad. En muchos lugares, los sistemas tributarios contemplan categorías especiales para la inscripción de trabajadores por cuenta propia (también llamados autónomos o independientes). En cualquier caso, es importante investigar y buscar asesoramiento con respecto a la

forma correcta de inscripción, debido a que así se tiene la tranquilidad de estar operando de acuerdo con la ley, se eleva la imagen de seriedad y profesionalismo, y se accede a la posibilidad de emitir comprobantes impresos (generalmente llamados facturas) por los servicios prestados a los clientes.

EL EMPRENDEDOR

Este rol tiene más puntos en común con el trabajador freelance que con el empleado, aunque se diferencia por su misión. El emprendedor busca la conformación

de una empresa, nacida de una idea que percibe como una oportunidad genuina de crecimiento. Esta empresa podrá vender servicios o productos, pero estos en general no serán fruto de la labor directa del emprendedor, ya que no serán hechos por sus propias manos. En cambio, este se ocupará de organizar, administrar y dirigir eficientemente los recursos disponibles para buscar la máxima rentabilidad económica (u otros objetivos, si no tiene fines de lucro), satisfaciendo las necesidades de su público a través de su oferta. ◀



¿CUAL OPCIÓN ES LA ADECUADA?

En muchos casos, los profesionales del diseño web pasan, de manera secuencial, por todas estas modalidades en alguna etapa de su vida. Al principio trabajan como empleados, adquiriendo experiencia en una empresa ya conformada, donde pueden aprender sobre cuestiones prácticas específicas que, por lo general, son poco profundizadas en las instituciones educativas. En esta instancia, conforman un primer portfolio personal. Más adelante, abandonan la empresa para desarrollarse de forma freelance. Esto suele suceder cuando alcanzan el techo de su crecimiento profesional o, quizá, debido a injusticias sufridas y a falta de reconocimiento. En general, resulta atractiva la idea de libertad horaria y organizativa, además de la posibilidad de tener mayores ingresos y desafíos. Finalmente, cuando aparece una idea valiosa y diferenciadora que necesita de una organización más compleja, se alcanza el rol de emprendedor, formando equipos de personas y realizando una inversión de capital para cumplir el objetivo propuesto.



A la hora de trabajar, es importante contar con un lugar cómodo y tranquilo, además de ordenado.

FREELANCE: ¿CÓMO INICIARSE?

Todos los nuevos comienzos requieren de mucha reflexión y análisis, ya que en su recorrido, nos encontramos con diversas etapas de aprendizaje. Para trabajar de manera freelance, debemos adquirir algunas destrezas que nos brinden mayor rendimiento. Veamos cuáles son.

Ser un trabajador freelance significa emprender un camino de responsabilidades y de compromisos en el que deberemos desempeñarnos como nuestros propios jefes y expresar al máximo nuestras habilidades. Un trabajador freelance desarrolla su profesión de manera independiente y, por lo general, a través de encargos sin vinculación contractual. Si deseamos comenzar nuestra carrera bajo esta modalidad, es aconsejable tener en cuenta las siguientes recomendaciones y tips.

EL MOMENTO

Una vez que hemos tomado la decisión y queremos iniciarnos como profesionales freelance, debemos focalizarnos en crecer y en darnos a conocer. Es importante presentar proyectos y recurrir a diversos medios para ofrecer nuestros servicios y llegar a potenciales clientes. Este es, quizás, uno de los principales objetivos de la actividad freelance: hacer que nos conozcan quienes estén necesitando de nuestro trabajo.

PRESENTACIÓN PROFESIONAL

Utilizar una tarjeta de presentación es casi obligatorio. La entregaremos en reuniones de trabajo, en encuentros con colegas y a todo aquel que consideremos como potencial interesado en nuestros servicios, aunque no lo esté en el momento.

Es importante dedicarle el tiempo necesario al diseño de este elemento, porque en muchos casos será la primera muestra de lo que podemos realizar como diseñadores.

ESTABLECER UN HORARIO DE TRABAJO

Trabajar de manera freelance nos permite manejar nuestros horarios y coordinar cómo queremos organizar el tiempo dedicado a cada uno de los proyectos. Sin embargo, es muy recomendable fijar el horario laboral que vayamos a cumplir diariamente y durante el cual limitemos al máximo las posibles distracciones. Dentro de ese lapso, ordenaremos nuestro tiempo para poder llevar a cabo todos los trabajos en los plazos estipulados.

ACOMPañAR A LOS CLIENTES

Cuando un cliente nos contrate, tendremos una reunión con él al comienzo del proyecto o, incluso, un acuerdo a distancia a través de Internet. En cualquier caso, es importante mantenernos bien comunicados durante todo el desarrollo del emprendimiento, enviando

resultados parciales e informando los adelantos obtenidos. De este modo, el cliente se sentirá respaldado y tranquilo, y podremos solucionar inconvenientes con mucha más facilidad si estos son detectados rápidamente.

ORGANIZARNOS

Ya hemos hablado de lo importante que es mantener una rutina de trabajo bien clara. Además, en el momento de trabajar, es fundamental tener a nuestra disposición todas las herramientas y los materiales que podamos llegar a necesitar. Siempre debemos organizar y ordenar nuestro espacio de trabajo, colocando al alcance lo más necesario y quitando del medio lo que no precisemos con frecuencia. Este orden nos permitirá enfocarnos más en nuestras tareas, y así aprovechar mejor el tiempo y el espacio de que disponemos.



En **www.rzmota.com** encontramos un diseño bastante creativo para ofrecer servicios de diseño web.

ACTUALIZARSE CONSTANTEMENTE

Vivimos en un mundo que cambia todo el tiempo, donde las tecnologías que se utilizan son reemplazadas en breve por otras nuevas y más eficientes. Es por eso que, cuando iniciamos nuestro camino como profesionales freelance, también debemos comenzar una exploración continua de nuevos recursos, observando los proyectos y las tecnologías que se van presentando para, así, ofrecer siempre lo último a nuestros clientes.

CREAR **PRESTIGIO** EN LÍNEA

Integrarse en comunidades online –como Twitter, Facebook, LinkedIn, Freelancer y FreelanceSwitch– nos permitirá socializar con colegas de nuestro ámbito y dar a conocer los trabajos que hagamos. A medida que interactuemos con quienes tienen nuestras mismas aficiones, compartiremos opiniones e ideas, además de experiencia. Esta es una excelente manera de darnos a conocer e, incluso, de descubrir nuevos proyectos en los que podremos participar.

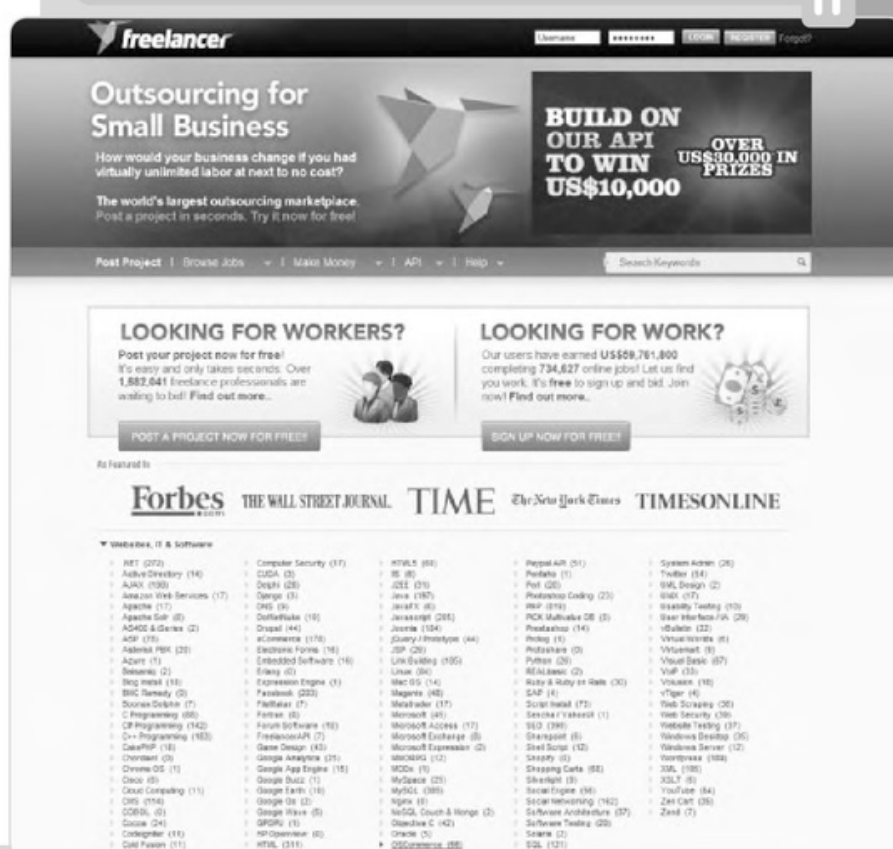
PRESUPUESTAR CORRECTAMENTE

Una de las mayores dudas que tienen los profesionales freelance está relacionada con lo que deben cobrar por su trabajo. La realidad es que debemos hacer valer la calidad de lo que realizamos, ofreciendo un precio razonable y justo. No necesariamente vamos a tener clientes si cobramos poco, ya que lo barato puede ser analizado como algo que no es realmente bueno. Es por eso que precisamos investigar un poco para establecer un precio que se encuentre dentro del promedio de lo que se cobra en el mercado. ◀

APRENDER DE CADA TRABAJO

Cuando trabajemos de manera independiente, seguramente aprenderemos algo nuevo en cada una de las tareas que realicemos. En todos los casos, será muy útil tomar las posibles barreras que aparezcan en el camino como nuevos desafíos por afrontar. Así, iremos resolviendo inconvenientes y aumentaremos nuestra experiencia.

Redes como Freelancer (www.freelancer.com), LinkedIn (www.linkedin.com) y FreelanceSwitch (www.freelanceswitch.com) están especialmente dedicadas a los profesionales independientes.



FREELANCE:

ADMINISTRAR EL TIEMPO

Anteriormente, mencionamos la singularidad principal de esta modalidad de trabajo: todas las tareas relacionadas directa e indirectamente con la actividad recaen en la responsabilidad del profesional. Veamos cómo administrarlas.

Uno de los primeros problemas para el trabajador freelance se presenta cuando debe lidiar con los trajines del negocio, ya que muchos de ellos son ajenos a su preparación académica. El tiempo se vuelve repentinamente escasísimo, pues además del trabajo de creación, programación y diseño, debe encarar tareas como las siguientes:

► **Llevar registros contables y de situación fiscal.** Esto implica registrar la facturación y el flujo de caja (o movimiento de ingresos y egresos de dinero), controlar los gastos y economizar; vigilar la situación del crédito,

de los depósitos y las deducciones bancarias; y realizar un seguimiento de la situación tributaria.

► **Cumplir con el pago de impuestos y servicios.** Los olvidos podrían conducir a deudas que generarían intereses y a cortes de suministros, lo cual elevaría los gastos.

► **Elaborar e implementar un plan de marketing.** Básicamente, esto significa planificar, para un tiempo delimitado, la forma en la que el público conocerá el servicio ofrecido y las razones para elegirlo. El objetivo de esta tarea es mantener un flujo de trabajo constante, que permita planificar inversiones y llevar adelante una vida normal.

► **Negociaciones y gestión de clientes.** Esta actividad cobra más relevancia, cuanto mayor complejidad, plazos de ejecución e inversión económica impliquen los desarrollos encomendados. Es frecuente que los profesionales bien posicionados encuentren sus mañanas totalmente ocupadas con reuniones.

► **Gestionar la cobranza.** Esta tarea, que en general queda relegada, es de importancia capital. Muchos clientes son muy exigentes en cuanto al cumplimiento en tiempo y forma del servicio, pero se revelan indiferentes a la hora de pagar por él. Entonces, el profesional deberá destinar parte de su tiempo a hacer un seguimiento de las cobranzas, tiempo que se multiplica a medida que se tienen más clientes.

Es posible que, en un comienzo, el diseñador freelance pueda realizar eficazmente cada una de estas tareas, pero encontrará dificultades al incrementar su cartera de clientes.

Lo que sucede entonces, si no se toman medidas precautorias, es que todas estas actividades pasan a relegarse para cumplir con algún servicio que ya está en marcha. Así, lo urgente quita tiempo a lo importante o, en otras palabras, lo que sirve a corto plazo se impone a lo que serviría a largo plazo.

Ante esta situación, es útil que el profesional despiece y grafique todas las tareas que forman parte de su quehacer y las relaciones existentes entre ellas. Incluso, puede hacer este análisis con el proceso de creación, diseño y programación, para obtener un panorama completo y tomar conciencia real del vasto campo de operaciones que debe llevar a cabo. El paso siguiente consistirá en decidir cuáles de esas tareas debe realizarlas por sí mismo, siguiendo una jornada laboral que le permita tener una vida personal y social satisfactoria, al mismo tiempo que obtiene los ingresos económicos esperados.

ORGANIZARSE Y DELEGAR TAREAS AYUDA A LIBERAR TIEMPO QUE SERVIRÁ PARA BUSCAR MÁS CLIENTES Y DESARROLLOS.





Luego, con estas tareas bien definidas, será tiempo de buscar ayuda. Por ejemplo, existen jóvenes profesionales de la contabilidad y la abogacía que se especializan en asesorar a trabajadores independientes. Mediante un pago mensual accesible, se encargan de llevar los registros de facturación, realizar seguimientos tributarios e impositivos, confeccionar contratos de prestación de servicios e, incluso, hasta de gestionar las cobranzas. También es posible recurrir a profesionales de corta trayectoria (recién recibidos) en el campo del marketing de servicios y negociar con ellos un asesoramiento. Algo similar ocurre con tareas simples que deben realizarse dentro de la programación y el diseño web. Ciertas partes de un desarrollo pueden encargarse a terceros, como algunas funcionalidades específicas, animación, vectorización y optimización de imágenes, ilustración, fotografía, redacción y carga del contenido, entre otras. De este modo, liberaremos tiempo propio para buscar más clientes y concretar más proyectos.

En suma, repasemos las recomendaciones más importantes:

- No debemos intentar hacer todo, sacrificando algún aspecto de nuestra vida personal.
- Es bueno confiar en otros profesionales que puedan asistirnos y a quienes delegarles parte de nuestro trabajo. Los profesionales independientes, lejos de ser trabajadores solitarios, funcionan mejor en equipo.
- La meta es mantener un flujo constante de ingresos monetarios. Pocas cosas llenan más de incertidumbre que el hecho de no saber qué sucederá luego de haber finalizado un encargo.
- Ciertas tareas no deben delegarse, sobre todo, la de gestión de los clientes y la negociación con ellos. Es importante tener en cuenta que el cliente contrata a un freelance, entre otras cosas, porque quiere tratar directamente con una persona que le asegure que dará todo su esfuerzo para obtener el mejor resultado.
- Ante cada tarea, es bueno preguntarse si se trata de algo importante o algo

urgente. Si la segunda alternativa se lleva la mayor parte de las respuestas, no nos alcanzará el tiempo ni la jornada de trabajo; todo deberá ser resuelto en el momento y perderemos la visión a largo plazo. Esta es la visión que nos impulsa a seguir, la que nos dice en dónde estamos, dónde queremos estar y cómo haremos para llegar ahí. ◀



EMPLEADO VS. FREELANCE

Como ya vimos, es el empleador quien se ocupa de las cuestiones financieras, administrativas, operativas, legales y de gestión de los clientes. También es él quien afronta los riesgos propios de la actividad empresarial. En este modelo, el quehacer del profesional está bien delimitado; el empleado debe cumplir con su tarea específica en la franja horaria pactada y, luego de ella, finaliza su jornada laboral.

FREELANCE: CONSEGUIR CLIENTES ONLINE

La Web es una gran fuente de herramientas a la hora de promocionarse y darse a conocer en línea. Aquí veremos algunas de las más útiles y famosas en la actualidad.

Los empresarios, gerentes y potenciales clientes en general están cada vez más conscientes de la importancia de Internet a la hora de asistirlos en la búsqueda de proveedores. Conozcamos las causas por las cuales esto sucede:

- **Ahorro de tiempo y dinero:** en solo un par de días, es posible conseguir varios presupuestos para el mismo proyecto, de cualquier estudio de diseño del mundo y sin costo alguno.
- **Variedad de ofertas:** además de la mayor cantidad de presupuestos, se consiguen más propuestas de ofrecimientos laborales que por los medios tradicionales o por recomendación.
- **Mayor información sobre cada potencial proveedor:** la posibilidad

de recorrer el portfolio, ver resultados en los buscadores web y pedir referencias en las redes sociales permite conocer el tipo de servicio, su calidad y la cartera de clientes.

EL PORTFOLIO ONLINE

A la hora de definir un portfolio en línea, hay una serie de preguntas que nos serán de utilidad para decidir cómo presentaremos la propuesta de servicios.

En primer lugar, debemos saber exactamente cuál es el servicio que ofrecemos y cuáles son sus principales ventajas, además de conocer los servicios que otros

profesionales pueden brindar.

También es importante plantearse por qué un cliente elegiría nuestro servicio y no otro similar, tal vez, de menor precio. En este sentido, debemos saber de qué manera nos diferenciamos de la competencia y en qué somos mejores.

Otro aspecto para tener en cuenta es el tipo de cliente al que dirigimos nuestras comunicaciones, y pensar en la clase de organización para la que nos gustaría trabajar. Lógicamente, no es lo mismo dirigirse al dueño de un molino aceitero que al director creativo de una agencia de publicidad.

Tenemos que investigar para saber qué buscan los clientes en un profesional especializado en la misma área que nosotros, es decir, qué necesidades poseen y deben cubrir. En este sentido, debemos elegir brindar la información que les resultará más útil e interesante, aquella que precisarían para favorecernos con su elección.

Un aspecto que no debemos perder de vista es la competencia: quiénes son, cómo se muestran y qué dicen de sí mismos es un dato relevante. Hoy en día, la competencia no se limita al



ámbito local, sino que se extiende a todos los profesionales que estén al alcance a través de Internet, aquellos que provean soluciones a la misma necesidad, en un rango aproximado de precios.

LAS COLABORACIONES EN PORTALES

Colaborar en portales de la especialidad mediante artículos y comentarios es una forma efectiva de elevar nuestra posición en el ranking de los buscadores, siempre y cuando sea posible incluir el vínculo al sitio web del portfolio. Buscadores como Google asignan mayor importancia a los sitios a los cuales apuntan vínculos de otros que tengan una gran cantidad de visitas. En efecto, los portales dinámicos más buscados serán aquellos que tengan más visitas, al menos, dentro de una especialidad relacionada, y que muestren una mejor posición en los resultados.

Una variante de la colaboración en portales consiste en la participación activa en foros y blogs, respondiendo inquietudes y generando debates. Es una buena idea intercambiar hipervínculos textuales con los responsables de los blogs, a menos que sean nuestra competencia directa. Asimismo, es posible investigar el tipo de publicaciones online que frecuentan nuestros potenciales clientes e intentar escribir contenido para ellas, aunque sea de forma gratuita. Por ejemplo, si deseamos llegar al público gastronómico, una posibilidad interesante es escribir artículos acerca de las formas en que Internet puede llegar a potenciar sus ventas y a mejorar la fidelidad de su público.

FACEBOOK

La primera decisión que deberíamos de tomar al crear una cuenta en esta red está relacionada con la finalidad de su uso, que podría ser solo placer, o un intento por lograr contactos profesionales y buscar clientes.

Muchas veces, los profesionales dedican grandes esfuerzos a construir una imagen de seriedad, confianza y responsabilidad en su portfolio personal; imagen que luego desaparece al mostrar fotografías de fiestas alocadas, situaciones embarazosas o borracheras, todo en el mismo perfil de Facebook que comparten con clientes actuales o potenciales, proveedores y empresas. Para evitar este problema, lo ideal es separar ambos mundos creando una página especial para la faceta profesio-

nal. Allí, atraeremos público mediante invitaciones, noticias de interés, análisis de casos de éxito, artículos útiles, etcétera. Otra opción es contar con dos perfiles, o solo con uno pero dedicado al ámbito profesional.

Si participamos con frecuencia en los grupos de interés para aquellos que queremos convertir en clientes, aumentaremos la visibilidad de nuestro nombre en la red social, y luego podremos invitar a los miembros del grupo a que visiten nuestro portfolio.

DEBEMOS PRESTAR ATENCIÓN
A NUESTRA PRESENCIA **ONLINE**
YA QUE **INTERNET ES**
MUY UTILIZADA POR
LAS EMPRESAS PARA
BUSCAR PROVEEDORES.

LAS REDES SOCIALES PROFESIONALES

A diferencia de la red anterior, **LinkedIn** (www.linkedin.com) es una red social online creada para generar relaciones comerciales. Permite buscar ofertas laborales; agregar a colegas de trabajo, clientes y proveedores; y pedirles recomendación para llegar a otros con una propuesta. Además, posee un completo sistema de sugerencias de contactos

según la localización geográfica, las profesiones y los trabajos en común. Es posible instalar plugins para mostrar presentaciones, encuestas, contenido del blog WordPress y hasta sincronización con Twitter.

LAS BOLSAS DE TRABAJO

Las bolsas de trabajo online constituyen una gran fuente de ofertas laborales, sobre todo, para profesionales dedicados a la programación de aplicaciones web y a analistas en sistemas. ZonaJobs (www.zonajobs.com) es una de las bolsas más grandes en Sudamérica y posee sitios diferenciados por país (Chile, Argentina, Colombia, México e, incluso, España). Permite crear un perfil y un CV de forma gratuita, además de ofrecer la tradicional búsqueda mediante filtros. Afirmar ser utilizado por las empresas más importantes de cada nación en la que está presente. Por su parte, la bolsa de empleo argentina Bumeran (www.bumeran.com.ar) busca posicionarse como la primera **emplotter** del mundo. En un ingenioso cambio de enfoque, integra la dinámica de la Web 2.0 con la búsqueda de empleos. Propicia que los postulantes afines compartan opiniones y experiencias sobre las empresas oferentes, a la vez que estas últimas pueden compartirlas sobre mejores candidatos, salarios justos, mejores prácticas y características del puesto en cuestión. ◀



FREE LANCE

Veamos el camino que debe recorrer el profesional del trabajo independiente.

NUEVOS CLIENTES

Podemos encontrar nuevos clientes de diversas formas; entre ellas:

Comunidades freelance en donde se publican proyectos y nosotros debemos ofrecer nuestros servicios y costos compitiendo con el resto de los usuarios para lograr la mejor relación calidad / precio.

Recomendaciones: es una de las formas más habituales, en la cual nuestros clientes nos recomiendan a otros por nuestras cualidades.

Contacto directo: somos nosotros quienes contactamos directamente a nuestros posibles clientes, ya sea vía e-mail, telefónicamente, por medio de publicidad, flyers, banners en otros sitios o avisos clasificados.

COMUNICACIÓN

Métodos de comunicación

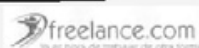
Es indispensable contar con varias vías de comunicación con nuestros clientes, ya sea por medio de e-mails, mensajería instantánea, telefonía IP o videoconferencias.



Trabajo

CLIENTE

La entrega del trabajo al cliente se hace de acuerdo con los plazos pactados.



Lugar de TRABAJO

Una de las ventajas principales de ser freelance es poder elegir cuando y dónde trabajar. Ya sea en un parque, en nuestra casa u oficina, solos o en colaboración con otros profesionales, estas son decisiones propias de cada uno.



Ps

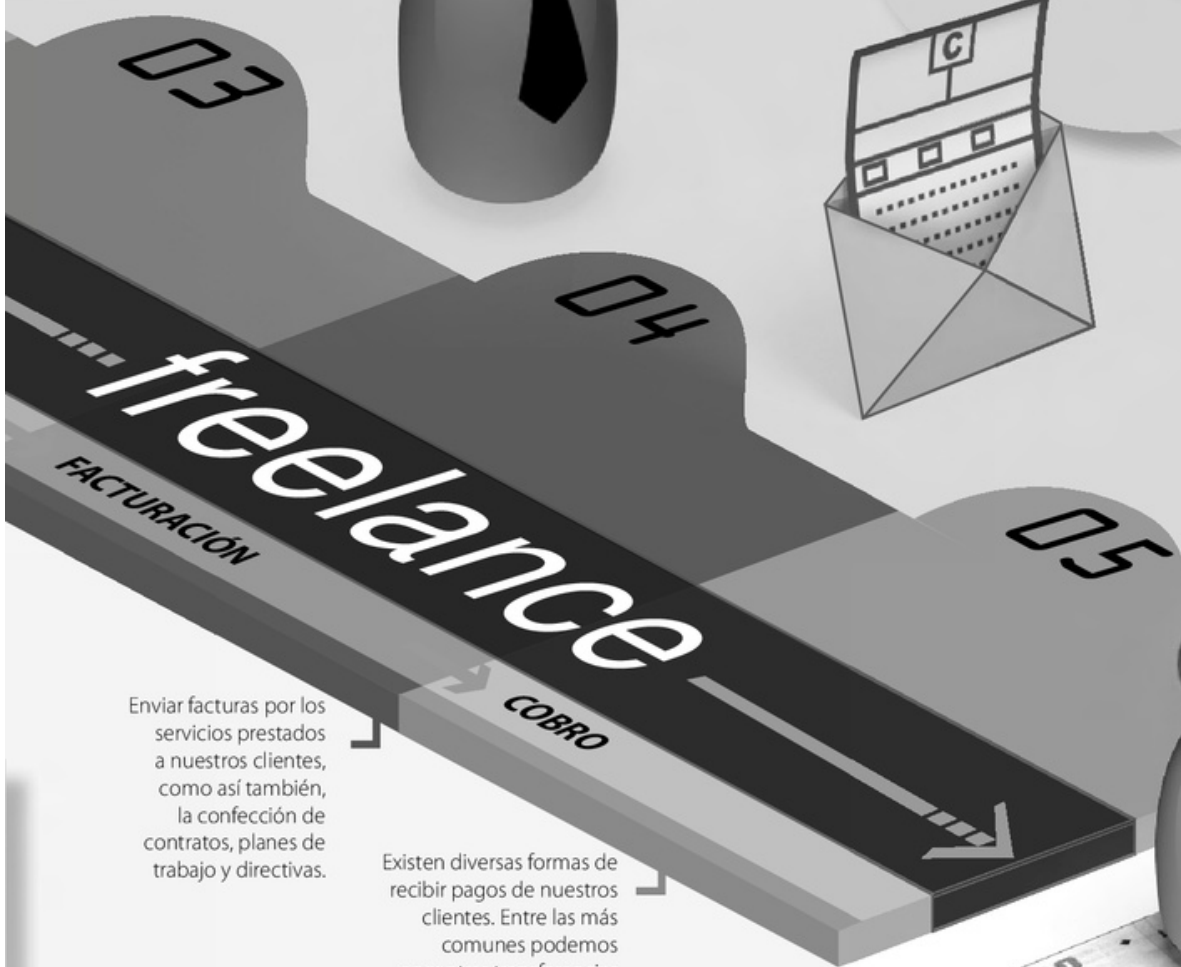
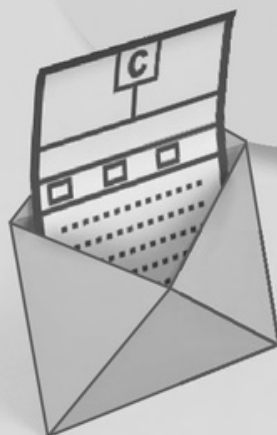
Dw

Fl

Ai

Elementos de TRABAJO

Los elementos básicos que todo trabajador freelance debe tener son una computadora, acceso a Internet, micrófono para realizar llamadas online y las aplicaciones necesarias para efectuar sus tareas (por ejemplo, Photoshop, Dreamweaver, y Flash, entre otras).



Enviar facturas por los servicios prestados a nuestros clientes, como así también, la confección de contratos, planes de trabajo y directivas.

Existen diversas formas de recibir pagos de nuestros clientes. Entre las más comunes podemos encontrar transferencias bancarias, pagos online con PayPal, envío de dinero por correo postal o tarjetas de crédito.

PayPal DineroMail

EMPRENDEDOR, ¿YO?

En esta sección, conoceremos algunas de las claves para definirnos como líderes emprendedores. Esto es fundamental si queremos comenzar nuestra propia empresa y ser exitosos en ella.

En algunas fundaciones para emprendedores, se practican tests a los postulantes para intentar averiguar si tienen una **personalidad emprendedora**.

Claro que esto no significa que quienes no posean estas características se vean imposibilitados de comenzar una empresa, ya que es posible desarrollarlas. Esto es ideal porque, en algún momento, estas cualidades serán necesarias para que la personalidad de los directivos no sea un gran obstáculo, sino un atajo.

CONFIANZA EN SÍ MISMO

Aunque no pensemos en cientos de años, no sabemos qué sucederá en el futuro; de hecho, no podemos saber con certeza qué ocurrirá en la próxima hora. Como el futuro aún no ha sucedido, solo es posible especular sobre distintos escenarios en alguna ínfima porción de la vida.

Ante este hecho, quienes poseen confianza en sí mismos tenderán a vivir más en el presente, pero también a imaginar escenarios optimistas y favorables, incluso, sin saber muy bien cómo se producirán. Con esta confianza, saben que llegarán a destino aunque solo vean los primeros metros de la ruta, como ocurre cuando se conduce en la noche. Como frase de cabecera, estos individuos dicen "de algún modo va a salir bien". Seguramente, tienen miedo y mantienen sus sentidos alertas, pero esto no logra paralizarlos. La confianza en sí mismos y, por consiguiente, en sus ideas, los impulsa a ir más allá, los lleva a tomar decisiones y a ser felices con esa situación.

INQUIETUD

El emprendedor es inquieto y, como invadido por hormigas, quiere avanzar. Se interesa por los nuevos desarrollos en su materia y los capitaliza en forma de proyectos. Está informado y es muy curioso. Se muestra capaz de detectar aquellas noticias y tendencias que le proveerán de ideas para un próximo proyecto, para mejorar la empresa o para diferenciarse de la competencia.

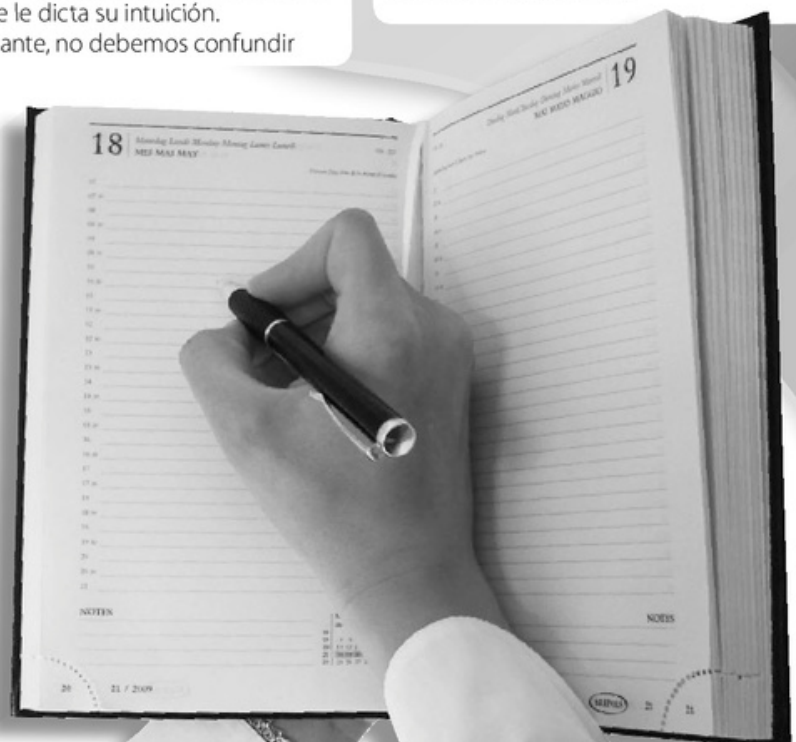
Este tipo de persona puede abandonar con facilidad su **área de confort**, el lugar en donde se siente seguro y con mínima incertidumbre, para explorar en la dirección que le dicta su intuición.

No obstante, no debemos confundir

inquietud con ansiedad. A diferencia de la inquietud, la aceleración de la ansiedad, en este caso, puede provenir de un estado de alerta, producto de creer que el futuro deparará algún peligro que puede superar la capacidad de la persona para solucionarlo. Esta ansiedad puede prevenir la acción positiva, en vez de favorecerla.

CONVIVENCIA CON EL RIESGO

Son varios los riesgos que un emprendedor tiene, incluso, desde el momento mismo en que decide serlo. Algunos de ellos son los siguientes:



► **Riesgo de emprender:** aunque se haya reflexionado, planificado y participado de varios cursos, nadie puede asegurar que todo se dará según lo calculado. La incertidumbre del futuro siempre está presente.

► **Riesgo de decidir:** sobre todo al principio, no se cuenta con toda la información necesaria y de todos modos hay que tomar decisiones. La tensión que esto genera será directamente proporcional al nivel de seguridad del emprendedor.

► **Riesgo económico y temporal:** existe el riesgo de perderlo todo (tiempo, dinero y relaciones) si el emprendimiento no alcanza los objetivos mínimos en un tiempo máximo.

► **Riesgo de la decepción:** preguntas como ¿qué haré si todo va mal?, ¿qué dirán los demás? y ¿qué pensaré de mí mismo? pueden llegar a bloquear la acción y a pensar que, directamente, es mejor no intentar emprender nuestra empresa.

LA VINCULACIÓN DEL PÚBLICO
CON UNA **EMPRESA** SE
BASA MUCHO MÁS EN LAS
RELACIONES PERSONALES
QUE EN EL **DINERO**.

SOCIALIZACIÓN

Mantener una empresa supone un continuo intercambio con diferentes públicos: proveedores, clientes, futuros clientes, empleados y socios, entre otros.

Aquellos que sean solitarios, con carácter duro, apático y áspero, sumarán un obstáculo importante. Las vinculaciones de una empresa con su público se basan mucho más en las relaciones personales que en el dinero, aunque no lo parezca. Si solo se sustentaran en este último factor, nociones como **lealtad**, **favor** y **entendimiento** carecerían de sentido. El público confía en la persona del emprendedor, en la simpatía y en la confianza que le despertó.

Si lo meditamos sobre experiencias propias, podemos recordar que cuando ingresamos a un local a comprar ropa, la actitud del vendedor resulta determinante. A veces, su carácter hace que compremos más de lo que pensábamos y que deseemos regresar. En otras ocasiones puede hacer que pasemos un



EL PROYECTO DE SITIO

Para reforzar nuestros conocimientos y poner en práctica nuestras habilidades podemos visitar el sitio de proyecto final en la versión **HTML** (<http://www.redusers.com/cursos/webdesign>), para conocer todos los detalles del código y maquetado. Además encontraremos un **CD interactivo** disponible para la descarga con **guías visuales**, **glosario de términos**, y una completa guía de **preguntas frecuentes**. Con los videotutoriales perfeccionaremos nuestras habilidades trabajando con las aplicaciones de la suite de diseño de **Adobe Web Premium CS5**. Con estas herramientas, nos convertiremos en los mejores profesionales.

mal momento, que no gastemos nada y que les contemos a nuestros conocidos lo mal que se atiende allí.

LIDERAZGO

El equipo de trabajo de una empresa necesita un líder con visión de futuro, capaz de establecer y compartir objetivos. Este líder debe motivar y estimular a los miembros para dar voluntariamente lo mejor de sí mismos. Abre el juego, permitiendo la participación y la toma de responsabilidades. El líder no se siente intimidado frente a críticas constructivas: las enfrenta y busca el acuerdo. Así, se transforma, en definitiva, en un referente, en alguien a quien las personas siguen y recurren ante cualquier problema. ◀



PLANIFICAR UN PORTFOLIO EFECTIVO

Ya vimos la importancia que tiene la creación de un portfolio online para un desarrollador web. En esta ocasión, analizaremos las estrategias para poseer uno interesante y atractivo destinado a los potenciales clientes.

En estos días, es imposible pensar en el portfolio profesional como una pieza estática y solitaria en el extenso océano de la Web actual. Por el contrario, la tendencia refleja que, cada vez más, se planifica una estrategia de marketing profesional online en la que el sitio web es solo una de las acciones destinadas a llegar al público potencialmente interesado en el servicio. Como ya hemos visto, a la hora de pensar el portfolio, resulta de suma utilidad la respuesta sincera a preguntas sobre el servicio que ofrecemos y el tipo de cliente al que queremos llegar. Luego, será hora de planificar y articular la comunicación de la oferta en redes sociales, en portales de información de diseño, en portales gratuitos de **microportfolios** (como www.illustrationmundo.com, www.ilovedesign.com y

www.deviantart.com), y en los de búsqueda de empleo y de clasificados profesionales; y, por supuesto, en el sitio web propio.

ORGANIZAR LA INFORMACIÓN

Al definir y conocer en profundidad el tipo de público que podría encontrar un valor genuino en el servicio ofrecido, estamos en condiciones de determinar la clase de material que espera encontrar en un portfolio; es decir, aquella información que le interesa para llegar a una contratación. Cualquier otro contenido solo contribuirá a complejizar la estructura, diluir el interés y aumentar la densidad de datos presentes. El potencial cliente se preguntará si somos nosotros quienes

podemos solucionar su problema, si seremos quien logre que su organización alcance la preferencia del mercado. Es importante tener esto en mente, ya que los clientes no se encuentran en una búsqueda por placer ni para deleitar sus sentidos en un momento de ocio. Tienen una necesidad puntual: comunicar las ventajas de un nuevo producto, mejorar la usabilidad de su portal o su estética, agregar funciones novedosas, y muchas más. La búsqueda de quien satisfaga esa contingencia genera, como es natural, un estado de ansiedad, una aceleración regulada. La organización de la información que contiene nuestro portfolio deberá



Sitios como www.illustrationmundo.com, www.ilovedesign.com y www.deviantart.com nos permiten crear un portfolio online dentro de una comunidad.

basarse en este hecho, proveyendo accesos intuitivos, rápidos y que, en pocos pasos, permitan llegar a la información que busca el visitante. Antes de incorporar material a nuestro portfolio, entonces, debemos preguntarnos si es algo realmente necesario y cuál es su nivel de importancia dentro del sitio. Así, iremos filtrando datos hasta conformar el sitio final. Por ejemplo, podemos preguntarnos si es más importante que el visitante observe en primer lugar nuestro trabajo más destacado y una breve reseña de él, o si será más útil que lea una descripción textual de nuestra filosofía de trabajo.

Al mismo tiempo que vamos decidiendo lo que queremos incluir, tenemos la tarea de categorizar el material. La división clásica recae, básicamente, en cuatro rubros: quién soy, qué hago, qué hice y dónde estoy, aunque existen tantos como profesionales diferentes pueda haber. Dentro de cada categoría es posible pensar en subdivisiones. En el caso de una sección que muestre las producciones anteriores, por ejemplo, podemos separarlas por el rubro del cliente o su nombre, por fecha, por el tamaño de la empresa o por el tipo de pieza (portal de e-commerce, periodístico, corporativo, blog, learning management system, portfolio personal, sitio promocional, etcétera).

NUESTRO PORTFOLIO DEBE TENER ACCESOS INTUITIVOS Y QUE, EN POCOS PASOS, PERMITAN LLEGAR A LA INFORMACIÓN DESEADA.

LA TECNOLOGÍA

Con frecuencia, la adopción de las últimas tecnologías se transforma en una barrera para la entrada de interesados, porque suelen requerirse accesorios de vanguardia, modernos procesadores y conexiones de varios megabits, cosas alejadas de la realidad de personas no vinculadas directamente a la informática. A menos que se haya definido como estrategia propia llegar a los estudios de diseño y agencias de comunicación de vanguardia para trabajar como proveedores, será preciso planificar el uso de herramientas tecnológicas con las siguientes características:

- ▶ Masivamente accesibles para el público promedio.
- ▶ Económicas en su costo de desarrollo, de implementación y de mantenimiento.
- ▶ Fácilmente actualizables. En pocos clics, el diseñador debe poder agregar, quitar o modificar toda la información del portfolio.
- ▶ Que no resignen atractivo visual en favor de la funcionalidad.
- ▶ Que permitan una óptima indexación y posicionamiento en los buscadores.
- ▶ Que brinden la posibilidad de vincularse a las redes sociales.

EL TRÁFICO

El paso siguiente consiste, lógicamente, en lograr que las personas ingresen en el portfolio. Para hacerlo, es necesario implementar alguna estrategia de comunicación online. Las redes sociales y los portales deberán colaborar en la generación de visitas. El agregado de la URL a la tarjeta personal impresa, a las carpetas institucionales, las facturas y los comprobantes de pago resulta de gran ayuda, así como también su inclusión en el discurso oral. Luego, tan importante como la generación de accesos es la tarea de medir el tipo y la cantidad de visitas. Google ofrece, de forma gratuita, una invaluable herramienta llamada **Google Analytics**.

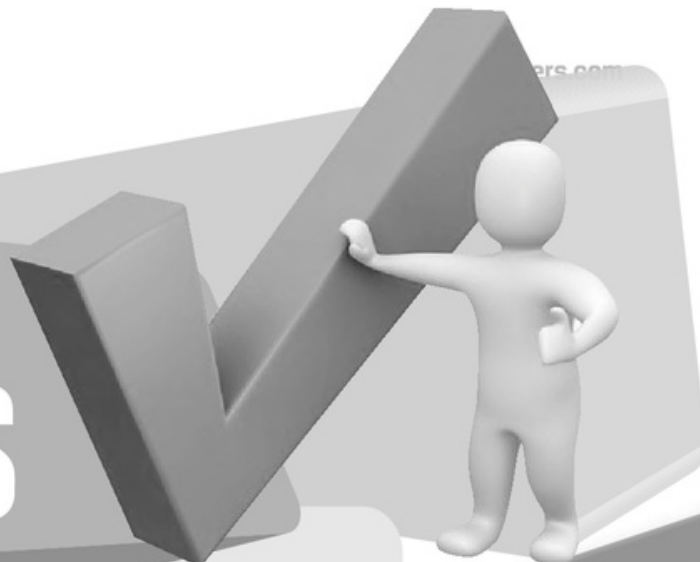


REUTILIZAR

Si no deseamos desarrollar un sitio de calidad, podemos recurrir a plataformas gratuitas Open Source, como **WordPress** (www.wordpress.com), **Joomla!** (www.joomla.org) o **Drupal** (www.drupal.org). Para operarlas, solo es necesario conocer lenguajes básicos de programación, como HTML, CSS y, quizás, un poco de PHP. Es posible comenzar a utilizarlos mediante plantillas gratuitas y fácilmente instalables, y también hay una gran cantidad de módulos y extensiones (muchas de ellas, sin costo) que permiten desde la incorporación de galerías multimedia hasta el agregado de completos formularios de contacto y envíos de comunicaciones vía e-mail.

En ella, es posible verificar la cantidad de visitas (por día, por semana y por mes), los lugares de procedencia, los sitios en los que estuvieron antes de llegar al nuestro (muy útil para estimar la efectividad de la estrategia) y las páginas visitadas, entre muchos otros datos. ◀

OPORTUNIDAD EN LAS IDEAS



Antes de lanzarnos a un nuevo emprendimiento, es importante tener en cuenta una serie de factores para evaluar si será beneficioso llevar a cabo el proyecto.

Si tenemos la inquietud de crear una empresa o un estudio propio, saltar directamente a alquilar una oficina, comprar el mobiliario y sentarse delante de la computadora puede resultar un movimiento por demás imprudente. Por definición, un emprendedor es quien tiene una idea (no siempre original o revolucionaria) que apunta a satisfacer una necesidad del mercado poco atendida o saciarla de una mejor manera. Si en vez de partir desde esa idea nos dejamos llevar por la inercia del trabajo independiente y lo que ya se ha aprendido a hacer, podemos encontrarnos sin un rumbo claro, ofreciendo un servicio que no es realmente valioso y enfrentándonos con demasiada competencia. La idea es una construcción mental significativa en general, espontánea. Está hecha de las observaciones, las

experiencias, los recuerdos y las habilidades de una persona. Muestra, sin precisiones, una situación ideal (óptima, mejor) a la que podría llegarse desde una situación existente y observable, que es la que dispara la idea.

EL ORIGEN DE LAS IDEAS

Además de provenir del interior de cada persona, distintas situaciones del contexto pueden conducir a una buena idea:

- ▶ **Las noticias y las publicaciones especializadas:** la simple lectura de un periódico el día domingo, el aprendizaje a partir de libros como el presente, los portales web y los blogs, todos ellos reflejan acontecimientos de última novedad o tecnologías que pueden derivar en nuevas ideas para un emprendimiento.
- ▶ **Congresos y charlas académicas:** son importantes tanto por el conteni-

do del discurso de los disertantes –en general, profesionales destacados y de vanguardia–, como por la posibilidad de generar intercambios con colegas y asistentes.

▶ **Tendencias sociales:** los cambios en el comportamiento de un rango de personas que forman parte de la sociedad indican tendencias; por ejemplo, la adopción masiva de teléfonos celulares para interactuar con las redes sociales. En ocasiones, estos cambios ocurren tan rápido, que las grandes empresas no tienen tiempo de adaptarse, y los nuevos emprendedores podrán aprovechar la oportunidad.

▶ **La calle y el entorno próximo:** podemos obtener una idea de algo tan simple como observar a un amigo con dificultades para conseguir un pintor de confianza, un electricista o un camión de mudanzas. Esto puede llevarnos a



idear un portal web que permita buscar profesionales por rubro y ubicación geográfica y que cuente con un sistema de recomendación de usuarios.

► **Los viajes y el entorno lejano:** existen emprendimientos exitosos que se basan en la adaptación de ideas efectivas surgidas en el extranjero, pero ausentes en sus lugares de origen.

VALORAR LAS IDEAS

Miles de ideas pasan por la mente de una persona en un solo día. La mayoría de ellas reciben atención durante pocos segundos, y luego quedan en el olvido definitivamente. Algunas darán vueltas en nuestra cabeza por un día y, quizás, una encuentre su lugar en un trozo de papel. La pregunta es, entonces, ¿cuál es una buena idea de empresa? O mejor aún, ¿cuál de todas nuestras ideas es una verdadera oportunidad, tan interesante, que amerita la creación de una empresa para explotarla? Proponemos valorar las ideas que persistan durante varios días a través de un filtro: la respuesta sincera y reflexiva a las siguientes preguntas:

► **Sobre uno mismo:** ¿realmente disfrutaría haciendo esto? La respuesta debe incluir el deseo de seguir adelante con la idea incluso en situaciones adversas, soledad, cansancio y críticas. ¿Creo que puedo hacerlo? ¿Tengo los conocimientos, habilidades y herramientas suficientes?

► **El mercado, el público:** ¿quiénes estarían interesados en este producto o servicio? ¿Por qué las personas elegirían este y no otro? ¿Qué elemento distintivo e interesante les ofreceré para que lo perciban como valioso y deseen pagar por él? A diferencia de las preguntas del punto anterior, es imposible obtener estas respuestas a partir de la pura cavilación. Es recomendable buscar información en libros, estudios de mercado, encuestas, medios periodísticos o, incluso, preguntando a aquellas personas que podrían tener el perfil de clientes que se aspira obtener.

► **La competencia:** ¿quiénes están realizando una oferta similar o ya han puesto en marcha una idea parecida? ¿Dónde están, cuáles son sus servicios, cuántos son, quiénes son sus clientes? Es útil recordar que, ante una mayor cantidad de oferta que el público perciba como similar, el precio bajará y será la



CONOCIMIENTOS TÉCNICOS

Recordemos que, para mejorar y potenciar nuestro aprendizaje, desde www.redusers.com/cursos/webdesign podemos descargar los archivos editables de los proyectos que realizamos en esta obra.

A partir de ellos podremos obtener ideas para un proyecto propio o, directamente, modificar los archivos para reutilizarlos. En el sitio también encontraremos **guías visuales**, **glosario de términos** y una completa guía de preguntas frecuentes sobre la suite CS5. Además, el libro **Desarrollo y animación en ActionScript 3.0** nos ayudará a comprender los fundamentos de la programación en Flash.

ANTE MUCHA COMPETENCIA EN UN MERCADO QUE NO CRECE, EL VALOR DEL SERVICIO BAJA Y APARECE LA GUERRA DE PRECIOS.

principal variable para lograr su preferencia. Lo ideal, si no disponemos de un gran capital de inversión, es ingresar en mercados en donde no haya muchos competidores o donde estos no sean fuertes, reconocidos, con gran experiencia y trayectoria. En general, estos mercados son aquellos en los que más especialización e innovación se requiere.

► **Monetizar:** muchas ideas son atractivas hasta el momento en que se piensa en cómo se obtendrán ganancias a partir de ellas. Volviendo al ejemplo del portal de búsqueda de profesionales, ¿cómo podríamos obtener dinero a partir de

él? Podríamos pensar en cobrar una mensualidad a cada profesional que allí se promocionara, en percibir ingresos en concepto de publicidad de otras empresas o, quizás, cobrar un porcentaje de cada contratación generada por el sitio, entre otras posibilidades. ◀



DETERMINAR EL PRECIO DEL TRABAJO

¿Cuánto debería cotizar este trabajo? ¿Qué práctica puedo seguir para establecer mis precios? Estas son preguntas frecuentes que se plantean los profesionales cuando dan sus primeros pasos en el mundo laboral.

En esta sección explicaremos una de las tantas propuestas y técnicas para evaluar el precio de un trabajo, dejando el camino libre para que el lector la modifique, critique y adapte a su realidad. Esencialmente, esta estrategia fija un precio según las necesidades y los objetivos económicos del profesional, calculado la cantidad de horas que insu-me un proyecto determinado. Se deja de lado y se evitan los precios "según la cara del cliente" y el irreal "precio de mercado" del diseño (¿cuánto se cobra el diseño de determinado producto?).

LAS HORAS

Como dijimos, el precio del servicio de diseño depende de quien lo realiza, no de quien lo paga, de lo que cobra la competencia o de la opinión de terceros. El primer paso consiste en hacer una detenida y sincera autoevaluación de la cantidad de dinero necesario por mes para:

- ▶ Cubrir las necesidades básicas: vivienda, comida, vestimenta, salud, sociales, etc.
- ▶ Mantener el funcionamiento de la oficina, estudio o profesión.
- ▶ Ahorrar ingresos, según los objetivos del profesional.
- ▶ Realizar capacitaciones y obtener material de formación.
- ▶ Invertir en la profesión o estudio.
- ▶ Pagar sueldos y hacer frente a otros costos fijos.

Sumando las respuestas a cada punto, se llegará a una suma de dinero mínima que se deberá reunir cada mes para alcanzar un equilibrio, sin ganar ni perder. Si no se logra reunirla, el negocio generará pérdidas. Los índices de variación de precios en cada país, junto con la inflación, hacen que este número deba ser replanteado y revisado con frecuencia.

EL PRECIO DE LA HORA

En paralelo, habrá que definir cuestiones relativas a las condiciones de la jornada de trabajo diaria:

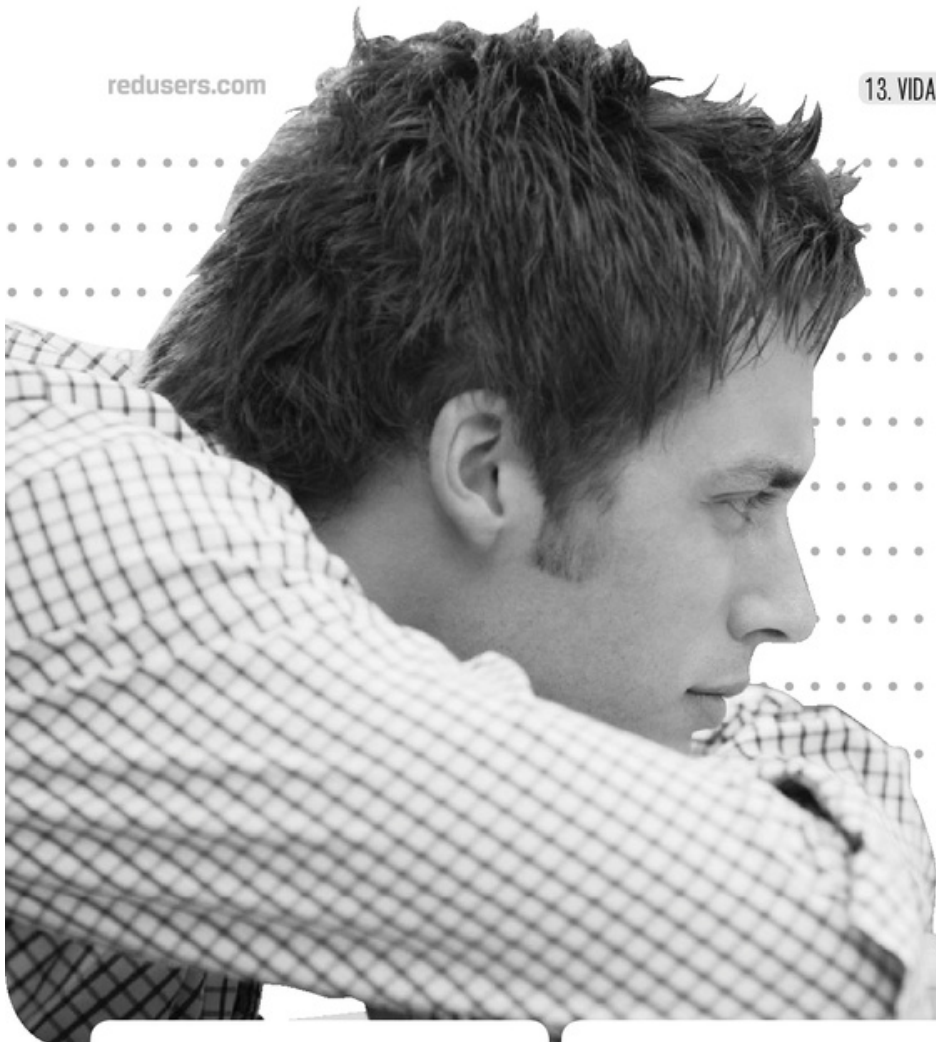
- ▶ Cantidad de meses de trabajo al año.
- ▶ Cantidad de semanas de vacaciones por año.
- ▶ Cantidad de días y horas de trabajo por semana.



PERFECCIONAMIENTO CONTINUO

La realización de un presupuesto justo será el producto de la práctica, y para esto no existen procedimientos instantáneos. Lo explicado hasta ahora es solo una propuesta de guía para someter a un frío análisis. Deja de lado gastos excepcionales y otras variables como meses de previsible baja demanda en los servicios. No tenemos una respuesta cierta a la pregunta "¿cómo puedo hacer para saber exactamente cuántas horas me llevará realizar la pieza de diseño encargada?". Solo la práctica, un detenido análisis de las capacidades del profesional, los plazos acordados con el cliente y la situación personal de cada freelance podrán comenzar a contestarla.





► Cantidad de horas de trabajo por día. Las horas que se afectan al desarrollo de un proyecto también incluyen las de reuniones y presentaciones con clientes.

ESTA ESTRATEGIA FIJA UN PRECIO SEGÚN LAS NECESIDADES Y LOS OBJETIVOS ECONÓMICOS DEL PROFESIONAL.

Veamos un ejemplo corriente, en el que delimitamos una jornada de trabajo de 8 horas diarias, 5 días por semana, de lunes a viernes, desde las 9 hasta las 17. Por un lado, tenemos entonces la cantidad de horas empleadas en trabajo por día, y por el otro, la suma de dinero que se debe recaudar para llegar a un punto de equilibrio. Es posible entonces establecer la siguiente relación:

- Dinero necesario por mes para el punto de equilibrio: US\$ 1500.-
- Horas laborables por día: 8
- Horas laborables por semana: 40, que resulta de multiplicar la cantidad de horas por la de días laborables por semana.
- Horas laborables por mes: 160 (multiplicando las horas por semana, por las semanas de un mes). Entonces, $US\$ 1500 / 160 \text{ hs.} = US\$ 9,375$.

Se arriba así a un rudimentario punto de equilibrio de US \$9,375, como valor de hora de trabajo profesional.

ASPECTOS PARA TENER EN CUENTA

Sin embargo, este cálculo de ejemplo presume que, de 9 a 17 horas, se trabajará constantemente y sin parar, hasta el último segundo hábil. Esta presunción deja afuera necesidades básicas como la comida, de información (para leer las últimas novedades locales o profesionales) y sociales (chatear brevemente con un amigo o participar en redes sociales). No es posible ni sano eliminarlas, por lo que podríamos adjudicarle una hora para almuerzo y otra para información y sociales. Con esto, la cantidad de horas que efectivamente se trabaja quedaría reducida a 6 y no a 8, como figura en el postulado anterior. Es decir:

- Dinero necesario por mes para el punto de equilibrio: US\$ 1.500
- Horas laborables por día: 6, en vez de 8
- Horas laborables por semana: 30, en lugar de 40
- Horas laborables por mes: 120, en lugar de 160 (40 horas menos de carga).
- Entonces, $US\$ 1500 / 120 \text{ hs.} = US\$ 12,5$ en vez de US\$ 9,375

RECOMENDACIONES GENERALES

Una vez alcanzado el valor inicial de la hora de trabajo, y al momento de poner-

lo en práctica, es interesante reflexionar sobre los siguientes aspectos:

- Todo se cobra. Si el cliente añade más trabajo o modifica de alguna forma el tipo de servicio que recibirá, aunque el presupuesto haya sido realizado y aprobado, el profesional debe replantearlo.
- Es tentador, pero no recomendable, cobrar más o menos de acuerdo con el cliente que se tenga en frente. Sin embargo, es bueno tener cierta flexibilidad que permita la negociación, sin que resulte en pérdida para el profesional.
- No es aconsejable tasar un proyecto guiados por lo que dijo otro profesional que cobró por algo similar. En general, poco se sabe de sus detalles, de sus necesidades profesionales y personales, y de la negociación llevada a cabo. ◀



MODELO

DE NEGOCIO

Lejos de ser un proceso que se planifica y ejecuta una sola vez en el tiempo, el modelado de un nuevo emprendimiento es una actividad permanente y continua. Veamos la primera parte de esta idea.

Si una empresa no innova, analizando y adelantándose a las tendencias, a las nuevas necesidades y a las preferencias del público, está condenada a estancarse, y a perder todo el terreno que ha sabido conseguir. Pensemos en aquellos cantantes pop atascados durante décadas en el estilo de canciones que, en una época, los llevaron al éxito, y que hoy solo son escuchados por un reducido grupo de memoriosos. Sin embargo, existen otros que, a fuerza

RECOMENDACIONES
FINALES

Recomendamos el mayor detalle posible y evitar respuestas con enunciados demasiado generales. No es posible abarcar todo el mercado, y no a todo el mundo le interesará la propuesta de nuestros servicios. Invertiremos mucho mejor la energía y el tiempo si enfocamos nuestros esfuerzos en contactar a organizaciones o personas que ya tengan la necesidad de contar con ese servicio, la voluntad de pagar por él y el dinero suficiente para hacerlo.

de actualización constante, adopción de nuevas tendencias musicales y modernización de su estilo, nunca perdieron su presencia en los primeros puestos de los rankings.

La innovación y la búsqueda progresiva de mejoras deben implementarse en todos los aspectos de la empresa, como los siguientes:

- ▶ La relación con los socios, proveedores y empleados.
- ▶ La infraestructura, organización física de la oficina, decoración.
- ▶ La gestión de los clientes y de ventas.
- ▶ La implementación y actualización de tecnología.
- ▶ El proceso de prestación del servicio, el de producción.
- ▶ El manejo de la competencia.
- ▶ La contabilidad y las cuestiones legales empresariales.

La realidad, la vanguardia, las noticias y las investigaciones del contexto proveerán las alertas de los cambios que se avecinan. Ante ellos, el emprendedor adoptará una postura flexible pero sólida, y liberará a su empresa de estructuras demasiado burocráticas que impidan los cambios rápidos.

COMENZANDO A **PLANIFICAR**

Luego de haber ponderado una idea, de acuerdo con los puntos analizados en otras secciones de este apéndice, se comenzará con una introspección profunda y detenida, en torno a los siguientes aspectos:

- ▶ Cuál es el núcleo del servicio.
- ▶ A quiénes se lo ofrecerá.
- ▶ Cómo y cuándo ofrecerlo.
- ▶ Cómo organizarse para brindarlo.
- ▶ Cómo se generarán ganancias económicas a largo plazo.
- ▶ Cómo diferenciarse.
- ▶ Cómo ser mejor cada día.

Es una buena idea evitar la reacción inicial de responder rápida y mentalmente a estos interrogantes, y la creencia de que, desde un principio, se conocen todas las respuestas. Algunas personas omiten esto, y saltan directamente a la apertura de una oficina, esperando llamados de clientes. Enfatizamos el hecho de que una oficina con una persona sentada detrás de una computadora, sin más planificación, difícilmente pueda llamarse "empresa".

EL NÚCLEO DEL SERVICIO

Un diseñador vende su tiempo, talento y recursos para agregar valor, potenciar las habilidades y solucionar problemas de índole comunicacional a un cliente. Este



concepto, hoy en día, es un estándar, dada la gran cantidad de instituciones educativas de diseño y la aparición de publicaciones formativas de gran calidad, como la que tienen en sus manos.

Pero el núcleo del servicio, que dará sentido a la formación de la empresa, partirá de la original idea motora, producto de la creatividad del emprendedor. Por supuesto, sus conocimientos y habilidades resultarán decisivos para llevarla adelante. Específica y detalladamente, se planificará:

- ¿Qué problemas o necesidades puntuales de la sociedad satisfará la empresa que llevemos a cabo?
- ¿A cuáles no dará respuesta (dentro del área), por estar fuera del foco del negocio que realizamos?
- ¿Este servicio será distinto de los existentes u original? ¿Por qué?

A QUIÉNES SE LO OFRECERÁ

Como paso siguiente, se planificará en forma secuencial:

- **Mercado potencial.** La porción general de la sociedad, con necesidades que podrían beneficiarse del servicio que se quiere prestar.
- **Tipo de sector.** Tipo de industria, profesionales, oficios, pequeñas y medianas empresas, grandes empresas, gobiernos, organizaciones no gubernamentales.
- **Quiénes componen este sector.** Listado, lo más exhaustivo posible, de los individuos, empresas y entidades que más se aproximan al ideal de futuros clientes del emprendimiento.
- **Ubicación geográfica.** Categorización del contenido del listado por ubicación geográfica. Selección de las ubicaciones más próximas o deseadas, según las posibilidades del emprendedor.
- **Mercado objetivo.** Dentro de esta lista, y evaluando los recursos reales del equipo emprendedor, se seleccionará a



quienes se desea contactar en un primer momento, con el propósito de que sean los primeros clientes.

► A quiénes se está en condiciones de contactar. Cómo y por qué.

- De ellos, ¿quiénes ya poseen un servicio similar al que se propondrá? ¿Cómo se lo podría mejorar?
- ¿Quiénes tienen la necesidad, pero no, ningún satisfactor hasta el momento?
- ¿A cuántos de ellos se podría atender con eficacia, sin perder calidad ni exceder nuestras capacidades?

► **Mercado efectivo.** Deben contrastarse estos datos con la realidad más próxima, y establecer metas para conseguir este mercado:

- Cantidad efectiva de contactos y visitas por realizar en el próximo mes y, luego, por mes en el año. ¿Con quiénes se empezará y cuándo?
- Cantidad de clientes que se necesita adquirir por mes y por año, para que el emprendimiento sea redituable.
- Tipo de proyectos que se comenzarán a ofrecer y proponer. ◀



MODELO DE NEGOCIO FINAL

Trataremos los aspectos más importantes para tener en cuenta antes de realizar el salto desde el lugar del profesional freelance al de dueños de un estudio o empresa propia. Veamos cómo lograrlo.

En la sección anterior, desarrollamos los primeros pasos de este ejercicio tan importante como es instalar en el mercado un estudio propio. Veamos ahora la segunda y última parte de esta empresa.

CÓMO Y CUÁNDO OFRECERLO

Una vez definida la cantidad de posibles clientes por contactar, saber quiénes son, y detectar su ubicación o área de influencia, estamos en condiciones de pasar a reflexionar acerca de cómo contactarlos y el momento propicio para hacerlo. Por ejemplo, imaginemos que escogimos el rubro de empresas relacionadas con las inversiones en bienes raíces (ya que resultó ser un servicio que los clientes encontrarían interesante y valioso). Ya las individualizamos y sabemos que se sitúan en cinco barrios cercanos a la zona céntrica. Comenzaremos, entonces, con la estrategia de acercamiento. Investigaremos, en primer lugar, con quién debemos hablar; es decir, la persona que tiene poder de decisión sobre la aprobación del proyecto. El objetivo es conseguir su atención durante un breve lapso, para



RELACIONES INTERPERSONALES

Es necesario tener presente que todos los consejos que hemos dado en este apartado son imprescindibles para el desarrollo de nuestra unidad de negocio. Sin embargo, es importante destacar que el universo mercantil es demasiado amplio y competitivo como para abordarlo solamente desde estos conceptos. Es por eso que necesitamos aportar una cuota extra de creatividad, no en el sentido de aplicación sobre nuestros desarrollos web, sino en términos de relaciones empresariales. Es decir, las relaciones interpersonales que se establecen antes, durante y después de haber cerrado un trato dan como resultado un gran valor agregado. Si nosotros no mantenemos una estrecha relación profesional con el cliente, aun luego de haber terminado nuestro trabajo, estaremos cometiendo un grave error.

que conozca el servicio. Una manera de lograrlo, es presentarle un proyecto con los beneficios y ventajas que tendrá para la empresa en un lugar muy destacado, y asegurarnos de que represente algún tipo de valor y ventaja también para él.

CÓMO ORGANIZARSE PARA BRINDARLO

El hecho de no detenerse a planificar el proceso que seguirá al estudio para realizar el servicio suele conducir a desagradables sorpresas una vez comenzado el emprendimiento. La suposición peligrosa de que se conocen todas las respuestas, por haber recibido educación superior o experiencias anteriores, hace que algunos emprendedores improvisen demasiado sobre la marcha, y sus primeros clientes sean

sus cobayos experimentales. En este punto, determinaremos las distintas áreas que harán al funcionamiento del estudio y planificaremos, a grandes rasgos, su manera óptima de funcionamiento. Transcribiremos a continuación las principales áreas básicas de una empresa, sin ánimo de ser exhaustivos:

- ▶ **Comercialización:** acercar el servicio a un público interesado en él, por medio de un precio, un lugar, la comunicación y el diseño de un servicio valioso.
- ▶ **Ventas:** acciones tendientes a convertir a personas interesadas, en clientes reales.
- ▶ **Cuentas:** asesoramiento e intercambio continuos con el cliente, hasta finalizar la prestación del servicio.
- ▶ **Gestión de proveedores:** administración de materia prima.
- ▶ **Producción:** se crea y se da forma a la solución para la necesidad del cliente.
- ▶ **Finanzas /contabilidad:** administración del dinero del estudio.
- ▶ **Legales:** área que ajusta la actividad de la empresa al marco legal vigente.
- ▶ **Cobranzas:** área administrativa comercial de la empresa.
- ▶ **Recursos Humanos:** área administrativa del personal.
- ▶ **Dirección del estudio:** ubicación geográfica (física).



CÓMO SE GENERAN GANANCIAS ECONÓMICAS

Para continuar construyendo el emprendimiento, habrá que razonar sobre las vías de ingreso de dinero al estudio; es decir, qué se dará a los clientes a cambio de su pago. La más típica resulta del cobro en concepto de honorarios por el servicio prestado. No obstante, es útil plantear si esta será la única vía y, si así fuera, cómo lograr un flujo más o menos constante, para poder pensar en la contratación de empleados y evitar el endeudamiento. Por ejemplo, algunos estudios, además de su servicio central, obtienen ingresos económicos de:

- Comercialización de objetos de diseño, vestimenta, accesorios de vanguardia y publicaciones dirigidas a diseñadores.
- Organización de eventos, también dirigidos a diseñadores y afines, como congresos, talleres, fiestas y muestras.
- Mercancía de producción propia: t-shirts estampadas, revistas impresas, portales web, etc.

CÓMO DIFERENCIARSE

El mercado objetivo debe encontrar valor, ventajas y beneficios en contar con el servicio propuesto, en vez de ir a la competencia, reemplazarlo por otra categoría de servicios o, directamente, no hacer nada. Este valor provendrá del nivel de satisfacción que encuentre en aquel y, sobre todo, de la singularidad de la propuesta. El cliente debe percibir

LA SUPOSICIÓN DE QUE SE CONOCEN TODAS LAS RESPUESTAS HACE QUE ALGUNOS EMPRENDEDORES IMPROVISEN SOBRE LA MARCHA.

claramente que hay pocos (o ninguno) que podrán ayudarlo de esa forma. Para eso, es posible comenzar por establecer diferencias con todos los servicios similares, que sean, obviamente, valiosas para el cliente, y comunicárselas. Las propuestas de valor que logran diferenciación pocas veces son técnicas, y siempre son fácilmente recordables y entendibles. "Calidad, servicio y limpieza" utilizó Mc Donald's para diferenciarse de su competencia y simplificar la inmensa complejidad de la empresa multinacional, a los ojos del cliente.

Es imprescindible recordar que el valor del servicio no solo se encuentra en él mismo, sino que también es construido por los clientes, el mercado objetivo y el público en general.

CÓMO SER MEJOR CADA DÍA

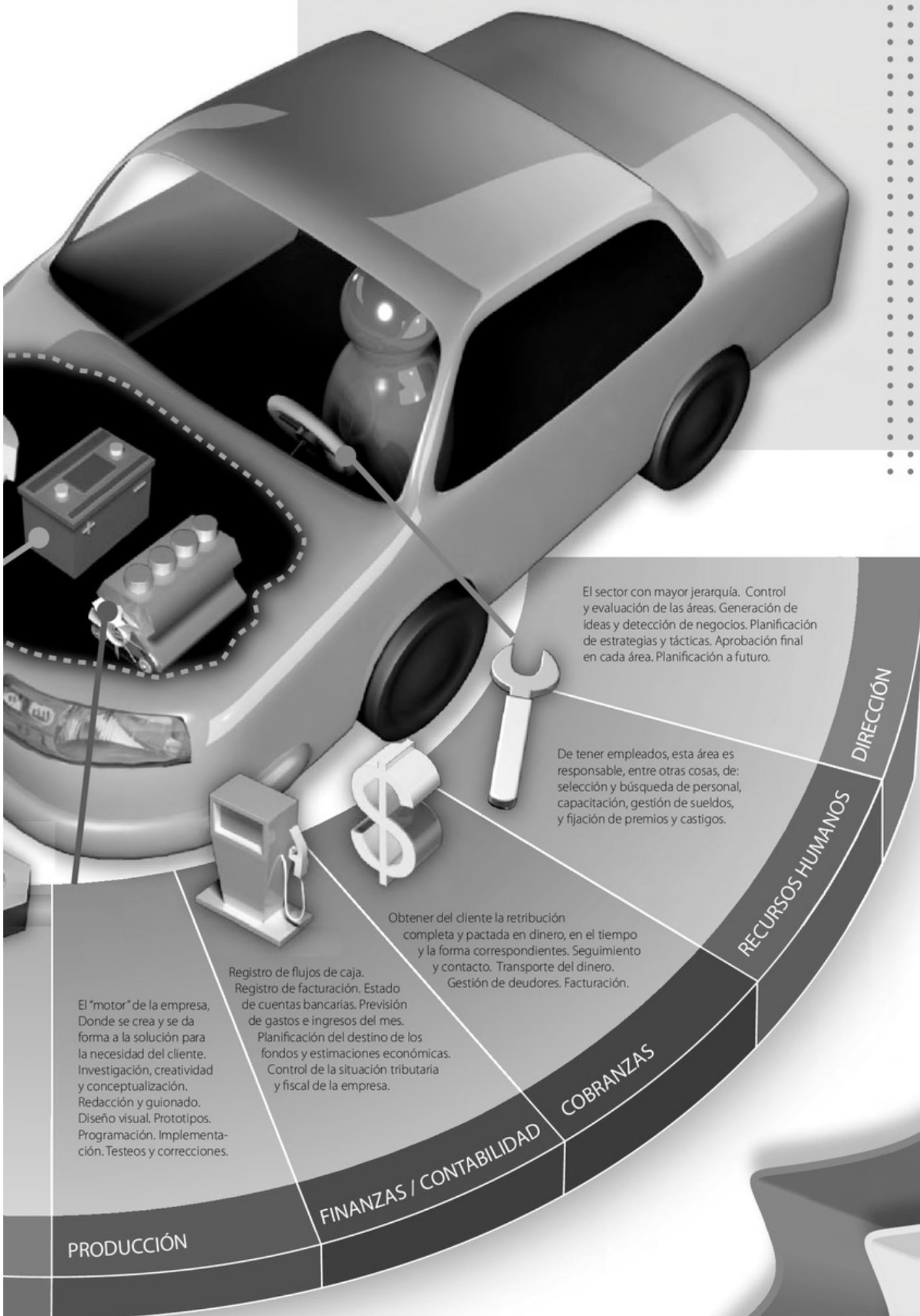
Yendo más allá de la diferencia y el valor del servicio para el público, los tiempos modernos revelan la insuficiencia de tener un servicio valioso para el mercado en un momento dado: es preciso hacerlo cada vez mejor, para asegurar la permanencia en los primeros puestos del ranking de preferencia. La evaluación continua del entorno, la competencia, la propia empresa y la capacitación constante revelarán aspectos y tendencias que, de ser descuidados, dejarán la puerta abierta para que otros jugadores tengan la oportunidad de hacerse fuertes en el mercado objetivo elegido. ◀



FUNCIONAMIENTO DE UNA EMPRESA

Veamos en esta infografía la estructura empresarial modelo. Además conozcamos sus sectores componentes.





PLANIFICAR PRESENTACIONES

EFFECTIVAS

En este apartado veremos los mejores consejos para realizar las presentaciones de nuestros trabajos y poder captar el interés de futuros clientes. La idea es ir de los objetivos generales a los particulares.

En esta ocasión, veremos la clase de presentaciones que los diseñadores deben realizar a la hora de dar a conocer un prototipo o primer modelo de la solución web requerida por el cliente. Nuestras presentaciones difieren de aquellas en las que se presenta el estudio al público con el objetivo de que conozcan sus servicios y se los tenga en cuenta en futuros pedidos de presupuesto. Asimismo, son distintas de las que intentan vender una propuesta concreta o un nuevo servicio.

LOS OBJETIVOS GENERALES

En general, las presentaciones de prototipos, ideas y modelos persiguen los siguientes objetivos:

- ▶ Que el cliente comprenda cabalmente la propuesta y se entusiasme con ella, evitando malos entendidos a futuro.
- ▶ Que el cliente acepte la propuesta en la forma planteada por el profesional.
- ▶ Que el cliente no descarte la solución a menos que encuentre un claro error por parte de sus productores.
- ▶ Pasar sin demoras a las próximas etapas del servicio, con miras a su finalización y al cobro de los honorarios completos.

En esencia, se buscará persuadir al cliente para que tome determinadas decisiones, más que solo informar sobre el avance del proyecto. Muchas veces, una propuesta es descartada (y hay que recomenzar el trabajo sin percibir pago alguno) solo porque no se la explicó con suficiente claridad, no se resaltaron sus beneficios y no se la comunicó con plena convicción.

LOS OBJETIVOS ESPECÍFICOS

Se trata de lo que el presentador desea que suceda a raíz del éxito de la futura

presentación. Estos objetivos serán elaborados de acuerdo con la situación y las necesidades propias de cada caso, ya que se transformarán en las guías en torno a las cuales se confeccionará la estrategia de la presentación. Es verdad que, con frecuencia, se asiste a una presentación sin los objetivos propios ni la estrategia, con la seguridad de que el modelo hablará casi "por sí solo" y el cliente quedará maravillado al estar expuesto a él. Lamentablemente, esa es una situación extraordinaria. Veamos un ejemplo de objetivo

SE BUSCARÁ **PERSUADIR** AL **CLIENTE** PARA QUE TOMÉ DETERMINADAS **DECISIONES**, MÁS QUE SOLO INFORMAR SOBRE EL AVANCE DEL **PROYECTO**.



específico: "Luego de la presentación, y en el mismo día, el cliente debe validar el modelo firmando una conformidad". O quizá, "que el cliente acepte comenzar con la siguiente etapa".

► **Quién y dónde:** ¿quiénes estarán presentes? Un aspecto clave para modelar las tácticas de presentación consiste en averiguar el perfil de los asistentes y el lugar en que se hará el encuentro. Si no se cuenta con estos datos, habrá que preguntar al cliente.

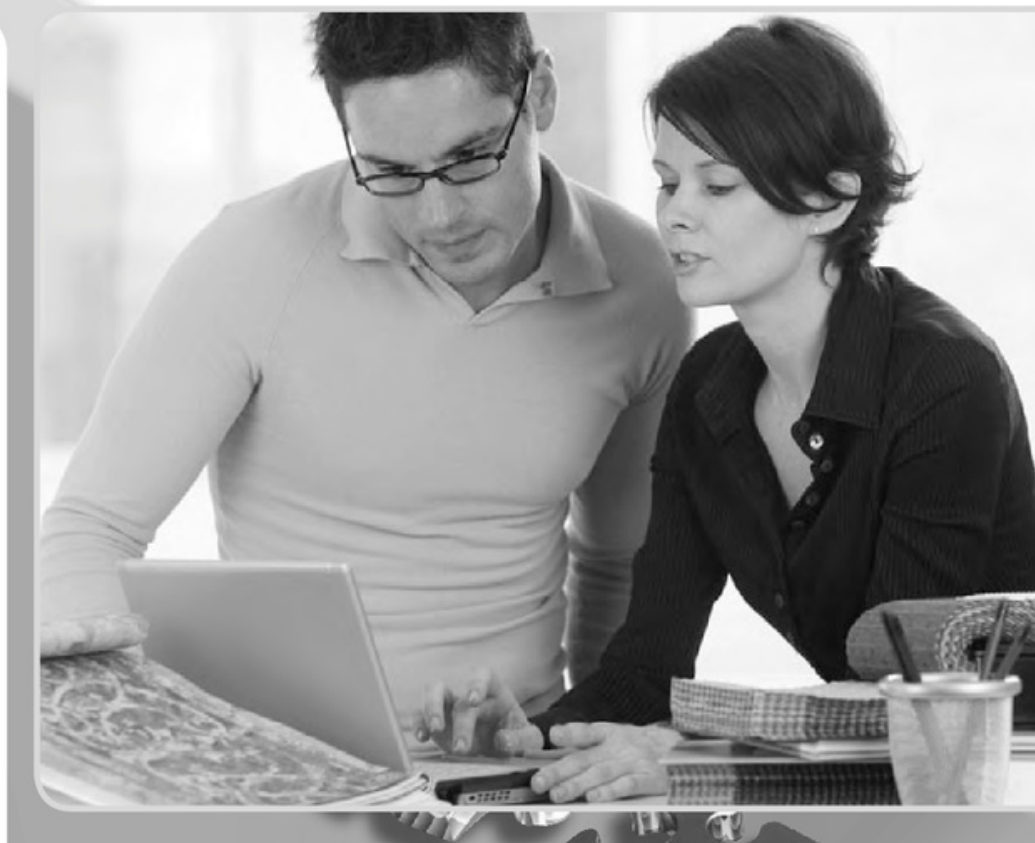
► **El número de personas del lado del cliente:** quién es el encargado de tomar las decisiones y saber si estará presente.

► **El tipo de relación previa con los asistentes:** amistad, trabajos anteriores, desconocidos y colegas.

► **Costumbres y cultura del cliente:** ¿cuál es el lenguaje y el contexto del cliente? Por ejemplo, si se trata de una empresa joven, con cultura informal, dedicada a la producción de vestimenta vanguardista, es posible que no valoren una presentación del tipo "hombre serio, con traje y PowerPoint".

► **Nivel de conocimientos que se tienen sobre la materia:** esto es importante para seleccionar el lenguaje que se usará y la profundidad en el detalle técnico.

► **Soporte de presentación disponible:** proyector multimedia (qué resolución tiene, posibilidad de conexión a una notebook), computadora (¿propia o del cliente?), impresos.



► **Tamaño y posibilidades que brinda el lugar:** por ejemplo, si la oficina del cliente no posee WiFi, se buscará la manera de utilizar el soporte, quizá, sin conexión a Internet.

► **Tiempo disponible:** si el cliente no fija una duración, el profesional deberá establecerla. Así, evitará desviarse del tema, enredarse en cuestiones superfluas y perder gran parte de su jornada laboral.

TÁCTICAS DE PRESENTACIÓN

A continuación, una serie de tácticas para ayudar a conseguir los objetivos específicos del estudio. La estrategia concreta reunirá algunas, descartará otras e incluirá las propias, en consonancia con los citados objetivos, el "quién" y el "dónde", y variables específicas del tipo de proyecto y estudio.

- Jamás mostrar el resultado visual en primer lugar y preguntar si le "gusta", sin haber hecho una introducción. El cliente tenderá a evaluar la pieza como si fuese una obra de arte, sin ningún parámetro más que su gusto personal.
- Es útil hablar sobre los objetivos iniciales del proyecto al comenzar la presentación, ya que estos habrán sido aceptados por el cliente en su oportunidad, y a partir de ellos se hizo el trabajo.
- Resaltar los beneficios y las ventajas que implica la adopción de la solución propuesta.



- A menos que se trate de colegas o personas con altos conocimientos de Internet, utilizar poco lenguaje técnico o jergas internas. Si la persona desconoce su significado, su atención se desviará y perderá el hilo de la charla.
- Es importante relacionar los puntos de la solución con los objetivos del proyecto, de forma que se la presente como el resultado justo para el requerimiento.
- No debemos detenernos demasiado tiempo en una objeción ni en detalles del proyecto. En muchas ocasiones, la presentación fracasa porque un fragmento del proyecto fue objetado y el presentador gastó todas sus energías en justificarlo, a la vez que olvidó que una gran parte del proyecto no obtuvo ninguna negativa. ◀



EL DISCURSO Y LOS GESTOS

En estas páginas hemos detallado los consejos más importantes que debemos tener en cuenta para realizar las presentaciones de nuestros proyectos. La clave, como pudimos observar, es la planificación de los objetivos y la táctica. Sin embargo, hay otro aspecto importante que no podemos pasar por alto y tiene que ver con la puesta en escena de la presentación. Tanto el lenguaje corporal como el oral desempeñan un rol fundamental en el proceso de persuasión. Es por eso que debemos seleccionar cada palabra del discurso, sin olvidar las formas de lo gestual. Es preciso mantener un equilibrio entre estos dos aspectos del lenguaje, de lo contrario, nuestros clientes podrían realizar una lectura equivocada de lo que queremos comunicar.

LAS ACTITUDES NOCIVAS DEL DISEÑADOR



Como diseñadores, ciertas reacciones ante situaciones propias de la profesión alejan a quienes las poseen de su desarrollo en forma saludable y sostenida. Veamos cuáles son y cómo evitarlas.

Muchas de las actitudes nocivas que un diseñador puede presentar son el resultado de una construcción de personalidad en la cual intervienen innumerables experiencias. Algunas de ellas resultan positivas, mientras que otras no lo son tanto. Veamos algunas manifestaciones que pueden resultar negativas para nuestra profesión:

- Personalidad e historias personales: en general, la profesión se encara de la misma manera que la vida personal.
- Universidades e instituciones educativas: además de educar sobre los aspectos técnicos, las instituciones (a través de la personalidad y la experiencia de sus profesores) influyen decididamente en las actitudes futuras de los profesionales que preparan, para bien y para mal.
- Experiencia propia y contexto: todas las interacciones de la vida profesional, sobre todo las más traumáticas, se convierten en aprendizaje, en una guía enciclopédica única que reside en la mente, y que moldea la manera de manejarse en la profesión.
- Modelos de acción, tomados de colegas famosos o más experimentados: quizá sus métodos y modelos mentales funcionaron para ellos y los llevaron al éxito. Pero eso sucedió en su



LA SORDERA

Algunos diseñadores parecen irritados al momento de escuchar el punto de vista del cliente. creen que la solución al requerimiento proviene exclusivamente de una suerte de inspiración genial, dominio propio del diseñador creativo. Cuando un cliente quiere aportar su conocimiento sobre el producto y el mercado (ya que, evidentemente, nadie los conoce mejor que él), este profesional se queja porque "se está metiendo en su trabajo" o "no le da libertad para crear". Por suerte, existen otros diseñadores, que saben que una gran parte de la solución se encuentra en el mismo problema. Por lo tanto, escuchar activamente a la fuente del problema, el cliente, se revela como una de las tareas más importantes del proceso de diseño.

propio tiempo, contexto y realidad, y no garantiza en forma alguna que funcione también para quienes los han tomado como paradigma.

LA COMODIDAD

Esta actitud se pone de manifiesto cuando un diseñador no muestra interés en capacitarse y evolucionar

continuamente, luego de terminar su formación académica. Simplemente, se limita a hacer, durante un largo período, solo aquello que le es transmitido en esa



oportunidad. La universidad y los cursos tienen como objetivo el aprendizaje efectivo de la forma en que se descubren, planifican y hacen las tareas, en un determinado sector del conocimiento humano. Además de algunas conjeturas, nada certero dicen acerca de los próximos tres años, por ejemplo. Si para ese momento el egresado aún continúa aplicando los mismos y exactos métodos aprendidos, significa que se ha estancado. Esta actitud tan dañina es una de las que propicia el escenario actual de gran competencia entre diseñadores y estudios de diseño. Muchos de ellos ofrecen un servicio estándar, sin variaciones con respecto a lo que obtuvieron de sus cursos, que, por cierto, están disponibles para miles de personas.

EL MIEDO A LOS NÚMEROS

En el momento en que llega la importantísima tarea de hacer los cómputos de ingresos y egresos monetarios y confeccionar un presupuesto, sobreviene con frecuencia una actitud de rechazo y ansiedad. En suma, podríamos hablar de una reacción hacia lo desconocido, ya que se trata de materias que casi no se mencionan en muchos cursos e instituciones educativas. Enfocados completamente en el área de producción, al lanzarse como freelancers, los diseñadores tienden a ignorar los aspectos legales, contables y administrativos, y así es que

quedan a merced de clientes y proveedores que aprovechan la situación. Es necesario entender que estas facetas, lejos de hacer al diseñador "menos creativo", son indelegables e impostergables, si es que se desea contar con una práctica económicamente sólida y previsible.

LA CRÍTICA DESTRUCTIVA

Hablamos aquí de la crítica superficial, sin conocimientos profundos de la realidad de lo criticado, en tono de queja, repleta de ironías y sarcasmos, hacia todo lo producido por otros diseñadores (o diseñadores no famosos). Por lo general, estas detracciones contienen las expresiones "no me gusta", "es horrible" y "esto está peor que antes", después de observar durante unos pocos minutos una producción de diseño, sin hacer ningún análisis ni estudio serio que justifiquen tal ataque. El diseño es el resultado de un proceso: información, contexto, cliente y limitaciones, presupuesto, y miles de variantes más que entran en juego a la hora de valorarlo. Por consiguiente, es imposible (e irresponsable) hacerlo solo basados en el aspecto visual, como si se tratase de una obra artística.

Por otra parte, ese diseñador que así actúa está utilizando las mismas palabras que le producen enojo al oír las de labios de un cliente: "no me gusta". Básicamente, entonces, se convierte en aquello de lo que también se queja.

LA SOLEDAD

Se refiere al estado o la actitud de encerrarse en una torre alta y solitaria, enchufado a una computadora, durante interminables horas, sin contacto humano real, más allá de las redes sociales y los mensajeros instantáneos. El trabajo en equipo es indispensable para la resolución de tareas de gran escala y complejidad, pero incluso las de menor tamaño se benefician de él. El intercambio y el complemento con otras personas (otros profesionales e, incluso, el mismo cliente) permiten:

- El contacto con otros puntos de vista, otras formas de pensamiento, ideas alternativas y conocimientos complementarios.
- La posibilidad de dividir las labores dentro de un proyecto, según la especialidad o la afinidad.
- Compartir responsabilidades y logros.
- Tener momentos de distensión y charlas sobre temas ajenos al proyecto.
- Acortar los tiempos de producción.
- El sano debate, el intercambio de ideas y las perspectivas diferentes.

Cabe aclarar que es necesario realizar una separación entre la esfera personal y la laboral; de este modo, evitaremos el error de transferir nuestros problemas hacia los proyectos. ◀



GESTIÓN FINANCIERA

Generalmente, no tenemos herramientas para verificar y optimizar el movimiento del capital. Veamos qué áreas generan movimientos de dinero.

INGRESOS

Proyectos

► Alcanzar los objetivos fijados será nuestra meta, respetando los límites del presupuesto. Cobro del servicio puntual que originó un requerimiento por parte de un cliente.

Ahorro e inversiones

► Como los que se producen al prestar dinero a un banco para que, a su vez, este lo preste a otras personas; o la compra de divisas extranjeras.

Comercialización de productos

► Algunos estudios, además, venden: revistas, ropa y accesorios de vanguardia, objetos "de diseñador" y juguetes, entre otros artículos.

Servicios extra y alquiler

► Alojamiento de sitios en servidores web, registro de dominios internacionales o muy específicos. Alquiler de notebooks, kioscos interactivos, y proyectores para eventos y presentaciones.

Gastos fijos

► Son los gastos que tenemos sucesivamente con el correr de un período. Incluye impuestos, servicios como telefonía, proveedores de Internet, asesoría legal y contable continua, alquiler del espacio de trabajo, expensas y sueldos, entre otros.

Gastos variables

► Proveedores, como asesores, redactores, locutores, imprenta, alojamiento en el servidor. Viáticos, que incluyen los viajes, alojamiento en hoteles, combustible, comida, transporte.

Gastos extraordinarios

Estos gastos llegan cuando menos los esperamos, por lo que deberemos ser previsores. Son los gastos de oficina ante el faltante de algún elemento, reparaciones, accidentes e imprevistos.

Inversiones

Capacitación, especializaciones y cursos; imagen, publicidad y promociones; adquisición de maquinaria y renovación de tecnología; e incorporación de empleados y oficinistas.

EGRESOS



NEGOCIACIONES

EFFECTIVAS I

Con toda seguridad, el diseñador se enfrentará con miles de situaciones en las cuales deberá buscar un acuerdo con un posible cliente mediante diversas tácticas, para poder obtener un trabajo.

En esta sección, hablaremos sobre la negociación como un proceso en el que personas con intereses divergentes (que, por eso mismo, se transforman en "partes") se comunican con el objetivo de acordar un curso de acción que posea beneficios para todos los involucrados. Sin embargo, existen algunos escenarios en esta profesión que no dan lugar a hablar de negociación.

INTERESES EQUIVALENTES

Esta situación se plantea en las raras ocasiones en que el cliente acepta absolutamente todo lo expresado en el presupuesto: precios, plazos, característi-

cas y condiciones del servicio. Entonces, no hay negociación, sino intereses equivalentes, que llevan al desarrollo natural de una acción predefinida por una de las partes.

LICITACIONES Y CONCURSOS

Los interesados pueden presentar un presupuesto disponible y condiciones propias, contratando al que las acepte o mejore. Este es el caso de las licitaciones públicas de organismos gubernamentales. De igual forma, los concursos de diseño ya poseen reglamentaciones, un premio (o pago por su trabajo al ganador), un jurado inapelable y un plazo de entrega. El concepto de negociación aquí

también es inaplicable: se acepta lo establecido o no se participa.

LA PERSONALIDAD DEL NEGOCIADOR

Antes de negociar, es importante tomarse un tiempo para hacer una breve planificación. Esto brinda seguridad, denota preparación y permite enfocarse directamente en el problema que se debe solucionar en la reunión. Para tal fin, se designará a una persona con autoridad en el estudio de diseño como la responsable de llevar la negociación adelante. Si se trata de un trabajador freelance, será él mismo, siempre que reúna las siguientes características:



- Saber exactamente en qué consiste el servicio, incluso, detalles técnicos.
- Conocer lo más posible acerca del cliente, su organización y el sector en el que opera.
- Tener poder de decisión conferido por el estudio (aunque a veces puede simular falta de poder, para ganar tiempo ante una cuestión difícil).
- Ser una persona segura de sí misma, sociable, paciente, no agresiva, capaz de identificar sus miedos y sobreponerse a ellos.
- Tener la capacidad de escuchar activamente para detectar miedos, prejuicios y necesidades reales en la otra persona.



UNA NEGOCIACIÓN NO IMPLICA MERAS IMPOSICIONES UNILATERALES, SINO LA BÚSQUEDA CONJUNTA DE UN ACUERDO.

OBJETIVOS DE MÁXIMA

Se trata de pensar las condiciones más favorables que el diseñador podría pedir a ese cliente puntual a cambio de su trabajo; naturalmente, sin ingresar en el terreno de los caprichos o abusos. Presentarse con un solo objetivo –por ejemplo, que se acepte el precio sin modificaciones– es una actitud demasiado estrecha. Si el cliente no lo hace, inmediatamente se comenzará a perder, y no existirán otras variables para balancear el debate. Entonces, habrá que concebir varios objetivos, cada uno formulado atendiendo al máximo beneficio para la parte prestadora del servicio; los llamaremos “objetivos de máxima”. Por ejemplo, se escribirá: “De esta reunión quiero:

- **Objetivo 1:** que el cliente acepte un determinado precio en efectivo,
- **Objetivo 2:** abonando un

determinado porcentaje al inicio,

- **Objetivo 3:** estableciendo cierto plazo expresado en días,

- **Objetivo 4:** con límites de correcciones al prototipo preliminar”.

OBJETIVOS DE MÍNIMA

Acto seguido, se planteará hasta cuánto se está dispuesto a ceder en cada objetivo, de acuerdo con el valor que el trabajo tenga para el estudio o profesional; estos son los “objetivos de mínima”, por debajo de los cuales se abandonará la negociación. Es posible establecerlos, considerando estas pautas:

- El valor que el cliente y el trabajo posee para el profesional, además de lo estrictamente monetario: si la pieza por desarrollar poseerá gran exposición pública, servirá para alcanzar un escalón superior de reconocimiento o se generará una relación comercial extendida en el tiempo, entre otros factores.
- La cantidad de trabajos que el profesional ya posee: si no se necesitan desesperadamente nuevos ingresos o se cuenta con gran número de clientes, los objetivos de mínima serán cada vez más próximos a los de máxima.

RECOMENDACIONES AL NEGOCIAR

Al momento de negociar, debemos tener presentes las pautas detalladas a continuación:

- Averiguar la mayor cantidad y calidad de información sobre la contraparte, su organización y su sector.
- Nunca prejuizar al cliente y actuar en consecuencia. No obstante, se tendrá prudencia sobre la base de la información obtenida de antemano.
- Crear un clima distendido, charlando al inicio brevemente sobre otros temas.
- Generar confianza mediante el cumplimiento de los compromisos asumidos, la muestra de casos de éxito y recomendaciones de clientes anteriores.
- Los gestos y la posición corporal deben transmitir el mismo mensaje que las palabras. En una comunicación interpersonal, más de la mitad de los significados se producen por gestualidad y proxémica.
- No tener miedo de decir “no”. Sin embargo, este debe estar claramente fundamentado, invitando al otro a que comprenda lo desfavorable de su propuesta. En seguida, se presentará una alternativa más viable, basada en la rechazada.
- Examinar el acuerdo al que se arribó, para compararlo con el planteamiento previo de objetivos de máxima y de mínima.

Los consejos mencionados son solo algunos de los que conviene tener presentes para negociar. Sin embargo, será necesario sumar otras tácticas según el desarrollo de la negociación. ◀



LA IRRACIONALIDAD

Se refiere a cuando el cliente pretende beneficios caprichosos, eliminando toda posibilidad para que el diseñador obtenga algo positivo. Es el ya célebre: “lo quiero gratis, y lo quiero ahora”. Tal actitud, claramente abusiva, revela en el cliente una total falta de respeto hacia la otra persona y su trabajo. El diseñador hará bien en recordar que una negociación no implica meras imposiciones unilaterales, y se marchará, rápida y cortésmente.

NEGOCIACIONES EFECTIVAS II

Los posibles clientes poseedores de actitudes manipuladoras utilizan ciertas frases que, en general, los ponen en evidencia. En esta segunda entrega del tópico "sobre negociaciones" comprenderemos cómo manejar estos argumentos falaces.



LAS ACTITUDES Y LAS PERSONAS

Aquí consideramos las actitudes como la forma en que un cliente aprendió a reaccionar ante situaciones que él percibe similares a otras de su pasado. Entiende que así pudo salir victorioso y las repite, poniéndolas en práctica de antemano. Quizá vea la negociación como una batalla en la que hay que mantener una posición, ganar y probar su valía, a expensas de los beneficios del otro. Es primordial alejarse del pensamiento de que sus malas actitudes son causadas, necesariamente, por el diseñador, la calidad del servicio o su desempeño.

de trabajar con él, a cualquier precio.

► Ponerse en el lugar de "cliente grande" y con muchas opciones.

No es probable que cumpla esta promesa porque, en cuanto tenga un nuevo encargo (si es que alguna vez lo tiene), hará la misma propuesta a otro diseñador, buscando continuamente quien aún crea en él y abarate sus presupuestos.

LA DESVALORIZACIÓN

"El amigo de una amiga, que hace lo mismo que tú, me cobra la mitad del dinero que me pides". Aquí es posible reconocer la **desvalorización**. El cliente está intentando:

- Hacerle creer al diseñador que no tiene nada de especial, que muchas personas hacen "lo mismo" que él. Y naturalmente, si la oferta por un servicio aumenta, su precio disminuye debido al incremento de la competencia.
- Tratar al diseño como a un producto industrializado, disponible en una góndola de supermercado.
- Bajar deliberadamente el precio que se le presentó, menoscabando el valor del servicio.

Es interesante preguntarse: si el precio es tan obscuro como dice, ¿por qué no contrató a los otros y continúa en esta negociación? A modo de sugerencia, el profesional podría recordarle que:

En esta oportunidad, repasaremos algunas expresiones argumentativas utilizadas por ciertos clientes, que fueron extraídas de charlas con colegas y experiencias propias. No son declaraciones absolutas ni válidas para todos los casos, pero es interesante tenerlas presentes en las reuniones de negocios.

EL ARGUMENTO PROMETEDOR

"Este es solo el comienzo. Vamos a hacer muchos trabajos más, todos grandes. Te voy a recomendar con mis amigos empresarios, que están necesitando un diseñador como tú". En este argumento, es posible reconocer la **falsedad**. El cliente está intentando:

- Tejer la ilusión de que el diseñador necesita de él, mucho más que él del diseñador.
- Venderle al diseñador el servicio de incremento de su cartera de clientes, confiando solo en su palabra y luego de finalizada la relación comercial.
- Generar en el diseñador la urgencia



- Si busca bien, seguramente encontrará quien lo haga aún más barato que eso, o gratis.
- Si lo que valora es el precio muy por encima de las características y la calidad del servicio, la reunión debería de llegar a su fin: se hace imposible competir contra lo gratuito o lo irrazonablemente barato.

No se puede desperdiciar tiempo. Existen muchos otros clientes que sí desean calidad y confiabilidad.

EL ARGUMENTO OMNIPOTENTE

"En realidad, esto no es nada complicado. Es muy sencillo. Lo haría yo mismo, pero no tengo tiempo". Otra vez, el profesional está en presencia de la **desvalorización**, con una pizca de omnisciencia. El cliente está intentando:

- Hacer creer que sabe de diseño, y puede juzgar cuándo un trabajo es sencillo y cuándo no lo es.
- Quitarle magnitud al trabajo y hacerlo percibir como fácil. Incluso, dice que podría encargarse él mismo. Si es algo fácil, obviamente, también es algo barato.
- Degradar la profesión, indicando que alguien sin conocimientos podría hacerlo de igual forma.
- Sugerir que, para hacer diseño, se requiere únicamente utilizar la computadora (algunas personas piensan que la máquina lo hace sola).

El trabajo de diseño no es sencillo. Requiere de un profesional, de estudio, capacitación, práctica, reflexión, planificación y responsabilidad. Es evidente que no cualquiera puede hacerlo.

LA PRISA

"Lo quiero para mañana". Con esta expresión de autoritarismo, el cliente está intentando:

- Acortar drásticamente los tiempos de producción, para reducir el presupuesto.

EL TRABAJO DE DISEÑO NO ES SENCILLO: REQUIERE DE ESTUDIO, PRÁCTICA, REFLEXIÓN, PLANIFICACIÓN Y RESPONSABILIDAD.



Su pensamiento es: "si lo hace en un día, en lugar de en diez, debería cobrarme menos".

- Restar importancia, otra vez, al trabajo del diseñador.
- Dar a entender que los plazos de ejecución presentados por el profesional son caprichosos, de manera que su presupuesto también lo es.
- Restar tiempo al diseñador para pensar en una respuesta que lo beneficie. La urgencia permanente, por definición, anula la reflexión.
- Quitarse la presión de encima. Quizá sea cierto que lo precisa para mañana, pero, ciertamente, esto no es responsabilidad ni culpa del diseñador.

Además, el cliente cree que es el único que el diseñador tiene y que está enteramente a su disposición. Si desea que trabaje en forma exclusiva para él, ya debería hablar de contratos y otras cuestiones legales.

LA VÍCTIMA

"Sé que eres un gran profesional, pero la situación está dura y no tengo dinero". El cliente, con ayuda de la actitud de víctima, está intentando:

- Generar un sentimiento de culpa en su contraparte, para dirigir sus acciones, poniéndose intencionalmente en un lugar de inferioridad.
- Que el diseñador, al estar en un lugar de mayor "poder" que él, tenga misericordia y compasión. Que lo ayude a pasar la mala situación.
- Obtener el mismo servicio pagando menos dinero por él, lisa y llanamente.

Si no conoce mucho a esta persona, el profesional hará bien en recordar que, si accede a bajar sus precios, quien luego se encontrará en una mala situación será él mismo. Estas frases no ocurren solo en el ámbito del diseño: se escuchan a diario en profesionales como médicos, ingenieros, arquitectos y abogados. ◀

CLIENTES MANIPULADORES

Identificaremos aquí distintos tipos de actitudes manipuladoras que algunos clientes podrían mostrar en una relación comercial.



AUTORITARISMO

- ▶ Todos los detalles del servicio deben estar realizados según lo que él dice.
- ▶ Confunde intencionalmente "lo que está bien", con "lo que le conviene".
- ▶ No permite y no escucha opiniones. No tiene flexibilidad ante las circunstancias.
- ▶ Sus dichos suenan como órdenes.
- ▶ Se enfurece y se ofusca ante el debate.
- ▶ No experimenta ningún tipo de solidaridad con el profesional.
- ▶ Utiliza la amenaza para lograr acciones originadas por el miedo.
- ▶ El interés en el profesional solo dura mientras que este acate las órdenes.



FALESDAD

- ▶ Falsea los hechos con el objetivo de lograr beneficios.
- ▶ Realiza promesas de las que no puede aportar ninguna prueba.
- ▶ Las promesas: recomendaciones, más trabajos, mayores ingresos crecimiento acelerado, reconocimiento público.
- ▶ Se irrita cuando se le pide más datos o se le pregunta sobre el tema.
- ▶ Simula falta de recursos económicos para afrontar el servicio.
- ▶ Sin embargo, se muestra entusiasta para comenzar lo antes posible.



DESVALORIZACIÓN

- ▶ Sistemáticamente resta valor a todo lo que realiza y representa el profesional que va a contratar.
- ▶ Utiliza la crítica con mucha frecuencia y no propone otros caminos viables.
- ▶ Espera que el diseñador adivine lo que le gusta o lo que está pensando.
- ▶ Pone de relieve los errores y puntos negativos, pero jamás reconoce los positivos.
- ▶ Induce al error, proveyendo datos inexactos, fechas de entrega imposibles, o estándares demasiado altos, dado el precio pactado.
- ▶ Se haga lo que se haga, jamás será "lo suficientemente bueno".
- ▶ Quiere dejar la sensación de que le está haciendo un favor al contratar sus servicios, por lo que el profesional debe mostrar su agradecimiento, por ejemplo, bajando sus precios.

► TIPS

Para poder sortear todos los obstáculos que nos pondrán los clientes malintencionados deberemos desarrollar una autoestima fuerte, gran seguridad en las propias capacidades y limitaciones. También deberemos reconocer abiertamente ambas. No es necesario esconder nada, nadie es perfecto.

Es importante tomar distancia de la situación, respirar hondo y ordenar los pensamientos. Dejar de lado la interpretación de un problema personal con el cliente, solo está mostrando la forma en que aprendió a actuar en situaciones pasadas, que percibe como similares.

Tenemos que evitar, siempre que podamos, la confrontación directa. Invitar al otro a que se ponga en el lugar del diseñador. Tratar de identificar los sentimientos de culpa, miedo, vergüenza o inferioridad, antes de que lleven a alguna acción no deseada.

Por último, siempre existe otra opción: abandonar la negociación y evitar una relación comercial enfermiza.



- Busca hacer que la otra persona actúe movida por un sentimiento de culpa.
- Se coloca intencionalmente en un lugar de inferioridad ante el diseñador.
- Puede decir sentirse "traicionado", "desilusionado" y "desengañado", por culpa de aquel.
- Implementa la adulación repentina para lograr su cometido.
- Utiliza el sentimiento de culpa del "victimario" sobre un hecho, para darle una oportunidad de expiar las culpas, si actúa como el cliente dice.
- Deja en claro lo que debería haberse hecho, con el propósito de que en el futuro se haga de esa forma.

VICTIMIZACIÓN



- Aunque no sea diseñador, ni sepa del tema, parece saberlo todo.
- Todo lo que dice es naturalmente cierto, para lo cual recurre a citas ficticias, estudios científicos difusos y experiencias de amigos.
- Para él existe la noción de "razón", nunca la de "puntos de vista".
- Utiliza una abrumadora cantidad de palabras, sin lugar a aportes de la otra parte.
- Usa la ilusión de "ignorancia" que genera en la otra persona, para hacerla sentir inferior.
- Busca lograr el lugar de "voz autorizada" sobre diversos temas.
- Su objetivo también es el de hacer que el profesional dude de sus conocimientos para sacar ventajas de ello.

OMNISCENCIA



GENERAR RELACIONES CONTRACTUALES

En este apartado conoceremos los mejores consejos para saber cómo proteger nuestro trabajo mediante contratos laborales. Es una tarea muy importante para el diseñador que se desempeña en condición de freelance.

Una vez redactado y aprobado el presupuesto que el diseñador realiza, donde figuran las características específicas del servicio por prestar, se hace necesaria la explicación de las particularidades de la relación comercial que se generará, en una pieza escrita. Se firma un contrato, entonces, para contribuir a asegurar los siguientes objetivos:

- Obtener la certeza acerca de los derechos y obligaciones que cada parte (cliente y diseñador) contrajo como consecuencia de dicha relación.
- Contar con una herramienta probatoria de su existencia, ante confusiones y entredichos.
- Establecer cursos de acción si ocurren incumplimientos de cualquiera de las partes.
- Definir la duración de la relación y las causas de su ruptura anticipada.
- Resolver controversias y evitar la problemática situación de "mi palabra contra la suya".

Además de los objetivos que acabamos de detallar, para realizar un buen contrato es necesario tener en cuenta otras cuestiones importantes, como las partes fundamentales que lo conforman.

LOS SUJETOS

Dado que todo contrato es un acuerdo entre dos o más partes que tienen un propósito común, aunque intereses disímiles, en él deben consignarse sus datos personales, y si actuarán por sí mismos o lo harán por medio de sus representantes. Si el cliente del diseñador es una empresa, resulta necesario verificar que la persona firmante (en nombre de ella) cuente con el poder suficiente para obligarla.

EL OBJETO

En todo vínculo comercial, de forma oral, los involucrados se comprometen a dar algo a cambio de un beneficio. El objeto del contrato, por consiguiente, se compone de las obligaciones y los derechos que cada parte asume. Típicamente,

consta de las modalidades de ejecución; es decir, la forma en que se pacta que se llevará adelante el proceso vinculado al servicio por prestar, como por ejemplo:

- Los plazos de ejecución del proyecto y las formas en que el cliente pagará por los servicios: en cuotas, por porcentajes de avance, en efectivo, cheques, depósitos y transferencias bancarias, con otros bienes, etcétera.
- Lugar y forma de entrega del proyecto finalizado.
- Posibilidad de subcontratación. Equivale a manifestar si el profesional recurrirá a otros para resolver ciertas partes del proyecto.
- Los cursos de acción ante posibles incumplimientos del cliente y el diseñador. Por ejemplo: aplicar sanciones ante retrasos en los pagos o en la provisión de información vital para el desarrollo del servicio.

LA FINALIDAD

Se trata del beneficio que cada uno obtendrá, a partir del cumplimiento efectivo del contrato. Es el "para qué" del compromiso mutuo, la pretensión de cada parte sobre la actividad de la otra. Recordemos que los beneficios son la condición elemental de cada uno de nuestros trabajos.

ANEXOS

Ya que el contrato, en sí mismo, regula la relación comercial, necesita de anexos (o agregados) que traten del contenido técnico y específico del servicio que el cliente obtendrá del diseñador. Por ejemplo: es buena idea incluir el presupuesto detallado dentro del anexo, como parte integrante del contrato.



ANTES DE COMENZAR A TRATAR CON CLIENTES, EL DISEÑADOR NECESITA GENERAR SU MODELO DE CONTRATO DE SERVICIOS.

SOLUCIÓN DE CONTROVERSIAS

A modo preventivo, pueden establecerse modos de solución de conflictos que pudieran surgir durante la ejecución del contrato. Por ejemplo, la posibilidad de recurrir a medios alternativos, como la mediación o el arbitraje; y la definición de la jurisdicción de los tribunales que entenderán en la causa, especialmente útil cuando se realizan trabajos a distancia, es decir, cuando ambas partes residen en distintas ciudades o países.

RECOMENDACIONES

Ahora que conocemos los objetivos y las partes del contrato, detallaremos algunas recomendaciones muy útiles:

- ▶ Tanto para confeccionar, como para aceptar y firmar un contrato, es imprescindible consultar previamente a un abogado en ejercicio. Nunca hay que firmar un contrato provisto por la otra parte bajo coacción o presión de algún tipo.
- ▶ Antes de comenzar a tratar con clientes, el diseñador profesional necesita generar su propio modelo de contrato de servicios, adaptado a sus requerimientos y situación particular, con el auxilio del abogado. Desalentamos el uso de modelos obtenidos de Internet o de otras empresas del rubro, aunque provengan de la misma región geográfica.
- ▶ Los acuerdos verbales –sobre todo, en proyectos de larga duración– se prestan a equívocos, suspicacias e intentos de obtener ventajas. Por lo tanto, es altamente recomendable volcar todos los acuerdos con el cliente por escrito, en forma de un contrato.
- ▶ El contrato también protege al diseñador ante cambios repentinos en la empresa cliente: despido o fallecimiento de la persona que lo contrató, cambios en el organigrama, y fusiones o adquisiciones.
- ▶ El profesional hará bien en desconfiar de quien se niegue a prestar su firma para el convenio. ¿Piensa que no lo cumplirá, no quiere compromisos? ¿Intenta

luego cambiar las reglas de juego con impunidad? ¿Desea tener el control absoluto del desarrollo del proyecto?

▶ Si bien es imposible abarcar cada detalle de la relación comercial, es importante definir con claridad y sencillez los aspectos que podrían suscitar problemas.

Es importante aclarar que lo detallado en esta sección no es completamente abarcativo, ya que cada una de las relaciones laborales estará sujeta a las negociaciones de ambas partes. ◀



OTROS TIPOS DE CONTRATOS

Es posible que, en proyectos de gran envergadura o con contenidos internos, el cliente condicione su realización a la firma de otros contratos, como los de confidencialidad. Esto es completamente razonable, porque en ellos, él busca proteger datos que producirían distintos tipos de pérdidas al revelarse a su competencia o al hacerse públicos. Tener esta presente importa, ya que los datos de los clientes tienen absoluta prioridad.



PRIMER AÑO DEL EMPRENDIMIENTO

Los primeros pasos en un emprendimiento son los más difíciles de dar, pero si realizamos una correcta planificación, podremos reducir la cantidad de imponderables. De este modo, lograremos una buena posición en el mercado laboral.

Según diversos estudios, nueve de cada diez empresas de reciente formación no llegan, en Latinoamérica, a su segundo año de vida. Esta alarmante cifra, lejos de desalentar al emprendedor, debe hacerle analizar y tomar conciencia sobre sus principales causas, para elaborar mínimos planes de contingencia o alternativas de acción antes de lanzarse a un nuevo emprendimiento. El hecho de fracasar y no alcanzar ciertos objetivos significa que todo se puede mejorar, que se ha aprendido mucho de la situación y que se seguirá adelante aún con más ímpetu. Es cierto que, en general, en esto intervienen una gran combinación de causas, siempre distintas, como las personas que conforman

las empresas. Sin embargo, es posible reconocer algunas como predominantes.

LAS CAUSAS EXTERNAS

Crisis económicas, inflación, cambios repentinos de políticas en ciertos ámbitos y aprobación de regulaciones desfavorables, entre otros, son factores imprevisibles y críticos para quienes no cuentan con contactos en el poder. Por otro lado, existen factores previsibles que quiebran la resistencia de muchos nuevos emprendimientos: insuficientes incentivos gubernamentales, la asfixiante cantidad de retenciones e impuestos, la limitadísima cantidad de crédito disponible y la dificultad para conseguir inversores, por nombrar solo algunos. Es importante tener en cuenta que,

desde aquí en adelante, todas las demás causas seleccionadas entrarán en el ámbito del equipo emprendedor.

LA ELECCIÓN DE LOS SOCIOS

No constituye una elección recomendable seleccionar socios con quienes nunca se trabajó en proyectos reales: "fuimos juntos a la universidad, por eso vamos a ser socios".

Una relación laboral dista mucho de una amistad, al incluir dinero, plazos de ejecución, presiones y negociaciones sobre la dirección que seguirá la empresa. Por otra parte, existe la posibilidad de crear y mantener una red de proveedores especializados en distintas partes de nuestro proceso de servicios y contratarlos por la duración del proyecto.





DESCONOCIMIENTO DEL MERCADO

Si bien es difícil sacar conclusiones sobre el comportamiento futuro de los consumidores, no es recomendable realizar estimaciones tan optimistas del requerimiento de un servicio, basados en supuestos propios. En cuanto a la competencia, las suposiciones juegan en contra. Una muy común consiste en subestimarla, actuando como si sus posibilidades y aptitudes fueran inferiores, y que el público se dará cuenta de esto. Otro error se da cuando la empresa no observa a los nuevos competidores que, seguramente, tendrá en muy poco tiempo, y que intentarán obtener la preferencia del mercado.

DESCONOCIMIENTO DE LA GESTIÓN DEL NEGOCIO

Existen estudios de diseño que nacen sin conocimientos sobre la gestión de una empresa. De repente, los nuevos emprendedores deben lidiar con:

- ▶ El flujo de caja, estados de resultados y situación fiscal.
- ▶ El pago de impuestos, servicios y proveedores.
- ▶ La elaboración e implementación de un plan de marketing.
- ▶ Negociaciones, gestión de clientes y gestión de cobranzas.
- ▶ Aspectos legales.

Estas cuestiones antes estaban reservadas a los responsables de las empresas, y los empleados tenían un acceso limitado a esta información. En consecuencia, los socios suponen que el saber técnico específico profesional, por sí solo, es suficiente para establecer su propia empresa. Se concentran en esto y dejan de lado

tareas tan fundamentales como las mencionadas. La falta de manejo de estos aspectos lleva a hacer trabajos de una calidad funcional y visual excelente, pero a tener una empresa deficitaria e impredecible.

FUERA DE FOCO

En los primeros tiempos de un emprendimiento de servicios de este tipo, existe la tentación de aceptar todos los trabajos que se presenten, sin importar que no correspondan al foco del negocio. Esta tentación se sustenta en la necesidad de tener dinero disponible en caja y en el miedo a que no se presenten nuevas ofertas para sostener la empresa. El problema de esta acción es que implica un desvío en la hoja de ruta previamente trazada, a la que se vuelve solo en los

tiempos libres. Pero se regresa siempre al mismo punto: el tiempo pasa, y no se avanza en la experiencia ni en la misión específica de la empresa.

EN RESUMEN

Es imprescindible planificar el servicio, los procesos productivos y otros procesos de la empresa. Sin embargo, es imposible prever todos los errores o fallas por falta de experiencia que se podrían presentar. Para los comienzos de un emprendimiento, es útil plantear el servicio en etapas, como una serie de tareas que componen el trabajo de producción: desde el contacto con el cliente, hasta la entrega del trabajo y los servicios de postventa, pasando por la elaboración de un brief, las instancias de presentación y aprobación de prototipos, la programación, etc. Es de gran ayuda ubicar estas etapas en un diagrama basado en una línea de tiempo. En cada tarea de este diagrama se intentará dar respuesta a las siguientes preguntas:

- ▶ ¿Qué podría ir mal en cada fase, por qué?
- ▶ ¿Se tiene algún plan por si, efectivamente, algo malo llegara a suceder?
- ▶ ¿Se puede corregir desde ahora? ◀

ES IMPORTANTE ELABORAR PLANES DE CONTINGENCIA O ALTERNATIVAS DE ACCIÓN ANTES DE LANZAR UNA NUEVA EMPRESA.



EL ENAMORAMIENTO DE LA IDEA

La pasión de emprender, del nuevo desafío y el amor hacia la propia idea pueden conducir a actuar de manera precipitada. Es posible que si la empresa no da los frutos esperados, aún después de implementar estrategias para revertirlo, sencillamente se esté equivocado con respecto al emprendimiento. Entonces, sería mejor abandonarlo antes de causar mayores daños al emprendedor y a quienes lo acompañan. Es aquí donde, con toda sinceridad y autocrítica, se diferenciará la utopía revolucionaria de la genuina oportunidad de negocios.

ACTIVIDADES

TEST DE AUTOEVALUACIÓN

01 ¿Qué necesitamos para trabajar Catalyst?

02 ¿De qué forma podemos administrar el tiempo?

03 ¿Cómo podemos conseguir clientes online?

04 ¿Qué necesitamos para planificar un portfolio?

05 ¿De qué manera podemos determinar el precio de nuestro trabajo?

06 ¿Cómo funciona una empresa?

07 ¿Qué necesita una presentación para ser efectiva?

08 Mencione algunas actitudes nocivas para un diseñador.

09 ¿Cómo podemos negociar en forma efectiva?

10 ¿Cómo podemos tratar con clientes manipuladores?

PRÁCTICAS

01 Planifique su portfolio de trabajo.

02 Determine el precio de su trabajo.

03 Realice una presentación efectiva.

04 Planifique su gestión financiera.

05 Genere un emprendimiento personal.

CURSO VISUAL Y PRÁCTICO



Servicios al lector



▶ ÍNDICE TEMÁTICO

▶ CATÁLOGO



ÍNDICE TEMÁTICO

A

Aceleración	114
ActionScript 3.0	018
Actitudes nocivas	296
Administrar el tiempo	274
Adobe	043
Adobe Media Encoder	192
APIs	208
Aplicaciones para Facebook	017
AVI	190

B

Banners	267
Bitmap	138
Blur de movimiento	135

C

Capas	095
Cargar coordenadas	215
CDs interactivos	014
Cinemática inversa	108
Classic Tween	096
Cliente-servidor	160
Clientes online	276
Content Management System	162
Contribute	043
Controladores de eventos	119
Crear texto clásico	058
Crear un formulario	153
Crear una scroll bar	068

D

Decó	078
Desenfoco de imágenes	135
Dibujo en Flash	076
DNS	247
Dominios regionales	251
Dreamweaver	042

E

E-Learning	017
Editar con Soundbooth	182
Efectos físicos	114
Elasticidad	114
Emprendedor	280
Empresa	292
Entornos 3D	080
Envío de formularios	160
Etiquetas HTML	065

F

Facebook API	218
Filtros dinámicos	136
Fireworks	043
Flash Builder	043
Flash Catalyst	043

Flash Player	021
FlashBlocks	166
FLV	190
Formas	074
Formas de trabajo	266
Formato con CSS	065
Formato limitado	064
Formatos de video	190
Formularios en Flash	150
Fotogramas	094
Fotogramas clave	095
Freelance	266, 272

G

Galería de fotos	140
Gestión financiera	298
Gestores de contenido	164
Google Maps	212
Gravedad	114
Guías de movimiento	097

H

Herramientas	046
Herramientas avanzadas	074
Herramientas básicas	074
Hosting	256
HTML	064

I

Illustrator	042, 047
Imágenes en la librería	132
Imágenes externas	132

Imágenes vectoriales	132
Incorporar imágenes	132
Interfaces	022
Intros en Flash	082
IP	246

L

Librerías de animación	116
Librerías de código abierto	123
Librerías de sonido	176
Liderazgo	281
Líneas de tiempo	094

M

Mapa de bits	132
Menú contextual	026
Metasecuencias	157
Migrar botón a Flash	051
Modelo de negocio	288
Motion Editor	104
Motion Preset	101
Motion Tween	100

P

Papervision 3D	216
Photoshop Extended	042
Planificar un portfolio	282

Pluma	074
Portfolios	015
Precio del trabajo	286
Preload	133
Presentaciones efectivas	294
Propiedades CSS	065
Protocolos	246



Receptor	242
Registrar un dominio	252
Remitente	242



Salida laboral	264
Sincronización de sonido	172

Sistema de partículas	122, 124
Sistema web	249
Sitios de publicidad	015
Sitios experimentales	016
Slideshows animados	085
Social Media	219
Socialización	281
Soundbooth	180
Suavizado tipográfico	057
Swfobject	224



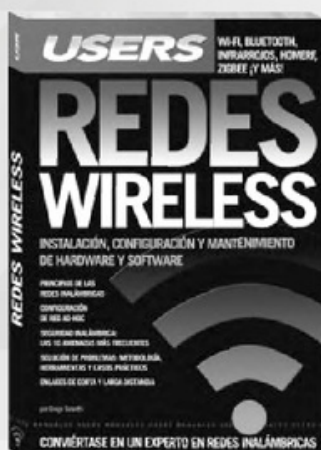
Texto clásico	054
Texto predictivo	158
Tipos de dominio	248
Tips de dibujo	075
TLF	060
TweenMax	118





Este libro está dirigido tanto a los que inician con overclocking, como aquellos que buscan ampliar sus experiencias.

→ 320 páginas / ISBN 978-987-1857-30-2



Este libro único nos introduce en el fascinante y complejo mundo de las redes inalámbricas.

→ 320 páginas / ISBN 978-987-1773-98-5



Esta increíble obra está dirigida a los entusiastas de la tecnología que quieran aprender los mejores trucos de los expertos.

→ 320 páginas / ISBN 978-987-1857-01-2



Esta obra se encuentra destinada a todos los desarrolladores que necesitan avanzar en el uso de la Plataforma Adobe Flash.

→ 320 páginas / ISBN 978-987-1857-00-5



Un libro clave para adquirir las herramientas y técnicas necesarias para crear un sitio sin conocimientos previos.

→ 320 páginas / ISBN 978-987-1773-99-2



Una obra para aprender a programar en Java y así insertarse en el creciente mercado laboral del desarrollo de software.

→ 352 páginas / ISBN 978-987-1773-97-8



Este libro presenta un nuevo recorrido por el máximo nivel de C# con el objetivo de lograr un desarrollo más eficiente.

→ 320 páginas / ISBN 978-987-1773-96-1



Esta obra presenta todos los fundamentos y las prácticas necesarios para montar redes en pequeñas y medianas empresas.

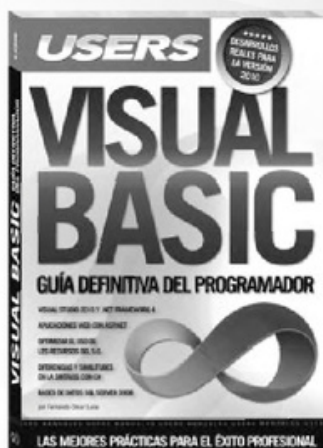
→ 320 páginas / ISBN 978-987-1773-80-0



Una obra única para aprender sobre el nuevo estándar y cómo aplicarlo a nuestros proyectos.

→ 320 páginas / ISBN 978-987-1773-79-4





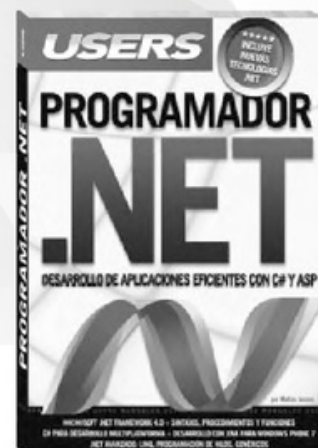
Un libro imprescindible para aprender cómo programar en VB.NET y así lograr el éxito profesional.

→ 352 páginas / ISBN 978-987-1773-57-2



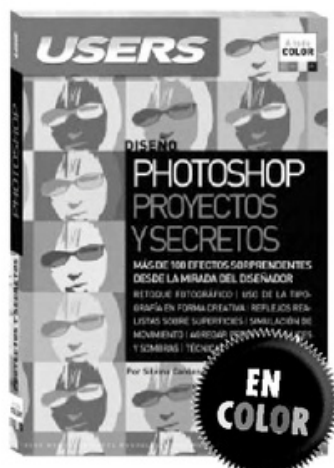
Una para quienes buscan aprender sobre los fundamentos de los microcontroladores y llevar adelante sus propios proyectos.

→ 320 páginas / ISBN 978-987-1773-56-5



Un manual único para aprender a desarrollar aplicaciones de escritorio y para la Web con la última versión de C#.

→ 352 páginas / ISBN 978-987-1773-26-8



Un libro imperdible para aprender a utilizar Photoshop desde la teoría hasta las técnicas avanzadas.

→ 320 páginas / ISBN 978-987-1773-25-1



Una obra imprescindible, para quienes quieran conseguir un nuevo nivel de profesionalismo en sus blogs.

→ 352 páginas / ISBN 978-987-1773-18-3



Un libro único para ingresar en el apasionante mundo de la administración y virtualización de servidores.

→ 352 páginas / ISBN 978-987-1773-19-0



Esta obra permite sacar el máximo provecho de Windows 7, las redes sociales y los dispositivos ultraportátiles del momento.

→ 352 páginas / ISBN 978-987-1773-17-6



Este libro presenta la fusión de las dos herramientas más populares en el desarrollo de aplicaciones web: PHP y MySQL.

→ 432 páginas / ISBN 978-987-1773-16-9



Este manual va dirigido tanto a principiantes como a usuarios que quieran conocer las nuevas herramientas de la Excel 2010.

→ 352 páginas / ISBN 978-987-1773-15-2



Esta guía enseña cómo realizar un correcto diagnóstico y determinar la solución para los problemas de hardware de la PC.

→ 320 páginas / ISBN 978-987-1773-14-5



Este libro brinda las herramientas para acercarse al trabajo diario del desarrollador los avances más importantes en PHP 6.

→ 400 páginas / ISBN 978-987-1773-07-7



Un libro imprescindible para quienes quieran aprender y perfeccionarse en el dibujo asistido por computadora.

→ 384 páginas / ISBN 978-987-1773-06-0



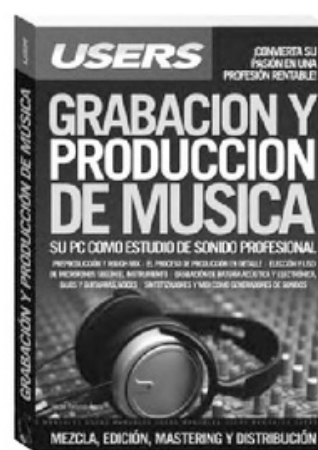
Este libro único nos permitirá alcanzar el grado máximo en el manejo de Windows: Administrador Profesional.

→ 352 páginas / ISBN 978-987-1773-08-4



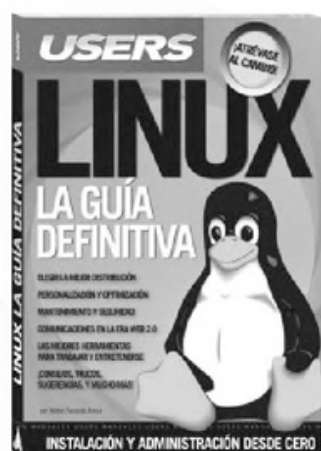
Una obra ideal para todos aquellos que busquen realizar manipulación y retoque de imágenes de forma profesional.

→ 320 páginas / ISBN 978-987-1773-05-3



Este libro presenta toda la producción musical, desde composición y masterizado, hasta distribución final del producto.

→ 320 páginas / ISBN 978-987-1773-04-6



Una obra imperdible para aprovechar al máximo las herramientas de código libre en la vida cotidiana.

→ 320 páginas / ISBN 978-987-26013-8-6



Una obra fundamental para aprender a trabajar de forma profesional con las herramientas audiovisuales de Adobe.

→ 320 páginas / ISBN 978-987-26013-9-3



Un manual ideal para aprender todo sobre la nueva versión y las posibilidades de trabajo online que ofrece.

→ 352 páginas / ISBN 978-987-26013-6-2



Un libro imperdible para aprender a usar este programa y aprovechar todas sus posibilidades al máximo.

→ 320 páginas / ISBN 978-987-26013-4-8



Esta obra única nos introduce en .NET para aprender sobre la última versión del lenguaje más utilizado de la actualidad.

→ 400 páginas / ISBN 978-987-26013-5-5



Este libro imprescindible nos enseña cómo mantener nuestra información protegida de todas las amenazas de la web.

→ 320 páginas / ISBN 978-987-26013-1-7



Un manual imperdible para guardar como guía de referencia y para aplicar siempre ante entornos complejos.

→ 368 páginas / ISBN 978-987-26013-0-0



Un libro imprescindible para expresar al máximo las capacidades multimedia que ofrecen Internet y Windows 7.

→ 352 páginas / ISBN 978-987-663-036-8



Un libro fundamental para aprender a programar desde cero con la última versión del lenguaje más robusto.

→ 368 páginas / ISBN 978-987-663-039-9



Este libro único nos brindará todas las respuestas para dominar los dos blogs más populares de la web: Blogger y WordPress.

→ 320 páginas / ISBN 978-987-663-037-5



Una obra única para aprender de manera visual cómo armar, actualizar y solucionar los problemas de la PC.

→ 320 páginas / ISBN 978-987-663-034-4



Esta obra nos dará la respuesta a todas las preguntas que necesitamos resolver para dominar por completo Windows 7.

→ 320 páginas / ISBN 978-987-663-035-1



Una obra única para exprimir al máximo el hardware del hogar sin necesidad de gastar dinero extra.

→ 352 páginas / ISBN 978-987-663-029-0



Un libro fundamental para dominar por completo el programa de bases de datos de Office.

→ 320 páginas / ISBN 978-987-663-025-2



Una obra imperdible para aprender de forma visual y práctica todo sobre las redes basadas en tecnología Cisco.

→ 320 páginas / ISBN 978-987-663-024-5



Un libro imprescindible que nos permitirá explorar todas las posibilidades que ofrece la suite a través de proyectos reales.

→ 352 páginas / ISBN 978-987-663-023-8



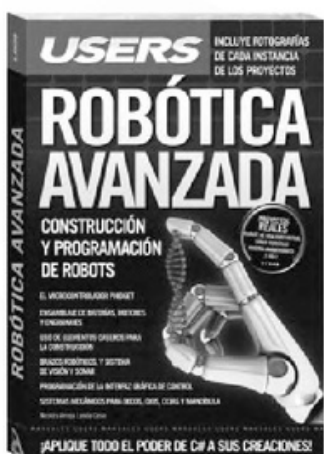
Este libro fundamental nos muestra de forma práctica cómo crear sitios web atractivos y profesionales.

→ 320 páginas / ISBN 978-987-663-022-1



Esta obra parte de la experiencia de muchos usuarios para presentar las respuestas más interesantes y creativas sobre Excel.

→ 336 páginas / ISBN 978-987-663-021-4



Un libro único para iniciarse en la creación y el desarrollo de robots complejos con elementos caseros.

→ 352 páginas / ISBN 978-987-663-020-7



Una obra imprescindible para aprender sobre todas las ventajas y novedades de Windows 7 de manera visual y práctica.

→ 320 páginas / ISBN 978-987-663-015-3



Un libro ideal para quienes quieran realizar la transición de Windows a Linux de forma sencilla y agradable.

→ 336 páginas / ISBN 978-987-663-013-9



CURSOS INTENSIVOS CON SALIDA LABORAL

Los temas más importantes del universo de la tecnología desarrollados con la mayor profundidad y con un despliegue visual de alto impacto: Explicaciones teóricas, procedimientos paso a paso, videotutoriales, infografías y muchos recursos mas.

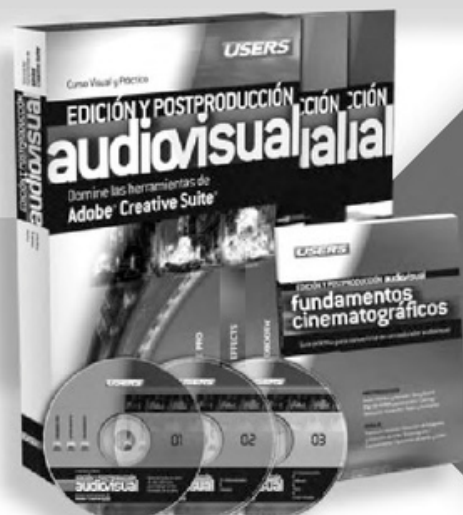


- » 25 Fascículos
- » 600 Páginas
- » 2 DVDs / 2 Libros

Curso para dominar las principales herramientas del paquete Adobe CS3 y conocer los mejores secretos para diseñar de manera profesional. Ideal para quienes se desempeñan en diseño, publicidad, productos gráficos o sitios web.

Obra teórica y práctica que brinda las habilidades necesarias para convertirse en un profesional en composición, animación y VFX (efectos especiales).

- » 25 Fascículos
- » 600 Páginas
- » 2 CDs / 1 DVD / 1 Libro



- » 25 Fascículos
- » 600 Páginas
- » 4 CDs

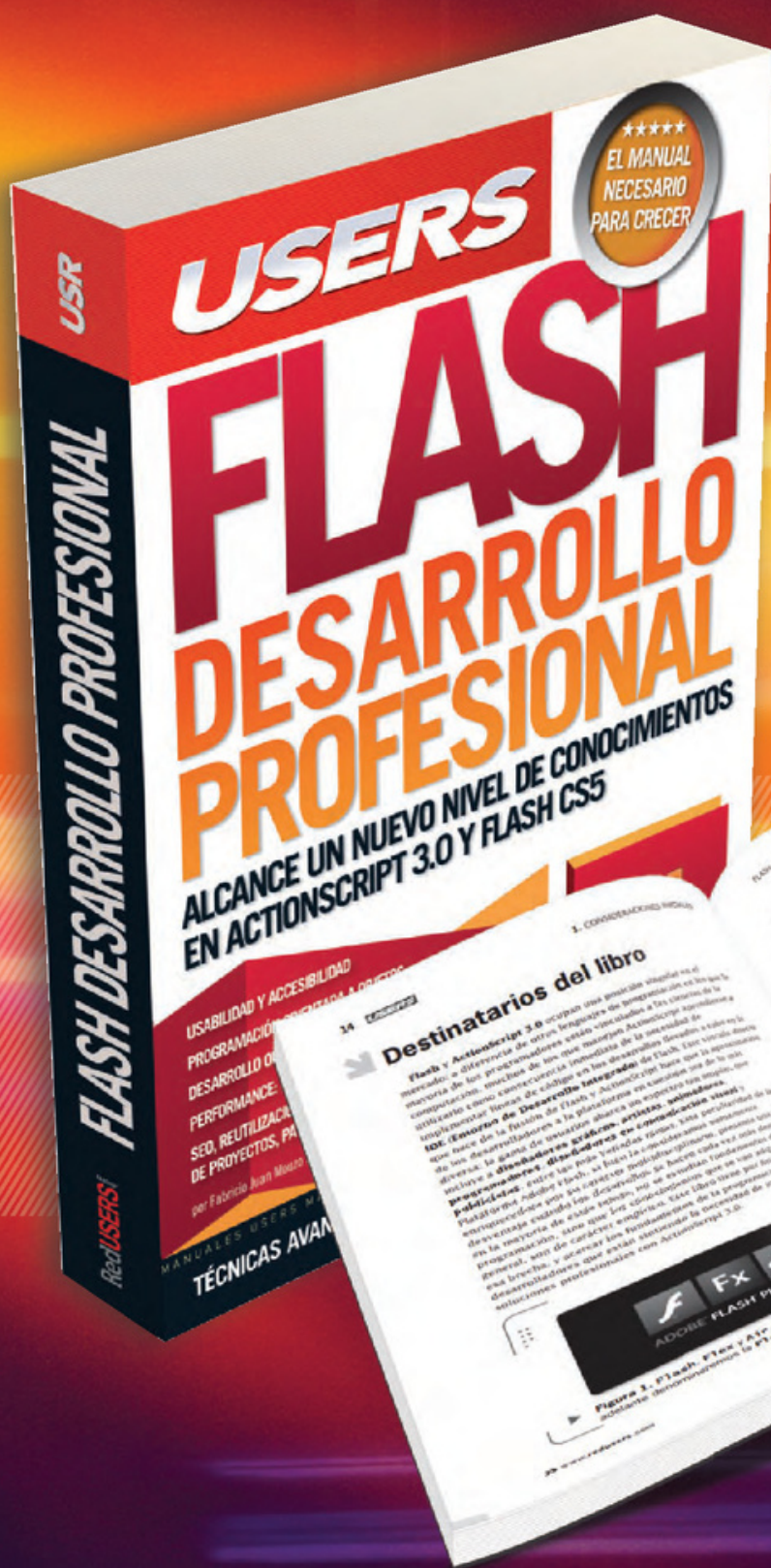
Obra ideal para ingresar en el apasionante universo del diseño web y utilizar Internet para una profesión rentable. Elaborada por los máximos referentes en el área, con infografías y explicaciones muy didácticas.

Brinda las habilidades necesarias para planificar, instalar y administrar redes de computadoras de forma profesional. Basada principalmente en tecnologías Cisco, busca cubrir la creciente necesidad de profesionales.

- » 25 Fascículos
- » 600 Páginas
- » 3 CDs / 1 Libro



CONÉCTESE CON LOS MEJORES LIBROS DE COMPUTACIÓN



Esta obra está destinada a todos los desarrolladores que necesitan avanzar en el uso de la plataforma Adobe Flash y sacarle su máximo provecho. A través de los distintos capítulos, se explica cómo crear un framework de trabajo personalizado.

» DESARROLLO / DISEÑO
» 320 PÁGINAS
» ISBN 978-987-1857-00-5

LLEGAMOS A TODO EL MUNDO VÍA OCA * Y DHL **

MÁS INFORMACIÓN:

usershop.redusers.com

CONTÁCTENOS: +54 (011) 4110-8700 usershop@redusers.com

* SÓLO VÁLIDO EN LA REPÚBLICA ARGENTINA // ** VÁLIDO EN TODO EL MUNDO EXCEPTO ARGENTINA

USERS



El contenido de este libro fue publicado en el curso en fascículos Web Design

CONTENIDO

1 | DISEÑO Y TECNOLOGÍA

Flash y ActionScript / Interfaces / Organizar un proyecto

2 | HERRAMIENTAS DE TRABAJO

Adobe CS Web Premium / Herramientas / Illustrator

3 | TEXTOS

Texto clásico / Suavizado / HTML / CSS / Scroll Bar

4 | DIBUJOS Y LOADERS

Dibujo en Flash / Entornos 3D / Intros / Creación de un cargador simple / Optimización de sitios

5 | ANIMACIONES BÁSICAS

Axiomas / Classic Tween / Motion Tween / Motion Editor

6 | ANIMACIONES AVANZADAS

Efectos físicos / Tween Max y Tween Lite / Partículas

7 | IMÁGENES Y GALERÍAS

Incorporación y optimización / Filtros / Componentes

8 | FORMULARIOS

Tabulación y validación / Texto predictivo / Envío / CMS

9 | SONIDO

Uso y sincronización / Librerías / Visualización de sonido

10 | VIDEO

Formatos / Adobe Media Encoder / Streaming

11 | APIS EXTERNAS Y COMPLEMENTOS

YouTube / Google Maps / Facebook / Flash Catalyst

12 | SERVIDORES Y TECNOLOGÍAS WEB

Clientes y servidores / Protocolos / DNS / URL

13 | VIDA LABORAL

NIVEL DE USUARIO

PRINCIPIANTE

INTERMEDIO

AVANZADO

EXPERTO

FLASH CREACIÓN DE SITIOS WEB

Esta obra constituye una guía indispensable para diseñadores y estudiantes que quieran perfeccionarse en el diseño de sitios web, así como también para los aficionados que deseen aprender sobre Flash y AS3.

En sus capítulos, los expertos muestran las bases para la creación de sitios, al mismo tiempo que exponen las últimas tendencias en diseño y desarrollo web. En el comienzo del libro, realizaremos un repaso de los fundamentos de Flash y ActionScript, así como también de las principales herramientas de trabajo. Luego, comenzaremos a trabajar con la incorporación de textos, dibujos y loaders en el proyecto. Más adelante, aprenderemos a crear animaciones básicas y avanzadas para otorgar un carácter más creativo e interactivo a nuestro desarrollo. Además, veremos cómo manejar elementos multimedia y crear formularios desde cero.

En definitiva, una obra imprescindible para quienes deseen diseñar sitios funcionales y atractivos en Flash. Una forma de obtener una salida laboral segura en el creciente mercado de trabajo relacionado con Internet.



En RedUSERS.com encontrará una gran variedad de recursos y software relacionado, que le servirán como complemento al contenido del libro. Además, tendrá la posibilidad de estar en contacto con los editores, y de participar del foro de lectores, en donde podrá intercambiar opiniones y experiencias.

Si desea más información sobre el libro puede comunicarse con nuestro Servicio de Atención al Lector: usershop@redusers.com

FLASH: CREATING WEB SITES



This manual will help the reader to design and put together websites using the best tools included in Adobe CS. He will learn step by step how to plan and set the structure of each page and prepare the text and images.

ISBN 978-987-1857-40-1



SERS MANUALES USERS MANUALES USERS

RedUSERS.com SITIOS ATRACTIVOS CON ADOBE CREATIVE SUITE