



USERS

DISEÑO

CREACIÓN DE SITIOS WEB

VERSIÓN
2.0

**DISEÑO DE SITIOS ATRACTIVOS
CON ADOBE CREATIVE SUITE**

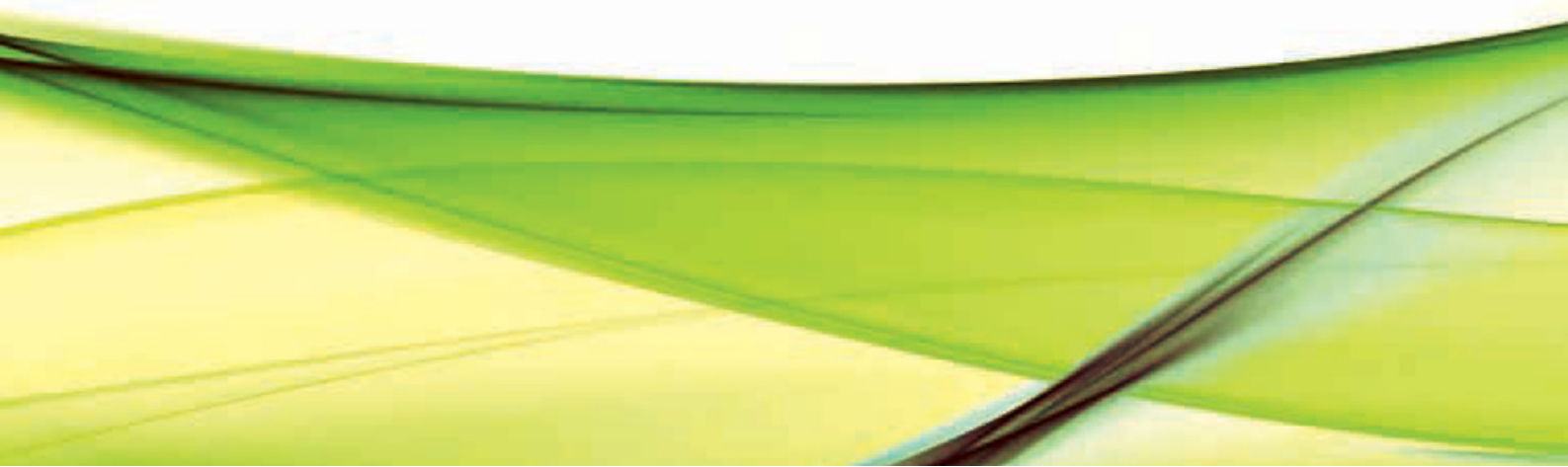
DREAMWEAVER, PHOTOSHOP E ILLUSTRATOR |
FUNDAMENTOS DEL DISEÑO Y LAS ÚLTIMAS
TENDENCIAS | ARMADO DE LA MAQUETA |
DESARROLLO DE SITIOS EN TECNOLOGÍA
XHTML Y CSS | TRATAMIENTO DE IMÁGENES

CREACIÓN DE SITIOS

WEB

VERSIÓN 2.0

DISEÑO DE SITIOS ATRACTIVOS
CON ADOBE CREATIVE SUITE





TÍTULO: Creación de sitios Web Versión 2.0

COLECCIÓN: Manuales USERS

FORMATO: 17 x 24 cm

PÁGINAS: 320

Copyright © MMX. Es una publicación de Fox Andina en coedición con Gradi S.A. Hecho el depósito que marca la ley 11723. Todos los derechos reservados. No se permite la reproducción parcial o total, el almacenamiento, el alquiler, la transmisión o la transformación de este libro, en cualquier forma o por cualquier medio, sea electrónico o mecánico, mediante fotocopias, digitalización u otros métodos, sin el permiso previo y escrito del editor. 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 IX, de MMX.

ISBN 978-987-1773-99-2

Creación de sitios Web Versión 2.0 / coordinado por Daniel Benchimol.

1a ed. - Buenos Aires: Fox Andina; Dalaga, 2011.

v. 219, 320 p. ; 24x17 cm. - (Manual Users)

ISBN 978-987-1773-99-2

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 TECNOLOGÍA



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 DHL**

* 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

Conoceremos cuáles son las características y habilidades que todo diseñador, programador o autodidacta debería sumar a su perfil profesional para convertirse en un experto del diseño y el desarrollo web.

▼ 02 HERRAMIENTAS DE TRABAJO

Analizaremos la suite Adobe Web Design Premium y elegiremos las herramientas que nos permitirán hacer realidad el sitio. Conoceremos la utilidad de Photoshop, Dreamweaver e Illustrator en el área web.

▼ 03 XHTML Y CSS: BÁSICO

Veremos los aspectos básicos del lenguaje HTML, así como también las principales características de XHTML. Estudiaremos los sectores básicos de cualquier página web.

▼ 04 XHTML Y CSS: TEXTO

Aprenderemos la forma de crear y utilizar las hojas de estilo. Estudiaremos las diferentes formas de escribir títulos y párrafos en Dreamweaver utilizando las etiquetas especiales para este tipo de elementos.

▼ 05 ENLACES

Estudiaremos los enlaces y su clasificación. Además, veremos las propiedades de estos. También analizaremos las clases especiales de CSS. Aprenderemos a dar estilos generales a los enlaces.

▼ 06 MAQUETADO

Conoceremos la técnica tableless, el arte de construir sitios sin recurrir a las tablas. Realizaremos la estructura básica de un sitio y aprenderemos a colocarle un fondo a nuestra web.

▼ 07 LISTAS

Aprenderemos todo sobre las listas: para qué sirven, su clasificación, cómo generarlas. Además, explicaremos las diferentes formas de personalizar las listas en pocos pasos.

▼ 08 IMÁGENES

Veremos la implementación de uno de los elementos más importantes de las páginas web: las imágenes. Comenzaremos conociendo cómo insertarlas mediante atributos específicos y exclusivos.

▼ 09 FORMULARIOS

Explicaremos cómo generar un formulario con imágenes e interacción entre los divs, con la ayuda de Dreamweaver. Además, veremos los conceptos básicos del lenguaje de servidor más conocido: PHP.

▼ 10 PROYECTO REAL I

Tomaremos medidas y definiremos el layout de nuestro sitio. Además, le daremos forma con CSS. Crearemos los menús de navegación, armaremos el footer y la página de inicio. También, generaremos un formulario de presupuesto.

▼ 11 PROYECTO REAL II

Realizaremos la segunda parte de la maquetación del sitio. Crearemos la página de servicios, el detalle, el portfolio y la página de contacto que incluye un mapa con la ubicación física del emprendimiento.

▼ 12 SERVIDORES Y TECNOLOGÍA WEB

Estudiaremos qué son los clientes y servidores. Además analizaremos los diferentes protocolos y el sistema de nombres de dominio. Registraremos nuestro sitio y lo subiremos a la web.

▼ APÉNDICE A VIDA LABORAL

Conoceremos las posibilidades laborales que nos ofrece el diseño web. Además, analizaremos las diferencias ser emprendedor, empleado y trabajador freelance. Aprenderemos a crear un portfolio efectivo.

▼ SERVICIOS AL LECTOR

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

PRÓLOGO

En menos de veinte años, Internet revolucionó por completo nuestras vidas y se transformó en un espacio virtual a través del cual nos comunicamos, consumimos productos y nos informamos. A medida que la Red de redes creció, los usuarios se fueron convirtiendo en expertos navegantes que comenzaron a demandar cada vez una mayor cantidad de soluciones basadas en la Web.

Estos requerimientos de los internautas generaron oportunidades de trabajo que fueron aprovechadas, en un comienzo, por profesionales con estudios relacionados con la programación. Sin embargo, cuando la apariencia de los sitios cobró casi tanta importancia como la funcionalidad, esos técnicos se quedaron sin respuestas para dar a los clientes y, entonces, recurrieron a los diseñadores gráficos.

Las herramientas de desarrollo y de diseño crecieron hasta permitir la creación de sitios con amplias funcionalidades, y con diseños sobresalientes y llamativos para los navegantes. En este nuevo mundo, podemos crecer profesionalmente sin realizar más inversión que la de nuestro capital intelectual. Para eso, deberemos adquirir los conocimientos técnicos que se convertirán en un negocio rentable cuando les sumemos una cuota de nuestra propia creatividad y buenas ideas.

Esta obra permite adquirir todos esos conocimientos, ya que propone un aprendizaje guiado por reconocidos profesionales que comparten con nosotros su amplia experiencia en cada una de las páginas del libro. Comencemos, entonces, el recorrido hacia el éxito en el diseño y en el desarrollo de sitios profesionales.

CONTENIDO

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

Adobe Illustrator CS5	068
Adobe CS5 Dreamweaver	070
La interfaz de Dreamweaver CS5	072
Actividades	074

▼ 01 DISEÑO Y TECNOLOGÍA 011

El diseñador Web	012
Evolución de Internet	014
Tecnologías de los sitios Web	016
Principios para un buen diseño Web	019
Evolución del diseño Web	022
Tipología de los sitios Web	026
Diseñar un portfolio	028
Coming soon page	030
Crear una coming soon page (paso a paso)	032
Un problema con buen diseño	034
Crear una página de error 404 (paso a paso)	036
La estructura de las páginas	038
Guía visual	042
Estructura y diseño de la página	044
Creación de un layout en Photoshop (paso a paso)	046
Teoría del color	052
Tipografía	055
Actividades	058

▼ 02 HERRAMIENTAS DE TRABAJO 059

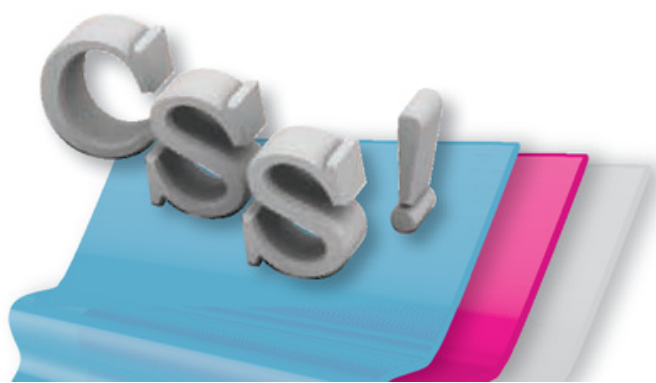
Adobe CS5 Web Premium	060
Herramientas para desarrollar HTML	062
Selección de las herramientas de diseño	064
Adobe Photoshop CS5	066

▼ 03 XHTML Y CSS: BÁSICO 075

¿Qué es HTML?	076
Estructura básica de una página: head y body	078
Etiquetas y atributos: definición	079
Dreamweaver, primer HTML (paso a paso)	080
Elementos HTML	082
Atributos	084
Nuestras primeras etiquetas HTML (paso a paso)	088
CSS cascading style sheets	090
Nuevas características de CSS3	092
Selectores	094
Tabla de estilos	097
CSS accesible	098
Actividades	100

▼ 04 XHTML Y CSS: TEXTO 101

Estilos CSS para textos	102
Hojas de estilo de Dreamweaver (paso a paso)	104
Texto: párrafos y títulos	106



Marcado básico y caracteres especiales	109
Escribir títulos y párrafos (paso a paso)	112
Actividades	114

▼ 05 ENLACES 115

Hipertextos	116
Enlaces relativos y absolutos	118
Enlaces de Dreamweaver	120
CSS: Clases especiales	122
Enlaces básicos y avanzados	124
Estilos para los enlaces en Dreamweaver (paso a paso)	126
Actividades	128

▼ 06 MAQUETADO 129

Tableless	130
Estructura básica de un sitio	132
Posicionamiento de elementos	133
Armaz la estructura de un sitio (paso a paso)	137
Modelo de cajas	142
Diagrama gráfico del modelo de cajas	143
Unidades de medida	144
Propiedades de tamaño	146
Margen y relleno	148
Bordes (CSS)	150
Tendencias en el diseño de fondos	152
Propiedades y aplicación de fondos	154
El background de nuestro sitio (paso a paso)	156
Actividades	158

▼ 07 LISTAS 159

Listas: definición	160
Listas ordenadas y no ordenadas	162
¿Dónde utilizamos listas?	164
Personalizar listas (paso a paso)	166
Creación de menús de navegación	168
Crear un menú de navegación (paso a paso)	170
Menú desplegable con CSS y listas (paso a paso)	174
Actividades	176

▼ 08 IMÁGENES 177

Imágenes en HTML	178
Tipos de imágenes	180
Optimización de imágenes para la Web	181
Optimización de imágenes para la Web (paso a paso)	182
Mapa de imágenes	184
Mapa de imágenes (paso a paso)	186
Propiedades de las imágenes en CSS	188
CSS sprites	189
Sprites en CSS	191
CSS sprites en menús (paso a paso)	192
Actividades	196

▼ 09 FORMULARIOS 197

Formularios	198
Elementos de los formularios	201
Formulario completo (paso a paso)	208
PHP Fundamentos	212
Hacer funcionar los formularios	216
Actividades	218

▼ 10 PROYECTO REAL I 219

Presentación del sitio	220
Tomar medidas y definir el layout (paso a paso)	222
Dar forma al layout con CSS (paso a paso)	224
Menús de navegación (paso a paso)	227
Armado del footer (paso a paso)	229
Formulario de presupuesto (paso a paso)	232
Crear la página de inicio (paso a paso)	235
Actividades	240

▼ 11 PROYECTO REAL II 241

Las páginas de servicio (paso a paso)	242
Página portfolio (paso a paso)	245
Página de detalles (paso a paso)	250
Página de contacto (paso a paso)	252
Actividades	260

▼ 12 SERVIDORES Y TECNOLOGÍA WEB 261

Cientes y servidores	262
Protocolos	266
DNS: sistema de nombres de dominio	267
URL y tipos de dominio	268
El sistema Web y su funcionamiento	269
Elegir un nombre de dominio	270
¿Cómo registrar un dominio?	272

Registrar un dominio (paso a paso)	273
Servicios de hosting	276
Subir nuestra web a Internet	278
Subir un sitio al servidor (paso a paso)	280
Actividades	282

▼ APÉNDICE A VIDA LABORAL 283

La salida laboral	284
Workflow	288
¿Emprendedor, empleado o freelance?	290
Freelance: ¿Cómo iniciarse?	292
Freelance: conseguir clientes online	294
Freelance: administrar el tiempo	296
Emprendedor, ¿yo?	300
Planificar un portfolio efectivo	302
Oportunidad en las ideas	304
Determinar el precio del trabajo	306
Modelo de negocio	308
Modelo de negocio final	310
Actividades	312

▼ APÉNDICE A VIDA LABORAL 313

Índice temático	314
Catálogo	317



INTRODUCCIÓN

La era digital generó nuevas oportunidades de trabajo. Este libro es 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. Nos permitirá una mejor inserción en el mercado laboral, tanto para armar el sitio de nuestro propio emprendimiento, como para postularnos a un nuevo trabajo o crear nuestro estudio de diseño web.

Realizar un sitio web, si tenemos la información adecuada, resulta una tarea sencilla. La idea detrás de este libro es proporcionar una guía práctica para diseñar un sitio web con Dreamweaver y la ayuda de herramientas gráficas, tales como Photoshop e Illustrator. Esta obra nos permitirá dominar las principales tecnologías empleadas, tanto para el diseño, como para el desarrollo de sitios web, como CSS y HTML. En sus páginas encontraremos un recorrido por todos

los elementos teóricos fundamentales en esta profesión y veremos la forma de llevar estos conceptos a la práctica.

En esta oportunidad, los mejores expertos en el tema exponen los fundamentos de este arte y nos enseñan todos los secretos para que triunfemos ante cualquier desafío que nos toque encarar, convirtiéndonos en excelentes profesionales de la Web. Este aprendizaje sentará bases sólidas y nos llevará a diseñar y desarrollar sitios web profesionales muy superiores a los que podría realizar un hobbysta o un diseñador no especializado, distinguiendo y haciendo resaltar nuestro trabajo.

La obra está dirigida a aquellos que quieran diseñar sitios, incluso a quienes no cuentan con educación formal previa en la materia. ¡Es nuestro deseo que sea una herramienta de ayuda para la creación de sitios web de carácter profesional!

Diseño y tecnología

- ▶ EL DISEÑADOR WEB
- ▶ EVOLUCIÓN DE INTERNET
- ▶ TECNOLOGÍAS DE LOS SITIOS WEB
- ▶ PRINCIPIOS PARA UN BUEN DISEÑO WEB
- ▶ EVOLUCIÓN DEL DISEÑO WEB
- ▶ TIPOLOGÍA DE LOS SITIOS WEB
- ▶ DISEÑAR UN PORTFOLIO
- ▶ COMING SOON PAGE
- ▶ UN PROBLEMA CON BUEN DISEÑO
- ▶ CREAR UNA PÁGINA DE ERROR 404
- ▶ LA ESTRUCTURA Y EL DISEÑO DE LAS PÁGINAS
- ▶ CREACIÓN DE UN LAYOUT EN PHOTOSHOP
- ▶ TEORÍA DEL COLOR
- ▶ TIPOGRAFÍA



EL DISEÑADOR WEB

Para comenzar, veremos cuáles son las características y habilidades que todo diseñador, programador o autodidacta debería sumar a su perfil profesional para convertirse en un experto del diseño y el desarrollo web.

Actualmente, es imposible referirse a un solo tipo de perfil de diseñador web y mucho menos a un solo tipo de persona. La creación de sitios web se ha convertido en una disciplina en la que se plantean proyectos que requieren de una conjunción de conocimientos **técnicos** (como lenguajes de programación y servidores), **conceptuales** (usabilidad y accesibilidad, entre otros) y **estéticos** (como percepción y legibilidad).
Atrás quedó la época en la que para realizar un sitio web alcanzaba con tener un mínimo conocimiento de HTML y de tratamiento de imágenes. Ahora es importante que un diseñador sepa cómo planificar un sitio web, y que tenga presentes los aspectos técnicos de diagramación y de estructura al momento de ponerse a diseñar.

LAS HERRAMIENTAS DE TRABAJO

Para realizar su trabajo, un profesional del diseño y el desarrollo web necesita herramientas de software, que debe utilizar con criterio para sacar el máximo provecho de ellas, además de conocer los fundamentos de su tarea. Debe ser un usuario intermedio de PC o Mac, utilizar los navegadores web y el correo electrónico, y estar al tanto de la existencia y el funcionamiento de las redes sociales. Como existen muchas herramientas de diseño en el mercado, lo importante es que hay un estándar muy bien definido de lo que se usa. Basta con echar una mirada a las solicitudes de trabajo que aparecen en los medios para saber qué es lo que se pretende hoy en día de un diseñador web. Un diseñador web debe tener conocimientos sobre **HTML** y **CSS**, trabajar

EL ÉXITO DE UN DISEÑADOR WEB DEPENDE DE SUS CONOCIMIENTOS TÉCNICOS, ESTÉTICOS Y CONCEPTUALES.

con **Dreamweaver** para armar los sitios, y poder armar por lo menos un banner en **Flash**. También debe utilizar las herramientas principales de diseño, como **Photoshop**, **Illustrator** y **Fireworks**, que permiten trabajar en forma profesional y con la mejor calidad. Además, resultarán importantes los conocimientos de animación así como también la programación en **ActionScript 3.0**.

LAS HABILIDADES DEL DISEÑADOR

Aunque es posible decir que un buen diseñador web es aquel que sabe utilizar las herramientas con las que desarrolla sus proyectos, éste es sólo un aspecto técnico que no resulta fundamental para tener éxito. Las habilidades principales del diseñador serán:

Buen criterio y sentido de la estética: conocer los principios del diseño tradicional y los específicos del diseño web para crear sitios agradables.

Elección de las herramientas: elegir la herramienta ideal según el tipo de



En la Web tendremos muchos sitios como referentes estéticos. Uno muy novedoso y atractivo es pikaboo (www.pikaboo.be).

proyecto web que deba realizar y saber utilizarla al máximo.

Conocer el ambiente: estar inmerso en el mundo virtual de Internet y conocer las últimas tendencias y tecnologías.

Estar actualizado: estar al tanto de todo lo que influye en el medio, como la aparición de nuevas técnicas de diseño, cambios en las tendencias y en los estilos, o nuevos dispositivos y utilidades que simplifiquen el trabajo.

Entender los principios básicos de funcionalidad en un sitio: de esta manera, puede ofrecer la mejor solución a cada proyecto que realice.

Capacidad de organización: esto es fundamental para poder llevar a cabo todo el proyecto de forma ordenada y prolija, sin sobresaltos.

Facilidad para la relación interpersonal: comunicarse de forma clara con los clientes y los colegas resulta fundamental para el éxito de los proyectos, sobre todo, cuando se trabaja de forma freelance y se tercerizan tareas.

TAREAS DEL DISEÑADOR

Para definir de forma más clara el perfil del diseñador web, podemos realizar una división de sus tareas sobre la base de las diferentes etapas en las que se podría ver involucrado dentro de la realización de un proyecto web.

Si pensamos en la concepción del diseño de un sitio, es imprescindible que sea capaz de realizar un **resumen** o **brief** de trabajo, en donde pueda determinar las necesidades estéticas y técnicas del proyecto, y a partir de esto, comenzar a diseñar. En este momento, será necesario que disponga de herramientas para confeccionar el diseño. Si el sitio es **HTML**, puede trabajar con aplicaciones como Photoshop, Illustrator o Fireworks, que ya mencionamos. En cambio, si el sitio es

en **Flash**, deberá determinar la mejor forma de crear un diseño optimizado, utilizar las herramientas gráficas del programa y conocer su lenguaje de programación, **ActionScript**. Para las etapas posteriores al diseño de los sitios, es necesario que sepa maquetar en HTML y utilizar CSS, y para estas tareas Dreamweaver resulta una herramienta

esencial. Por último, es importante que tenga conocimientos sobre la forma de alojar un sitio en un servidor, ya sea por medio de Dreamweaver o mediante un programa especial para **FTP (File Transfer Protocol)**, como por ejemplo, FileZilla. Como vemos, un diseñador necesita contar con muchos conocimientos para poder crear un sitio de calidad. ◀



ACTUALIZARSE

A la hora de mantenernos actualizados, podemos recurrir a los siguientes sitios:

- ▶ **TheFWA (www.thefwa.com):** uno de los sitios más conocidos, en donde premian a las mejores creaciones en Flash.
- ▶ **Best Web Gallery (www.bestwebgallery.com):** sitio de referencia para conocer los mejores diseños XHTML y CSS.
- ▶ **Ateneu Popular (www.ateneupopular.com):** blog en español con noticias, recursos y tutoriales.
- ▶ **Web Developer's Handbook (www.alvit.de/handbook):** excelente sitio con cientos de links de recursos, noticias e inspiración.



EVOLUCIÓN DE INTERNET

Para comprender en detalle el avance del diseño de sitios, nada mejor que una línea de tiempo que nos muestra cómo evolucionaron los navegadores web y los buscadores, dos piezas clave en el desarrollo de Internet.

Cada vez que surge una nueva tecnología o aparece una versión más moderna de alguna ya existente, todos los elementos relacionados con ella

se actualizan. De esta manera, los navegadores incluyen soporte para las novedades y, luego, los diseñadores recrean los sitios incorporando en ellos la nueva tecnología. ◀

BUSCADORES



► **Google.com:** nacido como motor de búsqueda, Google sólo exhibe un mensaje de bienvenida y un par de enlaces marginados a la izquierda.



- 1996

► **Yahoo.com:** primera versión, donde se diferencian tres sectores: cabezal, pie y cuerpo de página. Su contenido principal, diagramado a una sola columna alineada a la izquierda, se compone de textos y enlaces sin estilos, y de unos pocos iconos de aspecto sencillo.

- 1997

► **Yahoo.com:** la diagramación pasa a ser en dos columnas; surge un nuevo estilo para algunos textos, y la página se centra en la ventana del navegador.



- 1998

Welcome to Google
Search for anything
Google will find it for you

- 2000

► **Google.com:** el sitio vuelve a ser liviano visualmente. Los enlaces de contenido institucional se reducen, se eliminan los colores de fondo, y el logo sufre una suave metamorfosis.

- 2003

► **Yahoo.com:** la incorporación de soporte CSS en algunos navegadores trae mejoras en la calidad del diseño. Aumenta la cantidad de contenido, la diagramación se hace más rica y se suman las primeras fotografías al portal.

NAVEGADORES

1994



OPERA

► Es un navegador creado por la empresa noruega Opera Software. En la actualidad se encuentra en la versión 10.54.

1994



NETSCAPE NAVIGATOR

► Netscape fue el primer navegador web. Su última versión es la 9.0.0.6, lanzada en febrero de 2008.

1995



INTERNET EXPLORER

► Es el navegador desarrollado por Microsoft para Windows. Hoy en día existe la versión 8.0.

2003



SAFARI

► Es el navegador de Mac desarrollado por Apple Inc. En junio de 2010 se lanzó la versión 5.0.

2004



FIREFOX

► Navegador multiplataforma y de código abierto. Su última versión es la 3.6.6.

2008



GOOGLE CHROME

► Es el navegador desarrollado por Google. Es de código abierto y su última versión es la 5.0.375.86.



► **Google.com:** incorpora enlaces a las nuevas herramientas y recursos disponibles. Mejora los servicios de Gmail, Google Maps, Adwords y AdSense. Además, cambia la búsqueda instantánea, y la posibilidad de elegir una imagen para la home del buscador.

- 2005

► **Google.com:** mantiene sus principales características, incluyendo la personalización del logo, pero economiza recursos al reemplazar las pestañas por enlaces sin estilo.



- 2007

► **Yahoo.com:** el portal incorpora una pre-home, indicando las versiones de navegadores que soporta.

- 2008

► **Google.com:** incorpora enlaces a las nuevas herramientas y recursos disponibles. No aplica grandes cambios de estilo, más allá de las mejoras en los botones y en la definición del logo.

- 2010

► **Yahoo.com:** el sitio actual está optimizado para resoluciones de 1024 x 768, y es rico en diagramación, diseño, implementación de estilos y uso de JavaScript.



TECNOLOGÍAS DE LOS SITIOS WEB

Aunque casi todos los sitios que navegamos parecen iguales, es posible encontrar diferencias en la forma en la que están desarrollados, ya que utilizan distintas tecnologías o técnicas para cumplir sus objetivos.

Es habitual que a diario naveguemos por diversos sitios para revisar nuestro correo electrónico, leer las noticias, o hacer alguna compra o pago virtual. A simple vista, todos los sitios que encontramos parecieran estar hechos de la misma manera, aunque si hacemos un análisis en detalle, descubriremos que en ellos intervienen diversas

tecnologías que se combinan de diferentes modos para lograr las funcionalidades y prestaciones que brindan al usuario. Conozcamos las principales.

LENGUAJES DE EJECUCIÓN DEL LADO DEL USUARIO

En estos sitios, cuando el usuario accede a una página, los archivos que la compo-

nen son **descargados e interpretados** directamente por el navegador (Firefox o Internet Explorer). Pueden estar creados con las siguientes tecnologías:

► **HTML y XHTML:** la mayoría de los sitios por los que navegamos se crean con estos lenguajes. Estas páginas se componen por una estructura



TIPO DE SITIO	HTML	CSS	FLASH/ ACTIONSCRIPT	PHP/ASP
► SITIO ESTÁTICO	Usado sólo para incluir el archivo Flash.	Usado sólo para eliminar los márgenes.	Sitio creado totalmente en Flash y ActionScript.	No utilizado.
► SITIO DINÁMICO	Usado en toda la estructura, comentarios y videos relacionados.	Usado para darle todo el estilo visual.	Usado para el player de video.	Usado para crear las páginas en forma dinámica.
► BLOG	Usado en toda la estructura.	Usado para darle todo el estilo visual.	No utilizado.	Usado para crear las páginas en forma dinámica.
► E-COMMERCE	Usado en toda la estructura.	Usado para darle todo el estilo visual.	No utilizado.	Usado para crear las páginas en forma dinámica.
► PORTAL DE NOTICIAS	Usado en toda la estructura.	Usado para darle todo el estilo visual.	Utilizado sólo en banners publicitarios.	Usado para crear las páginas en forma dinámica.
► APLICACIÓN ONLINE	Usado en toda la estructura.	Usado para darle todo el estilo visual.	No utilizado.	Usado para crear las páginas en forma dinámica.
► BUSCADOR	Usado en toda la estructura.	Usado para darle todo el estilo visual.	No utilizado.	Usado para crear las páginas en forma dinámica.
► FORO	Usado en toda la estructura.	Usado para darle todo el estilo visual.	Utilizado sólo en banners publicitarios.	Usado para crear las páginas en forma dinámica.
► WIKI	Usado en toda la estructura.	Usado para darle todo el estilo visual.	No utilizado.	Usado para crear las páginas en forma dinámica.
► NEWSLETTER	Usado en toda la estructura.	Usado para darle todo el estilo visual.	No utilizado.	No utilizado.
► EXPERIMENTAL	Usado sólo para incluir el archivo Flash.	Usado sólo para eliminar los márgenes.	Sitio creado totalmente en Flash y ActionScript.	No utilizado.
► 3D	Usado sólo para incluir el archivo Flash.	Usado sólo para eliminar los márgenes.	Sitio creado totalmente en Flash y ActionScript.	No utilizado.

Tecnologías que se utilizan para desarrollar cada tipo de sitio.

Aquí se ve el código con el que está conformada parte de una página, con etiquetas que indican títulos principales, párrafos, links e imágenes.

```
<div id="Portfolio, Diseñando Web">
<div id="trabajos">
<div class="work" id="workA">
<div>Diseño web, Auto Neumen</div>
<div>
<div><a href="http://www.diegopazera.com.ar/neumen">Ver online</a></div>
<div><a href="portfolioWeb/sitiosWebNeumen.html">Ver detalles</a></div>
</div>
</div>
</div>
<div class="clear"></div>
<div>Desarrollo integral de diseño web y programación del sitio web, tanto se realiza un ajuste en la marca para la imagen corporativa y un nuevo diseño en la comunicación en vía pública.</div>
<div>
<div class="work" id="workB">
<div>Diseño web, New Fast Copy</div>
<div>
<div><a href="http://www.newfastcopy.com.ar">Ver online</a></div>
<div><a href="portfolioWeb/sitiosWebNewFastCopy.html">Ver detalles</a></div>
</div>
</div>
</div>
<div class="clear"></div>
<div>Se realiza tanto el diseño web como la programación del sitio web. Se trata de una empresa de servicios empresariales de primera línea y con gran experiencia y velocidad, todos valores que fueron contemplados a la hora de diseñar la interactividad y la interfase.</div>
</div>
```

de **etiquetas** que tienen significados semánticos (títulos importantes, párrafos o imágenes) para que los buscadores (como Google o Bing, por ejemplo) vinculen un determinado texto existente en la página al significado semántico que le da la etiqueta que lo contiene.

► **JavaScript/AJAX:** es un lenguaje de

programación usado para generar interfaces de usuario mejoradas, con más interactividad. Las sentencias y funciones de este código pueden estar en el mismo archivo HTML o en un archivo externo. AJAX es una técnica que se usa vinculando JavaScript a XML. JavaScript requiere la información alojada en el servidor en formato XML

para generar cambios o agregar funcionalidades en la página, sin interferir con su visualización ni comportamiento. ►



YOUTUBE

En YouTube podemos ver que el reproductor de video y sus controles están creados con Flash y ActionScript. El resto del entorno visual es HTML, pero tanto la información de los videos relacionados como los comentarios dejados por los usuarios están almacenados en una base de datos, lo que indica que se trata de un sitio creado con PHP o ASP. Finalmente, las funcionalidades que proveen los botones de cantidad de reproducciones o cantidad de videos están generadas con JavaScript y, tal vez, con AJAX.

JAVASCRIPT/ AJAX

BASE DE DATOS

URL DE EJEMPLO

No utilizado.	No utilizada.	www.metropolis-media.com.pl/2007
Usado para crear la interactividad.	Usada para leer la información respectiva de los videos y los comentarios.	www.youtube.com
Usado para crear la interactividad.	Usada para almacenar la información de los posts y comentarios.	http://es.wordpress.org
No utilizado.	Usada para almacenar los datos de los usuarios, productos e imágenes.	www.officenet.com.ar/basket.asp
Usado para crear la interactividad.	Usada para almacenar la información de los posts y comentarios.	www.lanacion.com.ar
Usado para crear la interactividad.	Usada para almacenar los datos creados por el usuario.	http://docs.google.com
Usado para crear la interactividad.	Usada para almacenar los datos indexados de las páginas.	www.google.com
No utilizado.	Usada para almacenar la información de los posts y comentarios.	www.psicofxp.com
No utilizado.	Usada para almacenar la información de los posts y comentarios.	www.wikipedia.org
No utilizado.	No utilizada.	***
No utilizado.	No utilizada.	www.davidlindseywade.com/#/portfolio/portfolio/19
No utilizado.	No utilizada.	www.vatican.va/various/cappelle/sistina_vr/index.html



HTML VS. XHTML

Aunque lo veremos en detalle más adelante, la diferencia entre HTML y XHTML radica, fundamentalmente, en que el último requiere una sintaxis más estricta en comparación con el primero, en el que pueden omitirse las etiquetas de cierre, por ejemplo, sin que esto provoque fallas.

► **Flash y ActionScript:** Flash se asomó al mundo de Internet como un programa destinado a generar vistosas animaciones vectoriales que se podrían incorporar en las páginas. En su evolución, desarrolló un potente lenguaje de programación que, hoy en día, permite crear sitios y aplicaciones web dinámicas, con conectividad hacia otros lenguajes y plataformas. También brinda la posibilidad de generar entornos con gran interactividad e impacto visual, además de manejar audio y video.

ALMACENAMIENTO DE DATOS

Cuando se genera un sitio web dinámico, es necesario contar con un repositorio de información. Para esto, disponemos de algunas alternativas que vemos a continuación:

► **XML (Extensible Markup Language):** es un lenguaje de sintaxis similar a HTML, pero no tiene etiquetas predefinidas con significados particulares, sino que éstas son establecidas por el usuario de acuerdo con la información que necesite almacenar. Suele usarse vinculado a Flash y ActionScript.

► **Bases de datos:** una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En el entorno web, las bases más utilizadas son SQL (SQL Server pertenece a Microsoft) y MySQL (open source). Se las usa para almacenar los datos que componen un sitio web, tales como textos e imágenes, entre otros. Suelen usarse vinculadas a ASP y PHP.

LENGUAJES DE EJECUCIÓN DEL LADO DEL SERVIDOR

Son lenguajes que ejecuta el servidor para crear una página cuando el usuario accede a un sitio. Es decir, buscan la información en una base de datos y, tomándola como referencia, generan archivos HTML que serán finalmente visualizados por el usuario. Conozcamos las alternativas más difundidas:

► **PHP:** es un lenguaje de programación al que pueden agregarse diversas librerías para otorgarle funcionalidades y

usarse sin costo alguno.

Es compatible con la mayoría de los servidores web y sistemas operativos, aunque suele implementarse en entornos Linux y con bases de datos MySQL, lo cual evita por completo los costos de licencias.

► **ASP:** es un lenguaje de programación creado por Microsoft, que se ejecuta sólo en servidores Windows. Su funcionamiento es igual al de PHP pero se utiliza, en general, vinculado a bases de datos SQL. La desventaja que presenta es que no es gratuito y que debe implementarse en un entorno 100% Windows, que tampoco es gratuito.

APLICACIÓN DE ESTAS TECNOLOGÍAS

Un caso en el que podemos ver muchas de estas tecnologías vinculadas es **YouTube**. En él se utiliza un reproductor con funciones especiales de video, creado en **Flash**, y en la interfaz de la página se pueden postear nuevos videos, crear perfiles y dejar comentarios. También encontramos sitios experimentales que generan nuevas situaciones de interacción y de experiencia con el usuario, al mismo tiempo que buscan un gran impacto visual sin limitaciones en cuanto al diseño. En ellos, tanto la interfaz como las situaciones interactivas son creadas en Flash y ActionScript de forma estática. Además, Flash también puede leer lenguajes de marcado (como XML) y bases de datos para generar sitios dinámicos. ◀

Wikipedia es un entorno colaborativo generado en PHP, donde los usuarios crean la información, que es almacenada en bases de datos y, luego, mostrada ante el requerimiento de cada persona.



Este sitio de alto impacto visual permite tener interactividad tanto con el mouse como con el teclado.



PRINCIPIOS PARA UN BUEN DISEÑO WEB

El diseño web posee características particulares que debemos tener en cuenta para obtener, como resultado, una interfaz estéticamente agradable y funcional.

A la hora de diseñar un sitio, es conveniente tener presentes algunas recomendaciones que harán que nuestro trabajo sea efectivo en cumplir su objetivo. A continuación, presentamos varios consejos para tener éxito o, al menos, para mejorar las decisiones que tomemos al crear el sitio.

EL PUNTO DE TENSIÓN

Toda pieza comunicacional presenta diversos elementos, niveles de lectura y jerarquías. Se generan relaciones entre esos componentes para guiar al usuario, llevándolo por la información que queremos que vea y generando puntos de tensión para hacer foco en esos elementos. Algunos de los recursos que podemos utilizar para trazar un recorrido son los siguientes:

► **Contraste:** la diferenciación, cromática y de tamaño, es una manera muy efectiva de conducir la mirada.

LOS ELEMENTOS GRÁFICOS NOS AYUDAN A INDICARLE AL USUARIO EL RECORRIDO QUE DEBE HACER DENTRO DE NUESTRO SITIO.

► **Color:** utilizando el color, podemos generar muchas conductas en el usuario y guiar su atención. Por ejemplo, si la palabra es azul, el navegante quizá la interprete como un vínculo. En cambio, si hay un texto rojo, seguramente lo leerá con mayor precaución.

► **Elementos visuales:** el uso de elementos tales como viñetas (bullets), recuadros y destacados facilita la lectura y conduce la mirada.

► **Posición:** el recorrido del usuario comienza por el logo y, luego, continúa

según el camino que tracemos. Entonces, si queremos que algo se vea en segundo plano, no deberemos ubicarlo en el pie de la página (footer), sino en un lugar superior.

MENOS ES MÁS: AIRE, ESPACIOS Y SIMPLICIDAD

Al diseñar una pieza comunicacional, debemos evitar llenar todos los espacios. El ojo necesita tener un descanso y no podemos bombardearlo con un sitio que tenga imágenes en todos lados, ya que el usuario no sabría por dónde comenzar a mirar. Pensemos en las sensaciones que experimentamos cuando pasamos por una vidriera y vemos una sola prenda en exhibición. Probablemente, tengamos la impresión de que se trata de una marca exclusiva. De algún modo, los espacios con pocas cosas parecen más elegantes, llaman más la atención. Por eso, es conveniente utilizar sólo contenidos relevantes para que los usuarios hagan foco en ellos.



Existen varias formas de destacar los elementos principales, como el tamaño, el color y la ubicación, entre otras opciones.

NAVEGACIÓN CLARA

La claridad del menú de navegación es fundamental, porque el usuario debe encontrar de forma rápida y sencilla lo que está buscando. Todos nos hemos sentido perdidos en algún sitio mientras intentábamos descubrir el teléfono de contacto. Esto hace que, luego de mover el puntero velozmente por toda la página, cerremos el navegador con un sentimiento de frustración por no haber hallado lo que buscábamos. Esta situación es, precisamente, lo contrario de lo que queremos que experimenten nuestros usuarios. Por eso, debemos intentar darles toda la información sin que tengan que hacer más de dos clics.

Los **botones** representan gráficamente los links del menú de navegación, por lo que deben ser fáciles de localizar e interpretar. En estos casos, lo obvio y predecible es lo más recomendado. Otro elemento que podemos utilizar son las **brújulas**, que funcionan como los letreros que vemos en los grandes comercios con indicaciones del estilo "Usted está Aquí". Éste es el objetivo del uso de **menús rastros o identificación de secciones** que solemos encontrar en los grandes sitios de Internet: nos muestran claramente cuál es la sección en la que estamos o el recorrido que hemos hecho para llegar hasta allí. Los menús rastro son importantes para guiar al usuario en el recorrido del sitio, sobre todo, cuando tiene mucha cantidad de páginas y secciones.

MÓDULO

CATEGORÍA

TÍTULO

BAJADA



La grilla constructiva permite organizar la información de manera ordenada y prolífica.

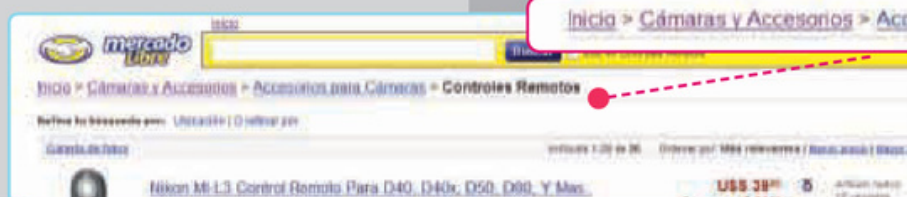
ES CONVENIENTE QUE
LOS **USUARIOS OBTENGAN
TODA LA INFORMACIÓN
CON SÓLO HACER DOS CLICS.**

GRILLA

Éste es uno de los elementos que, aun de forma inconsciente, conocemos y hemos visto en sitios y publicaciones. Es más, tal vez lo usemos incluso sin saberlo, ya que todo diseño presupone una **grilla constructiva**. Esta grilla es utilizada en los medios gráficos, como revistas y diarios, en donde la cantidad de información es muy amplia; poco a poco, también está cobrando importancia en la Web. El sentido de la grilla es mantener organizada la información, otorgando legibilidad y orden para una mejor lectura. Esto no significa que el diseño



Home > People > All Members



Inicio > Cámaras y Accesorios > Accesorios para Cámaras > Controles Remotos

Los menús rastro son importantes para guiar al usuario en el recorrido del sitio, sobre todo, cuando tiene mucha cantidad de páginas y secciones.

quedará cuadrado y aburrido, sino, por el contrario, que tendrá un valor agregado en cuanto al orden y la estructura.

La **grilla constructiva** permite organizar la información de manera ordenada y prolja.

En el ejemplo de grilla constructiva podemos apreciar el **esqueleto** del sitio web y ver que está compuesto por módulos. Cada módulo está determinado por un alto y un ancho y, a partir de su agrupación, se definen nuevas estructuras. Por ejemplo, el ancho de las columnas está formado por dos módulos. También podemos observar que en las notas se mantiene constante una estructura, donde aparece la categoría a la que pertenece la nota, el título y una pequeña bajada. Estos elementos mantienen siempre la misma letra, el mismo tamaño y el mismo color.

ESTÉTICA, TEMÁTICA Y TARGET

Nadie pensaría en vestirse con remera y bermuda para ir a un casamiento. Y así como creemos que ciertas cosas son más adecuadas para determinadas circunstancias, lo mismo sucede con los elementos que podemos hallar en el diseño de una interfaz. Por ejemplo, no encontraremos un periódico que utilice tipografías manuscritas para los textos de las notas.

En todos los diseños que hagamos, es importante saber qué queremos comu-



ZONAS CALIENTES

Jakob Nielsen, Ingeniero de interfaces, realizó investigaciones sobre los contenidos más vistos dentro de un sitio web. Concluyó en denominar como forma **térmica** a la presentación de los contenidos, marcando las zonas más vistas como **zonas calientes**. Utilizó colores rojo y amarillo para indicar lo más visto, y azul y gris para lo menos visto. Como resultado, las zonas más vistas forman una especie de letra F o triángulo sobre la estructura general del sitio, lo que confirma que las mejores ubicaciones para los contenidos más importantes son esas. Podemos leer más información en www.useit.com.

nicar y a quién lo dirigiremos. Teniendo en claro estas respuestas, podemos segmentar y definir el **target** de usuarios a los que apuntamos. De esta forma, será mucho más fácil seleccionar los elementos que resulten más apropiados para definir la identidad de la empresa o el producto que estamos mostrando. Cabe aclarar que decimos **más apropiados** porque, en realidad, no hay una regla que establezca para qué se debe usar una determinada tipografía o color. El diseño es subjetivo, y el concepto de belleza no es igual para todos; pero sí hay pautas que están muy arraigadas en la sociedad y que pueden ayudarnos a tomar algunas decisiones. Por ejemplo, sabemos que el rojo puede sugerir peligro y que el color negro se utiliza para luto, y podemos observar que los locales de comida rápida usan, generalmente amarillos y rojos para la decoración.

COLORES

El uso del color aplicado de forma consciente por diseñadores gráficos profesionales es un arma muy poderosa que provoca sensaciones y tiene un efecto psicológico en la conducta de la gente. Además, por supuesto, hay connotaciones culturales que, en ciertos casos, refuerzan el mensaje que los colores transmiten. Por ejemplo, los productos con packaging verde y blanco son light, los locales con colores rojo y azul son supermercados, y el rojo indica peligro. Así, a través del uso del color, podemos generar un punto de tensión en el sitio, llevando la mirada del lector hacia donde queremos que mire.

MISCELÁNEAS Y FORMAS

Los pequeños elementos que sirven como detalles en nuestros diseños deben ser coherentes con la estética general del sitio, ya que las formas también transmiten sensaciones. Si imaginamos un diseño para vender productos de cosmética para mujeres, sería más apropiado emplear líneas curvas de fondo, en vez de figuras geométricas. A modo de conclusión, vale repetir que el diseño es subjetivo, y que cada persona tiene sensaciones y vivencias distintas, por lo que las percepciones difieren en todos nosotros. Teniendo en cuenta esto y las recomendaciones anteriores, podremos lograr que nuestros sitios tengan un diseño efectivo y que cumplan sus objetivos principales: comunicar claramente, y provocar en los usuarios las reacciones previstas y deseadas.



En esta imagen podemos ver otro ejemplo de grilla constructiva.



EVOLUCIÓN DEL DISEÑO WEB



Para comenzar nuestro recorrido, es importante conocer qué significaba un sitio web hace 20 años y qué representa hoy. Esto nos ayudará a comprender el estado actual del diseño y del desarrollo web.

Como usuarios de Internet, navegamos decenas de sitios por día con el objetivo de revisar nuestro correo, informarnos, compartir imágenes, comprar productos, contar lo que estamos haciendo o, simplemente, divertirnos. Hemos aprendido a utilizar estos sitios casi sin darnos cuenta, haciendo uso de interfaces que, en general, podemos entender y dominar sin grandes dosis de esfuerzo o atención. Los elementos que hoy componen una página web y su diseño han evolucionado a través del tiempo. En sus comienzos, surgieron gracias al uso de metáforas basadas en el cine, la televisión, los libros o las galerías de arte, del mismo modo en que anteriormente los sistemas operativos tomaron la metáfora del **escritorio**. El uso de estos recursos ayudó a los usuarios a familiarizarse con las funcionalidades de estos productos interactivos y marcó el inicio de una disciplina que, luego, se convertiría en un trabajo interdisciplinario, frenético y lleno de inventiva.

UN RECORRIDO POR LA HISTORIA DEL DISEÑO WEB

El diseño web comenzó a dar sus primeros pasos en la década del '90 y hoy, a casi 20 años de su nacimiento, podemos decir que es una disciplina que ha tenido un crecimiento agitado, colmado de cambios, posibilidades, búsquedas e innovaciones. Esta metamorfosis fue la respuesta al ritmo vertiginoso de las necesidades y las tendencias que fueron mostrando los usuarios de Internet y a la evolución de los factores tecnológi-

cos que atravesó el medio web, entre los que podemos mencionar los siguientes:

- ▶ Avances en la velocidad y el tipo de conexión a Internet.
- ▶ Evolución del hardware: monitores, tarjetas de video y procesadores, entre otros componentes.
- ▶ Crecimiento del software, principalmente, el de uso libre.
- ▶ Masificación en el uso de los buscadores.
- ▶ Estandarización de los lenguajes según las normas de W3C (*World Wide Web Consortium*, www.w3c.org).
- ▶ Aparición de herramientas y aplicaciones web que ayudan a mejorar los procesos, los tiempos y la calidad de producción.

A partir de ahora, delinearemos los diferentes períodos del diseño web y analizaremos ejemplos característicos de cada uno.



PRIMEROS PASOS

En 1991, el físico inglés Tim Berners-Lee publicó la primera página web. Su contenido estaba conformado únicamente por **texto** e **hipervínculos** que hacían posible la navegación entre páginas. Ésta era una pequeña referencia sobre qué era la **World Wide Web** y qué se podía hacer en ella, y en menos de un año, recibió dos millones de visitas.

En esta primera etapa, el foco estaba puesto en la tarea de enviar o recibir datos. Las páginas solían ser hechas por programadores, y se caracterizaban por tener una interfaz lineal y limitada. Su diseño estaba generalmente diagramado a una sola columna y no se observaba el uso de cuadrículas que ayudaran a organizar la información. El acceso a Internet se efectuaba, principalmente, a través de conexiones **dial-up**, con una velocidad de navegación muy lenta, factor limitante del peso que podían tener los archivos. Por eso, las páginas de esa época contenían muy pocas imágenes, en general, dispuestas en orden consecutivo.

EN LOS COMIENZOS, LAS PÁGINAS SOLÍAN ESTAR HECHAS POR PROGRAMADORES, Y SE CARACTERIZABAN POR TENER UNA INTERFAZ LINEAL.

CREACIÓN DEL WORLD WIDE WEB CONSORTIUM (W3C)

En 1994, Tim Berners-Lee fundó el consorcio W3C, organismo destinado a establecer reglas y pautas que ayudarían a unificar criterios y a definir el futuro del diseño web.

En 1995 se publicó el estándar 2.0 del lenguaje **HTML (HyperText Markup Language)**, que a pesar de su número de versión, es el primer estándar oficial de ese lenguaje. Las primeras versiones de HTML sólo permitían dar a un texto el formato de encabezado, de párrafo y de hipervínculo (mediante **etiquetas**

básicas). En las versiones posteriores, se incorporó el uso de imágenes y de tablas, y de la mano de estos avances, fueron apareciendo varios navegadores web.

ICONOS, BOTONES Y BANNERS

El período en el cual aparecieron estos elementos comenzó alrededor del año 1996, momento en el que sucedieron varias cosas simultáneamente; veamos:

► Desarrollo de nuevos recursos gráficos: **iconos** que reemplazan palabras, **banners** que conforman cabeceras, **botones** con volumen que invitan a la interactividad y **fondos** que invaden las páginas. En el afán de experimentar y aprovechar los nuevos recursos, los desarrolladores web, conocidos en esa época como **webmasters**, empezaron a incluirlos de manera excesiva en sus diseños, lo que dio origen a páginas recargadas. ►

World Wide Web

The WorldWideWeb (W3) is a wide-area hypertext information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists, Policy, November's W3 news, Frequently Asked Questions.

What's out there?

Pointers to the world's online information, subjects, W3 servers, etc.

Help

on the browser you are using

Software Products

A list of W3 project components and their current state. (e.g. Line Mode, X11 Viola, NeXTStep, Servers, Tools, Mailboxes, Library)

Technical

Details of protocols, formats, program internals etc.

Bibliography

Paper documentation on W3 and references.

People

A list of some people involved in the project.

History

A summary of the history of the project.

How can I help?

If you would like to support the web.

Getting code

Getting the code by anonymous FTP, etc.

La primera página web, publicada en 1991 por Tim Berners-Lee, estaba compuesta sólo por texto e hipervínculos.



HTML

HTML (*HyperText Markup Language*) es el lenguaje básico con el que se escribe la mayoría de las páginas web. Está compuesto por etiquetas delimitadas por paréntesis angulares (<,>), que describen la estructura y el contenido. Más adelante, conoceremos en profundidad este lenguaje y lo utilizaremos para crear nuestras propias páginas.



Logo del consorcio W3C (www.w3c.org), organismo que aún en la actualidad es de gran importancia para el diseño web.

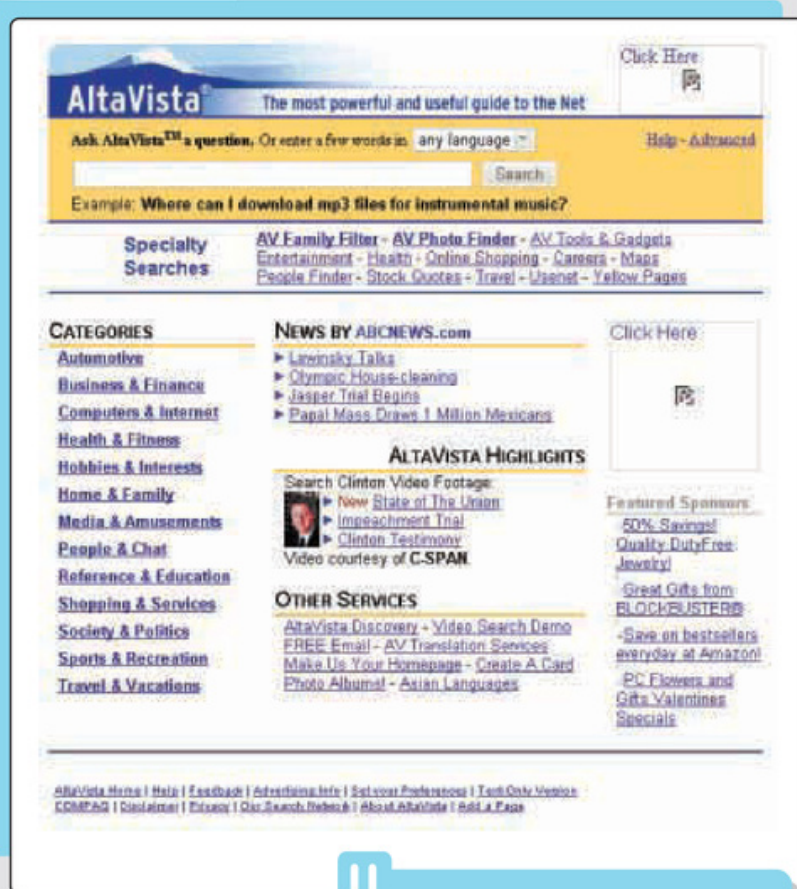
- ▶ Gracias a la definición de estándares HTML, se dieron los primeros pasos hacia la escritura de código semántico, compuesto por etiquetas bien estructuradas que describen el contenido.
- ▶ Para ordenar la información, comenzó a implementarse el uso de **tablas**, uno de los elementos más polémicos de HTML. Con este elemento, la diagramación de las páginas se enriqueció sobre las bases del diseño editorial (libros y revistas) e hizo uso de múltiples **columnas**. Aparecieron las primeras aplicaciones web basadas en el uso de tablas para la creación de páginas, lo que fomentó la adopción de la técnica por parte de los diseñadores. Si bien algunos desarrolladores la siguen utilizando, se trata de una técnica obsoleta y nada recomendable, concebida originalmente para mostrar información tabular.
- ▶ Ésta también fue la era del **spacer.gif**, una imagen transparente e invisible conformada por un píxel cuadrado. Su uso ganó popularidad rápidamente al permitir forzar y controlar los espacios vacíos dentro de una tabla, necesarios para separar los contenidos.
- ▶ En el terreno del hardware, se avanzó en términos de resolución y definición del color, tanto en los monitores como en las tarjetas gráficas. Esto trajo importantes mejoras en la calidad del diseño web.
- ▶ Comenzó a vislumbrarse lo que posteriormente se conocería como **guerra de los navegadores**, con las primeras diferencias en la adaptación de los estándares por parte de los dos principales browsers: Internet Explorer y Netscape Navigator.
- ▶ A finales de 1996, W3C publicó la primera recomendación oficial de CSS

(*Cascading Style Sheets*), conocida como **CSS nivel 1**, cuya adopción formal se produjo más adelante.

LA IRRUPCIÓN DE FLASH

Unos años más tarde, una aplicación surgida con el nombre **FutureSplash**

Interfaz de FutureSplash Animator (el precursor de Flash), utilizado para crear animaciones sencillas basadas en el uso de vectores.



Página principal del buscador AltaVista, lanzado en 1995. Fue el primero que permitió realizar búsquedas de texto sobre una base de datos.

Animator (hoy Adobe Flash) ganó popularidad entre los desarrolladores. Este software, que nació con una interfaz sencilla compuesta por una **línea de tiempo** y herramientas muy básicas, progresó hasta convertirse en una aplicación que permitió controlar el diseño y animar páginas web sin las limitaciones del código HTML de la época. La versatilidad de **Flash** permitió crear páginas dotadas de animación e interactividad, gracias a la edición de **fotogramas** y a la manipulación de objetos por separado. Hay quienes opinan que esta herramienta contribuyó a la evolución del diseño web, aunque muchos desarrolladores también sostienen que su uso desmedido no hizo más que degenerarlo. En 1998 se publicó la segunda recomendación oficial de CSS, conocida como **CSS nivel 2**. La versión de CSS que utilizan los navegadores actuales es CSS 2.1, una revisión de CSS 2 que aún se está elaborando. La siguiente recomendación de CSS, denominada **CSS**

nivel 3, continúa en desarrollo desde 1998, y hasta el momento sólo se han publicado borradores.

EL DISEÑO WEB ACTUAL

A partir de la aparición de mejoras en los navegadores, de la evolución en la velocidad de las conexiones y del desarrollo de nuevos dispositivos de navegación (celulares, PDA y consolas de juegos, entre otros), surgió una nueva generación de desarrolladores que deben tener en cuenta tanto las restricciones como las múltiples posibilidades existentes a la hora de implementar sitios.

La necesidad de reproducir los contenidos en los nuevos dispositivos y los avances constantes en los estándares web produjeron un nuevo impulso de sitios basados en **XHTML** (una combinación de **HTML** y **XML**) junto con **CSS** para separar cada vez más el **diseño** del **contenido**.

LOS DESARROLLADORES DEBEN TENER EN CUENTA LAS POSIBILIDADES Y LAS RESTRICCIONES A LA HORA DE DISEÑAR.

Por otro lado, la adopción de **JavaScript** (un lenguaje de programación sencillo capaz de generar contenidos interactivos) para la creación de **transiciones** relegó el uso de Flash a sitios de animación cada vez más específicos o al uso puntual de la tecnología en elementos como reproductores de video o banners. A lo largo de los años, los webmasters fueron reemplazados por equipos interdis-

ciplinarios de trabajo compuestos por programadores, arquitectos de la información, maquetadores y diseñadores web. Todos ellos construyen sitios basándose en las mejores prácticas, que también evolucionan continuamente.

La colaboración fluida entre pares, el acceso a tecnologías web de código abierto (gestores de contenidos), la disponibilidad de recursos gráficos en línea, y la incorporación de herramientas de testeo y de desarrollo modificaron el trabajo del diseñador web, y elevaron a un nivel muy superior la calidad de los productos finales. ◀



Página principal de uno de los sitios realizados en Flash más populares de la época: EYE4U.



CSS

Aunque lo conoceremos y utilizaremos más adelante en esta obra, vale aclarar que CSS es el lenguaje de hojas de estilo en cascada creado para controlar el aspecto visual de los documentos HTML. Es la mejor forma de separar los contenidos y su presentación.

TIPOLOGÍA DE LOS SITIOS WEB

Un diseñador web puede ofrecer muchas soluciones tecnológicas a las necesidades de un cliente. Las herramientas y tecnologías actuales nos permiten crear páginas con información y con diversos servicios.

Producto del avance tecnológico y del creciente número de personas que acceden a Internet, aparecieron variadas herramientas y servicios que es posible utilizar online. Gracias a esto, muchos aspectos de nuestra vida cotidiana se ven directamente influenciados por la proliferación de los sistemas web, como nuestros hábitos de compra, de pago de servicios o de comunicación. Para ofrecer estas utilidades, existen diversos tipos de sitios, que conoceremos a continuación.

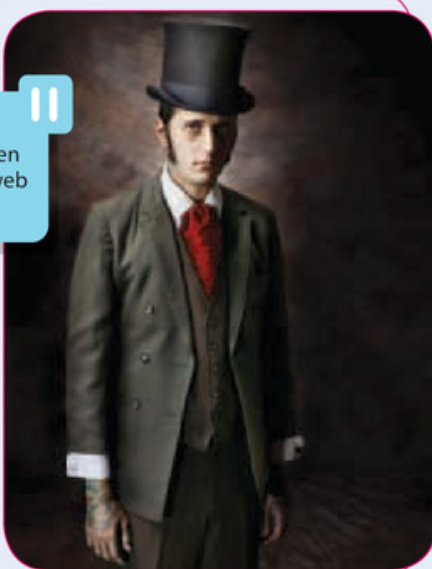
En este portfolio de fotografías, podemos ir navegando de una imagen a otra interactuando con la cámara web (www.davidlindseywade.com).

SITIOS DINÁMICOS

Aunque podríamos pensar que un sitio dinámico es aquél que tiene animaciones o que es veloz en algún aspecto, este concepto se refiere al sitio en el que, al hacer clic para acceder a una página, ésta se arma según nuestro pedido. En los sitios estáticos, en cambio, las páginas están armadas y no se modifican por lo que el usuario efectúe. El **armado a pedido** es realizado por dos partes: el motor de armado y la información con la que éste genera la página. El **motor de armado** está creado con un lenguaje de programación, y la **información** suele provenir de una base de datos. Las ventajas de un sitio dinámico son la facilidad de actualización y la sencillez en la carga de datos. Encontramos sitios dinámicos en cualquiera de los siguientes casos:



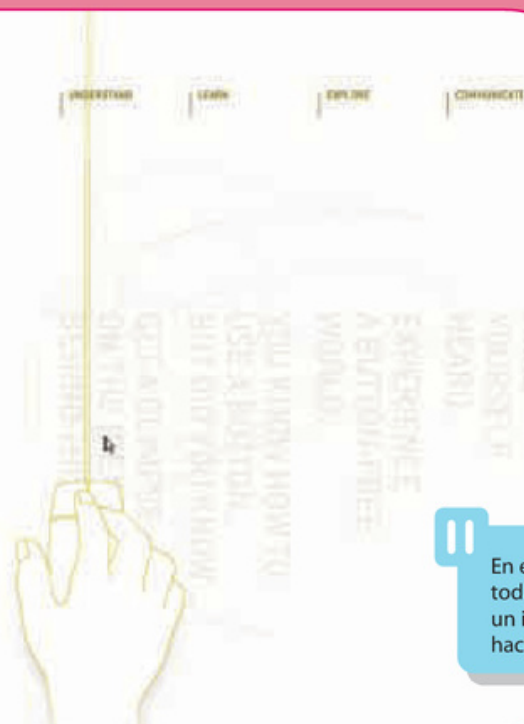
El caso ejemplar es Facebook (www.facebook.com), con una cantidad de usuarios que ronda los 124 millones a nivel mundial.



WordPress

Dentro del listado de blogs y microblogs que analizamos, WordPress merece una mención, ya que ofrece, en forma gratuita, los archivos fuente que componen el blog para descargar, instalar y personalizar, y así utilizar el blog en cualquier dominio y hosting de que dispongamos.

- **Foros:** plataformas donde los usuarios de una comunidad escriben consultas, comentarios y respuestas que son almacenados en una base de datos.
- **Blogs y microblogs:** sistemas que nos permiten tener un espacio personal, a partir de una plantilla prediseñada, en el que podemos compartir y comentar información como música (www.myspace.com), videos (www.youtube.com), imágenes (www.fotolog.com y www.flickr.com), textos y multimedia (www.blogspot.com y <http://wordpress.com>), y textos reducidos (<http://twitter.com>).
- **Wikis:** plataformas de creación de contenidos en forma participativa, como la enciclopedia <http://wikipedia.org> o <http://commons.wikimedia.org>, un repositorio de archivos de uso gratuito.
- **Portales de noticias:** sitios con contenidos generados, comúnmente, por periódicos de edición impresa. En este caso, los usuarios no pueden editar los contenidos, pero sí dejar comentarios en algunas noticias.
- **E-commerce (comercio electrónico):** sitios como www.ticketmaster.com

DONTCLICK.IT
INSTITUTE FOR INTERACTIVE RESEARCHCAN YOU
RESIST TO
CLICK?REPLAYING
MOVEMENT

En este sitio la navegación y el acceso a todas las secciones se hacen mediante un interesante sistema en el que no hacen falta los clics del mouse.

y **www.amazon.com** ofrecen entradas a espectáculos o productos como libros y películas en sus páginas, donde el usuario puede elegir, entre otros detalles, la cantidad y el tipo de artículo, la forma de envío y la de pago, que suele ser mediante tarjeta de crédito.

► **Redes sociales:** plataformas en las que podemos generar un perfil con datos personales, intereses, profesión y educación. A partir de él, es posible contactarnos con personas que tengan intereses similares o nuestra misma profesión, por ejemplo. También permiten agregar álbumes de fotos, videos, links y comentarios, estos últimos, tanto en el perfil propio como en el de los contactos.

OTROS TIPOS DE SITIOS

Muchos sitios ofrecen aplicaciones online en forma gratuita, entre las que encontramos editores de texto, de sonido y de imágenes; planillas de cálculo y agenda. La última tendencia es que las aplicaciones no estén instaladas en nuestras computadoras sino disponibles como un servicio online. El ejemplo más referencial es Gmail.com, que ofrece muchos de estos servicios y, además, permite personalizar colores, logos, fondos y la disposición de las interfaces. Por otro lado, encontramos novedades en el intercambio y alojamiento

de archivos. La calidad de las imágenes, los videos y el audio disponibles en las computadoras ha aumentado, lo que provoca cambios en el modo de distribuir y transferir estos archivos, cada vez más pesados. Sitios como **www.rapidshare.com**, **www.dropbox.com** o **www.yousendit.com** nos permiten almacenar y enviar archivos de gran tamaño.

SITIOS ESTÁTICOS

Son sitios donde los usuarios no pueden crear ni editar los contenidos. Los casos más comunes de este tipo son:

► **Sitios experimentales:** en estos casos se busca generar nuevas experiencias de navegación para el usuario, al producir interacciones no convencionales (**http://dontclick.it**), recorridos 3D (**www.vatican.va/various/cappelle/sistina_vr/index.html**) o navegación mediante la cámara web (**www.davidlindseywade.com/#/portfolio/portfolio/19**).

► **Newsletters y flyers:** son minisitios publicitarios que están alojados en un servidor y son enviados por e-mail con el fin de promover algún producto o servicio.

► **Sitios tradicionales:** este tipo de sitios es el más utilizado para presentar información. Se trata de sitios en los que el usuario puede navegar para leer contenidos y ver imágenes, y también ponerse en contacto con la empresa o profesional dueño del sitio. Ejemplo de este tipo de sitio es el portfolio de un diseñador, tal como el que haremos a lo largo de esta obra. ◀



DISEÑAR UN PORTFOLIO

Una tarea que debemos realizar los diseñadores web es el desarrollo de nuestro portfolio personal, en donde mostraremos lo que hemos hecho hasta el momento a quienes estén interesados en nuestros servicios. Analicemos este elemento promocional en detalle.

Al momento de diseñar nuestro portfolio, nos enfrentamos con una situación de incertidumbre respecto a la manera en que deberíamos exhibir nuestros trabajos y capacidades. Ésta es una decisión que debemos tomar con mucho detenimiento, ya que nuestro portfolio será un sitio en donde otros diseñadores podrán apreciar nuestro talento –lo que generará una mayor reputación–, y en el que los posibles clientes analizarán nuestras capacidades y experiencia. Por eso, a la hora de diseñar y crear nuestro portfolio, es fundamental tener presentes algunos aspectos muy importantes, que veremos en detalle a continuación.

DATOS RELEVANTES

Nuestros potenciales clientes precisarán ver datos que reflejen nuestra trayectoria y experiencia. Esta información tiene que incluir la cantidad de trabajos que hemos hecho, los detalles sobre el último que efectuamos, el tiempo que llevamos ejerciendo como diseñadores web freelance y las tecnologías empleadas en cada proyecto, entre otros puntos.

TRABAJOS REALIZADOS

Éste es uno de los elementos más importantes de nuestro sitio, ya que será una muestra directa de los proyectos que hemos llevado a cabo. Podemos mostrar lo que hicimos de dos maneras diferentes. Por un lado, si el proyecto desarrollado se encuentra **online**, es posible crear un enlace con la URL real correspondiente. Esto puede ser muy valorado

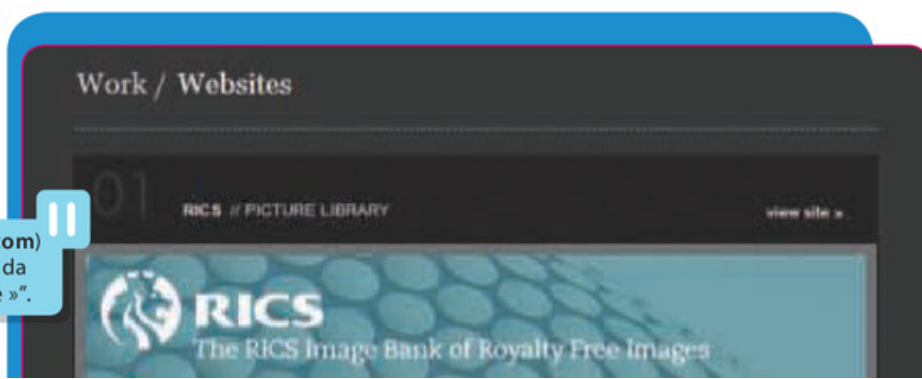


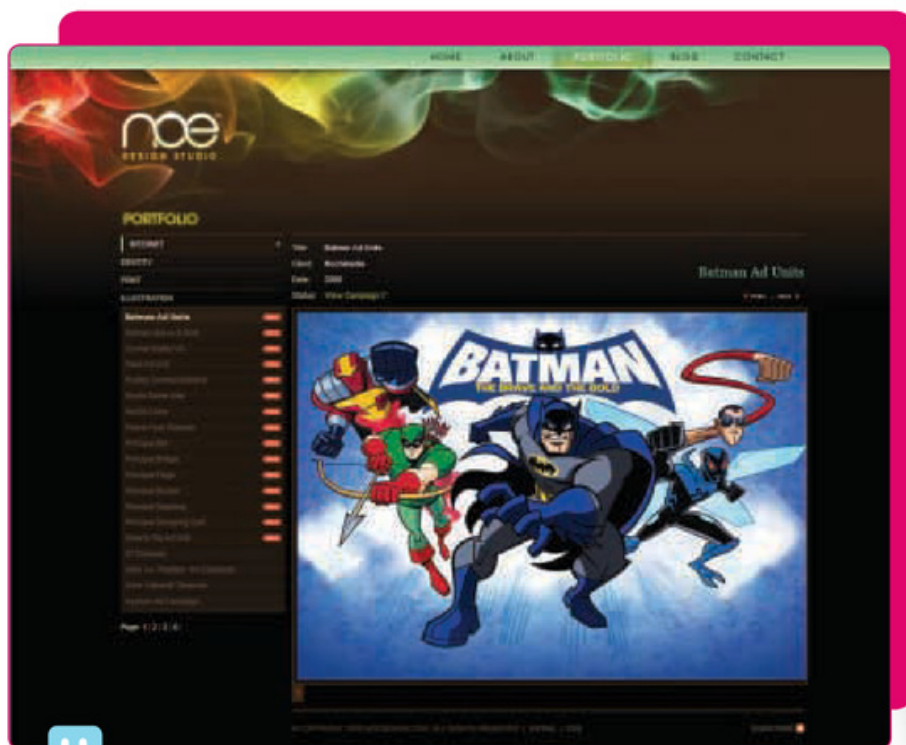
2Advanced (www.2advanced.com) nos permite revisar los proyectos organizados por fecha, por clientes o servicio, entre otras opciones.

por quienes están buscando un diseñador, ya que así podrán apreciar nuestro trabajo funcionando en línea. Además, el hecho de que el material aún esté online significa que el cliente ha quedado satisfecho y que, por eso, lo sigue utilizando. También podemos mostrar nuestros proyectos realizados a través de la inclusión de imágenes en el portfolio. En este caso, debemos asegurarnos de que las imágenes sean un fiel reflejo de la calidad de lo que entregamos a nuestros clientes.

LAS OPINIONES Y RELATOS DE NUESTROS CLIENTES PUEDEN SER ÚTILES EN LA CONFECCIÓN DEL PORTFOLIO

El portfolio de Minuco (www.minuco.com) nos ofrece la posibilidad de conocer cada uno de sus desarrollos desde "view site".





El estudio de diseño Noe (www.noedesign.com) resalta, dentro de cada categoría, los nuevos trabajos realizados.

NUESTRA ESPECIALIDAD

Si bien es cierto que en el portfolio mostraremos, seguramente, la totalidad de nuestros trabajos, es preciso distinguir en qué nos destacamos y cuál es nuestra mayor destreza. Esto puede resaltarse citando en qué nos diferenciamos del resto y cómo logramos esa diferencia.

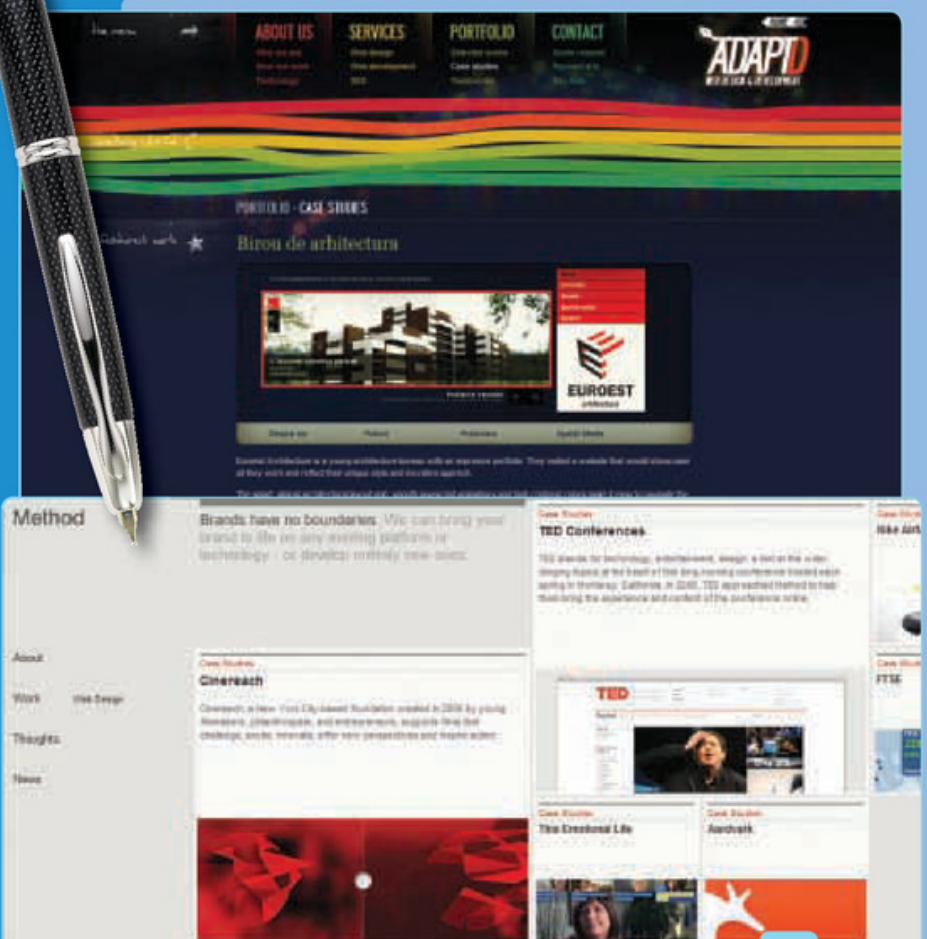
DEFINIR EL DISEÑO

Exponer nuestro trabajo no sólo implica colocar imágenes y enlaces sobre los proyectos que hemos desarrollado: el portfolio en sí mismo debe demostrar nuestra calidad profesional y lo que estamos capacitados para realizar. Cuando creamos el sitio, debemos tener en cuenta cuál es nuestro principal **enfoque** al diseñar el portfolio. Con esto nos referimos a los **objetivos predominantes** y a los destinatarios, es decir, a quiénes deseamos venderles nuestros servicios. Por ejemplo, podríamos optar por un diseño minimalista para intentar manifestar nuestro profesionalismo, aunque en otros casos quizá deseemos exhibir un diseño con un gran nivel de detalle y elementos gráficos de alto impacto, que pongan de manifiesto nuestra gran creatividad.

CASOS DE ÉXITO

Algunos visitantes de nuestro portfolio podrían requerir una mayor cantidad de información, más allá de los trabajos realizados. Por ejemplo, tal vez desearían encontrar redacciones que cuenten cómo logramos resolver los problemas de determinados clientes. Como podemos imaginar, este punto también es muy importante, ya que estos casos son una evidencia real de lo que hemos hecho y de la forma en la que lo conseguimos. También podemos acompañar cada caso de éxito con **opiniones** y **relatos** de nuestros clientes, quienes, voluntariamente, pueden colaborar con la confección de nuestro portfolio a través de sus palabras y sus anécdotas mientras trabajaron con nosotros.

Por todo lo que analizamos hasta ahora, es recomendable hacer varias revisiones del diseño y el contenido de nuestro sitio antes de darlo por terminado. Esto nos ayudará a lograr un portfolio que pueda sorprender al visitante, quien, gracias a lo que encuentre en el sitio, podría convertirse en nuestro próximo cliente. ◀



Los sitios www.adapttd.com y www.method.com muestran diversos comentarios sobre los trabajos que realizaron.

COMING SOON PAGE

Si ya contamos con un nombre de dominio, es común que, mientras estamos preparando nuestro próximo diseño, publiquemos la denominada coming soon page, es decir, un aviso indicando que, en breve, allí estará nuestro trabajo.

Para entender mejor de qué se trata una coming soon page, analizaremos un ejemplo muy simple y cotidiano. Cuando una película va a estrenarse en el cine, unos días antes de que esto suceda, se publican anuncios y publicidades que anticipan la fecha de estreno. De esta misma manera funcionan las coming soon pages: antes de que el diseño se encuentre implementado, se informa a quienes visitan el dominio que muy pronto verán el sitio online.

¿DE QUÉ SIRVEN LAS COMING SOON PAGES?

Supongamos que ya hemos hecho nuestras tarjetas personales y que, además, invertimos dinero en publicidad y en otras acciones de marketing, incluyendo en todas ellas la URL de nuestro sitio web. Las coming soon pages invitarán a quienes ingresen en nuestra web a regresar en unos días, prometiendo que, en breve, verán lo que estamos preparando. Para decirlo de otro modo, estas páginas serán una carta bajo la manga que utilizaremos para aprovechar mejor el tiempo durante el cual estemos trabajando en nuestro sitio web.

APROVECHAR AL MÁXIMO ESTE RECURSO

La creatividad abunda en el diseño de este tipo de páginas temporales, y no es para menos. Muchos expertos invierten mucho tiempo diseñando una idea creativa para exhibir como coming soon page, y el beneficio de esta inversión es doble. Por un lado, los visitantes sentirán una cálida bienvenida a nuestro sitio y serán invitados a regresar pronto. Por el otro, si nuestra idea es original y el trabajo es de gran calidad, este tipo de páginas suelen ser listadas por directorios web y blogs de diseño web, lo cual nos dará prestigio y publicidad adicional.

INFORMACIÓN SOBRE NUESTRA PÁGINA

En nuestra coming soon page, deberemos focalizarnos en dos tipos de datos: la información que brindamos y aquella que vamos a recolectar. Veamos los detalles de ambos casos:

- Los **datos exhibidos**: incluyen nuestro slogan, nuestros principales productos, algún medio de contacto (teléfono o correo electrónico) y la fecha de publicación del sitio, entre otros detalles.
- Los **datos solicitados**: son muy importantes porque podrían incluir desde la dirección de correo electrónico del interesado en nuestro sitio web, hasta su nombre y datos personales. Si recolectamos este tipo de información, al momento de realizar la implementación del sitio web, podremos enviar un correo electrónico anunciando sobre la puesta en marcha del sitio. Así, lograremos que la visita se repita en el momento adecuado.

LA IMPORTANCIA DEL POSICIONAMIENTO SEO

Es importante mencionar que las coming soon pages también deben contar con una optimización para el posicionamiento SEO. Al ser indexada por los buscadores, nuestra página será catalogada y ubicada en sus rankings, por lo que resulta fundamental tener presente el posicionamiento, incluso, de este tipo de páginas. Posteriormente, en la puesta en marcha del sitio, podremos comenzar con el posicionamiento en detalle. Para obtener más información sobre este tema, recomendamos consultar el sitio web de la editorial Users, en la dirección: www.redusers.com.



NO OLVIDAR LA SIMPLICIDAD

Cuando hablamos de simplicidad, nos referimos a enviar un mensaje claro y directo a quien nos visita. Debemos ser precisos con la información que brindamos, y dejar una buena imagen en el visitante, que esperaba encontrar el sitio web finalizado en el dominio que visitó. Tenemos que diseñar nuestra coming soon page con un mensaje limpio. Por ejemplo, podemos incorporar la palabra "Próxima-

mente" (un texto muy utilizado en estos casos), acompañada por los datos exhibidos y los solicitados. Respecto de estos últimos, es conveniente mencionar que cuantos menos datos le solicitemos al visitante, mayores probabilidades tendremos de que nos los envíe. En general, como lo que buscamos

con esta solicitud es lograr un acercamiento en el futuro, en lo posible, sólo le pediremos su correo electrónico, que bastará para mantenernos en contacto. ◀



LAS COMING SOON PAGES INVITAN AL VISITANTE A REGRESAR AL SITIO CUANDO ÉSTE SE ENCUENTRE ONLINE.



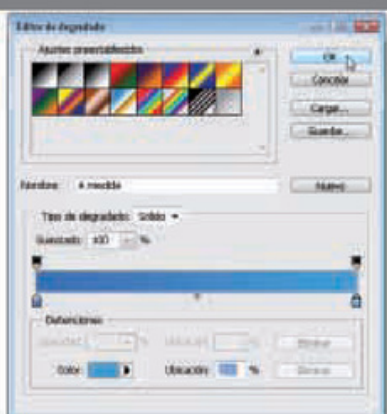
Podemos encontrar muy buenos ejemplos de coming soon pages en www.silverbackapp.com/holding, www.mynitelife.co.uk, www.pepsay.com y www.redkatana.com.

Las coming soon pages nos brindan una nueva posibilidad de destacarnos de nuestra competencia y de mostrar que, incluso en los detalles, contamos con una gran creatividad y alta capacidad de comunicación. Aquí, un ejemplo que encontramos en www.dmbnewstudioalbum.com.

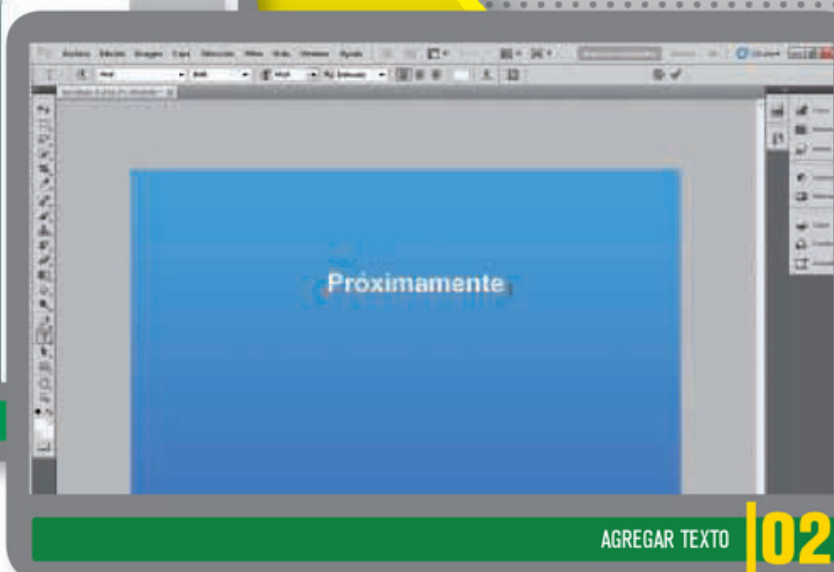
CREAR UNA COMING SOON

PASO A PASO

Diseñaremos una sencilla pero comunicativa coming soon page en Adobe Photoshop para que el visitante vuelva pronto.



01 | MODIFICAR FONDO



AGREGAR TEXTO | 02

Antes de crear nuestra página coming soon, sería ideal que tuviéramos alguna idea sobre el diseño que tendrá el sitio en el futuro. De esta manera, podríamos utilizar los mismos colores y tipografías como adelanto de lo que vendrá, mostrando desde el comienzo la capacidad que tenemos para diseñar. En cambio, si aún no tenemos definidos los colores y el estilo del diseño, podemos crear una coming soon page donde prevalezcan formas y colores neutros y simples. Una vez que decidimos qué es lo que vamos a incluir, abrimos Photoshop y creamos un nuevo documento (desde **[Archivo/Nuevo]**). Será suficiente con una medida de 1280 píxeles de ancho por 1000 de alto y un color blanco de fondo, ya que lo modificaremos como veremos a continuación. ◀

PASO | 01 Para modificar el fondo, que en este caso será un degradé, seleccionamos la herramienta **[Degradado]** (tecla <G>) y colocamos los colores **#0078e7** para el extremo izquierdo y **#01a7ff** para el extremo derecho. Luego, realizamos un clic en la parte inferior de nuestro documento y, sin soltar el clic del mouse, arrastramos el degradé hasta la parte superior del documento.

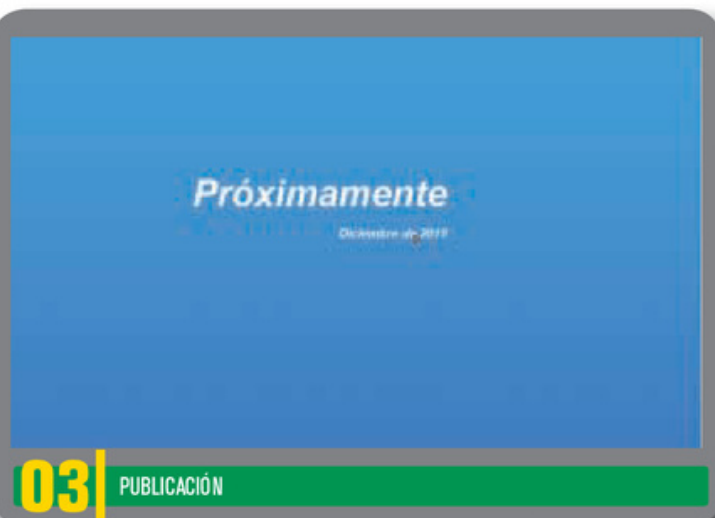
PASO | 02 Ahora, incluiremos el texto "Próximamente" en la parte superior-central del diseño. Para esto, seleccionamos la herramienta **[Texto horizontal]** (tecla <T>) y elegimos la fuente **[Arial]** con estilo **[Bold]**; le damos color blanco y 60 píxeles de tamaño. Luego, escribimos la palabra "Próximamente" en nuestro documento y ubicamos ese texto donde deseemos.

PAGE

04 | CREAR FORMULARIO



03 | PUBLICACIÓN



BORDE DE CAMPO | 05



PASO | 03

Para agregar la fecha en que publicaremos, creamos otra capa de texto. Mantenemos la fuente y su color pero elegimos el tipo **[Bold Italic]** y 20 píxeles como tamaño. Escribimos "Septiembre de 2011" y la ubicamos en la parte inferior-derecha de la palabra "Próximamente".

PASO | 04

Para recibir correos electrónicos a través de un formulario, seleccionamos la herramienta **[Texto horizontal]** con los mismos parámetros anteriores y escribimos "Ingrese su correo para notificarlo:". Para hacer los campos de texto dibujamos un rectángulo con la herramienta **[Marco rectangular]** y lo pintamos con el **[Bote de pintura]** usando el color **#07d6ff**.

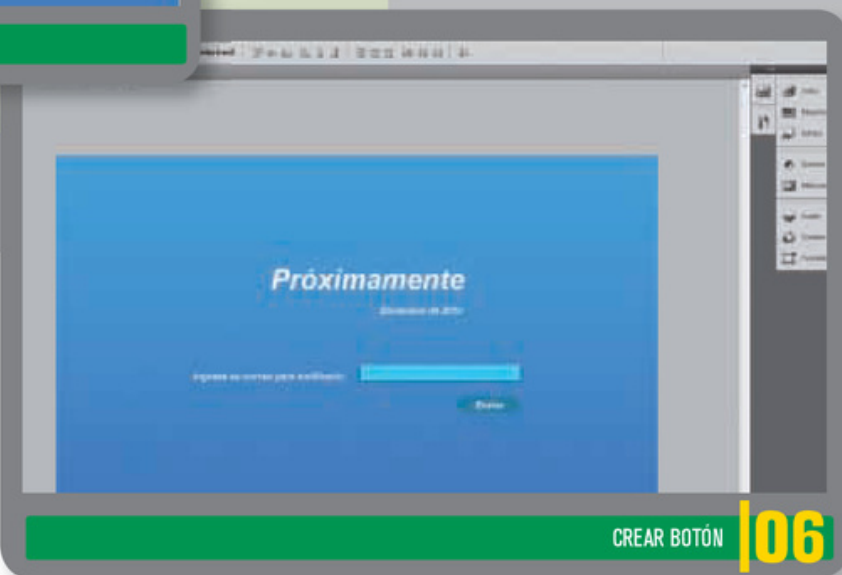
PASO | 05

Para el borde, vamos a **[Edición/Contornear]** y, en la ventana que aparece, ingresamos 1 píxel para el ancho del contorno y **#047da5** como color.

PASO | 06

Para crear el botón, elegimos la herramienta **[Rectángulo redondeado]** y configuramos 50 en el campo **[Radio]** de la barra de herramientas superior. Dibujamos la base, seleccionamos la herramienta **[Texto horizontal]** y escribimos el texto "Enviar" sobre él.

CREAR BOTÓN | 06



RESUMEN

Hemos diseñado una coming soon page. Gracias a esto lograremos que los visitantes, con una atractiva impresión, vuelvan a ingresar a nuestro sitio en un futuro muy próximo.

UN PROBLEMA CON BUEN DISEÑO

Las páginas de ERROR 404 son aquellas que aparecen cuando el visitante ingresa en una URL que no existe en nuestro dominio. A continuación, conoceremos la importancia de realizar un buen diseño para este tipo de elementos.

La manera en la que mostremos nuestras páginas de ERROR 404 determinará si el visitante se retirará del sitio web o continuará navegando en él. Por este motivo, resultan muy importantes y cumplen un papel fundamental en cualquier sitio que desarrollemos.

¿CUÁNDO SE PRESENTAN?

Estas páginas se visualizan cuando ocurre un error al intentar acceder a un sitio web. Los motivos más comunes por los que puede producirse este inconveniente son los siguientes:

- ▶ **Enlace roto dentro del sitio:** esto ocurre cuando enlazamos un recurso dentro de nuestro sitio que ya no existe o que se ha movido de lugar.
- ▶ **Enlace roto desde un buscador:** los buscadores son la fuente de visitantes más importante hacia el sitio web. Puede ocurrir que hayamos eliminado una página del sitio y que el buscador aún no haya indexado este cambio, con lo cual mantiene un enlace que no

existe. Esto se da porque los buscadores tienen una determinada frecuencia para realizar la indexación; por ejemplo, ingresan cada dos semanas en nuestro sitio para incluir en sus rankings los cambios que se hayan hecho.

404 error: File not found

The [URL](#) you requested was not found.

Did you mean to type <http://en.wikipedia.org/wiki/X>? You will be automatically redirected there in five seconds.

Maybe you would like to look at:

- [The main page](#)
- [The list of Wikimedia downloads](#)

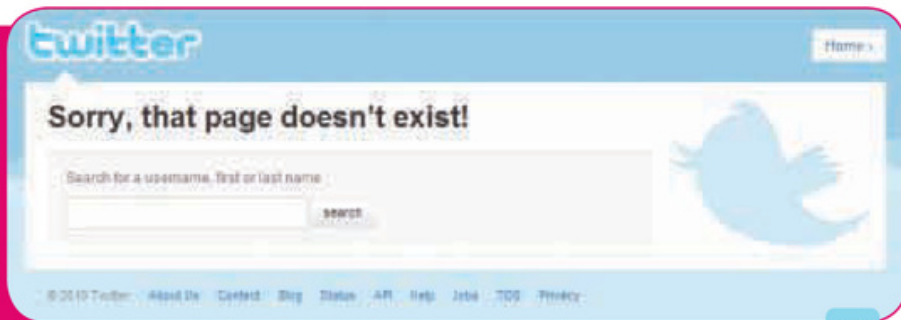
A project of the [Wikimedia Foundation](#)

Hasta hace un tiempo, las páginas de ERROR 404 no eran consideradas por los webmasters y, en consecuencia, se perdía al visitante que intentaba acceder a alguna parte del sitio web.

¿CÓMO TENDRÍA QUE SER?

A la hora de crear una página de ERROR 404, hay dos aspectos básicos que debemos tener en cuenta:

- ▶ Debería ser diferente del resto de las páginas del sitio, ya que con esto lograremos que el usuario perciba de forma rápida y clara que ha arribado a una página que no existe.
- ▶ **Mensaje claro:** tal como mencionamos con las coming soon pages, debemos transmitir nuestro mensaje de manera clara y directa hacia el visitante.



En la actualidad, abunda el ingenio a la hora de crear páginas de error. Éste es un ejemplo de www.twitter.com.

¿QUÉ DEBERÍA INCLUIR?

A grandes rasgos, la página de error debería incluir el logotipo de nuestro sitio web, el mensaje de error "Página no encontrada", un buscador interno del sitio y enlaces de referencia para ayudar al visitante.

Como vemos, existen dos objetivos en esta página. Por un lado, informar al visitante el sitio en el que está, a través del logotipo y el mensaje. Por el otro, atraerlo a continuar navegando por el sitio web, a través del buscador y los enlaces de referencia.

En algunos casos, también se implementa la opción de reportar o notificar el error por medio de un formulario directo de contacto. Esta opción suele ser utilizada por grandes portales que consideran que esta alternativa es valiosa para mantener la fidelidad del usuario, quien así tiene oportunidad de expresar la situación. La cordialidad también es parte importante de las páginas de ERROR 404. En este sentido, se colocan frases como "Le pedimos disculpas" o "Lo sentimos", para que el visitante reciba un grato mensaje ante una situación que suele ser confusa.

AL CREAR UNA PÁGINA DE **ERROR 404** DEBEMOS CENTRARNOS EN SU OBJETIVO, QUE ES EL DE SER **INFORMATIVA**.

EL OBJETIVO ANTE TODO

Una regla de oro al crear este tipo de páginas es centrarse en su objetivo, que es informar. Debe ser una página clara, concisa y explicativa, con el énfasis puesto en comunicar dónde ha llegado el visitante, por qué está en ese lugar y cómo puede continuar. En tanto informemos, podemos mostrar un diseño serio o informal, ofreciendo herramientas y opciones de navegación que ayuden a conservar esa visita que arribó a un lugar inexistente. Así, lograremos convertir en productivo algo que, en un principio, es todo lo contrario.

Desde TemplateMonster.com podemos bajar diferentes páginas de error (www.templatemonster.com).



El blog de diseño español Ateneu Popular 2.0 (www.ateneupopular.com) y su página 404.

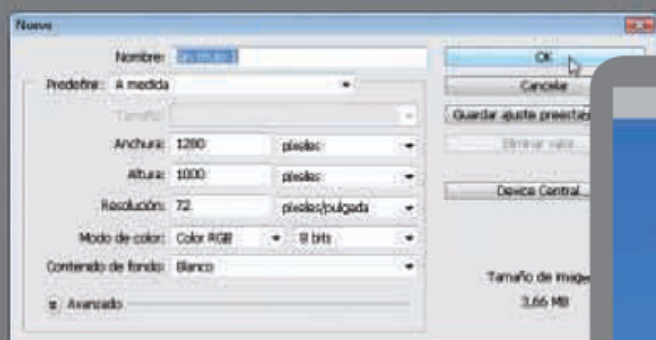


En ocasiones, Taringa también tiene inconvenientes. Un hombre llorando es su imagen por defecto.

CREAR UNA PÁGINA

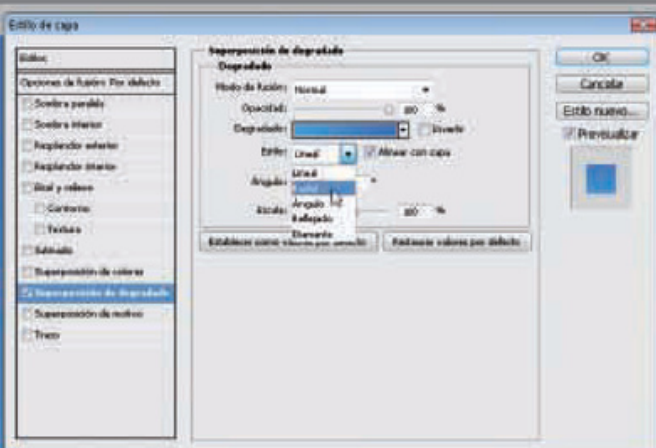
PASO A PASO

Una página 404 debe ser directa y bien informativa. Por este motivo, diseñaremos con Adobe Photoshop una página de ERROR 404.



01 | CREACIÓN DEL DOCUMENTO

Como ya vimos, las páginas de error son necesarias para avisarle al usuario que arribó a un lugar equivocado y, al mismo tiempo, para orientarlo hacia el punto que estaba buscando en un principio. ◀



CREACIÓN DE CAPAS | 02

PASO | 01 Para crear nuestra página de error, abrimos Photoshop y, desde el menú **[Archivo/Nuevo]**, creamos un documento que tenga el fondo blanco y un tamaño de 1024 píxeles de ancho por 1000 píxeles de alto.

PASO | 02 Creamos una nueva capa a través del comando **<Ctrl + Shift + N>**, la denominamos **"Fondo degradado"** y la rellenamos con la herramienta **[Bote de pintura]** para, después, poder definir el degradado de su interior. Luego, seleccionamos la capa y abrimos el menú **[Capa/Estilo de capa/Opciones de fusión...]** y, en la ventana que aparece, vamos a la sección **[Superposición de degradado]**. Allí, configuramos **#01a7ff** para el extremo izquierdo del degradado y **#0078e7** para el extremo derecho. Allí mismo, elegimos la opción **[Radial]** en el desplegable **[Estilo]**. Al finalizar, presionamos **[OK]** para aplicar los cambios.

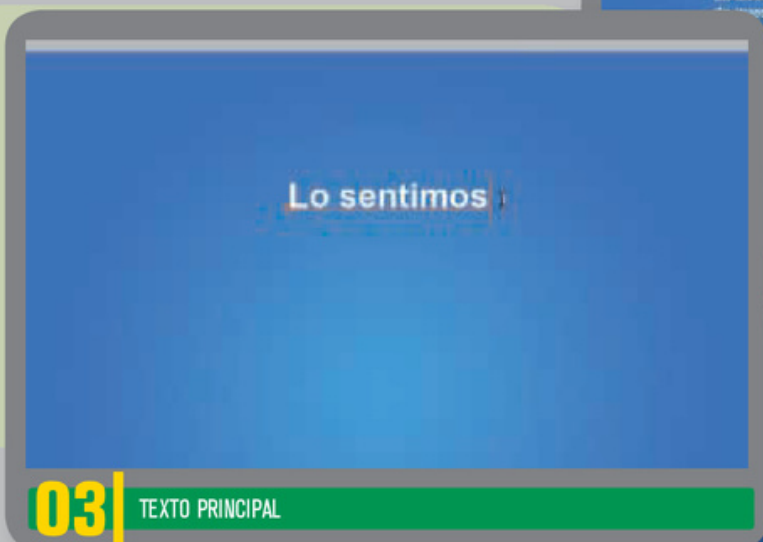


DE ERROR 404

04 | TEXTO EXPLICATIVO



LINKS | 05



03 | TEXTO PRINCIPAL



INCLUIR UNA IMAGEN | 06

PASO | 03 Para incorporar el mensaje principal, que dirá "Lo sentimos" en la parte superior-central del diseño, utilizamos la herramienta **[Texto horizontal]** (a la que accedemos con la tecla <T>) con la fuente **[Arial]**, color blanco, estilo **[Bold]** y 30 píxeles de tamaño.

PASO | 04 Para explicarle al visitante lo que ha ocurrido, agregamos un texto debajo del título que colocamos en el paso anterior. Utilizamos la misma fuente y el mismo color, pero esta vez en un tamaño de 24 píxeles. Podemos escribir el texto que deseemos; lo importante es que aclare que se trata de un error que no es culpa del visitante, ya que así lo mantendremos en calma.

Para que el usuario pueda salir de la página de error, agregaremos algunos enlaces. Elegimos la herramienta **[Texto horizontal]** con la fuente **[Arial]** en color negro, estilo **[Regular]** y 24 píxeles de tamaño. Escribimos los textos "Volver al inicio", "Utilizar nuestro buscador" y "Contactarse con nosotros", cada uno seguido de un salto de línea para espaciar las frases.

PASO | 06 Si queremos un mensaje formal y serio, podemos mostrar la fotografía de un oficinista. En cambio, si queremos dar una imagen más informal, elegiremos una ilustración o una fotografía creativa que pongan al visitante de buen humor ante el imprevisto. Incorporamos la imagen de un cartel de direcciones en una nueva capa y la ubicamos a la izquierda de nuestros enlaces.

RESUMEN

Podremos implementar la página de ERROR 404 y observar los asombrosos cambios en cuanto a las visitas en nuestro sitio. Veremos que aparecerá una nueva cantidad de visitantes.

LA ESTRUCTURA DE LAS PÁGINAS

Todos los internautas somos testigos del avance que han sufrido los contenidos de un sitio web en sus tecnologías, su diseño y su interactividad. Veamos ahora la evolución de su estructura.

Es hora de comenzar a presentar los aspectos relacionados con los elementos que conforman una **interfaz**, la manera en la que éstos se estructuran, y los criterios que debemos seguir para realizar una página atractiva y funcional.

Cada sitio web tiene un **objetivo comunicacional** y, debido a eso, todos sus aspectos visuales y la articulación de sus elementos deben garantizar que se cumpla ese objetivo.

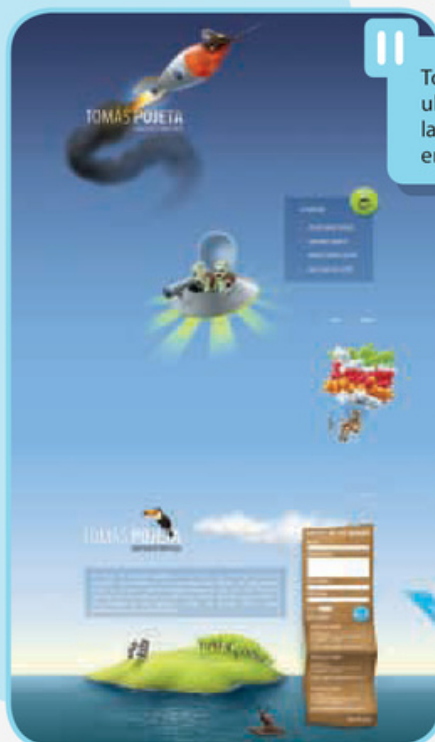
Es necesario que el internauta comprenda el objetivo del sitio en pocos segundos. El tiempo es tirano, y el navegante está expuesto a una innumerable cantidad de sitios, una avalancha de información que hace imprescindibles la claridad del mensaje y la buena legibilidad.

LA FORMA DE NAVEGACIÓN

A la hora de diseñar un sitio, tenemos la libertad de elegir qué forma de navegación utilizaremos, pero debemos hacerlo con criterio y adecuándonos a cada caso en particular. Veamos, a continuación, algunos sitios que proponen formas novedosas de navegación, con intervenciones poco comunes por parte de los usuarios.

► **Navegación sin clics:** www.dontclick.it propone una navegación sin clics, haciendo la elección de lo que deseamos con sólo aguardar unos segundos sobre el elemento al que queremos acceder.

► **Navegación vertical:** en lugar de tener una página para cada sección del



Todo el contenido del sitio está ubicado en una sola página muy larga, que se muestra por partes en la ventana del navegador.

sitio, todo el contenido se presenta en una sola página, que se mueve hacia arriba o hacia abajo según las opciones que vayamos eligiendo. Podemos ver ejemplos en www.volll.com, www.kitfolio.com y www.pojeta.cz.

► **Navegación con el mouse:** en <http://doorstepdairy.com> encontramos que la navegación se realiza mediante el movimiento de un elemento. Aquí, arrastramos un pequeño camión en lugar de hacer clic en botones.



INNOVACIONES SEGURAS

Sin recurrir a innovaciones que no son seguras, tenemos muchos recursos para impactar. En las siguientes direcciones podemos encontrar interfaces estructuradas que poseen un gran atractivo desde el diseño, el color, el uso tipográfico, y otros elementos:

- www.pieoneers.com
- www.meetemplates.com
- <http://skype.com>
- www.milrayas.com
- www.giraffe.net
- www.mcmillerssweetsemporium.co.uk

Si imaginamos una interfaz de navegación con movimiento vertical para un sitio con muchas secciones y que ofrezca varios productos, el largo necesario para la página sería excesivo. Si nos moviéramos del último producto al comienzo, la sensación al ver pasar palabras, imágenes y colores a gran velocidad quizá resultaría desagradable. Esto nos demuestra que no todos los recursos disponibles son aplicables a cualquier sitio, y si a pesar de eso los utilizamos, tal vez el usuario no tenga una buena percepción de nuestra propuesta.

¿QUÉ ES LA ESTRUCTURA?

Nadie espera encontrar un menú en el medio de un párrafo, ni tampoco un buscador o un logo en el pie de la página. Aunque sea de forma inconsciente, todos esperamos que los elementos estén dispuestos de cierta manera, donde los buscamos en primer lugar cuando accedemos al sitio. Un usuario tiene la capacidad de predecir el sitio, buscando de forma rápida lo que le interesa. Luego de reconocer la disposición de los elementos en la página de inicio, los seguirá buscando en el mismo lugar dentro de las diferentes secciones. Por eso, necesitamos mantener estándares de estructura para darle una coherencia



Cada vez son más los sitios con diseños de vanguardia que le otorgan un papel protagonista al header. En algunos casos éste mantiene el color del resto de la página.

**CADA SITIO WEB
TIENE UN OBJETIVO
COMUNICACIONAL, Y
TODOS SUS ASPECTOS
VISUALES DEBEN
GARANTIZAR ESE FIN.**

visual a la distribución del contenido en todas las páginas. Por supuesto que también podremos tomarnos pequeñas libertades de diseño para darle un ritmo especial a la navegación.

ENCABEZADO O HEADER

El **header** es uno de los aspectos principales que encontramos en un sitio web, ya que es lo primero que vemos. Ocupa todo el ancho superior del sitio y es uno de los elementos que se mantienen casi sin ninguna variación dentro de todas las páginas. Veamos, a continuación, los elementos que pueden estar presentes dentro del header de un site:

► **Logo:** dentro del header se sitúa el logotipo, es decir, la identidad gráfica, el sello distintivo de una empresa. Como es fundamental diferenciarse, el logo ocupa un lugar protagonista en el extremo superior izquierdo. Esto no es casual: está comprobado mediante estudios que éste es el primer lugar donde los usuarios dirigen su mirada. Además de ser un elemento gráfico importante, al logo se le otorga una funcionalidad extra: posee un enlace al inicio del sitio, convirtiéndose en un atajo para que el usuario pueda volver a la página principal. ►



Todo el contenido del sitio está ubicado en una sola página muy larga, que se muestra por partes en la ventana del navegador.

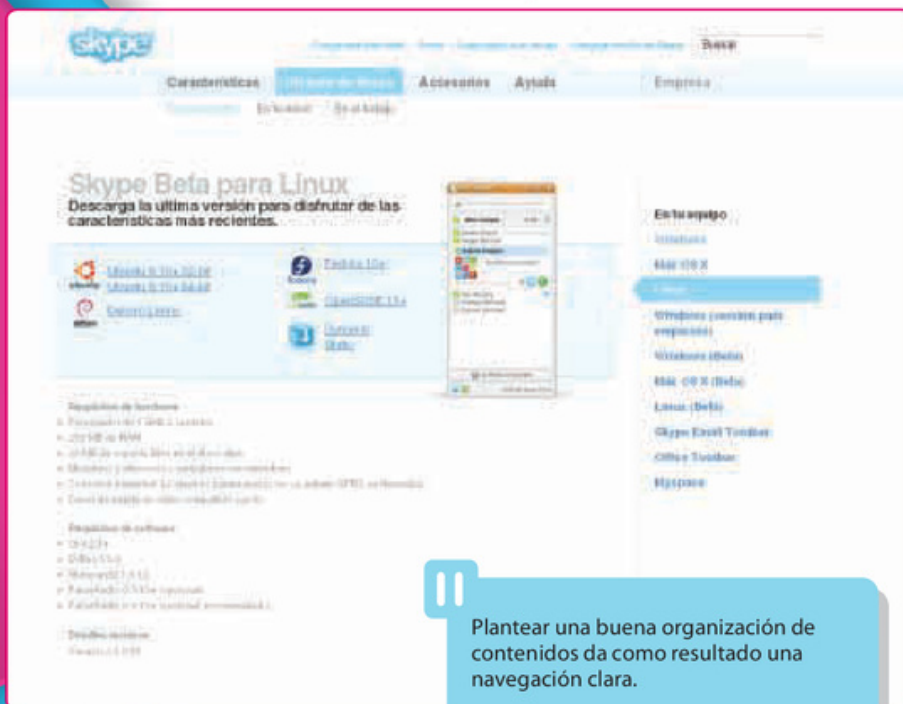
► **Menú:** en el header también encontramos la **barra de navegación**, que contiene los enlaces para acceder a las diferentes secciones del sitio. Los elementos del menú deben estar expuestos de modo sencillo, intuitivo y bien visible para que el usuario tenga acceso rápido a la información y siempre encuentre lo que busca. Cuando un sitio tiene pocos enlaces, todos pueden estar en el menú principal. En cambio, en uno más complejo (de más de seis o siete páginas), aparece el **menú principal** para navegar por las secciones más importantes y, luego, **submenús** para moverse dentro de cada una. Estos submenús deben mantener coherencia temática y lógica en su agrupación. Por ejemplo, si tenemos un sitio de venta de productos de electrónica, a nadie se le ocurriría buscar los productos dentro de la sección Contacto.

► **Menú de rastros:** como mencionamos antes, este menú es recomendable en los sitios que tienen muchas secciones y subsecciones, ya que informa al usuario su ubicación dentro del sitio. Por ejemplo, si estamos en la subsección **Aventuras**, dentro de la subsección **Playstation 3**, que a su vez está dentro de la sección **Videojuegos**, deberíamos de informarle al usuario algo como **Videojuegos/Playstation 3/Aventuras**. Esto representa de una manera visual la jerarquía del sitio y, en la mayoría de los casos, lo encontramos antes del contenido principal.

► **Identificación de secciones:** es importante que el usuario sepa en qué sección se encuentra y que pueda identificarla de inmediato. En combinación con el menú de rastros, la identificación de secciones debería tener mayor relevancia tanto en su ubicación como visualmente, y por lo general, se la coloca por debajo del menú principal de secciones, como encabezado de la sección a la que identifica.

► **Sliders:** este recurso interactivo ha ganado terreno gracias a las nuevas tendencias del diseño 2.0, donde los headers son cada vez más grandes, y la gráfica, cada vez más iconográfica. Los sliders muestran un mensaje o publicidad de forma rápida y eficaz, ya que captan la mirada por su movimiento e interactividad. Para hacerlo, se valen de iconos, colores e información bien jerarquizada, con un diseño simple y atractivo.

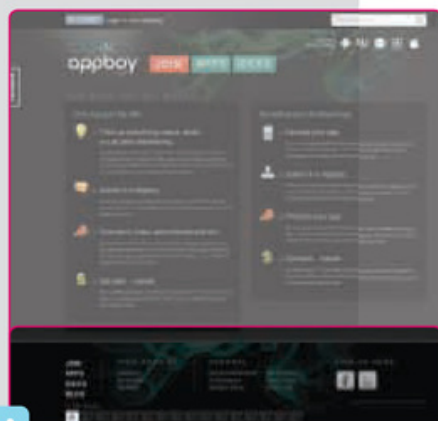
► **Buscadores internos:** en sitios grandes, estos buscadores suelen ubicarse arriba a la derecha o en los sidebars.



Plantear una buena organización de contenidos da como resultado una navegación clara.

Sirven para encontrar de forma rápida algo puntual. Esto quiere decir que no se accede a la información a partir de la navegación, sino que se ingresa una palabra exacta o aproximada a lo que se desea encontrar y se presiona el botón **[Buscar]** o similar para obtener los resultados.

Modernos, prácticos y muy utilizados, los sliders dan dinamismo y transmiten mensajes claros de una manera efectiva.



Actualmente, los footers son algo más que un buen cierre para el sitio. Su uso y actualización marca tendencia, y permite al diseñador ser original con el sitio.

CONTENIDO

Como mencionamos, está comprobado que el recorrido visual que realiza el visitante comienza por el extremo superior izquierdo (donde es usual que encontremos el logo) y luego se fija en el centro de la página, donde está el contenido principal.

Aunque la organización varía entre diseños, veremos denominadores comunes para jerarquizar los diferentes tipos de información. El uso de destacados es un gran punto de tensión, así como también los títulos, que son resaltados mediante el uso de fondos, bullets, iconos, u otros elementos, para luego continuar por la lectura de párrafos o de información secundaria.



COMPRENSIÓN INMEDIATA

No podemos dejar que el usuario tarde mucho tiempo en descubrir qué es lo que le ofrecemos. Si lo hacemos, corremos el riesgo de que cierre el navegador y perdamos nuestra posibilidad de mostrarle lo que queremos que vea, y quizá jamás regrese a visitarnos. La investigación sobre la forma de navegación sólo sería deseable si se tratara de sitios experimentales con interfaces no convencionales, en donde el usuario estaría predispuesto a tener una experiencia diferente.

NO ES **RECOMENDABLE** HACER **INNOVACIONES** EN **ASPECTOS** QUE DIFICULTARÁN **LA** **NAVEGACIÓN** DEL **USUARIO**.

BARRA LATERAL O SIDEBAR

Este elemento gráfico sirve para organizar contenidos importantes del sitio. Se puede ubicar a la derecha del cuerpo principal, a la izquierda o en ambos lados. Contiene enlaces externos e internos, información adicional organizada por categorías y, actualmente, se utiliza también para incluir publicidades (banners, links, etcétera).

PIE DE PÁGINA O FOOTER

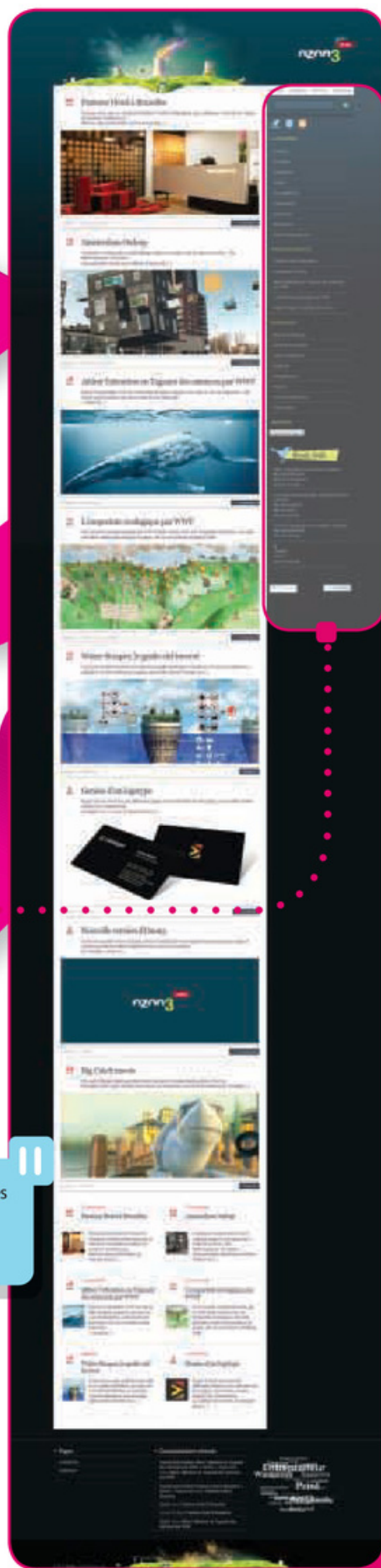
Así como los headers dieron un salto de lo tradicional a lo vanguardista, los footers también siguieron esos pasos. Pasaron de contener información como políticas de privacidad de uso y derechos de autor, a ser contenedores de elementos relevantes, como enlaces, mapas de sitios (para ayudar a la indexación en Google), información y formularios de contacto, y los infaltables links a las redes sociales, que nos invitan a unirnos, compartir, ver perfiles y hasta visualizar los últimos posts de Twitter.

Todos los elementos mencionados determinan lo que llamamos una **interfaz web**. Los contenidos, la forma

Los sidebars son infaltables en las redes sociales y en los blogs, donde su rol es claramente organizativo, debido a la gran cantidad de información que se necesita comunicar.

de navegación, los elementos de identificación y las acciones que podemos generar dentro de un sitio son parte de su interfaz.

Para transmitir un mensaje claro y de forma sencilla, al crear una interfaz debemos respetar las convenciones de estructura. En los próximos capítulos veremos cómo aplicar estos principios en un diseño real. ◀



GUÍA VISUAL

Para comprender mejor cuáles son las partes que componen la estructura de un sitio, analicemos un ejemplo de forma detallada en la siguiente guía visual.

1. LOGO

El logo es un punto de tensión por el cual se entra en el sitio y se regresa a la página principal. Luego de esto, debemos guiar la mirada del usuario, trazando un recorrido visual por la información que tiene más relevancia. Para hacerlo, podemos valernos de destacados, títulos, gráficos, etcétera.

2. MENÚ PRINCIPAL

El menú es la herramienta de navegación más importante del sitio y debe ser adecuada para que ningún visitante naufrague, perdiéndose en menús y submenús desorganizados o poco claros.

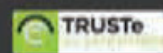


FOOTER

Careers | Online Privacy Policy | Terms of Use | Contact us | Accessibility | Report piracy | Permissions and trademarks | Product license agreements | Send feedback

Copyright © 2010 Adobe Systems Incorporated. All rights reserved.
Use of this website signifies your agreement to the Terms of Use and Online Privacy

Search powered by Google™



CONTENIDO

News

Listen to the Adobe Q3 FY2010 earnings conference call

Adobe Flash Player 10.1 released to device and mobile platform partners

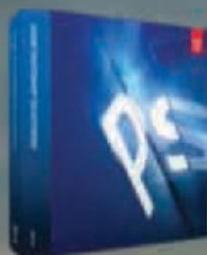
Registration now open: Adobe MAX 2010, October 23-27 in Los Angeles, California

Photoshop Lightroom 3 now available: Simply photography from shoot to finish

WIRED uses digital magazine solution from Adobe to deliver magazine on the iPad

Adobe helps you create without limits

HEADER



Photoshop CS5 Extended is here.

Learn more > Try now > Buy now >

Search

4

here.

Replay

HEADER

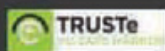
AYER

ITE 5
DARD

g print
the

Try now >

license agreements | Send feedback



6

3. MENÚ SECUNDARIO

Se utiliza para no sobrecargar el menú principal. En general, son enlaces con un nivel diferente de jerarquía o atajos a funciones muy utilizadas.

4. BUSCADOR

Permite encontrar información mediante palabras clave ingresadas en el cuadro de búsqueda. Muchas veces está provisto por **Google**.

5. SLIDERS O VIDEOS

Dentro del slider podemos mostrar imágenes y texto, el usuario los puedes cambia al hacer clic. Transmiten muchos contenidos en poco espacio.

6. DESTACADO

Este espacio se asigna a la promoción de elementos o productos que requieran una mención especial.

SIDEBAR

Ayuda a organizar los contenidos importantes del sitio. Se ubica en los laterales, y contiene enlaces, información adicional e, incluso, publicidad.

Products and solutions

Products

ESTRUCTURA Y DISEÑO DE LA PÁGINA

Para visualizar de forma sencilla y rápida la estructura del sitio, la diagramación del contenido y la relación entre sus diferentes partes, realizaremos esquemas para tener una referencia visual y sintética de todo.

Luego de conocer las necesidades del cliente y de confeccionar el brief, es conveniente realizar una serie de esquemas que expresarán, de forma gráfica, el relevamiento llevado a cabo en la etapa anterior.

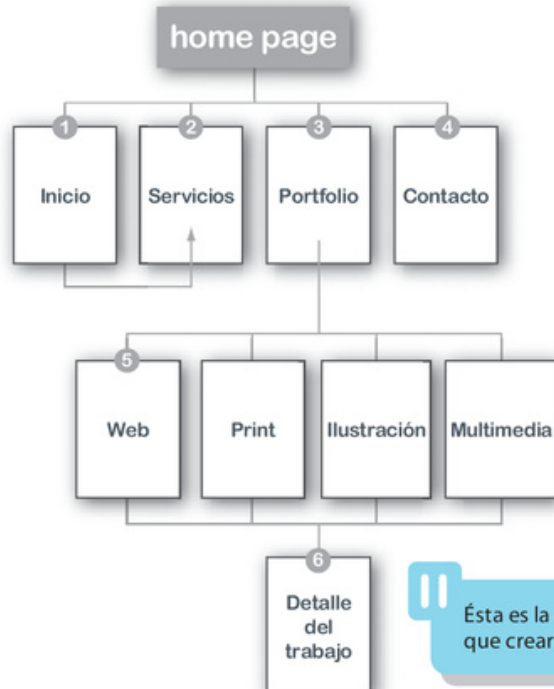
LA ESTRUCTURA

Lo primero que suele hacerse es un esquema con la **estructura del sitio**, donde se definen las diferentes páginas que están involucradas con la navegación y las conexiones entre ellas, pero no se especifican los contenidos de cada

una. De este modo, podemos tener una referencia rápida del tamaño, la profundidad y la complejidad de cada proyecto web que desarrollaremos. Para tener una idea más clara sobre las estructuras, podemos analizar algunos casos que corresponden a sitios de diferente magnitud. En el que desarrollaremos en esta obra, podemos notar que para visualizar el ejemplo de un trabajo, el usuario primero tiene que ir a Portfolio y, luego, seleccionar qué tipo de material quiere ver, donde tendrá la posibilidad de acceder al

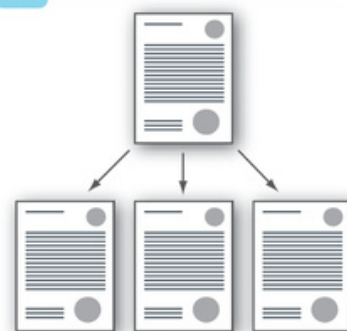
detalle correspondiente.

Si imaginamos la realización de la estructura de sitios del tamaño de Amazon, podremos darnos una idea de la cantidad de páginas y links que contienen, un esquema que resulta enorme. Un paso interesante que podemos dar para comprender la importancia que tiene la definición temprana de la estructura de un sitio es visitar **Webpages as Graphs (www.aharef.info)**, que muestra, de una manera visual muy atractiva, la estructura de cualquier sitio que deseemos.



Ésta es la estructura del sitio que crearemos en este libro.

Estructura en árbol, con jerarquías muy marcadas.



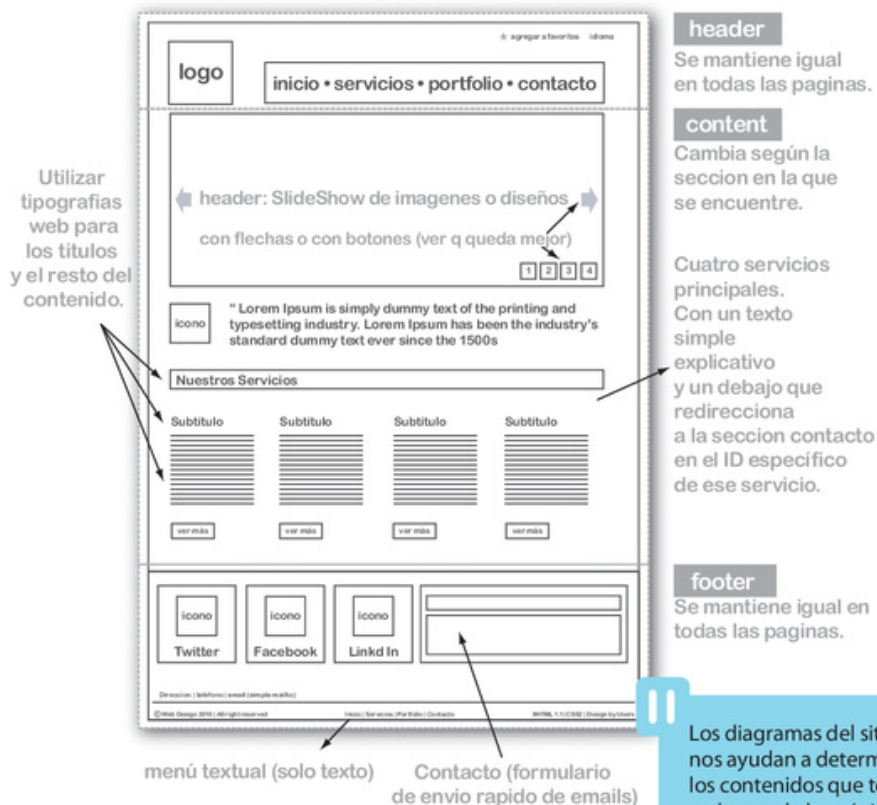
ANTES DE COMENZAR A DISEÑAR, ES RECOMENDABLE TENER LA **APROBACIÓN FINAL DEL CLIENTE**, PARA NO SUFRIR CON **LOS CAMBIOS**.

ESTRUCTURA Y NAVEGACIÓN

La confección de la estructura de un sitio no solamente nos permite disponer de una referencia visual de sus partes y la cantidad de páginas, sino que también nos provee de un método a través del cual podemos determinar el tipo de navegación y la manera en que los usuarios recibirán la información del sitio. Podemos elegir entre diferentes tipos de estructuras:

- **Estructura en árbol:** compuesta por una página principal desde la que se accede a otras que, a su vez, dan acceso a otras páginas en niveles inferiores.
- **En lista:** no existe una página principal sino que se accede a una página y luego a la siguiente en forma lineal, ya que todas tienen la misma profundidad.
- **En red:** todas las páginas tienen acceso entre ellas sin tener en cuenta la profundidad en la que se encuentre cada una.

- **Mixta:** esta estructura es una mezcla de las anteriores ya que, a partir de una página principal, se accede a las siguientes, y desde ellas, a otras en su mismo nivel de profundidad.

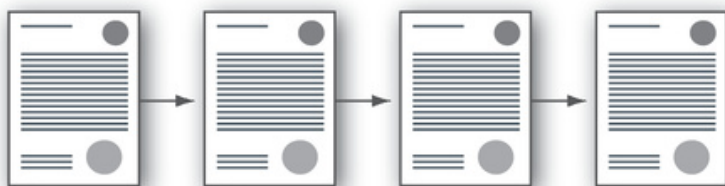


EL DIAGRAMA DEL SITIO

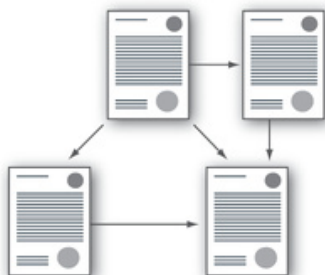
Al profundizar más en el esquema de nuestro sitio, será necesario definir los contenidos de cada página, su ubicación y los niveles de importancia. Para lograr este objetivo, se realiza un diagrama que sirve para presentar al cliente el contenido del sitio de forma específica. De este modo, podremos mostrarle qué tendrá su sitio, y detallarle qué va a pasar y cómo se va a comportar cada uno de los componentes de las diferentes secciones. El punto principal en esta etapa no es definir los textos e imágenes exactas, sino más bien la cantidad de texto y el tipo de imágenes que habrá. La función de estos diagramas es proveer de información general sobre los contenidos, pero no sobre su diseño ni la tecnología que utilizan. Por ejemplo, en ellos podríamos definir una zona de interactividad sin que sea necesario mencionar si será hecha en Flash, en JavaScript o utilizando algún otro método.

Como vimos, los esquemas que realizamos al encarar un proyecto web nos ayudarán a simplificar el trabajo, tanto para manejarnos con colegas como para tratar con nuestro cliente. A partir de ahora, comenzaremos a dar los pasos que llevarán nuestro proyecto a la práctica. ◀

Estructura en lista, sin niveles de profundidad.



Estructura en red, navegación en todo sentido.

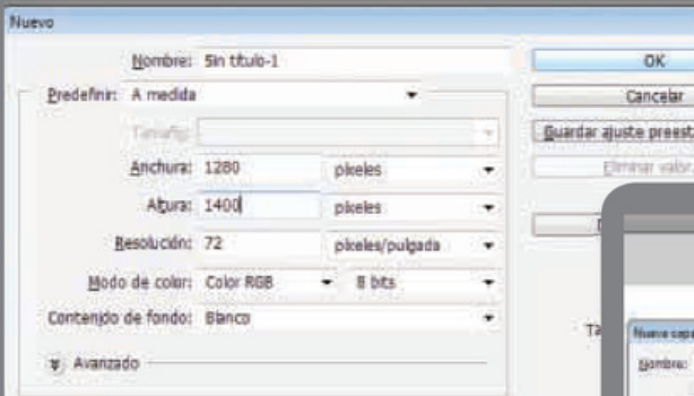


Estructura mixta, donde se combinan varios tipos de navegación.



CREACIÓN DE UN PASO A PASO

En las próximas páginas vamos a explicar todos los pasos que debemos realizar en Adobe Photoshop CS5 para diseñar un sitio de forma profesional.



01 | GENERACIÓN DEL DOCUMENTO



CARPETA DE CAPAS

02

Para organizarnos desde el principio, es recomendable **renombrar** cada una de las capas con una denominación que represente los objetos que se encuentran en ella. No es conveniente dejar nombres como **capa 48** porque, cuando tengamos muchas capas, perderemos bastante tiempo buscando la que contiene el elemento para modificar. Por otro lado, como es muy usual que los diseñadores pasen su archivo PSD para que lo maquete otra persona, es aconsejable que todo esté claro para ganar tiempo y evitar consultas innecesarias. Si queremos mejorar aún más la organización, podemos **agrupar las capas** por carpetas. En cada una ubicaremos todas las capas relacionadas, como podrían ser aquellas que forman parte del menú, las del encabezado o las del pie de la página. Luego de estas recomendaciones, comencemos con la creación del layout de nuestro sitio en Photoshop.

PASO | 01

Para generar un nuevo documento, vamos a **[Archivo/Nuevo]** (<Ctrl + N>) y, en la ventana que aparece, ingresamos las medidas que se muestran en la imagen (1280 píxeles de ancho por 1400 de alto). Luego, para guardar el documento, nos dirigimos a **[Archivo/Guardar como]** y lo almacenamos en el lugar de la PC donde vayamos a trabajar con el proyecto.

PASO | 02

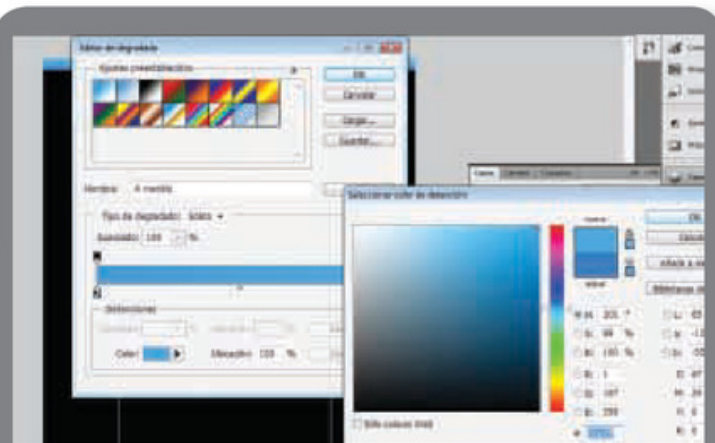
Como es conveniente trabajar de forma organizada, crearemos una carpeta para agrupar todas las capas correspondientes al header. Vamos a **[Capa/Nuevo/Grupo]** y escribimos el nombre que le daremos; en este caso, **[Header]**. A continuación, creamos una capa desde el menú **[Capa/Nuevo/Capa]** (<Ctrl + Shift + N>) y la nombramos para identificarla, tal como hicimos con la carpeta.

PASO | 03

Ahora, comenzaremos a diseñar la parte superior del sitio, es decir, el header.

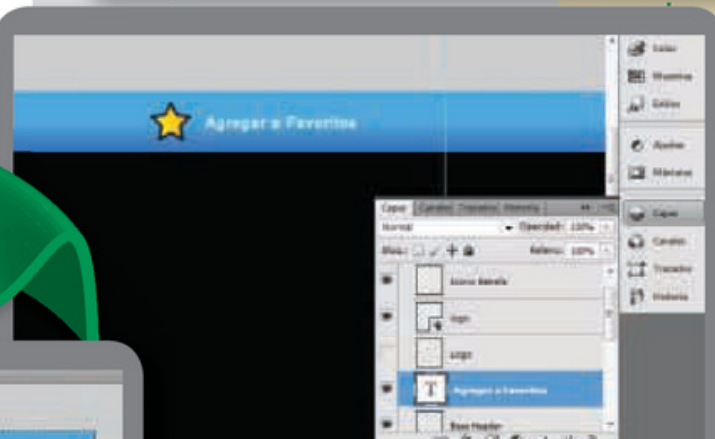
LAYOUT EN PHOTOSHOP

04 | CREAR DEGRADÉ

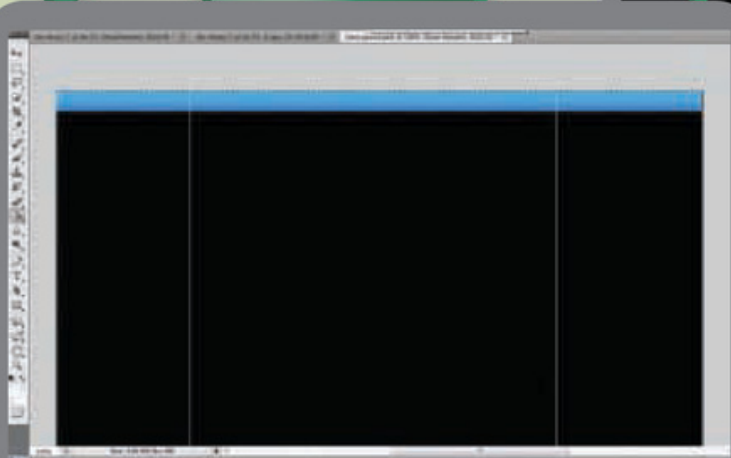


AGREGAR CAPAS

05



03 | DISEÑO DEL HEADER



MENÚ DE NAVEGACIÓN

06

Con la herramienta **[Marco Rectangular]** dibujamos el rectángulo que será la base del header. Para lograrlo, hacemos clic y arrastramos el mouse con el fin de delimitar su superficie.

PASO | 04 En la barra de herramientas, elegimos la correspondiente para hacer degradés y, luego, hacemos clic en la parte superior para editar sus colores. En este caso, ingresamos los tonos **#0078e7** (el más oscuro) y **#01a7ff**. Luego, usando la herramienta, arrastramos el mouse de arriba hacia abajo para pintar la figura.

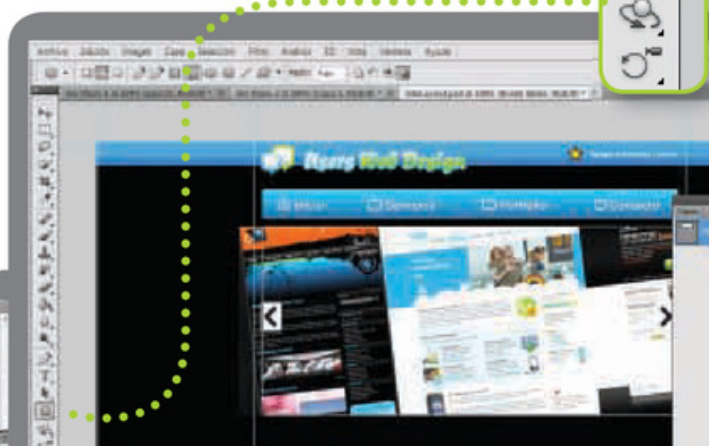
PASO | 05 Colocamos el logo en una nueva capa, y en otra, el icono que hayamos elegido para que el sitio sea agregado a la carpeta **[Favoritos]**. Hacemos clic en la herramienta **[Texto]** y escribimos **[Agregar a Favoritos]** al lado del icono.

A continuación del texto anterior, escribimos **[Idioma]** para completar el header.

PASO | 06 Para comenzar con la creación del menú de navegación, generamos un nuevo grupo de capas para trabajar dentro de él. Tomamos la herramienta **[Rectángulo Redondeado]** y trazamos la base del menú. Luego, aplicamos el mismo gradiente que le dimos a la base del header. Añadimos los iconos pertenecientes a cada botón y, después, con la herramienta **[Texto]**, escribimos el nombre que identifica a cada sección. ▶

CREACIÓN DE UN LAYOUT

PASO A PASO EN PHOTOSHOP



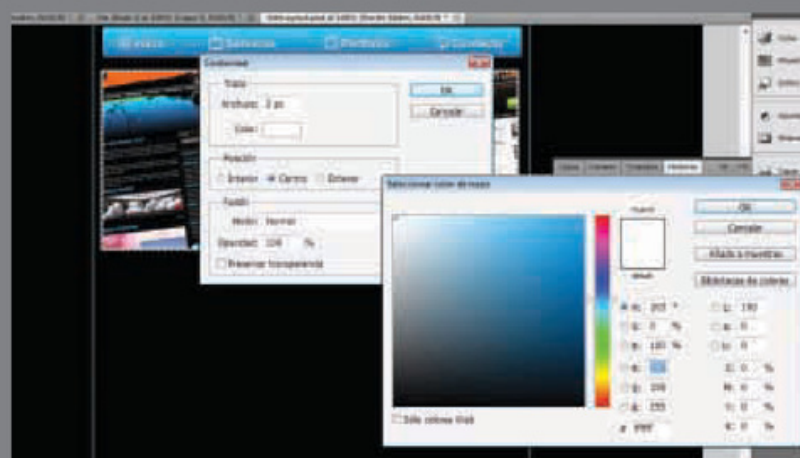
07 POSICIÓN DE LOS SLIDERS



08 APLICAR MÁSCARA

PARA TENER EN CUENTA...

En este libro nos centramos en los programas Flash y Dreamweaver. En este paso a paso en particular utilizamos Photoshop, y el lector deberá definir algunos parámetros según su criterio personal para seguir adelante, ya que sería imposible cubrir todas las opciones. Quienes tengan dudas podrán enviar su consulta al Profesor en línea.



TRAZAR BORDES | 09

PASO | 07 Vamos a indicar el lugar de los sliders mediante una imagen y una máscara con bordes redondeados. Para hacerlo, seleccionamos la herramienta **[Rectángulo Redondeado]** y definimos el contenedor. Veremos que en la ventana **[Trazados]** aparece una capa con esa forma delimitada.

PASO | 08 Para convertir el trazado en una selección, hacemos clic sobre él con el botón derecho y, del menú que se abre, elegimos **[Hacer Selección]**. Luego de aceptar la propuesta, notaremos que el perímetro de la selección queda delimitado con una línea punteada. Para aplicar la máscara, seleccionamos la capa que tiene la imagen y presionamos el tercer icono de la ventana **[Trazados]**.

PASO | 09 Para trazar el borde de los sliders, presionamos la tecla **<Ctrl>** y pulsamos el mouse sobre la máscara creada en el paso anterior. De esta forma, quedará seleccionado el perímetro del slider. A continuación, vamos a **[Edición/Contornear]** e ingresamos 3 píxeles para el ancho del contorno y **#ffffff** como color.

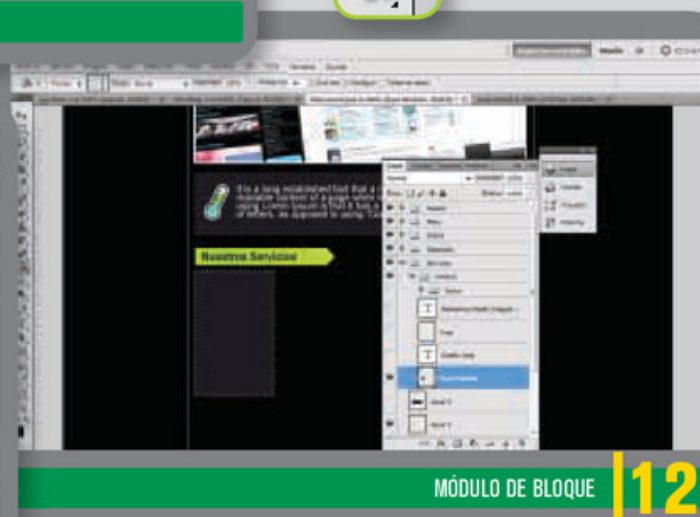
PASO | 10 Vamos a crear el área del texto destacado, para lo cual definimos una nueva forma rectangular de la misma manera en que lo hicimos con el header. En este caso, utilizamos la herramienta **[Marco Rectangular]** con el color **#1a171b**. Agregamos el icono que hayamos elegido para los destacados y, con la herramienta **[Texto]**, trazamos un rectángulo y pegamos el texto correspondiente.



10 | ÁREA DE TEXTO



11 | FONDOS



MÓDULO DE BLOQUE | 12



BASE DE MÓDULO | 13

PASO | 11 Para poner el fondo del título Nuestros Servicios, nos dirigimos a **[Archivo/ Colocar]** y seleccionamos la imagen que queremos usar. La movemos hasta su ubicación y presionamos la tecla **<Enter>**. Luego, seleccionamos la herramienta **[Texto Horizontal]** y escribimos el título.

PASO | 12 Ahora crearemos un módulo para el bloque de servicios y lo duplicaremos para colocar las diferentes ofertas de productos. Para lograrlo, trazamos la base y la pintamos del mismo color que la del destacado, como vimos en el **paso 11**.

PASO | 13 Para hacer el detalle de la base, definimos un triángulo con la herramienta **[Pluma]**, haciendo un clic

para establecer cada vértice. Realizamos la selección en la ventana **[Trazados]** como ya hicimos anteriormente y lo eliminamos de la base del módulo con la tecla **<Supr>**.



CREACIÓN DE UN LAYOUT

PASO A PASO EN PHOTOSHOP



PASO | 14 Agregamos el título del producto (Diseño Web) y, con la herramienta **[Pluma]**, trazamos una línea divisoria entre el título y el texto del detalle. Le aplicamos el color **#87b015** y le damos un grosor de 1 píxel. Luego, seleccionamos **[Contornear Trazado]** en la ventana **[Trazados]**.

PASO | 15 Después de colocar el texto del detalle, trazamos la base del botón tal como lo hicimos en el menú. Para conseguir extremos más redondeados, aplicamos 50 píxeles de radio. Los colores del gradiente irán de **#1a8cd3** a **#06d9fb**.

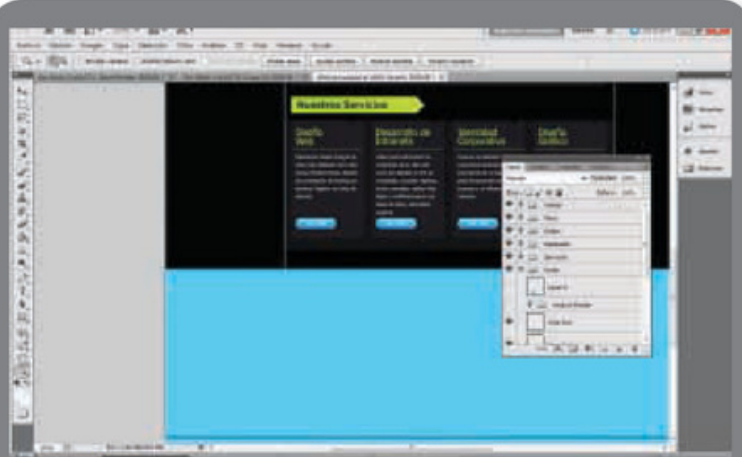
PASO | 16 Para obtener los otros módulos, hacemos clic derecho sobre la carpeta del que hemos creado y elegimos la opción

[Duplicar grupo]. Ubicamos el módulo duplicado y realizamos los mismos pasos para tener cuatro módulos.

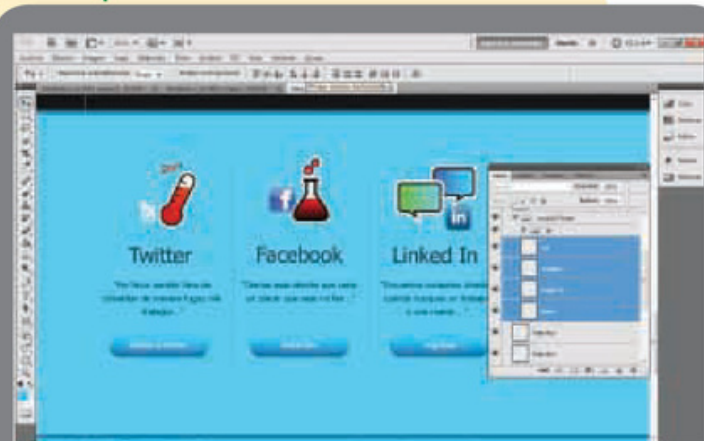
PASO | 17 Para crear el footer, definimos los campos con la herramienta **[Marco Rectangular]** y los pintamos uno por uno. El primero es de 5 píxeles de alto y lleva el color **#047da5**. El segundo tiene un alto de 200 píxeles y el color **#01e2ff**, además de una textura que aplicamos desde **[Archivo/Colocar]**. El tercero es el duplicado de la primera área, y el último, como llevará los detalles del copyright y otros datos, tendrá un alto de 85 píxeles y el color **#36c3ee**.

PASO | 18 La zona para las redes sociales también se manejará como módulos, ya que todas contienen los mismos elementos.

17 | CREAR EL FOOTER



18 | REDES SOCIALES



CAMPOS DE TEXTO | 19



RETOQUES FINALES | 20



Entonces, creamos uno de ellos y, después, lo duplicamos para tomarlo como base, cambiando la información específica de cada red. Para hacer la línea derecha que delimita el módulo, establecemos los dos extremos con la herramienta **[Pluma]** usando el color **#04c7e9**. Para alinear los elementos, seleccionamos todas las capas y, en la barra de herramientas superior, elegimos la opción **[Alinear centros horizontales]**.

PASO | 19 Para hacer los recuadros pertenecientes a los campos de texto, en una nueva capa dibujamos un rectángulo con la herramienta **[Marco Rectangular]** y lo pintamos con el **[Bote de Pintura]** usando el color **#07d6ff**.

PASO | 20 A continuación, realizamos el borde con un color un poco más oscuro para

que se destaque (**#07d6ff**) y bajamos la opacidad de la capa para darle el efecto de transparencia. Hacemos lo mismo para los dos campos restantes y, para finalizar, escribimos los textos correspondientes al copyright y otros datos complementarios que queramos agregar. ◀

RESUMEN

Con lo aprendido, tendremos el diseño del layout bien organizado, tanto para hacerle modificaciones como para enviarlo a maquetar. Por supuesto, hay elementos que se mantendrán estables en las diferentes secciones del sitio, como el header, los botones y el footer. Por lo tanto, cuando diseñemos las páginas interiores, vamos a copiar directamente las carpetas correspondientes a esos elementos fijos.

TEORÍA DEL COLOR

El color es parte de nuestro mundo cotidiano y es, sin dudas, uno de los elementos más importantes dentro de cualquier composición visual.

A la hora de realizar un diseño web, nos encontramos con elementos tanto subjetivos como objetivos, y como profesionales de la comunicación visual, debemos saber de qué manera trabajarlos para que los resultados no queden librados al azar.

EL COLOR COMO FENÓMENO FÍSICO, SENSORIAL Y CONTEXTUAL

Percibimos los colores gracias a la **luz**.

Esto se debe a que, dentro del **espectro electromagnético** conformado por las diferentes longitudes de onda (equivalentes a niveles de energía), el ser humano sólo ve una pequeña porción (denominada **espectro visible**) y distingue niveles que traduce en diversos colores.

Al excitar nuestro sentido de la vista, ocurre un fenómeno de carácter sensorial que posibilita la evocación de **relaciones sinestésicas** al utilizar cierto color. Esta experiencia se da dentro de un cierto contexto que acarrea diferentes significados según las vivencias y la cultura del individuo. Por ejemplo, en la cultura occidental, algunos posibles significados del color rojo son: dinamismo, pasión, amor, vibración, violencia, calor, lujuria y peligro.

Dentro del grupo de todos los colores es posible distinguir tres **subgrupos**. En primer lugar, encontramos los **colores cálidos** (rojo, naranja y amarillo) y los **fríos** (azul y violeta). El verde puede ubicarse como intermedio, ya que se encuentra en el paso de los cálidos a los fríos, o viceversa, y dependiendo del matiz que predomine en su mezcla, se definirá como de uno u otro tipo. Por último, están los **colores neutros**, como el gris, el blanco y el negro.

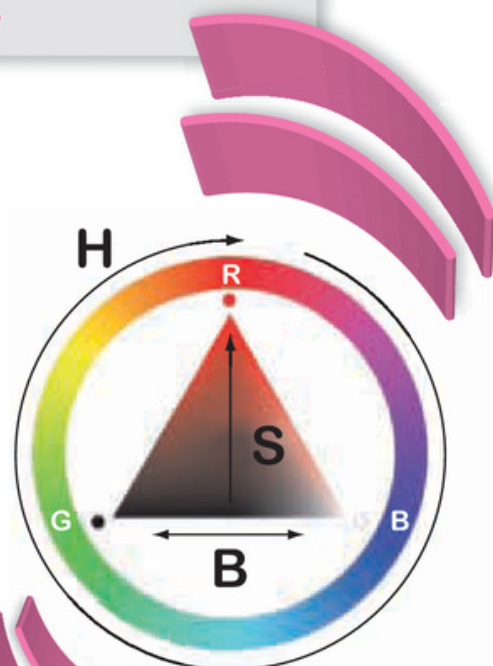
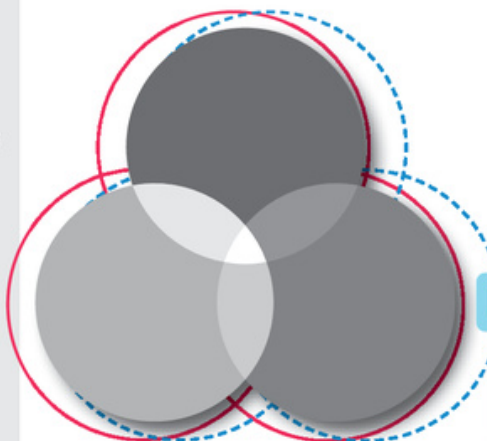
USO Y CONFORMACIÓN DEL COLOR

El color se debe utilizar teniendo en cuenta las características mencionadas previamente, para poder llegar a un buen resultado. Además, debemos saber que si existe una marca que tengamos que usar en nuestro diseño web, los colores que son parte de su sistema de identidad deberán tener presencia para lograr un resultado más armónico, pregnante y coherente.

MODELOS DE COLOR

Los modelos de color describen matemáticamente cómo éstos pueden ser representados. Algunos de los más conocidos son RGB, CMYK, HSB (HSV) y LAB. A continuación, conoceremos los referidos al ámbito del diseño web.

► **RGB**: el nombre deriva de las siglas en inglés de "rojo, verde, azul" (Red, Green, Blue), haciendo referencia a los colores de luces que el modelo toma como primarios. Estos colores, de acuerdo con la intensidad con la que intervengan individualmente, resultarán en un color secundario (**síntesis aditiva**).



El modelo HSB toma su nombre de los términos Hue, Saturation, Bright.

► **HSB (o HSV)**: el nombre de este modelo deriva de las siglas en inglés de "tono (o matiz), saturación y luminosidad (o valor)", haciendo referencia a la relación entre los diferentes colores. El **tono** representa el color que estamos tomando dentro del círculo cromático, y la **saturación**, la intensidad de ese tono (los colores más brillantes son los más saturados). Finalmente, tenemos el **brillo** o valor (brightness o value), que define los tintes (colores más cerca del blanco) o las sombras (colores cercanos al negro) de un mismo tono.

En el sistema RGB, gracias a la síntesis aditiva, los colores primarios superpuestos conforman nuevos colores secundarios. La mezcla de las tres luces genera el blanco.

COLORES EN EL DISEÑO WEB

Por lo general, en los programas que utilizamos a la hora de diseñar, encontramos un panel de selección de color donde podemos elegir valores para el modelo RGB (0-255 para cada uno de los colores) o el HSB (H en grados del ángulo en la rueda de color; S y B en porcentaje, con valores de 0 a 100).

► **Colores hexadecimales:** cuando trabajemos con documentos HTML, CSS, ActionScript o cualquier lenguaje de programación, utilizaremos el código hexadecimal para referirnos a los colores. El sistema hexadecimal es un sistema de numeración posicional en base 16 con el que disponemos de 16 dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F) para formar

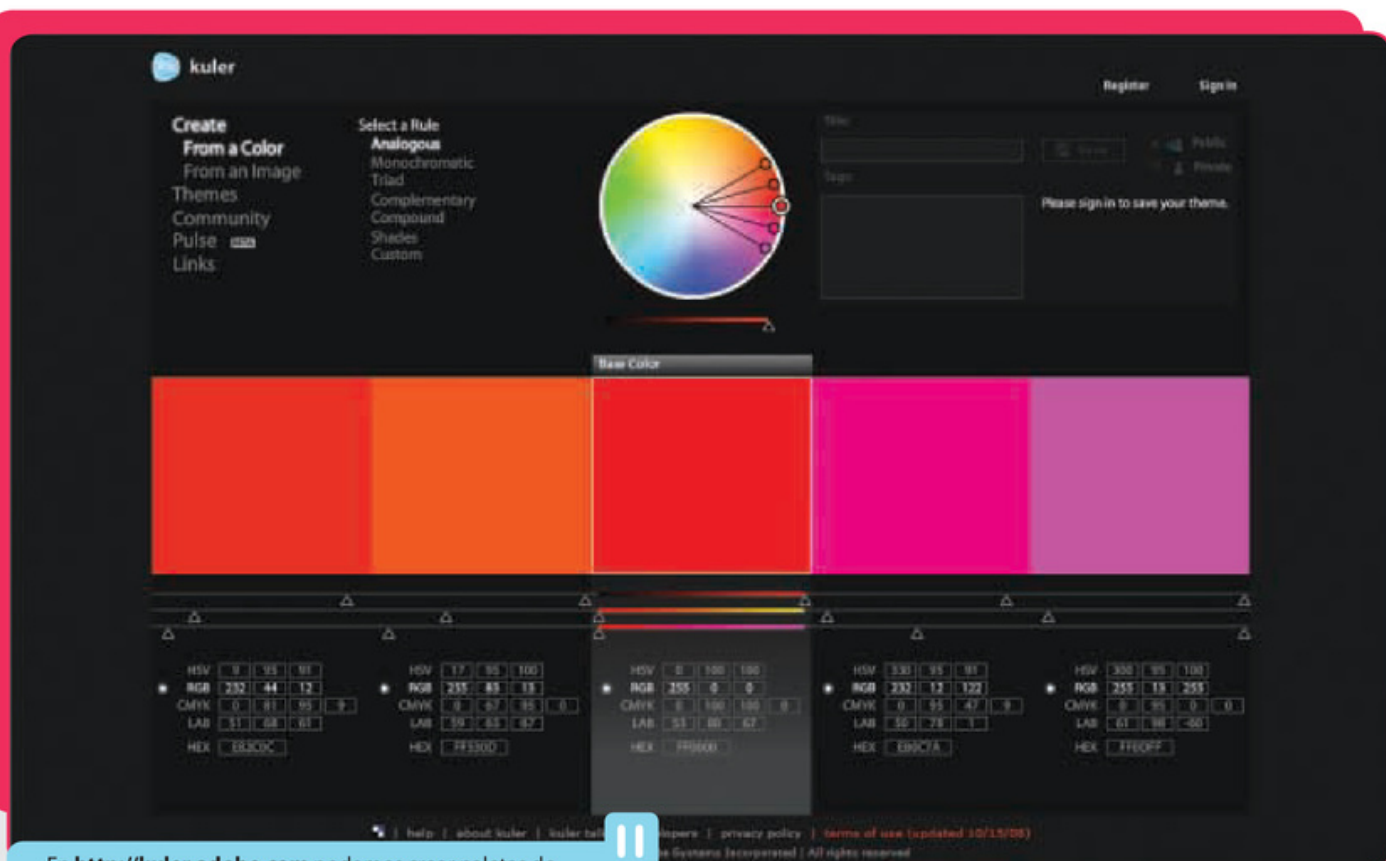
las distintas combinaciones posibles. Seguramente, habremos visto que, para un color azul, por ejemplo, se usa el valor **0000FF**. Éste determina la composición del color en el sistema **RGB** (Red, Green, Blue), constituido por tres pares de

números hexadecimales que nos indican qué cantidad de rojo, de verde y de azul, respectivamente, tiene el tono resultante. Es decir, los dos primeros dígitos determinan el rojo; los dos siguientes, el verde; y los dos últimos, el azul.

Composición del código hexadecimal, donde aparece la mezcla de rojo, verde y azul.

#RRGGBB

Valores hexadecimales que representen el azul del 0 al 255
Valores hexadecimales que representen el verde del 0 al 255
Valores hexadecimales que representen el rojo del 0 al 255



En <http://kuler.adobe.com> podemos crear paletas de colores de diferentes tipos para usar en nuestros diseños.

HERRAMIENTAS ONLINE

A la hora de elegir colores para los diseños, podemos recurrir a nuestra mente o utilizar herramientas que nos ayuden a seleccionar aquellos que generen una buena combinación. Para esta tarea, **Adobe Kuler** (<http://kuler.adobe.com>) es una gran opción. Este sitio nos permi-

tirá buscar esquemas de color creados por otros usuarios o, también, producir nuevas combinaciones. Si queremos crear nuestra propia combinación, en la home page del sitio presionamos [Create], del menú de la izquierda, para comenzar a trabajar. Accederemos al espacio de creación de esquemas de

color, donde, a partir de un color base, podremos obtener el esquema de tonos análogos, un esquema monocromático y el de los complementarios, entre otras posibilidades. Como información de los colores, veremos el modelo empleado, controles deslizantes para modificarlos y el código hexadecimal correspondiente. ◀

CÍRCULO CROMÁTICO EN EL MODELO RGB

El círculo cromático representa de manera visual la conformación de los colores.

Veamos cómo utilizarlo para determinar valores de luminosidad, sombras y saturación. Como mencionamos anteriormente, los colores primarios que conforman los demás tonos dentro del modelo RGB son el rojo, el verde y el azul. El círculo cromático, como podemos deducir a partir del nombre, es representado por una rueda multicolor que delimita cómo se conforma la gama completa de colores de este modelo y las relaciones entre ellos.

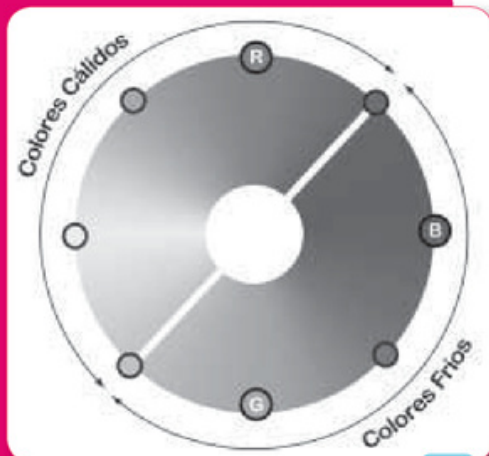
En el círculo cromático RGB podemos visualizar de qué manera los colores pri-

marios juegan en la mezcla y cómo, según a qué hemisferio pertenezca el tono o matiz que utilizemos, estaremos hablando de un color frío o de uno cálido.

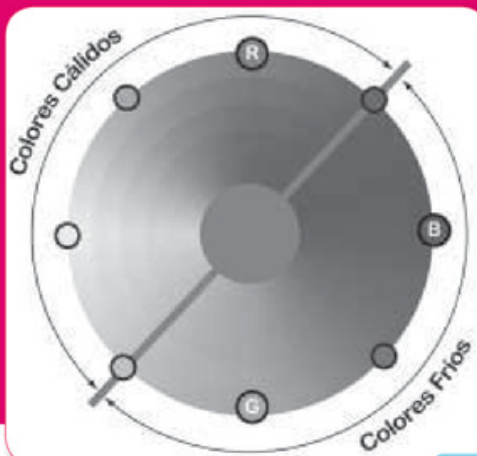
► **Luminosidad:** En el círculo cromático también es posible representar la luminosidad. Para hacerlo, agregamos diferentes ruedas que aumentan el valor (o luminosidad) de los distintos tonos a medida que se acercan al centro. Entonces, si tenemos un rojo furioso de valores máximos (255), conseguiremos sus tintes si aumentamos en la misma medida los otros dos colores primarios. Esto se debe a que, en este modelo, el blanco es la suma de los tres colores primarios en la máxima medida (R:255, G:255, B:255).

► **Sombras:** También podemos agregar al círculo diferentes ruedas que disminuyen el valor (o luminosidad) de los distintos tonos a medida que se acercan al centro. Volviendo al ejemplo del rojo, para conseguir sus diferentes sombras, debemos bajar el valor de R en la mezcla. Esto se debe a que, en este modelo, el negro es igual a la ausencia de los tres colores primarios (R:0, G:0, B:0).

► **Saturación:** En este caso, el círculo está intervenido por diferentes ruedas que disminuyen la saturación de los tonos a medida que se acercan al centro. En este caso, la saturación puede buscar que sus valores sean más bajos (más oscuros) o más altos (más luminosos). ◀



Esta división también puede verse en los programas orientados al desarrollo web dentro de la suite de Adobe.



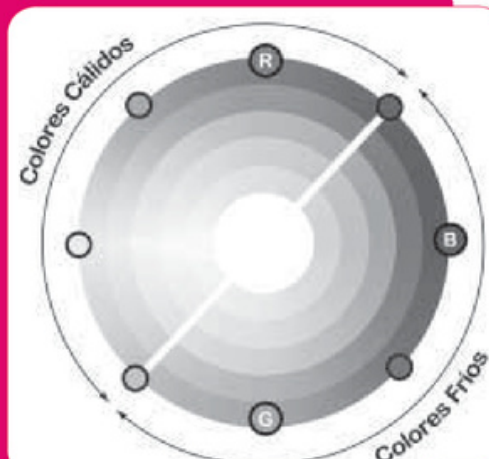
El círculo cromático RGB y su saturación.

R: 255
G: 0
B: 0

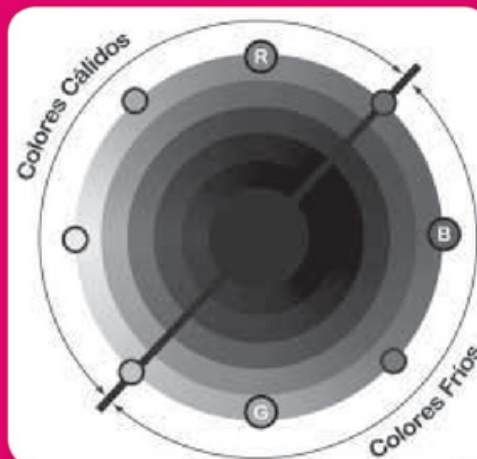
R: 255
G: 95
B: 95

R: 255
G: 176
B: 176

Aquí podemos ver los valores de R, G y B para un color y sus tintes.



El círculo cromático RGB con los tintes de los colores.



El círculo cromático RGB y sus sombras.

R: 255
G: 0
B: 0

R: 147
G: 0
B: 0

R: 63
G: 0
B: 0

Valores de R, G y B de un color y sus sombras.

TIPOGRAFÍA

En un diseño web, además de participar en el estilo, las tipografías son una parte determinante de la legibilidad del sitio. Por eso, es importante conocer qué factores debemos tener en cuenta a la hora de elegir las.

Cuando hablamos de tipografía, nos referimos a los caracteres que conforman un texto (letras, números o símbolos). Quienes trabajan con tipografías (tipógrafos) lo hacen con sus características visuales, como la forma, el peso, el tamaño, la relación entre los diferentes caracteres y la conformación de bloques de texto. En esta oportunidad, nos abocaremos a conocer, puntualmente, lo que significa la tipografía en los medios digitales. Como bien sabemos, las pantallas fueron posteriores a los escritos y a los impresos, y las leyes para tratar la tipografía no son las mismas en ambos casos. A continuación, veremos por qué sucede esto y cómo manejar las tipografías para que nuestros diseños tengan ciertos cuidados estéticos que influirán en el contenido y, como consecuencia, en el usuario.

TEXTOS EN LA WEB VERSUS TEXTOS EN PAPEL

Para empezar, analizaremos una tabla con características para conocer algunas de las diferencias que generan los soportes sobre los que se utilizan tipografías (digital o analógico). Teniendo en cuenta este punteo básico de las diferencias entre el soporte web y el tradicional, iremos viendo que los conceptos básicos que influyen en cualquier puesta tipográfica se aplicarán en la Web de otra manera.

LEGIBILIDAD Y CONTENIDO DE LOS TEXTOS

El medio digital es siempre menos legible que el analógico. Para comprender este concepto, comencemos por analizar a los que leen: los usuarios. Si pensamos en la manera en la que recorremos cualquier tipo de artículo en la Web, nos daremos cuenta de que, en realidad, no leemos, sino que escaneamos, lo que significa que

	PAPEL	WEB
▶ CANVAS	Fijo	Indefinido; hay diversos dispositivos y formatos de pantalla
▶ RESOLUCIÓN	Mayor (esto permite caracteres más definidos)	Menor
▶ UNIDAD MÍNIMA	El punto	El píxel
▶ SÍNTESIS DEL COLOR	Sustractiva	Aditiva
▶ SECUENCIA DE LECTURA	Lineal	Hipertextual
▶ LEGIBILIDAD DE TEXTOS	Mayor	Menor
▶ USO DE TIPOGRAFÍAS	Ilimitado	Restringido

Diferentes características según el soporte en el que estemos trabajando.

recorremos rápidamente las páginas de los sitios buscando la información que nos resulte útil o que llame nuestra atención. Esto se debe a que la secuencia de lectura es hipertextual (buscamos links o conexiones con otros contenidos relevantes al tema), con saltos y ramificaciones; sin mencionar que el soporte del texto es una fuente luminosa y tiene una resolución menor, que agota más rápidamente nuestros ojos. Entonces, además de elegir una tipografía correcta, debemos trabajar con otros factores de forma y de contenido:

- ▶ **La forma:** el espacio blanco entre los caracteres y las palabras (la interlínea, la interletra, el ancho de línea, el tamaño de los caracteres, etc.).
- ▶ **El contenido:** cada párrafo deberá cerrar una idea (independientemente del total del contenido). La jerarquía de la información y los niveles de lectura también son importantes, y la cantidad de información que utilizaremos deberá ser un 50% menor a la que emplearíamos en algún medio no digital. ▶

TYPOGRAPHY

LAS TIPOGRAFÍAS O FUENTES SEGURAS

Antes de elegir una tipografía para un diseño web, es importante saber que no todas las computadoras tienen las mismas fuentes instaladas. Entonces, el navegador interpreta los textos dispuestos en una página con las fuentes que el usuario tiene disponibles en su sistema operativo. Por esta razón, en los diseños web se utilizan las fuentes más comunes en todos los equipos: Arial, Arial Black, Comic Sans, Georgia, Impact, Times New Roman, Trebuchet, Verdana y Webdings. Nuestro diseño dependerá, en gran medida, de la elección que hagamos dentro de este grupo de tipografías, que podemos clasificar como serif, sans serif y de fantasía.

► **Serif:** son aquellas que tienen remates tanto inferiores como superiores. Son muy legibles en papel, pero en

medios digitales dificultan la lectura, sobre todo, en cuerpos pequeños, ya que sus remates son mal interpretados por falta de **antialiasing** (suavizado en los bordes). Ejemplos de este tipo son las tipografías Georgia y Times New Roman.

► **Sans Serif:** son aquellas que no tienen ningún tipo de remate, y son más legibles en los medios digitales, ya que la falta de antialiasing afecta en menor medida a la legibilidad de los textos. Ejemplos de este tipo son Arial, Arial Black, Comic Sans, Impact, Trebuchet y Verdana.

► **Fantasía:** son aquellas que no entran en ninguna de las dos categorías anteriormente mencionadas. Suelen ser ilustraciones, símbolos o signos, y no es recomendable usarlas para la redacción de textos. Ejemplo de este tipo es la Webdings.

Times New Roman 12 pt

Times New Roman 10 pt

Times New Roman 8 pt

Times New Roman 6 pt

Times New Roman 12 pt

Times New Roman 10 pt

Times New Roman 8 pt

Times New Roman 6 pt

Tipografía serif sin (izquierda) y con (derecha) antialiasing en diferentes cuerpos.

Verdana 12 pt

Verdana 10 pt

Verdana 8 pt

Verdana 6 pt

Verdana 5 pt

Verdana 12 pt

Verdana 10 pt

Verdana 8 pt

Verdana 6 pt

Verdana 5 pt

Tipografía sans serif sin (izquierda) y con (derecha) antialiasing en diferentes cuerpos.

EL PÁRRAFO, LA LÍNEA Y LA PALABRA

La tarea del diseñador web generalmente no está asociada a la creación de una tipografía desde cero (salvo raras ocasiones), sino que está relacionada con palabras, líneas y párrafos.

A la hora de plantear un trabajo, el diseñador no suele tener los textos definitivos y reales. En estos casos, para presentar un modelo, debe emplear un **generador de texto**, que brinda material para presentar y elegir tipografías y estilos. El generador más conocido es **Lorem Ipsum** (www.lipsum.com), muy recomendado para los textos en español

Bit, conseqi tem vitatem quae consectotat parupta spernatis ressi as autae nonsequaspis a sunda comnisi magnatus. Boreseque nia quas magnate porepellupta dolorep edictatur? Harum ex et eatemquiam, que eaqui omnihi lignatures et quatem res volorit qui re sit, non esequo temo et as dolum ne poresciis nihilliquod quis maximpos ex eum lam accus. Tem voluptam qui si acea aut ent eos aligent magnatus ende et eum, optatem ratiuntem quiatem et ulparist, tem nonsequi iliqui cum es sum es apera venti idelia quia qui dio bla nemquam sam esequi conem. To il mint, nime.

Bit, conseqi tem vitatem quae consectotat parupta spernatis ressi as autae nonsequaspis a sunda comnisi magnatus. Boreseque nia quas magnate porepellupta dolorep edictatur? Harum ex et eatemquiam, que eaqui omnihi lignatures et quatem res volorit qui re sit, non esequo temo et as dolum ne poresciis nihilliquod quis maximpos ex eum lam accus. Tem voluptam qui si acea aut ent eos aligent magnatus ende et eum, optatem ratiuntem quiatem et ulparist, tem nonsequi iliqui cum es sum es apera venti idelia quia qui dio bla nemquam sam esequi conem. To il mint, nime.

Bit, conseqi tem vitatem quae consectotat parupta spernatis ressi as autae nonsequaspis a sunda comnisi magnatus. Boreseque nia quas magnate porepellupta dolorep edictatur? Harum ex et eatemquiam, que eaqui omnihi lignatures et quatem res volorit qui re sit, non esequo temo et as dolum ne poresciis nihilliquod quis maximpos ex eum lam accus. Tem voluptam qui si acea aut ent eos aligent magnatus ende et eum, optatem ratiuntem quiatem et ulparist, tem nonsequi iliqui cum es sum es apera venti idelia quia qui dio bla nemquam sam esequi conem. To il mint, nime.

Bit, conseqi tem vitatem quae consectotat parupta spernatis ressi as autae nonsequaspis a sunda comnisi magnatus. Boreseque nia quas magnate porepellupta dolorep edictatur? Harum ex et eatemquiam, que eaqui omnihi lignatures et quatem res volorit qui re sit, non esequo temo et as dolum ne poresciis nihilliquod quis maximpos ex eum lam accus. Tem voluptam qui si acea aut ent eos aligent magnatus ende et eum, optatem ratiuntem quiatem et ulparist, tem nonsequi iliqui cum es sum es apera venti idelia quia qui dio bla nemquam sam esequi conem. To il mint, nime.

Comparación de párrafos alineados a la izquierda y justificados. En los justificados, se pueden observar los ríos que se forman.



ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 Mencione las habilidades principales del diseñador web.

02 ¿Qué herramientas debe conocer un diseñador web?

03 ¿Cuáles lenguajes permiten crear sitios para Internet?

04 ¿Qué recursos generan un recorrido de lectura del sitio?

05 ¿Qué es una **grilla**?

06 ¿Cómo definiría el lenguaje **CSS**?

07 ¿Qué diferencia a un sitio estático de uno dinámico?

08 ¿Cuáles son las ventajas de **Wordpress**?

09 ¿Qué tipo de datos se colocan en una **coming soon page**?

10 ¿En qué reside la importancia del posicionamiento **SEO**?



PRÁCTICAS

01 Cree una **coming soon page** en Adobe Photoshop con el objetivo de que el visitante regrese pronto.

02 Diseñe en Adobe Photoshop una página de **error 404**.

03 Haga una estructura de su sitio.

04 Diseñe el diagrama de su página.

05 Cree un **layout** para su sitio en Adobe Photoshop.





Herramientas de trabajo



- ▶ ADOBE CS5 - WEB PREMIUM®
- ▶ HERRAMIENTAS PARA DESARROLLAR HTML
- ▶ SELECCIÓN DE LAS HERRAMIENTAS DE DISEÑO
- ▶ ADOBE PHOTOSHOP CS5
- ▶ ADOBE ILLUSTRATOR CS5
- ▶ ADOBE CS5 DREAMWEAVER

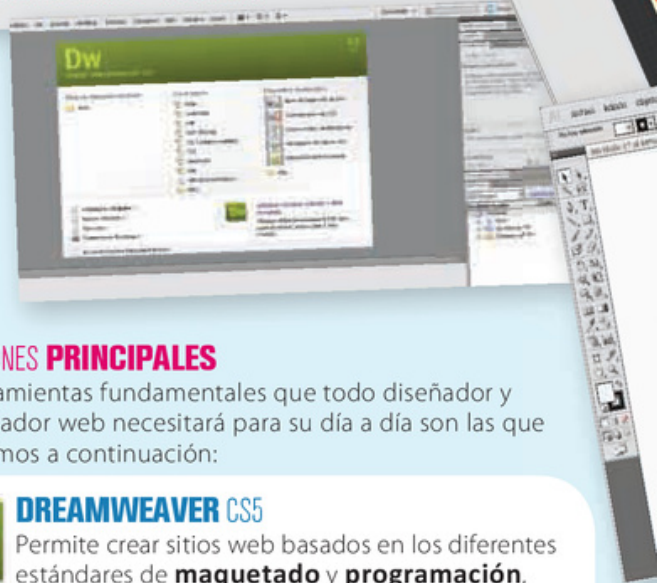


ADOBE CS5

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 ésta 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 enunciamos a continuación:



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.



FLASH PROFESSIONAL CS5

Éste 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 cada vez más sobresalientes en sus herramientas de creación y con un lenguaje de programación cada vez más potente: ActionScript 3.0.



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.



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:



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.



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.



FLASH BUILDER

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



CONTRIBUTE CS5

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



¿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 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.

HERRAMIENTAS PARA DESARROLLAR HTML

Como ya sabemos, el código básico de Internet es el HTML, por lo que una vez que tengamos nuestro sitio diseñado, necesitaremos transferir ese diseño a líneas de código que puedan ser interpretadas por los navegadores de los usuarios.

El proceso de crear páginas se denomina, **maquetación**, en la jerga, y da como resultado tener nuestro diseño convertido en un sitio navegable. Luego, en caso de que el proyecto lo requiera, faltará la adición de código JavaScript, AJAX, ASP o PHP para hacer que las páginas dinámicas obtengan contenido de las bases de datos.

Un problema que suele presentarse al desarrollar HTML es lograr que el diseño sea visualizado de igual forma en la mayoría de los navegadores de Internet.

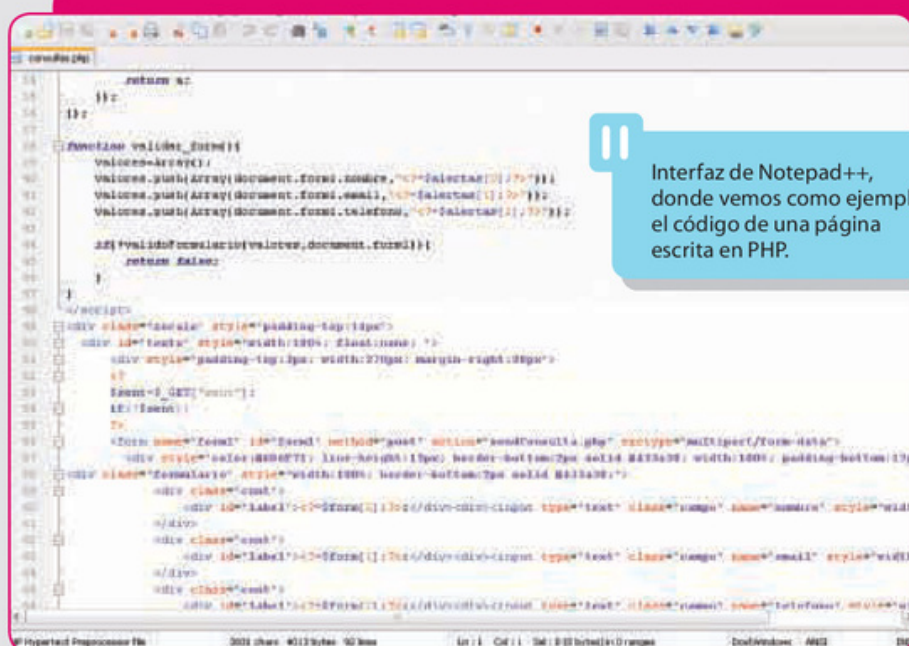
Este inconveniente se debe, principalmente, a que algunos fabricantes no han incorporado en sus productos los estándares que determinan la estructura de la Web.

En consecuencia, tendremos que aplicar **parches** en el código CSS y HTML para que la maqueta sea compatible con los navegadores más frecuentes: Internet Explorer 6, 7 y 8; Mozilla Firefox 3.5 y 3.6; Chrome 5.0 y Safari 4.0. Esta práctica se denomina

cross-browsing. Es importante distinguir entre las distintas versiones de cada navegador, ya que cada una interpreta el código de forma diferente. Si nos aseguramos de que nuestro sitio sea compatible con los siete navegadores mencionados, el 96% de los usuarios de Internet podrá ver, sin problemas, el trabajo que estamos desarrollando.

HERRAMIENTAS PARA GENERAR CÓDIGO HTML Y CSS

La aplicación más potente del mercado para generar código HTML es Adobe Dreamweaver CS5, que estudiaremos en las próximas páginas. Sin embargo, además de ella, existen otras que podemos utilizar para redactar y generar código



CSS y HTML para nuestros sitios.

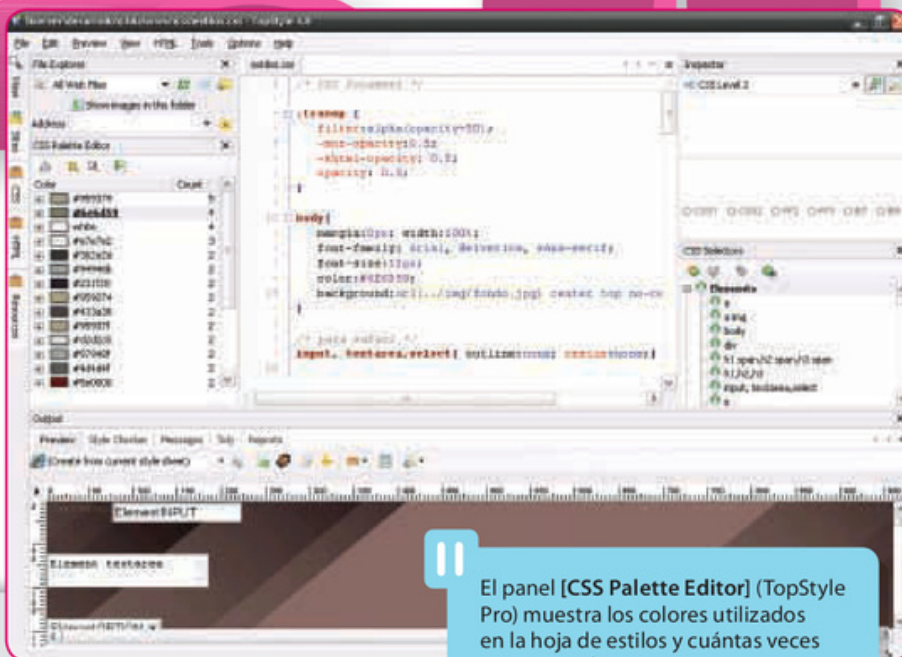
Si bien es cierto que cualquier editor de texto plano, como el Bloc de notas de Windows, es suficiente para redactar código HTML y CSS, nos convendrá recurrir a programas diseñados especialmente para tal fin, ya que en ellos tendremos ayudas visuales (como el uso de distintos colores para el texto) que nos orientarán en la tarea y simplificarán el trabajo.

NOTEPAD++

Notepad++ (<http://notepad-plus-plus.org>) es una aplicación que bien podríamos definir como una versión muy mejorada del editor de texto o el Bloc de notas incluido en todos los sistemas Windows. Si no estamos acostumbrados a redactar código

de programación web **a ciegas**, no será conveniente utilizarlo, ya que no cuenta con un preview incorporado. De todas maneras, desde un simple atajo de teclado, podemos visualizar el HTML en los navegadores de Internet que tengamos instalados. Entre los principales puntos a favor de esta aplicación encontramos los siguientes:

- ▶ Es gratuita.
- ▶ Sintaxis coloreada.
- ▶ Numeración de línea.
- ▶ Muy liviana, consume pocos recursos del sistema.
- ▶ Código **autocompletable** en diferentes lenguajes.



El panel [CSS Palette Editor] (TopStyle Pro) muestra los colores utilizados en la hoja de estilos y cuántas veces aparece cada uno.

► Es posible abrir y gestionar archivos de texto pesados (por ejemplo, logs de acceso a servidores) sin perder estabilidad.

TOPSTYLE PRO

TopStyle Pro (www.topstyle4.com) es uno de los mejores editores de código CSS y, en general, es recomendable utilizarlo como complemento del editor de HTML principal que empleemos para trabajar. Entre sus principales ventajas podemos mencionar:

- Preview del código escrito en tiempo real.
- Validador de estilos CSS.
- Sintaxis autocompletable del código que estamos escribiendo.

¿POR QUÉ DREAMWEAVER?

La elección de Adobe Dreamweaver como principal herramienta de trabajo se debe a que es una aplicación muy potente para desarrollar sitios web respetando los estándares CSS y HTML. Con ella, podemos diseñar sitios de forma visual o directamente sobre el código, y tenemos la posibilidad de trabajar con distintos entornos de programación, como XHTML, CSS, XML, JavaScript, AJAX, PHP y ASP. Dreamweaver nos permite configurar sitios y, de esta manera, definir un ámbito de desarrollo y testeo para los diseños y aplicaciones web que generemos, sin necesidad de usar un programa externo para tal fin. Además, esta nueva versión de la herra-

mienta (CSS) cuenta con una funcionalidad muy destacable: el **asistente de codificación inteligente**. Su función es optimizar la velocidad de escritura de código, dado que el programa reconoce y sugiere **Strings** de código en diferentes lenguajes de programación. Estas sugerencias aparecen tanto en funciones predefinidas del lenguaje como en otras personalizadas que hayamos usado en el mismo desarrollo. Por otro lado, con Adobe Dreamweaver tenemos una ventaja simple pero fundamental sobre otros editores de HTML: podemos insertar tags corriendo sólo un comando del programa, como **[Insertar/Imagen]**.

Veremos más detalles de la aplicación a medida que avancemos en la colección. ◀

Dreamweaver es la aplicación de la suite Adobe Web Premium especialmente desarrollada para la creación de sitios web.



SELECCIÓN DE LAS HERRAMIENTAS DE DISEÑO



Una vez que recopilamos la información brindada por el cliente, y luego de determinar las características conceptuales y técnicas del proyecto, llega el momento de elegir las aplicaciones que nos permitirán hacer realidad el sitio.

Como diseñadores web, debemos tener en cuenta que, para realizar cualquier proyecto, es de suma importancia determinar cuáles serán los recursos de los que nos valdremos para llevarlo a cabo. Éstos involucran las diferentes aplicaciones de diseño que utilizaremos y, también, nuestro nivel de conocimiento para hacer uso de ellas. Lo primero que debemos saber al momento de seleccionar las herramientas de trabajo es el **resultado estético** que queremos lograr con el diseño, ya que no es lo mismo un diseño basado en elementos fotográficos y efectos, que otro con formas simples, colores plenos y estilo más minimalista. Por ejemplo, si consideramos que nuestro diseño sólo tendrá un tratamiento fotográfico en el fondo o en algún otro sector, pero en forma mínima, no necesari-

PARA SELECCIONAR NUESTRAS HERRAMIENTAS DE TRABAJO, DEBEMOS SABER EL RESULTADO ESTÉTICO QUE QUEREMOS LOGRAR.

riamente debemos optar por desarrollar todo el diseño en **Photoshop**, dado que podríamos trabajar con **Illustrator** e incorporar los elementos fotográficos luego de editarlos con la aplicación anterior. Ésta es una de las maneras en que muchos diseñadores están acostumbrados a trabajar y, de hecho, es una técnica muy eficiente.

Por el contrario, si nuestro diseño requiere de fotomontajes, efectos y demás características que sólo podríamos lograr que queden bien con Photoshop, es recomendable que el proyecto esté basado íntegramente en esta aplicación y que, luego, agreguemos aquellos elementos que precisemos diseñar en Illustrator.

En pocas palabras, lo ideal es establecer en qué porcentaje utilizaremos una u otra aplicación según las necesidades que tengamos en cada proyecto en particular, porque así podremos lograr un flujo de trabajo más cómodo y eficiente. Para entender mejor a qué nos referimos con todo esto, analicemos algunos sitios que ejemplifican diseños basados, necesariamente, en Photoshop. En primer lugar, tenemos O'Reilly's Pub

Splendour in the Grass (www.splendourinthegrass.com) es un sitio muy artístico, repleto de pinceladas y fotomontajes hechos con Photoshop.



En el sitio Przeznaczenie (www.przeznaczenie.eu) aparece una composición de elementos fotográficos con uso de texturas y pinceles especiales.

En Roadify (www.roadify.com) veremos un diseño creativo, con formas plenas e ilustraciones que pueden efectuarse de manera vectorial.





PHOTOSHOP

- ▶ BASADO EN MAPA DE BITS
- ▶ MAYOR PESO DE LOS ARCHIVOS
- ▶ TRABAJA CON CAPAS
- ▶ DISPONEMOS DE UNA RESOLUCIÓN DEFINIDA PARA NUESTROS DISEÑOS

ILLUSTRATOR

- ▶ BASADO EN VECTORES
- ▶ MENOR PESO DE LOS ARCHIVOS
- ▶ TRABAJA CON CAPAS
- ▶ LOS DISEÑOS NO DEPENDEN DE LA RESOLUCIÓN AL ESTAR EN VECTORES

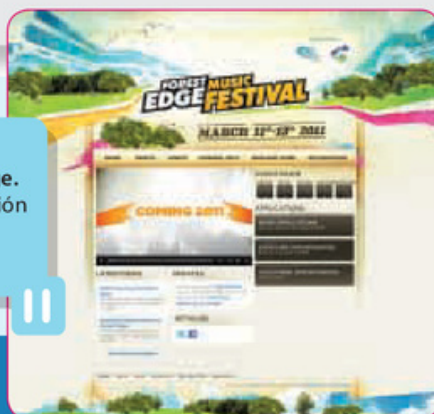
En esta tabla podemos encontrar las diferencias que comúnmente se aprecian entre Photoshop e Illustrator.

(www.oreillypub.com), donde encontramos fondos con texturas, elementos recortados y efectos en los títulos. Por su parte, el sitio de Andrei Arlovsky (www.arlovski.com) presenta imágenes y elementos manipulados con efectos, pinceles y ajustes de color. Finalmente, en Danny Diablo (www.dannydiablo.de) hay texturas, efectos de fuego, luces y volumen, que serían difíciles de lograr sin el mencionado programa. Por otro lado, también podemos encontrar ejemplos que nos permitan entender en qué casos sería mejor basar nuestros diseños en Illustrator. Blue Sky Resumes (www.blueskyresumes.com) es un sitio de formas llenas y colores definidos, sin elementos complejos que requieran intervenciones fotográficas. Por su parte, Just Party (www.justparty.ru) tiene elementos bien marcados, colores plenos y degradés, ideales para realizar con Illustrator. Aunque es importante elegir la aplicación según la experiencia o la habilidad que tengamos para utilizarla, la determinación de la herramienta ideal también debería de considerar si algún otro diseñador se desempeñará junto con nosotros, ya que entonces sería mejor compartir las mismas formas de trabajo. En este sentido, la elección del paquete Adobe Creative Suite 5 no es azarosa, porque las aplicaciones que nos ofrece son utilizadas por una gran mayoría de los diseñadores y desarrolladores de todo el mundo. Esto se debe, en gran medida, a sus características y utilidades realmente completas y profesionales. ◀

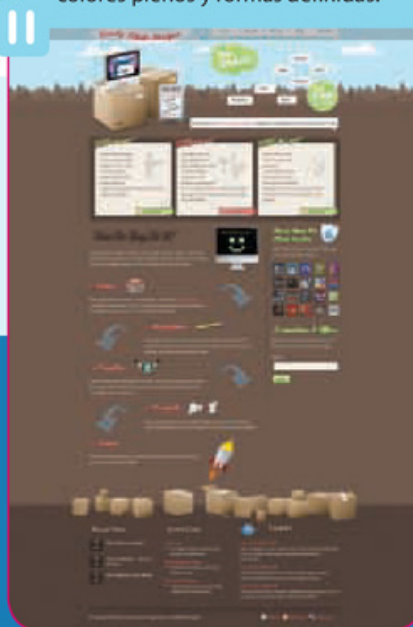


Ready Made Designs (www.readymadedesigns.co.uk) presenta ilustraciones vectoriales, colores plenos y formas definidas.

Claramente, el diseño de Forest Edge Music Festival (www.forestedge.org.au) fue una verdadera composición entre imágenes, pinceles y efectos de Photoshop.



The Gingerbread (www.gingerbreadusa.com) es una mezcla entre elementos vectoriales, fotográficos y texturas, que fácilmente pueden ser basados en Illustrator.



ADOBE PHOTOSHOP CS5

Llegó el momento de introducirnos en el fascinante mundo de Photoshop, el software por excelencia para diseño y retoque fotográfico. Aquí conoceremos su utilidad en el área web.

Aunque a lo largo de los años han aparecido distintos programas con características similares a las de Photoshop, ninguno resulta tan versátil y completo como esta aplicación de Adobe.

Es probable que, a medida que vayamos descubriendo las infinitas posibilidades de este programa, creamos que Photoshop nos ofrece más de lo que necesitamos. Sin embargo, es importante comprender que, utilizando este tipo de aplicaciones tan completas, podremos avanzar en nuestros diseños con un único límite: el que establece nuestra propia imaginación. En cada nueva versión de Photoshop, Adobe pone el acento en hacer más



Diseñadores y fotógrafos de todo el mundo utilizan Photoshop para editar sus imágenes; por ejemplo, para quitar fondos.



fluida la experiencia del usuario y en mejorar el acceso a las utilidades. En este camino de perfeccionamiento, la empresa genera una interfaz cada vez más intuitiva, que incluye nuevas funciones y variados paneles de herramientas. Optar por Photoshop al momento de diseñar nos permite disponer de más flexibilidad en cuanto a efectos, texturas o manipulación de imágenes, y así lograr resultados complejos y de excelente

Podemos encontrar muchísima información sobre Photoshop en su página del sitio de Adobe (www.adobe.com/es).

calidad. De hecho, sería prácticamente imposible alcanzar ciertos objetivos con una aplicación para retoque y manipulación de imágenes que no sea la mencionada.

LAS VENTAJAS DE PHOTOSHOP

A la hora de crear proyectos para la Web, Photoshop nos permite componer los distintos elementos que luego incluiremos en el diseño web.

A medida que nos vayamos interiorizando en su uso, aprenderemos cómo se utilizan los elementos de pintura, cómo se realizan selecciones para poder manejar determinadas áreas, y cómo trabajar

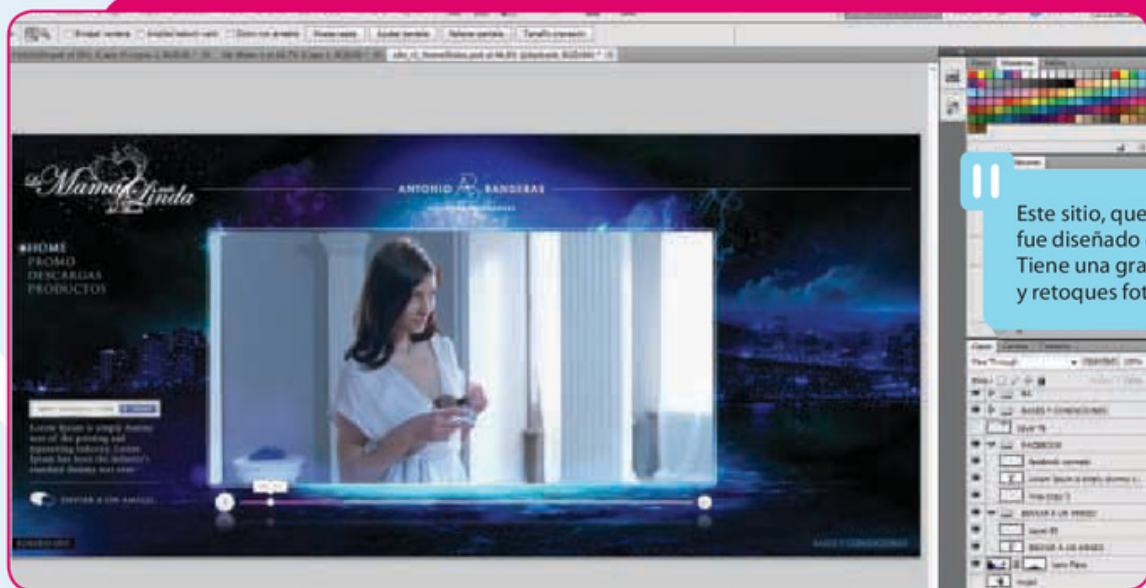
+ COMPLEMENTOS, PINCELES Y FILTROS

Al usar Photoshop, tenemos la ventaja de disponer de cientos de plugins, filtros, pinceles y otros complementos que nos facilitan el trabajo de creación de diseños y de efectos complejos. Existen muchísimos sitios donde podremos hallar este tipo de recursos, y bastará con descargarlos e incorporarlos al programa para poder usarlos de forma sencilla. Entre los sitios recomendados para descargar pinceles, recomendamos visitar My Photoshop Brushes (www.mypsdbrushes.com), PSBrushes (www.psbrushes.net) y Brusheezy (www.brusheezy.com). Para encontrar estilos predefinidos, son aconsejables Free Photoshop (www.freephotoshop.org/styles) y Photoshop Styles (www.photoshopstyles.net). A la hora de descargar plugins, podemos visitar Intercambios Virtuales (www.intercambiosvirtuales.org) y Alien Skin (www.alienskin.com). Si buscamos otros recursos, Blue Vertigo (www.bluevertigo.com.ar) y PSD Tuts (www.psd.tutsplus.com) son buenas fuentes de información.

con capas organizándolas, moviéndolas y transformándolas. También haremos recortes y montajes, y emplearemos los filtros y los efectos que la aplicación pone a nuestro alcance. Dentro de las posibilidades que nos brinda este programa, tanto en la edición de imágenes como en

lo relativo al diseño web, podemos destacar y tener en cuenta algunas características de suma importancia para nuestra tarea. Por ejemplo, la creación de elementos 3D en forma sencilla, los pinceles para la creación de texturas y fondos artísticos, los ajustes de imágenes, la manipulación de colores

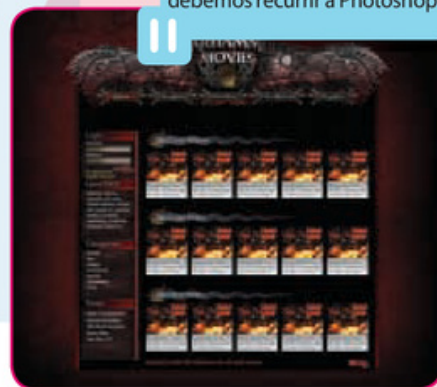
y de textos, y los estilos predeterminados para aplicar rápidamente a los diseños. Finalmente, una de las principales funciones de esta herramienta a la hora de hacer diseños para la Web, es la posibilidad de crear, recortar y optimizar las imágenes de nuestro sitio. ◀



Este sitio, que aún no está en línea, fue diseñado en Photoshop. Tiene una gran calidad de efectos y retoques fotográficos.



Para crear un diseño como el de www.warhammermovies.com, debemos recurrir a Photoshop.



Los pinceles son realmente útiles para crear texturas y fondos.

ADOBE

ILLUSTRATOR CS5

Ésta es la aplicación estrella de Adobe para el diseño vectorial. Es la herramienta favorita de todos los diseñadores, que realizan con ella sus trabajos de diseño gráfico y web.

Adobe Illustrator es la aplicación que más se utiliza en todo el mundo para crear diseños de todo tipo, para soportes tanto impresos como digitales, diseño de marcas, gráfica publicitaria y papelería. En los últimos tiempos, también es la preferida de los diseñadores web al momento de sentarse frente a sus computadoras y desarrollar sus sitios. Con este programa podremos diseñar sitios web fácilmente, gracias al empleo de sus herramientas de dibujo vectorial, a los pinceles y al excelente manejo tipográfico que ofrece. A partir de esta versión, nuestras ilustraciones se alinean con los píxeles de pantalla para lograr que los diseños se vean totalmente nítidos cuando son incorporados a la Web.



RECURSOS PARA NUESTROS DISEÑOS

Al ser Illustrator una aplicación utilizada por muchísimos diseñadores en todo el mundo, también dispone de recursos que podemos encontrar en Internet para facilitar nuestra tarea: desde templates, pinceles y formas vectoriales, hasta tutoriales, complementos y plugins interesantes. A continuación, recomendaremos algunos sitios en donde obtener recursos muy valiosos. Si necesitamos pinceles, podemos visitar Color Burned (www.colorburned.com/illustrator-brushes) o Createsk8 (www.createsk8.com/about). Para encontrar texturas e imágenes, BittBox (www.bittbox.com) es una gran alternativa. Si necesitamos recursos en general, podemos recurrir a Points and Paths (www.pointsandpaths.com), Illustrator World (www.illustratorworld.com) o Vector TutsPlus (www.vector.tutsplus.com). Encontraremos inspiración en sitios como Logopond (www.logopond.com), Dzine Blog (www.dzineblog.com) e Illustrator Tips (www.illustratortips.com); y tipografías, en DaFont (www.dafont.com) y Urban Fonts (www.urbanfonts.com).

Veamos, a continuación, algunas de las características más notables en detalle:

- **Diseños nítidos para la Web:** en esta versión podemos disponer de una cuadrícula en píxeles para nuestro espacio de trabajo. Los elementos se ajustarán a ellos y así podremos apreciar cómo se verán realmente los diseños en la Web.
- **Perspectivas:** podemos crear elementos en perspectiva fácilmente, y dotar a nuestros diseños de profundidad y distancia, gracias a las nuevas herramientas que brinda Illustrator.
- **Siluetas interactivas:** ésta es una nueva herramienta para la creación de siluetas a partir de trazos. Nos da la posibilidad de definir puntos de control interactivos y precisos, con los que podremos modificar el ancho de una silueta a lo largo del trazo.

LA RESOLUCIÓN NO ES PROBLEMA PARA LOS VECTORES

Al momento de decidir qué aplicación utilizar para nuestro trabajo, Illustrator es la ideal cuando lo que queremos obtener son diseños e ilustraciones definidos, donde no tengamos restricciones en lo que respecta a la resolución o al tamaño de los objetos o elementos creados. Illustrator es la opción en estos casos, ya que, al utilizar una herramienta de vectores para crear formas, fondos, iconos o menús, podemos manipular sus dimensiones de manera sencilla y sin perder calidad o definición. Esto se debe a que los elementos vectoriales son creados matemáticamente y no tienen una relación directa con la resolución en píxeles, como sí sucede con las imágenes de mapa de bits. Por esta razón, nuestros archivos pesan mucho menos y son más ágiles para manejar.



Podemos ver el site oficial de Illustrator en www.adobe.com/products/illustrator.

ILLUSTRATOR SE INTEGRA CON APLICACIONES COMO PHOTOSHOP PARA QUE NUESTROS PROYECTOS DE DISEÑO NO TENGAN LÍMITES.

También existe la posibilidad de trabajar con imágenes en píxeles (o mapa de bits), que podemos incorporar en nuestro espacio y, allí, manipularlas, ajustar sus colores, aplicarles efectos o, si lo preferimos, traerlas directamente desde Photoshop luego de retocarlas.

Lo ideal al momento de trabajar en el diseño de un sitio web es combinar las posibilidades que brinda Illustrator con las ofrecidas por Photoshop, de modo que podamos llevar contenidos desde una aplicación a otra. Este proceso es muy sencillo, porque Adobe nos permite pasar elementos de un programa a otro, simplemente, arrastrándolos.

Como recomendación, podemos decir que es conveniente basar el diseño de los sitios en Illustrator cuando en ellos no se precisen grandes efectos y otras características que sean limitadas en esta aplicación. También es aconsejable recurrir a él cuando los diseños van a usarse en sitios realizados en Flash, ya que ambas son aplicaciones vectoriales. Entonces podremos pasar nuestro trabajo directamente, optimizando el peso del sitio por no recurrir a otros tipos de imágenes en elementos que pueden ser vectores.

RECURSOS PARA NUESTROS DISEÑOS

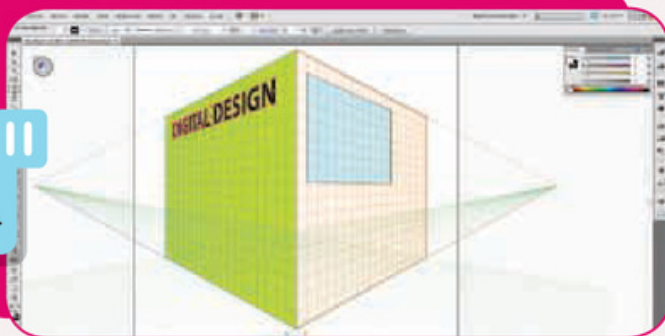
Al ser Illustrator una aplicación utilizada por muchísimos diseñadores en todo el mundo, también dispone de recursos que podemos encontrar en Internet para facilitar nuestra tarea: desde templates, pinceles y formas vectoriales, hasta tutoriales, complementos y plugins interesantes. A continuación, recomendaremos algunos sitios en donde obtener recursos muy atractivos. Si necesitamos pinceles, podemos visitar Color Burned (www.colorburned.com/illustrator-brushes) o Createsk8 (www.createsk8.com/about). Para encontrar texturas e imáge-



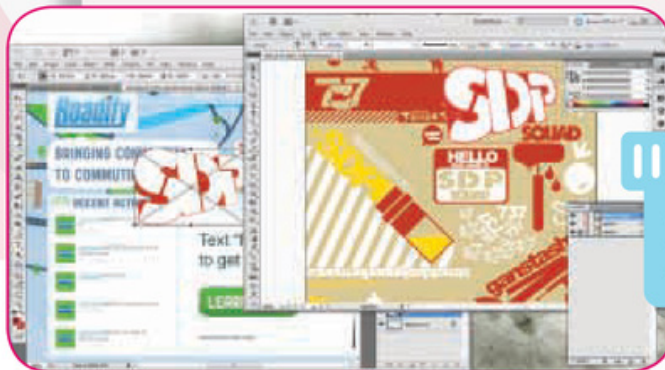
MAPA DE BITS VS. VECTORES	BITMAP	VECTOR
► FORMACIÓN	Imagen construida punto por punto en una grilla. Cada uno de los puntos se denomina píxel.	La imagen se construye a partir de cálculos matemáticos que dibujan la forma creada.
► PESO DEL ARCHIVO	Alto, aumenta con el tamaño de la imagen.	Bajo, no depende del tamaño de la imagen.
► USO IDEAL	Fotografías, imágenes con degradados o escala de grises.	Ilustraciones, imágenes con colores plenos.
► ESCALABILIDAD	Al aumentar el tamaño, comienzan a percibirse los píxeles.	Absoluta, no pierde definición al aumentar el tamaño ya que se redibujan las figuras.
► MODIFICACIÓN	No son editables, es decir, no se pueden recrear o modificar como si se los hubiese creado desde cero. Sí es posible modificarlos si se los guarda y se los vuelve a abrir en un programa como Photoshop.	Se pueden modificar luego de crearlos, es decir, son editables mientras se mantengan en formato vectorial.
► FORMATOS DE ARCHIVO	PSD, BMP, GIF, JPG, TIF	AI, SWF, PNG, EPS, CDR, DWG, SVG, VML
► APLICACIONES UTILIZADAS	Photoshop, Paint, Photopaint, Paint Shop Pro, The GIMP.	Illustrator, Flash, Fireworks, CorelDraw

Cada formato de imagen funciona mejor para un determinado uso.

Herramienta de perspectiva en pleno trabajo.



Pasar elementos de una aplicación a otra es muy sencillo con la suite de Adobe.



nes, BittBox (www.bittbox.com) es una gran alternativa. Si necesitamos recursos en general, podemos recurrir a Points and Paths (www.pointsandpaths.com), Illustrator World (www.illustratorworld.com) o Vector TutsPlus (www.vector.tutsplus.com).

Encontraremos inspiración en sitios como Logopond (www.logopond.com), Dzine Blog (www.dzineblog.com) e Illustrator Tips (www.illustratortips.com); y tipografías, en DaFont (www.dafont.com) y Urban Fonts (www.urbanfonts.com). ◀

ADOBE CS5

DREAMWEAVER

Esta aplicación combina una fuerte herramienta visual con funciones de desarrollo y edición de código de programación. A partir de ahora, comenzaremos a dar nuestros primeros pasos con la versión CS5 de este editor de páginas web de Adobe.

La flexibilidad es una de las cualidades más destacadas de Dreamweaver. Esta aplicación puede ser usada por aquellas personas que tengan poca experiencia en el armado de una página HTML, porque al insertar elementos desde los menús, el programa se encarga de redactar a la perfección el código HTML y CSS. También es útil para usuarios avanzados, ya que permite manipular manualmente los tags con asistencias de edición excepcionales, que aceleran el proceso de redacción de código. Luego de iniciar Dreamweaver por primera vez, es recomendable definir la distribución del **espacio de trabajo**. Esto determinará la ubicación de los distintos menús y opciones dentro de la aplicación, y es una selección que quedará registrada para las próximas veces que lo usemos. Para hacerlo, nos dirigimos al menú **[Ventana/Diseño del espacio de trabajo]**. La distribución llamada **[Clásico]** es sumamente flexible e intuitiva, y resulta familiar para quienes vienen trabajando con versiones anteriores de este programa. Sin embargo, cada usuario elegirá la que le resulte más adecuada.

EL ÁREA DE TRABAJO

Cuando abrimos un documento (o creamos uno nuevo, como veremos más adelante), por defecto aparece la pantalla dividida en dos zonas principales: **[Código]**

y **[Diseño]**. Como su nombre lo indica, en el área **[Código]**, ubicada en la parte superior, veremos el **código HTML** de la página. Aquí también podremos escribir cualquier otro código de programación web que necesitemos (como ASP, PHP, JavaScript, etcétera). Por su parte, en la sección **[Diseño]**, que se encuentra en la parte inferior, se muestra la previsualización (**preview**) del código interpretado. El resto de la pantalla se compone de la siguiente manera: en la parte superior están el menú y la barra de herramientas; en la derecha, la barra de ventanas; y en la parte inferior, la barra de propiedades HTML y CSS. Veremos estas secciones en detalle en las próximas páginas, donde conoceremos las partes de la interfaz del programa.

NUEVAS CARACTERÍSTICAS

Adobe Dreamweaver CS5 incluye mejoras respecto a su antecesor; las más destacadas son las siguientes:

► **Inspección de hojas de estilo:** se pueden cambiar las propiedades CSS sin necesidad de escribir código, simplemente, desde un panel ubicado a la derecha del área de trabajo.

► **Integración con Adobe BrowserLab:** éste es un servicio en línea de Adobe que permite probar y comparar nuestro desarrollo en diferentes navegadores de Internet.

► **Sugerencias de código PHP personalizado:** se trata de un asistente para escribir sintaxis adecuada de nuestras funciones personalizadas de PHP.

Aunque trabajaremos con CSS más adelante, vale destacar en este caso la nueva característica de Dreamweaver CS5 que realiza la **inspección de CSS** en tiempo real. Gracias a esta funcionalidad, es posible cambiar las propiedades de una hoja de estilos sin necesidad de interpretar código. Desde el panel **[Estilos CSS]**, podremos modificar colores y otros parámetros de manera muy sencilla.

La primera vez que ejecutamos Dreamweaver, debemos definir si será el editor predeterminado para los tipos de archivo que soporta.

Editor predeterminado

Adobe Dreamweaver CS5 se establecerá como editor predeterminado para los siguientes tipos de archivo:

- | | |
|--|---|
| ☒ Active Server Pages (.asp) | ☒ JavaScript (.js) |
| ☐ Java Server Pages (.jsp) | ☒ Hojas de estilos en cascada (.css) |
| ☒ ColdFusion (.cfm, .cfc) | ☒ Lenguaje de formato extensible (.xml) |
| ☐ ASP.NET (.aspx, .ascx) | ☒ Hojas de estilo XSL (.xsl, .xslt) |
| ☒ PHP (.php, .php3, .php4) | |

Seleccionar todo

Anular todas las selecciones

Aceptar



Al abrirse, Dreamweaver ofrece una lista de acciones que podemos realizar. Tenemos la opción de evitar que se abra este cuadro marcando la casilla ubicada en el ángulo inferior izquierdo.

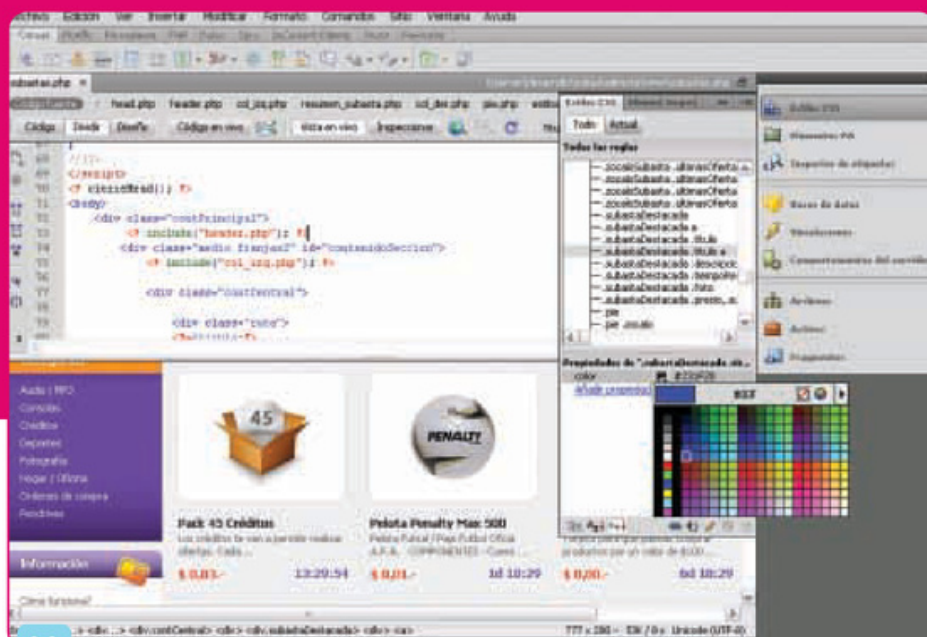
PRESTAR ATENCIÓN AL CÓDIGO

Aunque podemos trabajar exclusivamente en la vista **[Diseño]**, es conveniente utilizar la pantalla dividida de Dreamweaver. De esta forma, a medida que hagamos el diseño de una página, también veremos el código que se va generando. En etapas de aprendizaje, esto nos ayudará a familiarizarnos con la sintaxis HTML y relacionar cada uno de los elementos que insertemos con sus etiquetas y atributos HTML (veremos estos temas sobre el final del capítulo). Más adelante, cuando tengamos más conocimientos, ver el código nos permitirá cuidar que éste cumpla con los estándares definidos por el W3C (www.w3.org). Si bien éste es un tema que veremos en detalle en el próximo capítulo, cabe adelantar que, al trabajar con la interfaz dividida, encontraremos que algunas palabras comienzan con el signo **&** (*ampersand*). Tendremos que familiarizarnos con estos caracteres para poder leer HTML con naturalidad, ya que se utilizan para indicarle al navegador el uso de algunos símbolos o palabras especiales. Hasta aquí, hemos hecho un primer contacto con Adobe Dreamweaver CS5; a medida que avancemos en los capítulos, iremos descubriendo nuevas características y virtudes de nuestro editor de HTML y CSS preferido. ◀



INTEGRACIÓN ENTRE APLICACIONES DE LA SUITE

Además de las cualidades y beneficios de cada uno de los programas que componen la suite Adobe Web Premium, trabajar con el paquete completo ofrece ventajas adicionales. Entre ellas, una muy destacable es la que permite crear un objeto inteligente de imagen a partir de la inserción de un archivo proveniente de Photoshop o de Illustrator. Gracias al objeto inteligente, si hacemos cambios en la imagen original, la de Dreamweaver se actualizará para reflejar esas modificaciones.



Luego de seleccionar un estilo, buscamos la propiedad que queremos modificar y elegimos el nuevo valor que deseamos darle.

LA INTERFAZ DE

DREAMWEAVER CS5

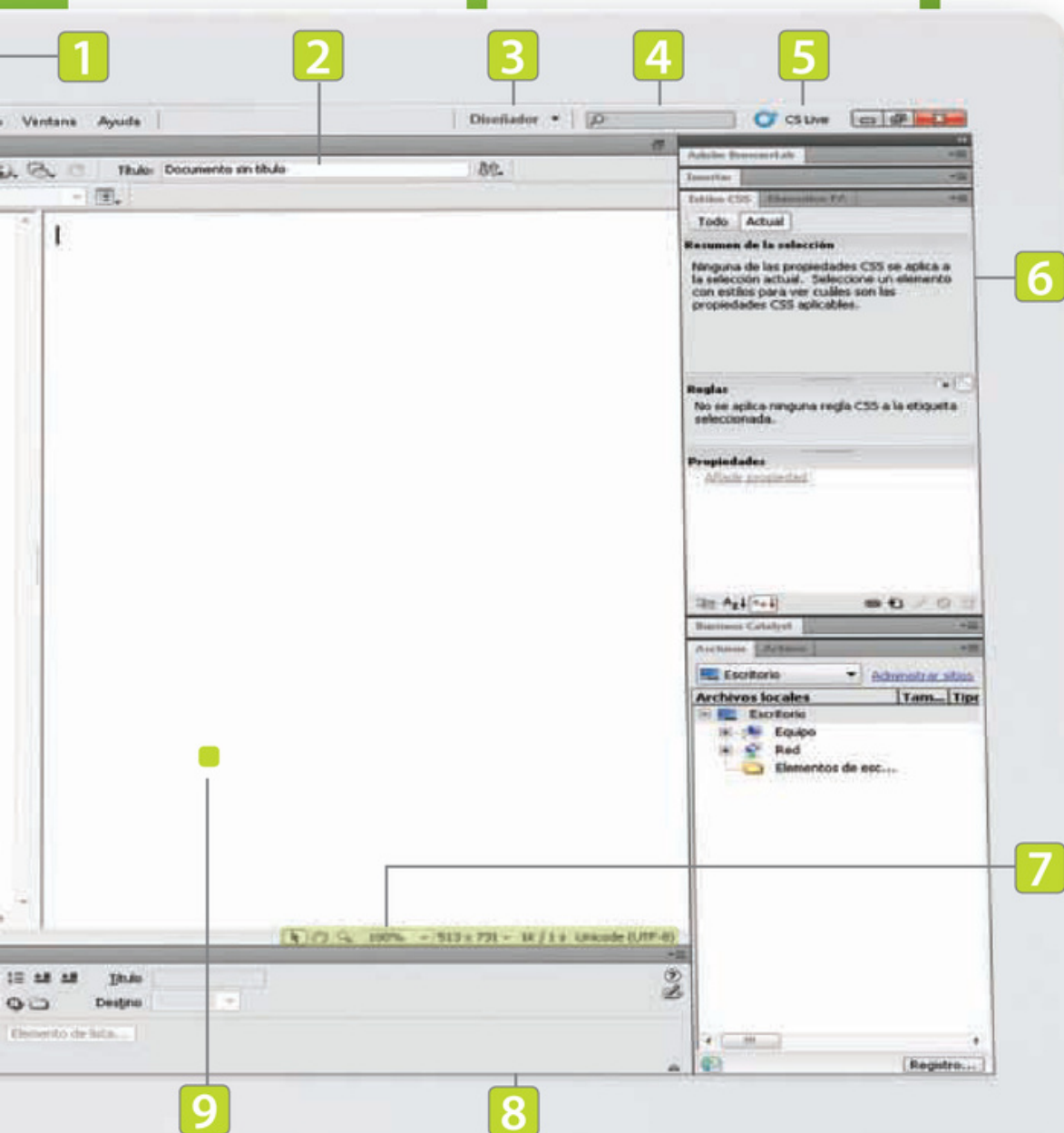
Antes de comenzar a trabajar, haremos un repaso de la ventana principal de Dreamweaver para conocer dónde están ubicadas las principales utilidades que nos brinda esta aplicación.

Adobe Dreamweaver posee una interfaz fácil de utilizar, que nos da la posibilidad de manipular los distintos archivos que componen un sitio de forma muy sencilla y cómoda. Conocerla nos permitirá ahorrar tiempo cuando empecemos a armar nuestras páginas. ◀

- 1** Barra de menú, que contiene todas las funciones del programa.
- 2** Barra del documento, en donde colocamos el título de nuestro sitio.
- 3** Vista de pantalla elegida. Podemos cambiarla de acuerdo con el rol que tenemos al utilizar el programa. Por ejemplo, diseñador, desarrollador o programador, entre otras opciones.
- 4** Cuadro de búsqueda que nos permite localizar texto o sintaxis de programación dentro del documento abierto.
- 5** CS Live. Nos da la oportunidad de conectarnos al sitio de Adobe de forma directa.
- 6** Paneles desde donde podemos agregar, cambiar o eliminar distintas propiedades de los documentos (nos permitirá desde crear un archivo nuevo hasta eliminar una clase en CSS).
- 7** Propiedades generales del documento: desde el tamaño hasta el UTF en el que está generado el archivo.



- 8** Panel [Propiedades], donde podemos ver las propiedades de una etiqueta en particular.
- 9** Vista previa para la posición de los objetos de un sitio. Ésta es una previsualización del sitio, que nos permite ubicar los objetos en el espacio de la página.
- 10** Permite cambiar entre las propiedades HTML y CSS del documento en ejecución.
- 11** Muestra las etiquetas definidas. Si hubiese una etiqueta <p>, la mostraría a continuación de <body>, ya que esta última contiene a la anterior. Por ejemplo, <body><p>.
- 12** Sección de la pantalla destinada a ver el código del sitio.
- 13** Barra de programación. Nos permite desde contraer el código del sitio hasta ordenarlo para obtener una lectura fácil y rápida.
- 14** Barra donde se muestra la URL de nuestro sitio, si tenemos el programa configurado para acceder al servidor.
- 15** Barra que permite elegir cómo ver la pantalla de Dreamweaver. Es muy práctica porque podemos ver sólo el diseño, el código o ambos (la forma más utilizada es la última).
- 16** Nombre del documento (del archivo HTML, PHP, etcétera).



ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 ¿Cuáles son las herramientas fundamentales para crear un sitio web?

02 ¿Cuáles son las herramientas complementarias para crear un sitio web?

03 ¿Cuáles son las ventajas de **TopStyle Pro**?

04 ¿Cuáles son las diferencias entre el software Photoshop e Illustrator?

05 Explique las ventajas de Photoshop.

06 Enumere las diferencias entre las imágenes de mapas de bits y las vectoriales.

07 Mencione las mejoras de la versión Dreamweaver CS5.

08 ¿Cuáles son las características más notables de Illustrator CS5?

09 ¿Cuáles son los puntos principales de **Notepad ++**?

10 ¿Por qué elegir Dreamweaver como elemento principal de trabajo?



PRÁCTICAS

01 Ingrese a **www.adobe.com/es** y conozca más sobre las aplicaciones principales para diseñar un sitio web.

02 Entre a los sitios recomendados para buscar pinceles y descargue algunos que pueda llegar a utilizar en su página.

03 Compare un archivo de mapa de bits con otro vectorial.

04 La primera vez que abra Dreamweaver seleccione qué archivos soportará ese software.

05 Abra Dreamweaver y cree un nuevo archivo **HTML**.



XHTML y CSS: básico

- ▶ ¿QUÉ ES HTML?
- ▶ ESTRUCTURA BÁSICA DE UNA PÁGINA: HEAD Y BODY
- ▶ ETIQUETAS Y ATRIBUTOS: DEFINICIÓN
- ▶ DREAMWEAVER, PRIMER HTML
- ▶ ELEMENTOS HTML
- ▶ ATRIBUTOS
- ▶ NUESTRAS PRIMERAS ETIQUETAS HTML
- ▶ CSS: CASCADING STYLE SHEETS
- ▶ NUEVAS CARACTERÍSTICAS DE CSS3
- ▶ SELECTORES
- ▶ TABLA DE ESTILOS
- ▶ CSS ACCESIBLE



¿QUÉ ES HTML?

Ahora que ya conocemos la interfaz de Adobe Dreamweaver, veremos los aspectos básicos del lenguaje HTML, así como también las principales características de XHTML. Con esta información, comprenderemos mejor la creación de páginas.

Toda página web está conformada por un código estructurado, denominado **HTML** (*HyperText Markup Language*). Básicamente, éste cuenta con una serie de instrucciones para indicarle al navegador que estemos utilizando la manera en que la página debe ser visualizada y representada. Una de las características de este lenguaje es su **simplicidad**, dado que para crear un documento HTML, lo único que necesitamos es un editor de texto, como WordPad o el Bloc de notas de Windows. Con estas dos sencillas herramientas, es posible estructurar una página web por medio de la inclusión de las **etiquetas** (**tags**) propias del lenguaje. Éstas indican el comienzo y el final de los elementos que integran el documento, desde las características de los textos hasta los elementos multimedia, como

HTML ES UN LENGUAJE DE MARCADO CON ETIQUETAS QUE LE INDICAN AL NAVEGADOR LO QUE DEBE MOSTRAR.

imágenes, videos y sonidos. Aunque HTML es un lenguaje, no lo es de programación, sino que es un **lenguaje de marcado**. Se trata de un sistema de etiquetas que indica al navegador, por ejemplo, cuándo subrayar un título, cómo definir el color del texto o dónde insertar una imagen. Si el navegador detectara algún error en el código que debe representar, mostraría el documento tal como lo hubiera interpretado.

XHTML

La sigla proviene de *EXtensible HyperText Markup Language*, y es un lenguaje que cuenta con un etiquetado más estricto que HTML, con lo cual permite una correcta interpretación del contenido del documento. Posee una mayor flexibilidad, ya que logra ser representado por todos los navegadores, incluyendo los de dispositivos portátiles, como PDAs y teléfonos celulares, entre otros. Gracias a XHTML, el mismo documento es útil para todos los soportes; es decir, no es necesario desarrollar un documento para navegadores web y otro diferente para contenido móvil, por ejemplo. Además, los elementos correspondientes al diseño –como colores, tipos de fuente o fondos– se representan en **hojas de estilo en cascada** (**CSS**) por separado, lo que permite tener un código mucho más limpio y de menor tamaño.

DIFERENCIAS ENTRE HTML Y XHTML

Hasta aquí hemos analizado cada lenguaje de marcado por separado. En esta sección, los detallaremos con más profundidad y analizaremos las principales características de cada uno, para comprender mejor sus diferencias.

El código XHTML tiene una menor cantidad de errores de compatibilidad y una mayor integración con las diferentes plataformas de visualización de contenidos web.



La evolución de HTML ha sido compleja, y avanzó sobre diferentes temáticas, que cambiaron aspectos del lenguaje año tras año hasta llegar a la versión actual (HTML 4.0). Ésta incluye elementos que en el pasado no estaban integrados y que fueron surgiendo gracias a los distintos fabricantes de navegadores web. Sin embargo, hoy en día no se navega por la Web únicamente desde una computadora, sino que cada vez se utilizan más los dispositivos móviles y de otro tipo. Debido a esto, se hizo necesario crear un mecanismo que permitiera manejar la información de manera más estandarizada y para que fuera universalmente accesible.

XHTML es el nuevo lenguaje estándar generalizado para la Web desde el año 2000. Como está basado en **XML** (*Extensible Markup Language*), es posible considerarlo como una combinación entre HTML y XML.

BENEFICIOS DE XHTML

Los principales beneficios de XHTML frente a otros lenguajes son los siguientes:



También podemos navegar por sitios HTML desde nuestros teléfonos móviles.

► **Compatibilidad con navegadores antiguos:** XHTML fue diseñado con el objetivo de cumplir con los requerimientos de todos los browsers.

► **Reducción de errores:** como mencionamos anteriormente, al ser más estricto, elimina una gran cantidad de errores propios de HTML.

► **Facilidad de edición:** por ser un código más estructurado –que, además, separa los elementos del diseño en hojas de estilo–, es más legible para editar.

► **Compatibilidad con estándares web:** es compatible con los nuevos estándares sugeridos por el W3C (*World Wide Web Consortium*). ◀

Contenidos
+
Presentación

XHTML
(Contenidos)

+

CSS
(Presentación)

HTML incluye los contenidos y la forma de presentación. XHTML sólo trata con contenidos, que se representan según lo indicado en CSS.



CÓDIGO VÁLIDO

Muchas herramientas de diseño web aún no producen código XHTML completamente válido, es decir, que cumpla con los estándares del W3C. Es por eso que, antes de publicar una página web en Internet, debemos realizar su comprobación en el sitio oficial del W3C (<http://validator.w3.org>), que nos indicará si existen errores y nos sugerirá la manera de corregirlos.

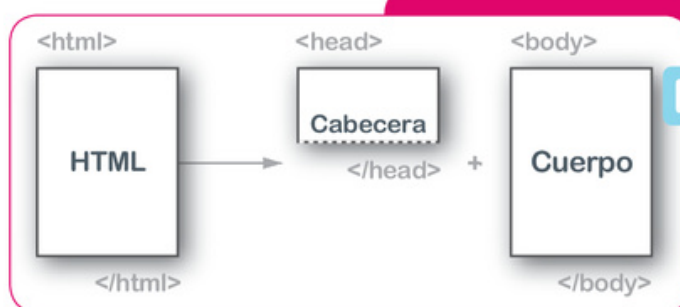
ESTRUCTURA BÁSICA DE UNA PÁGINA: HEAD Y BODY

En esta sección conoceremos dos partes fundamentales de la estructura del XHTML, elementos esenciales en el proceso de armado de una página web. Veamos los sectores básicos de cualquier página.

Por su estructura, las páginas web pueden dividirse en dos partes: **head** (cabecera) y **body** (cuerpo). El head es el encargado de determinar el tipo de documento y sus propiedades, además de otras características no visibles para el usuario que visita la página. Por su parte, el body es el responsable de darle a la página los elementos visibles, es decir, el texto, las imágenes, las animaciones Flash y todo lo que el usuario puede observar en pantalla.

EL HEAD

Esta sección contiene directivas que el navegador necesita para leer el documento correctamente. Entre estos elementos, encontramos el título de la página, las **meta etiquetas**, las **funciones** JavaScript y los **enlaces** de referencia.



El documento HTML le comunica al navegador cómo debe mostrar la información que contiene.

Podemos mencionar siete etiquetas principales en la cabecera, presentes en casi todas las páginas web: **DOCTYPE**, **html**, **title**, **meta DESCRIPTION**, **meta KEYWORDS**, **link** a hojas de estilo en cascada (CSS) y **link externo** a funciones JavaScript. Un encabezado básico estaría conformado como el **Código 1** indica. En este ejemplo, observamos la definición

del tipo de documento en la etiqueta **DOCTYPE**. Aunque no es obligatoria y puede ser omitida, su falta puede dar lugar a imperfecciones en la interpretación por parte del navegador. A continuación, la página está contenida entre las etiquetas **<html>** y **</html>** como apertura y cierre de su código fuente. Luego, aparece la apertura del encabezado a través de la etiqueta **<head>** y el título que será visualizado en la barra de título del navegador, definido entre las etiquetas **<title>** y **</title>**. Le siguen la descripción del sitio y las palabras clave que determinan su temática, y después, los enlaces a las referencias: por un lado, a las hojas de estilo (CSS), y por el otro, a una función JavaScript.

CÓDIGO 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>Título de la Página Web</title>
<meta name="DESCRIPTION" content="Descripción mínima de la
Página Web que debe contener preferentemente alrededor de 70
caracteres.">
<meta name="KEYWORDS" content="palabras, claves, que,
identifican, a, la, web, separadas, por, coma">
<link href="estilos.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="script.js"></script> </
```

EL BODY

El elemento body, cuya apertura es **<body>** y su cierre **</body>**, se introduce luego del cierre de la etiqueta **</head>**. Le indica al navegador los elementos que debe incluir, así como también la manera de hacerlo, es decir, su organización en el espacio de la página web. Luego del tag de cierre del contenido, aparece la etiqueta **</html>**, que corresponde al final de la página. ◀

ETIQUETAS Y ATRIBUTOS: DEFINICIÓN



Es momento de abordar más profundamente el lenguaje HTML. Entraremos en detalle a través de las etiquetas y atributos, dos peldaños que nos permitirán mostrar nuestro sitio de la forma en que deseamos.

El HTML está integrado por elementos que, en conjunto, forman la base de su estructura. De esta manera, cuando definimos una etiqueta, estamos indicando que, en función de ella, se represente algo, que puede ser visible al usuario o no.

DEFINIR LAS ETIQUETAS

Para definir una etiqueta, debemos realizar una apertura y un cierre. Entre ambos, tendremos el contenido que se adaptará a las condiciones que ella disponga. Por ejemplo, para subrayar un texto, utilizaríamos la siguiente línea de código:

```
<u>Texto Subrayado</u>
```

Existen 93 etiquetas estándar que tienen una representación específica, lo que nos facilita la realización de un documento si las utilizamos como herramientas para introducir la información por mostrar.

LOS ATRIBUTOS

Los atributos son directivas que nos permiten definir valores para cada una de las etiquetas agregadas en el código fuente. Son introducidos en las etiquetas con el objeto de otorgar algunas características puntuales, como el alto, el ancho, un borde, etcétera. Un atributo puede definir desde valores numéricos, como la altura de un determinado bloque, hasta colores y estilos de tipografías. Todo atributo

contiene un **valor**, que debe estar encerrado entre comillas. Veamos el ejemplo del **Código 2**. Siempre debemos recordar que las etiquetas deben llevar una apertura y un cierre, porque así lograremos informar cuál es el contenido que debe ser representado

a través de las declaraciones que hacemos en ellas. Cuando declaramos por completo la etiqueta, podemos observar cuatro intervinientes: la etiqueta propiamente dicha, los atributos, los valores de los atributos y el contenido en cuestión, como en el **Código 3**.



CÓDIGO 2

```
<a href="enlace" rel="de qué se tratará el enlace">Aquí el enlace</a>
```

CÓDIGO 3

```
<etiqueta atributo1="numérico" atributo2="alfanumérico">
Contenido</etiqueta>
```



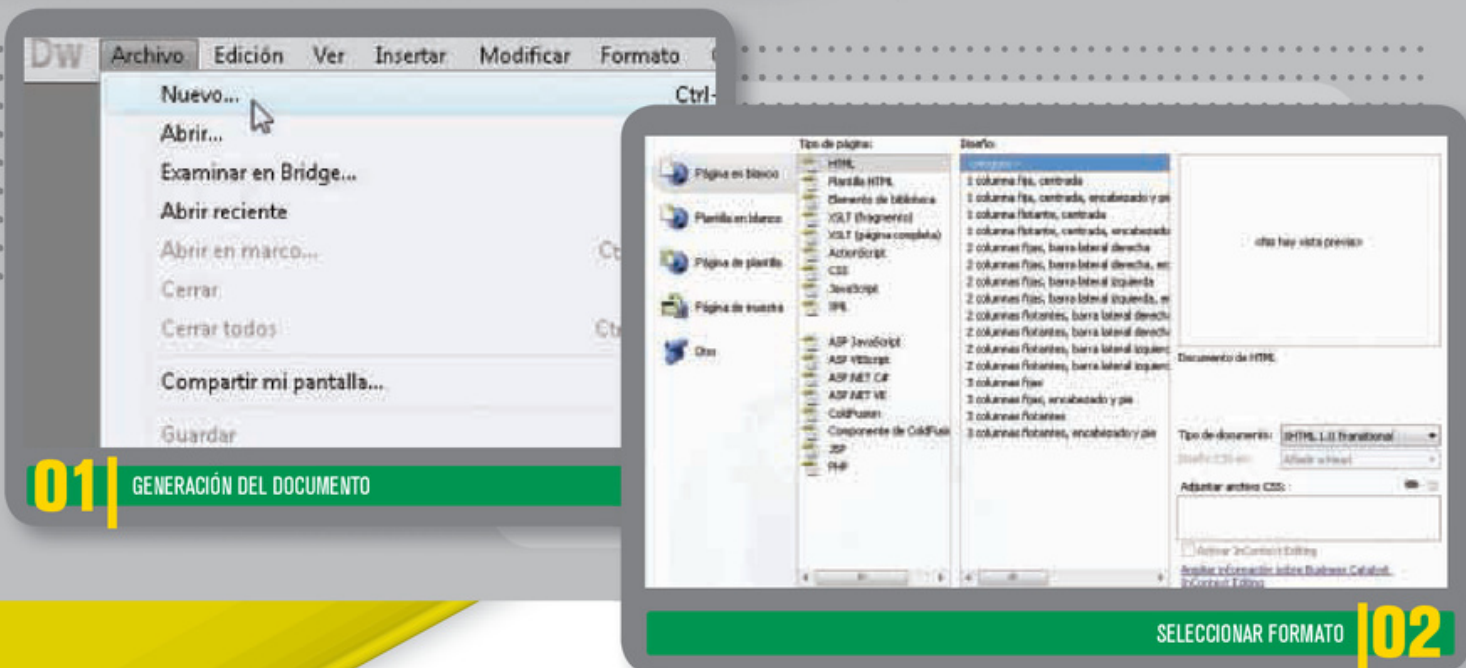
ETIQUETAS OBSOLETAS

Aunque podemos utilizar cualquiera de las etiquetas disponibles en HTML, en este caso recomendamos no emplear algunas en particular, ya que se consideran obsoletas. Entre ellas, podemos mencionar **applet**, **basefont**, **center**, **dir**, **font**, **isindex**, **menu**, **s**, **strike**, **u**.

DREAMWEAVER,

PASO A PASO

A continuación, realizaremos un nuevo documento en HTML utilizando Dreamweaver como editor.



01 | GENERACIÓN DEL DOCUMENTO

SELECCIONAR FORMATO | 02

Para continuar con nuestro aprendizaje e introducirnos en la práctica de HTML, Adobe Dreamweaver brinda la posibilidad de crear documentos en una gran variedad de formatos, al mismo tiempo que permite abrir todo tipo de archivos basados en texto, tales como ASP, PHP, JavaScript y hojas de estilo en cascada (CSS), aunque no hayan sido creados utilizando este programa. Además, podemos abrir y editar archivos de código fuente Visual Basic, Java y C#.

PASO | 01 Para crear un documento HTML, podemos hacer clic en la opción **[HTML]** que aparece en la sección **[Crear nuevo]**, del cuadro de bienvenida del programa. Otra alternativa, si ya tenemos el programa abierto, es dirigimos al menú **[Archivo/Nuevo...]**.

PASO | 02 Desde el cuadro que se presenta, podemos escoger entre una gran cantidad de formatos y, al crear un documento nuevo, optar por alguno de los diseños de página predefinidos o tomar un documento basado en una plantilla existente. En este caso, dentro de la pestaña **[Página en blanco]**, elegimos **[HTML]** en el recuadro **[Tipo de página]**. En la sección **[Diseño]** seleccionamos **[Ninguno]** y, luego, presionamos el botón **[Crear]**.



PASO | 04

Al comenzar a preparar el nuevo documento, revisamos el código de la izquierda hasta identificar la etiqueta `<title>`. Para colocar el título del sitio, escribimos, entre las etiquetas `<title>` y `</title>`, el texto que figurará en la ventana del navegador. En este caso, escribimos "Users Web Design".

PASO | 06 Por último, almacenamos el documento en nuestro disco a través del menú **[Archivo/Guardar como...]**. En el cuadro que aparece, colocamos el nombre que le daremos y presionamos el botón **[Guardar]**.

Hemos establecido los aspectos iniciales de la configuración del documento HTML. A medida que avancemos con nuestro aprendizaje, seremos capaces de incorporar textos, imágenes y otros elementos a nuestras páginas web.

ELEMENTOS HTML

En estas páginas, realizaremos una síntesis de los principales elementos de HTML y su aplicación en los sitios web. Más adelante, cuando comencemos a trabajar, los utilizaremos y sabremos más detalles sobre ellos.

En lo que respecta al lenguaje HTML, se considera a los elementos como las indicaciones o partes que el navegador interpretará más adelante. Durante la programación de un sitio web, encontraremos varios tipos de elementos que nos permitirán modelarlo a nuestro gusto.

UN REPASO POR LOS PRINCIPALES ELEMENTOS

Si bien hay una gran cantidad de elementos, a continuación citaremos algunos de los más utilizados:

- ▶ **<p>**: permite dividir un texto en párrafos. Si preferimos realizar un salto de línea en vez de comenzar un nuevo párrafo, recurrimos al elemento **
**.
- ▶ ****: corresponde a una **lista no numerada**. En ella, cada elemento debe estar identificado con la etiqueta ****.
- ▶ ****: en este caso, corresponde a una **lista numerada** y también se acompaña a cada elemento de la lista con la etiqueta ****.
- ▶ **<h1>**, **<h2>**, ..., **<h6>**: identifican a los **encabezados** del sitio web.

+ RECOMENDACIONES

Es muy interesante observar cómo diferentes maneras de introducir el código pueden hacer variar su tamaño en cuanto a líneas, sin modificar la estética del sitio web. Siempre es aconsejable que nuestro sitio web tenga su código validado por el W3C y que éste sea liviano, para acelerar la interpretación de los navegadores. En este sentido, más adelante veremos que, para determinar cuestiones de forma y estética, es conveniente utilizar CSS en vez de incluir todas las indicaciones dentro del documento HTML.

Su tamaño e importancia varía;
1 es el mayor y 6 es el menor.

- **<a>**: esta etiqueta indica la referencia a algún recurso, y es acompañada por el atributo **"href"**, que sirve para definir la dirección del vínculo.

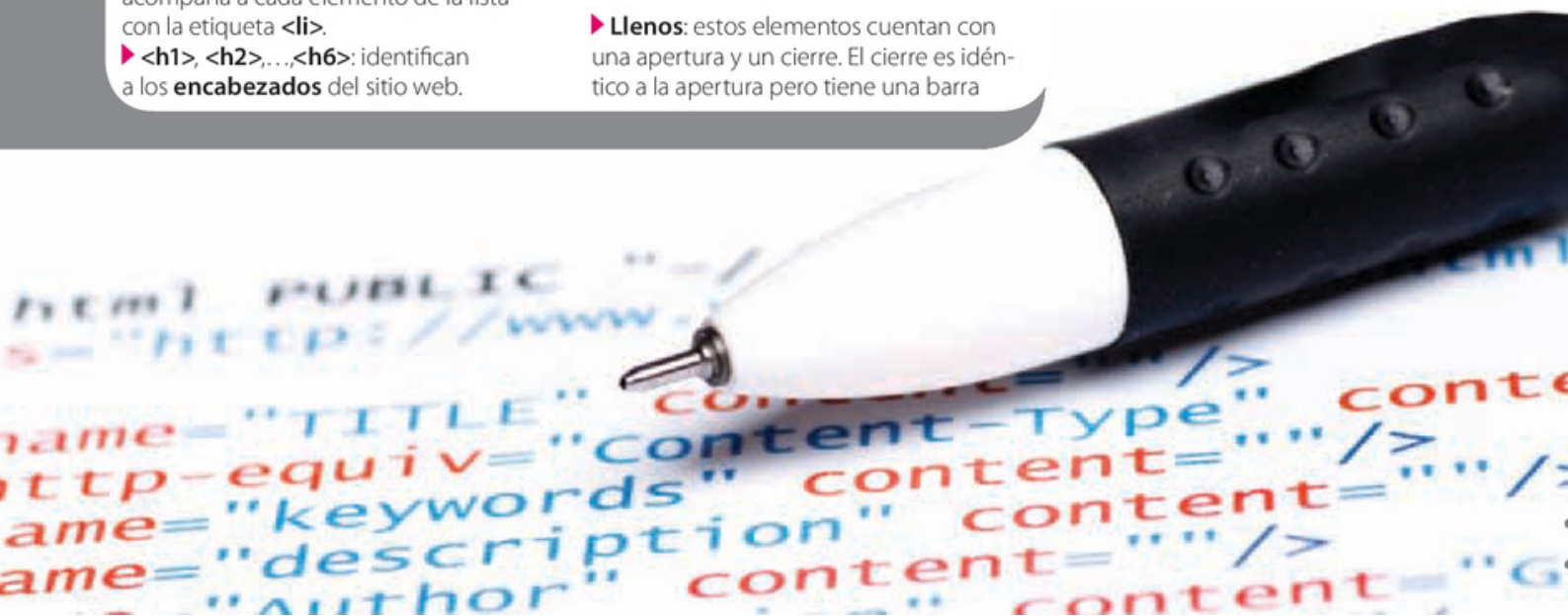
CLASIFICACIÓN DE ELEMENTOS: LLENOS, VACÍOS Y CON ARGUMENTOS

Podemos distinguir tres tipos de elementos en el lenguaje HTML:

- **Llenos:** estos elementos cuentan con una apertura y un cierre. El cierre es idéntico a la apertura pero tiene una barra

diagonal (/). Por ejemplo, si queremos agregar una línea de texto en donde una frase aparezca resaltada en negrita, ingresamos lo siguiente:

```
<p>Aquí una línea de texto  
<b>con una parte resaltada  
en negrita</b></p>
```



► **Vacíos:** en este caso, los elementos no requieren de un cierre. Un ejemplo de este tipo es la etiqueta `<br>`, que da la posibilidad de realizar un salto de línea.

► **Con argumentos:** aquí se encuentran los elementos que poseen atributos definidos; por ejemplo:

```
<table width="700px">Una tabla
de 700 píxeles de ancho</table>
```

CLASIFICACIÓN DE ELEMENTOS: EN LÍNEA Y DE BLOQUE

El lenguaje HTML clasifica los elementos en dos grandes grupos: en línea (**inline**) y de bloque (**block**). La diferencia entre ambos radica en la posición que abarcan sobre el **espacio** disponible en la página. Los elementos de **bloque** siempre comienzan en una **línea nueva**, de modo que ocupan todo el espacio disponible de ella, hasta el final. Por su parte, los elementos en línea ocupan únicamente el espacio necesario para su visualización. Otra diferencia entre ambos grupos es su contenido: mientras que los de bloque admiten en su interior otros elementos de bloque y en línea, los elementos en línea no admiten a los de bloque en su contenido. Esto significa que un elemento de bloque sólo puede aparecer contenido en otro elemento de bloque, pero uno en línea puede aparecer en otros en línea así como también en los de bloque. ◀

`<html>`

`<body>`

`<h1>`

`<p>`

`<img/>`

`<a>`

`<div>`

`<h1>`

`<p>`

`<em>`

`<h2>`

Los elementos permiten ver fácilmente las partes del código y mejoran la posterior visualización en los navegadores.



ELEMENTOS HTML

Conozcamos todas las posibilidades que tenemos con HTML, de forma simple, a través de la tabla presentada a continuación.

TIPO DE ELEMENTO

► Elementos en línea

► Elementos de bloque

► Elementos que pueden ser considerados en línea o de bloque según el caso

ELEMENTO

► a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

► address, blockquote, center, dd, dt, dir, div, dl, fieldset, form, frame-set, h1, h2, h3, h4, h5, h6, hr, isindex, li, menu, noframes, noscript, ol, p, pre, table, tbody, td, tfoot, th, thead, tr, ul.

► button, del, iframe, ins, map, object, script

ATRIBUTOS

En este apartado abordaremos de forma detallada los atributos, descubriremos qué nos permiten realizar y cómo podemos incluirlos en las páginas de nuestro sitio web.

Como ya hemos visto, cada documento HTML es estructurado por los elementos, que le indican al navegador de qué manera debería verse el sitio web que está abriendo. Cuando se precisa más información en los elementos, entran en juego los denominados **atributos**, que añaden datos extra para ser interpretados por los navegadores.

LA **COMBINACIÓN DE ETIQUETAS Y ATRIBUTOS NOS PERMITE LOGRAR DISTINTAS VISUALIZACIONES DE UN SITIO.**

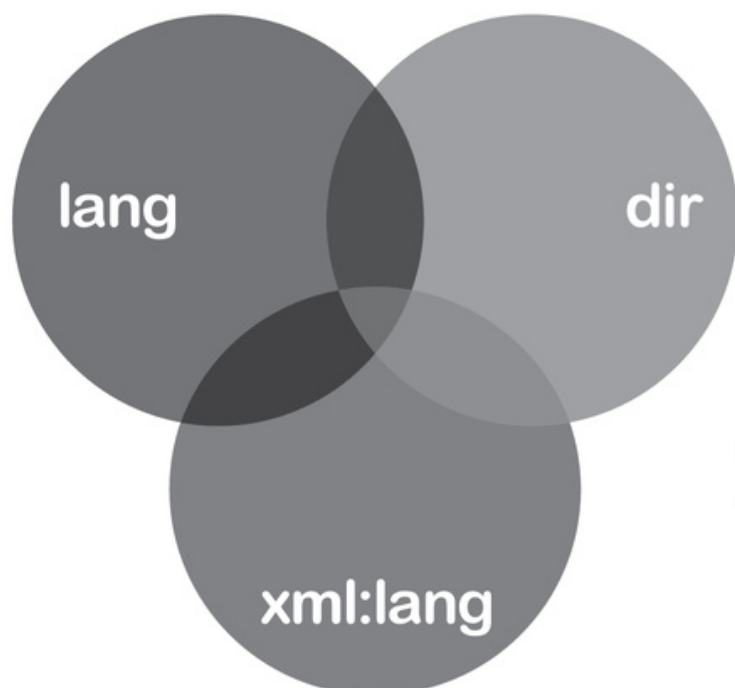
REPASO

Los atributos se incluyen siempre en las etiquetas de apertura de los elementos, y la forma de introducirlos es la siguiente:

```
</img>
```

Como podemos observar en el ejemplo anterior, lo primero que aparece en el atributo es su nombre, seguido por el carácter igual (=) y, por último, el valor del atributo incluido entre comillas. En este caso, hemos definido una imagen con la etiqueta **[img]** y, en su propiedad **[src]**, especificamos el nombre del archivo de imagen. La combinación entre las distintas etiquetas y los diferentes atributos nos abre un gran abanico de posibilidades en cuanto a las opciones de visualización de un sitio. Así, podemos modificar características visuales recurriendo exclusivamente al código fuente, sin que sea necesaria la inclusión de pesadas imágenes que harían retrasar la carga de la página en el navegador.

i18n



A este grupo de atributos también se lo denomina **i18n**, que es la abreviatura de la palabra **internacionalización**, utilizando el número de letras que tiene el término entre la **i** inicial y la **n** final.

LOS CUATRO GRUPOS DE ATRIBUTOS

Si bien cada etiqueta HTML cuenta con sus posibles atributos ya definidos, podemos distinguir algunos que son comunes en casi todas ellas. Es por esto que HTML divide los atributos en cuatro grupos diferenciados por sus funcionalidades: básicos, para internacionalización, de eventos y de foco. A partir de ahora, conoceremos en detalle los correspondientes a cada grupo.

LOS ATRIBUTOS MÁS FRECUENTES: BÁSICOS

Estos atributos son los que se presentan en el código fuente de los documentos HTML con mayor frecuencia, y la razón es bien clara: se utilizan en la mayoría de las etiquetas existentes. En el grupo de los atributos básicos podemos encontrar los siguientes:

- **Id:** identifica de manera única a cada elemento dentro de un documento HTML.
- **Class:** determina la clase CSS que afectará al elemento.
- **Style:** determina de manera directa las propiedades CSS que afectarán al elemento.



CONTENIDO DE LOS TAGS

En general, un tag puede contener otro tag e, incluso, partes de documentos HTML. Sin embargo, algunos tags no pueden contener a otros o sólo pueden contener determinados tags. Aunque hay excepciones, los elementos de bloque pueden contener elementos de línea y elementos de bloque, aunque no sucede lo mismo con los elementos de línea, que no pueden contener elementos de bloque.

- **Title:** añade un título al elemento. Esto es de utilidad para mejorar la accesibilidad del documento, ya que su nombre es visualizado al situar el cursor sobre el elemento. Cabe aclarar que los atributos **[id]** y **[class]** pueden contener letras, números y guiones medios y bajos como valor, pero no pueden empezar con números. Como recomendación, no se deberían utilizar letras como la ñ, ya que podrían no funcionar en algunos navegadores.

LOS ATRIBUTOS DE IDIOMA: PARA INTERNACIONALIZACIÓN

Son utilizados para definir y adaptar los elementos insertados en el documento a un idioma específico.

Se los emplea en páginas que muestran su contenido en varios idiomas o en aquellas que quieren especificar en qué idioma está su documento.

En este grupo encontramos los siguientes tres atributos:

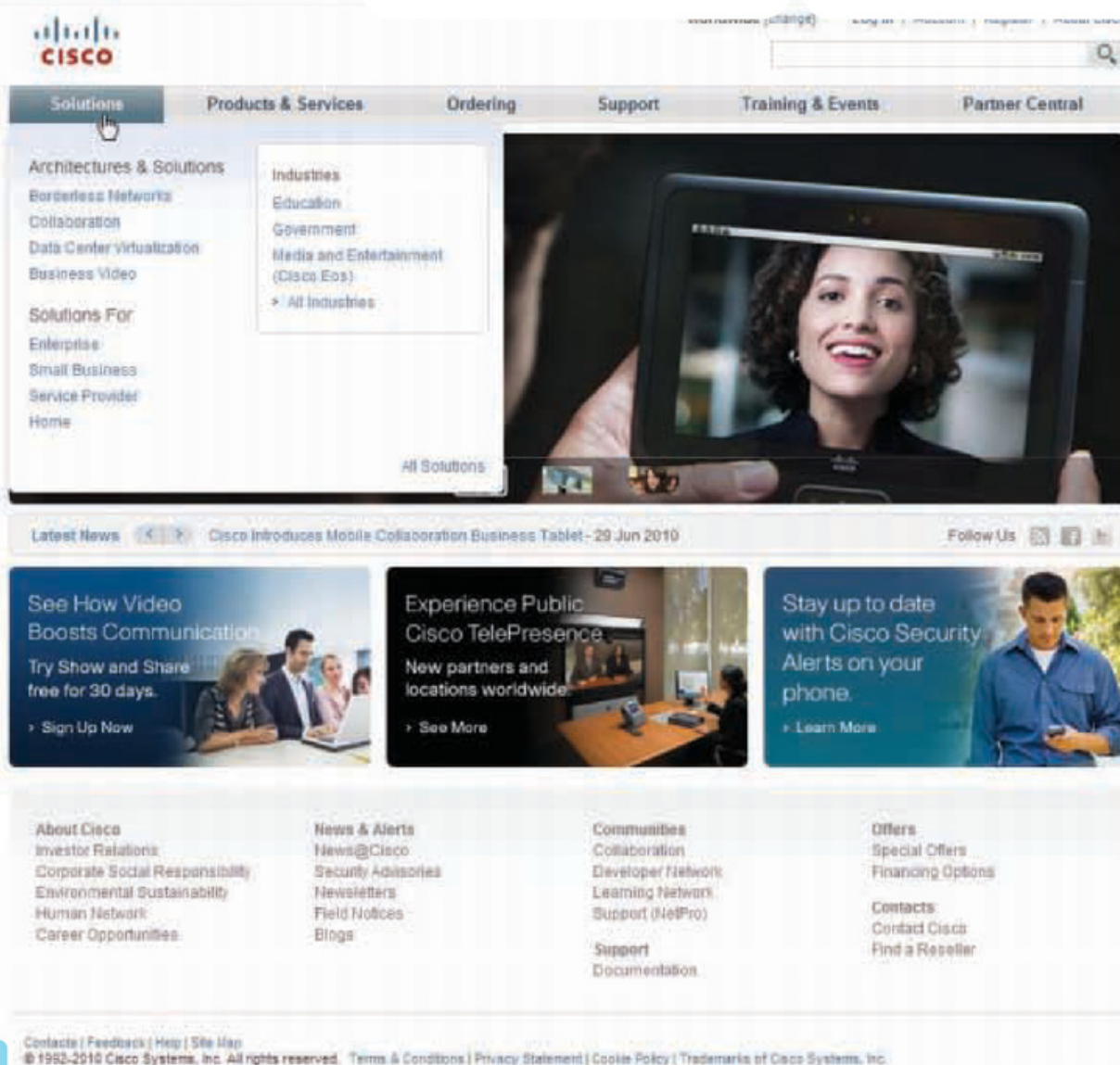
- **Lang:** indica el idioma del elemento.
- **Xml:Lang:** indica el idioma del elemento, aunque en este caso tiene más prioridad que el atributo anterior; es decir, prevalece este atributo por sobre Lang.
- **Dir:** indica la dirección del texto. Se lo utiliza al trabajar con idiomas que no se escriben de izquierda a derecha. ►



LOS ATRIBUTOS DINÁMICOS: DE EVENTOS

Encontramos dentro de este grupo muchos atributos, debido a que son los encargados de exhibir los **efectos** y las **animaciones** de las páginas que incluyen lenguaje JavaScript. Veamos, a continuación, cuáles son estos atributos:

- ▶ **OnClick:** establece la acción que se va a realizar cuando se haga un clic sobre el elemento.
- ▶ **OnDbclick:** establece la acción que se realiza cuando se hace un doble clic sobre el elemento.
- ▶ **Onmousedown:** se ejecuta la acción cuando se detecta el botón del mouse presionado.
- ▶ **Onmouseup:** se ejecuta la acción cuando se detecta que el botón fue soltado.
- ▶ **Onmouseover:** establece la acción cuando se detecta que el cursor se sitúa sobre el elemento.
- ▶ **Onmousemove:** establece la acción cuando se detecta que el cursor se encuentra en movimiento sobre el elemento.
- ▶ **Onmouseout:** se ejecuta la acción cuando el cursor abandona el elemento.
- ▶ **Onkeypress:** se ejecuta la acción cuando se presiona una tecla del teclado.
- ▶ **Onkeydown:** se ejecuta la acción cuando se detecta que está pulsada la tecla del teclado.
- ▶ **Onkeyup:** se ejecuta la acción cuando la tecla del teclado fue soltada.



Los atributos de eventos son muy importantes para las aplicaciones web actuales. En www.cisco.com podemos verlos en acción.

El grupo de atributos de eventos sólo es utilizado cuando el documento HTML incorpora el lenguaje **JavaScript**, ya que a través de ellos, es posible lograr que el sitio web responda ante la interacción del usuario. Por ejemplo, podemos verlos en acción y descubrir su potencial al visitar sitios en los que se emplean menús de navegación JavaScript, los cuales despliegan su lista de elementos cuando nos situamos sobre cada uno de los botones.

LOS ATRIBUTOS DE SELECCIÓN: DE FOCO

El foco aparece durante la interacción del visitante en el sitio. Estos atributos nos permiten asignar acciones que se realizarán en cada caso. Cuando nos referimos a detectar o perder el foco, estamos hablando de que el usuario ha seleccionado o ha deseleccionado el elemento. Por ejemplo, si tenemos un formulario web, diríamos que cuando el usuario está escribiendo en uno de sus campos, este elemento tiene el foco. Cuando el usuario continúa hacia otro campo, el elemento anterior ha perdido el foco, ya que lo tiene el elemento actual que está utilizando. Veamos, a continuación, cuáles son los atributos de foco:

- ▶ **AccessKey:** establece una tecla de acceso rápido a un determinado elemento.
- ▶ **TabIndex:** establece el orden de la tabulación del documento, y su valor está comprendido entre 0 y 32.767.
- ▶ **OnFocus:** se ejecuta la acción cuando se detecta el foco sobre el elemento, lo que significa que actúa cuando el elemento es seleccionado.
- ▶ **OnBlur:** se ejecuta la acción cuando se detecta que el elemento ha perdido el foco; es decir, actúa cuando el elemento se ha **deseleccionado**.

El atributo **accessKey** asigna teclas de acceso rápido a los elementos de un sitio web, lo cual tiene un problema implícito: los navegadores. Este inconveniente consiste en que, si bien HTML define qué teclas se deben presionar para acceder al elemento en cuestión, Internet Explorer interpreta el acceso rápido a través de la tecla **<Alt>**, mientras que Mozilla Firefox interpreta esta tecla a través de la combinación **<Alt + Shift>**, el navegador Safari realiza la acción a través de la tecla **<Ctrl>**



COMPLEMENTOS, PINCELES Y FILTROS

En HTML, los eventos son acciones que sirven para asociar tags HTML con los scripts que contiene el documento. Un script es un programa del lado del cliente y, mediante los eventos, cuando el usuario realiza una determinada acción (como pasar el mouse sobre un botón, por ejemplo), se realiza una llamada al script, que ejecuta su función (como podría ser, por ejemplo, la carga de una imagen). La sintaxis para definir un evento es similar a la que se emplea en la definición de atributos, y los eventos disponibles dependen del tag utilizado en cada caso.

JUNTO A JAVASCRIPT, LOS ATRIBUTOS DE EVENTO AYUDAN A DAR INTERACCIÓN A LOS DOCUMENTOS HTML.

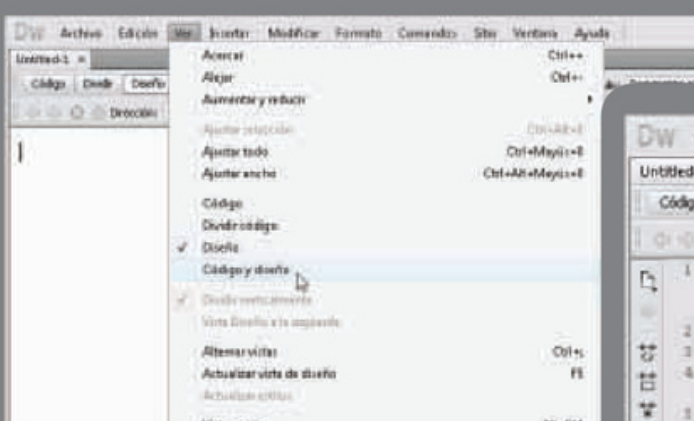
y, por último, Opera la interpreta a través de la combinación **<Shift + Esc>**. Esto quiere decir que el acceso a cada parte de una página utilizando las teclas depende, exclusivamente, del navegador que tenga el usuario. El atributo **tabIndex**, por su parte, exhibe su mayor funcionalidad cuando estamos manejando, por ejemplo, los formularios de login o de registro. Cuando un usuario se enfrenta a un formulario en el que debe ingresar sus datos en diferentes campos, al utilizar la tecla **<Tab>** para moverse entre ellos, lo hará en el orden indicado mediante este atributo. Para decirlo de otra manera, al definir el atributo **tabIndex** de cada elemento, estaremos definiendo la secuencia en la que el usuario entrará en cada campo si no recurre al mouse, es decir, si pasa de un campo a otro pulsando **<Tab>**. ◀



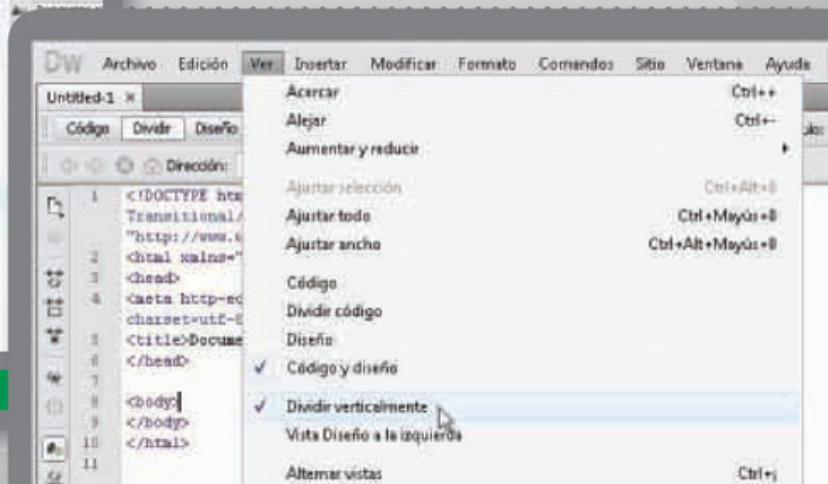
NUESTRAS PRIMERAS

PASO A PASO

Veremos, a continuación, cómo escribir código en Dreamweaver y observaremos los cambios que se producen en la sección de diseño.



01 | ESCRIBIR ETIQUETAS



DISPOSICIÓN DEL CÓDIGO | 02

Aunque ya creamos y modificamos algunas etiquetas en Dreamweaver para configurar el título de la página y agregar tags descriptivos, aún no hemos generado modificaciones en el contenido de la página, ya que no incorporamos textos ni imágenes. ◀

PASO | 01 Para escribir etiquetas en Dreamweaver desde el código, seleccionamos en la barra de vistas de la interfaz la que dice **[Dividir]**. Otra forma de hacerlo es ir al menú **[Ver]** y allí elegir **[Código y diseño]**.

PASO | 02 Luego, para que no se corten las líneas de código y el trabajo resulte más cómodo, en el mismo menú hacemos clic en **[Dividir verticalmente]**, de modo que la división de la pantalla sea horizontal.

PASO | 03 Dentro de la etiqueta **[body]**, presionamos la tecla **<Enter>** y comenzamos a escribir la primera etiqueta, que puede ser cualquiera de las existentes. En este caso, será **<div>**.

```

7
8 <body>
9
10 <div> I
11 </body>
12 </html>
13

```

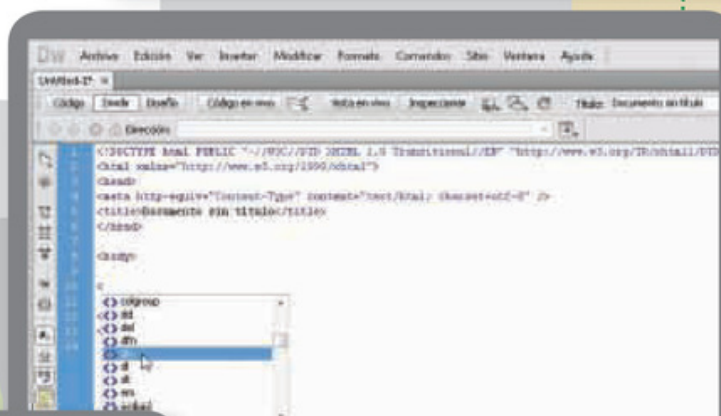
```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Documento sin título</title>
6 </head>
7 <body>
8 <div></div> I
9 </body>
10 </html>
11

```

04 | APERTURA Y CIERRE

ETIQUETAS HTML



GESTOR INTUITIVO

05

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Documento sin título</title>
6 </head>
7 <body>
8 <div>
9 </div>
10 </body>
11 </html>
12

```

03 | PRIMERA ETIQUETA

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; char
5 <title>Documento sin título</title>
6 </head>
7 <body>
8 <div>Esta es nuestra primera etiqueta</div> I
9 </div>
10 </body>
11 </html>
12

```

CIERRE FINAL

06

PASO | 04

Las etiquetas llevan una apertura y un cierre, lo que significa que, cuando abrimos una etiqueta, debemos cerrarla. Para cerrar **<div>**, debemos escribir **</div>**. Este cierre debe aparecer al final del contenido de la etiqueta; es decir, el contenido tiene que ir entre la apertura y el cierre.

PASO | 05

Dreamweaver nos da la posibilidad de utilizar un gestor de etiquetas. Esto nos facilita la escritura y nos da más rapidez a la hora de armar un sitio. Este gestor aparece cuando colocamos el signo **<** para abrir una etiqueta. Seleccionamos la etiqueta en el gestor, ayudados por el mouse o los cursores de dirección, y una vez hecho esto, presionamos la tecla **<Enter>** y escribimos el signo **>**.

PASO | 06

Cuando terminemos de escribir el contenido de la etiqueta, bastará con escribir **[</]** para que Dreamweaver cierre la etiqueta automáticamente con su utilidad de autocompletado. Luego, podremos ver el contenido que hayamos incorporado en la página.

RESUMEN

Con este paso a paso hemos visto cómo escribir etiquetas en el código de Dreamweaver. En las próximas páginas conoceremos las distintas etiquetas que podemos incorporar a nuestros documentos, y la forma en la que podemos utilizar y modificar cada una de ellas. ¡Con cada capítulo seremos más profesionales!

CSS CASCADING STYLE SHEETS

Como ya adelantamos al conocer las características de HTML, CSS nos permite mantener el contenido separado del diseño. La mayor ventaja de esta práctica es que podremos modificar muy fácilmente las características visuales de un sitio.

CSS es el lenguaje de **hojas de estilos** creado para definir determinados parámetros de un sitio web. Con él, se separa, por un lado, el contenido HTML, definiendo la función que tendrá cada elemento (como podría ser encabezado, párrafo, título, etcétera), y luego, se otorgan los aspectos visuales de cada uno de ellos. Los objetivos de usar hojas de estilo son los siguientes:

- ▶ Definir los aspectos visuales de un sitio web.
- ▶ Homogeneizar estos parámetros en todo el sitio.
- ▶ Actualizar o modificar su apariencia en forma clara y sencilla.

Algunos de los aspectos que es posible manipular con CSS son:

- ▶ Color de fondo o imagen de fondo.
- ▶ Tipografía (tamaño, familia, color, etc.).
- ▶ Aspectos de los links y sus estados (rollover, enlaces visitados y activos).
- ▶ Márgenes de los elementos, bordes, y otros.
- ▶ Efectos JavaScript.

¿CÓMO APLICAR CSS A NUESTRAS PAGINAS?

Entre las formas más comunes de aplicar CSS a un documento HTML, encontramos las siguientes:

- ▶ **CSS en el mismo documento:** esta forma se emplea cuando los estilos son específicos para un documento HTML o cuando son muy pocos. En este caso, se deben incluir dentro de una etiqueta **<style>** y siempre dentro de **<head>**, como vemos a la derecha:

Tengamos presente que una de las premisas fundamentales de CSS es la practicidad a la hora de modificar valores, y aplicar a todo el sitio el mismo estilo. Sin embargo, si aplicamos CSS de esta manera, el cambio que realicemos sólo afectará al archivo HTML en el que estemos trabajando.

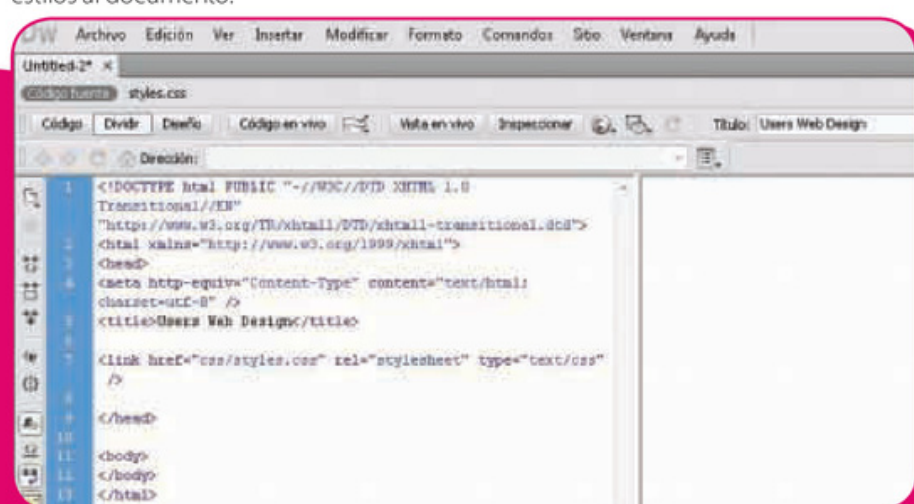
▶ **CSS como archivo externo:** ésta es la forma más usual y práctica, que consiste en aplicar un archivo CSS externo relacionado al HTML a través de la etiqueta **<link>**. Así, podremos aplicar todos los archivos CSS que hagan falta para darle estilos al documento:



```
<head>
```

```
<link href="css/styles.css"
rel="stylesheet" type="text/
css" />
```

```
</head>
```



```
<head>
<style type= text/css>
  h1{
    color: Blue;
    font-size:12px;
  }
</style>
</head>
```


Éste es el modo de trabajo más habitual, ya que, aplicando el mismo archivo CSS a las diferentes páginas de un sitio, garantiremos la homogeneidad de estilos. Esto también permite que, al variar algún valor de cualquier selector, éste se aplique a todo el sitio de forma rápida y sencilla.

UN CSS EXTERNO PERMITE ACTUALIZAR LOS ESTILOS DEL SITIO DE FORMA SENCILLA, MODIFICANDO SÓLO EL ARCHIVO CSS.

► **CSS en elementos HTML:** es un método que no resulta recomendable, aunque es necesario conocerlo. Este tipo de aplicación de CSS sólo se usa en circunstancias en las que, específicamente, es necesario aplicar un estilo a un elemento determinado. Por ejemplo:

```
<p style="color: black; font-family: Verdana;">Un párrafo de texto.</p>
```

SINTAXIS DE ATRIBUTOS Y PROPIEDADES EN CSS

Cuando creamos un estilo, estamos definiendo todas las características que tendrá un elemento HTML, es decir, su apariencia visual, su **estética**.

La sintaxis de CSS es muy simple y se mantiene siempre de la misma manera. Consta de un **selector**, que indica a qué elemento o etiqueta se le aplicará el estilo; y de **declaraciones** (una sola o varias), formadas por el nombre de la **propiedad** y un **valor**. Ambas partes, selector y declaración, determinan una **regla CSS**.

Es muy importante tener en cuenta la forma de escribir los estilos: primero, el **selector** y, luego, las **declaraciones**, que siempre van entre llaves ({ }). La declaración está formada por las **propiedades**, seguidas por dos puntos (;), y luego están los **valores**. Si hay más de una declaración, entonces utilizamos un punto y coma (;) y la nueva declaración.

Para que el código sea legible y fácil de recorrer con la vista, suele ponerse una declaración bajo la otra. Entonces, un archivo CSS quedaría de la siguiente manera:

```
body {
  color: #666;
  font-size: 11px;
  font-family: "Trebuchet MS", Arial, sans-serif;
  background-color: #4bc5ee;
  background-image: url(.../img/bg_body.jpg);
}
```

	PROPIEDADES	VALOR
A	{	font-family: Verdana;
SELECTOR		DECLARACIÓN



Con este libro tendremos todos los recursos del diseño a nuestro alcance.

NUEVAS CARACTERÍSTICAS DE CSS3

Hace tiempo se está hablando de la nueva versión CSS y, aunque no es operativa en todos los navegadores, ya podemos ver algunas de las novedosas ventajas que trae aparejadas. Vamos a conocerlas.

En muchos casos, las nuevas características de CSS nos harán ahorrar tiempo y simplificar el código. En otros casos, nos permitirán ampliar las posibilidades estéticas que podemos utilizar para nuestros sitios. De todas maneras, debemos ser pacientes antes de decidirnos por aplicar estos nuevos atributos, ya que no son compatibles con todos los navegadores.

BORDES

A la hora de darles estilo a nuestros bordes, contamos con las siguientes novedades:

- ▶ **Border-color:** los bordes pueden tener degradados.
- ▶ **Border-image:** los bordes pueden crearse con diferentes imágenes.
- ▶ **Border-radius:** las cajas pueden tener bordes redondeados.
- ▶ **Box-shadow:** es posible aplicar sombra a las cajas.

FONDOS

Para establecer el estilo de los fondos, las novedades son:

- ▶ **Background-origin y background-clip:** tres modos de posicionar el fondo con respecto a las cajas.
- ▶ **Background-size:** permite especificar el tamaño de la imagen de fondo en píxeles o en porcentaje.
- ▶ **Multiple backgrounds:** permite usar más de una imagen como fondo del mismo elemento, separando con comas los nombres de cada imagen.

COLOR

Novedades para trabajar con colores:

- ▶ **Colores HSL:** ahora se pueden llamar a los valores por color, saturación y luminosidad.
- ▶ **Colores HSLA:** en este caso, es posible agregar la opacidad al modelo anterior.
- ▶ **Colores RGBA:** al conocido RGB se le suma el canal alpha, con el que es posible otorgar transparencias a los elementos.

TEXTO

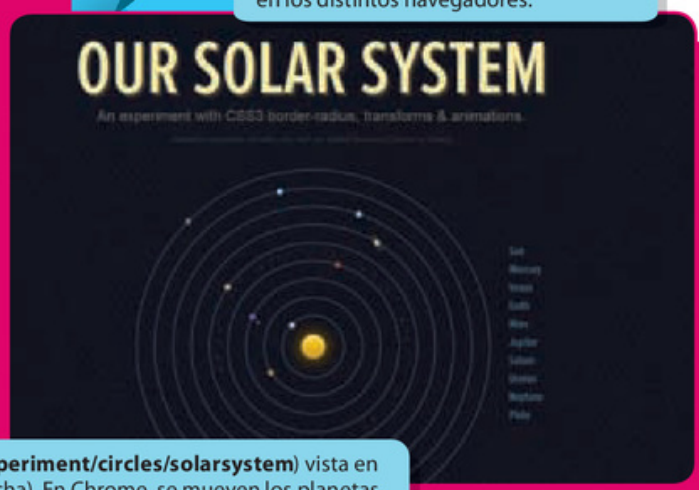
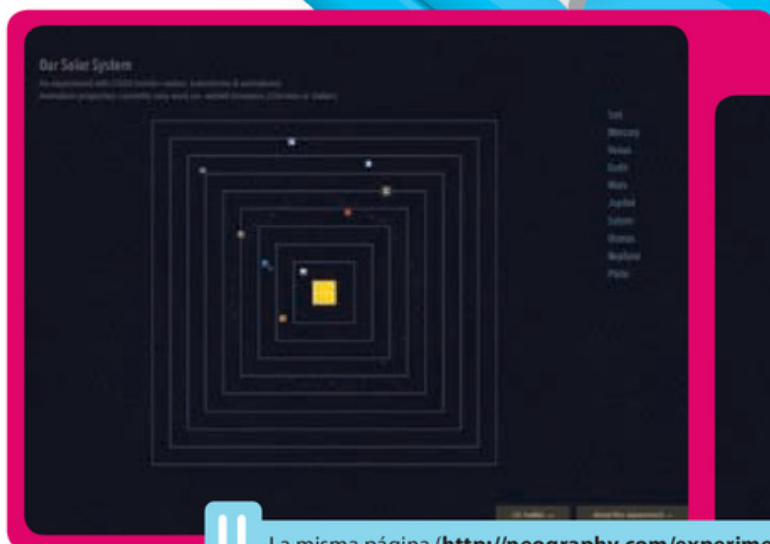
Para manejar el texto, contamos con las siguientes opciones:

- ▶ **Text-shadow:** efectos de sombras que evitan la necesidad de incluir imágenes, lo que hace que la página se cargue más rápido. Ya podemos usar esta propiedad porque es compatible con todos los navegadores.
- ▶ **Text-overflow:** cuando el texto no entra en su contenedor, agrega puntos suspensivos (...).
- ▶ **Text-Wrap:** separa las palabras largas que no entran en una línea.

INTERFAZ DE CSS 3

- ▶ **Box-sizing:** nuevo tributo que permite especificar las dimensiones de las cajas de forma diferente a la tradicional.
- ▶ **Resize:** gracias a este atributo, las cajas se pueden redimensionar.
- ▶ **Outline:** permite crear marcos y bordes dobles, especificando en píxeles la separación.

Esta página experimental se visualiza de diferentes maneras en los distintos navegadores.



La misma página (<http://neography.com/experiment/circles/solarsystem>) vista en Explorer (a la izquierda) y en Firefox (a la derecha). En Chrome, se mueven los planetas.

SOPORTE DE CSS EN LOS NAVEGADORES

Seguramente, nos encontraremos con más de un sitio que se ve diferente en distintos navegadores. Esto sucede porque cada browser **renderiza** (interpreta) el código CSS a su manera, y esto se debe al motor que posee cada navegador. Los motores se encargan de interpretar el código HTML y CSS, pero no todos lo hacen del mismo modo.

Por este motivo, si queremos que todos los usuarios de nuestros sitios puedan ver lo mismo en sus computadoras, deberemos tener cuidado al implementar CSS y testear los sitios en detalle.

Existen varias aplicaciones que nos permitirán realizar este testing, y algunas alternativas que podemos encontrar online. Vale la pena investigar sobre las novedades y opciones disponibles para mantenerse actualizado. ◀



CSS EN LOS NAVEGADORES

Soporte de las distintas versiones de CSS en los navegadores más conocidos.

NAVEGADOR	MOTOR	CSS 1	CSS 2	CSS 3
▶ Internet Explorer	▶ Trident	▶ Completo (desde la versión 6.0)	▶ Completo (desde la versión 8.0)	▶ Prácticamente nulo
▶ Firefox	▶ Gecko	▶ Completo	▶ Casi completo	▶ Selectores, pseudo-clases y algunas propiedades
▶ Safari	▶ WebKit	▶ Completo	▶ Casi completo	▶ Todos los selectores, pseudo-clases y muchas propiedades
▶ Opera	▶ Presto	▶ Completo	▶ Casi completo	▶ Todos los selectores, pseudo-clases y muchas propiedades
▶ Google Chrome	▶ WebKit	▶ Completo	▶ Casi completo	▶ Todos los selectores, pseudo-clases y muchas propiedades



SOPORTE CSS3 EN LOS NAVEGADORES

Conozcamos los diferentes soportes de los navegadores, según el sistema operativo que empleemos, a las novedades de CSS3.

	MAC				WIN								
	CHROME	FIREFOX	OPERA	SAFARI	CHROME	FIREFOX		OPERA		SAFARI	IE		
	5	3.6	10	4	4	3.6	3	0	5	4	6	7	8
RGBA	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No
HSLA	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No
MULTIPLE BACKGROUNDS	Sí	Sí	No	Sí	Sí	Sí	No	No	Sí	Sí	No	No	No
BORDER IMAGE	Sí	Sí	No	Sí	Sí	Sí	No	No	Sí	Sí	No	No	No
BORDER RADIUS	Sí	Sí	No	Sí	Sí	Sí	Sí	No	Sí	Sí	No	No	No
BOX SHADOW	Sí	Sí	No	Sí	Sí	Sí	No	No	Sí	Sí	No	No	No
OPACITY	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No
CSS ANIMATIONS	Sí	No	No	Sí	Sí	No	No	No	No	Sí	No	No	No
CSS COLUMNS	Sí	Sí	No	Sí	Sí	Sí	Sí	No	No	Sí	No	No	No
CSS GRADIENTS	Sí	Sí	No	Sí	Sí	Sí	No	No	No	Sí	No	No	No
CSS REFLECTIONS	Sí	No	No	Sí	Sí	No	No	No	No	Sí	No	No	No
CSS TRANSFORMS	Sí	Sí	No	Sí	Sí	Sí	No	No	Sí	Sí	No	No	No
CSS TRANSFORMS 3D	Sí	No	No	Sí	Sí	No	No	No	No	Sí	No	No	No
CSS TRANSITIONS	Sí	No	No	Sí	Sí	No	No	No	Sí	Sí	No	No	No
CSS FONTFACE	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí	Sí	Sí	Sí	Sí

SELECTORES

Las reglas CSS son la base de cualquier sitio que desarrollemos hoy en día. Su correcto armado y declaración favorecerá la optimización del sitio e influirá de forma positiva en su velocidad de carga.

Nuestra hoja de estilos CSS contendrá distintos tipos de **selectores**, que son los que le indican al navegador de Internet dónde aplicar los estilos que definimos. Cada regla CSS definida puede aplicarse a cuantos elementos HTML necesitemos y, a su vez, cada elemento HTML puede utilizar cualquier cantidad de reglas que hayamos creado. Aunque estos conceptos son un tanto confusos al principio, se irán aclarando a medida que avancemos y practiquemos con ellos.

LOS SELECTORES LE INDICAN AL NAVEGADOR DE INTERNET DÓNDE APLICAR LOS ESTILOS DEFINIDOS.

SELECTOR UNIVERSAL

Este selector se usa para aplicar estilos a todos los elementos HTML de una página, y se indica mediante un asterisco:

```
* {  
    height: 50px;  
}
```

El ejemplo anterior establece un alto de 50 píxeles para todos los elementos de las páginas a las que se encuentre vinculada nuestra hoja de estilos. Aunque es una forma sencilla de trabajar, no es muy utilizada porque no es común que el mismo estilo pueda aplicarse a todos los elementos de una página.

SELECTOR DE TIPO O ETIQUETA

Se utiliza para aplicar estilos a todos los elementos HTML que coincidan con el nombre del selector. Para usarlo, sólo es necesario nombrar el elemento HTML (tag) sin los caracteres de apertura y de cierre, es decir, sin < ni >. Por ejemplo, a continuación definimos un borde de color gris para todos los elementos div de nuestra página:

```
div {  
    border: 1px solid #CCC;  
}
```

Si queremos aplicar el mismo estilo a más de un elemento HTML, podemos encadenar los selectores, separándolos por una coma. Con el siguiente ejemplo veremos un borde gris en todos los elementos div, en todas las imágenes y en todos los párrafos de la página:

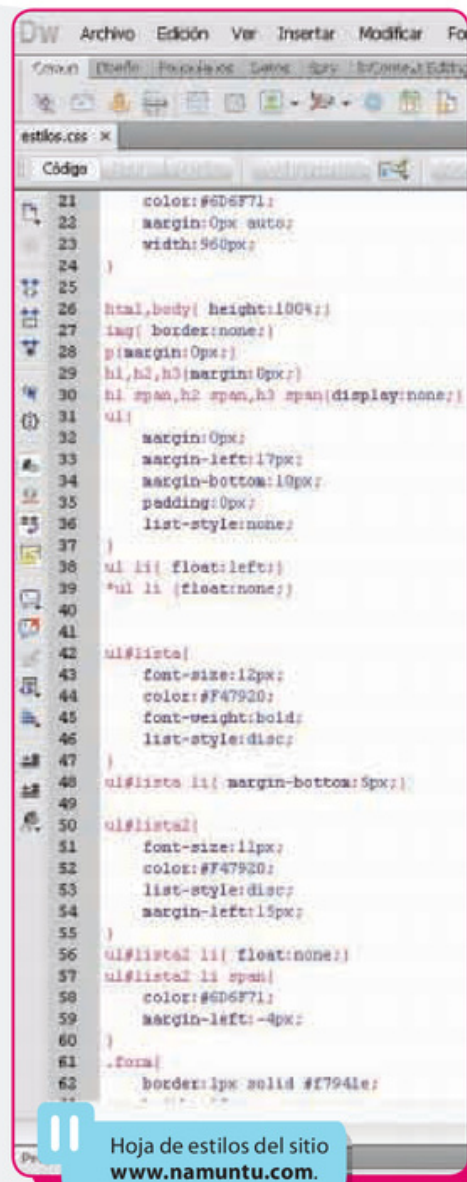
```
div, img, p {  
    border: 1px solid #CCC;  
}
```

En las hojas de estilo complejas, suelen agruparse los estilos comunes a varios elementos en una única regla y, luego, se definen los estilos específicos de cada elemento HTML por separado. Siguiendo con el ejemplo anterior, podemos lograr que todos los elementos div, las imágenes y los párrafos de nuestra página se vean con un borde gris, y que sólo los elementos div tengan una altura de 50 píxeles. Veamos cómo:

```
div, img, p {  
    border: 1px solid #CCC;  
}  
div {  
    height: 50px;  
}
```

SELECTOR DESCENDENTE

Este selector se utiliza para seleccionar un elemento HTML que se encuentra dentro de otro del mismo tipo. Los selectores descendentes siempre están formados por dos o más selectores, separados entre sí por un espacio.



El último selector indica el elemento HTML al que se le debe aplicar el estilo. A continuación, aplicamos color rojo a todos los elementos `<span>` que estén dentro de un párrafo (`<p>`). A aquellos que estén por fuera de los párrafos, no se les aplicará este estilo:

```
p span{
    color:#F00;
}
```

SELECTOR DE CLASE

Se define utilizando un punto al comienzo del selector, forma en la que el navegador de Internet lo distingue de los otros tipos de selectores. Supongamos que tenemos el siguiente código HTML:

```
<ul>
  <li>Primer lugar</li>
  <li>Segundo lugar</li>
  <li>Tercer lugar</li>
</ul>
```

Si quisiéramos aplicar un estilo sólo al primer elemento `<li>`, nos sería imposible hacerlo con los tipos de selectores que conocimos hasta ahora. Una de las soluciones es aplicar el atributo `[class]` sobre el elemento HTML al que le queremos aplicar el estilo. Entonces, nuestro código HTML quedaría así:

```
<ul>
  <li class="principal">Primer
    lugar</li>
  <li>Segundo lugar</li>
  <li>Tercer lugar</li>
</ul>
```

Luego, definimos el estilo (en este caso, fijamos la tipografía de color rojo), utilizando un selector de clase, de la siguiente manera:

```
.principal{
    color:#F00;
}
```

De igual modo, cualquier elemento HTML que tenga definido su atributo `[class]` como principal tendrá su tipografía de color rojo. Por ejemplo:

```
<span class="principal">Hola<
 /span>
<p class="principal">Chau</p>
```

Ahora bien, si quisiéramos la tipografía roja sólo en los párrafos cuyo atributo `[class]` ha sido definido como principal, deberíamos definir la regla CSS así:

```
p.principal{
    color:#F00;
}
```



A la izquierda, podemos ver el sitio www.namuntu.com con diferentes estilos aplicados en los elementos HTML. A la derecha, el mismo sitio sin la hoja de estilos.



SELECTOR ID

Este tipo de selector es igual al de clase, aunque lo aplicaremos a un único elemento HTML de nuestra página, porque no se permite usar varios atributos id con el mismo valor.

Para un solo elemento HTML, también podríamos utilizar un selector de clase, pero en ese caso nos convendrá emplear este selector de ID porque es más eficiente.

Para utilizar este tipo de selector, debemos definir el atributo **[id]** del elemento HTML al que le queremos aplicar nuestro estilo. La sintaxis es muy parecida a la del selector de clase, pero en vez de un punto, utilizaremos el numeral o **sharp** (#). En nuestro ejemplo, veremos el código HTML:

```
<ul>
  <li>Primer lugar</li>
  <li id="nuevo">Segundo
    lugar</li>
  <li>Tercer lugar</li>
</ul>
```

Y el respectivo código CSS:

```
#nuevo{
  color:#F00;
}
```

Al igual que en los selectores de clase, podemos restringir el alcance del selector combinando el de ID con otros, por ejemplo:

```
a#importante{
  font-size:14px;
}
```

Hasta aquí, hemos visto los distintos tipos de selectores que utilizaremos en nuestra actividad como diseñadores profesionales de páginas web. Este tipo de definiciones resultan complejas al principio, pero con horas de práctica y trayendo a la memoria lo que encontramos, cada vez con más frecuencia, con este tipo de trabajo, lo que hará que nos fijemos en ellos y que hagamos cosas con suma naturalidad.



SELECTORES DE CLASE

En las hojas de estilo CSS, debemos prestar mucha atención a diferencias como las que vemos aquí.

```
p.principal{
  color:#F00;
}
```

Aquí aplicamos nuestro estilo a todos los párrafos que tengan su atributo **[class]** definido como principal.

```
p .principal{
  color:#F00;
}
```

Aquí aplicamos nuestro estilo a todos los elementos HTML que tengan su atributo **[class]** definido como principal y se encuentren dentro de un párrafo.

```
p .principal{
  color:#F00;
}
```

Aquí aplicamos nuestro estilo a todos los párrafos y a todos los elementos HTML que tengan su atributo **[class]** definido como principal.

Otro aspecto importante de los selectores de clase es la forma de aplicar los estilos de varias clases a un mismo elemento HTML. Supongamos que tenemos los siguientes estilos:

```
.titulo{
  font-weight:bold;
}
.importante{
  color:#F00;
}
```

Podemos aplicar ambas clases a un mismo elemento HTML si separamos los valores de su atributo **[class]** con un espacio en blanco:

```
<p class="titulo importante">Hola que tal</p>
```

El contenido del párrafo se verá en negrita y con la tipografía de color rojo.



TABLA DE ESTILOS

A la hora de trabajar con CSS, tendremos que recordar una gran cantidad de propiedades. A continuación, encontraremos un listado que contiene las utilizadas con mayor frecuencia.

Para dar estilo a nuestros documentos HTML mediante CSS, utilizaremos una serie de propiedades cuyos nombres deberemos tener bien presentes. Cuando recién comenzamos, recordar todo el

listado resulta bastante difícil. Por eso, aquí desplegamos una tabla donde podremos ver las funciones y propiedades más usuales y útiles para dar formato al texto y al fondo de nuestras creaciones. De esta forma, tendremos un

panorama de las diversas opciones de estilos que nos brinda CSS y, al mismo tiempo, una ayuda a la memoria que nos ahorrará tiempo de búsqueda para recordar cómo se declara una determinada propiedad.



PROPIEDADES CSS PARA TEXTOS Y FONDOS

Propiedades más comúnmente utilizadas en CSS para textos y fondos.

	PROPIEDAD	DESCRIPCIÓN	VALORES	EJEMPLOS
TEXTO	word-spacing	Determina la cantidad de espacios en blanco entre palabras.		P {word-spacing: 0.5em;}
	letter-spacing	Determina la cantidad de espacio en blanco entre letras.		{letter-spacing: 0.5em;}
	text-decoration	Establece la decoración de la fuente, como el subrayado.	none; blink; line-through; overline; underline;	{text-decoration: underline;}
	vertical-align	Establece la alineación vertical respecto al elemento padre.	baseline; bottom; middle; sub; super; text-bottom; text-top; top;	{vertical-align: baseline;}
	text-transform	Define mayúsculas y minúsculas dentro del texto.	capitalize; uppercase; lowercase; none;	{text-transform: capitalize;}
	text-align	Establece la alineación horizontal del texto respecto a un elemento.		{text-align: justify;}
	text-indent	Tabulación de la primera línea de texto.		{text-indent: 10px;}
	text-height	Interlineado del texto.		{text-height: 150%;}
FONDO	color	Color de fondo.		{color: #ff3300;}
	background-color	Color de fondo del elemento.		{background-color: #BBBBBB;}
	background-image	Imagen de fondo.		{background-image: url (img/fondo.gif);}
	background-repeat	Forma de repetición de la imagen de fondo.	repeat; (mosaico), no-repeat; repeat-x; (horizontal), repeat-y; (vertical)	{background-repeat: no-repeat;}
	background-attachment	Determina si la imagen de fondo sigue o no el scroll.	scroll; fixed;	{background-attachment: fixed;}
	background-position	Punto de comienzo de un fondo (imagen o color).	position (x y) o (x% y%); top; center; bottom; left; right;	{background-position: top center;}
	background	Permite definir varias propiedades del fondo de forma simultánea. No es necesario que los valores respeten un orden.		{background: #BBBBBB url(img/fondo.gif) no-repeat fixed center;}

ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 ¿Cuáles son las diferencias entre **HTML** y **XHTML**?

02 ¿Cuáles son los beneficios de **XHTML**?

03 ¿En qué sitio podemos validar nuestra web?

04 ¿Qué es el **body**?

05 ¿Cómo definimos una etiqueta?

06 ¿Qué son los atributos?

07 ¿En qué grandes grupos **HTML** clasifica los elementos?

08 Mencione los atributos dinámicos de eventos.

09 ¿Cómo podemos aplicar **CSS** a nuestras páginas?

10 Nombre diferentes tipos de selectores de **CSS**.



PRÁCTICAS

01 Cree un nuevo documento en **HTML** con Dreamweaver como editor y guárdelo.

02 Dentro del código **HTML** de su sitio: divida un texto en párrafos, realice un salto de línea y transfórmelo en una lista numerada.

03 Cree un archivo de **CSS** y aplíquelo a su documento **HTML**.

04 En el archivo anteriormente creado, defina los bordes, fondo, color y texto de su sitio.

05 Busque un sitio que sea de su agrado e imite el diseño por medio de **CSS** en un archivo aparte del anterior.



CSS ACCESIBLE

Una de las razones por las cuales conviene usar CSS es que hace más práctico y veloz el mantenimiento de los sitios. Sin embargo, su uso incorrecto puede causar muchos inconvenientes.

No es raro que, en sitios relativamente grandes, el código supere las 1000 líneas. Si no nos manejamos con suma organización, encontrar algo en él será realmente difícil, y perderemos mucho tiempo hasta ubicar, por ejemplo, el estilo que controla a un elemento. Como estamos avanzando en nuestras creaciones, es momento de conocer algunas recomendaciones para hacer la tarea bien desde el comienzo, y evitar inconvenientes e incomodidades.

ID Y CLASS

Primero debemos diferenciar dónde debe usarse `id` y dónde, `class`, ya que es muy usual que se generen confusiones en torno a su aplicación, y un mal uso puede hacer que nuestro sitio no valide ante el W3C. Un `id` identifica a un elemento HTML, que debe ser el único en la página. Por el contrario, las clases pueden (y deben) ser reutilizadas cuantas veces sean necesarias.

Otra cuestión que debemos tener en cuenta es no abusar de estos elementos, es decir, no caer en una creación indiscriminada. Un ejemplo de este mal uso sería el siguiente:

```
<h1 class="titulo">Este
es un titulo</h1><ul
class="lista"> <li
class="item">Primer
elemento</li><li
class="item">Primer
elemento</li> <li
class="item">Primer elemen-
to</li> </ul>
<p class="texto">
Ejemplo de texto</p>
```

En una correcta utilización de las etiquetas HTML, podemos aplicar los estilos directamente, sin tener que utilizar clases o `id`. En este caso, el ejemplo correcto sería:

```
<h1>Este es un titulo</h1>
<ul> <li>Primer elemento</li>
<li>Primer elemento</li>
<li>Primer elemento</li> </ul>
<p> Ejemplo de texto</p>
```

ORDEN EN EL CSS

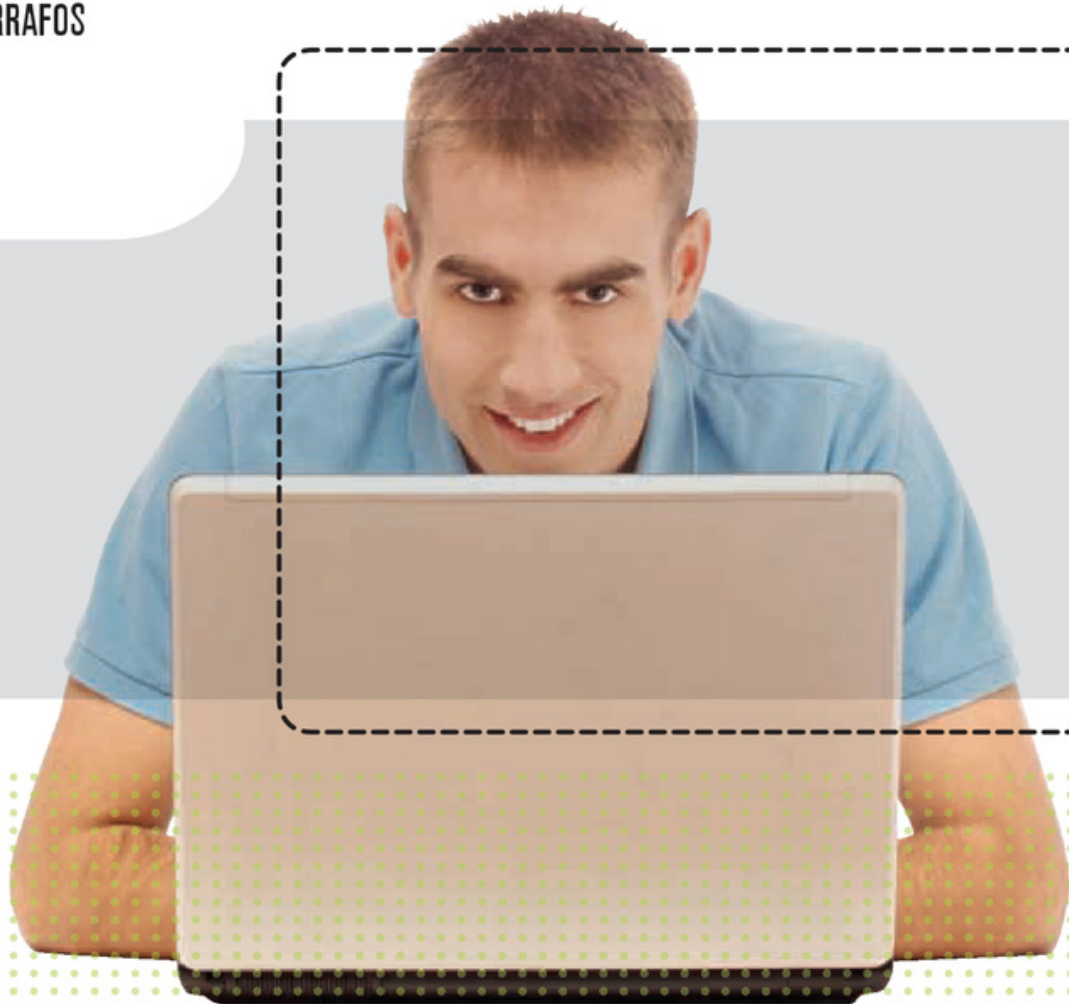
Con respecto a la escritura del CSS, es fundamental mantener el orden. Veamos algunas recomendaciones:

- ▶ Agrupar los estilos pertenecientes a una determinada estructura.
- ▶ Realizar encabezados para cada grupo de selectores.
- ▶ Utilizar comentarios para que los códigos sean más fáciles de leer y trabajar. Los comentarios se insertan entre los signos `/*` y `*/`. Un buen uso de ellos es para delimitar las secciones de código.
- ▶ Indentar el código correctamente simplifica su lectura.
- ▶ Utilizar el método de escritura shorthand cuando sea posible, debido a que acorta mucho el código y facilita su visualización.
- ▶ Como en CSS los elementos heredan atributos, si en nuestro elemento `<body>` definimos el color de las fuentes, no será necesario hacerlo si tenemos `<h1>`, `<h2>`, `<h3>` u otras clases que lleven el mismo color, ya que éstas tomarán el que hayamos definido para `<body>`. Si establecemos colores en todos los casos, tendremos muchos contratiempos cuando deseemos cambiar uno, porque habrá colores por todo el código.
- ▶ Si resulta necesario incluir otra hoja de estilo CSS, es aconsejable agregar pocas, dado que el objetivo debe ser centralizar el código.
- ▶ Debemos usar hojas de estilo **vinculadas** en vez de estilos incrustados directamente en las etiquetas, como así también tenemos que evitar las hojas de estilo incrustadas (directamente en la página).
- ▶ Siempre es mejor utilizar medidas relativas, excepto cuando las del elemento sean inamovibles, como sucede con las imágenes, por ejemplo.
- ▶ Mantener una misma forma de escritura a lo largo de todo el código. Hay dos maneras para hacerlo: en una línea (**Código 1**) o en bloque (**Código 2**).

XHTML y CSS: texto



- ▶ ESTILOS CSS PARA TEXTOS
- ▶ HOJAS DE ESTILO DE DREAMWEAVER
- ▶ TEXTO: PÁRRAFOS Y TÍTULOS
- ▶ MARCADO BÁSICO Y CARACTERES ESPECIALES
- ▶ ESCRIBIR TÍTULOS Y PÁRRAFOS



ESTILOS CSS PARA TEXTOS

Como ya mencionamos, con CSS podremos aplicar diferentes y variados estilos al texto de la página web que estemos creando. Veamos qué podemos modificar de esta manera.

En esta oportunidad, analizaremos las propiedades más comunes de CSS que podemos utilizar para modificar los textos de nuestros sitios. Con ellas será posible determinar desde el tamaño y la familia tipográfica, hasta el color y la decoración que tendrá la fuente elegida, entre otros aspectos.

COLOR

Como su nombre lo indica, esta propiedad sirve para definir el color de la tipografía. Por defecto, el que utilizan la mayoría de los navegadores es el negro.

```
div{ color:#069; }
span{ color: blue; }
a{ color:#FF9900; }
p{ color: rgb(120,40,40); }
```

En el ejemplo anterior, vemos las diferentes formas de establecer el color para un elemento HTML. Esta propiedad se hereda a todos los **hijos** que se encuentren dentro de un elemento HTML, con excepción de los enlaces. Por esta razón, es muy común definir el color en el elemento **<body>**. Si quisiéramos definir el mismo color para todos los elementos de la página, incluyendo los enlaces, tendríamos que utilizar esta propiedad de la siguiente forma:

```
body, a{ color:#006699; }
```

FONT-FAMILY

Esta propiedad sirve para indicar la tipografía que utilizaremos en el elemento HTML donde sea aplicada. La manera más común de establecer la tipografía del texto es mediante el nombre del tipo de letra, por ejemplo:

```
body{ font-family: Arial, Helvetica, sans-serif; }
```

Las tipografías que elegimos para nuestro elemento HTML deberán estar instaladas en la computadora del usuario que visite la página. Por esta razón, es conveniente elegir tipografías comunes, que sepamos que todos los usuarios tendrán en sus equipos. En el ejemplo anterior, en caso de que el usuario no tenga instalada la tipografía **Arial** (algo que es prácticamente imposible), la página visualizará la **Helvética**

y, si no, mostrará la familia tipográfica genérica **sans-serif**, que le indica al navegador que elija una tipografía sin serif entre las que estén instaladas en la computadora del usuario.

FONT-SIZE

Con esta propiedad definimos el tamaño de la tipografía, lo cual puede hacerse de diferentes maneras.

- Indicando su **tamaño absolute** con alguna de estas palabras clave: **xx-small**, **x-small**, **small**, **medium**, **large**, **x-large** y **xx-large**.
- Indicando su **tamaño relativo**: se utiliza **smaller** o **larger** para definir el tamaño de la tipografía en relación con el tamaño de la tipografía del elemento padre.
- Indicando su tamaño en **píxeles**: ésta es una de las prácticas más comunes.



En www.chilaweb.com.ar vemos cómo se aplicaron distintos estilos a los textos.



Es importante contar con un teclado cómodo para no tener una mala posición al tipear.

- ▶ Indicando su tamaño en **puntos**: es recomendable usar esta medida cuando estamos desarrollando una página optimizada para ser impresa.
- ▶ Indicando su tamaño con la unidad **em**: esta unidad indica la relación con el tamaño de tipografía del elemento padre. Por ejemplo, si el tamaño de la tipografía en `<body>` es de **10 px** y tenemos un `<div>` con tamaño **1.2 em**, el equivalente será **12 px**. Lo mismo sucede si queremos usar un porcentaje para definir el tamaño de la tipografía. Siguiendo con el ejemplo anterior, si el tamaño en `<body>` es de **10 px** y tenemos un `<div>` con tamaño **120%**, el equivalente será **12 px**.

CSS RECOMIENDA UTILIZAR LA UNIDAD DE **MEDIDA EM** O EL PORCENTAJE (%) PARA INDICAR EL **TAMAÑO DEL TEXTO**.

CSS recomienda utilizar la unidad **em** o el porcentaje (%) para indicar el tamaño del texto. Veamos, en el siguiente ejemplo, definiciones de tamaños realizadas de diferentes maneras:

```
div{ font-size:12px; }
span{ font-size: small; }
p{ font-size: 18pt; }
```

FONT-WEIGHT

Esta propiedad se utiliza en caso de querer transformar una tipografía en **negrita**. Los valores que emplearemos son **normal** o **bold**, tal como vemos en el siguiente ejemplo, que aplica **negrita** en un párrafo:

```
p{ font-weight: bold; }
```

Vale mencionar que, si bien utilizando esta propiedad con el valor **bold** podemos lograr el mismo aspecto visual que se obtiene al aplicar la etiqueta `<strong>`, la principal diferencia es que, usando código CSS, se pierde el concepto de **marcado semántico** otorgado por las etiquetas HTML.

FONT-STYLE

Esta propiedad se aplica, en general, cuando queremos mostrar la tipografía en **cursiva**. Su método de empleo es el siguiente:

```
p{ font-style: italic; }
```

TEXT-DECORATION

Esta propiedad se utiliza cuando necesitamos subrayar –o no– una tipografía. Es común usarla cuando queremos quitarle el subrayado por defecto que traen los **links**. Por ejemplo:

```
a{ text-decoration: none; }
```

Para subrayar, por el contrario, la utilizamos de la siguiente manera:

```
span{ text-decoration: underline; }
```

LINE-HEIGHT

Esta propiedad define el **interlineado** del elemento HTML donde la apliquemos. Es conveniente que la unidad de nuestro interlineado tenga relación con el tamaño de tipografía que estemos usando. Por ejemplo:

```
p{ font-size:12px;
  line-height: 1.5em; }
```

TEXT-ALIGN

Con esta propiedad definimos la alineación del contenido de los elementos de bloque y de las celdas de tabla HTML a donde la apliquemos. Alinear no sólo el texto que contenga el elemento, sino también todo su contenido, como las imágenes y los párrafos. Los valores que podremos usar son: **left**, **right**, **center** o **justify**.

Hasta aquí hemos visto los estilos que podemos aplicar a las tipografías y a las cajas de texto que vayamos a usar en una página. ◀



TAMAÑOS ESTÁNDAR

Por defecto, los navegadores asignan los siguientes tamaños a las etiquetas de títulos:

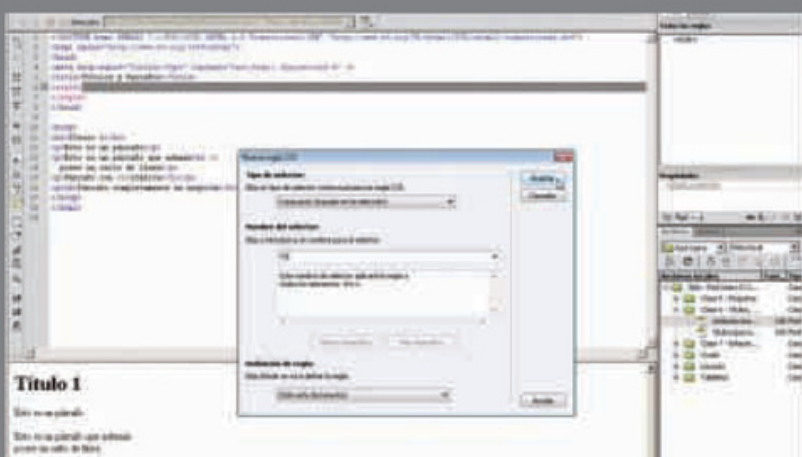
```
<h1> = xx-large
<h2> = x-large
<h3> = large
<h4> = medium
<h5> = small
<h6> = xx-small
```


HOJAS DE ESTILO DE PASO A PASO

En este apartado veremos la forma de crear y utilizar las hojas de estilo, también llamadas CSS. Hacerlo desde Dreamweaver nos facilita la tarea, ya que el programa cuenta con herramientas especiales para ello.



01 | REGLA



SELECTOR | 02

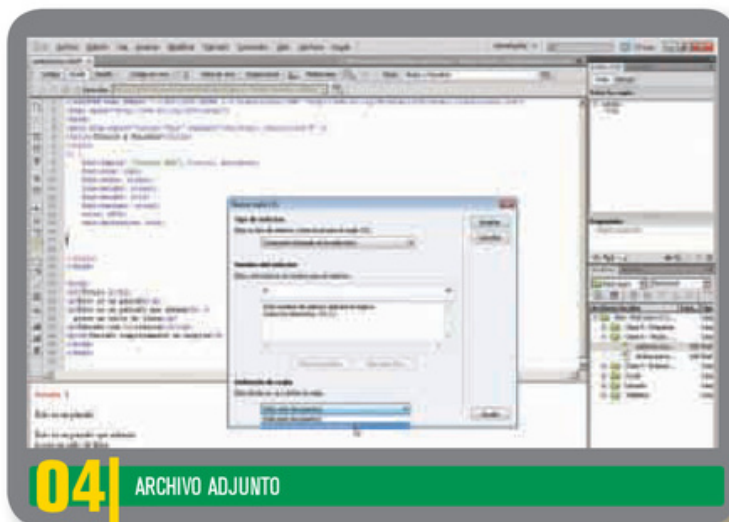
Usaremos las propiedades de las CSS desde el documento HTML, y veremos la forma en que podemos crearlas y modificarlas. ◀

PASO | 01 En el interior de la etiqueta `<head>`, creamos una etiqueta `<style>` con su cierre `</style>`. Luego, vamos al panel derecho que se denomina **Estilos CSS** y hacemos clic en **Nueva regla CSS**.

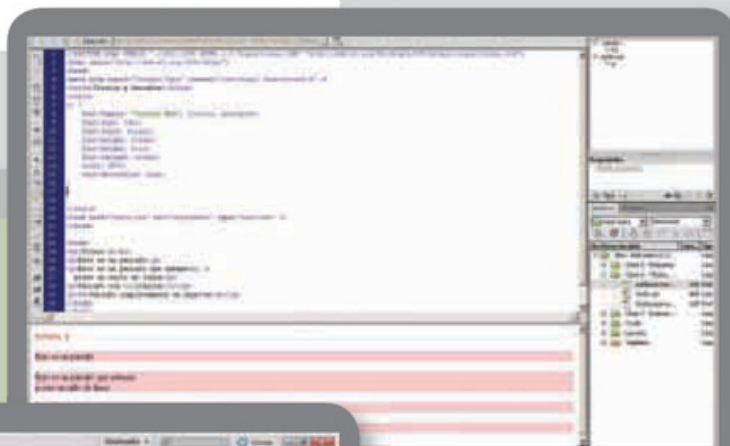
PASO | 02 En la ventana **Nueva regla CSS** que aparece elegimos, en **Tipo de selector**, la opción **Compuesto (basado en la selección)**. En **Nombre del selector** colocamos el nombre de la etiqueta a la que le aplicaremos las propiedades, en este caso, `<h1>`. En **Definición de regla** dejamos **[(Sólo este documento)]** y presionamos **Aceptar**.

PASO | 03 Se abrirá la ventana **Definición de regla para h1**, donde podremos modificar la regla. Dentro de la sección **Categoría**, la opción **Tipo**, contiene lo relacionado con la tipografía, y **Fondo** lo concerniente al fondo. **Bloque** afectará a los distintos bloques de texto y **Cuadro** determinará las medidas de los bloques (alto, ancho, márgenes, paddings, etcétera). **Borde** establece el tipo de borde y sus propiedades; **Lista**, las características de las listas ordenadas o desordenadas; **Posición**, la ubicación dentro del navegador, por ejemplo, y **Extensiones** permite aplicar un filtro o un tipo de cursor personalizado, entre otras opciones. En este caso, para la etiqueta `<h1>`, cambiamos las propiedades dentro de **Tipo** y observamos el efecto presionando **Aplicar**.

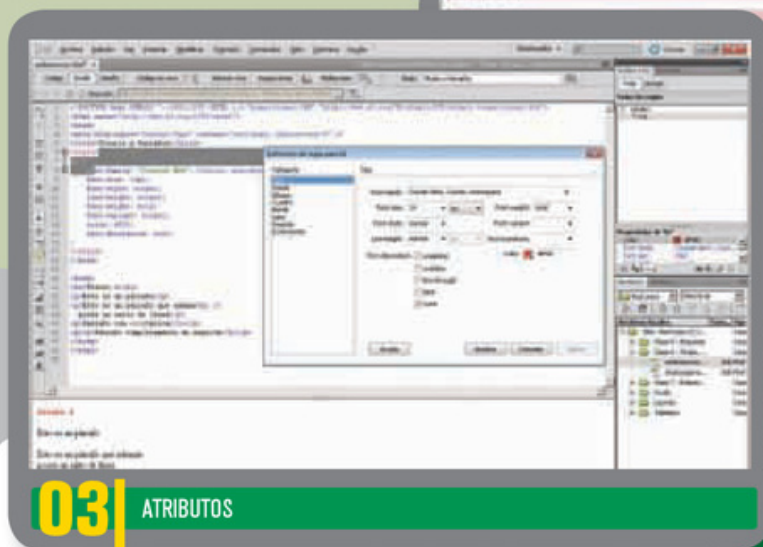
DREAMWEAVER



04 ARCHIVO ADJUNTO



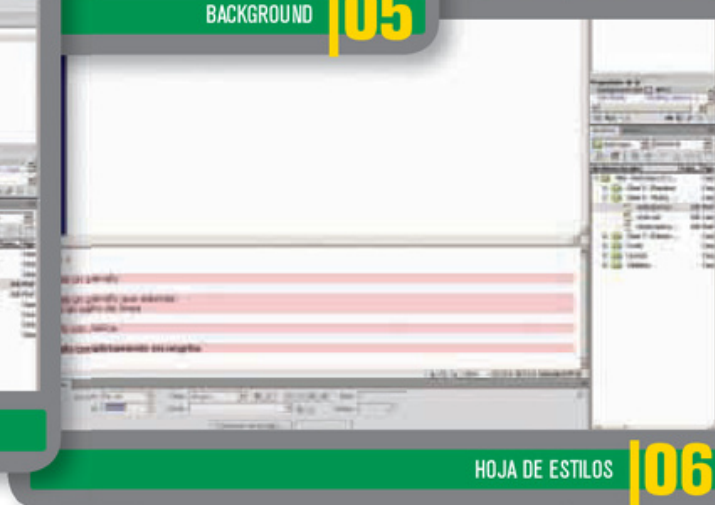
05 BACKGROUND



03 ATRIBUTOS

PASO | 04 Para crear un archivo adjunto que esté relacionado con nuestro archivo HTML, generalmente llamado **[style.css]**, lo generamos repitiendo las opciones del paso 2 pero eligiendo **[Nuevo archivo de hoja de estilo]** dentro de **[Definición de regla]** en la ventana **[Nueva regla CSS]**. Una vez hecho esto, deberemos elegir el destino donde será guardado el archivo y confirmaremos las preguntas para almacenarlo.

PASO | 05 Luego de elegir la etiqueta **<p>** en la ventana anterior, en la categoría **[Fondo]** definimos un nuevo color en **[Background-color]**. Ese color que definimos se aplicará a todas las etiquetas **<p>** que tengamos en el documento, como podemos ver en la imagen. Los estilos se incorporan al código dentro de la etiqueta **<head>** y podemos encontrarlos en el panel **[Estilos CSS]**.



06 HOJA DE ESTILOS

PASO | 06 Para modificar la hoja de estilos, podemos hacer clic en el botón que aparece debajo del nombre de la página con la denominación de la hoja que creamos en el paso 4 (en este caso, **[style.css]**). Veremos el código de nuestra hoja de estilos y podremos hacer cambios escribiendo las propiedades nuevas. Podemos hacer doble clic en los nombres de las reglas dentro del panel **[Estilos CSS]** para continuar con la ayuda de Dreamweaver.

RESUMEN

Es importante tener en cuenta que los estilos aplicados como hemos visto influyen de forma determinante en la etiqueta, como cuando aplicamos las propiedades a **[p]**. Por otro lado, el hecho de aplicar las propiedades a la etiqueta **[i]** en la forma **[p i]** sólo afectará a la etiqueta **[i]** que se encuentre dentro de **[p]** y no, por ejemplo, a la **[i]** que pueda estar dentro de **<h1>**.

TEXTO: PÁRRAFOS Y TÍTULOS

Los sitios web, cuyo escenario está compuesto por numerosos y variados elementos, fueron dueños desde sus orígenes de un actor principal que todos conocemos: el texto. Desde sus comienzos, las etiquetas HTML se enfocaron en su tratamiento, intentando imitar el uso del contexto editorial.

Como ya sabemos, en sus primeras versiones las páginas sólo estaban conformadas por algunos títulos y unos pocos párrafos, dentro de los cuales se marcaban los hipervínculos y algunos contenidos que se querían resaltar. Las primeras especificaciones oficiales del lenguaje HTML definidas por el W3C nacieron para ayudar a los desarrolladores a estructurar y marcar los textos mediante el uso de etiquetas específicas, pensadas para asignarle mayor jerarquía y un significado lógico a cada uno de sus elementos. Esta práctica, conocida en el ámbito de los desarrolladores web como **marcado semántico del contenido**, también resulta útil, entre otras razones, para que los motores de búsqueda puedan rastrear dentro de las interfaces, **entendiendo** su contenido para así indexarlo. Esto aumenta la importancia de conocer y familiarizarse con las etiquetas de texto, con el objetivo de aplicarlas de manera correcta cada vez que maquetamos una página. El lenguaje HTML diferencia dos grupos de etiquetas para el tratamiento del texto. Por un lado, existen aquellas que se utilizan para **estructurar** los contenidos en secciones y párrafos; por el otro, las que se usan para **marcar** elementos importantes dentro de ellos. A continuación, nos enfocaremos en las primeras y, más adelante, conoceremos las que se emplean para el marcado.

TEXTO ORIGINAL (sin estructurar)

El título define el contenido de una sección. Los subtítulos suelen subdividir los contenidos de una sección. Este es el primer párrafo de mi página. Cada párrafo está compuesto por una o más líneas de texto. Este texto corresponde al segundo párrafo de la página que estoy estructurando haciendo uso de etiquetas HTML.

TEXTO ESTRUCTURADO

El título define el contenido de una sección

→ TÍTULO

Los subtítulos suelen subdividir los contenidos de una sección

→ SUBTÍTULO

Este es el primer párrafo de mi página. Cada párrafo está compuesto por una o más líneas de texto. Este texto corresponde al segundo párrafo de la página que estoy estructurando haciendo uso de etiquetas HTML.

→ PÁRRAFOS



En este texto, podemos identificar tres elementos principales: un título, un subtítulo y dos párrafos.

PRIMEROS PASOS: ESTRUCTURACIÓN DEL TEXTO

Para estructurar correctamente los contenidos, sugerimos comenzar por analizar el texto, identificando y diferenciando los elementos que lo componen. Títulos, subtítulos y párrafos son los que vamos a etiquetar en primer término. Como referencia, podemos basarnos en el tratamiento que se le da al texto en cualquier ámbito editorial; por ejemplo, el capítulo de un libro o la noticia de un diario. Veamos ahora cuáles son las etiquetas adecuadas para definir esta estructura.

A TRAVÉS DE CSS, PODEMOS SOBREScribir LAS PROPIEDADES QUE ALGUNAS ETIQUETAS TIENEN POR DEFECTO, COMO LOS MÁRGENES.

PÁRRAFOS

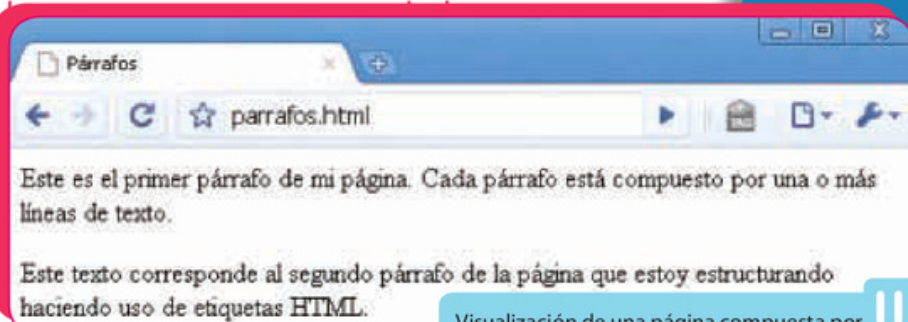
Una de las etiquetas más utilizadas en la maquetación de páginas es **<p>**, cuya función es delimitar los **párrafos** pertenecientes a un texto. Para aplicarla –al igual que la mayoría de las etiquetas–, debemos encerrar con ella el texto correspondiente a cada una de esas secciones, como se muestra en el **Código 1**.

CÓDIGO 1

```
<p>Este es el primer párrafo de mi página. Cada párrafo está compuesto por una o más líneas de texto.</p>
<p>Este texto corresponde al segundo párrafo de la página que estoy estructurando haciendo uso de etiquetas HTML.</p>
```

HTML

Como ya vimos, el estándar HTML clasifica los elementos en dos grandes grupos, que se diferencian, principalmente, por el espacio que ocupan en la pantalla de cualquier navegador. Los elementos en bloque son aquellos que comienzan en una nueva línea y llegan hasta el final del elemento que los contiene (por ejemplo, una etiqueta), aunque su contenido no necesariamente ocupe todo el espacio disponible. Las etiquetas **<h1>** y **<p>** son ejemplos de elementos en bloque. Los elementos en línea, al contrario de los en bloque, se visualizan a continuación del elemento declarado con anterioridad en el código HTML, en vez de comenzar en una nueva línea, y sólo ocupan el espacio necesario para ser visualizados. Las etiquetas **<a href>** y **** son ejemplos de este tipo de elementos.



Como la etiqueta **<p>** es un elemento en **bloque**, cada párrafo comienza en una nueva línea, y deja, por defecto, un salto de línea al finalizar, lo que ayuda a separar visualmente los párrafos entre sí.

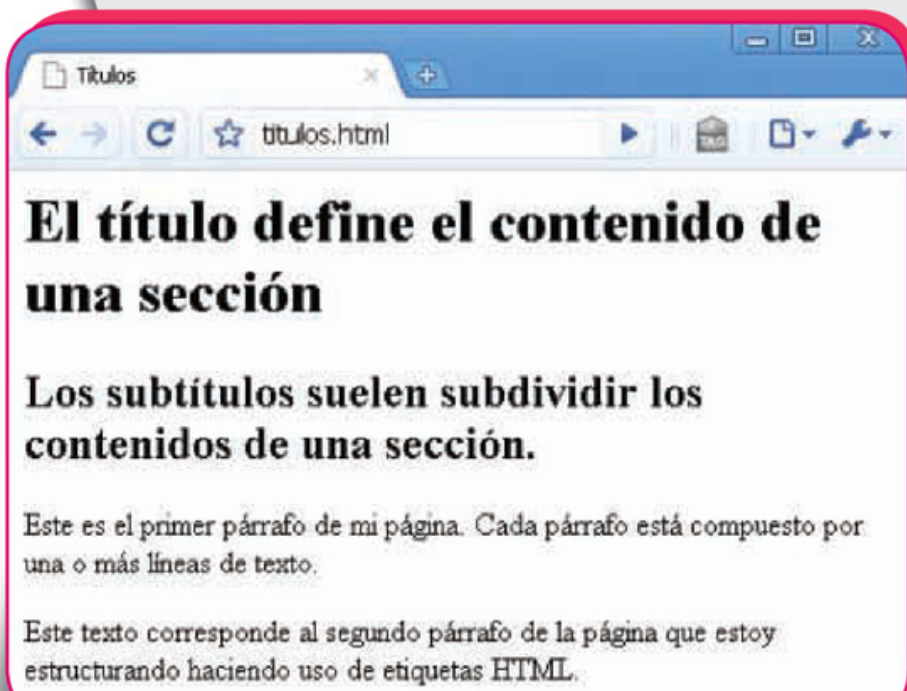
TÍTULOS

Como habitualmente las páginas están definidas por secciones más complejas que los párrafos, surge la necesidad de incorporar títulos y subtítulos que delimiten y diferencien estas secciones. El estándar HTML define seis etiquetas que se asignan según el nivel de importancia conceptual que cada título tiene dentro del contexto en el que se encuentra. En orden jerárquico descendente, estas etiquetas son **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>** y **<h6>**.

La etiqueta **<h1>** se utiliza para definir el título más importante de la página, en tanto que se usa **<h6>** para el de menor jerarquía. Esto no significa que todas las jerarquías deban usarse en la maquetación de un sitio web, porque esto dependerá de cuán rico sea el contenido y, en consecuencia, de su estructura. En páginas comunes, suele ser suficiente el uso de tres o cuatro niveles de títulos. Si al ejemplo anterior le añadimos dos títulos con sus correspondientes etiquetas, obtenemos la sintaxis del **Código 2**.

CÓDIGO 2

```
<h1>El título define el contenido de una sección</h1>
<h2>Los subtítulos suelen subdividir los contenidos de una sección.</h2>
<p>Este es el primer párrafo de mi página. Cada párrafo está compuesto por una o más líneas de texto.</p>
<p>Este texto corresponde al segundo párrafo de la página que estoy estructurando haciendo uso de etiquetas HTML.</p>
```



Los títulos, al igual que los párrafos, son elementos en bloque. Debido a esto, siempre comienzan en una nueva línea y se separan visualmente de los elementos que se disponen en la siguiente imagen. ▶

Para comprender exactamente de qué estamos hablando, podemos comparar el código escrito en un editor HTML con su visualización en un navegador. El código escrito sería el siguiente:

Para forzar a que un determinado texto salte a la siguiente línea, debemos utilizar la etiqueta `<br>` en el lugar del código donde queremos que se produzca el salto. Si bien podemos utilizarla numerosas veces consecutivas para conseguir la separación deseada, es conveniente aplicarla una sola vez (para que el salto se produzca) y, luego, complementar el trabajo con el uso de propiedades CSS hasta obtener el espaciado necesario. Esto se debe a que el uso desmedido de esta etiqueta hará más pesada la página, y el resultado será un código HTML desprolijo.

A diferencia de la mayoría de las etiquetas HTML, `<br>` (al igual que `<img>`) es una **etiqueta vacía**. Esto quiere decir que nunca encierra un determinado contenido, por lo que debe abrirse y cerrarse de la siguiente manera: `<br><br />`. Esta sintaxis se puede resumir aplicando un atajo, de la siguiente forma: `<br />`

Visualización de un texto con espacios en blanco y saltos de línea, con el formato de texto aplicado correctamente.

MARCADO BÁSICO Y CARACTERES ESPECIALES

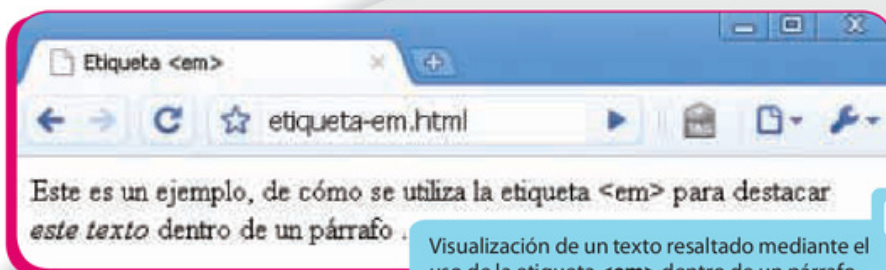
Ya hemos aprendido a estructurar un texto, identificando y etiquetando títulos, subtítulos y párrafos. Veremos ahora cómo marcar sólo algunos elementos de él, mediante la aplicación de nuevas etiquetas.

Con frecuencia, los textos contienen palabras o frases destacadas en negrita o cursiva, o incluyen citas textuales de otros contextos. La mayoría de las etiquetas de marcado que utilizaremos para crear esos estilos son elementos **en línea**, para que no se produzca un salto de página, aunque existen algunas excepciones. Comencemos a analizar cuáles son.

ETIQUETA `<EM>`

Es un elemento en línea que se utiliza para reforzar la importancia que tiene un determinado texto dentro de un párrafo. Como resultado, los navegadores muestran los textos encerrados por esta etiqueta de manera **cursiva**, como vemos en el siguiente ejemplo:

```
<p>Este es un ejemplo, de
cómo se utiliza la etiqueta
<em>este texto</em> dentro
de un párrafo .</p>
```



ETIQUETA `<STRONG>`

Conceptualmente, esta etiqueta es similar a `<em>`, pero se diferencia de ella en que ésta se utiliza para remarcar el texto más importante del párrafo. También es un elemento en línea, y su visualización en los navegadores es como la negrita. Su uso es como vemos a continuación:

```
<p>Este es un ejemplo, de
cómo se utiliza la etiqueta
<strong>que este texto es
el más importante</strong>
del párrafo.</p>
```



ETIQUETAS `<B>` Y `<STRONG>`

Si bien aplicando las etiquetas `<b>` y `<strong>` obtenemos idéntico resultado visual, la diferencia más importante entre ellas es conceptual. La función de `<b>` es meramente visual, al indicar que el texto debe verse en negrita; mientras que `<strong>` señala que el texto abarcado por ella es el más importante dentro de un párrafo, algo muy valioso para los motores de búsqueda (SEO). Por convención, los navegadores han decidido que esta última etiqueta se visualice también como negrita, lo que no significa que pueda aparecer en algún momento un nuevo navegador que la muestre con otro estilo.

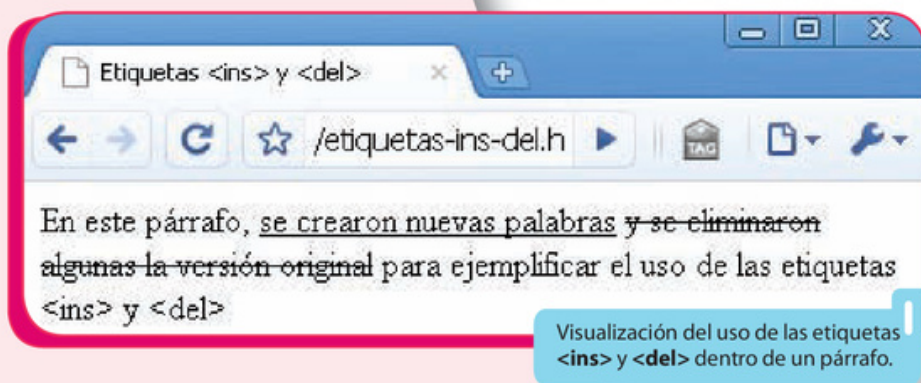
Visualización de un texto destacado mediante el uso de la etiqueta `<strong>` dentro de un párrafo.

ETIQUETAS **<ins>** Y ****

El uso de estas etiquetas no es demasiado frecuente. Se aplican para remarcar, dentro de un contenido, aquellas modificaciones que se han realizado respecto de su versión original. Concretamente, **<ins>** indica qué textos fueron insertados (se ven con un subrayado) y ****, cuáles fueron eliminados (se presentan de manera tachada). Ambas etiquetas son elementos en línea, y su aplicación puede complementarse mediante el uso del atributo **Datetime**, útil para adicionar información sobre la fecha y la hora en las que se realizaron los cambios. Vemos un ejemplo de la aplicación de estas etiquetas en el **Código 3**.

CÓDIGO 3

```
<p>En este párrafo, <ins datetime="20091026" cite="http://www.librosweb.es/mas_informacion.html"> se crearon nuevas palabras</ins> <del datetime="20091025" cite="http://www.librosweb.es/mas_informacion.html"> y se eliminaron algunas la versión original</del> para ejemplificar el uso de las etiquetas <ins> y <del></p>
```



Visualización del uso de las etiquetas **<ins>** y **** dentro de un párrafo.

ETIQUETAS **<blockquote>**

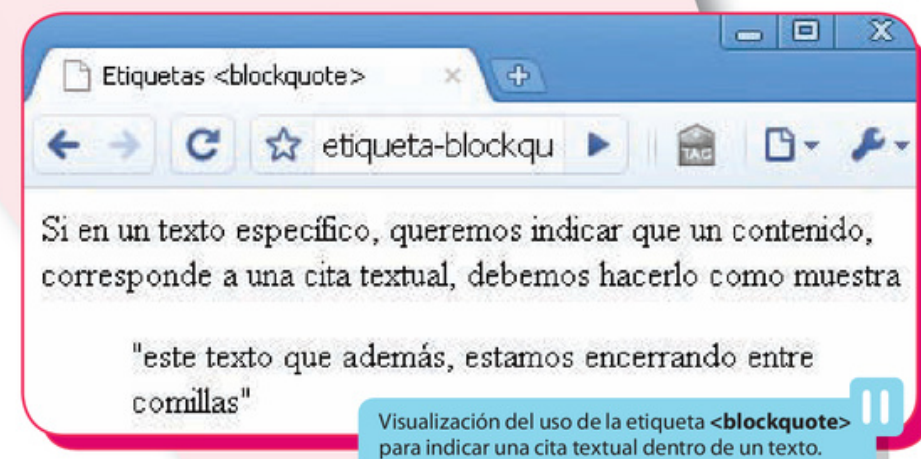
A diferencia de las anteriores, éste es un elemento en bloque, utilizado para marcar las **citas textuales** que fueron extraídas de otros contextos. Se complementa con el atributo **cite**, que permite indicar la URL de la que se obtuvo la cita textual. Veamos un ejemplo del uso de esta etiqueta en el **Código 4**.

CODIFICACIÓN DE CARACTERES

Además de utilizar el marcado, al escribir un texto en un documento HTML, tenemos que ingresar algunos caracteres de manera especial, ya que no son reconocidos por los navegadores de la misma forma en que lo haría un editor de texto. Las principales razones por las que esto sucede es que algunos de estos caracteres son usados para escribir las etiquetas HTML, de modo que los navegadores los interpretan como código en vez de tomarlos como texto normal. Por otro lado, aquellos caracteres que están por fuera del alfabeto inglés también pueden ocasionar problemas en su visualización; por ejemplo, ñ, á, ç, ï, j, etcétera. Para evitar el primero de los dos inconvenientes, al escribir determinados caracteres, debemos reemplazarlos por la entidad correspondiente del siguiente cuadro:

CÓDIGO 4

```
<p>Si en un texto específico, queremos indicar que un contenido, corresponde a una cita textual, debemos hacerlo como muestra</p>
<blockquote cite="http://www.w3.org/TR/html401/struct/text.html">"este texto que además, estamos encerrando entre comillas"</blockquote>
```



Visualización del uso de la etiqueta **<blockquote>** para indicar una cita textual dentro de un texto.



ENTIDADES PARA REPRESENTAR SIGNOS

Algunos de estos caracteres son reconocidos como código por los navegadores.

ENTIDAD	CARÁCTER	DESCRIPCIÓN	TRADUCCIÓN
<	<	less than	signo de menor que
>	>	more than	signo de mayor que
&	&	ampersand	ampersand
"	"	quotation mark	comillas
 	(espacio en blanco)	non-breaking space	espacio en blanco
'	'	apostrophe	apóstrofo

Por ejemplo, si queremos escribir correctamente el texto "Si escribimos en lenguaje html, los paréntesis angulares <, > tal cual son, no son visualizados correctamente en los navegadores", debemos hacerlo como se muestra a continuación, para que se vea como corresponde:

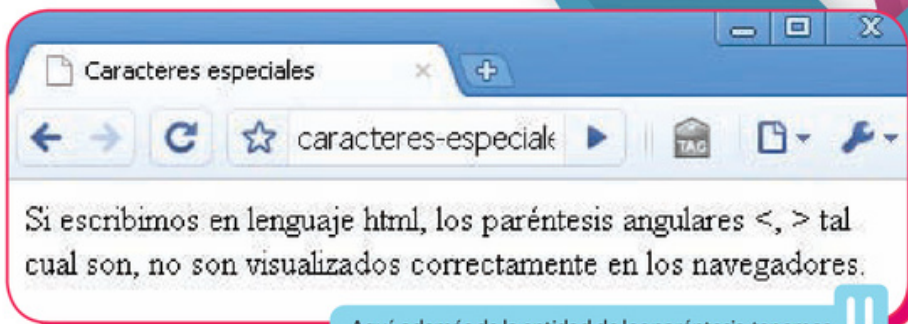
<p> Si escribimos en lenguaje html, los paréntesis angulares <., > tal cual son, no son visualizados correctamente en los navegadores</p>



Entidades HTML y Charset

Las entidades HTML están compuestas por una secuencia de caracteres que devuelve un determinado carácter. Un charset es una norma que define cómo se verán ciertos caracteres en toda la página HTML. Para definir el **charset** que queremos usar en XHTML, debemos especificarlo como atributo de una etiqueta meta:

```
<meta content="text/html; charset=[b]iso-8859-1[/b]" http-equiv="Content-Type"/>
```



Aquí, además de la entidad de los paréntesis, tenemos que utilizar la correspondiente a la "e" acentuada.

El segundo problema es un poco más complejo de resolver, ya que deviene del proceso de creación de un sitio web en el que intervienen diferentes responsables. Si no se acuerda en el tipo de codificación que se va a utilizar (normalmente, llamado **charset**), pueden intervenir codificaciones distintas que generarán inconvenientes en la visualización de los caracteres.

La solución acertada consiste en emplear el mismo **charset** en todos los procesos involucrados, por ejemplo, **UTF-8** o **ISO-8859**. Cabe mencionar que la codificación que más se adecua a los hispanohablantes es la ISO-8859-1, y que UTF-8 es el charset por defecto en cada estándar del W3C.

Como no siempre es posible utilizar el mismo charset, para asegurar una correcta visualización, dentro del texto podemos optar por reemplazar todos los caracteres por su entidad correspondiente.

Por lo tanto, para escribir el texto "Es probable que existan problemas de visualización, cuando existen caracteres acentuados dentro de un párrafo.", deberíamos reemplazar los caracteres especiales de la siguiente forma:

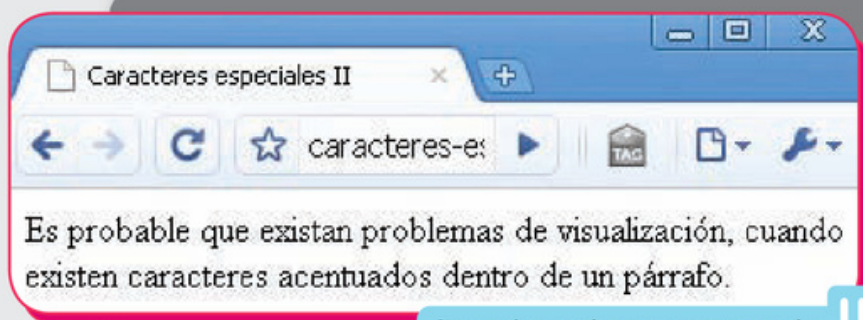
<p> Es probable que existan problemas de visualizaciön, cuando existen caracteres acentuados dentro de un párrafo. </p>



ENTIDADES PARA REEMPLAZAR CARACTERES

Representación correcta de los caracteres especiales o acentuados.

ENTIDAD	CARÁCTER	DESCRIPCIÓN OFICIAL
ñ	ñ	latin letter n with tilde
Ñ	Ñ	latin capital n letter with tilde
á	á	a acute
é	é	e acute
í	í	i acute
ó	ó	o acute
ú	ú	u acute
Á	Á	A acute
É	É	E acute
Í	Í	I acute
Ó	Ó	O acute
Ú	Ú	U acute
€	€	Euro



Si reemplazamos los caracteres acentuados, el texto se visualiza correctamente.

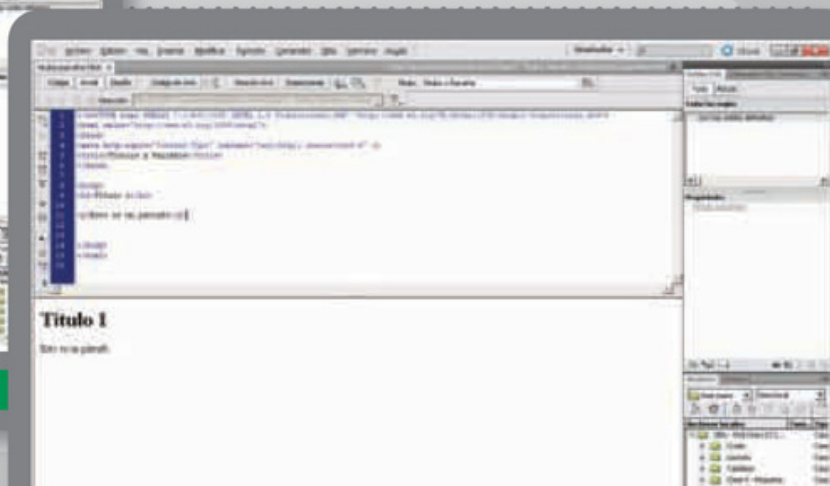
ESCRIBIR TÍTULOS

PASO A PASO

En este apartado, veremos las diferentes formas de escribir títulos y párrafos en Dreamweaver utilizando las etiquetas especiales para este tipo de elementos.



01 | TÍTULO



PÁRRAFO | **02**

Luego de haber conocido las etiquetas para marcar el texto, en estas páginas aplicaremos aquellas relacionadas con títulos y párrafos. Veremos una forma muy fácil y rápida de aplicarlas, y para eso, crearemos nuestros elementos paso a paso, de modo que el procedimiento sea bien claro. ◀

PASO | 01 Como ya vimos, existen etiquetas especiales para escribir un título (**<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>** y **<h6>**). Todas ellas tienen propiedades asignadas por defecto, y la elección de la adecuada estará determinada por el nivel de importancia del título. Escribimos la etiqueta que vamos a usar, el texto del título y, luego, cerramos la etiqueta con el formato **</h1>**.

PASO | 02 Un párrafo se escribe con la etiqueta **<p>**, que también posee propiedades predefinidas. Si queremos tener más de un párrafo, después de cerrar esa etiqueta, podemos abrir otra igual, ya que no hay un límite de párrafos en ellas. Escribimos el texto a continuación de la etiqueta de apertura y cerramos con **</p>**.

Y PÁRRAFOS

04 | PROPIEDADES ADICIONALES

05 | TEXTO EN NEGRITA

03 | SALTO DE LÍNEA

06 | MANTENER EL ORDEN

PASO | 03 Si queremos generar un salto de línea dentro de un mismo párrafo, debemos utilizar la etiqueta **br**. Existen varias formas de escribirla, y la correcta es **
**. Si ponemos **
** o **
**, la etiqueta puede llegar a funcionar, pero estará mal escrita.

PASO | 04 Dentro de los párrafos o títulos, podemos colocar diferentes etiquetas que nos permitirán aplicar propiedades adicionales. Por ejemplo, **i** convertirá su contenido en *itálica*. Como todas las etiquetas, ésta posee propiedades por defecto y debe cerrarse al finalizar; es decir, se escribe **<i></i>**, con el texto que quedará resaltado entre ambos tags.

PASO | 05 Otra etiqueta que podemos utilizar dentro de los títulos o párrafos es **b**, que convertirá nuestro texto en **bold** (también llamado **negrita**). Este tag funciona tal como el anterior: el texto que será resaltado se escribe entre la apertura **** y el cierre ****.

PASO | 06 Al terminar de escribir las etiquetas y su contenido, debemos ir a la barra de programación, buscar **[Formatear código fuente]** y desplegar sus opciones para, luego, hacer clic en **[Aplicar formato de origen]**. De esta manera, el código se ubicará de forma más legible.

RESUMEN

Recordemos que todas las etiquetas tienen que cerrarse y ser escritas en minúsculas. Además, como hemos visto con la etiqueta **
**, no todas llevan contenido. Si bien son pocas las de este último tipo, cuando pongamos manos a la obra, nos encontraremos con algunas más.

ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 ¿Qué propiedad define la Font-Family en **CSS**?

02 ¿Qué tamaños asignan, por defecto, los navegadores a las etiquetas de títulos?

03 ¿Qué son los saltos de página y cómo se indican en **HTML**?

04 ¿Cuál es el resultado en el navegador de una etiqueta ****?

05 ¿Qué diferencia a una etiqueta **** de una ****?

06 ¿Cuál es la diferencia entre las etiquetas **<ins>** y ****?

07 ¿Qué es un **charset**?

08 ¿Para qué sirven las etiquetas **<blockquote>**?

09 ¿Cómo podemos forzar los espacios entre las palabras?

10 ¿Cuál unidad de medida recomienda utilizar **CSS** para medir el tamaño del texto?



PRÁCTICAS

01 Cree una hoja de estilo de CSS en Dreamweaver y defina el color, la familia, el tamaño y la decoración de la letra.

02 En su archivo de texto, divida el contenido en título, subtítulo y párrafos. En base a eso, asigne las etiquetas de jerarquía a su texto en Dreamweaver.

03 Aplique la etiqueta **<blockquote>** a su texto.

04 Indique el interlineado de un texto y la alineación por medio de CSS.

05 Modifique la hoja de estilo creada anteriormente y agregue un color de fondo de color para la página.



Enlaces



- ▶ HIPERTEXTOS
- ▶ ENLACES RELATIVOS Y ABSOLUTOS
- ▶ ENLACES EN DREAMWEAVER
- ▶ CSS: CLASES ESPECIALES
- ▶ ENLACES BÁSICOS Y AVANZADOS
- ▶ TIPOLOGÍA DE LOS SITIOS WEB
- ▶ ESTILO PARA LOS ENLACES EN DREAMWEAVER



HIPERTEXTOS

Éste es uno de los conceptos más fuertes de la Web. El término “navegación” no hubiera sido posible sin la existencia de los enlaces, que permiten conectar entre sí la información disponible en ella.

El término **hipertexto** se refiere a un tipo de texto electrónico que representa, además de una nueva tecnología informática, una novedosa manera de edición. Gran parte del éxito del lenguaje HTML se debe a la posibilidad de **conectar** la información verbal y no verbal (visual, sonora, animación, etcétera) a través del uso de **enlaces** o **hipertextos** que, como nodos conformadores de una gran red, ponen a disposición de los usuarios recorridos alternativos no secuenciales. Lo más novedoso de este concepto es, precisamente, la libertad que se le da al usuario de moverse por la Web, siguiendo sus propios intereses y eligiendo qué información consumir o qué acción ejecutar como parte de un trayecto alternativo entre muchos otros.

+ ORIGEN HTML

En la definición del lenguaje HTML, se tomaron algunas características que ya existían para la publicación digital de contenidos, entre ellas, el hipertexto. Es por eso que las siglas HTML provienen de las palabras *HyperText Markup Language*, frase que podría traducirse como “lenguaje de marcado para hipertexto”.

ORIGEN DEL CONCEPTO DE HIPERTEXTO

El origen de la idea de **hipertexto** se remonta al año 1945, cuando **Vannevar Bush**, Jefe del Departamento de Investigación y Desarrollo Científico de

EE.UU., publicó el artículo “As we may think”, en el que criticaba los métodos utilizados hasta ese entonces para gestionar la información. Allí detalla que la estructura secuencial de los documentos –influenciada por la estructura verbal del discurso– es la causante de que los métodos de la época fueran incapaces de procesar grandes cantidades de información de manera correcta. La razón es simple: la mente humana no procesa naturalmente la información de modo secuencial, sino que lo hace a través de **asociaciones**. Si bien este autor nunca llegó a utilizar el término hipertexto, fue el primero en tener noción de lo que era la **multisecuencialidad** (no linealidad) de este elemento.

ENLACES BÁSICOS

Técnicamente, los enlaces son bloques compuestos por fragmentos de textos u otros elementos que vinculan dos recursos. Estos recursos suelen ser páginas web, aunque también se emplean enlaces para **relacionar** imágenes, para **descargar** documentos (PDF, Word, etc.) o para **ejecutar** acciones, como enviar un e-mail. La sintaxis de un enlace está compuesta por la etiqueta `<A>...</A>`, dentro de la cual siempre debe declararse el atributo **href** o **name** para indicar el destino del enlace.

ATRIBUTO HREF

Es el más importante para esta etiqueta y se usa para **enlazar** diferentes URL. Veamos cómo sería la sintaxis correcta de un enlace simple:

```
<A HREF="URL">Texto
del enlace</A>
```

Partes de una URL

<http://es.wikipedia.org/wiki/Url>

Protocolo

Dominio

Directorio

Tres partes de la URL le indican al navegador cómo llegar a la página solicitada.



En este caso, **Texto del enlace** es el contenido que se visualizará en el navegador, y **"URL"** es el destino al que apuntará el link.

ATRIBUTO NAME

Este atributo se implementa para nombrar secciones dentro de una misma página web, y suele ser útil para contenidos extensos, en los que el usuario podrá ir de una parte a otra sin necesidad de hacer scroll. Para aprender a utilizarlo, en primera instancia, crearemos un enlace de destino con el atributo **name**, asignándole un valor al azar, que nos servirá como identificador para el siguiente paso. Posteriormente, desde el enlace de origen, haremos referencia a este enlace anteponiendo el símbolo **#** al valor que le hayamos dado a **name**. Por ejemplo:

Enlace de destino:

`<A NAME="parte1">`

Primera parte`</A>`

Enlace de origen:

`<A HREF="#parte1">Ir a la primera parte</A>`

De esta manera, al hacer clic sobre el enlace de origen, el navegador se posicionará sobre el de destino.

URL (LOCALIZADOR UNIFORME DE RECURSOS)

Para comenzar a crear enlaces un poco más complejos, es necesario familiarizarnos con el concepto de **URL** (del inglés, *Uniform Resource Locator*), que, además de ser esencial para crear enlaces, también suele aplicarse a otros elementos. Las URL determinan que cada página HTML publicada en Internet posea un nombre único que la distinga del resto. Esto es lo que permite que los enlaces apunten inequívocamente a una determinada página. Por ejemplo, si queremos acceder a la página de Wikipedia, debemos escribir en el navegador la siguiente cadena de texto: **www.wikipedia.org**,

El artículo original "As we may think" se encuentra disponible en los archivos de *The Atlantic* (www.theatlantic.com).

que es la URL completa de la página principal de su sitio. Las URL poseen una estructura destinada a que los navegadores encuentren los recursos disponibles de manera eficiente. La mayoría de las que son simples están conformadas por tres partes: protocolo, dominio y directorio; aunque existen otras más complejas:

- **Protocolo:** es el mecanismo que usan los navegadores para acceder a los recursos. Todos los sitios web utilizan **http://**, y a aquellos que requieren seguridad se les agrega una letra "s" al final (**https://**).
- **Dominio:** especifica el nombre de la computadora (por lo general, es un nombre de dominio o una dirección IP) donde está alojado el contenido que compone un sitio web.
- **Directorio:** luego del dominio, se encuentra el o los directorios separados por barras ("/"), que definen el recorrido o ruta por seguir para llegar hasta el recurso. ◀

LA MENTE HUMANA
NO PROCESA NATURALMENTE
LA INFORMACIÓN
DE MANERA SECUENCIAL,
SINO A TRAVÉS DE
ASOCIACIONES.

Visualización de un enlace sin estilos (por defecto).



Enlaces - Etiquetas

ahref.html

Texto del enlace

ENLACES RELATIVOS Y ABSOLUTOS



Aprenderemos a interconectar diferentes tipos de elementos, que pueden pertenecer a nuestro propio sitio o estar en otro lado de la Web. Analicémoslo aquí.

Todo sitio web posee, comúnmente, numerosos enlaces. Éstos pueden apuntar a otras páginas o recursos dentro del mismo sitio o alojados en sitios externos. Esto categoriza a los enlaces en dos grandes grupos:

- **Internos:** son aquellos que apuntan a recursos o páginas ubicados dentro del mismo sitio.
- **Externos:** son los que enlazan un recurso o una página ubicada por fuera del sitio de origen. Una característica clave de este tipo de enlaces es que, al hacer clic sobre ellos, el navegador abandona el sitio en el que se encuentra para localizar el nuevo sitio.

Por otro lado, existe una clasificación que, además de tener en cuenta si los enlaces son externos o internos, tiene que ver con la manera en la que se escribe la sintaxis de las URLs:

- **Absolutas:** son las que incluyen la URL completa, con protocolo, dominio

y directorios. Por lo general, se utilizan para los enlaces externos, ya que necesitan toda la información de la localización del recurso.

- **Relativas:** por lo general, este tipo de URL se utiliza al crear enlaces internos. Sólo indica los **directorios** y, a partir de esa información, los navegadores pueden adivinar los datos faltantes (**protocolo y dominio**) para localizar el recurso.

Por ejemplo, si desde la página ubicada en <http://www.wikipedia.org/ruta1/ruta2/pagina1.html> quisiéramos acceder a otra presente en <http://www.ejemplo.com/ruta1/ruta2/pagina2.html>, no necesitaríamos escribir la URL absoluta, ya que ambas páginas poseen el mismo dominio y utilizan el mismo protocolo. Por lo tanto, podemos prescindir de esa parte de la URL y escribirla de manera relativa. Entonces, las dos formas de escribir la misma URL serían:

URL absoluta: <http://www.ejemplo.com/ruta1/ruta2/pagina2.html>
URL relativa: [/ruta2/pagina2.html](#)

Y el enlace aplicado:

```
<A HREF="/ruta2/pagina2.html">Texto del enlace correspondiente a la página 1</A>
```

Una de las ventajas de utilizar enlaces relativos es que podemos cambiar la dirección de un sitio web sin tener que reescribir las URLs de todos sus hipervínculos. Existen varios tipos de URL relativas. Veamos cada uno de ellos analizando ejemplos de enlaces que nos ayudarán a comprender cómo deben crearse:

- Cuando el **origen** y el **destino** del enlace están en el **mismo directorio**, el protocolo, el dominio y los directorios son exactos. La única diferencia es el nombre del recurso enlazado, por lo que la URL relativa será solamente este nombre.

Origen: <http://www.misitio.com/ruta1/ruta2/pagina1.html>
Destino: <http://www.misitio.com/ruta1/ruta2/pagina2.html>
URL relativa: [pagina2.html](#)

- Cuando el destino del enlace está en un nivel superior que el de su origen, aunque ambas páginas están muy cerca, es necesario indicar esa superioridad de nivel al escribir la URL. La manera correcta de hacerlo es escribir el nombre



ASPECTOS DE LOS ENLACES

En los diferentes sitios que recorramos, podemos reconocer fácilmente los enlaces porque al pasar el cursor por encima de ellos, éste se transforma en una mano con un dedo señalador. Si los enlaces no tienen estilos que modifican su aspecto visual, por defecto se muestran como textos subrayados de azul. Al incorporar estilos en un diseño, es conveniente codificar los enlaces asignándoles un color particular no utilizado en textos que no sean hipertextos, o mantener el subrayado tradicional. Esto ayudará a los usuarios a identificar claramente cuáles son los enlaces dentro del contenido.

del recurso anteponiéndole dos puntos y una barra (../), que le indicarán al navegador que debe subir un nivel para localizar el recurso.

Origen: <http://www.misitio.com/ruta1/ruta2/pagina1.html>
Destino: <http://www.misitio.com/ruta1/pagina2.html>
URL relativa: [../pagina2.html](#)

Si el destino se encuentra dos o más niveles por encima del origen, debemos incluir ../ tantas veces como niveles existan. Por ejemplo, si los niveles fueran dos, la URL relativa se escribiría así:

Origen: <http://www.misitio.com/ruta1/ruta2/ruta3/pagina1.html>
Destino: <http://www.misitio.com/ruta1/pagina2.html>
URL relativa: [../../pagina2.html](#)

► Si el destino del enlace está en un **nivel inferior** al de su origen, debemos escribir la URL indicando el nombre del recurso anteponiéndole el nombre del directorio que lo contiene:

Origen: <http://www.misitio.com/ruta1/pagina1.html>
Destino: <http://www.misitio.com/ruta1/ruta2/pagina2.html>
URL relativa: [ruta2/pagina2.html](#)

En este caso, no es necesario utilizar barras o puntos, pero, al igual que en el caso anterior, debemos escribir

AL USAR ENLACES RELATIVOS, SI CAMBIA LA DIRECCIÓN DEL SITIO, NO TENDREMOS QUE RESCRIBIR TODOS LOS HIPERVÍNCULOS.

la cantidad de directorios de diferencia que haya entre ambos:

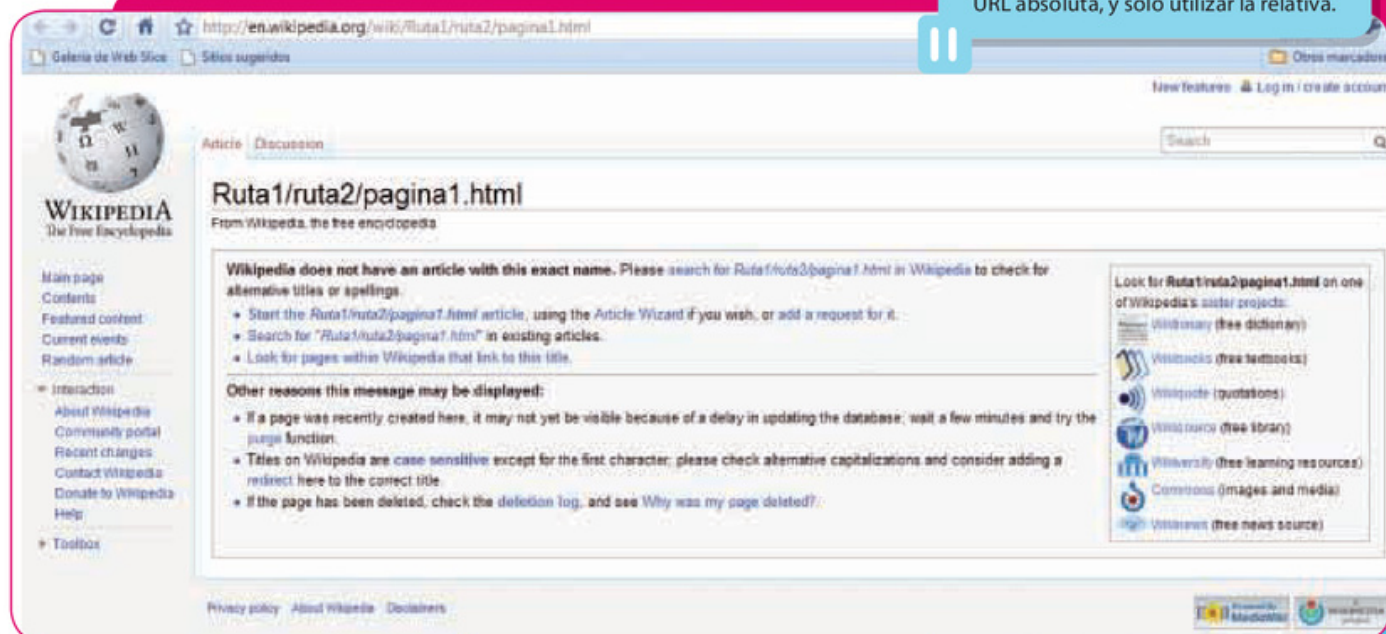
Origen: <http://www.misitio.com/ruta1/pagina1.html>
Destino: <http://www.misitio.com/ruta1/ruta2/ruta3/ruta4/pagina2.html>
URL relativa: [ruta2/ruta3/ruta4/ruta2/pagina2.html](#)

► Si el origen y el destino del enlace comparten el mismo

dominio pero se encuentran muy alejados, podríamos utilizar los dos métodos anteriores para simplificar la tarea. Sin embargo, lo conveniente es describir todos los directorios que existen a partir del servidor, anteponiendo una barra (/). De esta forma, los navegadores vuelven hasta la raíz del dominio y buscan el recurso a partir de allí, siguiendo el orden de los directorios que hemos especificado.

Origen: <http://www.misitio.com/ruta1/ruta2/ruta3/pagina1.html>
Destino: <http://www.misitio.com/ruta9/pagina2.html>
URL relativa: [/ruta9/pagina2.html](#) ◀

Como vimos, podemos prescindir de la URL absoluta, y sólo utilizar la relativa.



ENLACES EN

PASO A PASO

En este apartado veremos los distintos tipos de enlaces que se pueden generar a través de Dreamweaver. ¡Manos a la obra!

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Documento sin título</title>
6 </head>
7
8 <body>
9 <h1>Titulo para enlace interno</h1>
10
11
12 <p>Enlace interno del 1er título</p>
13
14
15
16
17
```

01 | PREPARAR EL DOCUMENTO

```
5 <title>Documento sin título</title>
6 </head>
7
8 <body>
9 <h1><a href="#">Titulo para enlace interno</a></h1>
10
11
12
13 <p>Enlace interno del 1er título<a href="#"></a></p>
14
15
16
17 </body>
```

ENLACE INTERNO | 02

```
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Documento sin título</title>
6 </head>
7
8 <body>
9 <h1><a href="#texto">Titulo para enlace interno</a></h1>
10
11
12
13 <p>Enlace interno del 1er título<a name="texto" id="texto"></a></p>
14
15
16
17
18 </body>
19 </html>
20
```

03 | CREAR EL ENLACE

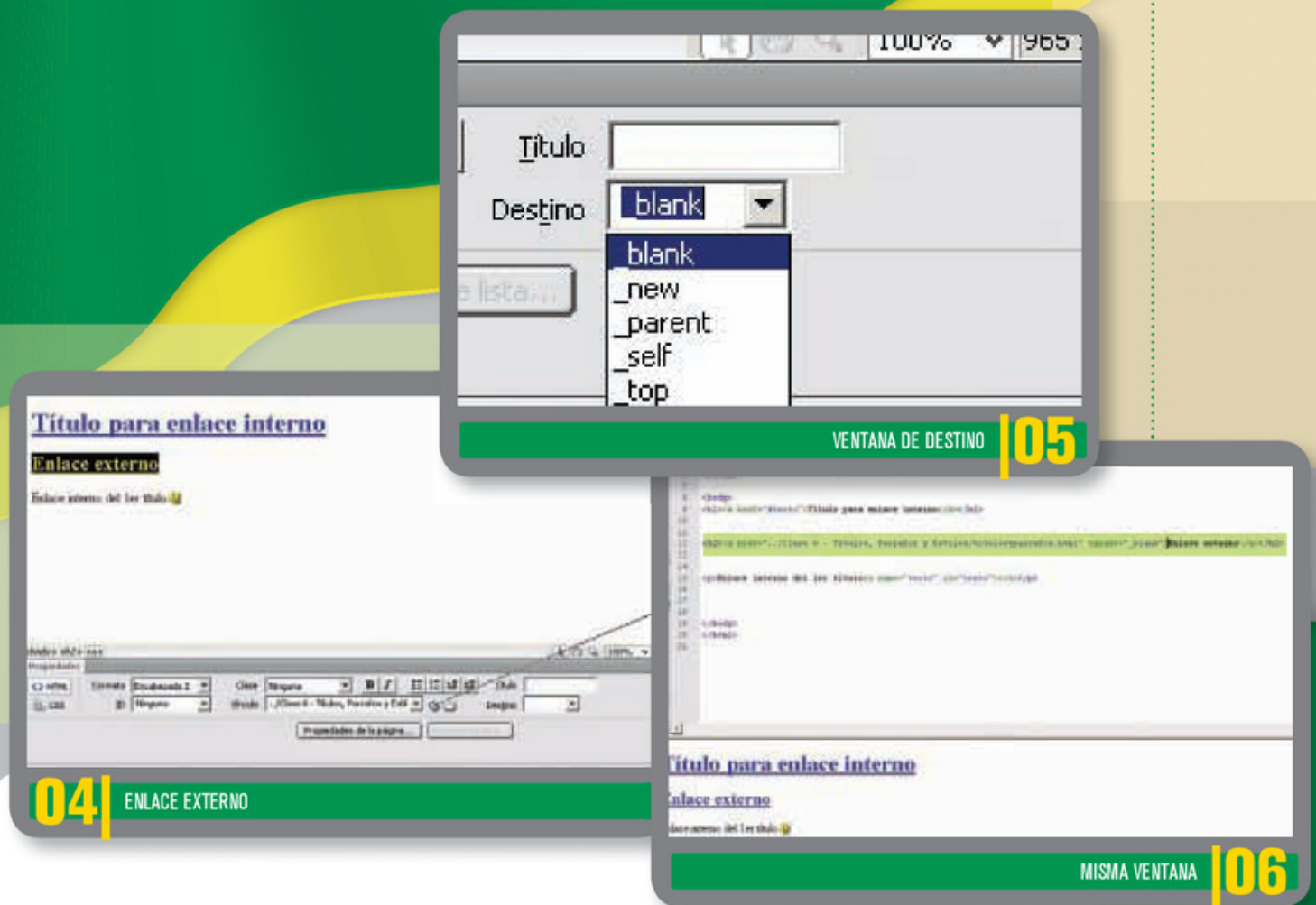
Adoce Dreamweaver brinda la posibilidad de generar enlaces desde el manejador gráfico ubicado en la barra [Propiedades] o desde el código. Seguramente, al principio nos sentiremos más cómodos con la primera opción, pero la segunda nos resultará ideal para escribir más rápido y sin recurrir al mouse. ◀

PASO | 01 Lo primero que haremos para preparar el documento en el que vamos a armar el enlace será crear un texto con la etiqueta `<h1>` y otro con `<p>`.

PASO | 02 Para generar el enlace, utilizamos la etiqueta `<a>`, seguida del parámetro **href**. Debemos tener en cuenta que, en un enlace interno, tanto `<h1>` como `<p>` deben contener una etiqueta `<a>`.

PASO | 03 A continuación, dentro del parámetro **href** de la etiqueta `<a>` correspondiente a `<h1>`, colocamos el signo # seguido del nombre **texto**. Luego, en la etiqueta `<a>` correspondiente a `<p>`, ponemos **texto** dentro del parámetro **name**. De esta forma, conseguiremos que la etiqueta `<h1>` enlace con `<p>` al hacer clic (tengamos en cuenta que en **name** la palabra **texto** no debe llevar #).

DREAMWEAVER



PASO | 04 Para generar un enlace externo al documento, vamos a usar una etiqueta **<h2>** y vincularemos su texto a otro archivo externo de carácter HTML. Entonces, seleccionamos el texto y utilizamos el manejador gráfico que se encuentra en la barra **[Propiedades]**, moviendo el mouse hasta el archivo de destino del enlace.

PASO | 05 Desde Dreamweaver, podemos indicarle al enlace externo el lugar que tendrá al abrirse. Si bien por defecto la etiqueta **<a>** reemplazará el archivo HTML que estamos viendo por el nuevo, mediante las opciones del desplegable **[Destino]** podemos indicar que éste se muestre, por ejemplo, en una pestaña nueva del navegador si el vínculo no pertenece a nuestro sitio. Como vemos, la propiedad target indicará en qué ventana debe abrirse el enlace. Este recurso sirve también para crear pop-ups.

PASO | 06 En este caso, dejaremos la etiqueta **<a>** sin ningún destino, porque si bien el enlace es externo, se encuentra dentro de nuestro propio sitio, y lo más conveniente es que una página (archivo HTML) reemplace a la otra. No sucede lo mismo cuando el enlace generado debe enlazar otro tipo de archivo, como podrían ser PDF, RAR o ZIP, entre otros, dado que lo ideal es que el usuario no pierda de vista nuestra web.

RESUMEN

Cuando creamos los archivos de nuestros sitios, siempre debemos recordar que sus nombres no pueden tener espacios ni caracteres especiales, porque si no, pueden ser tomados como un error dentro de los estándares. Además, es más seguro trabajar de esta forma.

CSS: CLASES ESPECIALES

CSS posee reglas especiales que permiten aplicar estilos a elementos que el lenguaje HTML no reconoce como independientes, y a los diferentes estados de un elemento.

Existen clases especiales de CSS, llamadas **pseudo-clases**, que clasifican a los elementos según su estado y nos permiten otorgar estilos a cada una de sus instancias.

Si quisiéramos asignar distintos estilos a los diferentes estados de un mismo elemento, sería imposible hacerlo a través del uso de los atributos **[id]** o **[class]**, y por eso CSS incorpora un nuevo concepto, el de **pseudo-clases**.

Como hemos visto anteriormente, los navegadores visualizan los enlaces no visitados de color azul y subrayados, mientras que los visitados son de color violeta. Aplicando pseudo-clases, es posible modificar por completo el aspecto de los enlaces en cada una de las instancias que se desprenden del comportamiento de los usuarios (al hacer clic, al posicionar el cursor sobre el elemento, y otras). La sintaxis correcta de una pseudo-clase se construye de la siguiente manera:

```
objeto:pseudoclase {
  propiedad: valor;
}
```

En este caso, los dos puntos (:) siempre se ubican entre el objeto y la pseudo-clase, sin dejar espacios. Estas clases especiales se aplican, principalmente, sobre la etiqueta de los enlaces (<A>), y existen cuatro tipos:

► **:link** se emplea para definir la apariencia de los enlaces que aún no han sido visitados por los usuarios.

► **:visited** define el aspecto de aquellos enlaces que ya fueron visitados por los usuarios. Es útil como una referencia que permite reconocer a simple vista los lugares que ya hemos recorrido dentro del sitio.

► **:hover** se usa para asignar determinado aspecto a los enlaces cuando el usuario posiciona el cursor sobre ellos (sin hacer clic).

► **:active** declara el aspecto del enlace cuando el usuario hace clic con el cursor sobre él, por lo que suele durar un lapso muy breve de tiempo.

El siguiente ejemplo muestra cómo se define el estilo para los diferentes estados de un enlace a través del uso de pseudo-clases:

```
A {
  text-decoration: none;
}
A:hover {
  text-decoration: underline;
}
A:visited {
  text-decoration: none;
  color: green;
}
```

Por otro lado, también podemos combinar las pseudo-clases con cualquier otro selector. Veamos algunos ejemplos. En el siguiente caso, los enlaces cuya clase sea **[sidebar]** serán visualizados de color rojo cuando hayan sido visitados:

```
a.sidebar:visited {
  color: red;
}
```

Como veremos en el ejemplo a continuación, los enlaces cuyo id sea **[caja]** serán visualizados con un subrayado al posicionar el cursor sobre ellos:

```
#caja:hover { text-decoration: underline; }
```

Con el siguiente código, los enlaces que contengan la clase **[destacado]** y que, además, se encuentren dentro de un listado contenido por el elemento div con id **[menú]**, serán visualizados con un fondo de color gris al hacer clic con el cursor sobre ellos:

Pseudo-elementos

Este es un ejemplo de uso de pseudo-elementos.

La primera letra de los párrafos tiene definido un estilo distinto al resto.

Podemos ver cómo se muestra el pseudo-elemento **:first-letter** en un navegador.

```
div#menu ul li
a.destacado:active { back-
ground-color: #c2c2c2; }
```

Cuando definimos varias pseudo-clases sobre un mismo enlace, pueden ocasionarse coaliciones de estilos. Para que esto no ocurra, el orden sugerido al escribir estos elementos en la hoja de estilos es **:link**, **:visited**, **:hover** y **:active**.

PSEUDO-ELEMENTOS

Los pseudo-elementos son reglas CSS utilizadas para identificar y dar estilo a aquellos elementos de una página a los que el lenguaje HTML no les otorga identidad propia.

Como son generados por las reglas de estilo, estos elementos no existen en el código fuente de la página. Su referencia está en los archivos CSS y, concretamente, son utilizados para aplicar estilos a determinadas partes de un texto, como podrían ser la primera frase y la primera letra. Su sintaxis es igual a la de las pseudo-clases; veamos:

```
objeto:pseudoelemento {
propiedad: valor;
}
```

Los pseudo-elementos disponibles a partir de CSS2 son cuatro:

- ▶ **:first-letter** permite darle estilos a la primera letra de un texto.
- ▶ **:first-line** permite darle estilos a la primera línea de un texto. La extensión de esta línea depende del espacio reservado en el elemento que lo contiene o del tamaño de la ventana del navegador.
- ▶ **:before** y **:after** están disponibles a partir de la versión CSS2 y sirven para insertar contenido generado por nosotros antes o después del elemento al que le asignamos este pseudo-elemento. La versión 6 de Internet Explorer no soporta su visualización.

Una característica particular de los dos primeros pseudo-elementos es que sólo admiten un subconjunto de propiedades CSS, que describimos a continuación:

- ▶ **:first-letter**: **color**, **background**, **text-decoration**, **vertical-align**,

CÓDIGO 1

```
P {
font-size: 13px;
font-family: arial,
tahoma, verdana;
font-style: normal;
font-weight: normal;
}
P:first-letter {
font-size: 200%;
font-style: italic;
font-weight: bold;
float: left;
}
```

CÓDIGO 2

```
P {
font-size: 13px;
font-family: arial,
tahoma, verdana;
font-style: normal;
font-weight: normal;
}
P:first-line {
font-size: 200%;
font-style: italic;
font-weight: bold;
float: left;
}
```

CÓDIGO 3

```
P {
font-size: 13px;
font-family: arial, tahoma, verdana; font-style:
normal; font-weight: normal;
}
P.noticia:before {content: "Texto generado por el usuario. "
```

text-transform, **line-height**, **float** y **clear**

- ▶ **:first-line**: **color**, **background**, **word-spacing**, **letter-spacing**, **text-decoration**, **vertical-align**,

text-transform, **line-height** y **clear**.

Analicemos algunos ejemplos del uso de **:first-letter** en el Código 1, **:first-line** en el Código 2, y **:before** en el Código 3. ◀



Cuando utilizamos el pseudo-elemento **:first-line** obtenemos un resultado como éste.

ENLACES BÁSICOS Y AVANZADOS

Los enlaces son una parte fundamental de cualquier sitio. Saber maquetarlos, distinguirlos y hacerlos obvios sin perder la sutileza del diseño es un desafío. Conozcamos sus variantes y las formas en las que deben aplicarse.

Como ya vimos, todos los sitios poseen diversos enlaces. Cada uno de los estados de un enlace puede ser maquetado con distintas propiedades, considerando la función que va a cumplir. Hay varias maneras de maquetar un **enlace** y de vincularlo, dependiendo de su función. Entender el funcionamiento y la estructura de un **enlace** puede facilitar mucho el código semántico y hacerlo, además, un código más breve, limpio y de fácil acceso. Veamos, a continuación, las tres formas más comunes de crear enlaces:

► La primera es el **enlace tipográfico**, aquel que se utiliza desde los comienzos de Internet, y que consta de un texto con propiedades de color y de subrayado especiales. En este caso, nuestro código CSS estaría definido de la siguiente manera:

```
a:link { color: #ededed;
text-decoration:
underline; }
```

► La segunda forma de definir un enlace es mediante una **imagen**. En la actualidad, se la utiliza comúnmente en galerías de imagen, aunque hay quienes siguen empleando esta forma fuera de los sitios de ese tipo. En esta clase de enlaces, el código en HTML se vería de la siguiente manera:

```
<a href="ruta_de_mi_nueva_
página"></a>
```

► Finalmente, la forma más habitual y correcta consiste en definir un **enlace por bloque**. De esta manera, obtendremos un enlace con diseño no sólo mediante imagen sino también con tipografía. En general, este modo de diseñar se utiliza para las **navbar** (barras de navegación) o **botoneras**. En el **Código 4** vemos la semántica que incorporamos en nuestro código CSS.

Para el código HTML, lo aplicamos como vemos en el **Código 5**.

Los enlaces pueden tener diferentes características, y son jerarquizados de acuerdo con su importancia y su relevancia. La jerarquización de un enlace tiene que ver con el diseño y la ubicación que le otorgamos dentro del sitio. Los diseños de los enlaces deben ser pensados en sus distintos estados, ya que, de esta forma, podremos saber si los espacios y las posiciones realmente están bien seleccionados y cuidados. Los links que definimos como texto suelen ser, en su mayoría, los términos de contrato, las políticas de privacidad, las direcciones de e-mail (que se definen como `<a href="mailto:tu_direccion_de_mail">tudirecciondemail</a>`) y los textos para ver más (**view more**) de los sitios dinámicos. También se usan en las sidebars de las webs dinámicas. Por otra parte, los enlaces generados por bloques la mayoría de las veces suelen contener tipografías que no son estándar



El sitio www.naomicampbell.com posee enlaces con texto (el copyright), enlaces con imagen (la cámara de filmar) y enlaces en bloque con diseño (bajo la navbar, denominado fashion for relief).

CÓDIGO 4

```
ul { Definir las propiedades }
ul li { Definir las propiedades de la lista }
ul li a { Definir las propiedades del enlace }
ul li a .nombre_de_la_clase { Definir las propiedades de la
clase }
ul li a span { Definir las propiedades de la etiqueta span }
ul li a: hover { Definir las propiedades de la pseudo-clase
para efectuar el comportamiento }
```

CÓDIGO 5

```
<ul><li><a href="enlace" class="nombre_de_la_
clase"><span>Contenido tipográfico</span></a></li></ul>
```

(no vienen instaladas por defecto con el sistema operativo), sombras, fondos realizados con degradados o con imágenes y, otras veces, con efectos visuales.

PROPIEDADES EN LOS ESTADOS DEL ENLACE

Las propiedades que podemos utilizar en los enlaces son diversas y variadas, y todas pueden aplicarse en los distintos estados de éstos. Por ese motivo, recordemos que, como dijimos anteriormente, el link debe ser pensado en sus cuatro

HTML 5 Y CSS 3 NOS BRINDAN NUEVAS ETIQUETAS PARA GENERAR DEGRADADOS, APLICAR SOMBRAS O USAR FUENTES NO ESTÁNDAR.

estados. Veamos las propiedades que tenemos disponibles:

► **background:** son aquellas que hacen referencia al fondo. Se utilizan cuando queremos colocar una imagen como botón en un bloque generado mediante una lista ordenada o desordenada, o cuando queremos utilizar un color de la misma forma.

► **color:** la usamos cuando queremos cambiar el color del enlace en alguno de sus estados o en todos ellos.

► **font:** sirve para generar cambios tipográficos, ya sea de familia tipográfica, variaciones en el cuerpo, inclinación o interlineado.

► **text-decoration:** nos permite quitar o colocar subrayado, tachado o, incluso, una sobrelínea.

► **visibility:** lo utilizamos cuando no queremos que se vea el texto de un bloque generado por lista.

Para finalizar, debemos tener en cuenta que el lenguaje HTML y CSS tienen un gran potencial, que día a día se desarrolla aún más con el objetivo de poder prescindir de ciertos objetos a la hora de construir enlaces y hasta el sitio completo. Esto es importante porque se han comenzado a implementar las etiquetas para HTML 5 y CSS 3, que nos permitirán generar los enlaces para navbar o botoneras sin necesidad de utilizar imágenes en muchos casos, como cuando queremos hacer uso de colores degradados, aplicar sombras o permitir el uso de tipografías no estandarizadas, que podrán ser alojadas en el servidor, leídas y cargadas por CSS. También nos permitirán generar composiciones con más de un fondo para producir enlaces o fondos web. ◀



En esta sección del sitio podemos usar las imágenes como enlaces para apreciarlas en un mayor tamaño.

ESTILOS PARA LOS ENLACES EN

PASO A PASO

En este apartado explicaremos cómo dar estilo a los enlaces de nuestro sitio en sus diferentes estados.

```

1  <html><head><title>Titulo :
2  </title>
3  <link rel="stylesheet" type="text/css" href="css/estilos.css">
4  </link>
5  <body>
6  <div class="wrapper">
7  <div class="header">
8  <h1>Titulo para enlace interno</h1>
9  </div>
10 <div class="content">
11 <h2>Titulo para enlace externo</h2>
12 </div>
13 </div>
14 </body>
15 </html>

```

Titulo para enlace interno

Enlace externo

01 | LOS ESTADOS

```

1  <html><head><title>Titulo :
2  </title>
3  <link rel="stylesheet" type="text/css" href="css/estilos.css">
4  </link>
5  <body>
6  <div class="wrapper">
7  <div class="header">
8  <h1>Titulo para enlace interno</h1>
9  </div>
10 <div class="content">
11 <h2>Titulo para enlace externo</h2>
12 </div>
13 </div>
14 </body>
15 </html>

```

Titulo para enlace interno

Enlace externo

Enlace interno del 1er título

LINK **02**

```

1  <html><head><title>Titulo :
2  </title>
3  <link rel="stylesheet" type="text/css" href="css/estilos.css">
4  </link>
5  <body>
6  <div class="wrapper">
7  <div class="header">
8  <h1>Titulo para enlace interno</h1>
9  </div>
10 <div class="content">
11 <h2>Titulo para enlace externo</h2>
12 </div>
13 </div>
14 </body>
15 </html>

```

Titulo para enlace interno

Enlace externo

03 | HOVER

Como ya explicamos en las páginas anteriores, los enlaces tienen cuatro estados, llamados **:active**, **:hover**, **:link** y **:visited**. Cada uno de ellos tiene que ver con un estado particular del enlace, y pueden ayudarnos a potenciar nuestro diseño y a mejorar la navegación de los usuarios. ◀

PASO | 01 Para empezar, vamos a definir los cuatro estados que posee un enlace. Crearemos una etiqueta **<style>** dentro de **<head>** y colocaremos estos cuatro estados como se muestra en la imagen.

PASO | 02 El primero que definiremos será el atributo **[link]**. Esta pseudo-clase nos indica que la etiqueta funciona como un botón. En este caso, definiremos un nuevo color, pero no modificaremos el decorado del texto.

PASO | 03 La próxima pseudo-clase que definiremos es **[hover]**, que marca el comportamiento del enlace cuando el mouse está sobre él. En este caso, cambiaremos el color y sacaremos el subrayado dándole el valor **[none]** a la propiedad **[text-decoration]**.

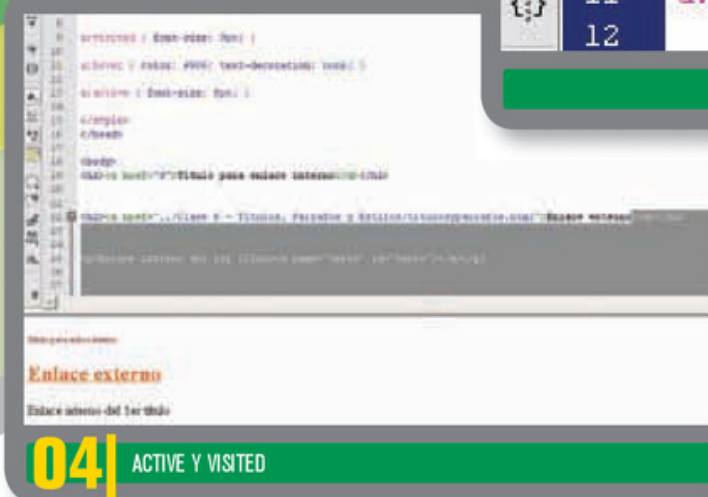
DREAMWEAVER

```

3 <head>
4 <meta http-equiv="Content-Type" content="text/html" charset="utf-8">
5 <title>Enlaces y Estilos</title>
6 <style>
7 h1 a:link { color: #F60; }
8
9 a:visited { font-size: 9px; }
10
11 a:hover { color: #966; text-decoration: underline; }
12 </style>

```

ESTILO ESPECÍFICO | 05



04 | ACTIVE Y VISITED

PASO | 04 Finalmente, vamos a definir un nuevo tamaño para los estados **[visited]** y **[active]**. No es necesario modificar **[active]** porque no tiene un uso real, sólo dura los segundos que tarda en cambiar de HTML. Por su parte, **[visited]** mostrará los links sobre los que ya hemos hecho clic.

PASO | 05 Podemos asignar un estilo específico para el comportamiento de un botón si, antes de la etiqueta **<a>**, colocamos el nombre de otra etiqueta. En este ejemplo, por delante del estado **[link]** escribimos la etiqueta **<h1>**, por lo que el comportamiento específico sólo se aplicará a los enlaces con ella.

PASO | 06 Otra forma de dar estilo a una etiqueta **<a>** es escribiendo el correspondiente dentro del código HTML que la compone. En este caso, cambiamos el tamaño tipográfico usando la

propiedad **[font-size]**. De esta manera, estamos utilizando un valor para font-size de 36 píxeles. Este parámetro queda a nuestro criterio, y podremos dar el tamaño que corresponda en cada ocasión.

RESUMEN

Como vemos, la forma de dar estilos generales a los enlaces es muy sencilla. Sólo debemos escribir **[a:]** seguido del nombre de la pseudo-clase. Estos estilos aparecerán en el panel **[Estilos CSS]**, donde podremos modificarlos como hemos visto anteriormente. Es importante mencionar que, en general, únicamente hay que acotar los estilos (tanto en los enlaces como en otras etiquetas) si existe un motivo real, ya que es más sencillo trabajar con estilos generales.

TAMAÑO DE FUENTE | 06

ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 ¿Qué diferencia hay entre los enlaces externos y los internos?

02 ¿Qué diferencia existe entre los enlaces absolutos y los relativos?

03 ¿Cuál es la ventaja de utilizar enlaces relativos?

04 ¿Qué son las pseudo-clases?

05 ¿Cuáles son los cuatro tipos de pseudo-clases?

06 ¿Cuáles son los pseudo-elementos disponibles a partir de **CSS2**?

07 Mencione las tres formas más comunes de crear enlaces.

08 ¿Cuáles son las propiedades en el estado del enlace?

09 ¿Qué subconjunto de propiedades permiten los pseudo-elementos **:first-letter** y **:first-line**?

10 ¿Con qué tiene que ver la jerarquización de un enlace?



PRÁCTICAS

01 Cree un enlace absoluto y transfórmelo en uno relativo.

02 Inserte un enlace interno y otro externo en su sitio.

03 Aplique diferentes pseudo-clases en los links creados.

04 Defina un enlace mediante una imagen.

05 Defina las propiedades en los estados de un enlace.



Maquetado

- ▶ TABLELESS
- ▶ ESTRUCTURA BÁSICA DE UN SITIO
- ▶ POSICIONAMIENTO DE ELEMENTOS
- ▶ ARMAR LA ESTRUCTURA DE UN SITIO
- ▶ MODELO DE CAJAS
- ▶ DIAGRAMA GRÁFICO DEL MODELO DE CAJAS
- ▶ UNIDADES DE MEDIDA
- ▶ PROPIEDADES DE TAMAÑO
- ▶ MARGEN Y RELLENO
- ▶ BORDES (CSS)
- ▶ TENDENCIAS EN EL DISEÑO DE FONDOS
- ▶ PROPIEDADES Y APLICACIÓN DE FONDOS
- ▶ EL BACKGROUND DE NUESTRO SITIO



TABLELESS

La forma actual de estructurar los sitios es bastante diferente de la utilizada hace algunos años. Conozcamos la técnica tableless, el arte de construir sitios sin recurrir a las tablas.

En sus comienzos, el lenguaje HTML tenía múltiples carencias, ya que no se pensaba en la Web como en un multimedio con diseños creativos, sino que la idea era transferir y compartir datos científicos. Por esta razón, no era posible alinear una página hasta darle la forma que se quisiera, sino que, simplemente, se ordenaba y se jerarquizaba el texto.

Por esta razón, los diseñadores comenzaron a usar lo único que les permitía formar de alguna manera una estructura: las tablas. Este recurso funcionó durante un tiempo, pero con la aparición de distintos navegadores, motores de búsqueda y diversas maneras en que la gente empezó a pensar la Web, fueron dejadas de lado.

¿POR QUÉ DEJAMOS DE LADO LAS TABLAS?

Esto se debe a que, fundamentalmente, las tablas se utilizan para presentar contenido tabulado, y su estructura está conformada por filas y columnas. Si las utilizamos para la estructura de un sitio, nos encontraremos con múltiples proble-

mas a la hora de diagramarlo, ya que deberemos anidar varias tablas para conseguir la disposición deseada, y tendremos que colocar el diseño y el contenido en el interior del sitio, algo que resulta poco práctico desde el diseño. Además, si maquetamos el sitio con tablas tendremos otras desventajas:

Aquí podemos apreciar un prototipo realizado con tablas. A la izquierda, la página abierta en un navegador y a la derecha, el código con tablas.

```

10 
11 <link rel="stylesheet" href="http://www.bastadepaginas.com.ar/favicon.ico" />
12 </head>
13 <body>
14 <table width="834" height="383" border="0" align="center">
15 <tr>
16 <td colspan="2" width="834" height="383">
17 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
18 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
19 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
20 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
21 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
22 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
23 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
24 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
25 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
26 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
27 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
28 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
29 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
30 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
31 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
32 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
33 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
34 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
35 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
36 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
37 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
38 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
39 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
40 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
41 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
42 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
43 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
44 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
45 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
46 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
47 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
48 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
49 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
50 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
51 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
52 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
53 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
54 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
55 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
56 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
57 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
58 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
59 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
60 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
61 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
62 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
63 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
64 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
65 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
66 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
67 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
68 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
69 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
70 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
71 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
72 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
73 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
74 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
75 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
76 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
77 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
78 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
79 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
80 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
81 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
82 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
83 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
84 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
85 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
86 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
87 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
88 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
89 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
90 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
91 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
92 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
93 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
94 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
95 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
96 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
97 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
98 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
99 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />
100 <div style="width: 234px; height: 30px; background-color: #FFFFFF; border: 1px solid #000000; margin: 0px 0px 0px 0px;" />

```

Bdep
DESIGN

Nosotros

En este momento estamos en un proceso de desarrollo de un nuevo sitio web, con el propósito de mejorar la experiencia de usuario.

Además, estamos en un proceso de desarrollo de un nuevo sitio web, con el propósito de mejorar la experiencia de usuario.

Objetivos

El objetivo principal es mejorar la experiencia de usuario, lo que se logrará mediante la implementación de un nuevo sitio web.

Idioma y Futuro

El idioma principal es el español, pero también se ofrece la opción de inglés. El futuro del sitio web es optimizar la experiencia de usuario.

Experiencia y Calidad

La experiencia de usuario es el foco principal, lo que se logrará mediante la implementación de un nuevo sitio web.

- ▶ El contenido del HTML se vuelve mucho más pesado.
- ▶ El servidor tarda más en acceder y, por lo tanto, habrá menos gente en el sitio.
- ▶ Los buscadores indexan el sitio de manera incorrecta, porque a veces leen su contenido de modo confuso, debido a la forma en la que el contenido se distribuye en el código.
- ▶ El rediseño del sitio resulta incómodo y, en muchos casos, inaccesible, lo que hace que debamos iniciar el proceso desde cero.
- ▶ La forma de pensar la estructura del sitio es complicada y hasta impráctica a la hora de efectuar su implementación.



TABLELESS

Por los motivos que mencionamos, el uso de tablas en la estructura web se está dejando cada vez más de lado, ya que se fue migrando a CSS+HTML/xHTML.

En los inicios del código CSS, no todos los navegadores lo soportaban de forma completa y, a veces, ni siquiera lo hacían de manera parcial. Esto hizo que el paso de las tablas a una estructura dinámica constituida en su mayoría por etiquetas tomara más tiempo. Pero desde hace unos cinco o seis años, casi todos los navegadores soportan el código CSS en su totalidad, lo que permite que haya cada vez más sitios desarrollados siguiendo esta modalidad.

Hace más de una década que el W3C (*World Wide Web Consortium*) se ha encargado de normalizar los estándares web, lo que facilita la accesibilidad y la correcta visualización de los sitios en los distintos navegadores, en tanto estos últimos cumplan las normativas que sugiere el W3C.

Veamos, a continuación, las ventajas de utilizar CSS y dejar de lado las tablas:

- Podemos tener un código más liviano (en muchos casos, hasta en un 50%); por lo tanto, se favorece el tiempo de acceso del servidor.

El mismo sitio (www.bastadepoquitez.com.ar) realizado con **divs**. No hay imperfecciones en el navegador y tenemos un código totalmente limpio.

- ▶ Es posible mantener dentro del código una estructura más ordenada y para nada confusa, ya que sólo hay contenido y no diseño. Esto facilita la indexación del contenido web en los distintos buscadores.
- ▶ Los navegadores muestran el sitio de manera correcta, incluso, en dispositivos móviles, como Palms y teléfonos. En cambio, si utilizamos tablas, esos pequeños equipos no lo mostrarán de forma adecuada.
- ▶ El diseño con CSS nos permite hacer un cambio parcial o total del sitio de manera fluida y práctica. Es así que se genera una optimización en el armado y el mantenimiento o rediseño.

A pesar de estas ventajas, la implementación del sistema tableless ha demorado debido a varios factores. Por un lado, las empresas que desarrollaban los navegadores no querían o no podían cumplir las normas para que CSS fuese implementado de forma óptima. Por su parte, los diseñadores, los desarrolladores y los programadores web consideraban tedioso tener que aprender un código nuevo, sobre todo, porque no se sabía si sería totalmente aceptado e implementado. Finalmente, quienes encargaban la realización de sus sitios no se preocupaban por la forma en la que éstos se estructuraban, ya que lo importante era que se vieran y funcionaran.

Afortunadamente, esto ha cambiado en la actualidad. Las compañías desarrolla-

doras de navegadores, quienes producen los sitios (tanto diseñadores, como desarrolladores y programadores), las empresas y los usuarios se han ido dando cuenta de que la evolución a nivel web no cesará y es progresiva.

Ya no es lo mismo que algo sólo funciona, sino que el funcionamiento debe ser correcto, y su estructura, simple, ordenada y de fácil modificación. Gracias a estas características, avanzamos paulatinamente hacia un código con mayor potencialidad y que nos brindará nuevos recursos. En este camino, el maquetado tableless es un paso hacia esa nueva forma de pensar, manejar y utilizar la Web. ◀

CSS NOS PERMITE
MANTENER DENTRO DEL
CÓDIGO UNA **ESTRUCTURA**
MÁS ORDENADA,
QUE SÓLO LLEVA EL
CONTENIDO Y NO,
EL DISEÑO.

ESTRUCTURA BÁSICA DE UN SITIO

La estructura básica de un sitio está compuesta por una serie de layouts que dependerán, pura y exclusivamente, del sitio que vayamos a maquetar. Veamos los diversos layouts que componen esta estructura fundamental.

Tras haber analizado conceptualmente las posibles estructuras de un sitio web para transmitir su contenido y su información de manera correcta, ahora veremos cómo concretar técnicamente estas estructuras por medio del maquetado HTML y la aplicación de estilos CSS. Para maquetar un sitio, tenemos que hacer un estudio previo del diseño propio o ajeno, dependiendo de si es nuestro o de un tercero. Una vez realizado este análisis, dividimos el sitio por áreas. De forma estándar, éstas llevan, en la mayoría de los casos, un nombre particular.

Cuando comenzamos a realizar el maquetado, debemos generar un contenedor general para todo el sitio. Éste será llamado **wrapper** o **page**, dependiendo de quien lo maquete. Ambos términos son correctos, aunque el segundo a veces puede generar confusión. Una vez que hemos dado los parámetros al **wrapper**, debemos dividirlo de acuerdo con el contenido que tengamos. Para hacerlo, vamos a colocar en su interior una serie de layouts con características particulares. En primer lugar, definimos el **header** o **cabecera**, que se ubicará en la parte superior del sitio y contendrá, en general,

la marca y el slogan de la empresa. Debemos tener en cuenta que este header no es lo mismo que la etiqueta head del documento HTML. También se lo puede llamar **top-header**, **top** o **logo**, y muchos lo utilizan para colocar banners de publicidad. A continuación, definiremos la **navbar**, también denominada **botonera**. Éste es el menú de navegación del sitio, y suele encontrarse en la parte superior y por debajo del **header**. Esta barra puede ser estática o dinámica, y también se la suele llamar **nav**, **navigation** o **nav-wrapper**. En el lateral izquierdo colocamos el **menú**, con los links generales, que suelen ser estáticos. Otras denominaciones pueden ser **links** o **sub-nav**.

En el centro del **wrapper** ubicamos el **main** o **content**, cuyo objetivo es albergar el contenido principal del sitio, ya sea texto, imágenes o ambos. Esta sección también se conoce como **main-content**.

En el lado derecho encontraremos la **sidebar**, que se usa en los sitios dinámicos para mostrar las últimas entradas o publicaciones de artículos. Otras denominaciones para este elemento pueden ser **sub-nav**, **side-panel** o **secondary-content**.

Finalmente, tenemos que ubicar el **footer** o **pie**, que se encuentra en la parte inferior del sitio. Allí tendremos los derechos de copyright, los datos de quien diseñó el sitio y, en algunos casos, información de contacto. También se lo denomina **copyright**.

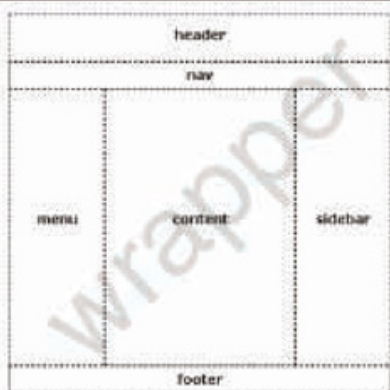
Si bien éstos son los nombres más utilizados para la estructura estándar de un sitio, no es obligatorio que el maquetador los nombre así, ni tampoco que el sitio tenga todos estos **layouts**, ya que esto dependerá del diseño que se esté realizando. Más adelante veremos cómo crear el del sitio que desarrollaremos en esta obra. ◀

Ejemplo de la estructura básica de un sitio. En la parte superior vemos el código semántico, y en la inferior, la forma en la que se verá en el navegador (sin las líneas de puntos).

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Layouts</title>
6 <link href="layouts.css" rel="stylesheet" type="text/css" />
7 </head>
8 <body>
9 <div id="wrapper">
10 <div id="header">header</div>
11 <div id="nav">nav</div>
12 <div id="menu">menu</div>
13 <div id="content">content</div>
14 <div id="sidebar">sidebar</div>
15 <div id="footer">footer</div>
16 </div>
17 </body>
18 </html>

```



POSICIONAMIENTO DE ELEMENTOS

Existe un subgrupo de propiedades CSS que se utilizan para posicionar los elementos de un HTML de diferentes maneras. Aprendiendo a declararlas, podemos generar ricas y variadas estructuras para exhibir los contenidos de nuestras páginas.

Como ya vimos, el lenguaje HTML considera a todos los elementos como cajas. Si éstas no se ven afectadas por reglas CSS de posicionamiento que sobrescriben su ubicación natural, se colocan en la página según su flujo normal. Además de las propiedades predefinidas que poseen los elementos, también existen otras características que los navegadores tienen en cuenta al posicionar un elemento dentro de una página. Algunas de ellas son:

- ▶ Qué tipo de elemento es: si es en línea o en bloque.
- ▶ Cuáles son las medidas del elemento: ancho y alto.
- ▶ Las dimensiones que posee la ventana del navegador.
- ▶ Las relaciones que existen entre los elementos de la página.

Según el estándar de CSS, existe un subgrupo compuesto por cinco propiedades empleadas para posicionar las cajas dentro de un documento HTML de diferentes maneras.

POSICIONAMIENTO NORMAL O ESTÁTICO

Es el que tienen por defecto todos los elementos de HTML según el flujo normal de la página. Si no se indica ninguna propiedad que afecte este parámetro, sólo se tiene en cuenta si el elemento es en bloque o en línea.

Como ya vimos, la regla general es que los elementos en bloque pueden ubicarse dentro de otros elementos en bloque, pero nunca dentro de aquellos que son en línea. Estos últimos, en cambio, no tienen restricciones, ya que pueden ser posicionados dentro de elementos tanto en línea como en bloque.

ELEMENTO EN BLOQUE

ELEMENTO EN BLOQUE

ELEMENTO EN LÍNEA

ELEMENTO EN LÍNEA

Visualización en los navegadores del posicionamiento normal de los elementos en bloque y en línea.

LA REGLA GENERAL ES QUE LOS ELEMENTOS EN BLOQUE PUEDEN UBICARSE DENTRO DE OTROS ELEMENTOS EN BLOQUE.

En general, el ancho de los elementos en bloque es definido por el del elemento que los contiene. Entonces, cuando estos elementos se ubican dentro de otro del mismo tipo, este último se denomina **contenedor**, y es el que define la posición y el tamaño de sus cajas interiores. Si el elemento en bloque no se encuentra dentro de otro elemento, entonces su tamaño y posición son definidos por la etiqueta **<body>**.

Los elementos en línea, en cambio, se ubican uno al lado del otro, y cuando no queda espacio disponible, ya sea en la etiqueta **<body>** o en otro elemento contenedor, entonces el elemento baja y ocupa una nueva línea. ▶

ELEMENTO EN LÍNEA

ELEMENTO EN LÍNEA

ELEMENTO

ELEMENTO EN LÍNEA

Comportamiento normal de los elementos en línea. Cuando no hay más lugar, el elemento baja a la línea siguiente.

POSICIONAMIENTO RELATIVO (POSITION: RELATIVE)

Este tipo de posicionamiento es considerado una variable del posicionamiento normal. Por eso, toma como referencia la ubicación original del posicionamiento estático y, luego, se desplaza en los ejes X o Y, según se indique en el valor de las propiedades declaradas.

El conjunto de propiedad:valor utilizado para controlar este posicionamiento es **[position:relative]**; después, para manejar la distancia del desplazamiento en medidas concretas, se completa con alguna o varias de las siguientes propiedades: **[top]**, **[right]**, **[bottom]** y **[left]**. Todas ellas pueden tener valores tanto positivos como negativos, por ejemplo, **[left: -10px]**.

Analicemos un ejemplo para comprender el comportamiento de esta propiedad. El código CSS es:

```
.caja1 {
  position: relative;
  top: 30px;
  left: 15px;
  background-color: #d5f9f4;
  width: 50px;
  height: 50px;
  margin: 20px;
}
.caja2 {
  position: normal;
  background-color: red;
  width: 50px;
  height: 50px;
  margin: 20px;
  left: 10px;
}
```

El código HTML es el siguiente:

```
<div class="caja1">caja1</div>
<div class="caja2">caja2</div>
<div class="caja2">caja2</div>
```

En el ejemplo anterior, la **[caja1]** se desplaza vertical y horizontalmente respecto de su posición normal. Como el resto de las cajas no modifica su posición, se produce una superposición o solapamiento entre elementos. También podemos observar que el resto de las cajas tampoco ocupa el espacio liberado por la **[caja1]**.

POSICIONAMIENTO ABSOLUTO (POSITION: ABSOLUTE)

Al asignarle posicionamiento absoluto a un elemento, éste se ubica en la página de manera independiente del resto de ellos. Es como si pasara a colocarse en un plano más cercano al espectador, y dejaría en un segundo plano a los demás, que no se ven alterados por su presencia.

Para controlar la ubicación de los elementos con este posicionamiento, también se utiliza la propiedad **[position]**, esta vez, acompañada del valor **[absolute]**. Al igual que el posicionamiento relativo, se combina con las propiedades **[top]**, **[right]**, **[bottom]** y **[left]**.

A diferencia del anterior, el origen de coordenadas del desplazamiento depende del posicionamiento de su elemento contenedor. Analicemos el ejemplo anterior, cambiando el valor del posicionamiento relativo por otro absoluto. En este caso, modificamos el código el CSS de **[caja1]** para que sea como el siguiente:

```
.caja1 {
  position: absolute;
  top: 50px;
  left: 20px;
  background-color: #d5f9f4;
  width: 50px;
  height: 50px;
  margin: 20px;
}
```

El resto del código CSS es el mismo, es decir, no hay modificaciones para la **[caja2]**. Por su parte, el código HTML también se mantiene como en el ejemplo del posicionamiento relativo.



TIPOS DE ELEMENTOS

Recordemos que, según el estándar HTML, todos los elementos se clasifican dentro de dos grandes grupos. Por un lado encontramos los elementos en bloque, que son los que comienzan en una nueva línea y ocupan todo el espacio que existe hasta el final de esa línea. Por ejemplo, las etiquetas P, H1, DIV, FORM, UL y OL. En cambio, los elementos en línea pueden comenzar en una nueva línea o no (generalmente no lo hacen) y sólo ocupan el espacio que necesitan para que se vean sus contenidos. Son elementos en línea las etiquetas SPAN, A, STRONG, IMG e INPUT.

POSICIONAMIENTO FIJO

(POSITION: FIXED)

Ésta es una variante del posicionamiento absoluto, que no se utiliza con demasiada frecuencia en la maquetación de páginas web. La diferencia más importante es que, al recibir esta regla, el elemento afectado se convierte en una caja fija e inamovible, independiente del resto de los elementos y que no se ve afectado si el usuario utiliza el scroll de la página. Las propiedades que se aplican son las mismas que en los dos posicionamientos anteriores, pero el valor para este caso es **fixed**. Veamos un ejemplo en CSS:

```
.cajal {
  position: fixed;
  background-color:
  #d5f9f4;
  width: 50px;
  height: 50px;
}
```

Como podemos notar, para el posicionamiento relativo, absoluto, fijo y estático de una caja se utiliza siempre la propiedad **[position:]**, acompañada, o no, de **relative**, **absolute**, **fixed** e **inherit**.

POSICIONAMIENTO FLOTANTE

Este tipo de posicionamiento del estándar CSS tiene características particulares, y aunque es el más complejo de comprender, es uno de los más utilizados en la maquetación web. Consiste en desplazar la caja todo lo posible hacia la derecha o hacia la izquierda del espacio disponible en la página.

La propiedad de posicionamiento utilizada en este caso es **[float:]**, y los valores que podemos asignarle son cuatro: **[left]**, **[right]**, **[none]** o **[inherit]**. Las cajas con posicionamiento flotante dejan de pertenecer al flujo normal de la página y se ubican lo más que pueden hacia la izquierda o hacia la derecha, según el valor asignado.

COMPORTAMIENTO DE LAS CAJAS FLOTANTES

Las cajas que poseen la propiedad **[float]** no se superponen entre sí, dado que siempre tienen en cuenta las demás cajas



En www.batman-3d.de encontramos un gran ejemplo de fondo fijo. Cuando hacemos scroll, la capa de Batman permanece en su lugar y se mueve sólo el área central de los contenidos.



En www.proxyapparel.com el fondo está compuesto por un div de posicionamiento fijo. Al hacer scroll en el navegador, ese div permanece en su lugar y sólo se desplazan los contenidos.

ELEMENTO CON FLOAT:LEFT

ELEMENTO CON FLOAT:RIGHT

Visualización de dos elementos: el primero, con posicionamiento flotante a la izquierda; y el segundo, a la derecha.

existentes en la página. Suponiendo que hay dos cajas con posicionamiento flotante de valor **[left]**, éstas se ubicarán hacia la izquierda de la página, una al lado de la otra. Más a la izquierda, en primer término, irá la que primero se haya declarado en el código HTML, y luego, la otra. De no contar con espacio suficiente para colocarse en la misma línea, el segundo elemento saltará a la línea siguiente. El mismo caso se puede tomar en cuenta para las cajas con posicionamiento de valor **[right]**.

Veamos un ejemplo CSS:

```
.caja1 {
  float: left;
  background-color: #d5f9f4;
  width: 80px;
  height: 80px;
  margin: 10px;
  padding: 10px;
}
.caja2 {
  float: left;
  background-color: #d5f556;
  width: 80px;
  height: 80px;
  margin: 10px;
  padding: 10px;
}
```

El HTML sería:

```
<div class="caja1">caja1</div>
<div class="caja2">caja2</div>
```

CLEARFIX

La propiedad **[float]** de CSS es muy utilizada para posicionar elementos, pero, en ocasiones, presenta un problema. Si el elemento está ubicado dentro de otro (por ejemplo, un div sin la propiedad **[float]** definida), este contenedor no envuelve al elemento flotante dentro de ella y colapsa. Veamos un código HTML de ejemplo:

```
<div style="border:1px
solid black;
padding:20px;">Elemento 1
sin float
<div style="float:left;
width:200px; back-
ground-color:#beeff4;
padding:20px;">Elemento 2
(ubicado dentro del elemento
1) CON float</div>
</div>
```

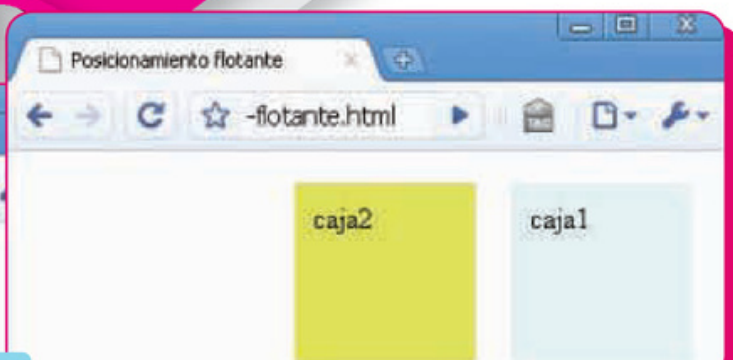
LA SOLUCIÓN ANTE
[FLOAT] ES CREAR
UNA CLASE **[CLEARFIX]**
Y APLICARLA A LOS
ELEMENTOS CONTENEDORES.

Para solucionar este comportamiento, existe un **hack** de CSS para que los elementos flotantes dentro de un div se adapten lo mejor posible a su elemento. La solución consiste en crear una clase **[clearfix]** en nuestro CSS y aplicarla a los elementos contenedores, tal como muestra el siguiente ejemplo CSS:

```
.clearfix:after {
  content: ".";
  display: block;
  height: 0;
  clear: both;
  visibility: hidden;
}
.clearfix {display: inline-
block;}
/* Hide from IE Mac */
.clearfix {display: block;}
/* End hide from IE Mac */
#divflotante {
  float: left;
}
```

El código HTML es el siguiente:

```
<div id="contenedor"
class="clearfix">
  <div id="divflotante">
    Div flotante.
  </div>
</div>
```



A la izquierda, podemos ver cómo quedan dos elementos con la propiedad **[float:left]**. A la derecha, los mismos elementos con la propiedad **[float:right]**.

ARMAR LA ESTRUCTURA DE UN SITIO

PASO A PASO

En este apartado explicaremos cómo armar la estructura básica de un sitio.

01 | REGLAS CSS

WRAPPER | 03

GUARDAR ARCHIVO CSS | 02

Antes de comenzar con el armado del layout, es importante tener el diseño lo más ajustado posible y claro. También es fundamental contar con los archivos editables, llamados crudo; éstos son los PSD o AI necesarios para dar inicio al maquetado del sitio. ◀

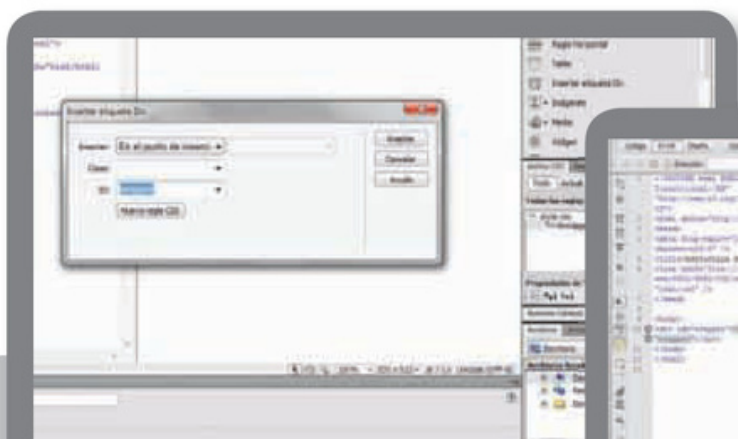
PASO | 01 Vamos al panel derecho [Estilos CSS] y hacemos clic en [Nueva regla CSS]. En el cuadro que se abre, en [Tipo de selector] colocamos [ID (sólo es aplicable a un elemento HTML)], en el campo [Nombre del selector] escribimos [wrapper] y, en [Definición de regla], seleccionamos [(Nuevo archivo de hoja de estilos)]. Una vez elegido estos parámetros, pulsamos [Aceptar].

PASO | 02 A continuación, debemos seleccionar la carpeta en la que queremos guardar el archivo CSS que estamos desarrollando. Para ser ordenados, crearemos una carpeta llamada **css** y, dentro de ella, alojaremos nuestro archivo, denominado **style.css**.

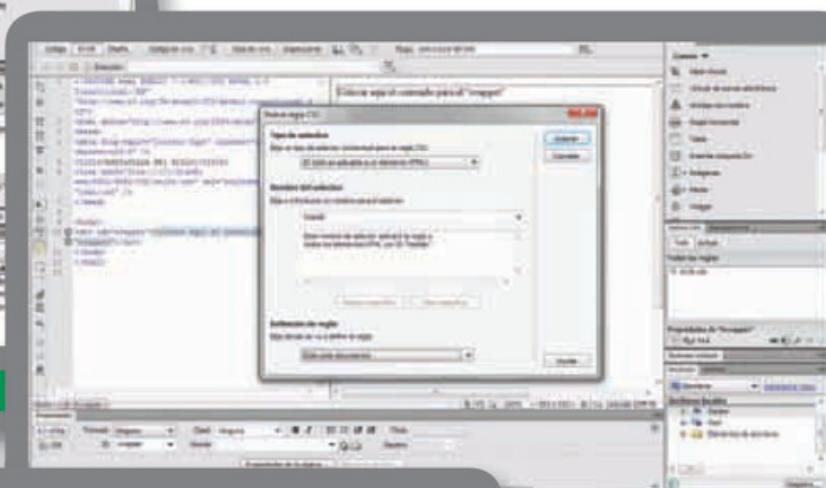
PASO | 03 Al guardarlo, se abre automáticamente la ventana para definir la regla que hayamos creado, en este caso, [wrapper]. En el recuadro [Categoría], hacemos clic en [Cuadro] y, luego, dentro de [Width], ponemos 800 con la medida px y, en [Height], ingresamos 600 con la misma medida. Finalmente, en la propiedad [margin] seleccionaremos [auto] en [Top], para tener centrado el wrapper. Cuando terminamos, presionamos [Aceptar].

ARMAR LA ESTRUCTURA DE UN SITIO

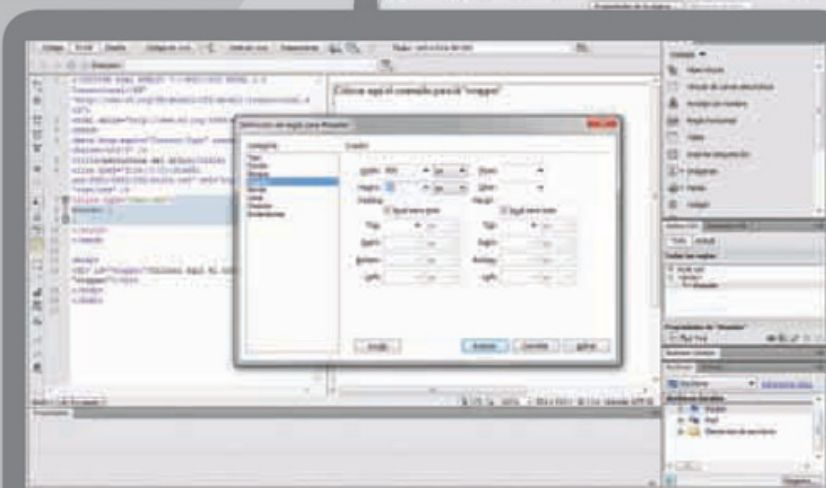
PASO A PASO



04 | AGREGAR DIVS



HEADER | 05

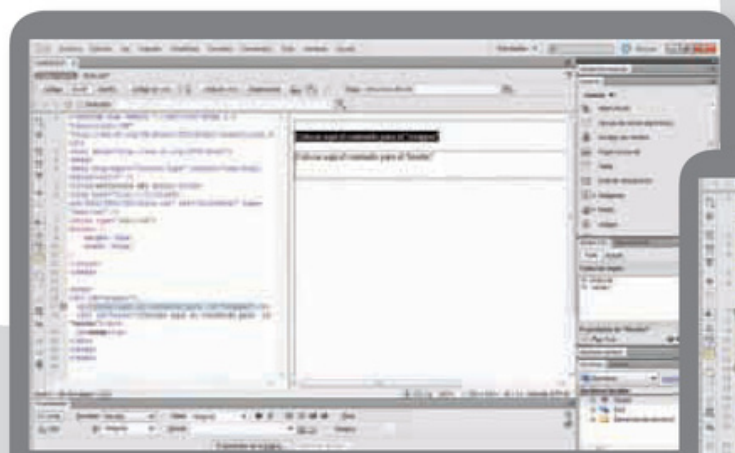


DEFINICIONES | 06

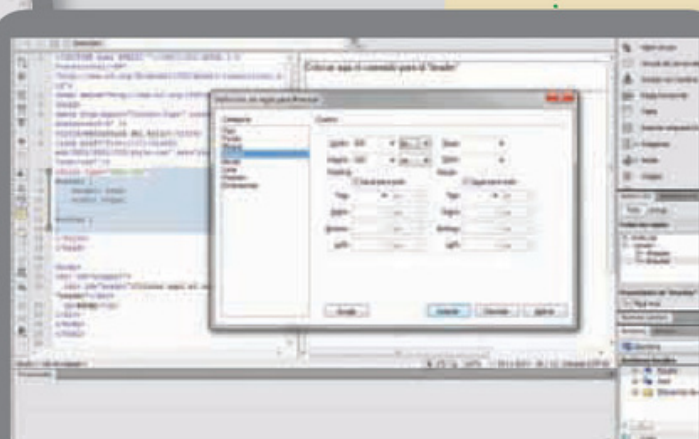
PASO | 04 Vamos al panel derecho y elegimos la solapa **[Insertar]**, dentro de la cual hacemos clic en **[Insertar etiqueta Div]**. Se abrirá la pantalla **[Insertar etiqueta Div]**, desde donde vamos a **[ID]** y seleccionamos la regla **[wrapper]** creada en el primer paso. Al presionar **[Aceptar]**, en la parte inferior de la pantalla, se mostrará nuestra etiqueta seleccionada.

PASO | 05 Ahora, para definir el header, volvemos a crear una nueva regla desde el panel **[Estilos CSS]**. Elegimos **[ID (sólo es aplicable a un elemento HTML)]**, escribimos **[header]** en el campo **[Nombre del selector]** y, en **[Definición de regla]**, seleccionamos **[(Sólo este documento)]**. Al terminar, hacemos clic en **[Aceptar]**.

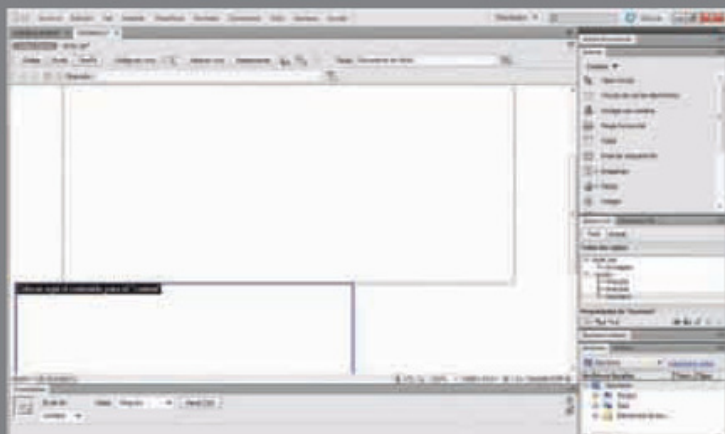
PASO | 06 En la ventana de definición de la regla, dentro del recuadro **[Categoría]**, vamos a **[Cuadro]** y definimos **[Width]** en 800 con la medida px; mientras que en **[Height]**, ingresamos 50 con la misma medida. Finalmente, presionamos **[Aceptar]**.



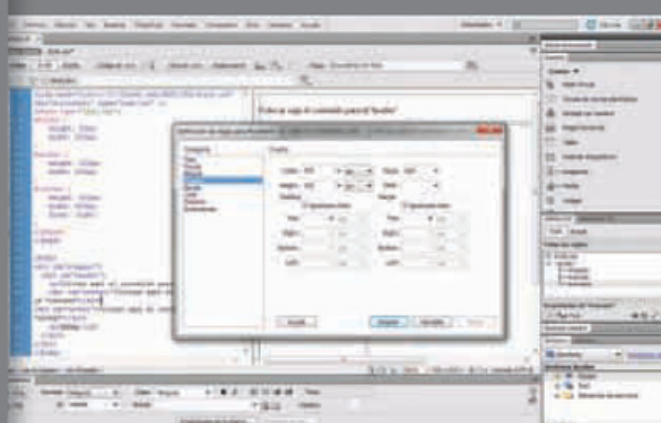
07 | POSICIONAR EL HEADER



ESTRUCTURAR EL SITIO | 08



09 | REGLA [CONTENT]



EDITAR [CONTENT] | 10

PASO | 07 Ahora, asignaremos la nueva regla **[header]** dentro de la regla **[wrapper]**. Para hacerlo, seleccionamos **[Insertar etiqueta Div]** del panel **[Insertar]** y elegimos **[header]** dentro de **[ID]**. Cuando presionemos **[Aceptar]**, se creará un desfase en las medidas del sitio, y para solucionarlo, bastará con elegir el texto **["Colocar aquí el contenido para id "wrapper"]** y presionar la tecla **<Sup>**.

PASO | 08 De la misma manera, crearemos la regla **[navbar]**, configurando las propiedades **[Width]** en 200 con la medida px, y **[Height]** en 520, también con esa medida. Al finalizar, hacemos clic en **[Aceptar]** y, desde el panel **[Insertar]**, incorporamos un nuevo div al que le aplicamos la regla **[navbar]**. Cuando lo hagamos, veremos que la estructura del sitio comenzará a tomar forma.

PASO | 09 Ahora, armamos la regla **[content]**, estableciendo **[Width]** en 600 y **[Height]** en 520. Al finalizar, hacemos clic en **[Aceptar]** y vamos al panel **[Insertar]** para agregar un nuevo div, al que le aplicamos la regla **[content]** en la ventana de creación. Cuando terminemos, veremos que aparece un error que es muy común.

PASO | 10 Para solucionarlo, lo primero que debemos hacer es editar la regla **[content]**, haciendo doble clic sobre ella en el panel **[Estilos CSS]**. En la ventana que se abre, dentro de **[Categoría]**, vamos a **[Cuadro]** y allí, en la propiedad **[Float]**, seleccionamos la opción **[right]**. Luego, pulsamos **[Aceptar]**.

ARMAR LA ESTRUCTURA DE UN SITIO

PASO A PASO

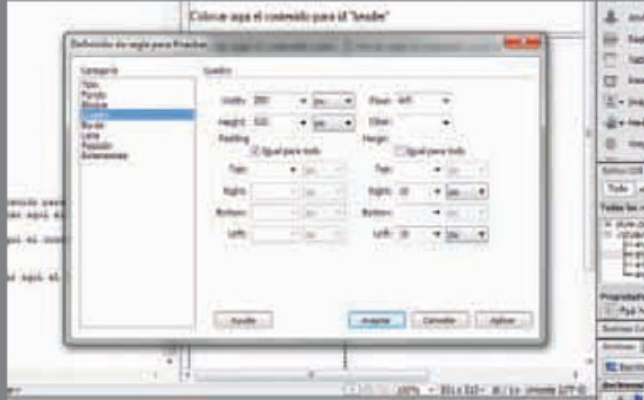


PASO | 11 Para terminar de solucionar el error, vamos a hacer lo mismo que en el paso anterior pero con la regla **[navbar]**. En este caso, a la propiedad **[Float]** le asignamos el valor **[left]**. Después de haber editado las reglas **[content]** y **[navbar]**, la estructura básica se habrá acomodado y tomado la posición debida.

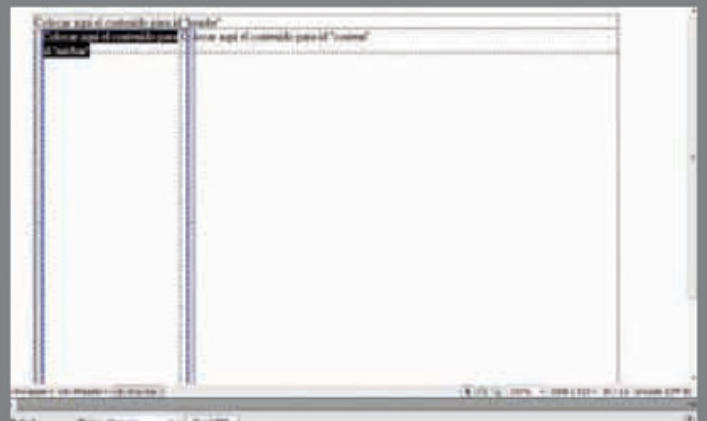
PASO | 12 Para finalizar, crearemos una nueva regla **[footer]**. Para esto, hacemos lo mismo que en las anteriores, pero establecemos la propiedad **[Width]** en 800 y **[Height]** en 30. Al finalizar, hacemos clic en **[Aceptar]** y vamos a incluir el div para esta regla desde el panel **[Insertar]**. Al hacerlo, nos encontramos con el mismo problema que había surgido cuando agregamos el div del contenido.

PASO | 13 Para solucionar el inconveniente, hacemos doble clic sobre la regla **[footer]**, dentro del panel **[Estilos CSS]**. En la ventana que aparece, dentro de la categoría **[Cuadro]** buscamos la propiedad **[Float]** y le damos el valor **[left]**. Cuando pulsamos en **[Aplicar]**, veremos la nueva posición de la regla.

PASO | 14 Ahora, nos encargaremos de asignar algunos márgenes. Comenzamos con la edición de la regla **[navbar]**, en cuya categoría **[Cuadro]** le damos, a la propiedad **[Margin]**, el valor 10 tanto en **[Right]** como en **[Left]**. Antes de hacerlo, quitamos la marca en la casilla **[Igual para todo]**.

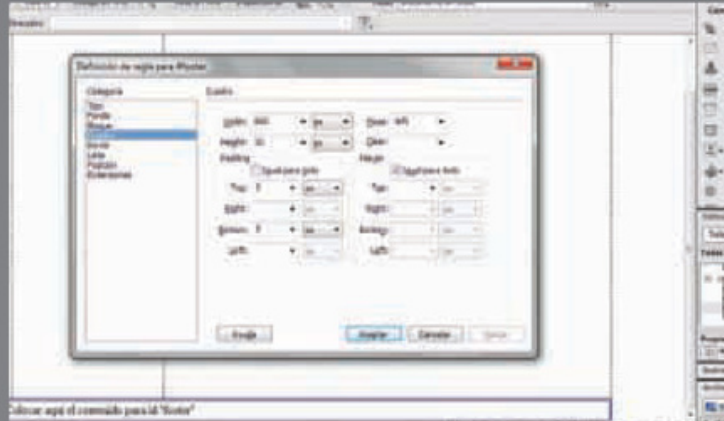


14 | EDITAR MÁRGENES



MÁRGENES PARA [NAVBAR]

15



16 | PADDING



RETOQUES FINALES

17

PASO | 15 Al hacer esto, el div perteneciente a la regla **[content]** pasará abajo. Entonces, editamos su regla y le restamos los 20 píxeles de ancho que le asignamos a los márgenes de la regla **[navbar]**.

PASO | 16 Abrimos la regla **[footer]** para editarla y, dentro de la categoría **[Cuadro]**, quitamos la marca **[Igual para todo]** de la sección **[Padding]**, y asignamos valor 5 a **[Top]** y a **[Bottom]**. Como al hacer esto el div será 10 píxeles más alto, tendremos que restarlos de la propiedad **[Height]**, para dejarla en 20.

PASO | 17 Vamos a editar una vez más la regla **[content]**, colocando **[auto]** en la propiedad **[Height]** de la categoría **[Cuadro]**. Para finalizar, editamos la regla **[wrapper]** y le asignamos el valor 100 con la unidad % para la propiedad **[Height]** de la categoría **[Cuadro]**.

RESUMEN

Hasta aquí, hemos realizado la estructura básica de un sitio paso a paso. Al hacerlo, debemos tener en cuenta que si utilizamos la propiedad **[float]**, todo lo que coloquemos después deberá usar esa propiedad. De hecho, es aconsejable usarla en todos los divs, excepto en el wrapper. Por defecto, si no se utiliza esta propiedad, los divs se ubicarán uno por debajo del otro.

Además, cuando aplicamos margen, debemos restarlo de las propiedades asociadas. Por ejemplo, si la propiedad **[Left]** dentro del **[Margin]** de **[content]** es 10, tenemos que restarle 10 píxeles a la propiedad **[Width]**.

El padding es muy utilizado para centrar textos cortos de forma vertical; tal como hacemos con los márgenes, debemos restar a su asociado la cantidad de padding que utilizemos.

MODELO DE CAJAS

El modelo de cajas es la característica más importante de CSS, ya que sobre su base se elabora el diseño de las páginas que vamos a desarrollar.

El modelo de cajas implica que cada elemento HTML de una página es representado como una caja rectangular.

Estas cajas son creadas automáticamente cuando generamos un elemento HTML y encierran todos los contenidos de ese elemento.

Las cajas son invisibles, ya que, por defecto, no muestran ningún color de fondo ni borde. Sin embargo, podemos modificar todas sus características mediante CSS. Cada caja está formada por seis partes, como vemos a continuación:

- ▶ **Content** (contenido, en inglés): es el contenido HTML que le agreguemos al elemento; puede ser texto, imagen, etcétera.
- ▶ **Padding** (relleno, en inglés): es el espacio entre el content y el border.
- ▶ **Border** (borde, en inglés): es la línea que encierra por completo el content y su padding.
- ▶ **Background image** (imagen de fondo, en inglés): es la imagen que se muestra por debajo del content y el padding.
- ▶ **Background color** (color de fondo, en inglés): es el color que se presenta por debajo del content y el padding.
- ▶ **Margin** (margen, en inglés): es el espacio entre la caja y el resto de las cajas que estén por fuera de ella.

Este orden establece, por ejemplo, que, al definir una imagen de fondo para la caja, la imagen jamás tapaná el contenido, ya que éste está definido primero. Por su parte, la imagen de fondo tiene prioridad sobre el color de fondo, pero, si



----- LÍMITE DE MARGIN ----- LÍMITE DE PADDING
 ——— LÍMITE DE BORDER ——— LÍMITE DE CONTENIDO

la que definimos no llega a cubrir la totalidad de la caja, se visualizará el color de fondo en el espacio sobrante. Tanto el padding como el margin son transparentes.

En el espacio padding se muestra el color o la imagen de fondo de la caja (siempre y cuando estén definidos). En el espacio margin, aparece el color o la imagen de fondo del elemento padre (siempre y cuando estén definidos). En el siguiente ejemplo definimos las seis características para una caja. Primero, el contenido en el HTML:

```
<div>Hola que tal, este  
es el contenido de nuestra  
caja</div>
```

Representación gráfica del modelo de cajas.

Y luego, en la hoja de estilos CSS: ◀

```
div{  
    padding: 20px;  
    border: 1px solid  
#CCCCC;  
    background-image:  
url(mi_fondo.jpg) left top  
no-repeat;  
    background-color:  
#000000;  
    margin: 10px;  
}
```

DIAGRAMA GRÁFICO DEL MODELO DE CAJAS

Como ya vimos, el modelo de cajas es la característica más importante de CSS, debido a que es bajo el cual se diagraman los elementos HTML de una página.

Para comprender mejor cómo funciona el modelo de cajas, en esta representación gráfica analizaremos la

composición de un sitio donde vemos distintos elementos HTML (divs, en su mayoría). Cada uno de los recuadros corresponde a una caja. ◀

```
#header {
  margin: 0 auto;
  padding-bottom:
  36px;
  width: 914px;
}
```

```
.mini_box h2 {
  border-bottom: 2px
  solid #AFE813;
  color:#AFE813;
  font-size: 23px;
  font-weight: lighter;
  margin-left: 10px;
  margin-top: 21px;
  min-height: 54px;
  padding-bottom: 8px;
  width: 185px;
}
```

```
.intro {
  background-color: #1A181D;
  height: 167px;
  margin-bottom: 30px;
  margin-top: 34px;
  width: 914px;
}
```

```
.mini_box {
  background: url("../img/
  mini_box_top.jpg")
  no-repeat scroll 0 0
  #1A171B;
  float: left;
  height: 340px;
  margin-left: 14px;
  margin-top: 20px;
  width: 210px;
}
```



UNIDADES DE MEDIDA

Las medidas en CSS se utilizan, por lo general, para definir el ancho, el alto, los márgenes y los tamaños de las tipografías de la página HTML que estemos desarrollando. Conozcamos sus características.

Las medidas se determinan con valores numéricos que pueden ser **enteros, decimales, positivos o negativos**. Luego de ellos, debemos poner la unidad que corresponda. Si indicamos el valor numérico en 0 (cero), podemos no especificar la unidad de medida; pero si el valor numérico es distinto de cero y no indicamos ninguna unidad, la mayoría de los navegadores de Internet ignorará la regla. Éste es un error habitual de los diseñadores que recién comienzan a manejar CSS, y es importante que le prestemos atención para evitar cometerlo.

A continuación, analizaremos tres casos de estilos aplicados a un párrafo. En el primero, la regla CSS está mal aplicada, porque no contiene una unidad de medida:

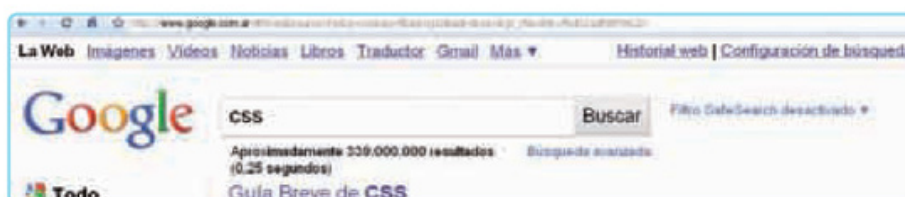
```
p{ width: 500; }
```

En cambio, aquí vemos dos reglas correctas:

```
p{ padding: 0; }
```

```
p{ margin-left: 10px; }
```

Las unidades se dividen en dos grupos: relativas y absolutas. Conozcamos las características de cada tipo.



Vemos una búsqueda de Google con la tipografía por defecto. Cuando usamos textos mayores, se aplican unidades relativas, ya que todo aumenta de manera proporcional.

UNIDADES RELATIVAS

Estas unidades se definen en función de la medida de algún otro elemento HTML. Las relativas son más flexibles que las absolutas, ya que pueden adaptarse a diferentes medios. Además, al incrementar o reducir la medida de referencia, todas las medidas relativas se incrementarán o se reducirán proporcionalmente, y por lo general, conservarán las relaciones de tamaño de nuestro diseño.

Las unidades relativas son **em**, **ex** y **px**. Las dos primeras son proporcionales a los tamaños de tipografía utilizados, y la última es proporcional a la resolución de pantalla del usuario. La unidad **em** tiene su origen fuera del ámbito de CSS y se define, inicialmente, como el ancho de la letra eme mayúscula (M) de la tipografía que se esté usando. En CSS, **em** hace referencia a la unidad de medida de la tipografía que esté utili-

zando el **elemento HTML padre**.

Por ejemplo, si definimos el siguiente código, la tipografía de nuestro párrafo será un 20% mayor que la de su elemento padre; en este caso, **<body>**:

```
body{ font-size:10px; }
p{ font-size:1.2em; }
```

Para calcular el tamaño que tendrá la tipografía del párrafo, debemos hacer una sencilla cuenta: 10 (tamaño de letra del **<body>**) * 1.2 (tamaño relativo de letra del párrafo). Esta operación da 12, que es el tamaño en **px** que tendrá la tipografía del párrafo.

Aunque al principio todo esto puede parecer muy complicado, con la práctica nos daremos cuenta de que es sencillo, e incorporaremos estas relaciones de proporción en nuestro trabajo habitual.

Como ya mencionamos, la principal ventaja que tiene el uso de unidades relativas es que nos permiten mantener las proporciones de un diseño independientemente del tamaño por defecto que esté utilizando el usuario que visita nuestra página. La lógica de la unidad **ex** es exactamente igual a la de la unidad **em**, sólo que se toma como referencia la altura de la letra equis minúscula (x). En CSS, lo más común es utilizar las unidades **px** y **em**, donde **px** (píxel) es relativa respecto a la pantalla del usuario.

PORCENTAJES

Otra unidad relativa en CSS son los porcentajes. Esta unidad de medida está formada por un valor positivo seguido del símbolo %. Los porcentajes se utilizan, generalmente, para definir el ancho de los elementos. Por ejemplo:

```
div#principal{ width:600px; }
div#contenido{ width:50%; }
```

Donde nuestro código HTML es el siguiente:

```
<div id="principal">
  <div id="contenido">
    ...
  </div>
</div>
```

En este caso, el **[div]** cuyo atributo **[ID]** es **[contenido]** tendrá 300 px de ancho, porque mide el 50% (la mitad) de su elemento padre, que es el **[div]** cuyo atributo **[ID]** es **[principal]**. Un problema habitual que tendremos al utilizar unidades relativas es que el texto se verá cada vez más chico o más grande. Por ejemplo, si nuestro código CSS es:

```
body { font-size:10px; }
div{ font-size:1.5em; }
```

El HTML debería ser como vemos en el ejemplo de la derecha.

En este caso, **[Texto 1]** tendrá un tamaño de 15 px ($10 * 1.5 = 15$); **[Texto 2]** tendrá un tamaño de 22.5 px ($10 * 1.1 * 1.1 =$

```
<body>
<div>
<p>Texto 1</p>
<div>
<p>Texto 2</p>
<div>
<p>Texto 3</p>
</div>
</div>
</body>
```

LAS MEDIDAS SE DETERMINAN CON VALORES NUMÉRICOS, QUE PUEDEN SER ENTEROS, DECIMALES, POSITIVOS O NEGATIVOS.

22.5) y **[Texto 3]** tendrá un tamaño de 33.75 px ($10 * 1.5 * 1.5 * 1.5 = 33.75$). Para solucionar este tema, debemos ser más específicos con las relaciones de los elementos HTML, utilizando los atributos **[id]** y **[class]** vistos anteriormente.

UNIDADES ABSOLUTAS

Las unidades absolutas definen las medidas de forma específica, debido a que

sus valores reales no dependen ni hacen referencia a los de otros elementos HTML. Las unidades absolutas son:

- **in**, del inglés *inches*, pulgadas (una pulgada equivale a 2,54 centímetros)
- **cm**, centímetros
- **mm**, milímetros
- **pt**, puntos (un punto equivale a 1/72 pulgadas, es decir, alrededor de 0,35 milímetros)
- **pc**, picas (una pica equivale a 12 puntos, es decir, alrededor de 4,23 milímetros)

La mayoría de estas unidades absolutas están en desuso, a excepción de los puntos (**pt**), que se utilizan para indicar el tamaño de tipografía de los documentos que vamos a imprimir.

RECOMENDACIONES

Por lo general, es aconsejable utilizar unidades relativas siempre que sea posible, ya que esto hará que nuestra página sea más flexible para ser visualizada en diferentes dispositivos. Lo habitual es emplear la unidad **px** (píxel) y porcentajes (%) para definir los elementos principales de las páginas, y **em** y porcentajes para el tamaño de la tipografía. ◀

En Dreamweaver, el texto se agranda. Para evitarlo, debemos ser más específicos con las relaciones de los elementos HTML.

```
16 <body>
17 <div>
18 <p>Texto 1</p>
19 <div>
20 <p>Texto 2</p>
21 <div>
22 <p>Texto 3</p>
23 </div>
24 </div>
25 </div>
26 </body>
27 </html>
```

Texto 1
 Texto 2
 Texto 3

PROPIEDADES DE TAMAÑO

Ya hemos visto las unidades de medida. Es hora de conocer las propiedades que nos permitirán determinar el tamaño de los elementos incorporados a nuestras páginas.

La altura y la anchura de un elemento HTML están controladas por los atributos llamados **width** y **height** en CSS, que se traducen como ancho y alto, respectivamente. Conozcamos cómo se utiliza cada una de ellos.

ANCHO (WIDTH)

La propiedad que controla la anchura es **width** y se emplea para determinar el ancho de la caja. Veamos su uso en el código CSS:

```
#caja {
width: 60px;
}
```

El código HTML sería:

```
<div id="caja">Esta sería la
caja contenedora de un div
con 60px de ancho
</div>
```

Mediante líneas punteadas, en el primer ejemplo de uso de **width** vemos que, al no tener un ancho especificado, el párrafo emplea todo el elemento contenedor. En el segundo caso se le asigna el 50% de su elemento padre y, luego, se expresa en ems y en píxeles.

ALTO (HEIGHT)

La propiedad que controla la altura es **height** y se emplea para determinar el alto de la caja. A continuación, veremos

cómo se establecen sus valores para un elemento HTML. En este caso, será para el **<div>** con un **[ID]** llamado **[caja]**. Veamos el código CSS:

```
#caja {
height: 60px;
}
```

El código HTML es:

```
<div id="caja"> Esta sería
la caja contenedora de un div
con 60px de alto

</div>
```

En la imagen donde aparece el alto modificado, en el primer caso no se le asigna valor de alto, y actúa de la misma

manera que cuando no se le asigna un ancho. Luego, le asignamos un porcentaje al alto que no es bien interpretado por el navegador (éstos tienen dificultades para interpretar este tipo de medida en el alto). Para finalizar, aplicamos un valor relativo en ems y píxeles.

Para definir el tamaño, es posible emplear, en ambos casos, cualquiera de las unidades estándar que explicaremos a continuación, aunque es común que siempre las encontremos especificadas en píxeles como su unidad de medida:

- **Inherit:** toma los valores del elemento padre. Si éste no tiene altura explícita, se le asigna valor **auto**.
- **Porcentaje:** se calcula tomando de referencia las medidas que tiene asignado el elemento padre.
- **Píxeles:** se le asigna el valor en píxeles.
- **Auto:** éste es el valor que se le asigna al elemento cuando no tiene una altura determinada por CSS. En este caso, el navegador calcula la dimensión sobre la base del contenido y el espacio disponible para establecer el tamaño final.

Ninguno de estos atributos puede tener valores negativos, ya que expresan el tamaño que tendrá el elemento. Ambos pueden aplicarse a cualquier elemento, excepto a aquellos en línea (**inline**) que no sean imágenes. Recordemos que los



Primero se especifican sólo alto y ancho; luego se aplica padding y un borde, que afecta sus dimensiones.

elementos **inline** no generan una nueva línea en el flujo del texto y, por eso, sería inútil asignarles valores de ancho y alto. Por otro lado, cuando definimos las medidas de una caja de otro elemento, debemos tener en cuenta que dichos valores se verán afectados por los de bordes, padding y margen que también se hayan aplicado.

En la imagen de las cajas, en la tercera (la C) se quiso lograr un elemento con las mismas dimensiones que en la primera, pero con bordes de 3 píxeles. Para conseguir el mismo tamaño debemos restar, tanto al alto como al ancho, 6 píxeles, que corresponden a 3 píxeles por cada lado (horizontal o vertical). Es decir, si al lado derecho le quitamos 3 y al izquierdo otros 3, tenemos 6 píxeles menos de ancho. Lo mismo sucede con el alto, para compensar el grosor de los bordes. Con la misma metodología debemos restar 10 px de padding en la segunda caja, dado que ambos factores afectan de forma directa las dimensiones de una caja.

LOS ATRIBUTOS **WIDTH Y HEIGHT** NO PUEDEN TENER VALORES NEGATIVOS, YA QUE **EXPRESAN EL TAMAÑO DEL ELEMENTO**.

LAYOUTS LÍQUIDOS

Como hemos mencionado anteriormente, una de las medidas que podemos utilizar para el ancho y el alto de los elementos es el porcentaje. Aunque prevalece el uso de las medidas definidas en píxeles, por ser más específicas y exactas, existen los llamados **layouts líquidos**, en los que se emplean medidas porcentua-



En www.w3schools.com encontramos un cuadro con las resoluciones más utilizadas a lo largo de los últimos años.

les para la estructuración del sitio. En ellos, para definir los anchos de los elementos se utilizan unidades relativas, como porcentajes o ems, por lo que los elementos se adaptan a la pantalla que emplee el usuario y a su resolución. De esta forma, se aprovecha al máximo el espacio disponible. A pesar de esta ventaja, son más utilizados los layouts fijos, aunque su uso puede generar algunas incomodidades. Por ejemplo, en primera instancia hay que determinar para qué resolución se va a maquetar el sitio, y si un usuario tiene una resolución diferente, verá el sitio distinto. En este caso, corremos el riesgo de que aparezca el scroll horizontal, que no es muy agradable. En la actualidad, se está trabajando para 1024 x 768, pero no todo el mundo lo está aplicando.

En www.mclibre.org podemos observar cómo actúan los diferentes valores de ancho y de alto aplicados a un elemento `<p>`.

Tamaño

En esta página se tratan los siguientes temas:

- Tamaño: width y height
- Tamaño mínimo y máximo: min-width, min-height, max-width y max-height
- Desbordamiento overflow

Tamaño: width y height

La propiedad `width` establece el ancho de un elemento mediante un valor numérico (con su unidad) o un porcentaje (con respecto al elemento padre). Si no se establece, el elemento ocupa todo el ancho disponible.

Propiedad	Valor	Resultado
<code>width: 100px;</code>	100px	El elemento ocupa 100 píxeles de ancho.
<code>width: 50%;</code>	50%	El elemento ocupa el 50% del ancho de su contenedor.
<code>width: 200px;</code>	200px	El elemento ocupa 200 píxeles de ancho.
<code>width: 100%;</code>	100%	El elemento ocupa todo el ancho disponible.

La propiedad `height` establece la altura de un elemento mediante un valor numérico (con su unidad), pero no en porcentaje. Si no se establece, el elemento ocupa el espacio necesario para mostrar su contenido (sin contar los elementos flotantes).

Propiedad	Valor	Resultado
<code>height: 100px;</code>	100px	El elemento ocupa 100 píxeles de altura.
<code>height: 50px;</code>	50px	El elemento ocupa 50 píxeles de altura.
<code>height: 100px;</code>	100px	El elemento ocupa 100 píxeles de altura.
<code>height: 100px;</code>	100px	El elemento ocupa 100 píxeles de altura.

Si se establece la anchura y altura de un elemento, el contenido del elemento puede no caber en el tamaño asignado. La propiedad `overflow` define qué hacer en estos casos.



MARGEN Y RELLENO

Los márgenes externos e internos de un elemento HTML son controlados en CSS por los atributos `margin` y `padding`, respectivamente. Conozcamos cómo utilizar estas propiedades para dar forma a nuestros diseños.

La estructura de un elemento se compone de cuatro áreas anidadas. Desde el interior hacia fuera, encontramos el contenido; luego el `padding`; después el borde, donde finaliza el `box`; y por último el `margin`. Veamos las propiedades que podemos utilizar para manejar y controlar estos elementos y, así, establecer las características del diseño.

MARGEN (MARGIN)

El margen o **margin** de un elemento es la distancia que hay entre una caja y los elementos adyacentes. Los valores que puede tomar el margen son positivos, porcentajes, `auto` o `0` (este último es el que adopta si no se especifica otro).

La propiedad **[margin]** controla los cuatro márgenes de una caja y, luego, disponemos de un atributo específico para cada lado: **[margin-top]** controla el margen superior, **[margin-right]** controla el margen derecho, **[margin-bottom]** controla el margen inferior y **[margin-left]** controla el margen izquierdo.

Estas propiedades son aplicables a cualquier elemento HTML, a excepción de `margin-top` y `margin-bottom`, que sólo valen para elementos que sean de bloque, como imágenes o `div`. Esto significa que un `link`, por ejemplo (`<a>`), no se verá afectado si le aplicamos márgenes verticales.

Si el layout necesita indefectiblemente utilizar este recurso, por lo general se le aplica por CSS la propiedad **[display: block]** y, de esta forma, el elemento sí es afectado por estos márgenes.

Para asignar valores a las cuatro propiedades que determinan los márgenes de un elemento –en este caso, la clase **[caja]**–, especificamos cada uno de ellos de la siguiente manera:

```
.caja{
  margin-top: 10px;
  margin-right: 12px;
  margin-bottom: 20px;
  margin-left: 13px;
}
```

A continuación, podemos ver, por ejemplo, diferentes valores aplicados a un elemento `<img>`. El código en CSS es el siguiente:

```
img{
  margin: 0px 50px 80px
5px;
}
```

Más abajo, vemos la forma abreviada, en donde se indica que los cuatro valores son iguales. Se llama **shorthand**:

```
objeto{
  margin: [mismo valor para
los cuatro márgenes];
}
```

En un código real, esto quedaría así:

```
.caja{
  margin: 5px;
}
```

Otra manera que podemos utilizar es asignar márgenes verticales, por un lado, y horizontales, por el otro:

```
objeto{
  margin: [valor para
márgenes verticales] [valor
para márgenes horizontales];
}
```

En el siguiente código vemos cómo se indica que los márgenes superior e inferior son de 5 píxeles, y el izquierdo y el derecho, de 10 píxeles. Ambos márgenes son iguales entre ellos; y el izquierdo y el derecho también son iguales entre sí.

```
.caja{
  margin: 5px 10px;
}
```

Otra forma de agrupar los valores es como vemos a continuación:

```
objeto{
  margin: [valor para
márgenes horizontales]
[margen superior] [margen
inferior] ;
}
```

En este caso, el primer valor corresponde a los márgenes derecho e izquierdo; el segundo, al margen superior; y el último, al inferior. Es decir que izquierdo y derecho son iguales.

```
.caja {
    margin: 5px 10px
  8px;
}
```

Ahora, veamos cómo trabajar con márgenes que son diferentes. Siempre el primer valor corresponde al margen superior; el segundo, al derecho; el tercero, al inferior; y el último, al izquierdo. Es decir, comenzamos con el primer valor (el superior), y luego el orden continúa en el sentido de las agujas del reloj.

```
objeto {
  margin: [margen superior]
  [margen derecho] [margen inferior]
  [margen izquierdo] ;
}
```

El código es el siguiente:

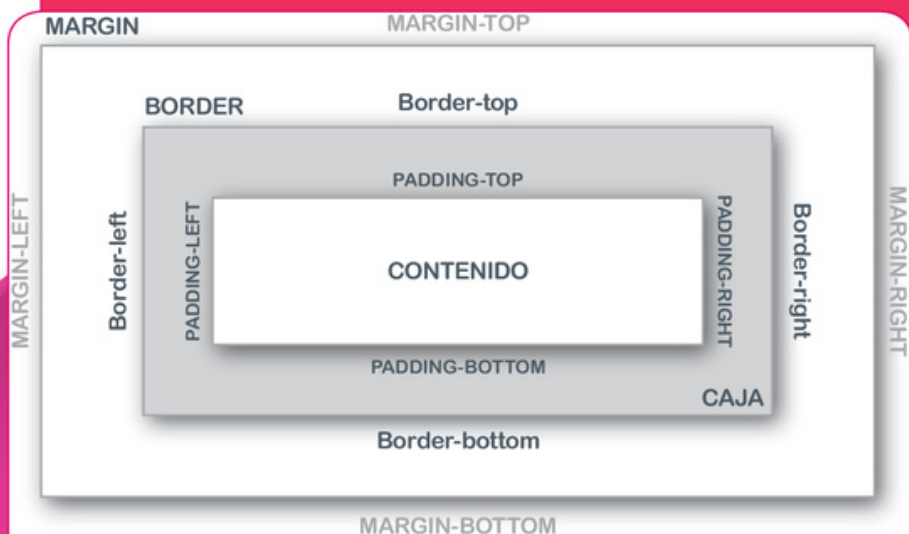
```
.caja {
    margin: 5px 10px 8px
  6px;
}
```

Las unidades de medida que podemos utilizar son las siguientes:

- ▶ **Píxeles:** es la medida más utilizada y proporciona una precisión total.
- ▶ **Em:** es utilizada para hacer diseños que mantengan las proporciones.
- ▶ **Porcentajes:** se usa para hacer diseños líquidos o fluidos (*liquid layout*).

RELLENO (PADDING)

El padding es el espacio interno dentro de un elemento, y separa el contenido de los márgenes; es decir, es la distancia que hay entre el borde y el contenido. Imaginémoslo como lo opuesto a los márgenes, ya que éstos se miden hacia afuera del borde, y el padding, hacia adentro. La propiedad **[Padding]** es genérica; luego, tenemos las que controlan cada lado específico: **[padding-top]** controla el margen superior, **[padding-right]** controla el margen derecho, **[padding-bottom]** controla el margen inferior y **[padding-left]** controla el margen izquierdo. A esta propiedad se le asignan los valores de la misma manera que al atributo



La configuración de **margin** y **padding** nos ayudará a ubicar y a relacionar los elementos de las páginas y establecer posiciones coherentes a nuestras necesidades.

[margin], y sus formas abreviadas son iguales. Esto significa que podemos utilizar la propiedad de tipo **shorthand** de la siguiente manera:

```
Objeto {
  Padding: 10px;
}
```

Si tomamos el portfolio de **www.brunetgarcia.com** como ejemplo y aumentamos el padding a las miniaturas de los trabajos, veremos cómo se agranda el espacio que hay entre el borde y la imagen. El código sería el siguiente:

```
#details img {
  border: 1px solid #999999;
  padding: 25px;
}
```

En cambio, si modificamos el padding a 0, la imagen se pegará al borde. Si además aumentamos el margen, se ampliará el espacio entre las miniaturas. Para lograr esto último, el código sería así: ◀

```
#details img {
  border: 1px solid #999999;
  margin: 25px;
}
```

A las miniaturas de este portfolio (www.brunetgarcia.com) les aumentamos el padding a 25 px.



BORDES (CSS)

A través de la manipulación del código CSS, podemos modificar el aspecto visual de los bordes de nuestros elementos HTML; por ejemplo, ancho, color y estilo. Veamos cómo hacerlo.

Cuando trabajamos con HTML, debemos imaginar que cualquier elemento está inscripto en un rectángulo, cuyos bordes podemos controlar. La propiedad genérica es **[Border]** y afecta a los cuatro bordes que posee la figura (imaginaria), contemplando los atributos de ancho, estilo y color (**[border-width]**, **[border-style]** y **[border-color]**). Esta propiedad vale para cualquier elemento HTML. A continuación, veremos cada una de sus propiedades en detalle, así como la forma de aplicarla en los diferentes lados.

ANCHO DE BORDES

El **grosor** de los bordes puede ser modificado a través de las siguientes propiedades:

- ▶ **border-top-width:** controla el borde superior.
- ▶ **border-right-width:** controla el borde derecho.
- ▶ **border-bottom-width:** controla el borde inferior.
- ▶ **border-left-width:** controla el borde izquierdo.

A cada una de estas propiedades podemos asignarle alguno de los valores permitidos, que son:

- ▶ **thin:** grosor fino.
- ▶ **medium:** especifica un grosor normal. Viene determinado por defecto.
- ▶ **thick:** grosor ancho.



En las capturas observamos que, aunque el diseño sea circular o irregular, la imagen siempre está circunscripta dentro de un cuadrado.

- ▶ **inherit:** toma el ancho del borde del elemento padre.
- ▶ **pixeles:** éste es, tal vez, el más utilizado, ya que permite tener mayor control y precisión sobre el grosor.

A la propiedad **[border-width]** podemos asignarle varios valores en forma resumida, lo que nos ahorra bastante tiempo. Esta modalidad debemos tenerla siempre en cuenta, ya que su sintaxis es igual y aplicable a muchos atributos.

```
.caja {
  border-width: 8px;
  width: 200px;
  height: 200px;
  background-color: gray;
  border-color: red;
  border-style: solid;
}
```



Distintos ejemplos de grosores que podemos aplicar en los bordes. Las medidas en píxeles son determinadas por el usuario; las otras son estándar de CSS.

Por otro lado, cuando se indican dos valores, el primero se aplica a los bordes superior e inferior; y el otro, a los bordes derecho e izquierdo.

```
.caja{
  border-width:8px 1px;
  width: 200px;
  height: 200px;
  background-color: gray;
  border-color: red;
  border-style: solid;
}
```

Cuando se indican tres valores, el primero se aplica al borde superior; el segundo, al derecho e izquierdo; y el último, al inferior.

```
.caja{
  width: 200px;
  height: 200px;
  border-width:12px 1px
5px;
  background-color: gray;
  border-color: red;
  border-style: solid;
}
```

COLOR DE LOS BORDES

Los bordes admiten la asignación de colores a través de las siguientes propiedades:

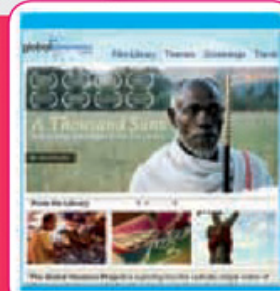
- ▶ **border-top-color:** controla el color para el borde superior.
- ▶ **border-right-color:** controla el color para el borde derecho.
- ▶ **border-bottom-color:** controla el color para el borde inferior.
- ▶ **border-left-color:** controla el color para el borde izquierdo.

Los valores disponibles para el color de los bordes son los que están en notación hexadecimal, por ejemplo, #123456; o los predefinidos, como White, Black, Red, etcétera.

ESTILO DE LOS BORDES

La tercera opción que podemos modificar de los bordes es el estilo. Su propiedad genérica es **[border-style]** y se asigna de la siguiente manera:

- ▶ **border-top-style:** controla el estilo para el borde superior.
- ▶ **border-right-style:** controla el estilo para el borde derecho.



Aquí vemos el resultado de indicar un solo valor para la propiedad **[border-width]**. En el segundo ejemplo aplicamos dos valores y, finalmente, tres.



DOTTED



DASHED



SOLID



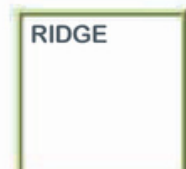
DOUBLE



GROOVE



GROOVE



RIDGE



INSET



OUTSET



Aquí vemos los diferentes estilos de bordes que podemos aplicar mediante CSS.

- ▶ **border-bottom-style:** controla el estilo para el borde inferior.
- ▶ **border-left-style:** controla el estilo para el borde izquierdo.

Como en otros casos, podemos usar el método de escritura llamado **shorthand** para abreviar múltiples reglas CSS. De esta forma, en una sola línea asignamos varios valores a una propiedad; en el caso de los bordes, el ancho, el estilo y el color. Por ejemplo, si tuviésemos nuestro código con las siguientes declaraciones:

```
border-top-style:dotted;
border-top-width:5px;
border-top-color: White;
```

Podríamos abreviarlo de esta manera:

```
border-top: 5px dotted White;
```

Si quisiésemos el mismo estilo para los cuatro bordes, tendríamos el código:

```
border-top: 5px dotted White;
border-right: 5px dotted White;
border-bottom: 5px dotted White;
border-left: 5px dotted White;
```

Gracias al método abreviado, podríamos resumirlo en una sola línea:

```
border: 5px dotted White;
```

Hacer la declaración de esta forma es lo más frecuente, ya que nos ahorra tiempo de escritura y, además, acorta el código. ◀

TENDENCIAS EN EL DISEÑO DE FONDOS



El fondo de un sitio web es el elemento que contextualizará su contenido, transmitiendo cierta atmósfera que dependerá de dos factores: el lenguaje gráfico que trabajemos y su interrelación con el resto de los elementos.

El fondo de un sitio puede darle un toque distintivo y actuar como contexto de la información. Trabajado de manera apropiada, podrá reflejar una determinada época, representar el concepto de minimalismo o, simplemente, dar aire al diseño. Podemos categorizar los fondos según su **aspecto visual**, es decir, el tratamiento gráfico que se aplique; o según su relación con el resto de los elementos de la interfaz. Veamos las opciones del primer grupo.

► **Colores plenos:** son fondos sencillos que transmiten orden y claridad visual. Son pertinentes en los sitios que necesitan priorizar los contenidos, como los de fotografías o tipografías. La aplicación de estos fondos es sencilla, ya que podemos crearlos con la declaración de la propiedad **[background-color]** en el elemento **[body]** de la hoja de estilos.

► **Gradientes:** otorgan mayor profundidad al diseño y se convierten en un detalle que eleva su calidad visual. Para aplicarlos,

es aconsejable repetir una pequeña imagen a modo de patrón, utilizando las propiedades **[background-image]** y **[background-repeat]**. La dirección del gradiente debe ser vertical u horizontal, en ningún caso diagonal.

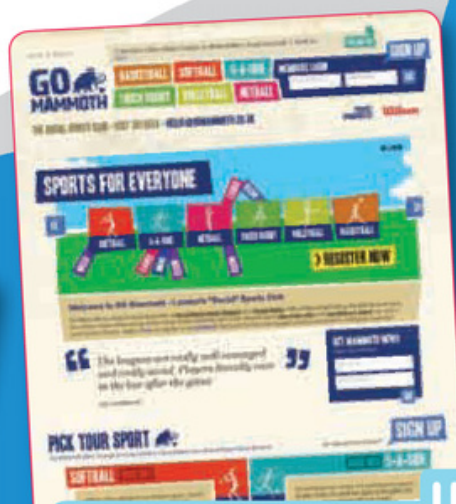
► **Texturas:** son apropiadas para aquellos diseños en los que queremos representar una metáfora o transmitir una cierta atmósfera. Por ejemplo, el uso de papeles viejos denotará el paso del tiempo, mientras que la elección de una textura de acero transmitirá conceptos como asepsia, modernidad o volumen. Las texturas suelen ser aplicadas como una gran imagen de fondo, utilizando la propiedad **[background-image]**, y también es posible hacerlo como **patrón**, repitiendo una pequeña imagen.



Como vemos en www.ljetovanjerab.com/en, los fondos conformados por gradientes aportan profundidad y delicadeza al diseño.



En www.davidfooks.com se destaca el contenido por sobre el contexto, al utilizar plenos de color.



El uso de texturas enriquece el lenguaje gráfico. En www.gomammoth.co.uk vemos un ejemplo.

EL FONDO PUEDE SER UN CONDIMENTO ESPECIAL PARA DAR UN TOQUE DISTINTIVO Y CONTEXTO PARA LA INFORMACIÓN DEL SITIO.



Las fotografías crean entornos realistas. Encontramos un gran ejemplo en www.alexarts.ru/en.



En www.pixelimagedubai.com vemos un sitio cuyo fondo actúa como marco de los contenidos principales.



Las ilustraciones generan ambientes muy ricos, y la creatividad trasciende, como vemos en www.adobeusergroup.nl.



The Blizzards (www.theblizzards.ie) presenta un fondo que disminuye el nivel de profundidad.

► **Fotografías:** si encontramos una fotografía apropiada, podemos lograr un ambiente muy realista al utilizarla como fondo. Habitualmente, la foto utilizada recibe un importante retoque y se aplica de manera fija a través de la propiedad **[background-attachment]**. Los contenidos quedarán por encima de ella y, al hacer scroll, se deslizarán independientemente del fondo. No es aconsejable aplicar una fotografía a modo de patrón, porque su repetición genera saltos visuales.

► **Ilustraciones:** los fondos creados a partir de ilustraciones suelen tener un toque de personalidad u originalidad. Sin embargo, esta técnica en general transmite un espíritu algo infantil, que no siempre resulta apropiado. Las ilustraciones pueden aplicarse como patrones o como imágenes de fondo y, a diferencia de las fotografías, están compuestas por pocos colores, por lo que pueden ser optimizadas con formato GIF o PNG, de peso reducido.

RELACIÓN DEL FONDO CON LOS ELEMENTOS

Además del tratamiento gráfico, para crear la atmósfera adecuada en un sitio debemos tener en cuenta la relación que los fondos mantienen con el resto de los elementos, ya que podemos encontrar varias formas de aplicación:

► **Fondo y contenido en diferentes niveles visuales:** el fondo permanece en un segundo nivel, actuando como marco para el resto de los contenidos del sitio. El contenido principal se ubica dentro de su propio contenedor, que se localiza por delante del fondo, como si se tratara de un plano más cercano al espectador. En ocasiones, el fondo del contenedor se aplica con transparencias, acercando visualmente, en la página, ambos planos.

► **Fondo y contenidos integrados visualmente:** al integrar el fondo con los elementos, disminuye la sensación de profundidad que otorga el uso de diferentes planos. Como estos sitios no están enmarcados por contenedores, transmiten sensación de aire, apertura y, en algunos casos, horizontalidad. Por lo general, el trabajo de implementación del fondo se realiza



PATRONES

Esta técnica consiste en repetir —en el eje X, Y o en ambos— un módulo cuadrado o rectangular para componer un fondo de mayor tamaño. Los patrones pueden ser creados a partir de texturas, fotografías, ilustraciones o gráficos, y su ventaja principal es que el peso total es inferior al de utilizar una gran imagen. Al crear patrones, lo conveniente es generar módulos que, cuando se repitan, compongan una imagen mayor que la original. Existen sitios específicos para crear patrones, como www.bgpatterns.com y <http://es-es.colourlovers.com/patterns/add>.

con mayor cuidado, dado que el riesgo de que los elementos queden desfasados es más notorio que en la implementación de fondos ubicados en diferentes planos.

► **Fondos compuestos por varios fondos:** aquí se aprovecha el fondo para separar los contenidos. De esta manera, se utiliza, por ejemplo, una ilustración para el cabezal y un color pleno u otra imagen como fondo para el cuerpo. Así se logra un corte visual entre ambos sectores. ◀

PROPIEDADES Y APLICACIÓN DE FONDOS



Como ya vimos, el fondo es una gran parte del diseño de un sitio. Una vez que determinamos el tipo de fondo que queremos darle a nuestro trabajo, tenemos que incorporarlo al HTML. Veamos cómo hacerlo.

El fondo de un elemento HTML puede controlarse a través de la propiedad **[background]**, que contempla **[background-color]**, **[background-image]**, **[background-repeat]**, **[background-attachment]** y **[background-position]**.

Vale aclarar que, cuando hablamos de fondo, nos referimos a la zona que ocupa el contenido. No se verán afectados los bordes ni los márgenes, pero sí el padding, si éste existiera, ya que forma parte del contenido.

BACKGROUND-COLOR

Este atributo se usa para determinar el color de fondo de cualquier elemento HTML. Es posible asignarle un valor hexadecimal o RGB, establecerlo por el nombre del color o, simplemente, usar el valor **[transparent]**.

En el siguiente ejemplo, al elemento **<body>** le asignaremos color gris, y a la clase **[caja]**, naranja. En cambio, si al elemento **[caja]** no le asignáramos un color determinado, éste adoptaría el establecido por defecto, que es **[transparent]**.

```
body {
    background-color: gray;
}
.caja {
    width: 200px;
    height: 200px;
    background-color: orange;
}
```

BACKGROUND-IMAGE

Esta propiedad establece una imagen de fondo para cualquier elemento HTML. En este caso, se indica dentro del código la ruta de ubicación de la imagen, es decir, el lugar donde se encuentra el archivo. También es posible usar un color de fondo al aplicar una imagen, ya que estas propiedades pueden emplearse juntas (la imagen se posicionará delante del color de fondo). La sintaxis de esta propiedad es: **url ("ruta a la carpeta donde se ubica la imagen / nombre del archivo y su extensión")**. Veamos un ejemplo:

```
body {
    background-color: #FFFFFF;
    background-image: url ("img/bg-b.jpg");
    margin: 0;
    padding: 0;
}
```

BACKGROUND-REPEAT

Otro de los aspectos del fondo que controlamos con CSS está relacionado con el tamaño de la imagen y de los elementos que la contienen. Si la imagen es más grande que el elemento HTML que la contiene, ésta se verá cortada. En cambio, si es más chica, se repetirá automáticamente en el eje X (horizontal) y en el Y (vertical) formando un mosaico, a menos que se establezcan otros atributos.

EN LA ACTUALIDAD, LOS SITIOS DE VANGUARDIA POSEEN GRANDES ILUSTRACIONES E IMÁGENES EN EL FONDO.

Esta repetición de una imagen puede controlarse a través de los valores de la propiedad **[Background-repeat]**:

- ▶ **repeat-x**: repite la imagen horizontalmente en la página.
- ▶ **repeat-y**: la iteración de la imagen en la página es vertical.
- ▶ **no-repeat**: no repite la imagen.
- ▶ **repeat**: repite la imagen en X e Y.

En el siguiente código se utiliza una imagen como fondo del contenedor y, dado que no se indica otra cosa, ésta se repetirá en ambos sentidos:

```
#contenedor {
    width: 300px;
    height: 300px;
    background-image:
    url(../img/textura.jpg);
    float: left;
}
```


En cambio, al asignarle la repetición sólo en el eje X, la imagen se repite horizontalmente:

```
#contenedor {
    width: 300px;
    height: 300px;
    background-image:
    url(../img/textura.jpg);
    background-repeat:
    repeat-x;
    float: left;
}
```



A la izquierda, podemos ver una imagen repitiéndose en ambos ejes del contenedor. A la derecha, sólo se repite en el eje X.

BACKGROUND-POSITION

A través de esta propiedad, indicamos la posición de la imagen de fondo de un elemento. Es posible definir este parámetro a partir de **palabras clave**:

- **Ubicación horizontal:** **left** (izquierda), **center** (centro), **right** (derecha).
- **Ubicación vertical:** **top** (arriba), **center** (centro), **bottom** (abajo).

O mediante medidas como las siguientes:

- **Porcentajes:** fija la distancia en porcentaje desde la posición inicial, que es 0,0 y está ubicada en el margen superior izquierdo. Su valor por defecto es 50%.
- **Píxeles:** fija la distancia en píxeles desde la posición inicial, que es 0,0 y está ubicada en el margen superior izquierdo.
- **Inherit:** hereda la posición del elemento padre.

En el siguiente ejemplo podemos ver cómo se pueden combinar ambas posiciones utilizando las palabras clave. La primera hace referencia a la ubicación en X, y la segunda, a la posición en Y:

```
body {
    background-image: url
    ("../imagenes/bg.png");
    background-position: left top;
    background-repeat: no-repeat;
}
```

También podemos utilizar para el background la **propiedad shorthand**, que permite definir, primero, el color; a continuación, la ruta de la imagen; luego, su repetición y, finalmente, su posición en X e Y. Veamos un ejemplo:

```
body {
    background: #FFFFFF
    url("grafik/topBack.png")
    repeat-x scroll 0 0;
}
```

Un recurso utilizado al maquetar sitios es el empleo de una pequeña imagen de fondo que se repita en el eje X y, luego, un color pleno. Esto es común en sitios que son muy largos, ya que es posible usar una imagen vistosa o un gradiente y luego, sólo se visualizará el color de fondo.

BACKGROUND-ATTACHMENT

Esta propiedad no es muy utilizada. Su efecto es mantener fija la imagen de fondo de un elemento mientras el contenido se desplaza por delante de ella cuando nos movemos con el scroll.

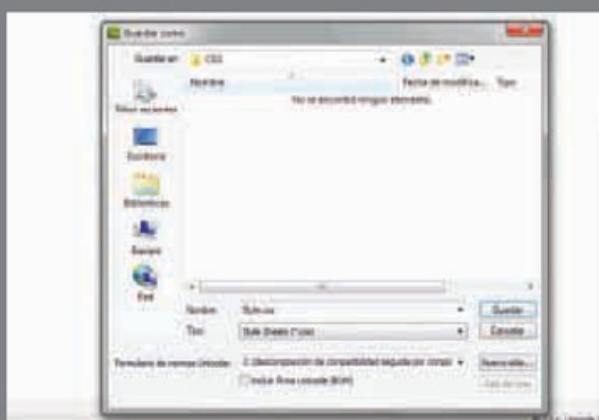


En www.anthonynolan.org, la imagen del degradé se repite en la parte superior del sitio; después se usa un color plano para continuar con el alto de la página.

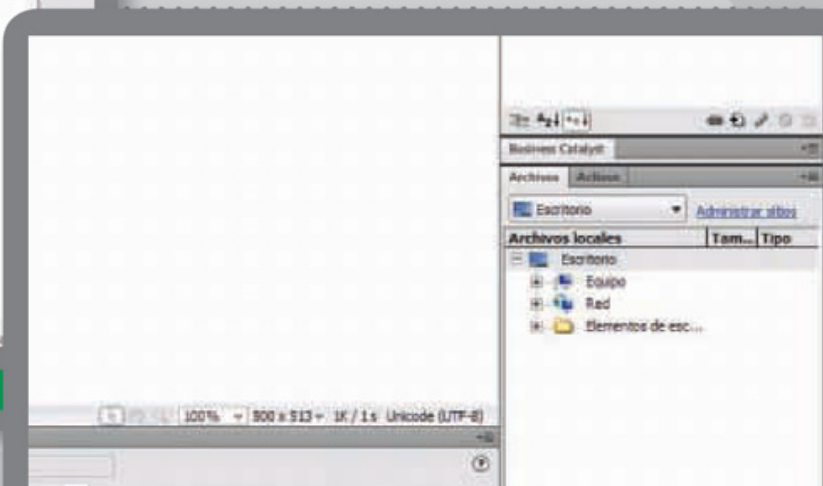
EL BACKGROUND

PASO A PASO

En este apartado, veremos en seis pasos las formas de colocar un background.



01 | CSS



HTML | 02

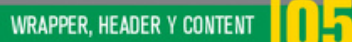
Antes de colocar un background, debemos abrir el archivo de nuestro fondo en el programa de diseño que utilicemos y analizar cómo conviene emplearlo con el código CSS y HTML para integrarlo al sitio. Luego, tenemos que dividir la imagen de la forma que resulte más conveniente. En este caso, cortamos un trozo de header de 16 píxeles de ancho por 56 píxeles de alto, una sección de body de 123 píxeles de ancho por 123 píxeles de alto y, finalmente, hacemos un recorte del footer de 56 píxeles de ancho por 552 píxeles de alto. ◀

PASO | 01 Después de guardar los recortes como archivos JPG, abrimos Dreamweaver y creamos un archivo CSS desde **[Archivo/Nuevo...]**, eligiendo **[CSS]** en el recuadro **[Tipo de página]** y presionando **[Crear]**. Luego, vamos a **[Archivo/Guardar como...]** y ubicamos la carpeta donde almacenaremos la hoja de estilos.

Escribimos style en el campo **[Nombre]** y elegimos **[Style Sheets (*.css)]** en **[Tipo]**. Presionamos **[Guardar]** y cerramos el archivo.

PASO | 02 Creamos un archivo HTML y lo guardamos para trabajar más seguros. En el nuevo documento, dentro del panel **[Estilos CSS]**, hacemos clic en el icono **[Adjuntar hojas de estilo]**. En la ventana que aparece, pulsamos **[Examinar...]** para ubicar la hoja de estilo que creamos anteriormente y presionamos **[Aceptar]**.

PASO | 03 Presionamos **[Aceptar]** en la ventana **[Vincular hoja de estilos externa]** y guardamos el archivo HTML para salvar los cambios realizados. A continuación, hacemos clic en el botón **[style.css]** para comenzar a trabajar en el archivo CSS.



PASO | 05 Para definir el wrapper, escribimos `#[#wrapper { background: url(..../imagenes/header.jpg) 0 0 repeat-x; margin: 0 auto; width: 100%;}]`. Luego, ingresamos `#[#header { margin: 0 auto; padding-bottom: 36px; width: 914px;}]` para generar el header y, finalmente, establecemos el área de contenido con `#[#content { margin: 0`

PASO | 06 Para terminar, definimos el footer con `[#footer { background: url(../imagenes/footer.jpg) 0 0 repeat-x; min-height: 552px; margin: 0 auto; width: 100%;}]`. En este caso, fijamos la altura porque el footer sólo contendrá elementos que están definidos, lo que lo hace invariable. Al hacer clic en **[Código fuente]**, veremos que se aplica el estilo al body de nuestra página.

Más adelante en este libro, cuando aprendamos a crear un layout, veremos cómo debemos generar los divs que darán estructura al sitio para que incorporen estas reglas y, así, la página luzca como deseamos en cada una de sus áreas.

ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 ¿Por qué se dejó de lado la maquetación por tablas?

02 Diferencie el posicionamiento estático del dinámico.

03 Diferencie el posicionamiento absoluto del fijo.

04 ¿Qué es el posicionamiento flotante?

05 ¿Cuál es el comportamiento de las cajas flotantes?

06 ¿Para qué sirve la clase `[clearfix]`?

07 ¿Cuál es la diferencia entre el **padding** y el **margin**?

08 ¿Cuál es la propiedad que determina el ancho?

09 ¿Qué es el **inherit**?

10 ¿Qué son los layouts líquidos?



PRÁCTICAS

01 Cree la estructura de su sitio en Dreamweaver, en base al diagrama que había armado.

02 Edite los bordes, márgenes y padding.

03 Controle el alto y ancho de las cajas con las propiedades **width** y **height**.

04 Elija una caja y defina el grosor, color y estilo de cada borde.

05 Coloque el fondo para su sitio.



Listas

- ▶ LISTAS: DEFINICIÓN
- ▶ LISTAS ORDENADAS Y NO ORDENADAS
- ▶ ¿DÓNDE UTILIZAMOS LISTAS?
- ▶ PERSONALIZAR LISTAS
- ▶ CREACIÓN DE MENÚS DE NAVEGACIÓN
- ▶ CREAR UN MENÚ DE NAVEGACIÓN
- ▶ MENÚ DESPLEGABLE CON CSS Y LISTAS



LISTAS: DEFINICIÓN

Las listas, como seguramente sabemos, sirven para presentar elementos, links, párrafos o imágenes. Listar es una forma de simplificar y mejorar la manera de presentar la información.

Debido a la gran utilidad que tienen, las listas son uno de los elementos HTML que con mayor frecuencia encontraremos en los sitios web. Es importante conocer sus ventajas y las razones por las que es conveniente utilizarlas. Las listas nos permiten mostrar la información de manera clara, porque ordenan visualmente el contenido de un sitio. Gracias a esto, el ojo puede recorrerlo con mayor rapidez y, así, focalizar en el contenido que es de interés para el usuario. Por otra parte, en sitios con contenidos muy extensos, una lista sirve para organizar y separar algunos elementos y, de este modo, evitar lo que podría ser un gran bloque de texto. De esta manera, su lectura se hace más llevadera, y lo importante es más fácil de encontrar y reconocer.

Para analizar el uso de listas podemos visitar el sitio de un periódico, por ejemplo, **www.clarin.com**. Este diario argentino tiene un gran volumen de información que se encuentra segmentada y organizada por temáticas, secciones e importancia, entre otras categorías. Para lograr una diagramación clara del contenido, es importante utilizar elementos que le simplifiquen al lector la tarea de hallar

lo que está buscando. Para esto, además de valernos de títulos, subtítulos, viñetas, recuadros, destacados y copetes, también emplearemos listas. El uso de listas es sencillo y necesario para que el sitio respete los estándares. Este recurso puede usarse en sus formas **ordenadas** o **desordenadas**, y también **anidarse** (es decir, poner

una lista dentro de otra) para conseguir diferentes niveles de lectura. Más adelante veremos cómo deben organizarse las listas y cuál es su estructura. Por ahora, presentemos el código utilizado en **www.copenhagen.chopeh.com** para crear el menú:

```
<ul id="nav">
  <li class="home"><a href="index.php">Home</a></li>
  <li class="about"><a href="about.php">About Me</a></li>
  <li class="work"><a href="work.php">My Work</a></li>
  <li class="help"><a href="help.php">Help Me</a></li>
  <li class="contact"><a href="contact.php">Contact Me
</a></li>
  <li><a class="download" target="_blank" href="CV-Pete-Lacey.pdf">Download CV</a></li>
</ul>
```

La manipulación de listas a través de CSS nos brinda la posibilidad de cambiarles totalmente el estilo, para lograr estéticas muy atractivas y vanguardistas que están bastante lejos de la idea que comúnmente se tiene de ellas. En **www.pingdom.com/free** encontramos en el banner principal un listado en dos columnas, cada uno de cuyos ítem contiene una imagen de fondo que funciona como icono. El uso de imágenes que representen el tema listado es bastante frecuente y enriquece la apariencia visual del sitio. En el mismo sitio, dentro del footer, vemos dos listas con el clásico bullet circular, el uso más estándar de este recurso. En este caso, la información es secundaria, menos importante que la anterior, por lo cual la lista tiene una

aparición menos atractiva, que no compite visualmente con la principal. Como podemos notar, las listas pueden usarse y manipularse a través de CSS según nuestras necesidades, otorgándoles mayor o menor relevancia según su tratamiento.

TRES MOTIVOS PARA UTILIZAR LISTAS

Aunque podemos conseguir los mismos resultados mediante el uso de CSS aplicado a cualquier elemento HTML, la diferencia al utilizar listas está en su practicidad; veamos las razones. En principio, aplicar un estilo a los elementos **** de una lista hace que, luego, si queremos cambiarlo, nos resulte mucho

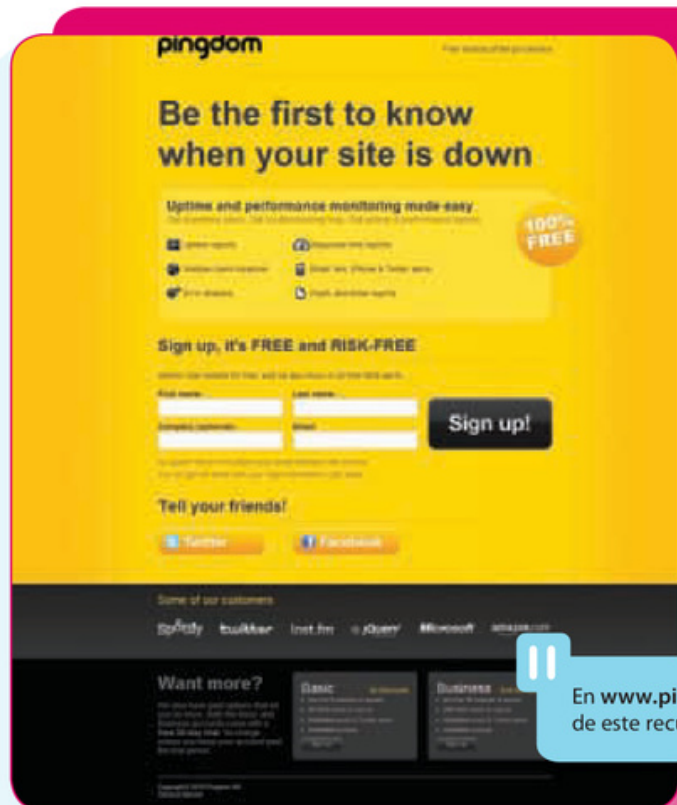
GRACIAS A LAS LISTAS,
EL OJO RECORRE LA INFORMACIÓN
MÁS RÁPIDAMENTE
Y FOCALIZA EN
LO IMPORTANTE.

más sencillo hacerlo. Esto se debe a que no tendremos todos los elementos separados, en cuyo caso deberíamos cambiar el estilo en cada uno de ellos. Aunque más adelante conoceremos con más detalles los tipos de listas y sus características, cabe mencionar que, si tenemos una lista ordenada y queremos cambiar de lugar un elemento, el navegador interpretará que es una `<ol>` (lista ordenada) y, automáticamente, numerará los elementos. En cambio, si hubiésemos escrito los números a mano, tendríamos que reenumerarlos uno por uno, algo realmente incómodo, sobre todo, en listas muy extensas. El uso de listas sirve para que el lector las interprete como tal, lo que ayuda mucho a la lectura y a la ubicación de contenido, dado que las listas poseen, además, una estructura semántica. Todo esto significa que las listas deben implementarse con las etiquetas que tienen este fin, es decir, `<ol>` y `<ul>`. Para nosotros, esta forma de trabajo aumentará la practicidad en el manejo de estos elementos. Para los usuarios son una ayuda porque alivian mucho la lectura, en especial, en sitios que tienen una gran carga de contenido, cuya información puede ser jerarquizada. Si tenemos un sitio con bloques de texto muy extensos, y otro con información destacada, listas, textos en negrita y diferentes

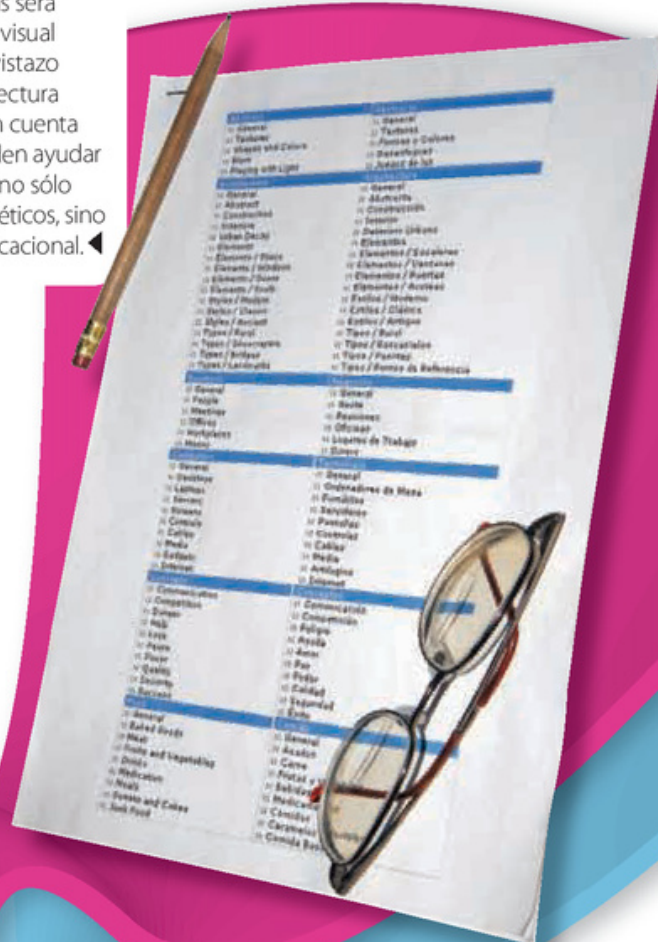


En www.copenhagen.chopeh.com encontramos un ejemplo donde el menú se estructura a partir de una lista desordenada.

tamaños de tipografías, sin dudas será mucho más sencillo el recorrido visual del segundo, tanto para dar un vistazo general como para realizar una lectura detallada. Es importante tener en cuenta todos estos elementos que pueden ayudar a enriquecer nuestras interfaces, no sólo porque resultan agradables y estéticos, sino también por su utilidad comunicacional. ◀



En www.pingdom.com/free podemos ver dos formas de uso de este recurso: una en el listado central, y otra en el footer.



LISTAS ORDENADAS Y NO ORDENADAS

A la hora de presentar elementos con formato de lista, tendremos que decidir entre utilizar una de tipo ordenado o una que no lo sea. Conozcamos sus características y diferencias.

Ya hemos mencionado que, en ciertas ocasiones, debemos recurrir a listas ordenadas cuando precisamos, por ejemplo, numerar los ítem. En cambio, en otros casos, simplemente necesitamos mejorar la presentación de la información y, entonces, recurrimos a listas no ordenadas.

LISTAS ORDENADAS

Estas listas se caracterizan por estar encabezadas por algún tipo de elemento que deja implícito su orden, como podrían ser números o letras.

Todos los elementos pertenecientes a

una lista ordenada se encuentran dentro de las etiquetas `<OL>` y `</OL>`, que significan **Ordered Lists**. La primera le indica al navegador que comienza una lista ordenada, en tanto que la última señala que ésta termina. A partir de la primera etiqueta, se comienza la numeración, y si no indicamos otra cosa a través de CSS, se la implementará con el sistema decimal, que es el más difundido. Dentro de las etiquetas que vimos, encontramos los elementos que componen la lista. Cada uno debe tener, al comienzo, la etiqueta `<li>`, que significa **list item**; y terminar con `</li>`. Como podemos imaginar, este tipo de listas no utiliza imágenes ni bullets, ya que su propósito es sólo mostrar un orden. Es posible manipularlas a través de CSS por medio de la propiedad **[List-style-type]**, que puede tomar los siguientes valores:

- ▶ **decimal-leading-zero**: enumera los elementos a partir de 0.
- ▶ **decimal**: enumera los elementos a partir de 1.
- ▶ **lower-roman**: lista los elementos identificándolos con números romanos en minúsculas.
- ▶ **upper-roman**: lista los elementos identificándolos con números romanos en mayúsculas.
- ▶ **lower-alpha**: identifica los elementos con letras a partir de la "a", en minúsculas.
- ▶ **upper-alpha**: identifica los elementos con letras a partir de la "A", en sus versiones mayúsculas.

Veamos, a continuación, un ejemplo del código HTML con la estructura de una lista ordenada:

```
<ol>    <li>Elemento número
uno</li> <li>Elemento número
dos</li> <li>Elemento número
tres </li> <li>Elemento
número cuatro</li>
<li>Elemento número
cinco</li> </ol>
```

A través de CSS, controlaremos, en este caso, el tipo de numeración y otros aspectos que ya conocemos, como el padding de cada elemento, el color, la familia tipográfica y su tamaño. Veamos el ejemplo:

```
ol{ list-style-type: decimal;
padding: 20px;
    font-family: arial;
    font-size: 15px;
    color: olive;}
ol li{ padding-left: 20px;
}
```

Cuando trabajamos con listas, es fundamental cerrar tanto la lista como cada uno de los elementos `<li>`. Si olvidamos hacerlo, los sitios no validarán su código HTML y, además, tendremos problemas con el código CSS. Por ejemplo, puede ocurrir que el estilo de nuestros elementos `<li>` se propague por todo el sitio si no cerramos esta etiqueta, o que la lista adopte otras posiciones con respecto a los elementos adyacentes si no la finalizamos.

1. Elemento número uno
2. Elemento número dos
3. Elemento número tres
4. Elemento número cuatro
5. Elemento número cinco

- Primer elemento de nuestra lista
- Segundo elemento de nuestra lista
- Tercero elemento de nuestra lista
- Cuarto elemento de nuestra lista
- Quinto elemento de nuestra lista

- ✓ Primer elemento de nuestra lista
- ✓ Segundo elemento de nuestra lista
- ✓ Tercero elemento de nuestra lista
- ✓ Cuarto elemento de nuestra lista
- ✓ Quinto elemento de nuestra lista

En primer lugar, encontramos una lista ordenada con estilo decimal. A continuación, en una lista no ordenada utilizamos un cuadrado como bullet y, finalmente, una imagen. Aquí podemos apreciar cómo se vería en un navegador la lista ordenada con estilo decimal creada con el primer código.

LISTAS NO ORDENADAS

Las listas no ordenadas tienen la misma estructura de las anteriores, excepto que las etiquetas que las encierran son `<ul>` y `</ul>` (**unordered list**). Estas listas se diferencian de las ordenadas porque, en vez de identificar los elementos a través de números o letras, en general se emplean bullets o imágenes. Entre los diferentes tipos de bullets que existen, es conveniente elegir alguno que se adecue bien a nuestro diseño o crear los propios. Para una lista no ordenada, el código CSS que utiliza un cuadrado como bullet es el siguiente:

```
ul {
    list-style-type:
    square;
    font-family: arial,
    helvetica, verdana, sans-ser-
    if;
    font-size: 15px;
    color: #4b0082;
}
```

ES IMPORTANTE CERRAR LAS ETIQUETAS DE LA LISTA Y DE CADA UNO DE LOS ELEMENTOS `<li>`.



El uso de este tipo de listas es muy frecuente en sitios de enciclopedias o diccionarios, como www.wordreference.com.

+ MENÚS CON LISTAS

Aunque más adelante nos enfocaremos específicamente en los menús de navegación que se generan a partir del uso de listas, a continuación podemos ver un ejemplo de ese uso, en el que debemos prestar atención al lugar donde se colocan la etiqueta `<a>` y su cierre. Dentro de los tags `<ul>` y `</ul>`, el código correspondiente a cada elemento sería: `<li><a ref="http://paginaejemplo.com.ar">Primer elemento de nuestra lista</a></li>`.

LISTAS ANIDADAS

Usamos esta denominación para definir una lista que está dentro de otra. Puede aplicarse a listas tanto ordenadas como no ordenadas, de modo que podemos crear una combinación de ambas. Lo que obtenemos de esta forma es una lista dentro de otra: una contenedora, que es la principal; y otra secundaria, que estará indentada con respecto a la primera. En este tipo de listas, es fundamental cerrar y abrir las etiquetas donde corresponde, para así lograr la estructura correcta. Veamos un ejemplo de lista anidada:

```
<ul> <li>Letras del alfabeto
<ol> <li>Letra A</li>
<li>Letra B</li>
<li>Letra D</li>
</ol> </li>
<li>Colores</li>
<li>Números</li>
<li>Otros</li>
</ul>
```

WordReference.com

Pura Suavidad. Nuevo Milka Soft.

Entra y muestra que tus amigos son un poquito Milka.

Look up: Spanish-English Buscar

crujido: Definición | Sinónimos | Conjugación | En Francés | En Portugués | Imágenes

Del verbo **crujir** (conjugación)

crujido es:

- el participio
- Múltiple Entrées:
- crujido crujir

Power Online Spanish Dictionary © 2005 Online University Press

crujido sustantivo masculino

- a. (de fabrica, muebles, ramas) creaking
- b. (de papel, hojas secas) rustling
- (de seda) rustle
- c. (de los nudillos, las rodillas) cracking
- d. (de la grana, maula) crunching
- e. (de los dientes) grinding

crujir (conjugación) verbo intransitivo

- i. (table/muebles/ramas) to creak
- ii. (seda) to rustle

Report an inappropriate ad.

Analisis Google

Link:

- Abbreviations
- Pron. Symbols
- Conjugation

¿DÓNDE UTILIZAMOS LISTAS?

Seguramente, pensamos que las listas sólo se utilizan para enumerar elementos, pero en la actualidad, sus funciones son mucho más variadas. Descubramos, entonces, diferentes usos de listas en los sitios web.

Cuando pensamos en listas, tal vez lo primero que nos viene a la mente es una sucesión de palabras sin formato, una debajo de la otra. Aunque pensemos que las listas no se usan demasiado o que no son de utilidad, son elementos muy empleados y totalmente controlables a través de CSS. A continuación, veremos ejemplos de sitios que utilizan listas y estudiaremos cómo las implementan.

LISTAS ESTÁNDAR

Comencemos por la tradicional lista con un bullet circular. Podemos verla en www.mycurli.com, donde se la utiliza para hacer un punteo de texto principal y, además, en el footer, donde conforma tres columnas para enumerar los diferentes links.

LISTAS CON IMÁGENES

Es muy frecuente utilizar listas para los links de navegación de un sitio. Como ejemplo, en www.redvelvetart.com notamos que el menú principal está compuesto por un listado de links. Cada `<li>` tiene, en su interior, un elemento `<a>` al que se le aplica la imagen icónica mediante CSS. En este caso, en una sola imagen encontramos los estados del botón, es decir, el botón en descanso y el rollover, que también se define por CSS mediante la propiedad `[hover]`. De la misma manera se tratan los otros menús, con excepción de los tags en el ángulo inferior izquierdo, en donde no se utilizan imágenes sino fuentes. Hallamos otro ejemplo de este tipo en el sitio de Adobe (www.adobe.com/es), donde podemos ver dos columnas de

listas, cada una con una miniatura que representa el programa al que se hace mención. Luego hay un listado de noticias, cada una de las cuales representa un link hacia su ampliación. Por debajo, aparece un menú horizontal; en él, cada link está flotado hacia la izquierda, con

- Startseite
- Merkmale
- Gebrauchshinweise - Tips
- Blog
- Händlerkonto eröffnen
- curli Worldwide
- Onlineshop Schweiz
- Onlineshop Deutschland
- Onlineshop Österreich
- Trade
- Presse & Media
- Events - Ausstellungen
- Links
- Kontakt
- Impressum

www.revistaohlala.com,
www.mycurli.com y
www.storypixel.com son excelentes
ejemplos del correcto uso de las listas.

margin y padding configurados para generar la separación entre ellos. En **www.storypixel.com** podemos ver listas en varios lugares. Primero tenemos el menú principal, y luego aparecen las categorías, a las cuales, mediante CSS, se les aplica una miscelánea como background de etiqueta, de la siguiente manera:

```
#nav_categories li {
background:url("http://
s3.amazonaws.com/
creativeharvest/
blt-tag-a.gif") no-repeat
left 6px transparent;
}
```

En este ejemplo vemos el uso de la propiedad shorthand para definir, en una misma línea, el uso de una imagen como background, la asignación de no-repeat para que la imagen sea una sola (y no obtener un mosaico como resultado), una posición en X y en Y, y el color de fondo transparente.

Por otro lado, en el menú **[My Diet]**, cada link está compuesto por una imagen de fondo, que se utiliza mediante un código

donde se define la clase del link, se aplica una imagen como background y se establece su altura:

```
#my-diet-list
li.item-flash {
background:url("http://
s3.amazonaws.com/
creativeharvest/
my-diet-flash.gif")
no-repeat scroll left top
transparent;
height:80px;
}
```

LISTAS EN MENÚ DE NAVEGACIÓN

En **www.revistaohlala.com** tenemos un claro ejemplo de cómo usar listas para realizar un menú. En esta estructura, el primero y el último link se controlan con CSS de manera separada del resto, es decir, aplicándoles clases diferentes:

```
<ul> <li class="first"><a
href="">Link1</a></li>
<li><a
href="">Link2</a></li>
<li><a
href="">Link3</a></li>
<li><a href="">...</
a></li>
<li class="tab-
sedal"><a href="">Link4</
a></li>
</ul>
```

Las distintas longitudes de cada botón están determinadas por el largo de la palabra. Cada **** tiene un margin-left, que si fuera utilizado para todos los links, se aplicaría también al primero, y entonces éste no quedaría alineado. Por otro lado, el último link posee una clase especial donde se aplican una imagen y un color diferente como background. El resto de los botones posee el mismo código CSS, que es el siguiente:

```
ul#nav li {
background-
color:#7F7F7F;
float:left;
height:19px;
margin-left:2px;
padding:6px 25px;
text-align:center;
}
```

En **www.zaum.co.uk** primero vemos listas con ****, con un elemento debajo del otro; luego, tanto en el menú como en el footer, bajo los títulos **[Who]**, **[What]**, **[Where]** y **[Why]**, encontramos listas con **** flotados hacia la izquierda. En el menú tenemos el siguiente código, que muestra la sección seleccionada con un borde blanco debajo: ◀

```
#top-nav ul li a.active {
border-bottom:3px solid
#FFFFFF;
}
```



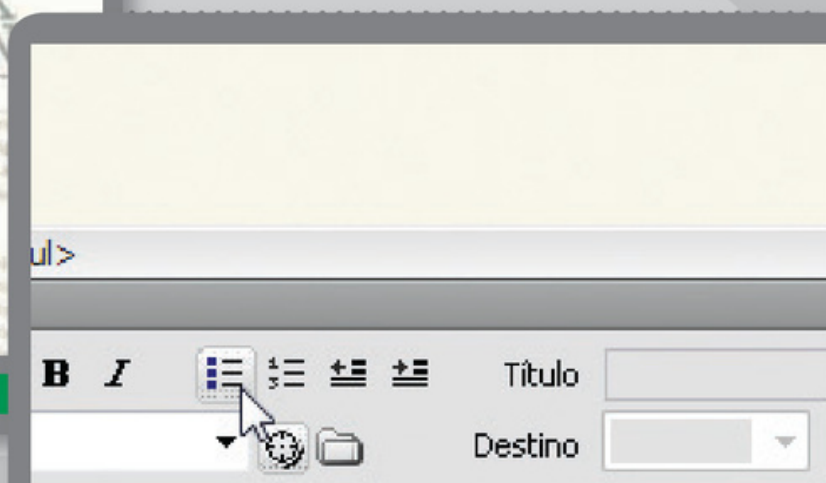
PERSONALIZAR

PASO A PASO

En este apartado, explicaremos las diferentes formas de personalizar las listas en pocos pasos.



01 | PÁRRAFOS SUELTOS



LISTA SIN ORDENAR | 02

Las listas son una herramienta muy utilizada a la hora de diseñar un sitio. Como ya vimos, nos permiten enumerar puntos que queremos que el visitante tenga en cuenta, pero también nos dan la posibilidad de realizar diversos tipos de botonerías. ◀

PASO | 01 Luego de crear un documento HTML y un archivo de hoja de estilos llamado **[lista.css]**, escribimos la cantidad de párrafos que queremos convertir en lista. Las listas no necesariamente deben ser extensas. Aquí, los párrafos que transformaremos se encuentran por debajo de **[SERVICES]**, dentro de un div con una clase denominada **[nav]**.

PASO | 02 Seleccionamos todos los párrafos que queremos convertir en listas. Luego, nos dirigimos a la barra **[Propiedades]** y hacemos clic en **[Lista sin ordenar]**. Observaremos cómo cambian el código y el aspecto.

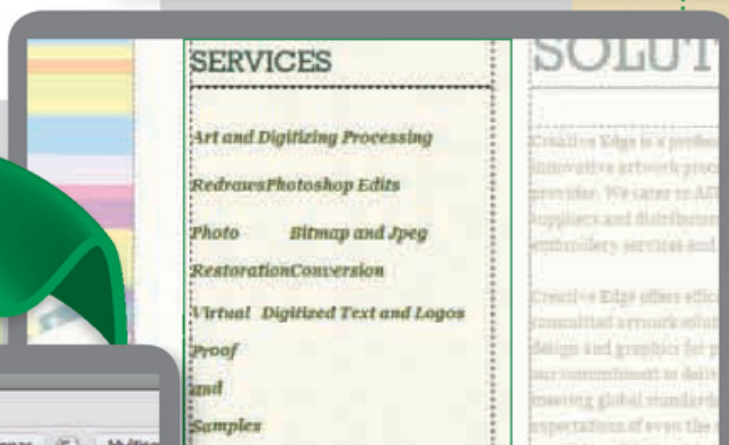
PASO | 03 Para comenzar a darle propiedades a la lista, vamos al archivo **[lista.css]** y, dentro de él, escribimos **[.nav ul { color: #36511a; font: normal 12px "Georgia",**



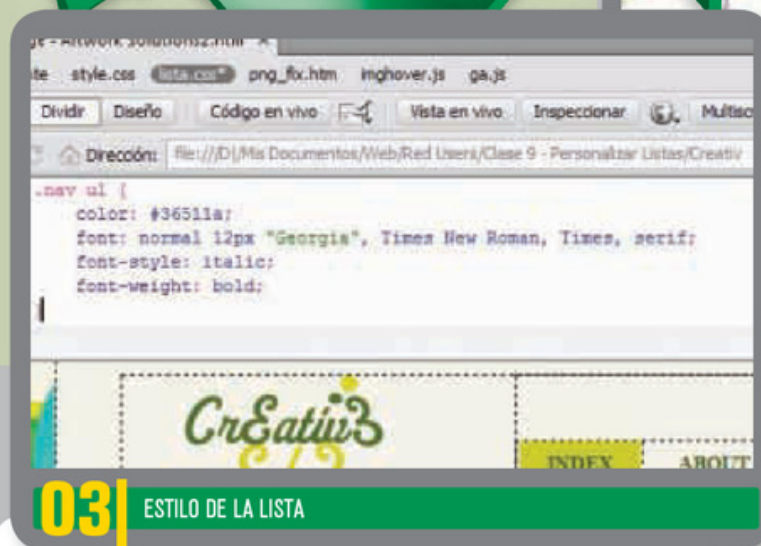
LISTAS



04 | ESTILO DE ELEMENTOS



ESPACIO ENTRE LÍNEAS 05



03 | ESTILO DE LA LISTA

Times New Roman, Times, serif; font-style: italic; font-weight: bold; }}. Aquí podemos ver aplicadas las propiedades que le daremos a la lista desordenada a nivel general.

PASO | 04 Ahora, vamos a mejorar la presentación de las etiquetas . Para hacerlo, a la hoja de estilos le agregamos el texto `[.nav li { color: #36511a; float: left; padding: 4px 0px 4px 0px; }]` para que las etiquetas se ubiquen una al lado de la otra, que tengan un color determinado y una distancia particular entre ellas.

PASO | 05 Para aumentar el espacio entre elementos, agregamos en el código anterior un alto a la etiqueta , mediante la aplicación de una altura en cada línea. El código es el siguiente: `[.nav li { color: #36511a; height: 29px; float: left; line-height: 29px; padding: 4px 0px 4px 0px; }]`.

PASO | 06

Para finalizar, a la etiqueta le damos un borde y un ancho, para así tener la lista diseñada y ordenada como deseamos. El código semántico debe quedar así: `[.nav li { border-bottom: dotted #000000 1px; color: #36511a; height: 29px; float: left; line-height: 29px; padding: 4px 0px 4px 0px; width: 231px; }]`.

RESUMEN

Como dijimos al comienzo de este capítulo, las listas son una herramienta muy útil cuando necesitamos enumerar cierta información. También podemos observar que las listas poseen múltiples aplicaciones, ya que nos permiten generar navbars e, incluso, pueden usarse para ciertas estructuras en reemplazo de las tablas.

LÍNEAS SEPARADAS 06

CREACIÓN DE MENÚS DE NAVEGACIÓN



Gracias a las listas, podemos generar diversos menús de navegación que tengan distintas propiedades y que podamos ubicar en el sector del sitio que nos parezca más adecuado.

La creación de menús de navegación mediante **listas** es una de las prácticas más comunes en la actualidad. Como ya vimos, cuando generamos una lista desordenada, por defecto el navegador muestra la viñeta bullet, en tanto que si creamos una lista ordenada, la viñeta es decimal. Ya sabemos que una de las ventajas

de utilizar reglas CSS es que podemos cambiar las características de las listas y controlar sus propiedades básicas, que son **[List-style-type]** para definir el estilo de la viñeta en la lista, **[List-style-image]** para cambiar la imagen de la viñeta y **[List-style-position]** para ubicar la viñeta por fuera del texto (outside) o por dentro de él (inside).

La modificación de estas propiedades permite obtener una gran variedad de resultados, ajustando el menú al diseño que hayamos realizado. Para hacerlo de forma simple, CSS nos brinda el shorthand **[list-style]**, cuyo objetivo es escribir menos, dado que permite aplicar todas las propiedades de la lista en un formato abreviado.

- elemento
- elemento

- elemento
- elemento

- elemento
- elemento

- elemento
- elemento

- elemento
- elemento

- elemento
- elemento

- elemento
- elemento

- elemento
- elemento

- elemento
- elemento

- elemento
- elemento



Aquí podemos observar ejemplos de los distintos tipos de viñetas que es posible aplicar a las listas.

CÓDIGO 1

```
#botonera ul li a {
height: 50px;
text-decoration: none;
width: 142px;
}
.quienessomos { background:
url(..../imagenes/jpg/
botonera/quienessomos.jpg)
center top;}
.granjaeducativa {
background: url(..../
imagenes/jpg/botonera/
granjaeducativa.jpg)
center top;}
...
```

LA CREACIÓN DE MENÚS DE NAVEGACIÓN MEDIANTE LISTAS ES UNA DE LAS PRÁCTICAS MÁS COMUNES EN LA ACTUALIDAD.

CREACIÓN DE UN MENÚ DE NAVEGACIÓN VERTICAL

Como mencionamos, las listas, además de cumplir su función natural, se aplican para la creación de navbars verticales u horizontales. Para este caso, lo primero que debemos hacer es definir la etiqueta `<ul>` dentro del código CSS, quitándole el estilo a la lista; el código es **#botonera ul { list-style: none; }**. Luego, definimos la etiqueta `<li>` mediante el código **#botonera ul li { font: normal 12px/12px Arial, Helvetica, Sans-serif; }**, que permite elegir la tipografía que se usará si las imágenes no se cargan. Para finalizar, definimos la etiqueta `<a>` y armamos las clases para cada botón, repitiendo la creación de la clase y aplicando las propiedades del background de acuerdo con la cantidad de botones que tengamos (Código 1).

Dentro de nuestras listas, hemos escrito el nombre de cada botón. Cuando damos a esos botones las propiedades con reglas CSS, notaremos que ese primer texto (que se utiliza cuando se ejecutan navegadores que no muestran los estilos) se superpone con el estilo. Por este motivo, vamos a agregar una etiqueta `[span]` de modo que el texto se vuelva invisible. Además, asignaremos las propiedades en el estado **hover** del botón:

```
#botonera ul li { float:
left; } #botonera ul li a {
float: left;
height: 50px;
text-decoration:
none; width: 142px;
}
```

CREACIÓN DE UN MENÚ DE NAVEGACIÓN HORIZONTAL

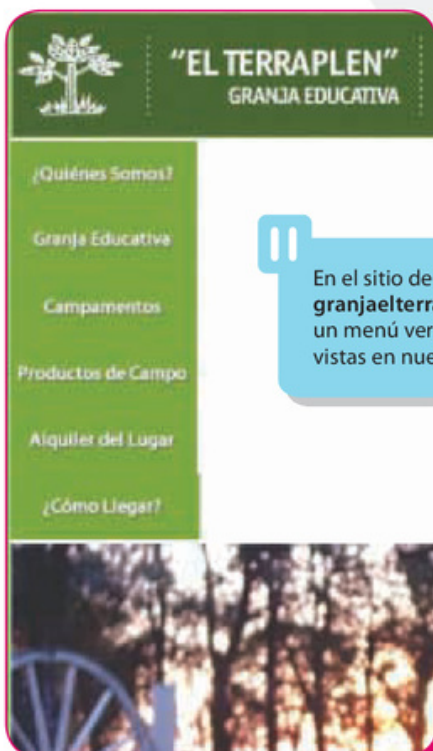
A partir de cualquier menú vertical, es posible crear uno horizontal. Para lograrlo, el cambio más importante lo generamos en la etiqueta `<a>`, aunque también pueden hacerse modificaciones en `<li>` o en ambas. Esto último es recomendable debido a que algunas versiones de IE mostrarán y cumplirán los cambios de propiedades dependiendo de dónde se encuentren éstos. Entonces, tanto en la etiqueta `<li>` como en `<a>`, agregamos la propiedad **float: left**. Las etiquetas deberían quedarnos así:

```
#botonera ul li a span {
text-decoration: none;
visibility: hidden;
} #botonera ul li a: hover {
background-position: center
bottom;
}
```

Además de los menús verticales y horizontales básicos, si utilizamos las reglas CSS para las pseudo-clases **hover** y **active**, podríamos crear navbars avanzadas que tuviesen pestañas, sombras, tipografías no estándar e, incluso, submenús internos que funcionen y se desplieguen (esto con la ayuda del lenguaje JavaScript). Recordemos que las botoneras o navbars deben estar contenidas dentro de un `div` o, si es en HTML5, dentro de la etiqueta `[nav]`. ◀

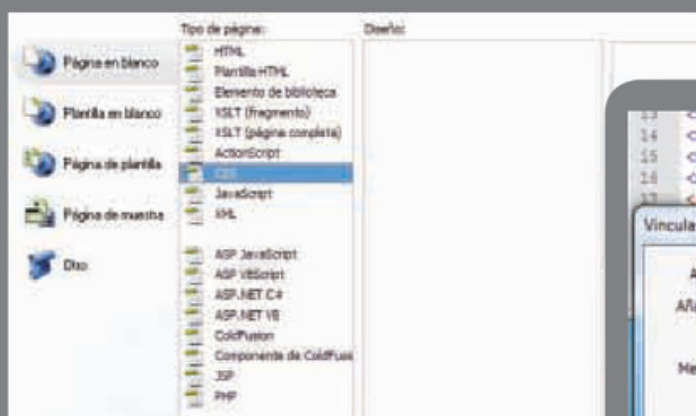
En el sitio de El Terraplén (www.granjaeltraplen.com.ar), encontramos un menú vertical creado con las propiedades vistas en nuestro ejemplo.

En esta imagen se muestra el menú vertical convertido en horizontal, gracias a las propiedades que se han colocado más arriba.

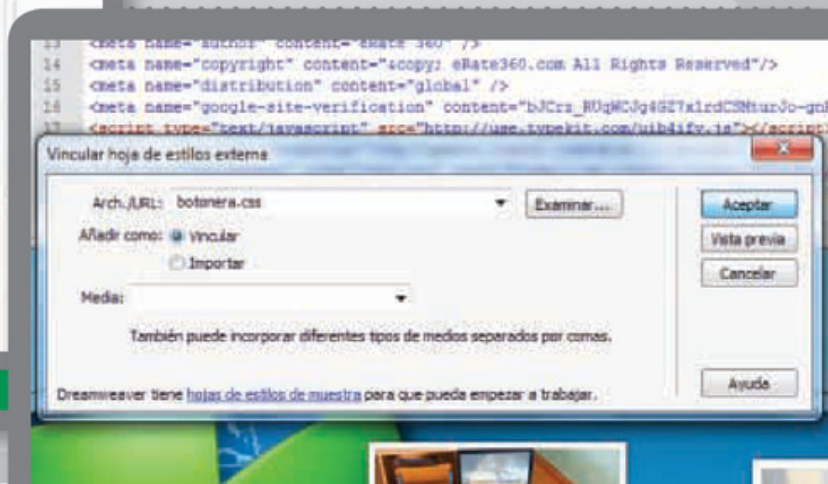


CREAR UN MENÚ DE PASO A PASO

En este apartado, explicaremos cómo crear un menú de navegación con listas.



01 CREAR CSS



HTML Y CSS **02**

Como vimos hasta aquí, en la actualidad las listas son fundamentales para la creación de navbars, también llamadas menús de navegación. Aunque más adelante veremos cómo generar el menú de nuestro sitio, aquí exploraremos la creación y la manipulación de listas para obtener un menú un poco más complejo y con más elementos. Esto nos ayudará a adquirir más conocimientos, para luego adecuarlos y aplicarlos en cualquier sitio. ▶

PASO | 01 Lo primero que haremos será crear un archivo CSS que contendrá nuestro menú de navegación. Para hacerlo, vamos a **[Archivo/Nuevo...]**, elegimos **[CSS]** en **[Tipo de página]** y presionamos **[Crear]**. Luego, guardamos el archivo como **botonera.css**.

PASO | 02 Creamos un documento HTML, lo guardamos y, luego, nos dirigimos al panel derecho **[Estilos CSS]** y hacemos clic en **[Adjuntar hoja de estilo]**. Cuando se abre **[Adjuntar hoja de estilos externa]**, presionamos **[Examinar...]**, buscamos el archivo **botonera.css** y pulsamos **[Aceptar]**. Finalmente, volvemos a hacer clic en **[Aceptar]**.

PASO | 03 Desde el panel **[Insertar]**, creamos un div llamado **[botonera]** y en él escribimos el texto de los botones que formarán la

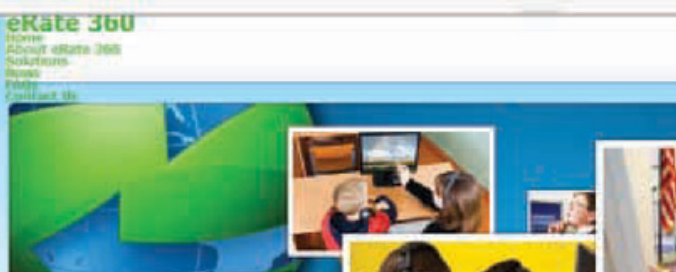
NAVEGACIÓN

04 | ENLACES EN LA LISTA

```

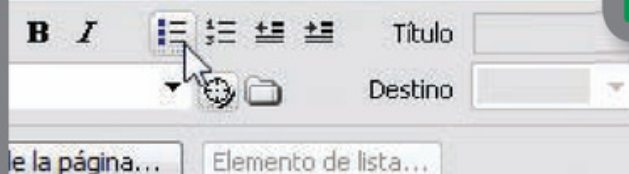


```



ESTILO BOTONERA | 05

A Compliance Team of



03 | TEXTO PRINCIPAL

navbar. En este caso, además, incluimos una línea vacía al comienzo y otra al final del texto. Seleccionamos los elementos y, en la barra inferior [Propiedades], presionamos [Lista sin ordenar].

PASO | 04 Ahora, terminaremos de definir la lista desordenada de nuestro div **botonera** para luego dedicarnos a las reglas CSS.

Dentro de las etiquetas `<li>`, colocamos cada enlace mediante etiquetas `<a>`, excepto en el logo del sitio, al que previamente le ponemos una etiqueta `<h1>`. Además, colocamos una etiqueta `[class="izquierdo"]` en el espacio en blanco inicial, una `[class="derecho"]` en el espacio en blanco final y `[class="sans-divider"]` en la que tiene el logo o marca.

PASO | 05 Vamos al archivo **botonera.css** y allí escribimos `[div#botonera {display: block; margin: 0 auto; padding: 0 0 0 2px; width: 988px;}]` para definir las características del div creado

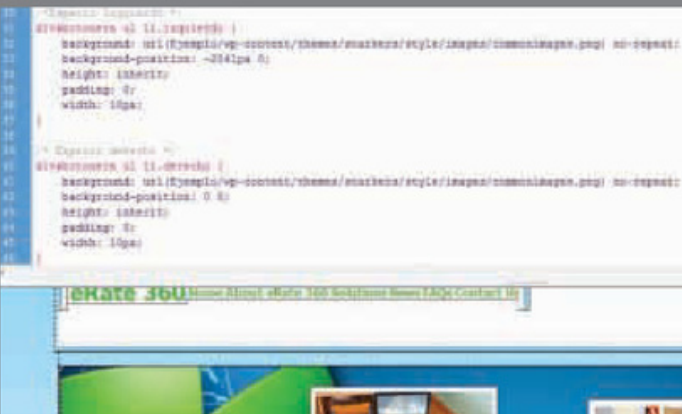
en el paso anterior. Además, definimos las propiedades de la etiqueta `<ul>` escribiendo `[div#botonera ul {background: url(Ejemplo/wp-content/themes/starkers/style/images/tile_menu.jpg) repeat; height: 67px; list-style: none; margin: 0; padding: 0;}]`, donde la URL de la imagen corresponde a la ubicación que le hayamos dado al fondo del menú.

PASO | 06 Definimos la etiqueta `<li>` de la botonera con `[div#botonera ul li {background: url(Ejemplo/wp-content/themes/starkers/style/images/commonimages.png) no-repeat; background-position: -2047px -67px; display: block; float: left; height: inherit; padding: 0 0 0 3px; position: relative;}]`.

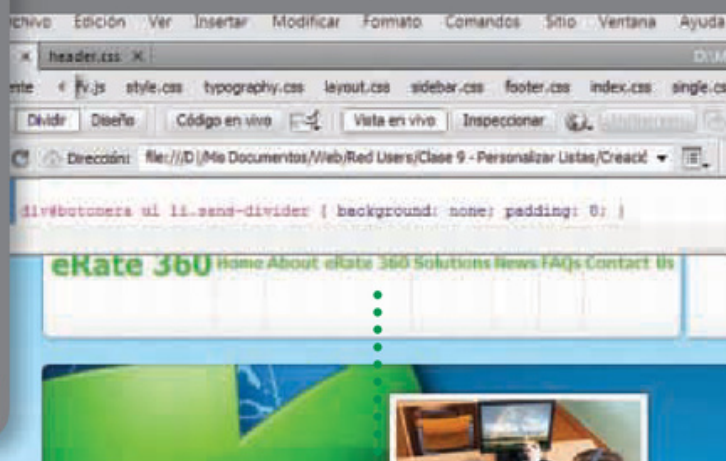


CREAR UN MENÚ DE NAVEGACIÓN

PASO A PASO

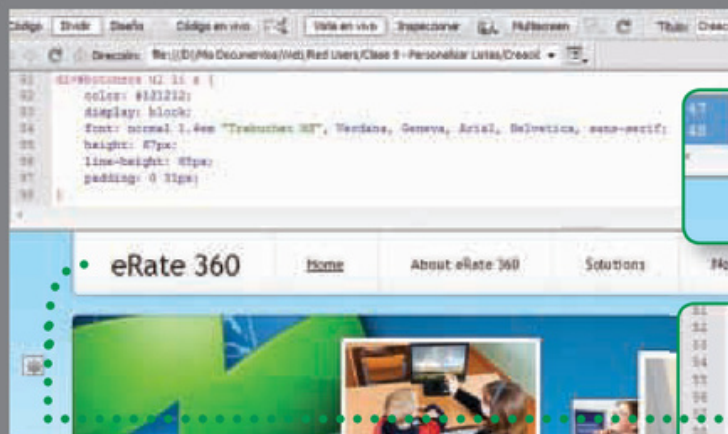


07 | DEFINIR LOS EXTREMOS



ESTILO DE LOS EXTREMOS

08



09 | ESTILO DE ENLACES



PASO | 07

Para definir los espacios en blanco que tenemos a los lados de la botonera, en el CSS escribimos

el código `[div#botonera ul li.izquierdo {background: url(Ejemplo/wp-content/themes/starkers/style/images/commonimages.png) no-repeat; background-position: -2041px 0; height: inherit; padding: 0; width: 10px;}]`, para el lado izquierdo; y `[div#botonera ul li.derecho {background: url(Ejemplo/wp-content/themes/starkers/style/images/commonimages.png) no-repeat; background-position: 0 0; height: inherit; padding: 0; width: 10px;}]` para el lado derecho.

Además, agregamos al código HTML las clases que corresponden ([.izquierdo] y [.derecho]).

PASO | 08

Ahora, definimos la clase `[.sans-divider]` con el código `[div#botonera ul li.sans-divider {background: none; padding: 0;}]`.

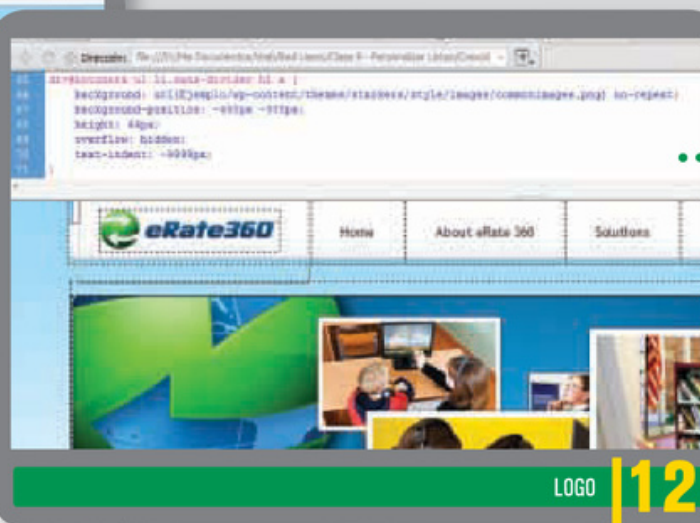
PASO | 09

La etiqueta `<a>` debe ser definida con `[div#botonera ul li.sans-divider a {background: none; padding: 0;}]` y `[div#botonera ul li a {color: #121212; display: block; font: normal 1.4em "Trebuchet MS", Verdana, Geneva, Arial, Helvetica, sans-serif; height: 67px; line-height: 65px; padding: 0 31px;}]`.

PASO | 10

Luego de aplicar la etiqueta `<a>`, definimos todos sus estados escribiendo en el archivo `botonera.`

css lo siguiente: `[div#botonera ul li a:hover {background: url(Ejemplo/wp-content/themes/starkers/style/images/tile_menuhover.jpg) repeat; text-decoration: none;}]`. Con este código, modificamos todos los botones en su estado `hover`.



PASO | 11

Lo próximo que definiremos será el tamaño que deberá tener la etiqueta `<li>` de la marca o logo. En este caso, escribimos

```
[div#botonera ul li.sans-divider h1{height: 67px; padding: 10px 31px 10px 22px; width: 191px;}]
```

PASO | 12

Vamos a definir ahora las propiedades de la etiqueta `<h1>` cuando tiene en su interior una etiqueta `<a>` mediante

```
[div#botonera ul li.sans-divider h1 a {background: url(Ejemplo/wp-content/themes/starkers/style/images/commonimages.png) no-repeat; background-position: -450px -300px; height: 44px; overflow: hidden; text-indent: -9999px;}]
```

PASO | 13

Una vez que hemos realizado los pasos anteriores, podemos decir que tenemos nuestra primera botonera o menú armado.

Hacemos clic en [Vista en vivo] para ver su funcionamiento.

RESUMEN

Hemos creado de esta forma nuestro primer menú con listas, para conocer mejor uno de los múltiples usos que tiene este elemento. Es preciso tener en cuenta que los menús o botoneras deben ser diseñados previamente y que, antes de comenzar a maquetar, hay que tener en claro lo que queremos obtener, ya que de eso dependerá la forma en la que asignemos las propiedades.

MENÚ DESPLEGABLE

PASO A PASO

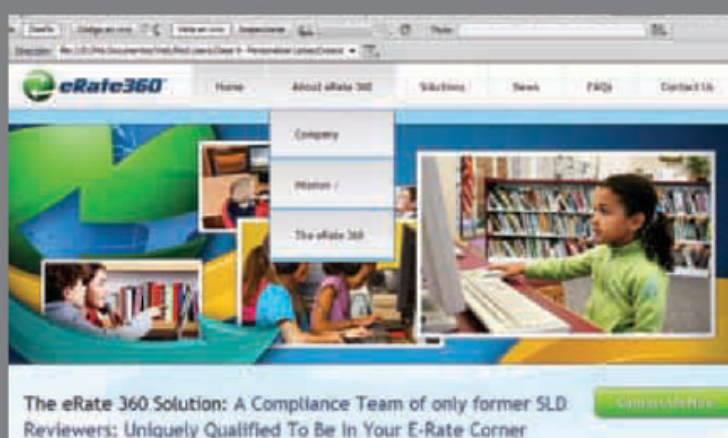
En este apartado explicaremos cómo crear un menú desplegable utilizando listas.



01 | LISTA ANIDADA



ESTILOS NUEVA LISTA | 02



03 | CREACIÓN DEL SUBMENÚ

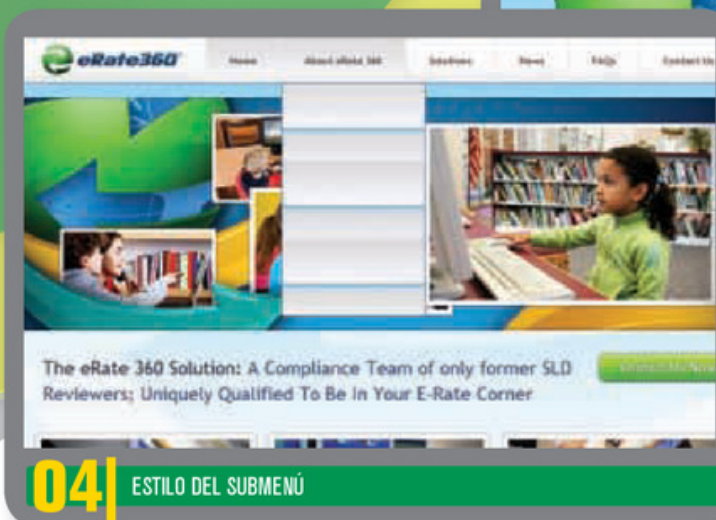
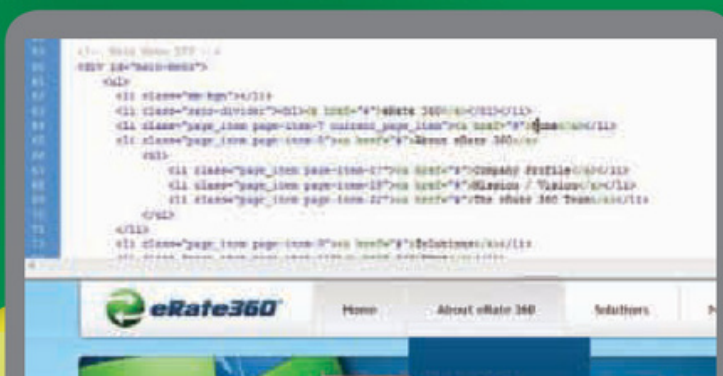
PASO | 01 Lo primero que haremos será seleccionar uno de los botones antes generados y colocar, dentro de su etiqueta `<li>`, una lista desordenada con sus respectivas etiquetas `<li>`. En este caso, lo haremos dentro de **[About eRate 360]**.

PASO | 02 Ahora, a cada una de las nuevas etiquetas `<li>` le otorgaremos una clase y colocaremos en su interior la etiqueta `<a>`. Esto hará que hereden ciertas propiedades que ya hemos configurado previamente.

PASO | 03 Para generar un bloque para el submenú, aplicamos nuevas propiedades a la etiqueta `<li>` en su posición **hover**: `[div#botonera ul li: hover ul {display: block;}]`. Luego, generamos las propiedades de la etiqueta `<ul>` del submenú mediante `[div#botonera ul li ul {background: url(Ejemplo/wp-content/`

En esta oportunidad explicaremos, en seis pasos, cómo generar un menú desplegable con listas y CSS. Para hacerlo, nos basaremos en el menú de navegación creado anteriormente. ◀

CON CSS Y LISTAS



04 | ESTILO DEL SUBMENÚ

themes/starkers/style/images/tile_dropdownbg.jpg)repeat-x; background-color: #004971; border: 1px solid #005c8f; display: none; height: 0; height: auto !important; max-width: 454px; position: absolute; top: 62px; z-index: 1;}}

PASO | 04 Para diferenciar estéticamente el submenú del menú principal, escribimos [div#botonera ul li ul li {background: none; border-bottom: 1px solid #FFFFFF; clear: left; float: left; padding: 0; position: relative; width: 100%;}]. Luego, otorgamos las propiedades a la etiqueta <a> del submenú con [div#botonera ul li ul li a {color: #FFFFFF; font-weight: bold; font-size: 1.2em; height: 100%; padding: 20px; width: 154px;}]

PASO | 05 Ahora, establecemos las propiedades del submenú en su estado **hover**: [div#botonera ul li ul li a: hover {background: none; color: #9df315; text-decoration: underline;}]

ESTADO HOVER **05**



SUBMENÚ DIFERENCIADO **06**

PASO | 06

Finalmente, eliminamos el fondo del submenú para que quede perfectamente diferenciado. Lo hacemos con el código [div#botonera ul li.current_page_ancestor ul li a, div#botonera ul li.current_page_item ul li a, div#botonera ul li.wax-on ul li a, div#botonera ul li: hover ul li a { background: none;}]

RESUMEN

Es importante aclarar que para que el desplegable funcione, debemos usar algún comportamiento de jQuery o JavaScript, que podemos conseguir fácilmente en Internet. Además, podemos cambiar todos los estilos de los submenús si así lo deseamos, de la misma forma en que generamos la primera lista.

ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 ¿Qué son las listas?

02 ¿Cuáles son las ventajas de utilizar listas?

03 ¿Cuáles son las diferencias entre las listas ordenadas y las no ordenadas?

04 ¿Dónde conviene utilizar listas?

05 ¿Cuál es la diferencia entre una lista estándar y una lista con imagen?

06 Defina los valores que toma la propiedad **[List-style-type]**.

07 ¿Qué son las listas anidadas?

08 ¿Cuáles son las características de las listas en menú de navegación?

09 ¿Cómo se define el estilo de la viñeta en la lista?

10 ¿Cómo se crea un menú horizontal a partir de uno vertical?



PRÁCTICAS

01 Genere una lista ordenada que identifique sus elementos con letras, a partir de la "a" minúscula.

02 Genere una lista no ordenada con un bullet creado por usted.

03 Busque en Internet ejemplos de listas anidadas.

04 Cree un menú de navegación horizontal a partir de uno vertical.

05 Cree un menú desplegable a partir de la lista realizada anteriormente.



Imágenes

- ▶ IMÁGENES EN HTML
- ▶ TIPO DE IMÁGENES
- ▶ OPTIMIZACIÓN DE IMÁGENES PARA LA WEB
- ▶ MAPA DE IMÁGENES
- ▶ PROPIEDADES DE LAS IMÁGENES EN CSS
- ▶ CSS SPRITE
- ▶ SPRITES EN CSS
- ▶ CSS SPRITES EN MENÚS



IMÁGENES EN HTML

En este capítulo veremos la implementación de uno de los elementos más importantes de las páginas web: las imágenes. Comenzaremos conociendo cómo insertarlas mediante atributos específicos y exclusivos.

Prácticamente, todos los sitios web contienen imágenes, en mayor o menor cantidad. Entre ellas, encontramos dos tipos: las de contenido y las de complemento o adorno.

Las imágenes de **contenido** son las que agregan información adicional a la información textual que tengamos en la página, como fotos, gráficos, ilustraciones, etcétera. Las incluiremos en el código HTML mediante la etiqueta `<img>`.

Las imágenes de **complemento o adorno** son las que utilizaremos para hacer, por ejemplo, cajas con bordes redondeados, fondos con degradé, iconos en listados o en links, etcétera. No deberíamos incluirlas en el código HTML, sino a través de hojas de estilo CSS. Esto responde a dos razones principales. En primer lugar, cuando queramos modificar el estilo del sitio, lo haremos directamente desde la hoja CSS. En segundo, les quitaremos tarea a los robots de los buscadores al indexar sólo las imágenes que son importantes para el sitio.

ATRIBUTOS OBLIGATORIOS DE LA ETIQUETA `<img>`

Como dijimos, la etiqueta que utilizaremos para incluir imágenes en el código HTML es `<img>`, que tiene dos atributos requeridos, u obligatorios, y cuatro opcionales.

El primer atributo obligatorio es **src**, que proviene de la abreviatura de source (fuente, en inglés). A través de él, le indicamos al navegador la URL de la imagen que queremos mostrar. Ésta puede ser absoluta o relativa, tal como vimos cuando conocimos las rutas. El segundo atributo obligatorio es **alt**, que se utiliza para describir el contenido de la imagen en un texto breve de hasta 1024 caracteres.

Veamos, a continuación, un ejemplo de uso de la etiqueta `<img>`:

```

```



ALT

Originalmente, el atributo **alt** se utilizaba como sustituto de la imagen en caso de que ésta no pudiera ser mostrada por diferentes razones (estar deshabilitada desde el navegador, error de conexión, etcétera). Un uso importante que le damos hoy es para ayudar a los buscadores a describir el contenido de la imagen, tarea fundamental a la hora de hacer SEO.

En este portfolio (www.ilustradordigital.com.br), encontramos ilustraciones y ejemplos de trabajos realizados.



ATRIBUTOS OPCIONALES

El atributo **longdesc** es muy poco usado y sirve para especificar dónde encontraremos más información sobre la imagen. Está pensado para complementar el límite de 1024 caracteres que tiene el atributo **alt**. Veamos un ejemplo:

```

```

El atributo **name** sirve para establecerle un nombre al elemento imagen, en tanto que **width** y **height** se aplican para definir el ancho y el alto con el que se muestran las imágenes en la página HTML. Estos atributos pueden parecer un tanto contradictorios porque, como hemos visto antes, todo lo referido al aspecto gráfico de la página debe definirse en la hoja de estilos CSS. Por lo tanto, en principio, el ancho y el alto de una imagen también deberían estar definidos allí. No obstante, es virtualmente imposible que CSS determine los anchos y los altos de todas las imágenes que tengamos en un sitio, ya que la hoja de estilos aumentaría su tamaño según la cantidad de imágenes presentes, lo que resultaría contraproducente. Por esta razón, los atributos **width** y **height** son la excepción a la regla de que el código HTML no debe hacer referencia al aspecto gráfico de los elementos. Si el valor de estos atributos es un **número entero**, el navegador de Internet interpretará que hacemos referencia a una medida en **píxeles**. A continuación, vemos una imagen con un ancho de 320 píxeles y un alto de 240 píxeles:

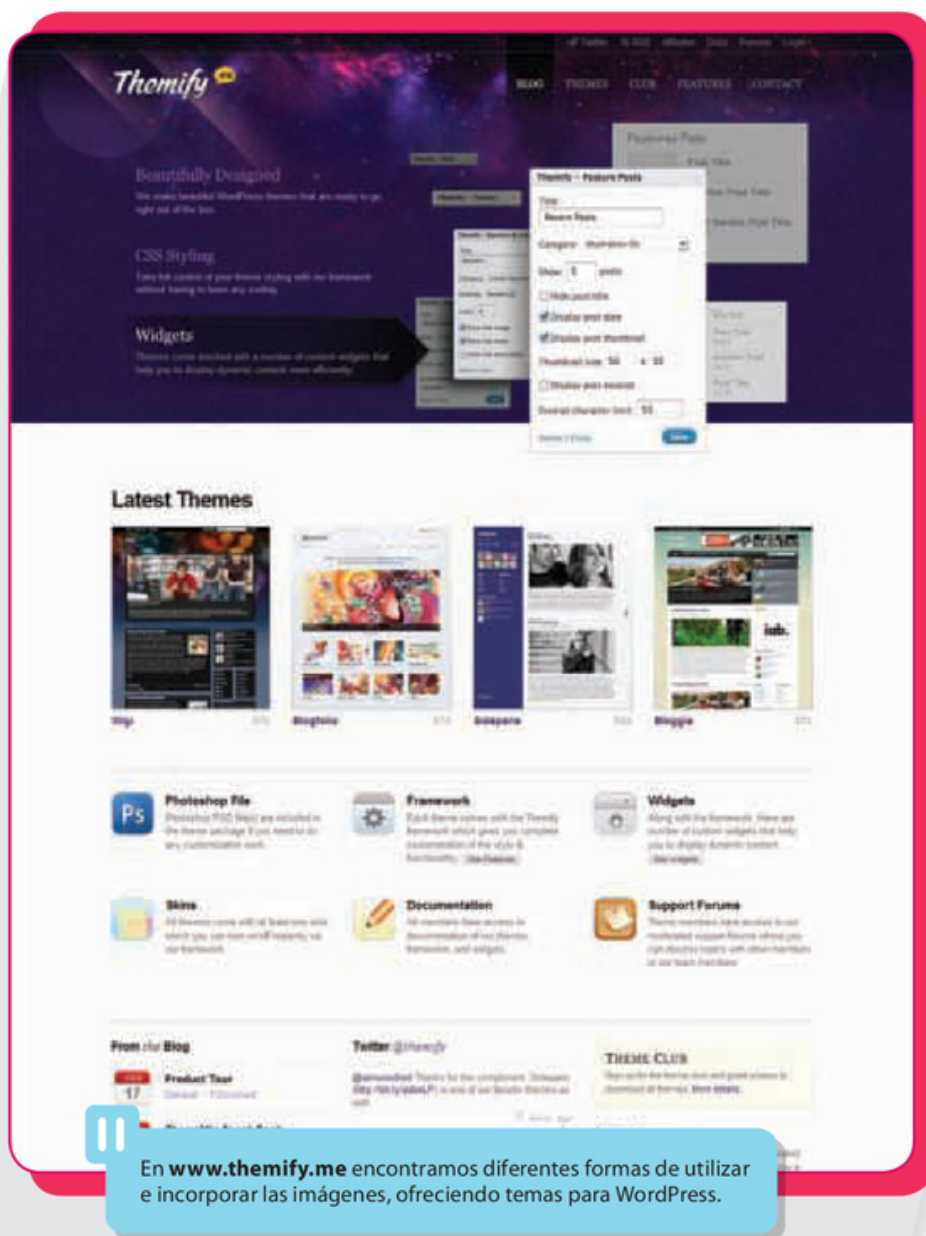
```

```

También es posible definir el ancho y el alto de una imagen con **medidas relativas**, indicando un **porcentaje**. En este ejemplo, mezclamos los dos tipos de medidas que podemos utilizar: el ancho aquí equivale a la mitad del ancho del elemento contenedor (en este caso, un div), y el alto será de 240 píxeles.

```
<div>

</div>
```



Los valores que definimos en los atributos **width** y **height** no tienen por qué coincidir con los valores de ancho y alto reales de la imagen. Sin embargo, si estos valores no coinciden, las imágenes se mostrarán deformadas y con un aspecto que lucirá desprolijo. En caso de que definamos sólo uno de los dos atributos, el navegador de Internet calculará el restante para que se mantenga la proporción original de la imagen. Si bien **** es una etiqueta HTML que no contiene cierre, para que la página XHTML sea válida, todas las etiquetas deben estar cerradas. Por lo tanto, para cerrar la etiqueta de imagen, utilizamos **</>**.

UNA PRÁCTICA RECOMENDADA
ES ALMACENAR
TODAS LAS IMÁGENES
EN UNA CARPETA
ESPECÍFICA PARA
ESE FIN, COMÚNMENTE
LLAMADA IMAGES.

TIPOS DE IMÁGENES



Es hora de conocer cuáles son los formatos de imágenes más comunes que utilizaremos en el desarrollo de los sitios por crear. Cada uno de ellos es ideal para un tipo de imagen; veamos sus características.

Los tres formatos de imágenes más comunes son GIF, JPG y PNG. No podemos decir que uno de ellos sea mejor que otro porque, en realidad, elegiremos el adecuado según la imagen que vayamos a insertar en el HTML.

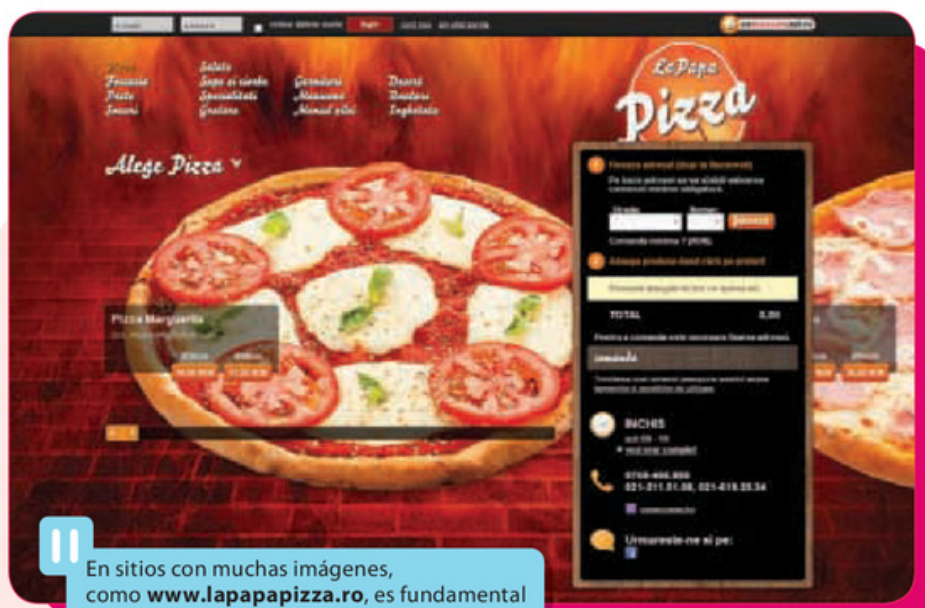
GIF

La sigla GIF proviene de *Graphics Interchange Format*, o formato de intercambio de gráficos. Este formato utiliza un máximo de 256 colores, por lo que no es adecuado para fotografías de mediana y alta resolución. En cambio, es muy usado para imágenes con grandes áreas de un mismo color, pequeños iconos o **bullets** e imágenes de complemento o **adorno**, como cajas con bordes redondeados y degradés cortos. La principal ventaja del formato GIF es que permite reemplazar un color por transparencia, dándonos mayor flexibilidad a la hora de aplicarlo sobre un fondo HTML. Además, en archivos de baja resolución, su ventaja sobre JPG es que, al no comprimirlos, conserva con mayor nitidez la imagen por mostrar.

JPG

También conocido como JPEG, por *Joint Photographic Experts Group*, o grupo de expertos en fotografía, este formato soporta hasta 16 millones de colores, de modo que resulta ideal para fotos o gráficos.

Tiene una muy buena relación entre el peso del archivo y la calidad de la imagen, lograda mediante su algoritmo de compresión, que reemplaza varios píxeles por un área del mismo tamaño,



En sitios con muchas imágenes, como www.lapapapizza.ro, es fundamental que todas ellas estén optimizadas.

con un color promedio. Cuanta mayor compresión de imagen tengamos, más calidad resignaremos, pero ganaremos en la reducción del peso del archivo. Una de las ventajas del formato JPG es lo que se conoce como **modo progresivo**, que los navegadores interpretan mostrando, en primer lugar, una versión en baja calidad de la imagen, para luego ir mejorándola a medida que se va cargando.

PNG

La sigla de este formato proviene de *Portable Network Graphics*, o gráficos portátiles de red. Este formato es muy usado, ya que las imágenes se comprimen sin pérdidas visibles de calidad. La principal ventaja de los PNG es que

admiten imágenes de 24 bits, incorporando un **canal alpha** que da la posibilidad de utilizar transparencias totales o parciales dentro del archivo. La desventaja es que no es compatible con todos los navegadores. En algunos más viejos (como Internet Explorer 6), al usar PNG veremos la imagen pero no su transparencia, que será reemplazada por un cuadro gris. Este formato también soporta el **modo progresivo** pero, además, tiene soporte para **metadata** (o metadatos), información adicional que poseen los archivos PNG para ser indexados por los motores de búsqueda. Esta información no es visible en la imagen, sino que está oculta dentro del código fuente del archivo. ◀

OPTIMIZACIÓN DE IMÁGENES PARA LA WEB

La preparación de las imágenes que utilizaremos en una página HTML es un proceso fundamental, que exige algunas precauciones para lograr que el sitio sea más ágil y rápido de cargar.

Los diseñadores gráficos están acostumbrados a trabajar con resoluciones de 300 dpi (*dots per inch*, puntos por pulgada). En diseño web, debemos darles importancia al ancho y al alto de las imágenes que utilizaremos, y no tanto a la resolución, ya que la de cualquier monitor es siempre de aproximadamente 72 dpi. Si utilizamos 300 dpi para una foto que incluiremos en una web, obtendremos imágenes demasiado pesadas y que no se verán mejor que si tuvieran el mínimo de resolución. Una imagen pesada ocupará mayor espacio en el servidor, generará más tráfico de datos, tardará más tiempo en cargarse y, probablemente, agote a los usuarios del sitio cuando tengan que esperar su carga.

El peso de un archivo de imagen dependerá, directamente, de la cantidad de colores, su tamaño y la resolución. Entonces, cualquier acción que recorte estos parámetros implicará una reducción en el peso del archivo y, por lo tanto, del tiempo de descarga necesario para visualizarla en cualquier navegador de Internet. Al guardar el archivo, debemos elegir el formato adecuado entre los tres más habituales para uso web: GIF, JPG y PNG.

RED, GREEN Y BLUE

Al trabajar con imágenes para la Web, no tiene sentido emplear el método CMYK utilizado en diseño gráfico para impresión. En cambio, optaremos por el modo de color RGB (Red, Green, Blue; rojo, verde y azul), método que establece un valor de 0 a 255 para cada uno de los componentes.

JPG VS. GIF

El formato JPG soporta 256 colores como mínimo y 16 millones como máximo, por lo que la reducción de peso por este lado es irrelevante. Lo que permite ajustar mejor el peso de un archivo JPG es la **compresión**. Al aplicarla, es aconsejable ir probando cuál es la mayor compresión que podemos obtener sin corromper visualmente la imagen. En cambio, para los archivos GIF, la cantidad de colores es importante, ya que

podemos tener 256 como máximo. En este caso, mediante la reducción de colores, podremos disminuir el peso del archivo. Muchas veces, encontraremos que, visualmente, no existe diferencia entre un archivo GIF con 256 colores y otro con 128, y de esa forma, podremos reducir el peso hasta en un 60%. Aunque Photoshop es una gran herramienta a la hora de optimizar imágenes para la Web, existen otras utilidades online y offline que permiten realizar esta tarea. Entre las opciones online, una muy recomendable es Web Resizer (www.webresizer.com), que se utiliza para reducir la resolución de una imagen (y, por consiguiente, su peso) como Web Service, sin tener que descargar nada. ◀



En www.vintagechiccannock.co.uk, las fotografías se presentan con un pequeño marco, y las ilustraciones son parte del diseño.



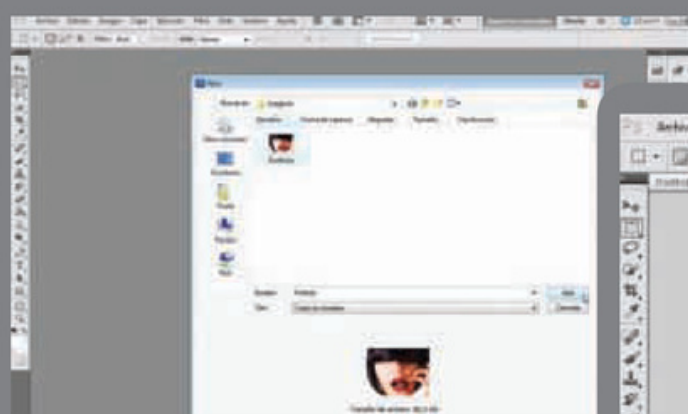
72 DPI

Cuando cambiemos el ancho y el alto de las imágenes por medio de algún programa de edición, deberemos asegurarnos de que, además del tamaño, estemos modificando la resolución para que ésta sea de 72 dpi. Siempre que trabajemos con archivos de imágenes, es aconsejable guardar una copia original que conserve el tamaño, la resolución y la cantidad de colores necesarios que nos permitan manipularla posteriormente sin limitaciones. Los formatos ideales para este fin son TIFF y BMP.

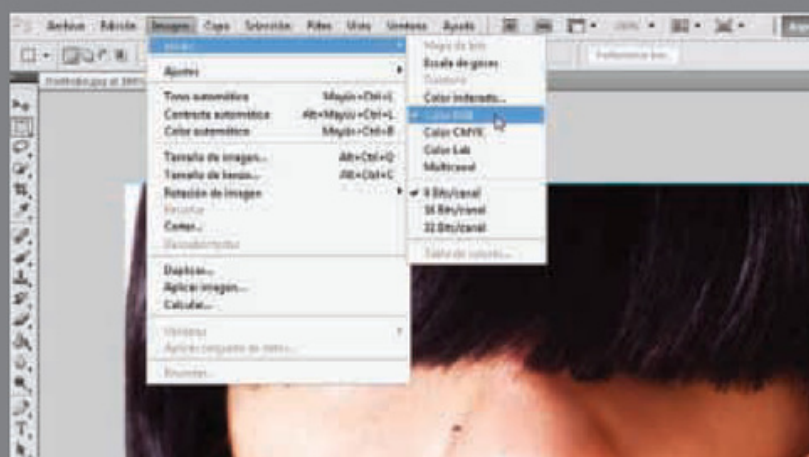
OPTIMIZACIÓN DE

PASO A PASO

En este apartado, explicaremos cómo optimizar las imágenes para poder utilizarlas en Internet.



01 | ABRIR ARCHIVO ORIGINAL



MODO DE COLOR | 02

La optimización de imágenes es fundamental para que la velocidad de acceso a un sitio sea adecuada. Durante el proceso, las fotografías deben perder la menor calidad posible, ya que si bien es preferible tener menor peso, no en todos los casos deberíamos aplicar la misma premisa. ◀

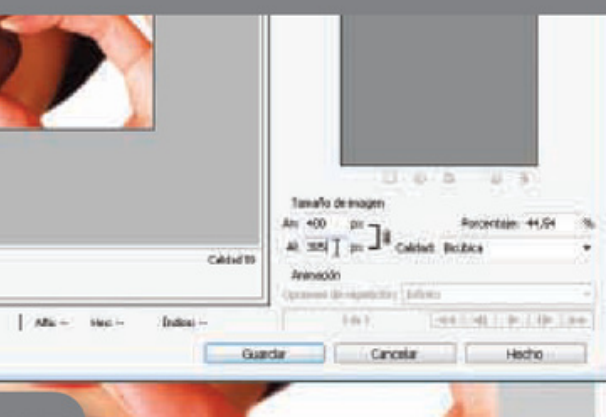
PASO | 01 Abrimos Photoshop CS5 y nos dirigimos al menú **[Archivo/Abrir...]**. En la ventana que aparece, buscamos la carpeta donde se encuentra la imagen que queremos optimizar, seleccionamos el archivo y presionamos el botón **[Abrir]**.

PASO | 02 Vamos al menú **[Imagen]** y, dentro de **[Modo]**, nos aseguramos de que estén marcadas las opciones **[Color RGB]** y **[8 Bits/canal]**, por ser éstos los estándares de color utilizados con los monitores.

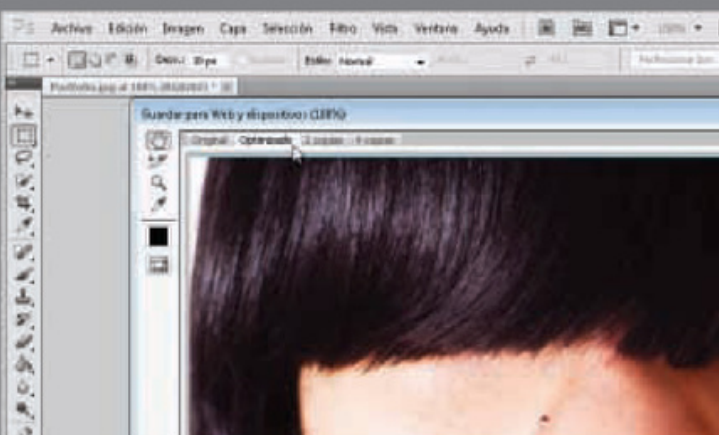
PASO | 03 Nos dirigimos al menú **[Archivo]** y hacemos clic en **[Guardar para Web y dispositivos...]**, o presionamos la combinación de teclas **<Alt + Shift + Ctrl + S>**. El panel que se abre nos dará la posibilidad de controlar todas las características de nuestro archivo optimizado. En primer lugar, verificamos que estamos ubicados en la solapa **[Optimizado]**, para observar el resultado; o hacemos clic sobre **[2 copias]**, para ver la imagen original y la mejorada al mismo tiempo.

IMÁGENES PARA LA WEB

04 | REDUCIR LA CALIDAD



CAMBIAR EL TAMAÑO | 05



03 | VENTANA DE OPTIMIZACIÓN

PASO | 04 A continuación, debemos elegir el tipo de archivo que queremos generar, que en este caso será JPEG, porque se trata de una fotografía. Luego, en la opción **[Calidad]**, elegimos un valor que mantenga la imagen sin que se vea pixelada. Debajo de la foto, podemos observar el peso que tendrá el archivo resultante.

PASO | 05 Si queremos reducir el tamaño de la imagen, en la parte inferior del panel podemos modificar el valor de **[Porcentaje]** o, directamente, ingresar un nuevo valor para ancho y alto en **[An]** y **[Al]**, respectivamente. Si la línea que conecta estos dos campos tiene un icono de cadena a la derecha, al modificar un valor, el otro cambiará automáticamente para mantener la proporción de la imagen. En cambio, si queremos deformar la fotografía, hacemos clic sobre la pequeña cadena y variamos la propiedad que deseamos.

PASO | 06 Finalmente, presionamos en el botón **[Guardar]** y, en la ventana que se abre, elegimos el lugar donde queremos almacenar el nuevo archivo. Escribimos el nombre que tendrá la imagen y pulsamos **[Guardar]**.

GUARDAR ARCHIVO OPTIMIZADO | 06

RESUMEN

En este caso, la imagen original pesaba 91 KB, y cuando terminamos el proceso de optimización, la nueva versión pesa sólo 19 KB. La pérdida de calidad fue mínima, por lo que podríamos decir que nuestro trabajo ha sido un éxito. Al optimizar, lo ideal es generar la menor pérdida posible de calidad. Si bajamos más allá del 70%, el resultado es riesgoso, aunque algunas imágenes soportan hasta un 40%. Esto depende mucho de la toma original y del uso que vayamos a darle a la fotografía optimizada.

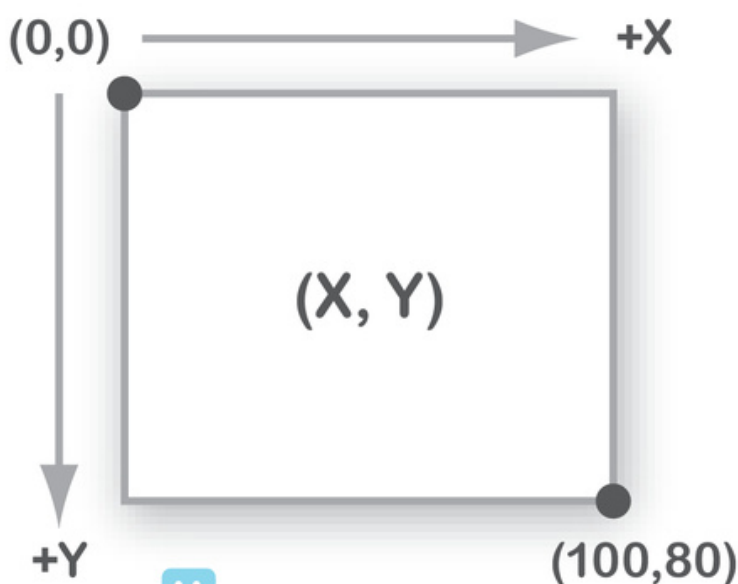
MAPA DE IMÁGENES

Los mapas de imágenes se han convertido en una técnica bastante olvidada debido al uso del mouseover. Sin embargo, pueden ser una alternativa muy práctica, efectiva y, en algunos casos, rápida de implementar.

Los mapas de imágenes son **áreas** que se colocan sobre una imagen de un sitio web para generar más de un enlace en ella. Antiguamente, no eran reconocidos por los navegadores, y se debía recurrir al servidor para utilizarlos. En la actualidad, en cambio, se los elabora mediante código HTML y casi todos los browsers los admiten. Como la definición del mapa de imágenes se encuentra dentro del archivo HTML, es rápidamente accesible y permite, en muchos casos, trabajar sin conexión. Este recurso es sencillo de usar y resulta ideal cuando queremos utilizar una porción de una imagen como enlace. A diferencia de lo que muchos diseñadores creen, los mapas sí soportan **comportamientos (scripts)**, ya que a todos ellos les podemos asignar acciones, desde el estado **off** hasta el **on**. Estos mapas tienen algunas ventajas en relación con otras formas de diagramar o actualizar un sitio, entre ellas, las siguientes:

- ▶ Se cargan automáticamente aunque la imagen no lo haya hecho, y pueden utilizarse mientras tengan su texto de referencia.
- ▶ Se adaptan fácilmente a distribuciones complicadas o complejas, lo que nos evita tener que cortar o separar las porciones de la imagen que emplearemos como enlace.
- ▶ Podemos volver a utilizarlos una vez definidos, de modo que ahorramos mucho tiempo a la hora de ubicarlos.
- ▶ Al aplicarlos en toda una barra de navegación, por ejemplo, sólo deberemos cambiar la imagen, porque si los elementos se mantienen en el mismo lugar, no será preciso reubicar o modificar el mapa, y así tampoco tendremos que editar el archivo **HTML**.

Los mapas de imágenes están compuestos por dos partes. La primera es la imagen que se utiliza dentro del sitio, y la segunda está dada mediante la **etiqueta o tag <map>**, que delimita el espacio empleado como enlace. Estos mapas de imágenes están restringidos por un espacio geométrico del que obtendremos las dimensiones mediante las **coordenadas X**, para el **ancho**, e **Y**, para la **altura**, teniendo en cuenta que el vértice superior izquierdo es **0,0**. Para obtener manualmente las coordenadas donde queremos ubicar el mapa, debemos recurrir a un programa como Photoshop.



Este recuadro mide 100 por 80 píxeles, contados desde el ángulo superior izquierdo. Entonces, la coordenada X es 100, y la Y es 80, para delimitar el ángulo inferior derecho.



Cada etiqueta **<map>** posee la etiqueta **<area>**, que puede ser una sola o más, dependiendo de cuántos enlaces necesitemos en una misma imagen. La etiqueta **<area>** tiene algunos atributos que le son propios y que enumeramos a continuación:

- El atributo **[alt]** muestra un texto cuando situamos el mouse encima del área, a diferencia de lo que sucede con el **[alt]** de las imágenes, que nos permite nombrarlas si el navegador no puede cargarlas.
- El atributo **[shape]** indica el tipo de área. Ésta puede ser **[rect]** si se trata de un área recta, como un cuadrado o un rectángulo, que siempre tendrá dos pares de coordenadas: X1, Y1, X2, Y2. Por su parte, **[circle]** es un área circular, que tendrá un par de coordenadas seguidas del radio: X1, Y1, R. Finalmente, **[poly]** es un área editable que puede tener la forma que deseemos, con la cantidad de coordenadas que necesitemos. Por ejemplo: X1, Y1, X2, Y2, X3, Y3, X4, Y4.
- El atributo **[coords]** corresponde a los valores numéricos que definen el área en donde se encontrará el mapa de imágenes; son sus coordenadas de posicionamiento.
- El atributo **[href]**, como en el resto de las etiquetas, se utiliza para enlazar el mapa de imágenes con el archivo o dirección que queramos relacionar.

Debemos decir, además, que para que la imagen actúe como mapa, es preciso colocar **[usemap="#nombre_del_mapa"]** dentro de su etiqueta. En el **Código 1**, podemos ver la forma en la que debemos colocar un mapa de imágenes.

CÓDIGO 1

```

<map name=" Imagen" id=" Imagen" >
<area shape="rect" coords="44,139,124,189" href="enlace_al_
sitio_que_deseemos" alt="Rectángulo" />
<area shape="circle" coords="71,70,37" href=" enlace_al_sitio_
que_deseemos " alt="Círculo" />
<area shape="poly" coords="287,236,222,309,209,296,214,291"
href=" enlace_al_sitio_que_deseemos" alt="Polilínea o Línea
adaptable" />
</map>
```



En <http://theall-knowing4.blogspot.com> encontramos un ejemplo de mapa de imágenes combinado con CSS para lograr el mouseover.

Los mapas de imágenes son realmente útiles si sabemos aplicarlos y utilizarlos. En muchos casos, nos ahorrarán bastante tiempo que perderíamos cortando imágenes para enlazarlas. Las formas rectas y circulares son las más sencillas de crear usando Dreamweaver, y aunque la herramienta con la que podemos dibujar una polilínea lleva un poco más de tiempo de manejo, no difiere del modo de uso de, por ejemplo, la **[Pluma]** de **Adobe Illustrator**.

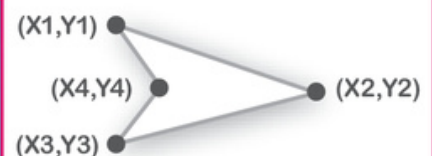
SHAPE = "RECT"



SHAPE = "CIRCLE"



SHAPE = "POLY"



En esta imagen podemos observar los tres tipos de área que se le pueden asignar al atributo **[shape]**.

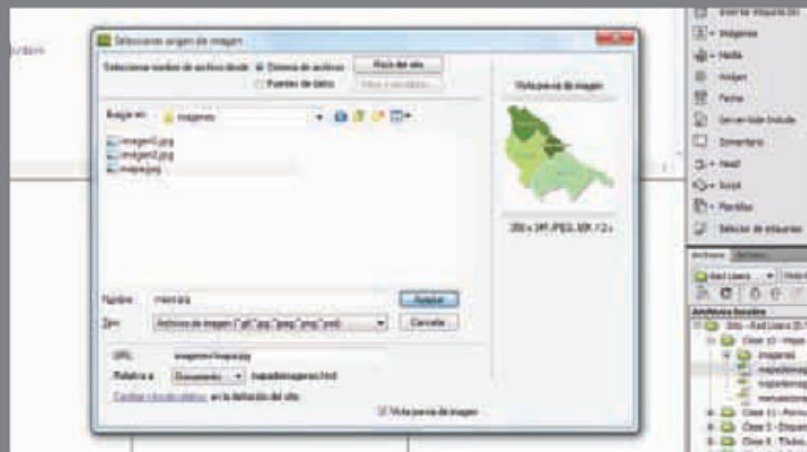
MAPA DE IMÁGENES

PASO A PASO

Conoceremos la forma de utilizar los mapas de imágenes para establecer diferentes secciones interactivas en ellas.



01 | LA IMAGEN ORIGINAL



INSERTAR LA IMAGEN | 02

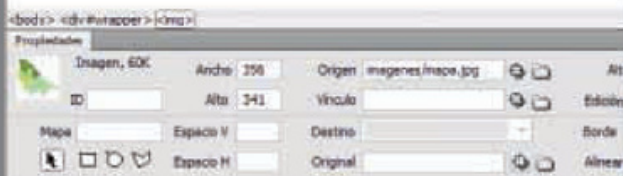
Los mapas de imágenes pueden ser muy útiles cuando queremos asignar diferentes enlaces a una de ellas, ya que así nos ahorraremos la inclusión de varias. Para esta técnica, Dreamweaver es una gran ayuda, ya que nos permite crear mapas de imágenes de forma muy sencilla, como si estuviéramos dibujando. ◀

PASO | 01 Lo primero que haremos será seleccionar la imagen que queremos utilizar. En este caso, vamos a usar un mapa, una opción ideal para aplicar esta técnica porque podemos establecer un enlace en cada una de las zonas geográficas, sin tener que recurrir a varias imágenes.

PASO | 02 Dentro de Dreamweaver, en el panel derecho **[Insertar]** hacemos clic en **[Insertar etiqueta Div]**, escribimos **wrapper** dentro de **[ID]** y hacemos clic en **[Aceptar]**. Eliminamos el contenido del div y presionamos la combinación de teclas **<Ctrl + Alt + I>**. En la ventana **[Seleccionar origen de imagen]** que se abre, buscamos la imagen y pulsamos **[Aceptar]**. Escribimos el texto alternativo y volvemos a presionar **[Aceptar]**.

PASO | 03 Con la imagen seleccionada, nos dirigimos a la barra inferior **[Propiedades]** y allí, debajo del recuadro **[Mapa]**, seleccionamos la herramienta que mejor se adecue a lo que necesitamos hacer. En este caso, como la forma es irregular, elegimos **[Herramienta Zona interactiva poligonal]**.

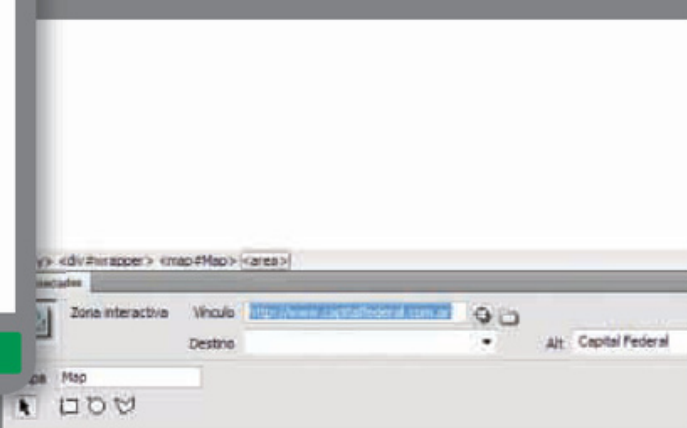
03 | CREAR EL MAPA



CREAR UN ÁREA

04

05 | AGREGAR ÁREAS



VÍNCULOS E INFORMACIÓN

06

PASO | 04

Al hacer clic sobre la imagen, aparecerá un cartel recordándonos que debemos colocarle el texto alternativo (campo **[Alt]**) al mapa de la imagen. Hacemos clic en **[Aceptar]** y comenzamos a dibujar sobre el mapa la zona en la que queremos generar el primer enlace. Para cerrar el área, presionamos en la flecha denominada **[Herramienta Puntero de zona interactiva]**.

PASO | 05

Volvemos a repetir la operación del paso anterior hasta dibujar todas las zonas a las que queremos dar un enlace independiente. Al hacer esto sobre una misma imagen, Dreamweaver no creará varios mapas, sino que generará distintas áreas dentro de él.

PASO | 06

Para terminar, en el panel **[Propiedades]** completamos el campo **[Alt]** de cada una de las áreas del mapa. Desde allí, también estableceremos el enlace que le corresponde a cada área, escribiendo la dirección dentro de **[Vínculo]** o utilizando el señalador de archivo.

RESUMEN

Utilizando este recurso, podemos generar zonas dentro de una misma imagen para usar como enlaces diferentes. Estos mapas interactivos nos permiten ahorrar imágenes y, por lo tanto, tener menor peso en el sitio. Además, podemos colocarlos de manera exacta editando el código HTML e indicando las coordenadas de pantalla precisas en las que queremos que se muestren.

PROPIEDADES DE LAS IMÁGENES EN CSS

Las imágenes en HTML son, básicamente, nombres de archivos colocados dentro de una etiqueta `<img>`, con atributos específicos que podemos modificar mediante CSS. Veamos las propiedades de las imágenes en CSS.

Cuando se empezaron a utilizar imágenes en los sitios web, éstas tenían muy pocas propiedades. Sin embargo, con el correr del tiempo, se expandieron y se volvieron más flexibles, gracias al empleo de las reglas CSS.

Al insertar una imagen en el espacio que le hemos designado dentro del HTML, ésta obtendrá, por defecto, sólo dos propiedades. La primera es **[width]**, que determinará su ancho en píxeles; y la segunda es **[height]**, para la altura. Otra que podemos agregarle a la etiqueta `<img>` es la propiedad o atributo **[alt]**, que sin duda es una de las más importantes, ya que con esta información, si el navegador no consigue cargar la imagen, mostrará un texto que hará referencia a ella.

PROPIEDADES CSS

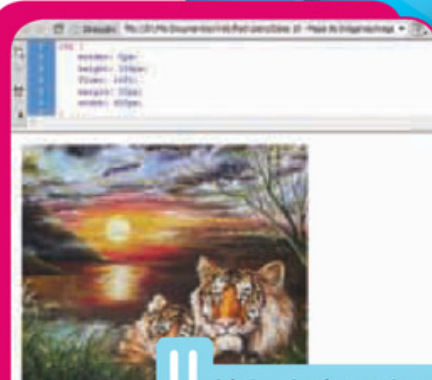
Las propiedades que le otorga CSS a una imagen son variadas. No sólo podemos modificar **[width]** y **[height]**, sino que también podemos darle propiedades de margen para establecer su marginado e, incluso, aplicarle propiedades de borde



En www.digiglass.com podemos ver un buen ejemplo de la aplicación de una imagen como fondo.

mediante **[border]**. Esto resulta útil si la imagen es un enlace, dado que, por defecto, dicho elemento tiene atribuida la propiedad **[border: 1px solid]**, que podemos hacer que no se vea o que únicamente se vea en uno de sus lados. También podemos utilizar la propiedad **[float]**, que nos permitirá hacer flotar la imagen y ubicarla donde consideremos más conveniente. Esta opción es práctica cuando queremos colocar algo al lado de la imagen sin necesidad de generar un div, sino, simplemente, dándoles a las etiquetas contiguas la propiedad **[float]**. Por su parte, **[resize]** permite redimensionar, con lo cual nos da la posibilidad de mostrar la imagen en un tamaño diferente del original. Ésta es una propiedad que se ha agregado a CSS3.

Para finalizar, hablaremos de una propiedad que, en muchos casos, resulta controvertida. Se trata de **[background]**, mediante la cual una imagen puede ser ubicada o posicionada como background dentro de casi cualquier etiqueta. En este caso, si la usamos como fondo, podremos generar mosaicos por repetición, escribir encima de él y hasta producir sprites con una única imagen. Aunque es una práctica muy utilizada para generar los fondos de los sitios o, incluso, los menús de navegación, debemos tener en cuenta que no es favorable para los motores de búsqueda, ya que les impide indexar la imagen como información. Por este motivo, debemos utilizar esta propiedad sólo cuando es realmente necesario, para no restar la posibilidad de lograr un mejor posicionamiento en los buscadores. Recordemos que éstos, a la hora de indexar, ingresan todo aquello que puedan leer como información, algo que no sucede cuando una imagen está colocada con la propiedad **[background]**.



A la izquierda, una imagen con propiedades establecidas en el código CSS. A la derecha, la misma imagen colocada como background. En el código HTML no hay ninguna etiqueta `<img>`.

CSS SPRITES



Ésta es una técnica pensada para optimizar el uso de imágenes. En la actualidad, se aplica, esencialmente, en las botoneras de los sitios web. Veamos de qué manera implementarla.

La técnica de sprites de imágenes se ha empleado durante años, incluso, antes de que se la pusiese en práctica para desarrollar partes de un sitio web. En sus inicios, fue utilizada para generar los movimientos básicos de los videojuegos elaborados en 2D, que mediante programación algorítmica, mostraban al personaje moviéndose y haciendo lo que nosotros deseábamos. Un ejemplo concreto de esto es el juego Mario Bros, que a lo largo de más de una década, usó sprites en sus diversas versiones.

¿QUÉ SON LOS SPRITES DE IMÁGENES?

Se trata de una técnica que consiste en tener varias construcciones de imágenes dentro de una única imagen. En el ámbito web, esto podría significar

incluir varias posiciones de un mismo botón, como su estado off, hover y on. Otra utilidad es colocar, en una sola imagen, todas aquellas que utilizaremos repetidas veces en un sitio web, tal como hace Google, en su buscador, o Yahoo Mail, en su casilla de correo.

¿CÓMO SE UTILIZAN LOS SPRITES DE IMÁGENES?

Si bien al principio el uso de sprites no es sencillo, con algo de práctica y atención, se transformará en una herramienta útil y de fácil implementación. Para utilizarla, primero debemos colocar la imagen como background; es decir, si se trata de una botonera, la imagen funcionará como fondo de ella. Luego, mediante la propiedad **[background-position]**, vamos asignando las distintas ubicaciones que

LOS EJES X E Y PUEDEN TENER VALORES TANTO POSITIVOS COMO NEGATIVOS, ALGO FUNDAMENTAL EN LA TÉCNICA DE SPRITES.

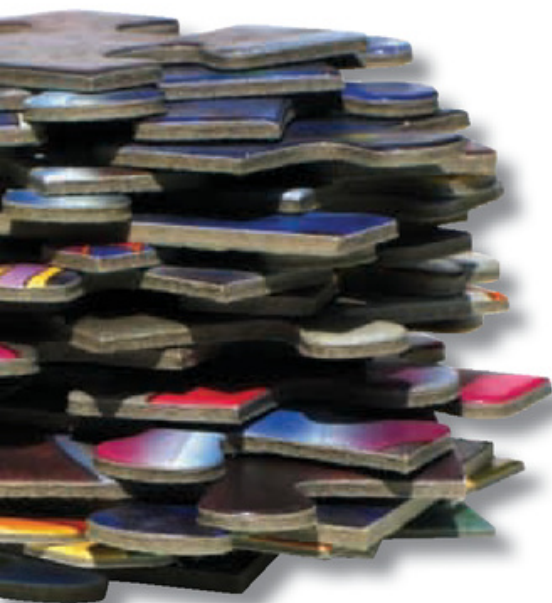
se mostrarán de la imagen mediante los ejes X e Y. El primero corresponde a la línea horizontal, también llamada ancho de la imagen; y el segundo, a la línea vertical, o alto de la imagen. Ambos pueden establecerse con valores positivos o negativos.

VENTAJAS DE ESTA TÉCNICA

La técnica de sprites es muy efectiva y, cuando nos familiaricemos con ella, es probable que sea una de las primeras opciones que evaluemos a la hora de realizar una gran cantidad de imágenes que se repitan en un sitio, tanto para los botones como para alguna porción del fondo (es menos empleada para este último fin, pero hay quienes la aplican). Como ventajas significativas, debemos mencionar que, al tener todo agrupado en una, dos o tres imágenes, su carga es más rápida. La transferencia de datos entre el servidor y la máquina del usuario se acelera, lo que le brinda al visitante una mayor velocidad de respuesta al acceder al sitio. ▶



En esta imagen, podemos observar el sprite utilizado para el juego Super Mario 3.



Dos ejemplos que debemos tener en cuenta son los mencionados anteriormente: el buscador de Google y Yahoo Mail. El primero sólo utiliza dos imágenes para generar la representación gráfica. En este caso, nos concentraremos en la imagen con la que se generan los resultados de búsqueda. Ésta no tiene más que 150 píxeles de ancho por 106 de alto con formato PNG y fondo transparente, lo que permite colocarla sobre cualquier tipo de fondo (también podría haber sido un GIF con fondo transparente). Esta imagen está constituida por la marca, las flechas hacia adelante y hacia atrás, las correspondientes a arriba y abajo, la cruz para cerrar una búsqueda, las estrellas de favoritos, el signo más y el signo menos, los signos de conversión y, finalmente, **Google Checkout**. Para analizar mejor este ejemplo, podemos decir que Google utiliza, en el caso de las O amarillas del logo, un **[background-position]** que se repite de acuerdo con la cantidad de páginas que tenga el resultado. Con esta técnica, la carga se genera cuando ingresamos por primera vez al buscador.



Podemos observar el sprite que se utiliza para el buscador Google (www.google.com); se trata de un archivo PNG.

REDUCIR LA CANTIDAD DE PETICIONES AL SERVIDOR DISMINUYE CONSIDERABLEMENTE LA VELOCIDAD DE CARGA DEL SITIO.

Por otro lado, tenemos el ejemplo de Yahoo Mail, la casilla de correo de Yahoo. Esta empresa utiliza una única imagen para generar no sólo los botones, sino también los fondos de los tabs de la casilla de correo. De esta forma, la optimización en la carga de las imágenes podría denominarse única, pues hace que el sitio resulte accesible para todos los usuarios, sin importar el ancho de banda que posean.

Recordemos que la creación de un sitio web no significa, simplemente, lograr un buen diseño, y optimizar el código HTML, CSS, PHP o el que usemos. También implica optimizar las imágenes, porque si son bien utilizadas, harán que el sitio sea más amigable y tenga un concepto efectivo de usabilidad.



Esta es una pequeña porción del sprite que se utiliza en el sitio de Yahoo Mail (www.mail.yahoo.com).



UTILIDADES WEB

Si no tenemos ganas de crear las imágenes de sprites por nosotros mismos, podemos recurrir a sitios de Internet que brindan este servicio. Dos de ellos son www.csssprites.com y www.css-sprit.es. Ambos generadores brindan una gran cantidad de opciones para configurar y lograr exactamente el sprite que necesitamos.

SPRITES EN CSS

Haremos el análisis de un sitio muy conocido para comprender de qué forma su sprite es aplicado a lo largo de él.

Además de ser popular por los videos que ofrece, YouTube (www.youtube.com) debe parte de su éxito a su accesibilidad y usabilidad. Es simple, rápido y posee una diagramación fácilmente entendible. Está diseñado y pensado para gente de todas las edades. El sprite del sitio incluye su logo, el de Google y los diferentes iconos que encontramos al navegar por él (banderas, manos de aprobación o desaprobación, flechas hacia todas las direcciones y cruces de cerrar), los botones y sus diversos estados (off, hover y on), también el signo de mail, y los logos de Facebook y Twitter, entre otros. Además, están los fondos de los tabs y sus respectivos colores.



CREAR SPRITES

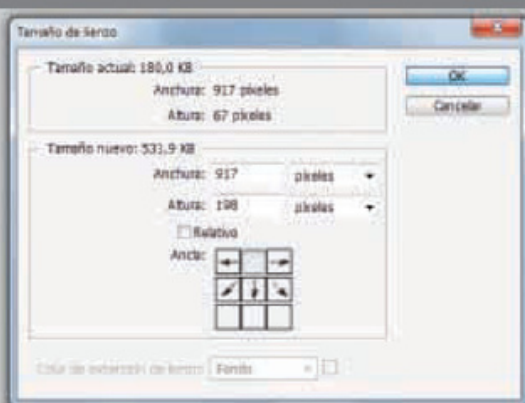
No solo podremos descargar sprites desde la Web para incluir en nuestro desarrollo, también podremos crearlos. Con **CSS Sprites Generator** (<http://csssprites.com>) solo deberemos seleccionar las imágenes, configurar las opciones, y ¡voilà!, tendremos nuestros sprites listos para incluir en las páginas de nuestro sitio.



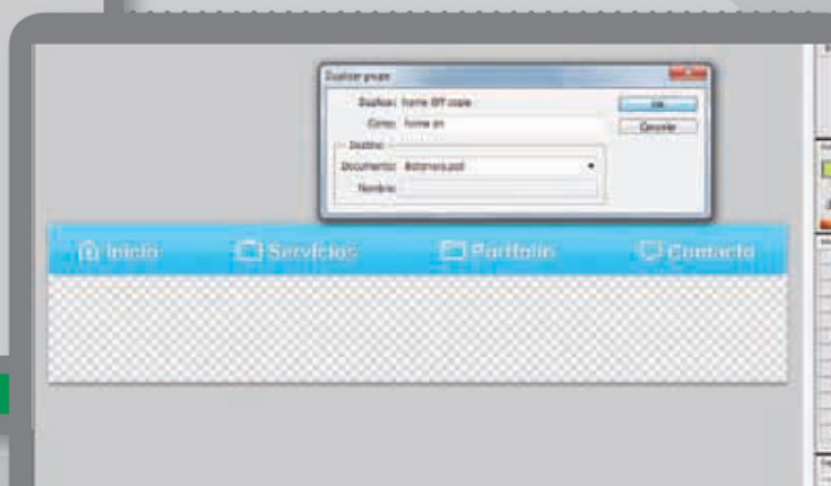
CSS SPRITES

PASO A PASO

En este apartado aprenderemos la forma de utilizar sprites en nuestros menús.



01 | TAMAÑO DE IMAGEN



ESTADOS | 02

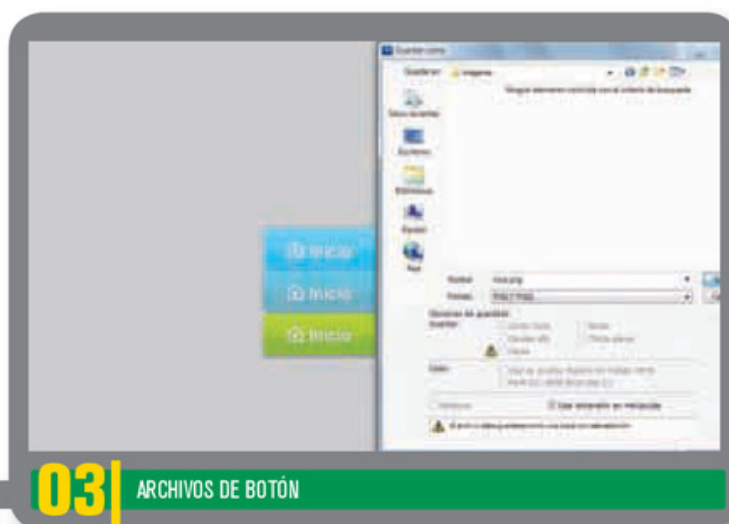
Los menús con sprites son muy utilizados en la actualidad, pero esta técnica no sólo se emplea en menús, sino que una gran cantidad de sitios también la aplican para determinados iconos o fondos que se irán repitiendo, con lo cual logran la carga de varios elementos desde una misma imagen. ◀

PASO | 01 Lo primero que hacemos es tomar nuestra botonera, abrirla en Photoshop CSS, ir al menú **[Imagen]**, hacer clic en **[Tamaño del lienzo]** o presionar **<Alt + Ctrl + C>** y colocar el tamaño de la botonera multiplicado por 3, ya que tendremos un estado **off**, uno **hover** y uno **on**.

PASO | 02 A continuación, duplicamos la capa de la botonera dos veces y ubicamos esas copias debajo de la original, haciendo los cambios de colores que hayamos determinado en el diseño para que se noten las diferencias entre los estados **off**, **hover** y **on**.

PASO | 03 Ahora vamos a cortar la imagen para separar cada botón, creando una imagen con los tres estados del botón para cada uno de ellos. Luego, los guardamos en formato PNG con fondo transparente, de modo que, si cambiamos el fondo del sitio, no necesitaremos volver a retocar la imagen.

EN MENÚS



03 | ARCHIVOS DE BOTÓN



LISTA DESORDENADA **04**



05 | ETIQUETAS NECESARIAS



ESTILO BOTONERA **06**

PASO | 04 Con las cuatro imágenes cortadas (una por cada botón con sus respectivos estados), abrimos el sitio en Dreamweaver y armamos un div al que denominaremos **[botonera]**. Éste contendrá una etiqueta **** (lista desordenada) y, en este caso, cuatro etiquetas **** con los nombres de los botones en el interior (**Inicio**, **Servicios**, **Portfolio** y **Contacto**).

PASO | 05 Dentro de cada etiqueta **** agregamos una **<a>** y, en su interior, una clase con el nombre del botón en cada caso. Entre la apertura y el cierre de esta última etiqueta colocamos una etiqueta ****, que contendrá el texto escrito en el paso anterior. De esta manera, podremos generar las reglas CSS sin tener que volver al código HTML.

PASO | 06 Creamos un archivo CSS denominado **botonera** y lo adjuntamos al HTML, como ya hemos hecho anteriormente. Ingresamos en ese archivo y escribimos la regla CSS para el div de la botonera: **[#botonera {background: #ccc; float: left; height: 66px; margin-top: 102px; width: 916px;}]**. De esta forma, le damos el tamaño (ancho y alto), el fondo que tendrá y la ubicación con respecto a la parte superior del sitio. Recordemos que usamos la propiedad **[float]** porque nos permitirá, en el futuro, agregar más divs a su alrededor.

CSS SPRITES

PASO A PASO EN MENÚS

07 | ESTILO LISTA

PASO | 07 Definimos la etiqueta `<ul>` mediante `[#botonera ul {list-style: none; margin: 0px 0px 0px -40px; width: 916px;}]`. Así, quitamos el estilo de lista (los bullets), la ubicamos en relación al div botonera (no tendrá diferencia de márgenes, y el último es negativo para que quede pegado al margen izquierdo), y le damos un ancho para que tenga el mismo tamaño que el div.

PASO | 08 Para definir la etiqueta `<li>` escribimos `[#botonera ul li {float: left;}]`. De esta manera, cada una de las etiquetas se colocará al lado de la otra (si no, quedaría una por debajo de la otra, y formaríamos un menú vertical).

ETIQUETA `<li>` | 08

ESTILO ETIQUETA `<a>` | 09

PASO | 09 Definimos la etiqueta `<a>` mediante `[#botonera ul li a {display: block; float: left; height: 66px; text-decoration: none;}]`. La primera propiedad es para que la etiqueta `<a>` tome forma, la segunda es para que flote como lo hace la etiqueta `<li>`, la tercera es la altura, y la cuarta es para que el texto no se subraye por defecto al ser un enlace.

PASO | 10 Creamos las cuatro reglas CSS para las clases de la etiqueta `<a>` como la siguiente: `[#botonera ul li a.inicio {background: url(imagenes/inicio.png) center top; width: 174px;}]`. En cada clase, elegimos el fondo (las imágenes que cortamos en el paso 3), lo ubicamos al centro y en la parte superior (`[top]`) y le damos el ancho si los botones tienen medidas diferentes, ya que si todos son iguales, se puede colocar en la etiqueta `<li>`, como hicimos con la altura, y repetirlo en `<a>`.



10 | ESTILO ETIQUETA <A>



OCULTAR LISTA | 11



12 | ESTADO HOVER



ESTADO ACTIVE | 13

PASO | 11 Para definir la etiqueta `<span>` de cada etiqueta `<a>`, escribimos `#[botonera ul li a span {text-decoration: none; visibility: hidden;}]`, de modo que el texto no se subraye y no se vea, y así no se superponga con los botones.

PASO | 12 Definimos el estado hover de la botonera con `#[botonera ul li a: hover.inicio {background-position: 0px 132px; #botonera ul li a: hover.servicios {background-position: 0px 132px; #botonera ul li a: hover.portfolio {background-position: 0px 132px; #botonera ul li a: hover.contacto {background-position: 0px 132px;}]`. Para la propiedad `[background-position]`, el primer valor es el eje X (el ancho de la imagen), que en este caso siempre será 0. El segundo valor es el eje Y (el vertical de la imagen), que será 132 px para que se muestre la posición hover.

PASO | 13 Para definir el estado active, escribimos lo siguiente para cada una de las clases: `#[botonera ul li a: active.inicio {background-position: center bottom;}]`. Así, definimos `[background-position]` como al inicio, mantenemos el centro de la imagen (aunque no sea necesario) y, en el eje Y, le pedimos que muestre la parte inferior de la imagen (que es el estado activo del botón).

RESUMEN

Esta técnica también es útil para colocar distintos botones que se repetirán a lo largo del sitio. Dependiendo del diseño que éste tenga, deberemos determinar el formato de archivo en el que nos convendrá guardar, valga la redundancia, los archivos (GIF o PNG si precisamos fondo transparente).

ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 ¿Qué son las imágenes de contenido?

02 ¿Qué son las imágenes de complemento?

03 ¿Cuáles son los atributos opcionales?

04 ¿Cuáles son los formatos más comunes de imágenes?

05 ¿Cuál es la diferencia entre **JPG** y **GIF**?

06 ¿Qué son los mapas de imágenes?

07 ¿Cuáles son las propiedades de las imágenes en **CSS**?

08 ¿Qué son los sprites de imágenes?

09 ¿Cuáles son las ventajas de los sprites de imágenes?

10 Explique los usos de la tecla **ALT** en diseño web.



PRÁCTICAS

01 Elija una imagen Web que va a utilizar en su sitio y optimícela para la web.

02 Con la misma imagen, mediante CSS, agréguele un borde y rediménsionela.

03 Posicione esa misma imagen como fondo, mediante la etiqueta **[background]**.

04 Cree un mapa de imagen y establezca diferentes secciones.

05 Utilice un **sprite** en su menú.



Formularios



- ▶ FORMULARIOS
- ▶ ELEMENTOS DE LOS FORMULARIOS
- ▶ FORMULARIO COMPLETO
- ▶ PHP: FUNDAMENTOS
- ▶ HACER FUNCIONAR LOS FORMULARIOS



FORMULARIOS

Los formularios facilitan la comunicación entre los usuarios de un sitio y los responsables de él. Además de ponerse en contacto, el usuario también puede realizar otras tareas, que veremos a continuación.

Los formularios web son muy similares a los impresos en papel y poseen su misma finalidad o función. Estos elementos permiten transmitir información entre un equipo cliente (el del usuario) y el servidor. Pueden ser de muy diversos tipos, dependiendo del objetivo que deban cumplir. Entre las opciones más comúnmente utilizadas, encontramos los de contacto, los de suscripción, los de pedido, los de votación, los de encuesta y los que se utilizan para ingresar un currículum en una bolsa de trabajo. Todos ellos se realizan en cuatro partes: la primera es la estructura dada por el código HTML, la segunda es la estética otorgada por el código o las reglas CSS, la tercera es la validación e interacción del lado del cliente (mediante los eventos y el código JavaScript) y la cuarta es la programación del lado del servidor me-



WEBAPPS

Además de los formularios que estamos mencionando, es importante saber que también existen completas aplicaciones web llamadas **webapps**, que los usuarios pueden utilizar accediendo a un servidor a través de Internet. Estas aplicaciones utilizan al navegador como ejecutor, y su popularidad se debe a que son fáciles de mantener actualizadas y que no es necesario instalar un software para poder usarlas. Ejemplos de esto son los weblogs, las wikis, las tiendas online e, incluso, Wikipedia. De esta manera el usuario puede acceder a datos de forma interactiva.

dante el código PHP (hablaremos de este lenguaje más adelante). Según su complejidad, encontramos dos clases de formularios:

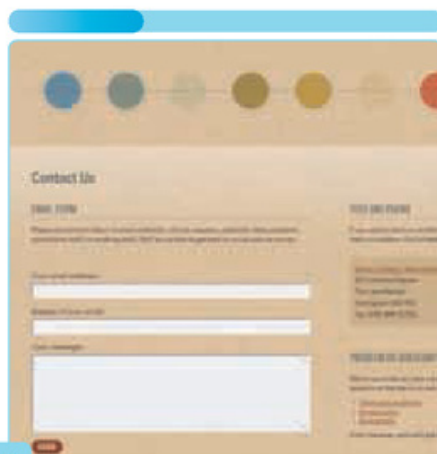
► **Básicos:** son aquellos que constan de muy pocas opciones y que están elaborados, principalmente, con etiquetas `<form>` e `<input>`. Dentro de este

grupo, podemos mencionar los formularios de suscripción o de login, entre otros.

► **Avanzados:** son aquellos formularios complejos que están elaborados no sólo con las etiquetas `<form>` e `<input>`, sino que además utilizan `<fieldset>` para agrupar o embloquear la información. Suelen requerir, además, la etiqueta `<label>` para generar una descripción de lo que se espera que se coloque en los campos que hay que completar. Ejemplo de este caso son los formularios de registro, los de compra y los de ingreso de currículum, entre otros.

Si analizamos los formularios según su finalidad, podemos hacer la siguiente división:

- **De contacto:** son los más utilizados en los sitios web. En ellos se piden los datos mínimos para poder comunicarnos con los responsables del sitio y, por lo general, sirven para evacuar alguna duda que tengamos.
- **De suscripción:** se utilizan cuando queremos que nos llegue información relacionada con algún tema de nuestro



Los formularios de contacto son los más utilizados en cualquier tipo de sitio. Aquí vemos los de www.newadventuresconf.com y www.smart-web-design.co.uk.



El captcha es un mecanismo para evitar que los formularios sean completados por procesos automáticos. Encontramos ejemplos en www.senior.co.uk y www.simplycreate.net.

LOS FORMULARIOS SON LA PRINCIPAL VÍA DE COMUNICACIÓN ENTRE LOS VISITANTES Y LOS DUEÑOS DE UN SITIO.

interés o que nos llama la atención. En la mayoría de los casos, sólo nos solicitan una dirección de e-mail.

► **De registro:** son el segundo grupo de formularios más utilizados en los sitios web, después de los de contacto. En ellos debemos colocar nuestros datos personales, un nombre de usuario y una contraseña, para tener acceso a ciertas partes del sitio restringidas a quienes no se registren. Son muy habituales

para formar parte de redes sociales, abrir cuentas de e-mail o participar en foros.

► **De login:** permiten acceder a los sitios en los que nos hemos registrado previamente. Suelen pedirnos un nombre de usuario y una contraseña.

► **De compra:** son aquellos que debemos completar cuando nos encontramos dentro de las llamadas **tiendas online**.

Estos formularios nos pedirán no sólo nuestros datos personales, sino también información sobre tarjeta de crédito o cuenta bancaria.

► **De encuesta:** estos formularios se utilizan en las páginas que brindan un servicio online. En general, su finalidad es ilustrar la experiencia del usuario en el sitio para mejorarla en el futuro.

► **De CV:** se encuentran en las páginas de bolsas de trabajo o de ciertas empresas y, como su nombre lo indica, nos permiten colocar todos nuestros datos para postularnos a un puesto laboral. En algunos de ellos, podemos incluir una foto o el archivo con nuestro propio CV.

CAPTCHA

Captcha es una prueba de seguridad destinada a diferenciar entre los humanos y las máquinas. Consiste en pedir la introducción de los caracteres que se muestran en una imagen distorsionada (imagen movida, caracteres tachados o incompletos, entre otras posibilidades). Su finalidad más importante es evitar que los robots (pequeños programas que tienen por misión utilizar ciertos servicios para generar correo o cuentas spam) utilicen sistemas de registro en cuentas de e-mail o redes sociales. ►



Los formularios de registro varían en cuanto a la cantidad de datos que solicitan, como vemos en www.getfitinspiredtraining.com, www.blogger.com y www.puntomio.com.

Con pocos elementos, los formularios de login son necesarios para controlar el acceso de los usuarios, como vemos en www.deploycms.com y en www.bundlehunt.com.

COMPOSICIÓN DE UN FORMULARIO

Aunque en las próximas páginas conoceremos en detalle cada uno de los elementos que integran un formulario, vamos a adelantar que, como sucede con otras páginas, este elemento está compuesto por una serie de etiquetas. La primera es `<form>`, que se utiliza como la estructura básica de todo formulario. Dentro de ella definimos los atributos, los métodos y la acción que llevará a cabo este recurso. Luego, está la etiqueta `<input>`, que puede tener varios usos diferentes:

- **Text:** una caja de texto simple para textos cortos.
- **Radio:** para seleccionar una única opción dentro de una casilla circular.
- **Checkbox:** para seleccionar una o más opciones dentro de una casilla cuadrada.
- **Submit:** propiedad que se asigna a las etiquetas `<input>` que hacen referencia a un botón (botones de envío, de borrado, etcétera).
- **File:** botón que permite buscar archivos en nuestro equipo y subirlos al sitio.
- **Reset:** botón para indicar al navegador que debe llevar los campos del formulario a su configuración predefinida (borra lo que se ha completado).

Encontramos también una etiqueta `<textarea>`, un campo de texto destinado a contener grandes cantidades de caracteres. La etiqueta `<select>`, por su parte, tiene por finalidad darnos a elegir la opción que queramos dentro de una lista, por lo cual tendrá en su

LA ETIQUETA `<FORM>` ES LA ESTRUCTURA BÁSICA DE TODO FORMULARIO E INCLUYE OTRAS ETIQUETAS, ATRIBUTOS Y DEFINICIONES.

interior la etiqueta `<option>`. Por otro lado, `<label>` se ocupa de mostrarnos la referencia del campo de texto y, finalmente, `<fieldset>` sirve para agrupar los distintos tipos de datos según cómo resulte más conveniente. Ésta contiene por defecto una etiqueta `<legend>`, que utilizaremos para colocar el título que le daremos al grupo de datos.

XFORM

Fue diseñado como una alternativa a los formularios tradicionales y representa la próxima generación de formularios HTML/xHTML. XForm posee muchas características avanzadas, ya que es un formulario que puede actuar en tiempo real cuando nos da a elegir las opciones: si hacemos clic en una lista desplegable, es probable que se estén pidiendo los datos de ella al servidor en el momento en el que lo hacemos. Este formulario puede especificar cómo serán validados los distintos datos a medida que se van ingresando, y no suele utilizar scripts separados. En la actualidad, XForm no es soportado por la mayoría de los navegadores.

EL CLIENTE Y EL SERVIDOR

Más adelante profundizaremos en el funcionamiento y la programación de los formularios, pero en este punto es importante mencionar que éstos pueden ser combinados con scripts para volverse dinámicos, es decir, funcionar. Entre los scripts llamados **del lado del cliente**, el más estandarizado es el lenguaje **JavaScript**. Los que trabajan **del lado del servidor** son un poco más complejos, dado que permiten crear todo tipo de tareas para lograr un uso más minucioso del formulario, que puede abarcar la autenticación de un inicio de sesión, el envío y el almacenamiento de información en una base de datos o la corrección ortográfica del envío de un e-mail. ◀

En www.informeddesign.com.au y www.34interactive.com vemos ejemplos de formularios de suscripción. En Informed Design, el formulario se presenta con el de contacto.

ELEMENTOS DE LOS FORMULARIOS

Permiten recibir información de los usuarios de nuestro sitio, y se componen de diversos elementos con funciones específicas, a los que podemos asignar funciones particulares.

Los formularios son la manera más utilizada para que el usuario de un sitio se comunique con el administrador o dueño. Allí, el visitante encuentra una serie de campos de texto que debe completar, y botones que se encargarán de que el mensaje sea procesado y enviado al administrador del sitio o a un programa que lo hará automáticamente. Existen distintos modos de hacer que el formulario que tenemos en nuestro sitio web funcione. Por un lado, podemos usar la opción que brindan muchos servi-

dores, que consiste en alojar el contenido del formulario completado en un archivo dentro del servidor. Por otro, podemos aprender a programar en **PHP** o **ASP** para que el contenido del formulario, una vez procesado, llegue a la dirección de e-mail que deseemos. Todos los formularios tienen por estructura básica una etiqueta denominada **<form>**, que se cierra con **</form>**. Dentro de ella, vamos a colocar el contenido que deseemos. Esta etiqueta posee ciertos atributos que veremos a continuación:

► **action:** define la acción que debe realizar el formulario. Tenemos dos formas de usar este atributo. La primera es asociarlo a un archivo que se encargará de procesar el contenido del formulario. En este caso, el código es **<form action="mi_archivo.php o .asp"></form>**. La segunda alternativa consiste en enviar el formulario sin procesar a una dirección de e-mail, de la siguiente manera:

<form action="mailto:midireccion@decorreo.com"></form>.

► **method:** se ocupa de especificar cómo debe ser enviado el contenido del formulario. La primera manera de hacerlo es mediante el método denominado **[GET]**. Éste utiliza la barra de direcciones del navegador para enviar el contenido, volviéndolo en cierto sentido público, porque se visualiza en dicha barra. Este sistema no soporta el envío de imágenes ni más de 500 caracteres. Está muy difundido en las páginas web dinámicas para la elaboración de consultas que se encuentran asociadas a una base de datos. La segunda opción es el método denominado **[POST]**, el más utilizado para enviar el contenido de un formulario porque lo hace sin recurrir a la barra de navegación del browser, con lo cual oculta el contenido. Además, permite adjuntar y enviar imágenes, y la cantidad de caracteres es, por decirlo de alguna forma, ilimitada.

► **enctype:** se usa dependiendo de cómo procesemos el contenido del

nos la jugamos o lo cambiamos

Aporta tu idea para eliminar la pobreza infantil en España

El 25% de la población infantil en España sufre:

1. Insuficiente
2. Mala calidad de la educación
3. Sanidad insuficiente
4. Trastornos de salud y padres
5. Malnutrición
6. Acceso desigual al agua

vuestros ideas

RECOMENDADO

Más enlaces

¡Ayuda a las madres jóvenes para conseguir un trabajo, mejor presión por maternidad, guarderías gratuitas, ayudas directas para la educación de los hijos hasta al menos los 16 años, actividades extracurriculares con la participación de las madres y padres, repartición de los programas en los de TIC, educación sexual, orientación sexual, al respecto a la diversidad y la discriminación.

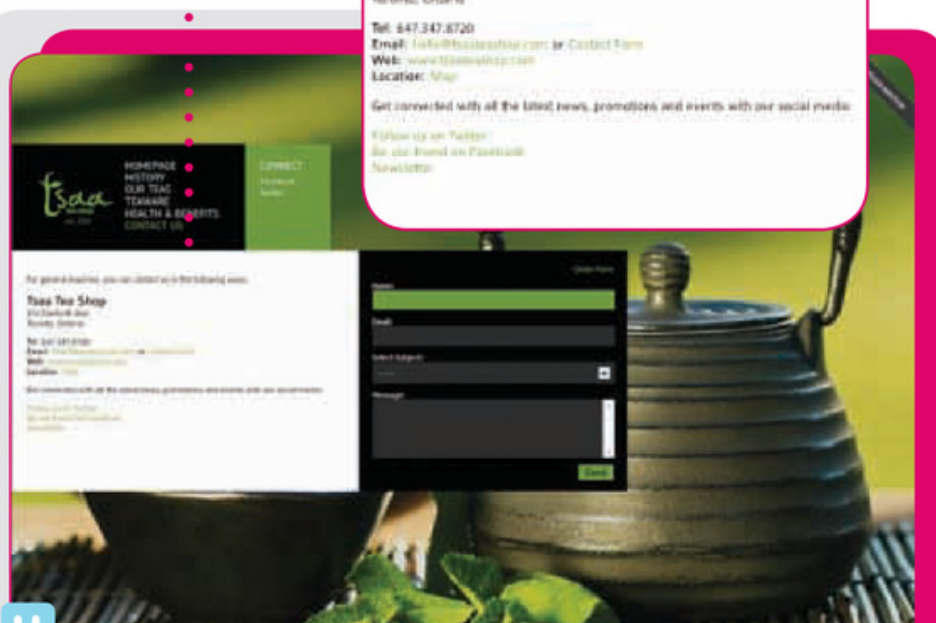
El sitio www.noslajugamosolocambiamos.org presenta un formulario con el estilo completamente ajustado al diseño general.

formulario. Si lo hacemos a través de un programa automáticamente, el atributo no se utiliza. En cambio, si el contenido será enviado a una dirección de correo electrónico, se coloca **[text/plain]** para que el mail llegue en texto plano. Por ejemplo: `<form action="mi_archivo.php o .asp (o dirección de mail)" method="post" enctype="text/plain">`.

A partir de ahora, conoceremos los distintos elementos que componen los formularios. Todos ellos, como cuadros de texto y botones, pueden denominarse **campos de texto** y **controles de formularios**, respectivamente.

ETIQUETA INPUT

Se utiliza para textos cortos, se asigna mediante `<input>` y podemos aplicarle dos atributos: **[type]** y **[name]**. La etiqueta se define de la siguiente manera: `<input type="text" name="nombre" value="">`. Con este ejemplo, creamos una caja de texto corto, cuyo contenido será **nombre**. Al atributo **[name]** es posible colocarle cualquier palabra que deseemos, aunque es aconsejable que esté relacionada con su contenido. El atributo **[name]** es muy importante, debido a que es el nombre que utilizaremos para tomar la información del campo de texto cuando procesemos el formulario. No debemos utilizar caracteres especiales o latinos para nombrar los campos, y siempre tenemos que estar de acuerdo con el programador que trabajará en el sitio, ya que la aplicación usada para enviar el formulario debe tener, entre sus **variables**, el nombre que nosotros le otorgamos al atributo **[name]**. También es fundamental definir el atributo **[type]**, ya que la etiqueta `<input>` posee múltiples usos y es este atributo el que determinará la función que cumplirá. Los valores que podemos darle a **[type]** son: **[text]**, **[password]**, **[radio]**, **[checkbox]**, **[submit]**, **[reset]**, **[file]**, **[hidden]**, **[image]** y **[button]**. Un atributo menos importante, pero que también podemos definir, es **[size]**, que



Los campos del formulario de esta casa de té (www.tsateashop.com) cambian su color a verde cuando los estamos editando.

EL ATRIBUTO [SIZE] ASIGNA UN TAMAÑO A LA ETIQUETA, Y [MAXLENGTH] LIMITA LOS CARACTERES QUE SE PUEDEN ESCRIBIR.

nos permitirá establecer el tamaño del campo de texto o etiqueta en un número de caracteres visibles. Si no definimos este atributo, cuando el campo de texto se muestre, lo hará en un tamaño preestablecido por el navegador que estemos empleando. El atributo **[maxlength]** nos permite definir la cantidad máxima de caracteres que se pueden colocar dentro del campo de texto. De esta forma, si le asignamos cierto valor, una vez que sea alcanzado, el navegador impedirá seguir escribiendo. Aunque es un error común confun-

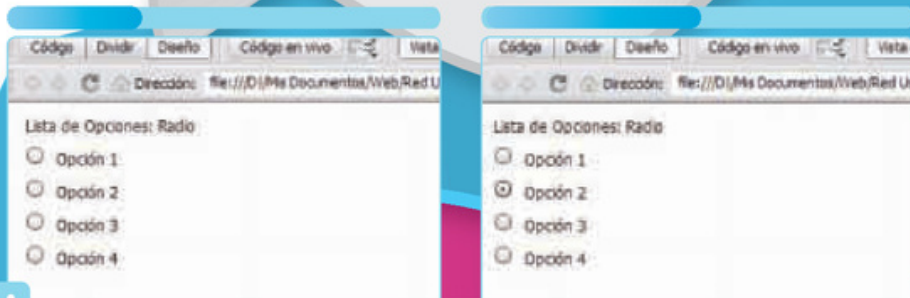
dir **[maxlength]** con **[size]**, debemos recordar que **[size]** simplemente asigna un tamaño a la etiqueta, mientras que **[maxlength]** restringe la cantidad de caracteres que podemos ingresar. El atributo **[value]** sirve para otorgarle un valor inicial al campo si no queremos que aparezca vacío. En algunas ocasiones, se lo utiliza para detallar, mediante un ejemplo, la forma en la que debe completarse el campo de texto, como vemos a continuación: `<input type="text" name="nombre" value="Escriba su nombre" />`. El atributo **[value]** siempre funciona dentro de una etiqueta `<form>`, nunca fuera de ella.

TEXTO OCULTO

Otro caso en el que usamos el elemento `<input>` es cuando deseamos ocultar el texto que queremos escribir por medio de asteriscos o círculos. Para esto, la única diferencia estará en el atributo **[type]**, en donde debe figurar **[password]**. Este atributo es muy utilizado cuando queremos generar campos para claves o logins. Veamos



Aquí podemos encontrar campos de texto utilizados de distintas maneras: tal como aparece por defecto, con texto de ayuda y configurado para ocultar los caracteres.



En este caso, podemos ver una lista de opciones sin selección predeterminada, a la izquierda; y la segunda opción elegida por defecto, a la derecha.

un ejemplo: `<input type="password" name="contrasena" value="" />`

ETIQUETA RADIO O BOTÓN RADIO

El tipo **[radio]** nos permite hacer elegir una única opción de una lista. La etiqueta empleada es `<input>`, pero con su atributo **[type]** establecido como **[radio]**. Al usar este elemento, debemos generar los textos y los saltos de línea en el código HTML. Veamos un ejemplo:

```
<input type="radio"
name="opciones" value="1"
/>Opción 1 <br /> <input
type="radio" name="
opciones" value="2"
/>Opción 2 <br />
<input type="radio"
name="opciones" value="3"
/>Opción 3 <br /> <input
type="radio" name="
opciones" value="4"
/>Opción 4 <br />
```

Como vemos en nuestro ejemplo, el atributo **[name]** es el mismo en todos los casos, pero **[value]** es diferente. Por lo tanto, cuando recibamos el e-mail sobre la opción seleccionada, veremos lo siguiente: **[Opciones = 2]**. Es posible **preseleccionar** una opción mediante el atributo **[checked]**. Para lograr la selección automática de la segunda opción, el código es `<input type="radio" name="lista" value="2" checked />Opción 2`.

El formulario que encontramos en www.quinttodigital.com es simple pero aparece en todas las páginas, ya que está incluido en el footer.



VARIABLES

Las variables son parámetros que nos permiten almacenar una información determinada durante cierto lapso de tiempo, que dependerá de la función que queramos realizar. Los nombres de las variables son sensibles al uso de mayúsculas y minúsculas en su denominación, y se definen con el signo \$. En un formulario, una variable almacenará, por ejemplo, la información de los campos, asociándolos al archivo PHP que nos permite enviar el formulario a la casilla de e-mail.



Aquí vemos el resultado de los distintos valores para el atributo **[type]**: **[submit]**, **[file]** e **[image]**.

CHECKBOX O CAJA DE SELECCIÓN O VALIDACIÓN

El tipo **[checkbox]** es más conocido como caja de validación, que puede ser activada o desactivada.

A diferencia de los botones radio, en este caso es posible seleccionar más de una opción al mismo tiempo. El código para usarla es: `<input type="checkbox" name="validacion" />` Caja de Validación.

Al igual que con **[radio]**, en este caso es posible seleccionar una opción de manera predefinida mediante el atributo **[checked]**. La información seleccionada puede ser indicada en nuestro correo como **[on]** u **[off]**, según corresponda.

HACER QUE LOS BOTONES SEAN REEMPLAZADOS POR IMÁGENES NOS PERMITE CONVERTIRLOS EN PARTE DEL DISEÑO.

BOTÓN SUBMIT O DE ENVÍO DE FORMULARIO

El tipo **[submit]** se utiliza para enviar el formulario una vez que esté completo. Se diferencia en estructura sólo en el atributo **[type]**, como vemos a continuación: `<input type="submit" value="Enviar" />`. Con **[value]** aplicamos al botón **[submit]** el mensaje que queremos que se muestre.

BOTÓN RESET O DE BORRADO DE FORMULARIO

El tipo **[reset]** se emplea para borrar el contenido del formulario si nos equivocamos al completarlo, ya que se vacían todos los campos y se los vuelve a su inicio. La forma de uso es: `<input type="reset" value="Borrar" />`, y es de empleo optativo.

BOTÓN FILE O DE ARCHIVOS ADJUNTOS

El atributo **[type]** con valor **[file]** se usa para permitir el envío de archivos mediante el formulario de contacto. Su código es: `<input type="file" name="adjunto" size="30" />`. Es importante que en este botón le demos un valor al atributo **[size]**, para establecer el

tamaño de la caja en la que se mostrará el texto con la ruta al archivo.

CAMPO HIDDEN O DATOS OCULTOS

El tipo **[hidden]** no es muy utilizado en sitios HTML, pero sí en aquellos que necesitan cierto tráfico con el servidor. Permite enviar datos adicionales para acelerar el proceso; por ejemplo, los formularios muchas veces se envían a dos cuentas de e-mail. Es un campo oculto que el usuario no podrá ver, a menos que observe el código fuente. Su uso es: `<input type="hidden" name="sitio" value="www.tuweb.com">`.

BOTÓN IMAGE

El tipo **[image]** sirve para colocar una imagen en reemplazo del botón **[submit]**. El código es: `<input type="image" src="imagenes/typeimage.jpg" alt="Submit" />`, donde podemos observar que aparecen dos atributos no utilizados hasta ahora. El primero es **[src]**, que permite seleccionar el archivo que emplearemos para que se muestre en reemplazo de **[submit]**. El segundo es **[alt]**, que se usa sólo cuando el navegador no puede mostrar la imagen.

En el sitio de Mooze (www.moozedesign.com), el formulario está modificado para lograr que se integre completamente con el diseño.

Contact Mooze



Gijsbert Koren
+31 (0)622843094

Shyam Soni
+91-9228431952

Email
contact@moozedesign.com

Name

Company

Phone

Email

Message

Submit

+ IMÁGENES Y EXPLORER

En general, cuando colocamos una imagen de fondo al botón **Submit** o le cambiamos el aspecto, nos encontraremos con que Internet Explorer, en la mayoría de sus versiones, sigue mostrando el botón predefinido que coloca el navegador. La manera más efectiva para evitar que esto suceda es definir la estética de este botón mediante una regla CSS asociada al ID (#). Así, no tendremos que redefinir la estética del botón en el archivo **iebugs.css**, que controla los pequeños desfasajes entre IE y el resto de los browsers.

El formulario en www.mustardmonkey.co.uk muestra un botón con imagen para efectuar el registro.

TYPE BUTTON O BOTÓN COMÚN

El atributo **type** con valor **[button]** es la definición de un botón normal. No tiene gran utilidad, aunque se define en algunos formularios si se quiere utilizar JavaScript para ocultar o mostrar determinados contenidos. Su uso es: `<input type=button value=" botón">`.

ETIQUETA PARA TEXTO LARGO

La etiqueta **[textarea]** sirve para que el usuario pueda escribir más de una línea en el campo de texto. Se inserta como `<textarea>` y debe cerrarse al finalizar. Suele utilizarse para que los usuarios envíen un mensaje o un comentario, o cuando deben completar alguna opción que necesita un desarrollo un poco más complejo que una línea. No se emplea para textos cortos, y posee los atributos **[name]**, **[rows]** y **[cols]**. El atributo **[rows]** define el número de líneas que se podrán escribir en el campo de texto, mientras que **[cols]** define el número de columnas en el que éste puede dividirse.

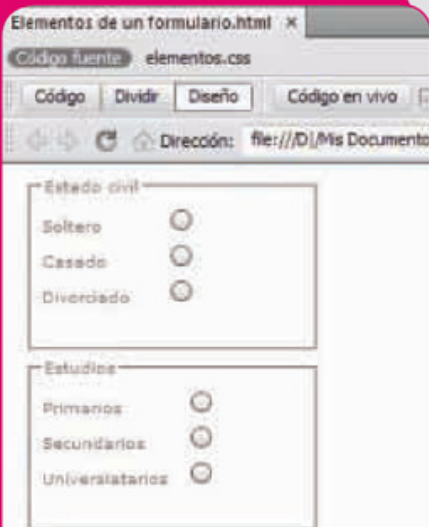
Un ejemplo de su uso es: `<textarea name="comentario" rows="10" cols="20"></textarea>`. En la etiqueta **[textarea]** no es posible definir el atributo **[value]**, por lo que si queremos dejar algún comentario para guiar al usuario, tendremos que hacerlo entre la etiqueta de apertura y la de cierre, como vemos a continuación: `<textarea name="comentario" rows="10" cols="20">Escriba su comentario...</textarea>`.

Aquí podemos ver la etiqueta **[textarea]**, tal como aparece por defecto y configurada para que muestre un mensaje al usuario.

LISTA DE OPCIONES

La etiqueta **[select]** también es conocida como **lista de opciones**, ya que permite elegir entre una serie de alternativas predefinidas por el diseñador del formulario. Está compuesta por una etiqueta **<select>** y su correspondiente cierre. Podemos establecer su nombre mediante el atributo **[name]**, determinando las opciones disponibles por una etiqueta **<option>** y su respectivo cierre. Veamos el código de ejemplo:

```
<select name="opciones">
<option>Opción 1</
option> <option>Opción
2</option>
<option>Opción 3</
option> <option>Opción
4</option> </select>
```

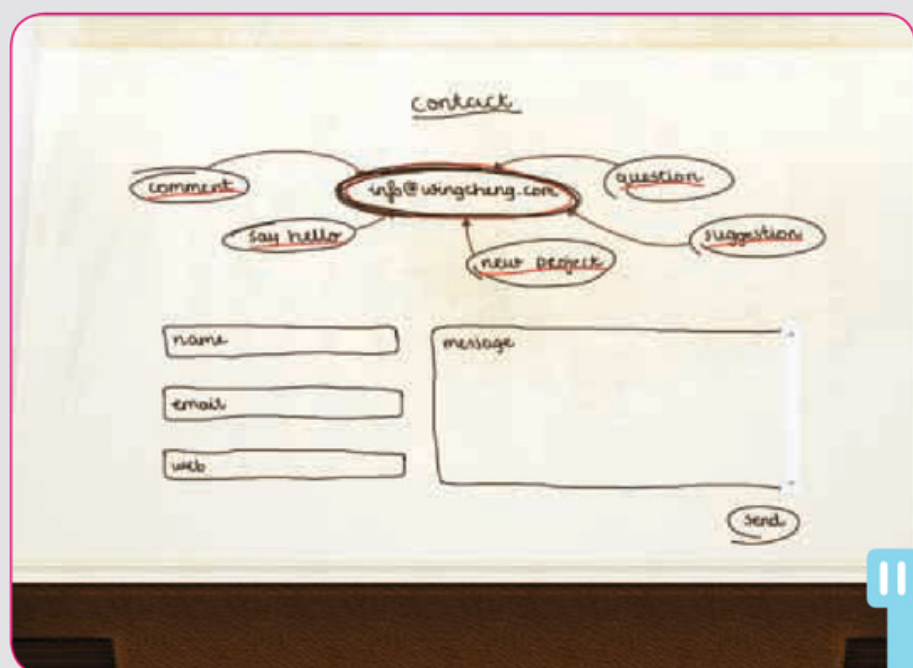


Aquí podemos observar claramente cómo están agrupados varios elementos mediante esta etiqueta.

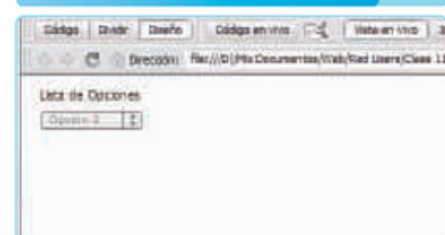
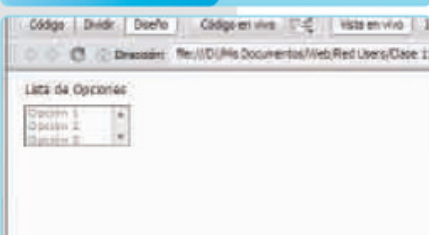
La etiqueta **[select]** puede tener tres atributos más. Por un lado, tenemos **[size]**, que se utiliza para definir el tamaño de la caja que posee la lista de opciones. Esto se traduce en la cantidad de opciones que se mostrarán a la vez, aunque de esta forma perderemos el campo desplegable y aparecerá, en el lado derecho de la etiqueta **[select]**, una barra de desplazamiento. Por otro lado, podemos definir **[multiple]**, para que sea posible seleccionar más de un elemento de la lista, utilizando la tecla **<Shift>** o **<Ctrl>**, de acuerdo con la versión de Windows que tengamos. Aunque no es recomendable utilizar este atributo, porque puede ser desconocido para el usuario que esté visitando el sitio, veamos un ejemplo:

```
<select name="opciones"
size="3" multiple>
<option>Opción 1</
option> <option>Opción
2</option>
<option>Opción 3</
option> <option>Opción
4</option> </select>
```

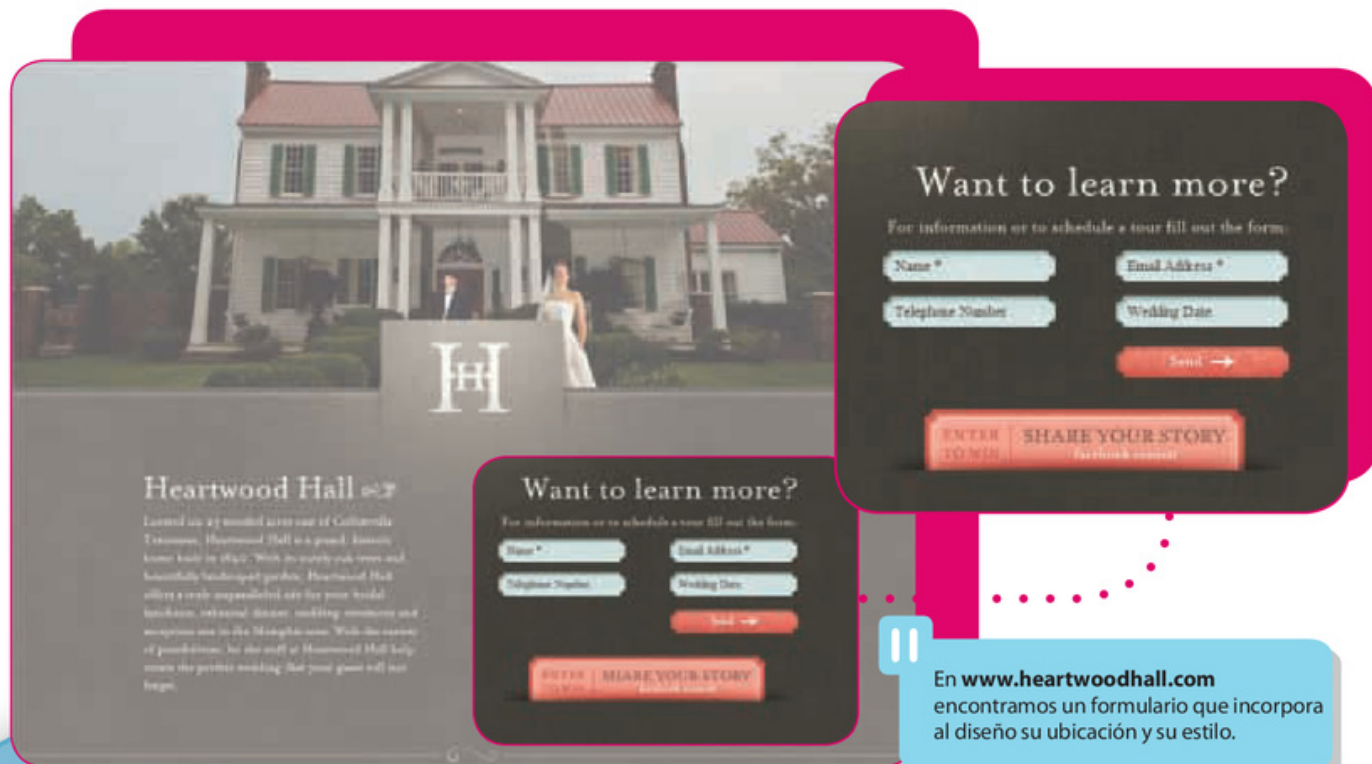
El último atributo es **[selected]**, cuya función, como su nombre lo indica, es hacer que la opción elegida aparezca seleccionada por defecto. Su uso es **<option selected>Opción 2</option>**. Por otra parte, el atributo **[value]** nos permite asociar una letra o un número al valor que elijamos de la lista. Es una función muy útil para aquellos formularios



En este caso, el formulario respeta perfectamente el diseño del sitio (www.wingcheng.com).



Primero, la etiqueta **[select]** como aparece por defecto; luego, con los atributos **[size]** y **[multiple]** configurados; y finalmente, con el atributo **[selected]** predeterminando una opción.



En www.heartwoodhall.com encontramos un formulario que incorpora al diseño su ubicación y su estilo.

LAS LISTAS SON MUY USADAS PARA EL LUGAR DE RESIDENCIA Y EL AÑO DE NACIMIENTO EN LOS FORMULARIOS DE REGISTRO.



ETIQUETAS Y NOMBRES

Bajo ningún concepto debemos utilizar tablas para armar el formulario. Aunque hasta no hace mucho tiempo ésta era una práctica frecuente para la mayoría de los desarrolladores web, un formulario debe realizarse con sus propias etiquetas y mediante reglas CSS. Debemos tener en cuenta el uso de clases y de ID, y definir la función de la etiqueta mediante `id`. Trabajando de esta manera, ahorraremos no sólo código en la estructura del HTML, sino también procesamiento en la apertura del archivo. Por otro lado, siempre es conveniente hablar con el programador para saber quién se hará cargo de asignar los nombres a los campos del formulario. En caso de que el implementador o maquetador (seríamos nosotros) deba hacerlo, convendrá utilizar nombres de campo claros, sencillos y que puedan ser fácilmente reconocidos. Por ejemplo, si la etiqueta `<label>` detalla **[Nacionalidad:]**, el nombre del campo debería ser **[nacionalidad]**. Probablemente, el que asigne el programador a la variable sea el mismo.

que deben ser procesados con posterioridad. Mediante el código `<option value="1">Opción 1</option>`, cuando el formulario sea enviado por correo electrónico, se mostrará **[Opción = 1]**.

BLOQUES DE ELEMENTOS

Los bloques de elementos nos dan la posibilidad de agrupar los distintos elementos de los formularios. Ya no necesitamos aplicar etiquetas externas al formulario para hacerlo (en una época, se utilizaban tablas, y luego se empleaban etiquetas `div` para agrupar en bloque los elementos; todas ellas, externas, que no pertenecen al form).

ETIQUETAS **FIELDSET** Y **LEGEND**

La etiqueta `<fieldset>` sirve para agrupar los distintos elementos de un formulario e, incluso, agregarles un título mediante la etiqueta `<legend>`. Esta última etiqueta permite utilizar el atributo `[align]` con

las propiedades `[center]`, `[top]`, `[right]`, `[bottom]` y `[left]`, según como queramos mostrar el título. Por defecto, `<fieldset>` genera un recuadro, y `<legend>` se coloca en la parte superior izquierda de éste, cortando la línea continua que lo conforma.

```
<fieldset> <legend>Bloque
de elementos</legend>
  Escribe lo que quieras
  <input type="text"
  size="20"> </fieldset>
```

ETIQUETA **LABEL**

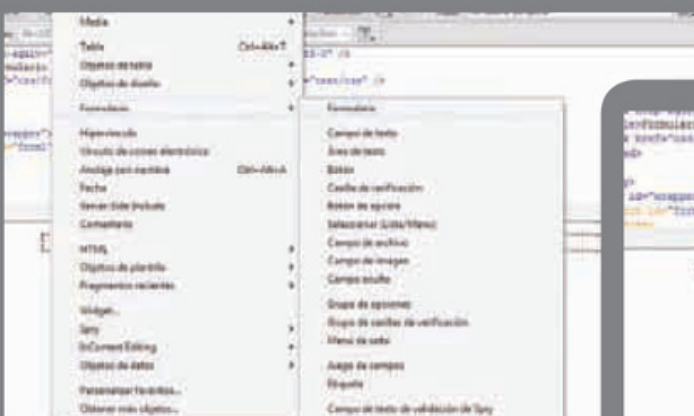
Permite colocar texto en su interior, que queremos mostrarle al usuario. Es muy usada para presentar el texto que corresponde a la etiqueta `<input>`. Por ejemplo, si el formulario tiene un campo llamado **[nombre]**, este texto estaría en el interior de `<label>`.

```
<label>Ingrese su E-Mail:
<input type="text"
name="email"> </label>
```

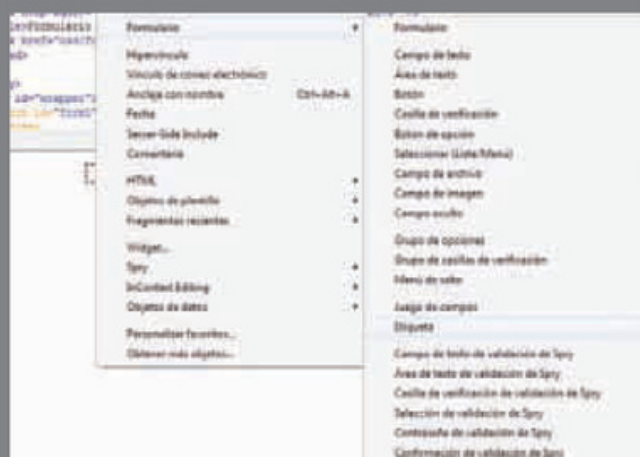

FORMULARIO COMPLETO

PASO A PASO

En este apartado veremos cómo generar un formulario con la ayuda de Dreamweaver.



01 | FORMULARIO



ETIQUETA | 02

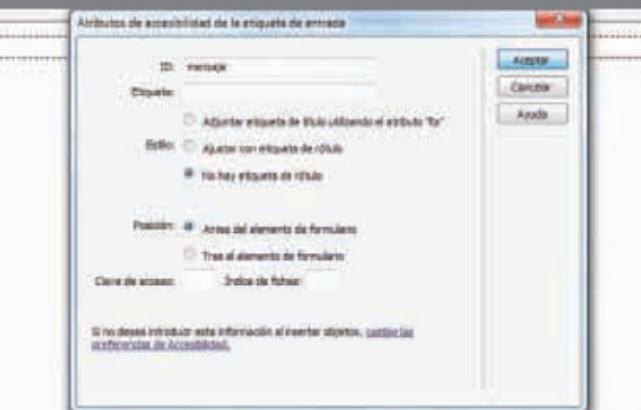
Los formularios son una parte fundamental del sitio de una empresa. Mediante ellos, los usuarios pueden comunicarse y dejar sus mensajes, sus datos personales o sus comentarios ▶

PASO | 01 Creamos un div **[wrapper]** y, dentro de él, insertamos la estructura básica del formulario. Para hacerlo, vamos al menú **[Insertar]**, seleccionamos **[Formulario]** y, en el desplegable que aparece, elegimos la opción también llamada **[Formulario]**.

PASO | 02 Nos dirigimos otra vez al menú **[Insertar]**, elegimos **[Formulario]** y, luego, hacemos clic en **[Etiqueta]**. Dentro de **<label>**, escribimos el detalle del campo que vamos a crear, en este caso, **[Nombre:]**.

PASO | 03 Nos dirigimos otra vez al menú **[Insertar]**, seleccionamos **[Formulario]** y, en el desplegable, hacemos clic en **[Campo de texto]**. En el recuadro que aparece, dentro de **[ID]** escribimos **[nombre]**, y entre las opciones dentro de **[Estilo]**

04 | APELLIDO Y E-MAIL



ÁREA DE TEXTO | 05



03 | CAMPO DE TEXTO



BOTÓN Y ORGANIZACIÓN | 06

marcamos **[No hay etiqueta de rótulo]**. Finalmente, presionamos **[Aceptar]**.

PASO | 04 Repetiremos los pasos 2 y 3 dos veces más para agregar las etiquetas y los campos correspondientes a los datos Apellido y E-mail, recordando que el campo **[ID]** debe ser igual al que hemos escrito en el interior de la etiqueta **<label>**.

PASO | 05 Vamos otra vez al menú **[Insertar]**, seleccionamos **[Formulario]** y escogemos **[Área de texto]**. En el cuadro que se abre, dentro de **[ID]** escribimos **[mensaje]** y hacemos clic en **[Aceptar]**.

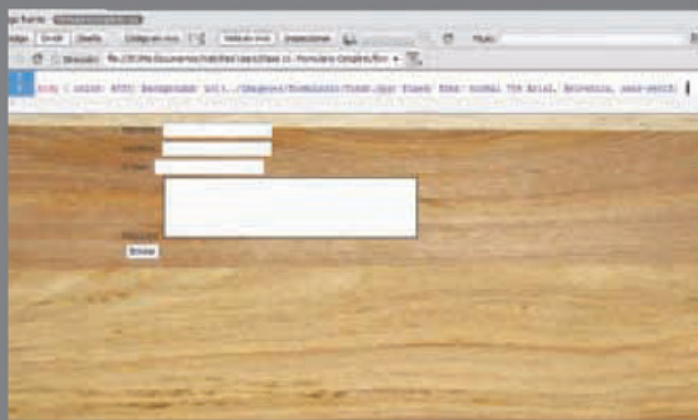
PASO | 06 Para terminar con el código HTML, dentro del menú **[Insertar/Formulario]** elegimos **[Botón]** y, en la ventana que se

abre, escribimos **[enviar]** dentro de **[ID]**. Cuando se inserte el botón, debemos verificar que en la barra **[Propiedades]** diga **[Enviar]** dentro del campo **[Valor]**, y que esté seleccionada la opción **[Enviar formulario]** en **[Acción]**. Luego, creamos un div **[#remiteente]** para colocar, dentro de él, los campos generados anteriormente (nombre, apellido y mail). Entre cada uno de ellos ponemos una etiqueta **
** y, para terminar, creamos un div **[#escribir]** donde ubicaremos el cuadro del mensaje.

FORMULARIO COMPLETO

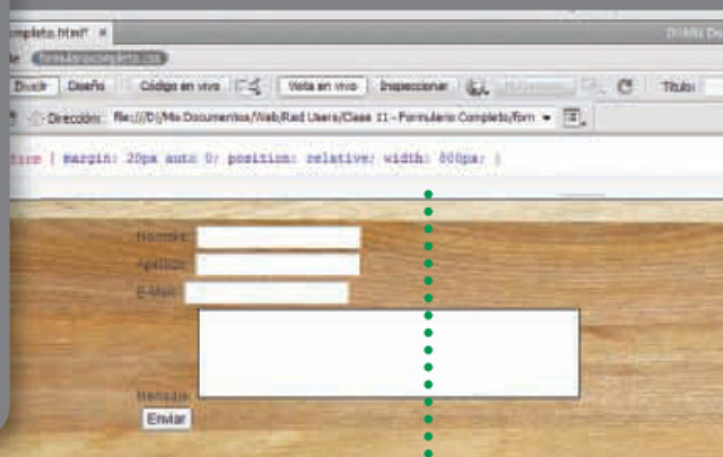
PASO A PASO

07 | ESTILO BODY

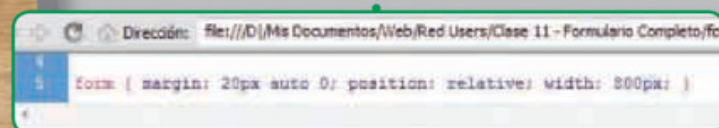


ESTILO FORMULARIO

08



09 | DATOS PERSONALES



PASO | 07 Le damos estilo al body mediante `[body {color: #333; background: url(../imagenes/formulario/fondo.jpg)`

`fixed; font: normal 75% Arial, Helvetica, sans-serif;}]`. Con esto, elegimos su color, le aplicamos un fondo que, si se viera fuera de cuadro, se arreglaría gracias a `[fixed]`, y definimos las características de la tipografía.

PASO | 08 Definimos la etiqueta `<form>` con `[form {margin: 20px auto 0; position: relative; width: 800px;}]`. Así, establecemos los márgenes: superior `[20px]` y laterales `[auto]` para que el formulario quede centrado, e inferior `[0px]`. A continuación, establecemos la posición (que, en este caso, es relativa al tamaño total del formulario) y el ancho.

PASO | 09 Damos estilo al div `#[remiteinte]` mediante `#[remiteinte {background:`

`transparent url(../imagenes/formulario/campos.png) no-repeat top left; height: 282px; padding: 30px 30px 30px 40px; position: absolute; left: 0px; top: 30px; width: 467px;}]`. Aquí definimos el fondo con la propiedad `[transparent]`, ya que el archivo png utilizado es transparente. Además, definimos el estado `hover` del div de la siguiente manera: `#[remiteinte: hover { z-index: 1000;}]`. Esto permitirá que, cuando estemos sobre el div `#[remiteinte]`, éste se vea por delante de todo lo demás.

PASO | 10 Definimos el div `#[escribir]` con `#[escribir {background: transparent url(../imagenes/formulario/mensaje.png) no-repeat top left; height: 353px; padding: 20px; position: absolute; top: 50px; right: 0px; width: 337px; z-index: 500;}]`.

Aquí definimos el eje Z (para que quede superpuesto al div `#[remiteinte]`).

```

#mensaje {
background: transparent url(../imagenes/formulario/mensaje.png) no-repeat top left;
height: 353px;
padding: 20px;
position: absolute;
top: 50px;
right: 0px;
width: 337px;
z-index: 500;
}

```

10 | CUADRO DE MENSAJE

ESTILOS VARIOS

12

11 | BOTÓN MEJORADO

TIPOGRAFÍA DEL MENSAJE

13

PASO | 11

Para definir el botón **[Enviar]**, escribimos **[#enviar {background: transparent url(../imagenes/formulario/btn_enviar.png); height: 163px; position: absolute; top: 280px; left: 150px; width: 168px; z-index: 2000;}]**. Como la función **hover** de nuestro **div** **[#remiteente]** tiene un eje **Z** definido en **1000** (recordemos que éste corresponde a la profundidad), definimos el eje **Z** de nuestro botón para que quede por encima de ella. En el código HTML del botón, dentro de **[Value]**, debemos colocar **[]** (esto significa un espacio) para que no quede superpuesta la palabra **Enviar**.

PASO | 12

Definimos las etiquetas **[textarea]** e **[input]** mediante **[textarea, input {background-color: transparent; margin-bottom: 10px;}]**, y **<label>** con **[label {display: block;}]**. Definimos estas etiquetas de forma conjunta, ya que tendrán algunos valores comunes.

PASO | 13

Para finalizar, definimos otros detalles de la etiqueta **[textarea]** de la siguiente manera: **[textarea {color: #C00; font: normal italic 18px Georgia, "Times New Roman", Times, serif; width: 330px;}]**. Así establecemos el color y la familia tipográfica que se mostrará mientras el usuario escribe su mensaje, además del ancho del área.

RESUMEN

Aquí pudimos ver cómo elaborar un formulario con imágenes e interacción entre los **divs**. Tengamos en cuenta que los recuadros que muestra Dreamweaver –por ejemplo, el transparente alrededor de ciertas imágenes, como la del botón **[Enviar]**– no se verán en el navegador.

PHP

FUNDAMENTOS

Aquí veremos los conceptos básicos del lenguaje de servidor más conocido: PHP. Aprenderemos qué es, para qué sirve y cómo funciona este lenguaje cuyo principal uso es tomar las variables de un formulario y enviarlas por e-mail.

PHP es un acrónimo recursivo que significa **PHP Hypertext Pre-processor**. Su creación se remonta al año 1995, y como todo lenguaje, pasó por diferentes versiones, algunas más populares que otras. La más reciente es la 5.3.3 (lanzada en julio de 2010), aunque una de las más usadas fue la 4.4.8. Si bien la diferencia entre ellas se concentra en funciones avanzadas que no veremos aquí, en nuestros ejemplos asumimos que el trabajo se realizará en servidores con la versión 5.2.6 o superior.

PHP es uno de los lenguajes más difundidos porque, prácticamente, todos los sistemas operativos para servidores lo soportan, aunque entre ellos el de mayor uso y robustez es Linux. Podemos incluir el código PHP dentro del HTML de la página en la que estemos trabajando. Para indicar que iniciaremos una secuencia de código PHP, usamos la etiqueta `<?php` y, para señalar su cierre, la `?>`. Cuando incorporamos código PHP al HTML, es fundamental almacenar el archivo con extensión PHP, ya que ésta es la manera en que el servidor reconocerá el tipo de página y podrá interpretar el código.

LENGUAJES DE SERVIDOR

Estos lenguajes son los que se interpretan y ejecutan en el lugar donde está

```

1 <html>
2 <head>
3 <title>Mi primer PHP</title>
4 </head>
5 <body>
6 <?php
7 $miVariable="Hola Mundo!";
8 echo $miVariable;
9 ?>
10 </body>
11 </html>
12 |

```

Dreamweaver colorea la sintaxis según el lenguaje que usemos y cada comando dentro de él para mejorar la lectura.

alojado el sitio. Cada vez que un usuario accede a una página, el servidor ejecuta las líneas de código antes de que el contenido sea mostrado en el navegador de Internet del visitante. El código de los lenguajes de servidor es inaccesible para el usuario. Como PHP es un lenguaje de este tipo, el usuario no podrá ver el código, sino que su navegador lo recibirá ya interpretado.

LENGUAJES DE CLIENTE

Son aquellos que se interpretan y ejecutan en la computadora o dispositivo del usuario. En este caso, el encargado de interpretar y ejecutar los comandos es el navegador de Internet que esté utilizando quien visita el sitio. El lenguaje de cliente más utilizado es **JavaScript**, actualmente masificado por su **framework** más conocido: **jQuery**.

PROCESO DE EJECUCIÓN EN PHP

Gracias a PHP, podremos mostrar contenido dinámico en nuestro sitio web; es decir, aquel que está almacenado en una base de datos o en variables, que a través de la interpretación de PHP, se traduce en HTML en el navegador. Este proceso es relativamente fácil de comprender, y podemos resumirlo en estos pasos:

1. El navegador de Internet del usuario solicita una página PHP determinada.
2. El servidor localiza el documento pedido y, a través del **intérprete de PHP**, ejecuta el código que esté en esa página.
3. Una vez que se ejecuta el código PHP, se genera el resultado como código HTML.
4. El resultado en HTML es enviado al usuario que solicitó la página y se muestra en su navegador de Internet.

Es importante entender que una página PHP no puede visualizarse de manera local como sí sucede con las HTML; es decir, no podemos abrirla directamente desde el navegador de Internet. Para ver una página PHP de forma correcta necesitamos tener instalado un servidor en nuestra computadora o conectarnos a uno externo.

VARIABLES

Una variable es una posición en la memoria del servidor, reservada para



FRAMEWORK

Los frameworks simplifican el desarrollo de una aplicación web mediante la automatización de funciones que se usan con frecuencia. Gracias a ellos, las operaciones complejas quedan encapsuladas para que, luego, nosotros apliquemos operaciones sencillas y fáciles de entender.



En este esquema, podemos ver cómo funciona el proceso de ejecución de un archivo PHP alojado en un servidor.

asignar cualquier valor o dato. A diferencia de otros lenguajes de programación, en PHP no es estrictamente necesario declarar las variables, ya que sólo debemos anteponer el símbolo **\$** al nombre de aquella que queremos utilizar. Las variables en PHP son **case sensitive**, es decir, el lenguaje diferencia entre las minúsculas y las mayúsculas. Por ejemplo, **\$miVariable=1** no es lo mismo que **\$mivvariable=1**, y PHP las interpretará como dos variables diferentes. Una variable puede almacenar distintos tipos de datos. En PHP, las opciones son las siguientes:

- ▶ **Integer:** números enteros positivos y negativos.
- ▶ **Double:** números decimales positivos y negativos.
- ▶ **String:** cadenas de texto.
- ▶ **Boolean:** valores true o false.
- ▶ **Array:** tipo de variable de múltiples posiciones.
- ▶ **Object:** tipo especial de dato complejo.

VARIABLES DEFINIDAS

En PHP existen diferentes tipos de variables predefinidas por el lenguaje que utilizaremos para manejar la información

en las distintas páginas de un sitio. Las dos que más utilizaremos son **[\$_GET]** y **[\$_POST]**, y las veremos con mayor detalle más adelante. Mientras tanto, a continuación veamos nuestro primer ejemplo de script en PHP, dentro del código HTML:

```

<html> <head> <title>Mi
primer PHP</title> </head>
<body> <?php
$miVariable="Hola Mundo!";
echo $miVariable; ?> </body>
</html>

```

En el ejemplo anterior vemos que la palabra **[miVariable]** comienza con el símbolo **\$**. Como mencionamos anteriormente, de esta forma le estamos indicando a PHP que se trata de una variable. También observamos que al final de cada línea hay un punto y coma **;**, para indicar a PHP que allí finaliza una línea de código. Por último, la función **[echo]** sirve para mostrar en pantalla el contenido de una variable o una cadena de texto.

CUANDO INCLUIMOS
CÓDIGO PHP, LOS
ARCHIVOS NO PUEDEN
TENER EXTENSIÓN HTML,
SINO QUE DEBEN SER PHP.

El formulario de contacto que encontramos en www.jud-ecamo.com está realizado en PHP.

jud Ecamo
Constructeur français de matériel de façonnage

Nous contacter
Formulaire contact

Rechercher un produit

nom* Prénom*

Rue/Bois

Code postal Ville Pays

Téléphone* Fax*

E-mail*

Site web

☐ Accepte de recevoir, par courrier électronique, des informations concernant votre société.

SV JUD **JUD ECAMO**

JUMP! Interactivo
Primera Jornada de Publicidad Interactiva
UN SALTO A LOS NUEVOS MEDIOS

31 de Agosto 2010
Carmelitas Center
14 a 20 hs.
Asunción • Paraguay

INICIO EL EVENTO DISERTANTES SPONSO

PRE-INSCRIPCION

Pre-inscríbete a Jump! Interactivo completando el siguiente formulario, y nos comunicaremos con usted para coordinar los detalles posteriores.

O bien, llame a los teléfonos: (021) 623 438 - (0971) 347 111.

Nombre y Apellido

E-mail

Empresa

Teléfono

Celular

Cantidad de entradas

Profesional ☐ Estudiante (Precisamos recibo de último pago de cuota) ☐

¿Desea un cobrador a domicilio? SI ☐ NO ☐

Dirección

ENVIAR

En www.jumpinteractivo.com se utiliza PHP para crear formularios de registro, como el que observamos.

comienzan en 0 (cero), lo que significa que uno que contenga diez posiciones irá de la posición cero a la nueve.

FUNCIONES

Las herramientas más importantes de cualquier lenguaje de programación son las funciones. En PHP, podemos definir nuestras propias funciones y, también, utilizar muchas que ya están definidas por el lenguaje. Una que usaremos muy a menudo es `[mail()]`, que, como su nombre lo indica, sirve para enviar contenido por correo electrónico. Se define de la siguiente manera: `[mail ($destinatario,$asunto,$mensaje);]`.

VARIABLES ENTRE PÁGINAS

Muchas veces, nos encontraremos ante la necesidad de transmitir variables de una página a otra. Para hacerlo, existen los métodos **POST** y **GET**. Esta transmisión de una variable entre páginas se conoce, en la jerga, como “pasar una variable”. Para pasar una variable por **GET**, debemos indicarla en la URL de la página a la que estamos accediendo. La primera variable va precedida del carácter **?** (símbolo de interrogación), y a partir de la segunda, se emplea **&** (ampersand). Por ejemplo: `miPagina.php?nombre=Juan&apellido=Perez`. En este caso, el archivo `miPagina.php` recibirá por **GET** la variable **nombre**, cuyo valor es **Juan**; y **apellido**, cuyo valor es **Perez**. Luego, dentro del código de `miPagina.php`, por medio de PHP interpretamos esas variables que estamos pasando, utilizando la variable definida `$_GET`, que actúa como vector. Vemos entonces que `$_GET` es, en realidad, un vector cuyas claves o keys son las diferentes variables que le hemos pasado a la página.

```
<?php $nom=$_GET["nombre"]; $ape=$_GET["apellido"]; ?>
```

El uso de **POST** es similar, sólo que para pasar variables no utilizamos la URL de la página de destino sino un formulario. Luego, interpretamos esas variables con

ESTRUCTURAS DE CONTROL

La estructura de control que utilizaremos con mayor frecuencia es **if** (en inglés, el condicional si). Ésta analiza una condición y, en caso de que se cumpla, ejecuta un script; de lo contrario, ejecuta otro. Por ejemplo, el código que se muestra a continuación se traduce de la siguiente manera: si la variable “edad” es mayor o igual al valor 18, mostrar la frase “Eres mayor de edad”; y si la variable edad no cumple con la condición mencionada, mostrar la frase “Eres menor de edad”.

```
<?php if($edad>=18){
    echo "Eres mayor de edad"; }else{ echo "Eres menor de edad"; } ?>
```

INCLUDES

En muchas ocasiones, tendremos la necesidad de incluir código desde un archivo externo, que puede ser PHP o HTML. Esto sucede, por ejemplo, cuando precisamos incluir en todas las páginas de un sitio un archivo HTML que contiene un menú de navegación.

```
<body> <?php
include("menu.html");
?> </body>
```

La ventaja de utilizar **[include]** es que, si necesitamos realizar un cambio sobre el archivo agregado, no tendremos que hacerlo más de una vez. Siguiendo el ejemplo, si queremos agregar una opción al menú, no hará falta modificar cada una de las páginas HTML, dado que bastará con hacerlo en la página `menu.html`.

VECTORES

Aunque el empleo de vectores es un tema complejo, debemos mencionar que un array (por vector, en inglés) es una variable que contiene múltiples posiciones para almacenar datos. Para acceder a los datos guardados en un vector, existen dos formas. La primera consiste en indicar el índice del vector; por ejemplo, `[$miVector[0]]`. La segunda es indicar la clave (o key) del valor al que queremos acceder, por ejemplo, `[$miVector["dato"]]`. Las posiciones de los vectores siempre

la variable definida **\$_POST**. Veremos que **\$_POST** también es un vector cuyas claves son las diferentes variables que le hemos pasado a la página a través de un formulario.

```
<?php $nom=$_  
OST["nombre"]; $ape=$_  
POST["apellido"];?>
```

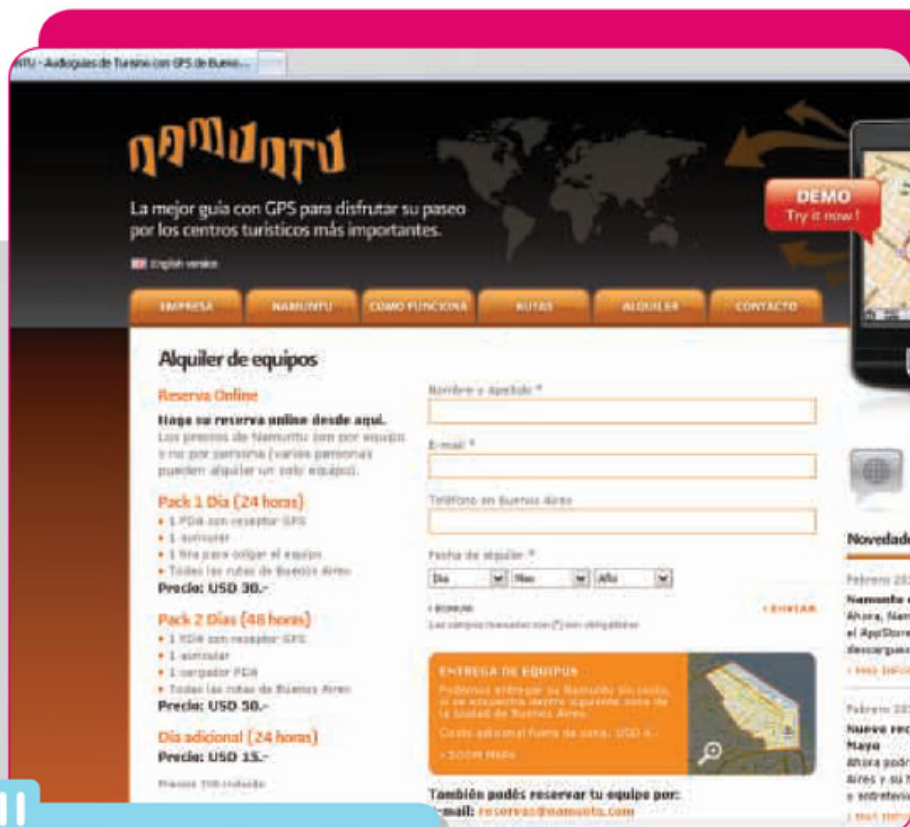
El uso de **POST** es más seguro y discreto, ideal para la transmisión de datos, por ejemplo, de un formulario de contacto. En cambio, **GET** se usa para pasar pocas variables y de información poco relevante para el usuario.

CONCATENAR

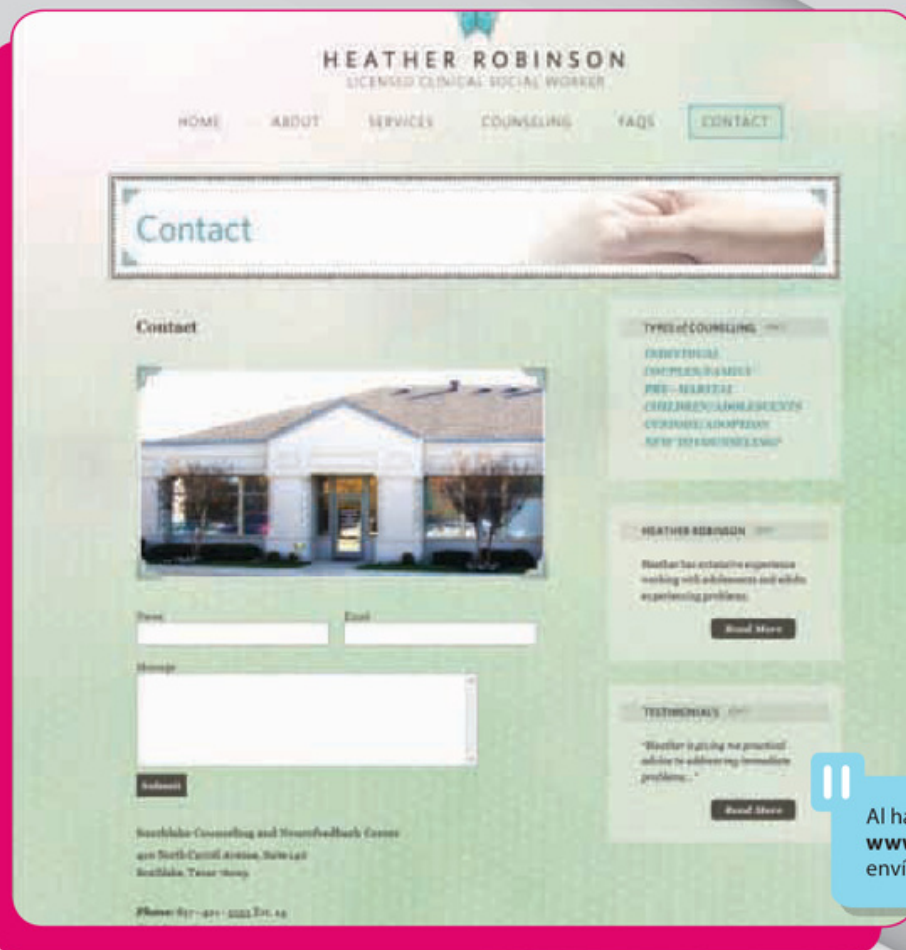
El carácter **.** (punto) se utiliza para concatenar cadenas de texto en PHP. Por ejemplo, con el siguiente código obtenemos como resultado el texto **20/10**:

```
<? $dia=20;  
$mes=10; echo  
$dia."/".$mes; // ?>
```

El sitio www.namuntu.com utiliza un formulario para reservar equipos. Los datos llegan por e-mail.



Hasta aquí llegamos con los primeros pasos en PHP, un lenguaje extenso y con infinitas posibilidades. Para estudiar y conocer más sobre él, es aconsejable visitar su sitio, www.php.net, donde encontraremos novedades y un manual completo con todas sus funciones. ◀



Al hacer clic en el botón **[Submit]** de www.hrobinsoncounseling.com, se envía la información a **[Mailer.php]**.

HACER FUNCIONAR LOS FORMULARIOS

Al trabajar con formularios, una vez que se los ha diseñado y maquetado, éstos deben funcionar. Sin embargo, es necesario incorporar algo de programación para que eso suceda.



Cuando terminamos de diseñar y armar un formulario, éste no funcionará por sí solo, sino que deberemos agregar las instrucciones correspondientes para que, efectivamente, ocurra la acción deseada. Preguntarse cómo hacer funcionar un formulario es muy frecuente al comenzar a elaborar este tipo de páginas. La manera más común de hacerlo es asociar un archivo PHP o ASP en el atributo **[action]** del formulario. El código constará de cuatro partes básicas:

- Obtener o recuperar la información del formulario.
- Elaborar el texto del mensaje.
- Definir la cabecera del e-mail.
- Enviar el e-mail.

LAS VARIABLES SON POSICIONES DE MEMORIA QUE GUARDAN LA INFORMACIÓN HASTA QUE ÉSTA SE TERMINA DE PROCESAR.

En PHP, para recuperar u obtener la información del formulario utilizaremos POST, que es el sistema de envío de datos. La orden usada es **[\$_POST]**, y el archivo PHP es el que se encuentra en el **Código 1**. Los datos del formulario que obtenemos mediante **[\$_POST]** se guardan en

variables. Éstas son posiciones de memoria que mantendrán la información hasta que se la haya terminado de procesar. Por esta razón, la mejor manera de asignar un nombre al campo es que lleve el mismo al que hará referencia la variable. Por su parte, **[\$mensaje]** es la variable donde se guardarán las distintas partes del formulario, siempre seguidas de un salto de línea para que se muestre de manera ordenada en el e-mail. Además, tenemos las variables **[\$para]** y **[\$asunto]**: la primera indica a dónde debe enviarse el e-mail; y la segunda, el nombre del asunto que llevará el correo.

DEFINIR LA CABECERA DEL E-MAIL

Después, generamos las cabeceras del correo electrónico que se va a enviar, ya que es conveniente que aparezcan al

CÓDIGO 1

```
<?php $para      = "mimail@misite.com.ar"; $asunto
to      = "Tu Web".$_POST['asunto']; $mensaje  .= "Nom-
bre: ".$_POST['Nombre']."\r\n"; $mensaje  .= "E-Mail:
".$_POST['Email']."\r\n"; $mensaje  .= "Teléfono: ".$_
POST['Telefono']."\r\n"; $mensaje  .= "Consulta: ".$_
POST['Mensaje']."\r\n"; $cabeceras = 'MIME-Version: 1.0' .
"\r\n"; $cabeceras = 'Content-type: text/
html; charset=iso-8859-1' . "\r\n"; $cabeceras = 'From:
'.$_POST['email'] . "\r\n" . 'Reply-To: '.$_POST['email'] .
"\r\n" . 'X-Mailer: PHP/' . phpversion(); @mail($para,
$asunto, $mensaje, $cabeceras); header('Location:
misegundapagina.html'); ?>
```



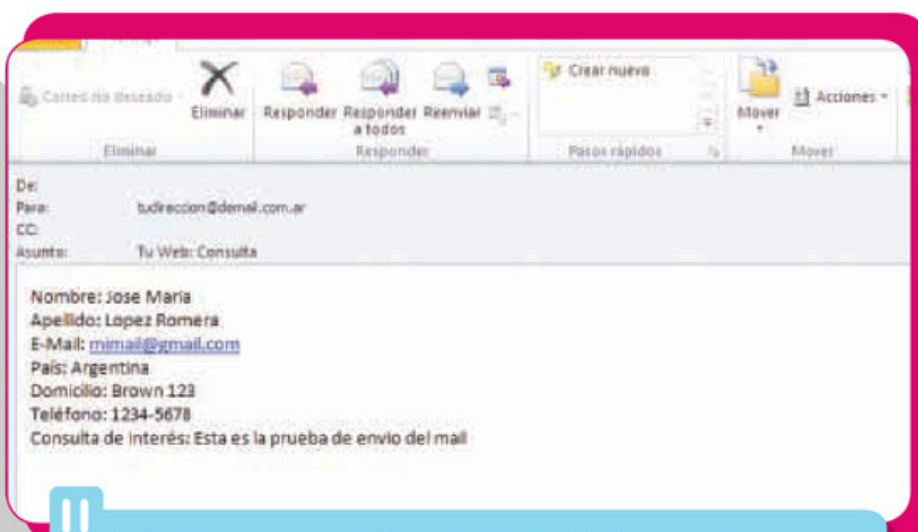
+ ASP O PHP

Para saber qué archivo utilizaremos para enviar el e-mail, es decir, si será PHP o ASP, debemos conocer en qué tipo de servidor se encuentra alojado el sitio web. Si es Linux, el archivo de envío de e-mail será PHP. En cambio, si es Windows, será ASP. Si el servidor es Linux, además deberemos tener en cuenta la versión de PHP alojada en él, ya que de esto dependerá la forma en la que el e-mail será mandado. Si la versión no es la última, antes de escribir `[r\n]` es recomendable colocar una etiqueta `<br/>` para generar el salto de línea. Por el contrario, si la versión de PHP es la más reciente, es mejor evitar esa etiqueta, porque, de no hacerlo, aparecerá en el cuerpo del correo.

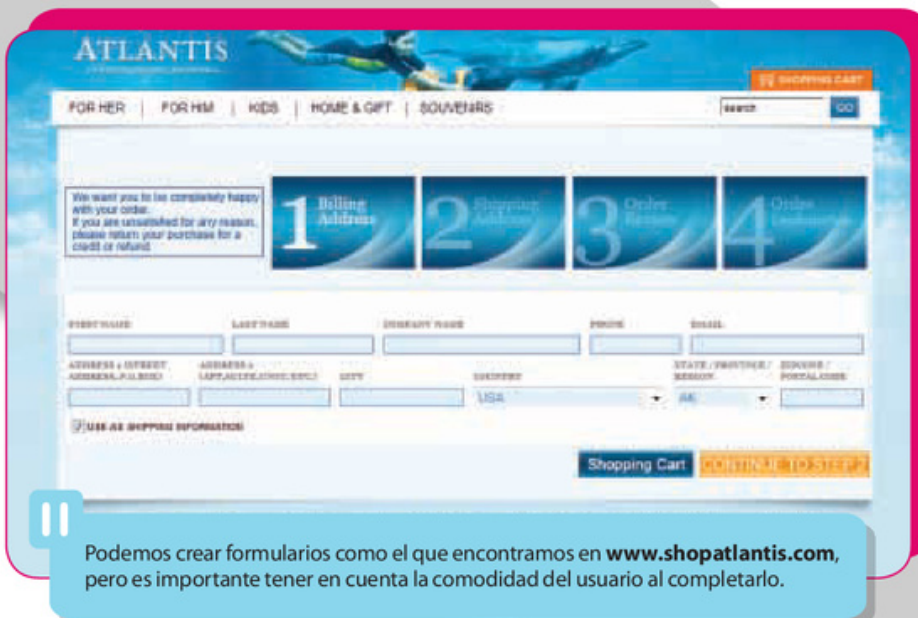
menos los datos mínimos para identificar la versión del e-mail, el programa que lo generó y quien lo envía, entre otros detalles. Esto es lo que tenemos en nuestras variables `[$cabeceras]`, que conformarán la cabecera del mensaje. La primera cabecera marcará la versión de mail, y la segunda, el tipo de contenido que éste contendrá. La tercera se divide en tres partes: `[FROM]` indica desde dónde se envía el correo; `[Reply-To]` indica a dónde debemos contestar el mail; y `[X-Mailer]` es el programa que se encargará de enviar el mensaje. Todos estos datos están separados mediante `[r\n]`, que significa salto de línea.

ENVIAR EL E-MAIL

Para finalizar, utilizamos `[@mail]`, la orden que sumará las variables y mandará el correo; y a continuación, `[header]`, para redireccionar al usuario a un archivo HTML en donde mostraremos el clásico texto "Su formulario se ha enviado con éxito". Siempre que generamos un formulario, debemos encargarnos de tener herramientas que nos permitan probarlo de manera offline, ya que esto nos servirá para ahorrar tiempo si el formulario no funciona como es debido. Para cumplir esta tarea, una buena opción es el programa de distribución gratuita llamado Test Mail ServerTool (www.toolheap.com), que emula al servidor. De este modo, es posible simular el envío del e-mail mandándolo a un archivo local en nuestro equipo. ◀



Aquí podemos observar, en Microsoft Outlook 2010, la forma en la que recibimos el correo electrónico enviado por el formulario que creamos.



Podemos crear formularios como el que encontramos en www.shopatlantis.com, pero es importante tener en cuenta la comodidad del usuario al completarlo.



Las páginas principales de Facebook (www.facebook.com) y de LinkedIn (www.linkedin.com) son, principalmente, formularios de registro.

ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 ¿Para qué sirven los formularios?

02 ¿Qué son las **webapps**?

03 Según su complejidad, indique las clases de formularios.

04 ¿Según su finalidad, cómo se clasifican los formularios?

05 Mencione los usos que puede tener la etiqueta **<input>**.

06 ¿Qué nos permite la etiqueta **<hidden>**?

07 ¿Qué es **PHP**?

08 ¿Qué son las variables en **PHP**?

09 ¿Qué son los vectores en **PHP**?

10 ¿Qué es concatenar?



PRÁCTICAS

01 Busque en Internet diferentes tipos de formularios, analice su tipología y pruebe su funcionamiento.

02 Cree un formulario básico de contacto.

03 Agregue un **botón Image** a su formulario.

04 Incluya un fondo en su formulario.

05 Defina que el borde transparente del **botón enviar** no se vea en el navegador.





Proyecto real I



- ▶ PRESENTACIÓN DEL SITIO
- ▶ SITIO HTML
- ▶ TOMAR MEDIDAS Y DEFINIR EL LAYOUT
- ▶ DAR FORMA AL LAYOUT CON CSS
- ▶ MENÚS DE NAVEGACIÓN
- ▶ ARMADO DEL FOOTER
- ▶ FORMULARIO DE PRESUPUESTO
- ▶ CREAR LA PÁGINA DE INICIO



PRESENTACIÓN DEL SITIO

A continuación, vamos a conocer el sitio web en el que nos basaremos para nuestro aprendizaje. Éste es el desarrollo con el que aprenderemos en esta obra y que nos servirá en la creación de un portfolio personal.

El objetivo principal de este sitio es mostrar el trabajo de un diseñador; por lo tanto, todo el contenido y las diferentes secciones van a ser pensados en función de esto. Está diseñado en Photoshop, maquetado en Dreamweaver y validado por la W3C. Es un sitio pensado para que su indexación en los buscadores sea la ideal. ◀

1 En la página principal (home), mediante una serie de *sliders*, vamos a mostrar algunos de los diseños más importantes que hayamos hecho. Por otro lado, tendremos un listado de los principales servicios que ofrecemos.

2 Como queremos comunicar rápidamente lo que hacemos, no es conveniente extenderse y agregar muchos párrafos explicativos. Es mejor incorporar un título contundente y algún texto destacado donde aparezca un enlace del estilo "leer más".

3 Servicios: aquí sí vamos a profundizar en los servicios que brindamos, ya que si el usuario está interesado, se tomará el tiempo necesario para conocernos y ver si ofrecemos lo que está buscando.



SITIO MODELO

El proyecto que creamos en este libro es un sitio web modelo que podrán utilizar nuestros lectores para promocionar y ofrecer sus servicios en Internet de manera efectiva. Este sitio respeta los mejores métodos de desarrollo web, y tiene en cuenta conceptos de usabilidad e interfaz que lo hacen rápido y sencillo de usar.

4 Portfolio: esta sección debe ser el área más fuerte de todo el sitio, porque el usuario va a comprar según lo que vea en ella. Lo que se muestra aquí es un ejemplo de lo que somos capaces de hacer.

5 Vamos a categorizar el contenido en web, print, ilustración e identidad, para que el usuario pueda hacer foco en la rama que le interese sin perder demasiado tiempo.

6 En un portfolio, es usual mostrar una imagen pequeña, representativa del trabajo, más un título y un texto breve. Si el usuario quiere ver más información sobre este diseño en particular, puede acceder a su página de detalle.

7 Acá podemos visualizar los datos más relevantes, como el nombre del cliente, el tipo de trabajo que se hizo y la URL, si es un sitio web (la dirección será un enlace que permitirá acceder al sitio online). Además, podemos agregar todo lo que creamos conveniente y que aporte datos interesantes al trabajo.

8 Contacto: esta sección es fundamental; no debe faltar en nuestro sitio y es importante que sea bien visible. Desde aquí, los potenciales clientes podrán comunicarse con nosotros.



9 En el footer, es de suma importancia ofrecer los medios alternativos a través de los cuales pueden contactarse con nosotros, como Facebook, LinkedIn y Twitter. También resulta útil incorporar un formulario, además de teléfono y dirección.

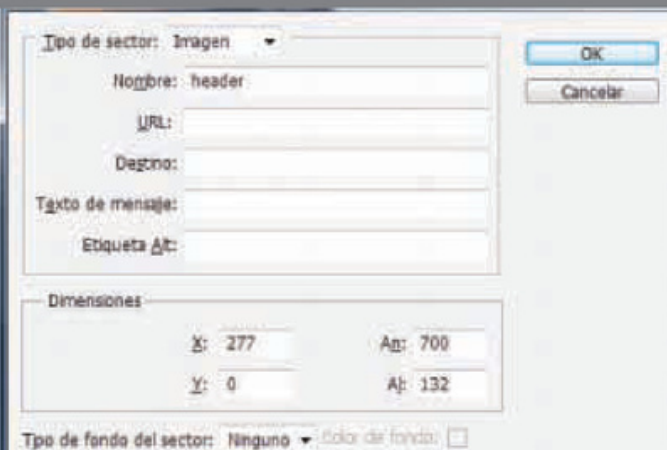
TOMAR MEDIDAS Y

PASO A PASO

En primera instancia, debemos determinar cómo conviene establecer el layout.



01 | HEADER



MEDIDAS DEL HEADER | 02

Una vez que tenemos abierto en Photoshop el archivo PSD con el diseño del sitio, hacemos el análisis de las partes que conformarán el layout. En esta etapa, tomamos las medidas del header, del content y del footer. Con ellas, podremos delimitar las áreas en nuestra página HTML, creando zonas específicas para cada una de las partes. ◀

PASO | 01 Con la herramienta **[Sector]**, marcamos el área que ocupa el header en nuestro sitio. En este caso, dicha zona tendrá el mismo ancho que el contenido principal del sitio, e incluirá el logo, el menú de navegación y los botones **[Agregar a Favoritos]** e **[Idioma]**.

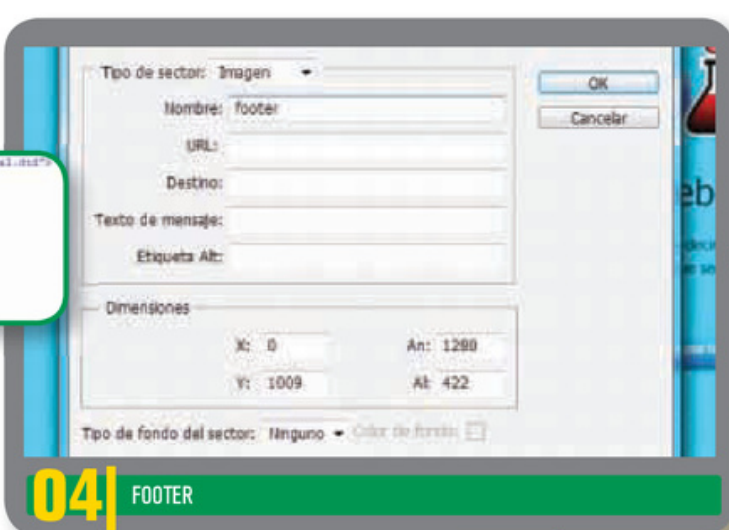
PASO | 02 Para saber las medidas de este sector (en inglés, *slice*), hacemos doble clic en él y, en la ventana **[Opciones de sector]**, observamos el alto y el ancho. En este caso, son 700 píxeles de ancho por 132 píxeles de alto.

PASO | 03 El sector del contenido deberá tener el mismo ancho que el del header, pero en este caso, no importa la altura, porque ésta variará según la cantidad de contenido que incluyamos en esta zona en cada una de las secciones del sitio.

```

<DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>USERS Web Design</title>
</head>
<body>
</body>
</html>

```



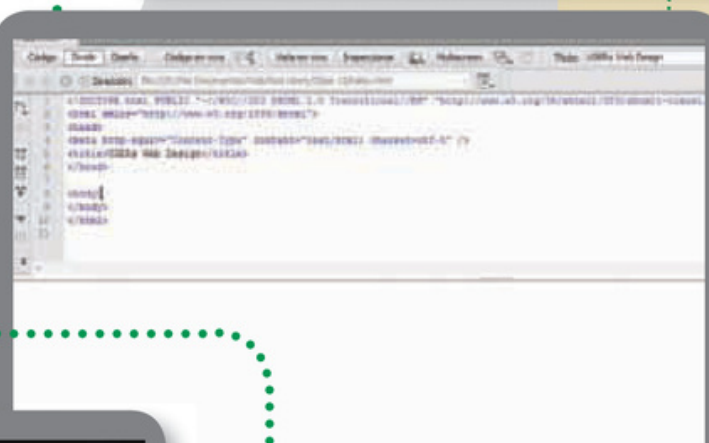
04 | FOOTER

DEFINIR EL LAYOUT

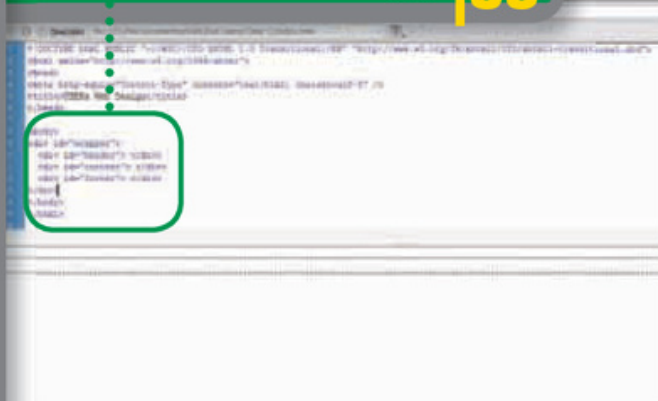
```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>USERS Web Design</title>
</head>
<body>
<div id="wrapper">
<div id="header">
<div id="content">
<div id="footer">
</div>
</div>
</div>
</body>
</html>

```



TÍTULO | 05



LAYOUT | 06

03 | CONTENIDO



PASO | 04

Finalmente, al marcar el sector del footer, debemos tener en cuenta, principalmente, su altura (en este caso, 422 píxeles), ya que ocupará todo el ancho del documento, para que luego se ajuste al ancho del navegador.

PASO | 05

Creamos una carpeta en nuestro equipo para colocar todos los archivos del proyecto y abrimos Dreamweaver, donde generamos un nuevo archivo HTML llamado **index.html**; lo guardamos dentro de la carpeta creada anteriormente. Éste será el archivo principal del sitio, donde definiremos su estructura por medio de diferentes etiquetas HTML. Primero, dentro de **<title>** escribimos el título del sitio: **USERS Web Design**.

PASO | 06

Para comenzar a armar la estructura del layout en el body del documento, creamos el div **[wrapper]**, que será el

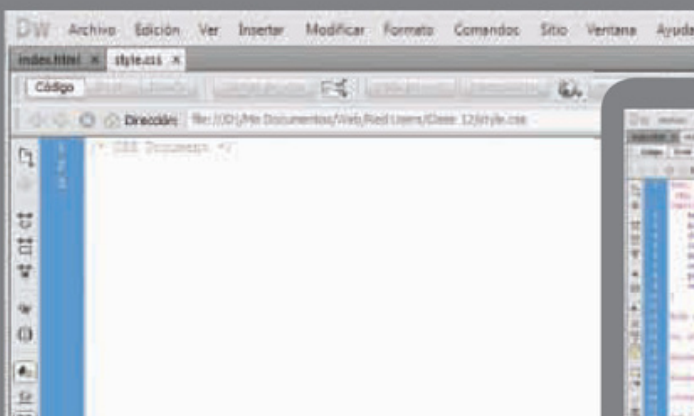
contenedor general de todo el material del sitio. Dentro de él, incluimos las etiquetas correspondientes a **[header]**, **[content]** y **[footer]**.



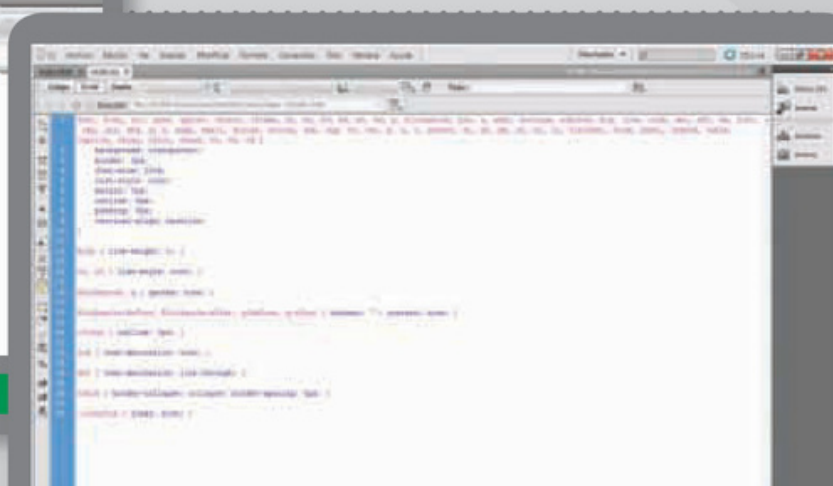
DAR FORMA AL LAYOUT

PASO A PASO

Como ya vimos, las reglas CSS nos permiten establecer las características del layout.



07 | STYLES.CSS



RESET.CSS | 08

PASO | 07 Creamos un nuevo archivo de hojas de estilo desde **[Archivo/Nuevo...]** y lo guardamos con el nombre **style.css** dentro de una carpeta llamada **css**, que generamos dentro de la principal del proyecto. Éste será el archivo que contendrá los estilos utilizados en el sitio.

PASO | 08 Creamos otro archivo CSS y lo almacenamos como **style.css** dentro de la misma carpeta. Éste servirá para resetear los valores CSS predeterminados del navegador, como hemos visto en el **Capítulo 9**. En él incluimos las reglas que aparecen en la imagen.

PASO | 09 Volvemos al archivo **styles.css** y escribimos **@import url("reset.css");** para incluir el archivo **reset.css** en el archivo de estilos principal.

Los pasos siguientes nos permitirán ajustar los tamaños y las posiciones de los diferentes sectores incluidos anteriormente. Para este fin, aplicaremos las propiedades CSS correspondientes a cada uno.

CON CSS

10 | CARGAR CSS

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>USERS Web Design</title>
6 <link href="css/style.css" rel="stylesheet" type="text/css" />
7 </head>
8
9 <body>
10 <div id="wrapper">
11 <div id="header"></div>
12 <div id="content"></div>
13 <div id="footer"></div>
14 </div>
15 </body>
16 </html>
```

```
/* Reset CSS */
body { background: #000 0 0; font-family: Arial, sans-serif; }

/* Wrapper */
#wrapper {
  background: transparent #000 repeat-x;
  width: 100%;
  margin: 0 auto;
}

/* Header */
#header {
  background: #f90;
  height: 33px;
  margin: 0 auto;
  padding-bottom: 3px;
  width: 700px;
}

/* Content */
#content {
  background: #00c;
  margin: 0 auto;
  min-height: 80px;
  padding-bottom: 3px;
  width: 700px;
}

/* Footer */
#footer {
  background: #c0c;
  height: 42px;
  margin: 0 auto;
  width: 100%;
}
```

REGLAS LAYOUT | 11

index.html x style.css x reset.css x

Código Dividir Diseño Código en vivo

Dirección: file:///D:/Mis Documentos

```
1 @import url("reset.css");
```

09 | CAMPO DE TEXTO

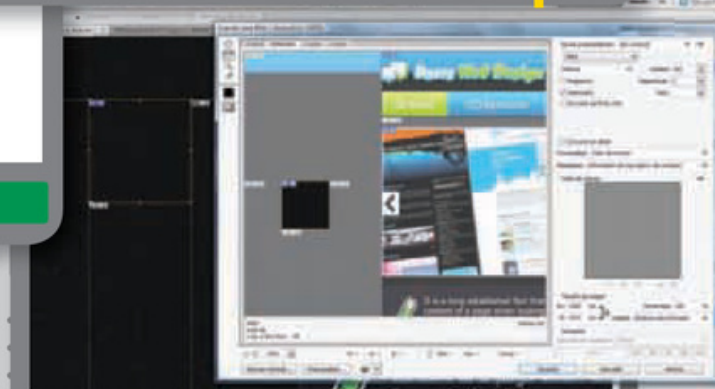


IMAGEN DE FONDO | 12

Al hacer esto, cada vez que incluyamos el archivo **style.css** en los documentos HTML, también lo hará **reset.css**. Como lo hemos incluido antes de las reglas de estilo que definiremos para el sitio, primero se resetearán los valores del navegador y, luego, se cargarán las reglas.

PASO | 10 Para cargar nuestra hoja de estilos dentro del documento **index.html**, colocamos la siguiente línea de código en la etiqueta **<head>**: **<link href="css/style.css" rel="stylesheet" type="text/css" />**.

PASO | 11 Ahora, aplicamos las reglas que vemos en la imagen para darles forma a los diferentes ids definidos en el HTML. Al hacerlo, veremos que nuestro layout se va armando, aunque sólo hemos aplicado un color de background a cada elemento

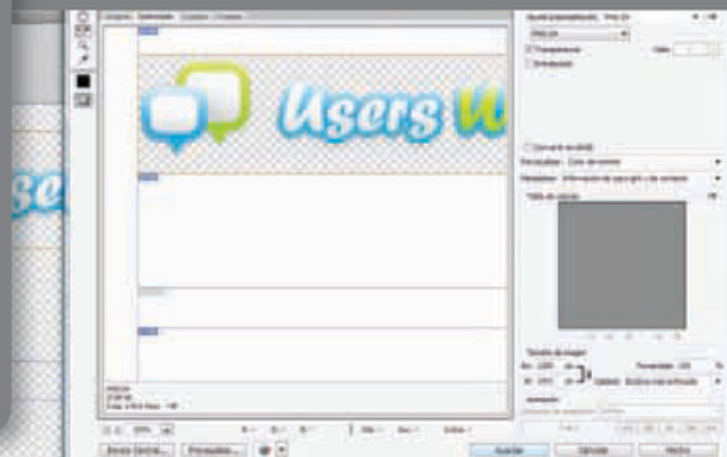
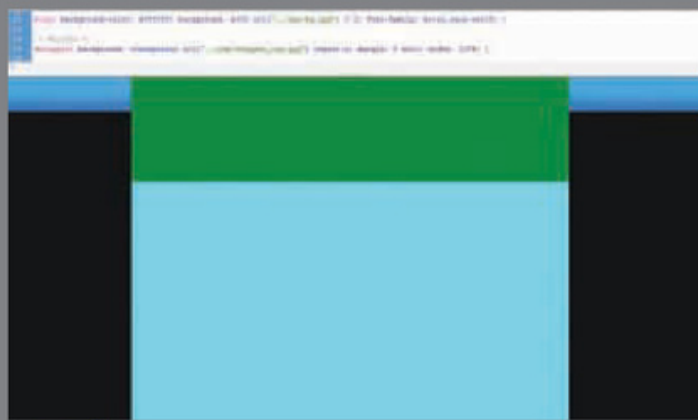
para poder diferenciarlo y, así, verificar su forma. Luego, podremos sustituir o eliminar el color según sea necesario.

PASO | 12 Ahora, para crear el background, vamos a Photoshop y marcamos un sector del fondo del sitio que hemos diseñado con una textura que compone una trama. Vamos a **[Archivo/Guardar para Web y dispositivos...]** y almacenamos el sector dentro de una carpeta llamada **img**, que creamos dentro de la principal del proyecto. En ella guardamos todas las imágenes que utilizaremos para las reglas de estilo CSS.

DAR FORMA AL LAYOUT CON CSS

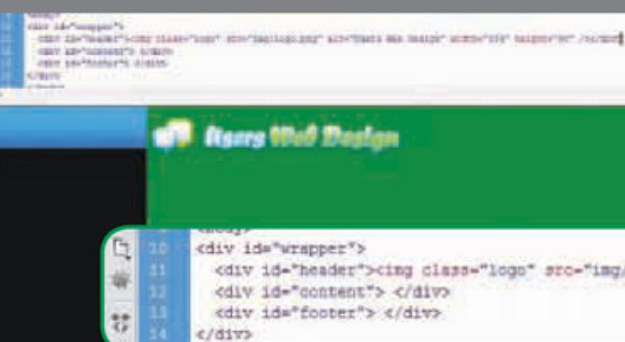
PASO A PASO

13 | FONDO WRAPPER



RECORTE LOGO

14



15 | UBICACIÓN LOGO

PASO | 13

En el archivo **style.css**, agregamos la siguiente propiedad a la regla correspondiente al **body**:

[background: #000 url("../img/bg.jpg") 0 0;]. Con esto, logramos que el archivo de imagen generado se repita en ambos ejes de coordenadas y abarque todo el fondo del sitio. Luego, para superponer la franja celeste sobre el fondo del sitio, volvemos a Photoshop y marcamos una parte de esa franja, que guardamos como **wrapper_top.jpg**. Para aplicarla al fondo de **[wrapper]**, en el archivo **style.css** escribimos **#[wrapper]{background: transparent url("../img/wrapper_top.jpg") repeat-x; width: 100%; margin: 0 auto; ;}**.

PASO | 14

Volvemos a Photoshop y creamos un sector que contenga el logo. Para utilizarlo, lo guardamos como archivo PNG de fondo transparente con el nombre **logo.png**.

PASO | 15

Incluimos el logo por medio de HTML y le asignamos una clase para poder aplicarle estilos con CSS y así ubicarlo. Dentro del div **[header]**, escribimos **** y, luego, en el archivo **style.css**, creamos la regla **.[logo]{ margin-top:10px; ;}** para que el logo esté en la posición correcta respecto del límite superior del sitio. ◀

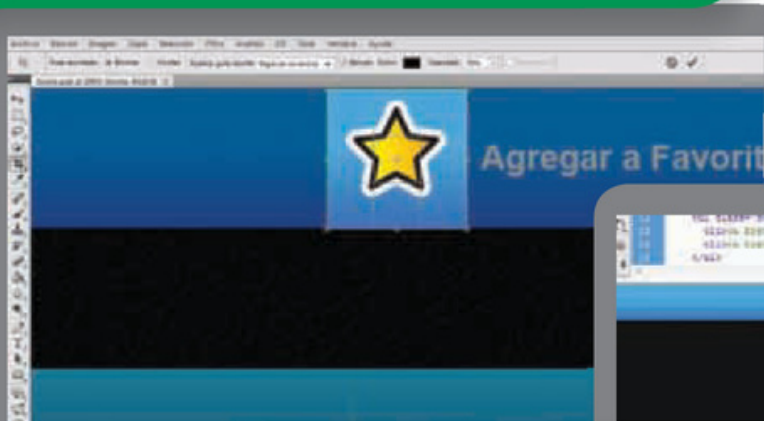


MENÚ DE NAVEGACIÓN

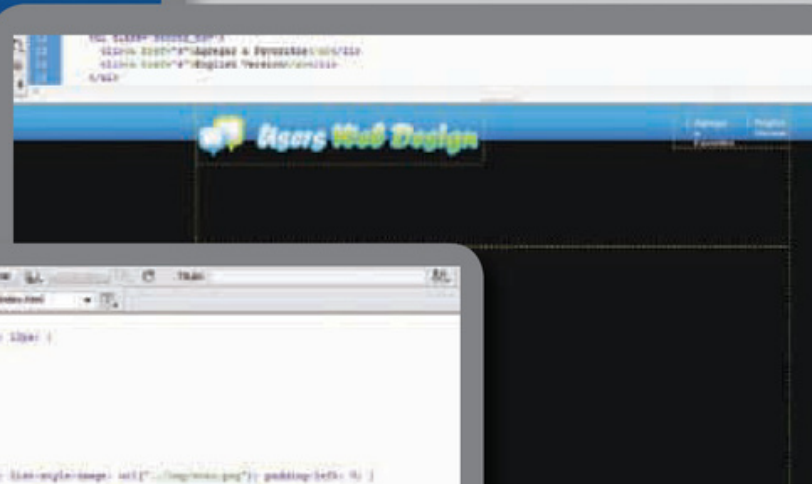
PASO A PASO

En esta etapa, crearemos los menús de navegación de nuestro sitio.

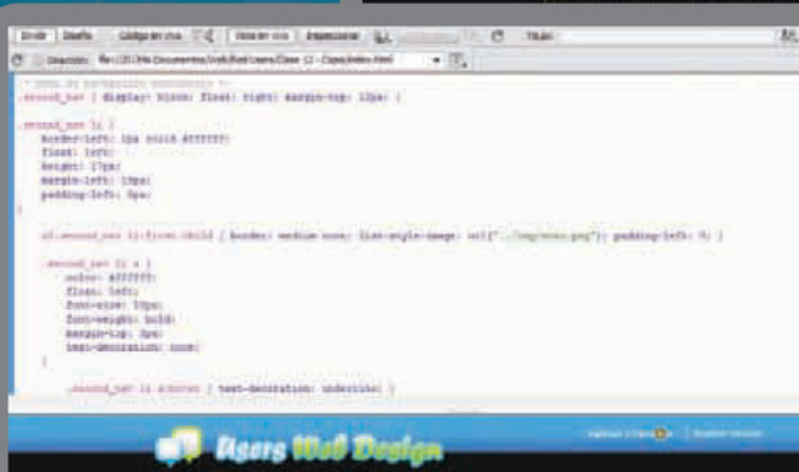
16 | FAVORITOS



MENÚ SECUNDARIO | 17



ESTILOS MENÚ SECUNDARIO | 18



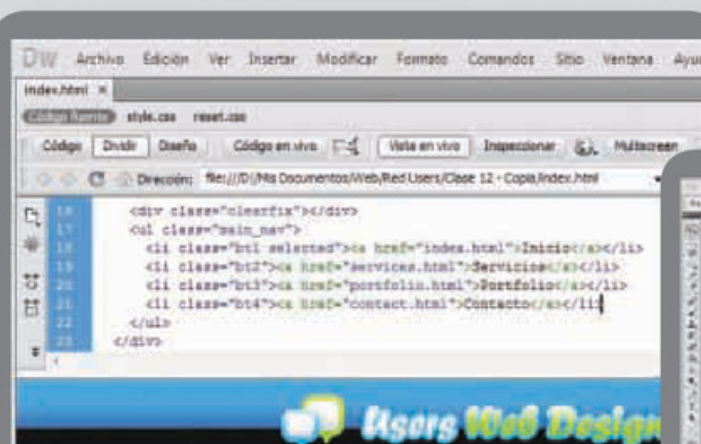
Primero, vamos a generar el menú de navegación secundario, donde se encuentran los botones para agregar a favoritos y acceder a la versión en inglés del site. Luego, crearemos el menú principal, con los botones que nos permitirán acceder a las diferentes secciones. ▶

```
<ul class="second_nav">
  <li><a href="#">Agregar a Favoritos</a></li>
  <li><a href="#">English Version</a></li>
</ul>
```

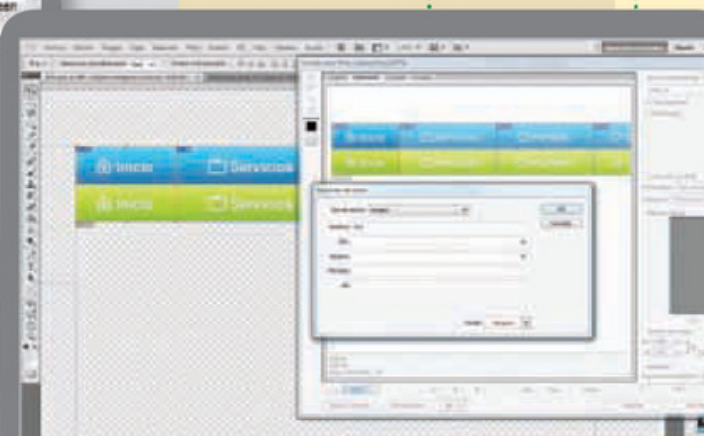
PASO | 16 En Photoshop, creamos una imagen transparente con la pequeña estrella de favoritos y la llamamos **star.png**.

PASO | 17 Luego, en el archivo HTML, escribimos lo siguiente dentro de **[header]**, a continuación del logo:

PASO | 18 Por medio del selector de clase aplicado a la etiqueta ****, vamos accediendo a los diferentes **** para dar estilos a los botones. Haremos flotar el menú secundario hacia la derecha, mientras que los diferentes **** que lo componen lo harán hacia la izquierda para estar correctamente alineados. Aplicamos estilos a los **** que componen la lista y, por medio de la pseudo-clase **[first-child]**, accedemos al primero de estos elementos y le aplicamos un estilo diferente para incluir la imagen de la estrella. Para lograr todo esto, ingresamos los estilos que vemos en la imagen.

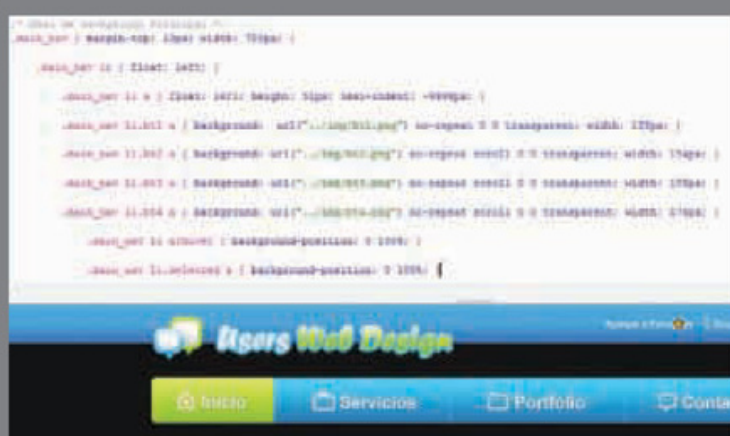


19 LISTA MENÚ PRINCIPAL



IMÁGENES PARA BOTONES

20



21 ESTILOS MENÚ PRINCIPAL



FONDO FOOTER

22

PASO | 19 Para crear el menú principal, también utilizaremos listas, incluyendo las líneas que se observan dentro de la etiqueta **[header]**, a continuación del menú de navegación secundario. Como podemos observar, cada uno de los botones del menú está formado por una etiqueta **** en la que incluiremos el enlace por medio de la etiqueta **<a>**, y a cada **** le asignaremos una clase diferente.

PASO | 20 En este caso, utilizaremos el método CSS sprites para los botones. Como ya vimos anteriormente, para usar esta técnica debemos duplicar el menú en Photoshop, de modo que queden los botones con el estado normal justo encima de los mismos botones para el estado **hover**. Luego, creamos un sector para cada par de botones y guardamos las imágenes resultantes con los nombres **bt1.png**, **bt2.png**, **bt3.png** y **bt4.png**.

PASO | 21 Ingresamos los estilos que vemos en la imagen para definir los diferentes elementos que componen el menú de navegación: definimos un margen superior para el menú, alineamos los botones, usamos la propiedad **[text-indent]** para que el texto de cada enlace no se vea en el diseño, asignamos a cada botón su imagen de fondo y ancho, generamos una clase para que la posición del background se mueva 100% hacia arriba (de esta forma, hacemos visible el estado hover por medio de CSS Sprite) y, por último, definimos otra clase para identificar la sección actual por la que estemos navegando.

PASO | 22 Para el fondo, dentro de Photoshop definimos un sector que abarque verticalmente todo el alto del footer, pero sólo una porción en sentido horizontal, para repetirlo mediante CSS. Guardamos el sector como **footer_bg.jpg** junto con las demás imágenes.

ARMADO

PASO A PASO

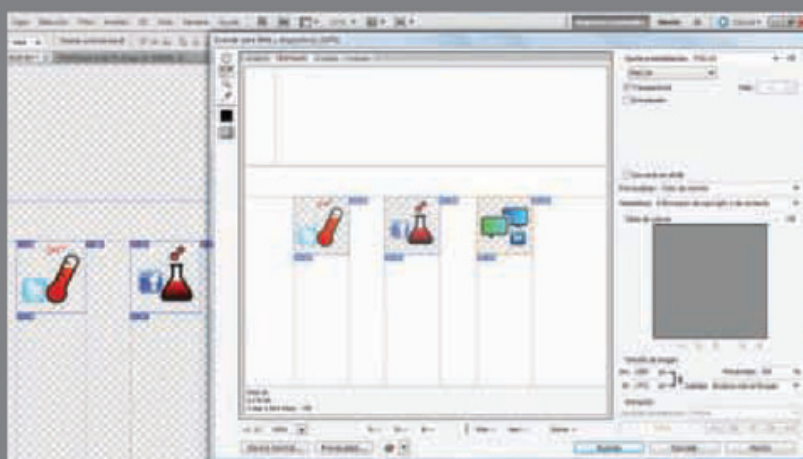
DEL FOOTER

Ahora, veremos cómo crear y definir el pie de nuestra página, donde habrá elementos fijos.

```
/* FOOTER */
#footer {
  background: transparent url("../img/footer_bg.jpg") 0 0 repeat-x;
  height: 420px;
  margin: 0 auto;
  width: 100%;
}
```

```
<div id="contenedor"></div>
<div id="footer">
  <div class="cont_footer"> </div>
</div>
```

23 | ESTILO DEL FOOTER



IMÁGENES REDES SOCIALES

24

CONTENEDOR FOOTER

25

Como vimos al crear el diseño, el footer contiene varios elementos que le permitirán al visitante ponerse en contacto con nosotros o comunicarse a través de redes sociales. Veamos los pasos para crear esta zona.

PASO | 23 Dentro del archivo de estilos, escribimos lo siguiente para definir el fondo del footer: `[#footer{ background: transparent url("../img/footer_bg.jpg") 0 0 repeat-x; height:420px; margin:0 auto; width:100%; }]`.

PASO | 24 Para incorporar las imágenes de las redes sociales, dentro de Photoshop creamos un sector para cada una de estas redes y las guardamos por separado como archivos PNG de fondo transparente. Nombramos `footer_tw.png` a la de Twitter, `footer_fb.png` a la de Facebook, y `footer_in.png` a la de LinkedIn.

PASO | 25 Dentro de las etiquetas del footer, generamos otro contenedor que utilizaremos para centrar horizontalmente los elementos internos, y le asignamos una clase llamada `[cont_footer]`.



ARMADO DEL FOOTER

PASO A PASO



26 | DIV TWITTER



```

social_box {
border-right: 1px solid #047da6;
float: left;
height: 288px;
width: 150px;

.social_box img { margin-left: 36px; margin-top: 11px; }

.social_box h2 {
color: #2f2d40;
font-size: 32px;
font-weight: lighter;
margin-top: 7px;
text-align: center;
}

.social_box p {
color: #2f2d40;
font-size: 10px;
margin-top: 19px;
min-height: 87px;
padding: 0 10px;
text-align: center;
}

.social_box .more { margin: 0 auto; width: 120px; }

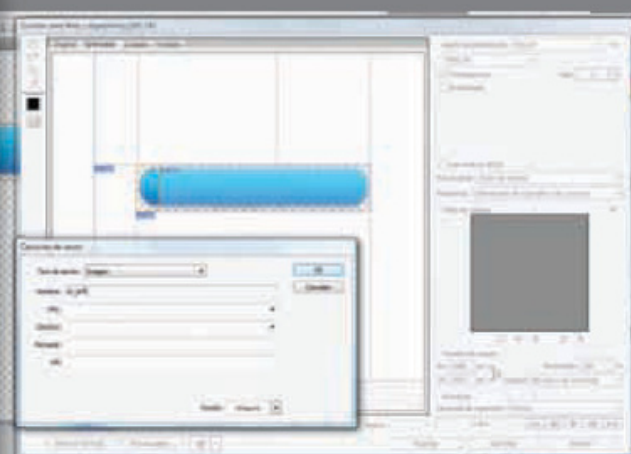
```

ESTILOS REDES SOCIALES

27



28 | OTRAS REDES SOCIALES



BOTÓN REUTILIZABLE

29

PASO | 26 Para definir la estructura HTML de cada una de las cajas, vamos a generar un div que contendrá una imagen, un título, un párrafo de texto y un botón, que crearemos de forma especial para que pueda utilizarse en otros lugares del sitio. En el caso de Twitter, el div es el siguiente:

```

<div class="social_box"> </img>
<h2>Twitter</h2> <p>"Por favor sentite
libre de comentar de manera fugaz mis
trabajos..."</p> <div class="more"><a href="#"
class="bt"><span>Visitar</span></a></div>
</div>

```

PASO | 27 Para dar estilo a las áreas de las redes sociales, incorporamos en nuestro archivo de estilos las reglas CSS que vemos en la imagen.

PASO | 28 Para continuar, creamos los divs correspondientes a las otras redes sociales, agregando las imágenes y el texto adecuados en cada caso.

PASO | 29 Para crear los botones de las redes sociales, debemos definir los sectores en Photoshop. Al guardar las imágenes, la esquina izquierda se llamará **bt_left.png**, y la derecha, **bt_right.png**.



30 | ARMAR BOTÓN



31 | ESTILOS BOTONES

```
background: url("../img/bt_left.png") no-repeat scroll 0 0 transparent;
color: #ffffff;
float: left;
font-size: 11px;
font-weight: bold;
height: 31px;
padding-left: 18px;
text-decoration: none;

a.bt span {
  background: url("../img/bt_right.png") no-repeat scroll right top transparent;
  display: block;
  height: 31px;
  padding-left: 22px;
  padding-right: 46px;
  padding-top: 7px;
}

a.bthover { color: #025685; }
```

PASO | 30

El maquetado de los botones es sencillo, ya que éstos son enlaces que contienen el título dentro de una etiqueta `<span>`. En la etiqueta `<a>` definimos como background las imágenes correspondientes a los bordes redondeados del diseño. En `<span>` definimos como background el resto del diseño, pero en forma más extensa, de modo que, si colocamos un texto más largo, el fondo se siga viendo. El código del botón [Visitar] es: `<div class="more"><a href="#" class="bt"><span>Visitar</span></a></div>`.

PASO | 31

Para los botones, escribimos las reglas CSS que observamos en la imagen.



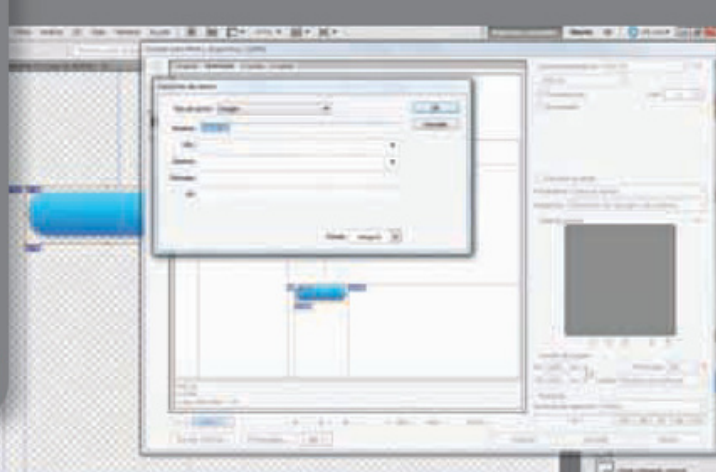
FORMULARIO DE

PASO A PASO

Veremos cómo crear el formulario de pedido de presupuesto que aparece en el footer de nuestro diseño.



32 | ICONO FORMULARIO



BOTÓN FORMULARIO **33**

Aunque le daremos funcionalidad más adelante mediante el lenguaje PHP, en esta instancia crearemos los campos y botones del formulario, dejando todo listo para cuando llegue el momento de hacerlo funcionar. ▶

PASO | 32 Para comenzar a crear el formulario destinado a pedir un presupuesto, en Photoshop marcamos el sector perteneciente al icono de este elemento y lo guardamos como **contact_icon** en formato PNG transparente.

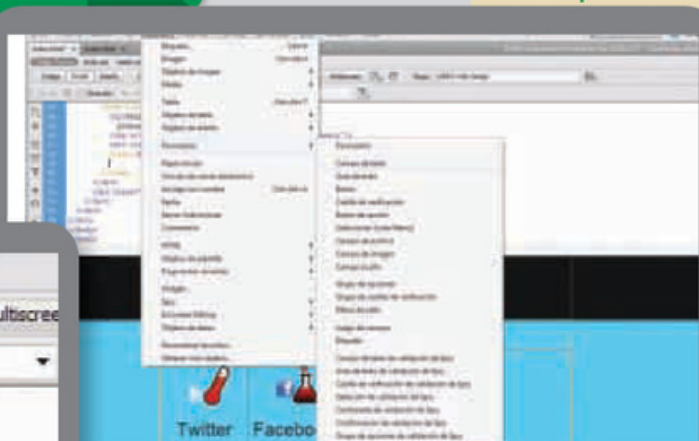
PASO | 33 En el mismo programa, generamos el sector para el botón **Submit** del formulario y lo guardamos con el nombre **submit.png**.

PASO | 34 Creamos un div con la clase **[contact_box]**, vamos al menú **[Insertar]** y, dentro de **[Formulario]**, elegimos **[Formulario]**. Al código del formulario le agregamos una clase **[contact_form]**, una etiqueta **<h2>** para el título y la imagen correspondiente. Al final, insertamos un div con la clase **[clearfix]**. Ésta nos permite generar una división invisible, con el objetivo de alinear todo aquello que se coloque debajo de



PRESUPUESTO

35 | ETIQUETA



CAMPOS Y BOTÓN | 36

```
css reset.css:
Diseño Código en vivo Vista en vivo Inspeccionar Multiscreen
Acción: file:///D:/Mis Documentos/Web/Red Users/Clase 12 - Copia/index.html
<form class="contact_form" method="post" action="#">
  <h2>Pedi un<br/>
  presupuesto</h2>
  
  <div class="clearfix"></div>
</form>
```

34 | FORMULARIO

```
.contact_box {
  float: left;
  height: 236px;
  margin-left: 12px;
  width: 233px;

  .contact_box img { float: left; margin-left: 8px; }

  .contact_box h2 {
    color: #047da5;
    float: left;
    font-size: 23px;
    font-weight: lighter;
    margin-top: 11px;
  }

  .contact_box label { color: #2f2d40; font-size: 13px; }
}
```

ESTILOS IMG, H2 Y LABEL | 37

forma más clara, rápida y fácil. Es una clase muy utilizada y está definida en el archivo **reset.css** de la siguiente manera: `[.clearfix(clear:both;)]`.

PASO | 35 Vamos otra vez al menú **[Insertar]** y, dentro de **[Formulario]**, hacemos clic en **[Etiqueta]**. Luego, colocamos el nombre que deseamos en la etiqueta `<label>`, en este caso, **[Nombre]**.

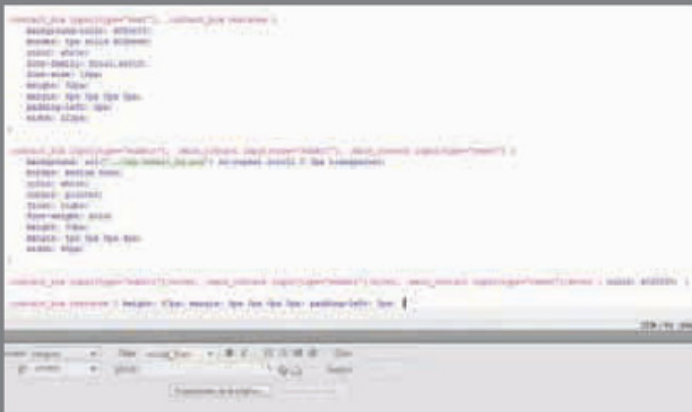
PASO | 36 Dentro del menú **[Insertar]**, elegimos **[Formulario]** y, después, **[Campo de texto]**. A continuación, volvemos a agregar una etiqueta (que llamaremos **[Comentario]**). Desde el mismo menú, hacemos clic en **[Área de texto]** y, finalmente, en la opción **[Botón]**.

PASO | 37 Las reglas CSS que aplicaremos al formulario deben ser las que vemos

en la imagen. Generamos la clase **[contact_box]**, le pedimos que flote a la izquierda, y le otorgamos una altura, un margen izquierdo y un ancho determinado. Luego, a la etiqueta `<img>` para la imagen que colocamos dentro del div le pedimos que flote a la izquierda y le damos un margen izquierdo determinado. A la etiqueta `<h2>` le otorgamos un color, le indicamos que flote a la izquierda, le damos una variable de tamaño y otra de cuerpo, y le asignamos un margen superior determinado. Finalmente, a `<label>` le damos un color y le otorgamos una variable de tamaño.

FORMULARIO DE PRESUPUESTO

PASO A PASO



38 | ESTILOS INPUT Y TEXTAREA

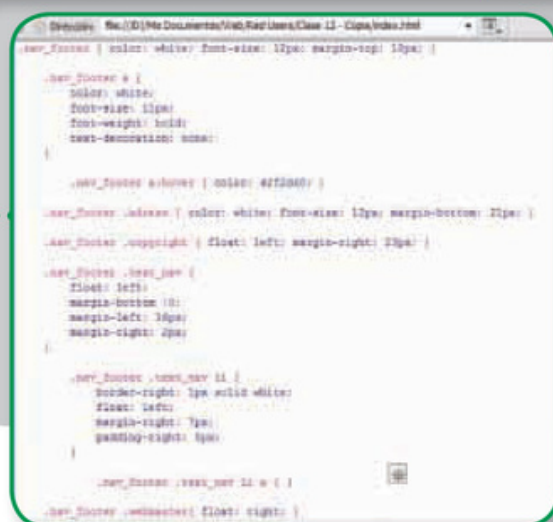


INFORMACIÓN FOOTER

39



40 | ESTILOS FOOTER



PASO | 38

Con el resto de las reglas CSS que vemos en la imagen, a las etiquetas `<input>` de tipo `[text]` y `<textarea>` les damos un color de fondo, las características de borde y tipografía, el alto, los márgenes, el padding izquierdo y el ancho. Luego, a la etiqueta `<input>` de tipos `[submit]` y `[reset]` le aplicamos una imagen como fondo, un borde, un color, un tipo de cursor, un flotado a la izquierda, una variable de cuerpo, una altura, un margen y un ancho.

A los estados **hover** de estas etiquetas `<input>` les otorgamos un color y, finalmente, a la etiqueta `<textarea>` le redefinimos la altura, los márgenes y el padding izquierdo.

PASO | 39

Incorporamos al pie de la página la información textual sobre los medios de contacto, copyright y accesos a las diferentes secciones. Para realizar esta acción, agregamos el código en el footer.

PASO | 40

Luego, escribimos las reglas CSS necesarias para decorar esta información en el footer, tal como vemos en la imagen.



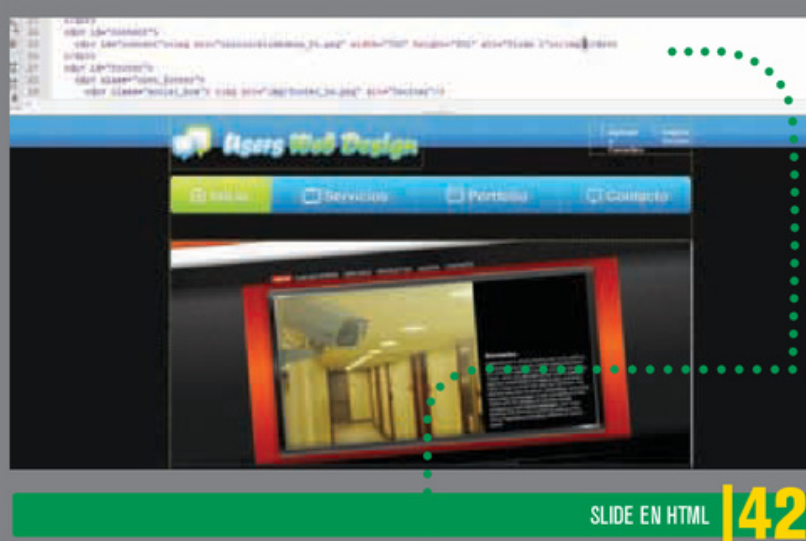
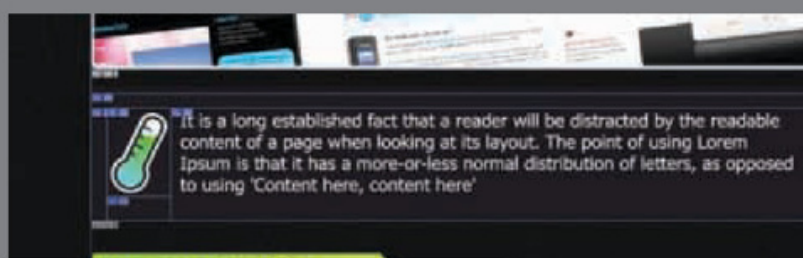
CREAR LA PÁGINA DE INICIO

PASO A PASO

Ahora que creamos los elementos básicos del layout, vamos a agregar la información de la página principal.



41 IMAGEN SLIDE



SLIDE EN HTML 42

ICONO DESTACADO 43

Hasta aquí hemos creado la parte de la estructura que se mantiene fija a través de todas las secciones del sitio. A continuación, comenzaremos a incorporar los elementos particulares de la página de inicio.

```
</div>
<div id="content">
  </img>
</div>
<div id="footer">
  <div class="cont_footer">
    <div class="social_bar">
      
    </div>
  </div>
</div>
```

PASO | 41 Para crear el banner de contenido de la página principal, primero usaremos una imagen simple, porque luego de terminar el maquetado, deberemos modificar el archivo de esta página para reemplazar la imagen por una librería JavaScript que nos permita visualizar un **slideshow** con un efecto de transición entre las fotos. Para crear la imagen actual, marcamos su sector en Photoshop y la guardamos como **slide.jpg**.

PASO | 42 Dentro del div **[content]**, colocamos la imagen en forma provisoria de esta manera:

```
<div id="content">
  </img>
</div>
```

PASO | 43 Dentro de Photoshop, marcamos el sector perteneciente al icono del recuadro destacado y lo guardamos con el nombre **intro_icon.png**.

CREAR LA PÁGINA DE INICIO

PASO A PASO



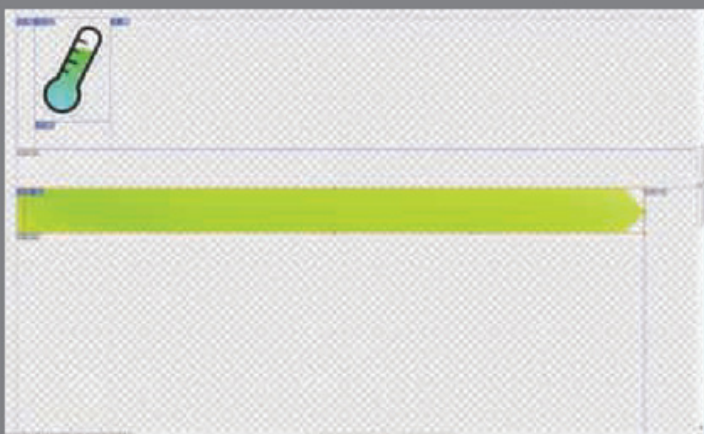
44 | RECUADRO DESTACADO

```
.intro {
  background-color: #f1f1f1;
  height: 167px;
  margin-bottom: 30px;
  margin-top: 34px;
  width: 700px;

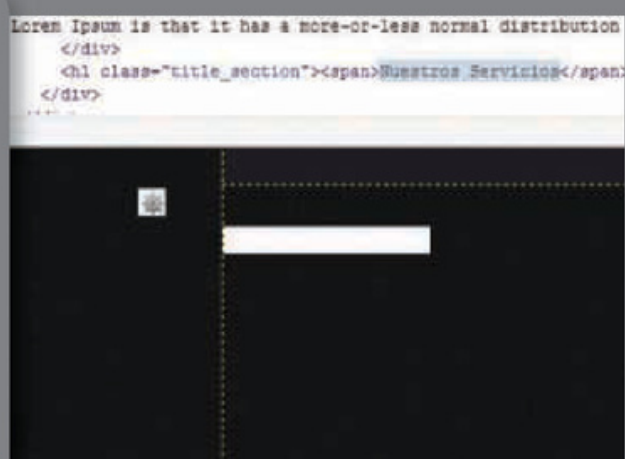
  .intro img { float: left; margin-left: 37px; margin-top: 41px; }

  .intro p {
    color: white;
    float: left;
    font-size: 19px;
    line-height: 25px;
    margin-left: 27px;
    margin-top: 32px;
    width: 560px;
  }
}
```

ESTILO DESTACADO **45**



46 | FONDO TÍTULO



TÍTULOS SERVICIOS **47**

PASO | 44 Para realizar el cuadro destacado con el icono y el texto, debemos crear una caja contenedora y sus diferentes elementos dentro de `[content]`. Para hacerlo, creamos un `div` con la clase `[intro]`, insertamos el icono y, luego, el texto, que en este caso es una simulación para realizar la prueba.

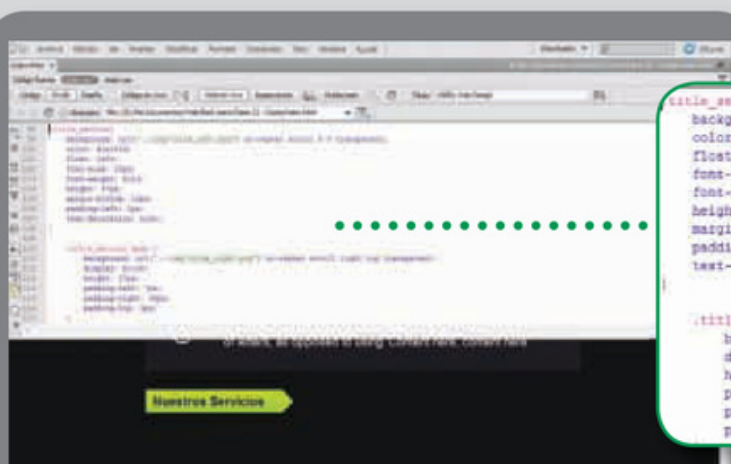
PASO | 45 Por medio de CSS, le damos estilo al destacado con las reglas que vemos en la imagen.

PASO | 46 En el caso de los títulos para identificar los servicios, optamos por realizar una

sola clase que nos permita ajustar el fondo según el contenido de texto que incluyamos. Para eso, extendemos el diseño de fondo en Photoshop y, tal como hicimos con los botones de las redes sociales, marcamos dos sectores. Uno es para definir los bordes laterales izquierdos del título, que guardamos como `title_left.png`; el otro es para el resto del fondo extendido, que almacenamos como `title_right.png`.

PASO | 47 El código HTML que utilizaremos para todos los títulos será el siguiente:

```
<h1 class="title_section"><span>Nuestros
Servicios</span></h1>
```



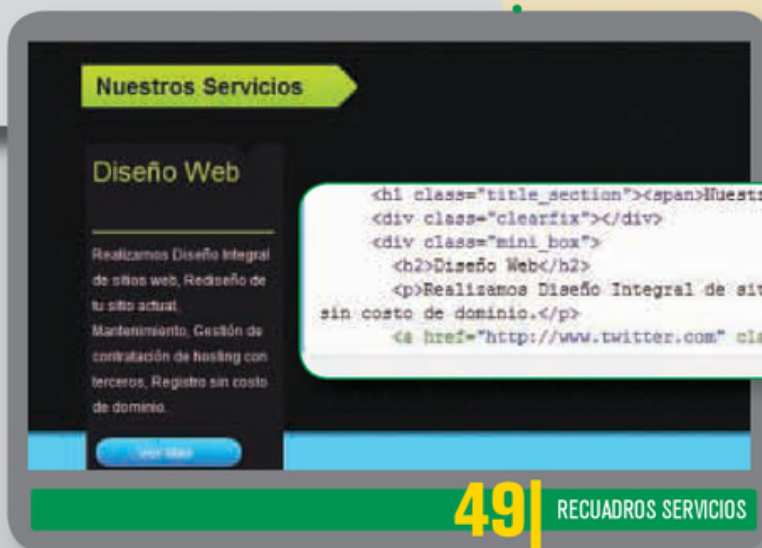
48 | ESTILOS SERVICIOS

```

title_section {
  background: url("../img/title_left.png") no-repeat scroll 0 0 transparent;
  color: #1a181d;
  float: left;
  font-size: 20px;
  font-weight: bold;
  height: 37px;
  margin-bottom: 12px;
  padding-left: 5px;
  text-decoration: none;

  .title_section span {
    background: url("../img/title_right.png") no-repeat scroll right top transparent;
    display: block;
    height: 37px;
    padding-left: 7px;
    padding-right: 46px;
    padding-top: 9px;
  }

```



49 | RECUADROS SERVICIOS

```

<h1 class="title_section"><span>Nuestros Servicios</span></h1>
<div class="clearfix"></div>
<div class="mini_box">
  <h2>Diseño Web</h2>
  <p>Realizamos Diseño Integral de sitios web, Rediseño de tu sitio actual, Mantenimiento, Gestión de
  sin costo de dominio.</p>
  <a href="http://www.twitter.com" class="bt"><span>Ver Más</span></a> </div>

```

PASO | 48 Luego, incluimos en nuestro archivo CSS las reglas que vemos en la imagen.

PASO | 49 Para crear los recuadros de servicios, hacemos uno solo y lo duplicamos las veces que sean necesarias, ya que todos son iguales, a excepción de sus contenidos. El código HTML es el que disponemos a continuación:

```

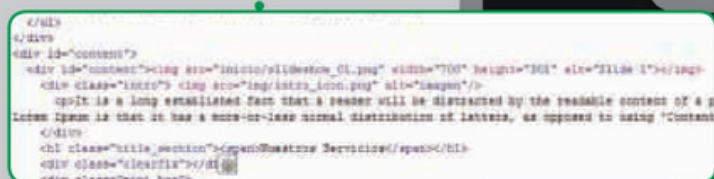
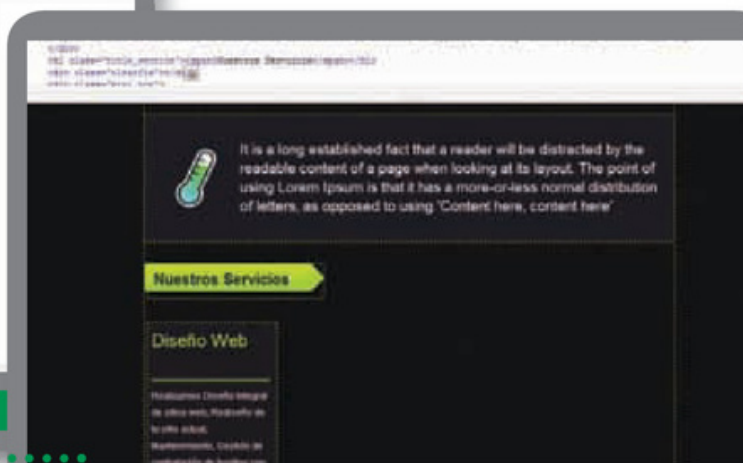
<div class="mini_box">
  <h2>Diseño Web</h2>
  <p>Realizamos Diseño Integral de sitios web,
  Rediseño de tu sitio actual, Mantenimiento,
  Gestión de contratación de hosting con terceros,
  Registro sin costo de dominio.</p>
  <a href="#" class="bt"><span>Ver Más
  </span></a>
</div>

```



CREAR LA PÁGINA DE INICIO

PASO A PASO



PASO | 50 Creamos las reglas CSS para la clase `[mini_box]` y sus elementos internos, tal como vemos en la imagen.

PASO | 51

Ahora, si revisamos el resultado del maquetado, notaremos un inconveniente con esta caja de servicios. Esto se debe a que tanto el título de la sección como las cajas están flotadas (float) a la izquierda, y la caja se posiciona a la derecha del título. Para solucionar este problema, simplemente colocamos un div con clase **[clearfix]** entre el título y las cajas de servicios, de modo de generar una división horizontal entre estos elementos. La línea es `[<div class="clearfix"></div>]`.

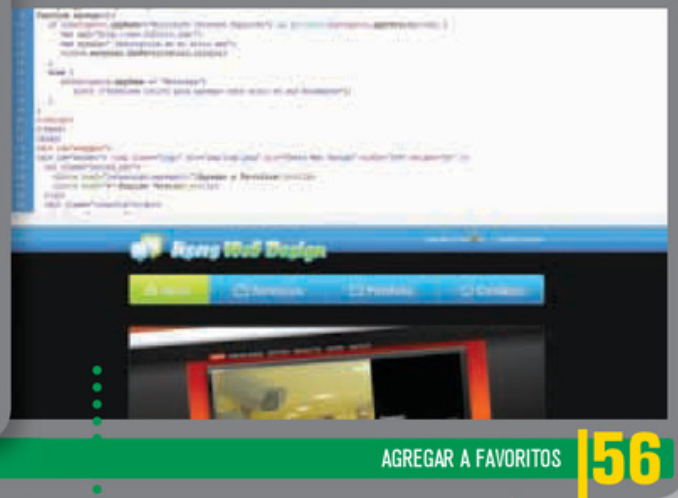
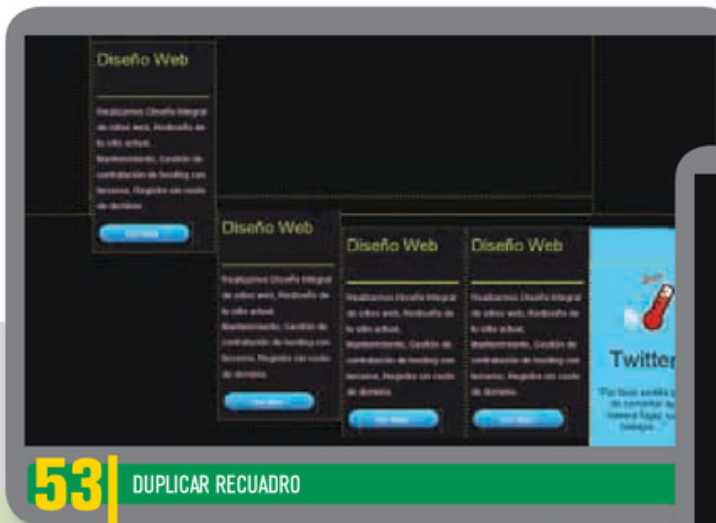
PASO | 52 Si volvemos a comprobar el maquetado, veremos que se ha solucionado el inconveniente anterior.

PASO | 53 Ahora, al duplicar los diferentes boxes, veremos cómo quedan los cuatro recuadros, uno junto al otro.

PASO | 54 Para terminar, escribimos los textos correspondientes a cada uno de los servicios ofrecidos y sus títulos.

PASO | 55 Al duplicar los diferentes boxes, vemos cómo quedan los cuatro recuadros, uno junto al otro. También colocamos la línea `[<div class="clearfix"></div>]` justo antes de cerrar `[content]`, para evitar que el footer se pose sobre el contenido.

PASO | 56 Finalmente, agregamos la función JavaScript que hará que el botón **[Agregar a favoritos]** cumpla su propósito. Para esto,



dentro de `<head>` escribimos el código que aparece en la imagen, desde la etiqueta `<script type="text/javascript">` hasta su cierre `</script>`. Luego, modificamos la etiqueta `<a>` del botón **[Agregar a favoritos]**, que ya habíamos creado al comienzo de este capítulo, para que diga `<a href="javascript:agregar()">Agregar a Favoritos</a>`. De esta manera, se ejecutará el comportamiento de la función JavaScript al hacer clic en el botón.

```
<link href="css/style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript">
function agregar(){
  if (navigator.appName=="Microsoft Internet Explorer") && (navigator.appVersion>=4) {
    var url="http://www.tufuturo.com/";
    var titulo=" Descripción de mi sitio web";
    window.external.AddFavorite(url,titulo);
  }
  else {
    if(navigator.appName == "Netscape")
      alert ("Pensione Ctrl+D para agregar este sitio en sus Bookmarks");
  }
}
```

RESUMEN

Luego de los pasos realizados a lo largo de todo el capítulo, contamos con una página principal completa, a la que sólo le falta la programación para hacer funcionar el formulario del footer y el slide del contenido. Con esta primera parte del sitio terminada, podemos comenzar a desarrollar el resto de las secciones, que crearemos basándonos en este mismo archivo, para ahorrar tiempo de trabajo.

ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 ¿Con qué herramienta podemos marcar el área que ocupa el **header** en nuestro sitio?

02 ¿Cómo cargamos la hoja de estilo dentro del documento?

03 ¿Qué nos permiten los elementos del **footer**?

04 Nombre algunos elementos particulares que puede tener la **página de inicio**.

05 ¿Qué debemos tener en cuenta principalmente para marcar el sector del **footer**?

06 ¿Qué se debe mostrar en el **portfolio**?

07 ¿Para qué sirve la sección de **contacto**?

08 ¿Qué información podemos incluir en la sección denominada **servicios**?

09 ¿Cómo nos conviene categorizar el contenido de nuestra web?



PRÁCTICAS

01 Tome las medidas para el **layout**.

02 Defina el **layout** con CSS.

03 Cree el **menú de navegación**.

04 Arme el **footer** de su sitio y establezca el **header**.

05 Cree la **página de inicio**.

06 Genere un **formulario de presupuesto**.



Proyecto real II

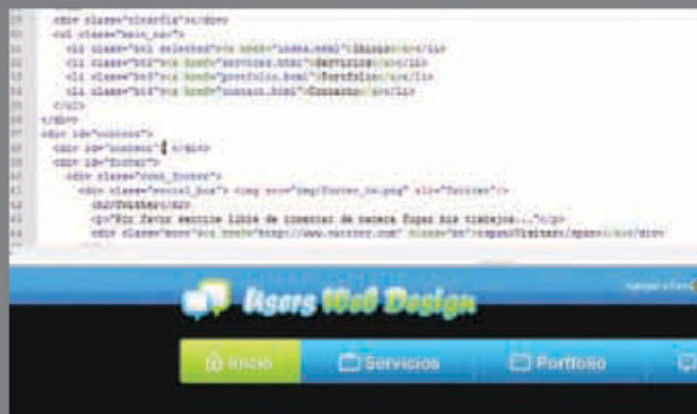


- ▶ SITIO HTML
- ▶ LAS PÁGINAS DE SERVICIO
- ▶ PÁGINA PORTFOLIO
- ▶ PÁGINA DE DETALLES
- ▶ PÁGINA DE CONTACTO



LAS PÁGINAS DE PASO A PASO

Para continuar con la diagramación de nuestro sitio, comenzaremos a crear el resto de las páginas.



01 | DOCUMENTO

```
<div class="clearfix"></div>
<ul class="main_nav">
<li class="bt1"><a href="index.html">Inicio</a></li>
<li class="bt2 selected"><a href="servicios.html">Servicios</a></li>
<li class="bt3"><a href="portfolio.html">Portfolio</a></li>
<li class="bt4"><a href="contact.html">Contacto</a></li>
</ul>
</div>
<div id="content">
<div id="content"> </div>
<div id="footer">
<div class="cont_footer">
<div class="social_box"> 
<h2>Twitter</h2>
<p>Por favor sentite libre de comentar de manera fugaz mis trabajos</p>
<div class="more"><a href="http://www.twitter.com" class="bt"><span>
```



SECCIÓN ACTUAL | 02

Luego de las definiciones iniciales y de maquetar la página principal, pondremos manos a la obra para generar el resto de las secciones del sitio. Comenzaremos con la página de servicios, una parte fundamental de nuestro sitio, ya que muestra lo que tenemos para ofrecerle al potencial cliente que decide visitarnos. ◀



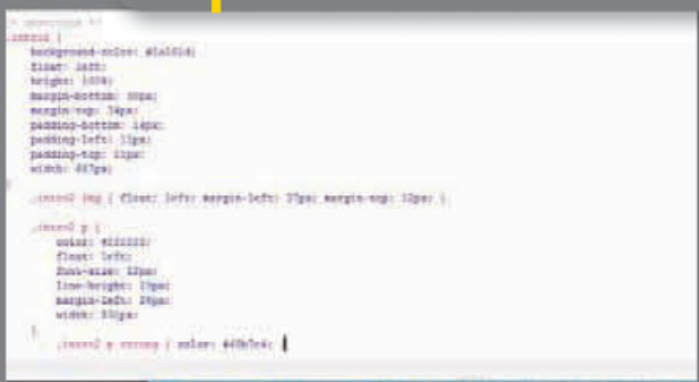
PASO | 01 Dentro de Dreamweaver, creamos un nuevo archivo HTML y lo guardamos como **servicios.html**. En él copiamos todo lo que hicimos en **index.html** pero eliminamos todo el contenido del div **[content]**, a excepción de la línea **[<div class="clearfix"></div>]**, que aparece antes del footer, porque es la que evitará que se superpongan los elementos.

PASO | 02 Ahora vamos a señalar la sección actual en el menú de navegación principal, quitando la clase **[selected]** del primer **** e incluyéndola en el segundo, para que nos quede como vemos en la imagen.

PASO | 03 Para crear la frase destacada, que es el primer elemento que encontramos en la sección **Servicios**, marcamos su sector en Photoshop y lo guardamos como **destacado_serv.png**.

SERVICIO

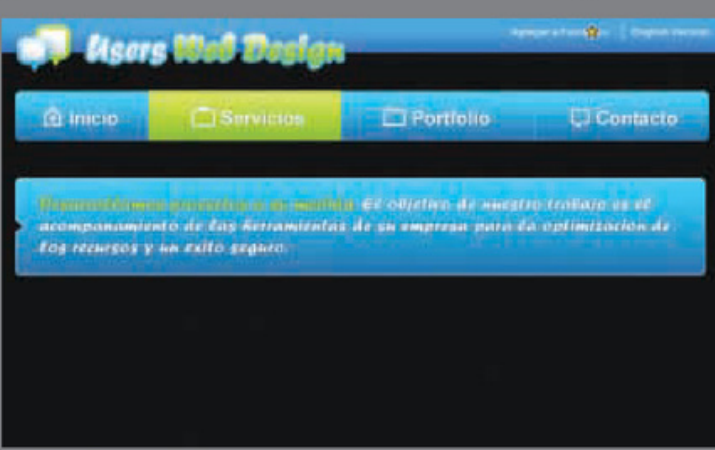
04 | ICONO DESTACADO



ESTILO DESTACADO | 05



03 | FRASE DESTACADA



LISTADO DE SERVICIOS | 06



A continuación, incluimos la imagen dentro del div `[content]` mediante la línea `</img>`.

PASO | 04 Para crear el cuadro destacado, marcamos en Photoshop el sector del icono del recuadro y lo guardamos como `servicios_icon.png`. Incorporamos el título de la sección y, luego, el icono y el texto del recuadro con una estructura HTML similar a la usada en el cuadro destacado de la página de inicio.

PASO | 05 Luego, aplicamos las reglas CSS que podemos ver en la imagen.

PASO | 06 Para realizar el listado de los servicios en esta sección, utilizaremos el elemento `<ul>` y, dentro de cada `<li>`, colocaremos una caja contenedora que identificará a cada uno de los servicios, como vemos en el

código de la imagen. A cada ítem le hemos asignado un ID con un valor `service_x`, que utilizaremos como ancla para direccionar a cada servicio por medio de enlaces como `<a href="servicios.html#servicio_3">Ir a Servicio 3</a>`.

LAS PÁGINAS DE SERVICIO

PASO A PASO



07 MEDIDAS SERVICIOS



08 ESTILOS SERVICIOS



09 SECCIÓN PORTFOLIO

PASO | 07 En Photoshop, marcamos los sectores correspondientes al recuadro contenedor de cada servicio y a la imagen que contiene, para poder tomar las medidas de ambos.

PASO | 08 Luego, aplicamos las reglas CSS que vemos en la imagen para estos elementos.

PASO | 09 Ahora, ya tenemos terminada esta sección. Si presionamos el botón **[Vista en vivo]**, podremos ver el resultado obtenido.



PÁGINA

PASO A PASO

PORTFOLIO

Ahora crearemos la sección que presenta nuestra experiencia como diseñadores.

10 | LISTA SUBMENÚ

```
<div id="content">
  <ul class="sub_nav">
    <li class="selected"><a href="portfolio.html">Web</a></li>
    <li><a href="portfolio2.html">Print</a></li>
    <li><a href="portfolio3.html">Ilustración</a></li>
    <li><a href="portfolio4.html">Multimedia</a></li>
  </ul>
</div>
```

Para optimizar nuestro trabajo y ahorrar tiempo, en este nuevo comienzo volveremos a utilizar lo que ya diagramamos hasta ahora. Para eso, creamos un nuevo documento HTML, que guardamos como **portfolio.html**, y limpiamos de contenido el div **[content]**. Luego, aplicamos la clase **[selected]** al ítem de la lista del menú que corresponde a esta página, es decir, **[bt3]**. Una particularidad de esta sección es que encontraremos cuatro categorías en ella (**web**, **print**, **ilustración** y **multimedia**). Cada una deberá estar contenida en una página HTML diferente, a la que accederemos por medio del submenú de navegación. Por defecto, la categoría que aparece al ingresar en la sección **Portfolio** será la correspondiente a **web**.

PASO | 10

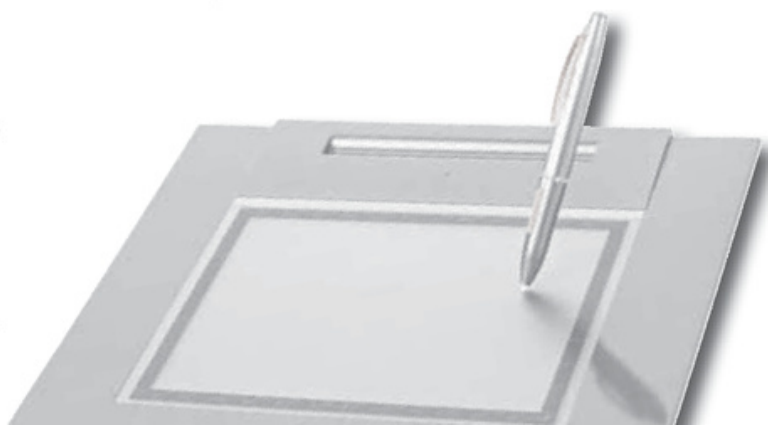
Para crear el submenú de navegación, incluimos un submenú dentro de **[content]**, utilizando la misma estructura del menú principal.

LISTA SUBMENÚ

11

PASO | 11

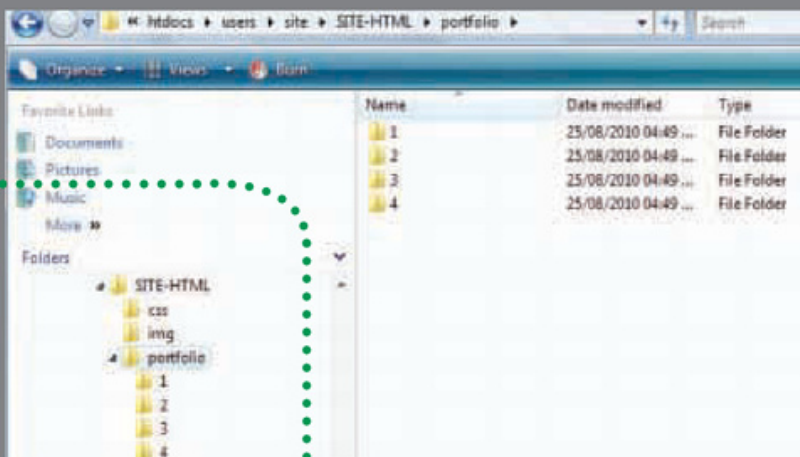
En Photoshop, marcamos un sector con el alto del menú y con un ancho pequeño, de modo de poder utilizar la imagen para hacer una repetición en el eje X a lo largo del menú. En este caso, al no usar el método CSS Sprite, no será necesario definir otra imagen para los diferentes estados de los botones.



PÁGINA PORTFOLIO

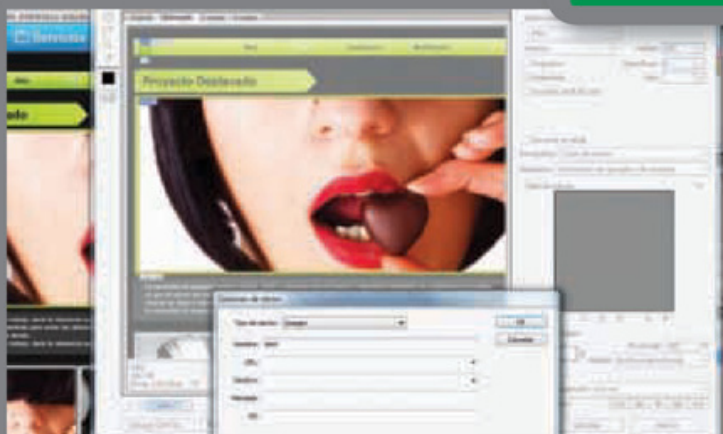
PASO A PASO

12 | ESTILO SUBMENÚ



CARPETAS DE PROYECTO

13



14 | IMAGEN PROYECTO DESTACADO

```

.sub_nav {
  background: url("../img/sub_nav_bg.jpg") repeat-x scroll 0 0 transparent;
  height: 29px;
  padding-left: 15px;
  margin-bottom: 40px;
}

.sub_nav li {
  border-left: 1px solid gray;
  float: left;
  margin-right: 20px;
  margin-top: 6px;
  padding-left: 18px;
}

.sub_nav li:first-child {
  border-left: none;
}

.sub_nav li a {
  color: #222222;
  font-size: 14px;
  font-weight: bold;
  text-decoration: none;
}

.sub_nav li a:hover {
  color: #ffffff;
}

.sub_nav li.selected a {
  color: #ffffff;
}

```

PASO | 12

A continuación, escribimos las reglas CSS necesarias para este submenú, tal como vemos en el siguiente código:

```

.sub_nav {
  background: url("../img/sub_nav_bg.jpg")
  repeat-x scroll 0 0 transparent; height: 29px;
  padding-left: 15px;
  margin-bottom: 40px; }
.sub_nav li { border-left: 1px solid gray;
  float: left; margin-right: 20px;
  margin-top: 6px; padding-left: 18px; }
.sub_nav li:first-child { border-left: none; }
.sub_nav li a {
  color: #222222; font-size: 14px;
  font-weight: bold; text-decoration: none; }
.sub_nav li a:hover { color: #ffffff; }
.sub_nav li.selected a { color: #ffffff; }

```

PASO | 13

Antes de comenzar a recortar las imágenes para esta sección, con el fin de almacenarlas creamos una carpeta llamada **[portfolio]**, que contendrá una subcarpeta para cada una de las categorías. En este caso, la estructura de carpetas es la que vemos en la imagen.

PASO | 14

Para el sector del proyecto destacado, lo primero que haremos será marcar el área correspondiente a la imagen de éste sin los bordes, porque éstos serán generados mediante CSS. Tomamos sus medidas y guardamos la imagen en la carpeta **[portfolio/1]** como **dest.jpg**.

PASO | 15

Ahora, crearemos la estructura HTML para este destacado, como observamos a continuación:

```

50 <div class="proy_dest"> 
51 <div class="box">
52 <p>La necesidad de asegurar nuestro trabajo...</p>
53 <a href="#" class="bt"><span>Ver Más</span></a>
54 </div>
55 </div>

```

15 | PROYECTO DESTACADO



ESTILO PROYECTO DESTACADO II

17



16 | ESTILO PROYECTO DESTACADO II

```

* portfolio - proyecto destacado *
.proy_dest img { border: 6px solid #b6d948; margin-bottom: 13px; }

.proy_dest .box {
  background-color: #333333;
  float: left;
  height: 100%;
  padding-bottom: 12px;
  width: 700px;

  .proy_dest .box p {
    color: #ffffff;
    float: left;
    font-size: 13px;
    line-height: 17px;
    margin-left: 14px;
    margin-right: 18px;
    padding-top: 8px;
    width: 514px;
  }

  .proy_dest .box a { margin-top: 38px; }

```

```

<div class="proy_dest"> 
<div class="box"> <p>La necesidad de
asegurar nuestro trabajo...</p>
  <a href="#" class="bt"><span>Ver
  Más</span></a>
</div>
</div>

```

```

.proy_dest { }
.proy_dest img { border: 6px solid #b6d948;
margin-bottom: 13px; }
.proy_dest .box {
  background-color: #333333; float: left;
height: 100%; padding-bottom: 12px;
width: 700px; }
.proy_dest .box p {
  color: #FFFFFF; float: left; font-size: 13px;
line-height: 17px; margin-left: 14px; margin-
right: 18px; padding-top: 8px; width: 514px;
}
.proy_dest .box a { margin-top: 38px; }

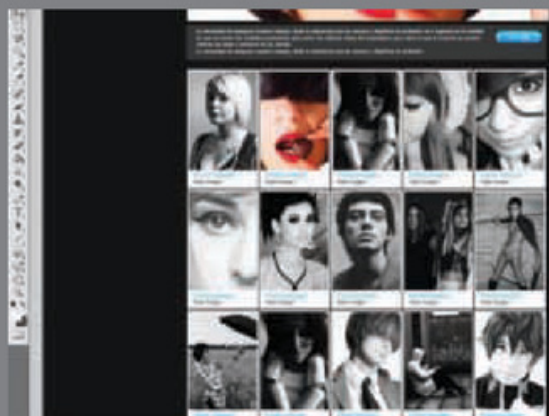
```

PASO 16 A continuación, escribimos las reglas CSS para el proyecto destacado, tal como podemos observar en el siguiente código:

PASO 17 Luego de lo que realizamos, obtenemos el resultado que se observa en la imagen.

PÁGINA PORTFOLIO

PASO A PASO



181 FICHAS DE PROYECTO

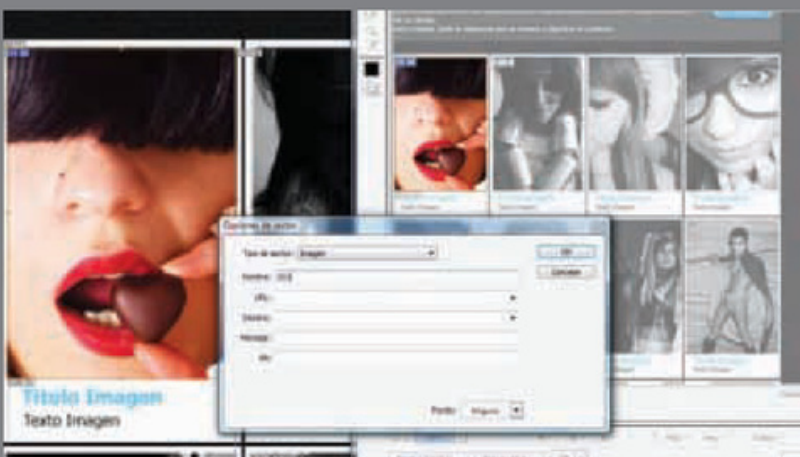
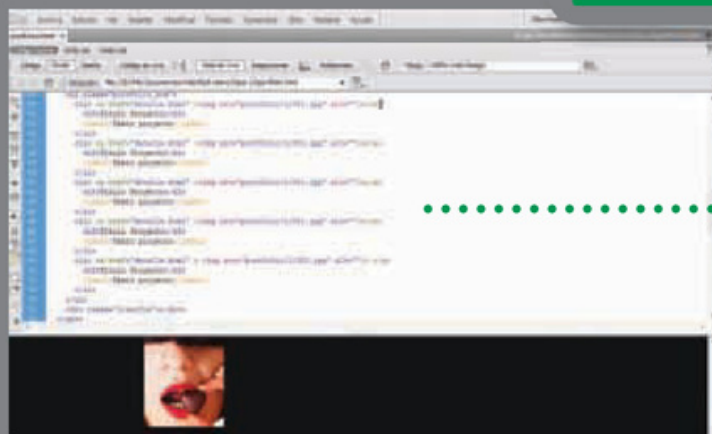


IMAGEN DE FICHA 19



201 HTML FICHAS

```
cul class="portfolio_item">
<li> <a href="detalle.html" ></a>
    <h3>Titulo Proyecto</h3>
    <div>Texto proyecto</div>
</li>
<li> <a href="detalle.html" ></a>
    <h3>Titulo Proyecto</h3>
    <div>Texto proyecto</div>
</li>
<li> <a href="detalle.html" ></a>
    <h3>Titulo Proyecto</h3>
    <div>Texto proyecto</div>
</li>
<li> <a href="detalle.html" ></a>
    <h3>Titulo Proyecto</h3>
    <div>Texto proyecto</div>
</li>
<li> <a href="detalle.html" ></a>
    <h3>Titulo Proyecto</h3>
    <div>Texto proyecto</div>
</li>
</ul>
<div class="clearfix"></div>
</div>
```

PASO | 18 A la hora de crear las fichas de proyectos, podemos observar que en el diseño de cada una de las categorías o subsecciones del portfolio, los trabajos se muestran en una estructura reticular con fichas para cada uno de ellos. Estas fichas contienen una imagen, un título y una línea de texto descriptivo, y están ubicadas en cuatro columnas y las filas necesarias según la cantidad de fichas que se van a mostrar.

PASO | 19 Comenzamos tomando las medidas de una ficha y de la imagen que contendrá, y guardamos esa imagen como **001.jpg** dentro de la carpeta correspondiente a esta sección del portfolio (utilizaremos números correlativos para nombrar a cada imagen).

PASO | 20

Definimos el código HTML de las fichas que usaremos para acceder a la información detallada de cada proyecto presentado en el portfolio. Como se trata de un listado de proyectos, utilizaremos una estructura de listas. Cada ítem **** del listado está compuesto por un enlace que incluye una imagen insertada con HTML por medio de la etiqueta ****, un título y una línea de texto utilizando la etiqueta **<label>**.

PASO | 21

A continuación, definimos el tamaño de cada ficha `<li>` y sus elementos internos, al mismo tiempo que convertimos en bloque (`[display: block;]`) cada elemento interno para poder asignarle una ubicación correcta. Junto a las demás propiedades CSS otorgadas, escribimos las que vemos a continuación:

[illegible]

21 | ESTILOS FICHAS

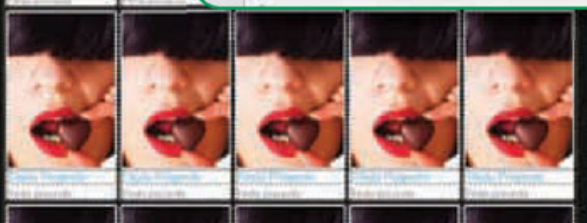


La necesidad de asegurar nuestro trabajo, dale la relevancia que se merece y dignificar la profesión va a lograrse en la medida en que se tomen las medidas preventivas para evitar las clásicas frotas del empresario poco serio al que le importa un conino rebasar las ideas y esfuerzo de los demás. La necesidad de asegurar nuestro trabajo, dale la relevancia que se merece y dignificar la profesión.

SECCIÓN PORTFOLIO | 23

[illegible]

```
112 </li>
113 <li> <a href="detalle.html" >  </a>
114 <h3>Titulo Proyecto</h3>
115 <label>Texto proyecto</label>
116 </li>
```

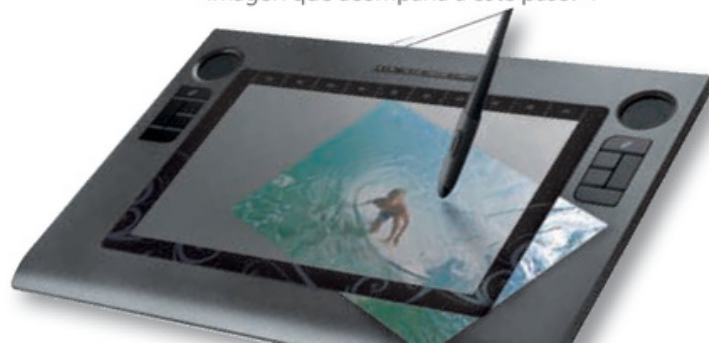


22 | MULTIPLICAR FICHAS

```
.portfolio_box{ margin-left:1px; margin-top:14px; width:702px; }
.portfolio_box li{ background-color:#FFFFFF; float:left; height:232px; margin:0 3px 4px 0; padding:1px; width:135px; }
.portfolio_box li a{ border:2px solid #FFFFFF; display:block; height:190px; width:131px; }
.portfolio_box li h3{ color:#49B7E4; display:block; font-size:13px; margin-left:2px; margin-top:3px; }
.portfolio_box li label{ color:gray; display:block; font-size:12px; margin-left:2px; margin-top:6px; }
.portfolio_box li a:hover{ border:2px solid #49B7E4; }
```

PASO | 22 Para ir finalizando, copiamos y pegamos dos veces más nuestro div **[portfolio_box]**, de manera de tener todos los trabajos de nuestro sitio.

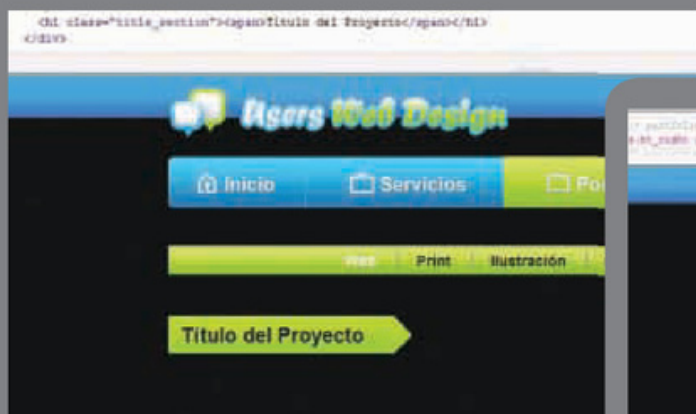
PASO | 23 Cuando terminemos, la sección Portfolio habrá quedado como vemos en la imagen que acompaña a este paso. ◀



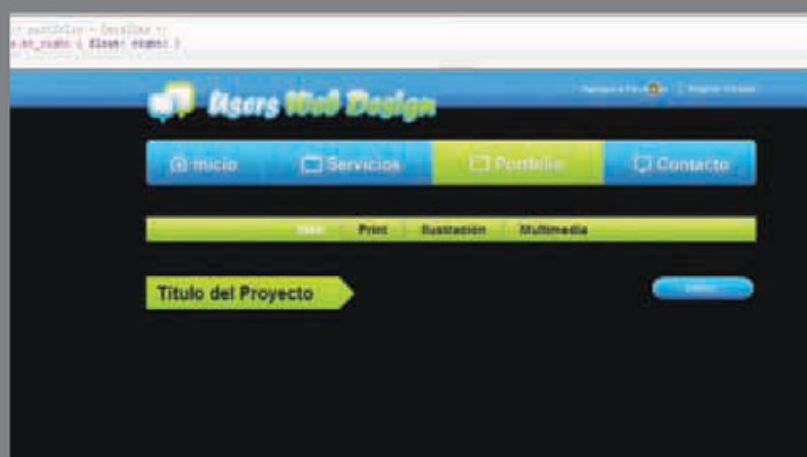
PÁGINA DE DETALLES

PASO A PASO

Cuando tenemos la sección Portfolio terminada, es hora de mostrar los detalles de cada trabajo realizado.



24 | TÍTULO DEL PROYECTO



BOTÓN VOLVER | **25**

Como vimos, cada ítem (ficha) tiene un enlace que debería abrir una nueva página HTML con información y detalles sobre ese proyecto en especial. Para crear una de esas páginas, generamos un nuevo archivo HTML y lo guardamos como **detalle.html**. Tal como hicimos en casos anteriores, allí incluimos el código HTML de la estructura que es común a todas las secciones y, en este caso particular, también ponemos el submenú de la sección Portfolio para poder identificar la categoría a la que pertenece el proyecto que estemos visitando. ◀

PASO | 24 Debajo del submenú, incluimos el título del proyecto utilizando el mismo código HTML y las reglas CSS que en los títulos de otras secciones, es decir, `<h1 class="title_section"><span>Título del Proyecto</span></h1>`.

PASO | 25 Para crear el botón **[Volver]**, utilizaremos el mismo botón que en casos anteriores. Si usáramos la misma clase que en los otros botones, el actual aparecería ubicado a la izquierda, aunque en este caso debe estar a la derecha. Para solucionar este problema, creamos una nueva clase `[bt_right]` cuya propiedad `[float]` tenga valor `[right]`, de la siguiente manera: `[a.bt_right{ float: right; }]`. Luego, concatenamos esta clase a la que ya tiene aplicada el botón, de modo que la nueva propiedad `[float]` pise a la anterior y la reemplace: `<a href="#" class="bt_right bt"><span>Volver</span></a>`.

```

<div class="detail"> 
<div class="box">
  <h3>Fecha</h3>
  <p>04 de Junio de 2019</p>
  <h3>Cliente</h3>
  <p>Realizamos Diseño Integral de sitios web, Rediseño de tu sitio actual, Mantenimiento, Gestión de p
sin costo de dominio.</p>
  <h3>Descripción</h3>
  <p>Realizamos Diseño Integral de sitios web, Rediseño de tu sitio actual, Mantenimiento, Gestión de p
sin costo de dominio. Realizamos Diseño Integral de sitios web, Rediseño de tu sitio actual, Mantenimiento,
terceros, Registro sin costo de dominio. Realizamos Diseño Integral de sitios web, Rediseño de tu sitio actual
hosting con terceros, Registro sin costo de dominio.</p>
  <h3>Enlace</h3>
  <p><a href="http://www.google.com/www.sss.hic/srd/p">
</div>

```

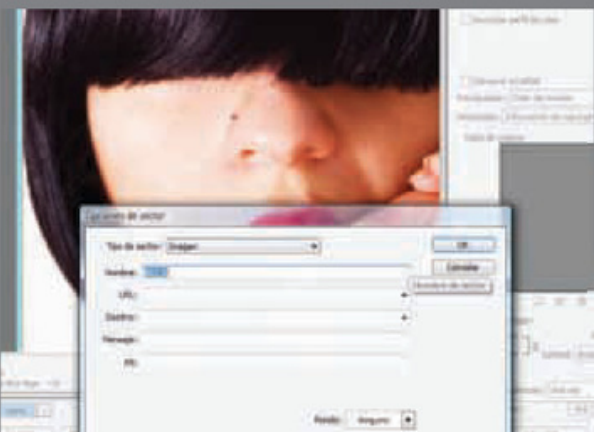
27 | DETALLES

```

<div class="detail"> 
<div class="box">
  <h3>Fecha</h3>
  <p>04 de Junio de 2019</p>
  <h3>Cliente</h3>
  <p>Realizamos Diseño Integral de sitios web, Rediseño de tu sitio actual, Mantenimiento, Gestión de p
sin costo de dominio.</p>
  <h3>Descripción</h3>
  <p>Realizamos Diseño Integral de sitios web, Rediseño de tu sitio actual, Mantenimiento, Gestión de p
sin costo de dominio. Realizamos Diseño Integral de sitios web, Rediseño de tu sitio actual, Mantenimiento,
terceros, Registro sin costo de dominio. Realizamos Diseño Integral de sitios web, Rediseño de tu sitio actual
hosting con terceros, Registro sin costo de dominio.</p>
  <h3>Enlace</h3>
  <p><a href="http://www.google.com/www.sss.hic/srd/p">
</div>

```

ESTILOS | 28



26 | BOTÓN VOLVER II



PÁGINA DE DETALLES | 29

PASO | 26 Para incorporar la imagen principal y los detalles del proyecto, marcamos el sector en Photoshop y lo guardamos como **001_h.jpg** dentro de la carpeta **[portfolio/1]**. Utilizando el sufijo **_h (high)**, tendremos la imagen grande y la pequeña ordenadas en la misma carpeta.

PASO | 27 Creamos un nuevo contenedor HTML al que asignaremos una clase llamada **[detail]** y, dentro de él, incluimos los diferentes elementos HTML que constituirán los detalles del proyecto. Como vemos, incluimos la imagen por medio de HTML y creamos un nuevo contenedor (**[box]**) para colocar los detalles en forma de títulos y párrafos.

PASO | 28 Las reglas necesarias para dar estilos a este apartado son las que vemos en la imagen.

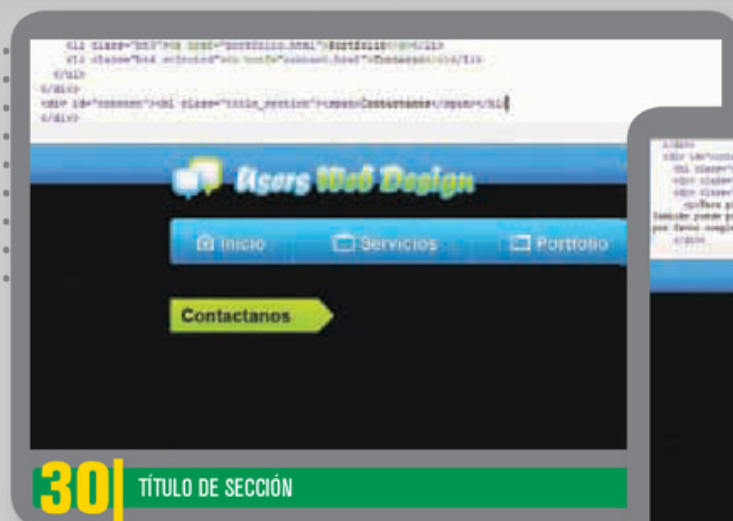
PASO | 29 Luego de los pasos anteriores, tendremos terminada la página de detalles, tal como observamos en la imagen.



PÁGINA DE

PASO A PASO

En los próximos pasos veremos cómo crear la página que permitirá a los visitantes comunicarse con nosotros.



Tal como hicimos en casos anteriores, para generar la página de contacto partiremos de una nueva en la que incluiremos todos los elementos HTML que se repiten en la estructura. Luego, dejaremos el div `[content]` vacío para poder incorporar en él los contenidos específicos de esta página. ◀

PASO | 30 Al igual que en las demás secciones ya realizadas, lo primero que insertaremos es el título de sección dentro de `[Content]`, aplicando la misma estructura y reglas CSS: `<h1 class="title_section"><span>Nuestros Servicios</span></h1>`.

PASO | 31 Para el recuadro debajo del título, vamos a utilizar el mismo recuadro que maquetamos en la sección **Servicios**, eliminando el título que tiene dentro para incorporar el nuevo texto y la imagen, que deberemos tomar desde Photoshop. Colocamos un div con la clase `[clearfix]` antes del recuadro, para evitar que se coloque junto al título.

PASO | 32 Lo primero que haremos para maquetar el formulario será crear un contenedor al que le aplicaremos la imagen del diseño

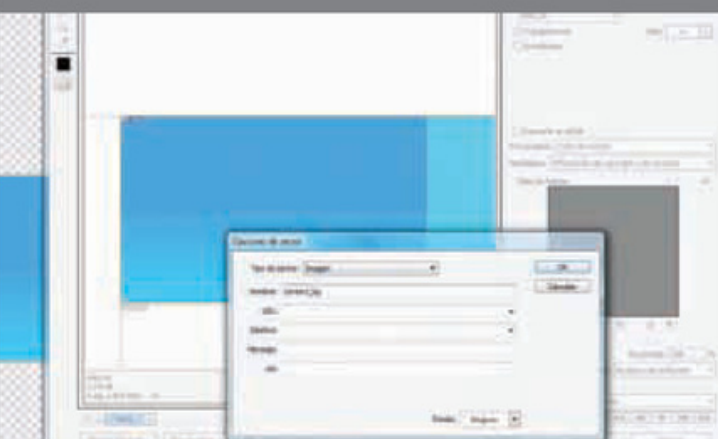
```

<div class="title_section"><span>Contáctenos</span></div>
</div>
<div id="content">
<div class="title_section"><span>Contáctenos</span></div>
<div class="clearfix"></div>
<div class="intro2">
<p>Para poder garantizarle un mejor servicio, por favor utilice nuestro formulario de contacto. También puede ponerse en contacto con nuestras sedes directamente. Si está interesado en recibir por favor complete el formulario apropiado y estaremos encantados de atenderle.</p>
</div>
<div class="main_contact">
</div>
</div>
<div id="Footer">
&ltdiv class="font_footer">

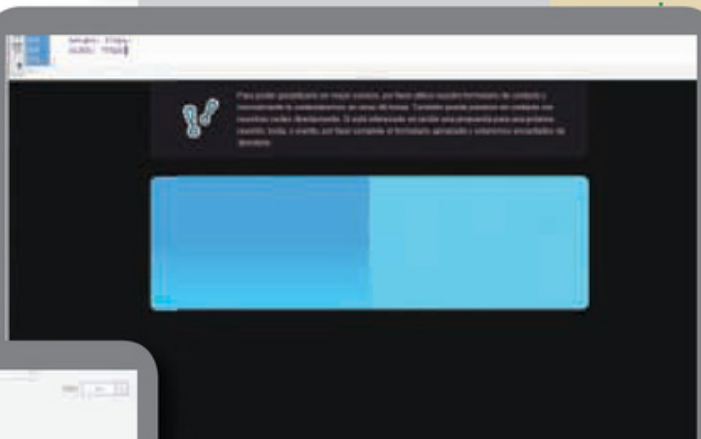
```

CONTACTO

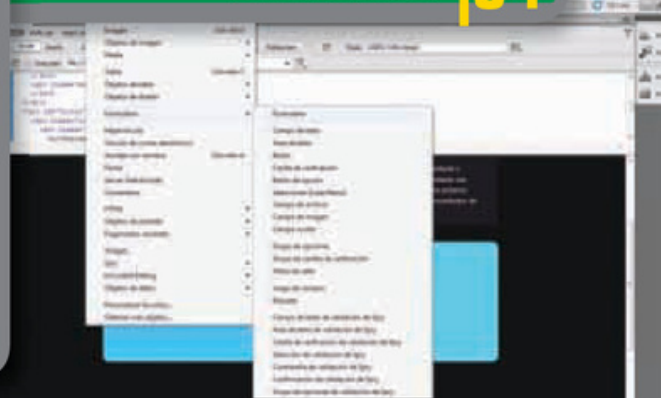
33 | CONTENEDOR FORMULARIO



32 | FONDO DEL FORMULARIO



FONDO DEL FORMULARIO II | 34



ESTRUCTURA FORMULARIO | 35

como background. Para hacerlo, la marcamos en Photoshop y la guardamos como **contact_bg.png**.

PASO | 33 El código HTML del contenedor del formulario es el que podemos ver a continuación:

```

<div class="main_contact">
</div>

```

PASO | 34 Acto seguido, ingresamos las reglas CSS necesarias para darle estilo al fondo del formulario: `[.main_contact{ background: url("../img/contact_bg.png") no-repeat scroll 0 0 transparent; float: left; height: 214px; width: 700px;}]`. El resultado será el que vemos en la imagen.

PASO | 35 Ahora, armaremos la estructura del formulario dentro del contenedor creado. Para hacerlo, vamos al menú **[Insertar]**, hacemos clic en **[Formulario]** y, en el desplegable que aparece, elegimos **[Formulario]**.



PÁGINA DE CONTACTO

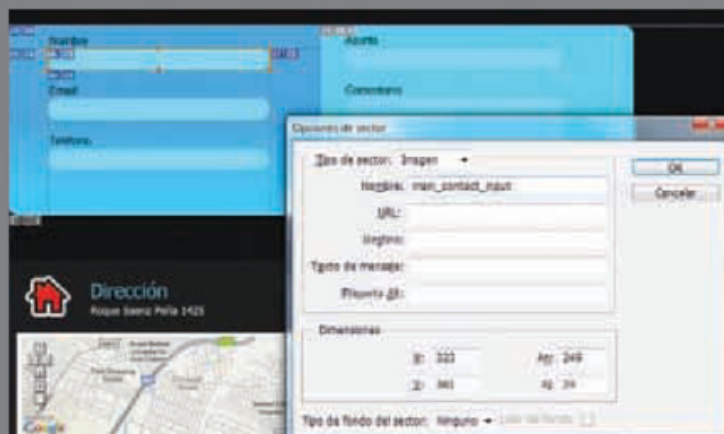
PASO A PASO



36 | CONTENEDORES FORMULARIO

```
<div class="main_contact">
  <form method="post" action="#">
    <div class="box_left">
      <label>Nombre</label>
      <input type="text" size="37" value="" name="your-name"/>
      <label>Email</label>
      <input type="text" size="37" value="" name="your-name"/>
      <label>Teléfono</label>
      <input type="text" size="37" value="" name="your-name"/>
    </div>
    <div class="box_right">
      <label>Asunto</label>
      <input type="text" size="37" value="" name="your-name"/>
      <label>Comentario</label>
      <textarea rows="4" cols="78" name="your-message"></textarea>
      <input type="submit" value="Enviar"/>
      <input type="reset" value="Borrar"/>
    </div>
  </form>
</div>
```

ETIQUETAS FORMULARIO | 37



38 | IMAGEN CAMPO

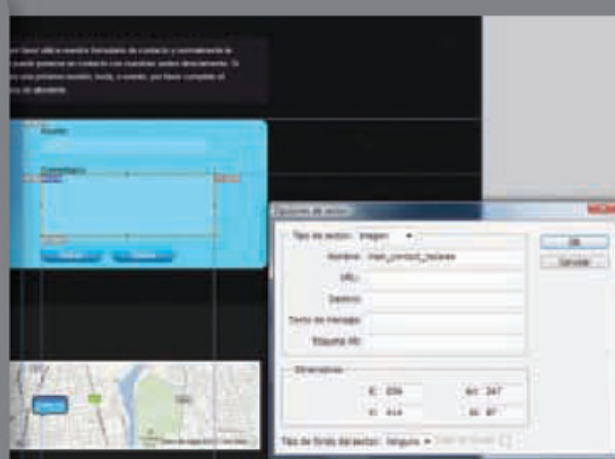


IMAGEN TEXTAREA | 39

PASO | 36 Luego, creamos dos contenedores dentro del formulario, uno de ellos para agrupar los elementos a la izquierda y otro destinado a los elementos de la derecha, tal como podemos observar en la imagen.

PASO | 37 Vamos otra vez al menú [Insertar] y, desde las opciones de [Formulario], agregamos las etiquetas de acuerdo con la imagen, que deberemos dividir en los contenedores (las tres primeras a la izquierda y las últimas dos a la derecha).

PASO | 38 A continuación, asignamos estilos a los campos de entrada de texto. Para hacerlo, marcamos el sector correspondiente a uno de ellos en Photoshop y lo guardamos como **main_contact_input.png**.

PASO | 39 Hacemos lo mismo con el diseño del cuadro de texto (**textarea**) para el comentario, que nombramos como **main_contact_textarea.png**.

PASO | 40 A continuación, aplicamos las reglas CSS que vemos en el código:

```
.main_contact{ background:url("../img/contact_bg.png") no-repeat scroll 0 0 transparent; float:left; height:214px; width:700px; }.main_contact .box_left{ float:left; padding-left:46px; width:303px; }.main_contact .box_right{ float:right; margin-right:45px; width:257px; }.main_contact label{
```

```

main_content_div_left { float: left; padding-left: 40px; width: 300px; }
main_content_div_right { float: right; margin-right: 40px; width: 250px; }

main_content_label {
  display: block;
  font-size: 15px;
  margin-bottom: 3px;
  margin-left: 4px;
  margin-top: 13px;
}

main_content_input[type="text"] {
  background: url("../img/main_content_input.png") no-repeat scroll 0 0 transparent;
  border: medium none;
  height: 23px;
  padding-left: 4px;
}

main_content_password {
  background: url("../img/main_content_password.png") no-repeat scroll 0 0 transparent;
  border: medium none;
  height: 23px;
  padding-left: 4px;
  width: 250px;
}

```

```
<form method="post" action="mail.php">
```

401 ESTILOS FORMULARIO

[illegible]

Contactar

 Para obter resultados em tempo real, por favor utilize nossos kits de teste de diagnóstico de referência em casa. O teste "point of care" permite ao usuário realizar o teste em casa. O teste é realizado em menos de 15 minutos e fornece uma resposta para uma primeira decisão, seja, a nível, por favor, consulte o formulário anexo e informe o resultado da decisão.

```

        .contact_box input[type="submit"], .main_contact input[type="submit"], .main_contact input[type="reset"] {

```

```
.contact-box input[type="submit"]:hover, .main-contact input[type="submit"]:hover, .main-contact input[type="reset"]:hover { color: #025F85; }
```

421 PROPIEDAD ACCION

[illegible]

41 | ESTILOS BOTONES

```
display:block; font-size:15px;
margin-bottom:3px; margin-left:9px;
margin-top:13px; } .main_contact
input[type="text"]{ background:url
("../img/main_contact_input.png") no-repeat
scroll 0 0 transparent; border:medium
none; height:25px; padding-left:8px; }
```

```
.main_contact textarea{
background:url("../img/main_contact_
texarea.png") no-repeat scroll 0 0
transparent; border:medium none;
height:87px; padding-left:6px;
width:250px; }
```

PASO | 41 Como los botones **Borrar** y **Enviar** deberían de tener los mismos estilos que aplicamos a los utilizados en el footer, buscamos las reglas anteriormente realizadas para ellos y las modificamos tal como podemos ver en la imagen.

PASO | 42 Para darle funcionalidad al formulario, debemos generar un archivo PHP o ASP según el servidor. En nuestro caso, será un PHP denominado **mail.php**, nombre que colocamos en la estructura del formulario dentro de la propiedad **[action]**.

PÁGINA DE CONTACTO

PASO A PASO

```
<div class="main_content">
  <div method="post" action="mail.php">
    <div class="box_left">
      <label>Nombre</label>
      <input type="text" size="37" value="" name="nombre"/>
      <label>Email</label>
      <input type="text" size="37" value="" name="email"/>
      <label>Teléfono</label>
      <input type="text" size="37" value="" name="telefono"/>
    </div>
    <div class="box_right">
      <label>Asunto</label>
      <input type="text" size="37" value="" name="asunto"/>
      <label>Comentario</label>
      <textarea rows="4" cols="28" name="mensaje"></textarea>
      <input type="submit" value="Enviar"/>
      <input type="reset" value="Borrar"/>
    </div>
  </div>
</div>
```

[illegible]

43 | PROPIEDAD NAME

ARCHIVO MAIL.PHP | 44

The screenshot shows a MATLAB script in the editor and its corresponding plot. The script defines a function `f` and calculates the integral of `f` from 0 to 1 using the `integral` function. The plot displays a series of green dots forming a quarter-circle arc in the first quadrant of a Cartesian coordinate system.

```

function f = f(x)
    f = 1 - x^2;
end

% Calculate the integral of f from 0 to 1
I = integral(f, 0, 1);

% Display the result
disp(I);

```

The plot shows a series of green dots forming a quarter-circle arc in the first quadrant of a Cartesian coordinate system. The dots are arranged in a grid, with the arc starting at (0, 1) and ending at (1, 0).

[illegible]

451 ASUNTO Y MENSAJE

ICONOS CONTACTO | 46

PASO | 43

A continuación, dentro de la propiedad **[name]** de cada etiqueta **[input]** o **[textarea]**, colocamos el nombre con el que haremos referencia al campo en el archivo PHP. Como sabemos, éste es el nombre que asignamos a cada variable.

PASO | 44

Dentro del archivo **mail.php**, primero definimos, dentro de una variable, a qué dirección de e-mail debe ir el formulario.

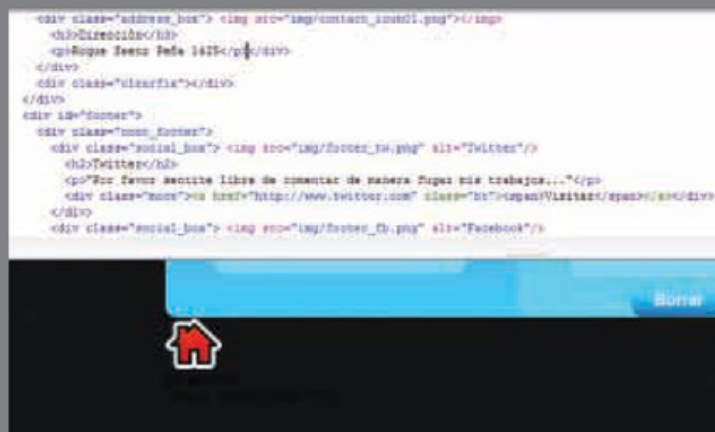
Luego, definimos las cabeceras (la primera, con la versión de programa; la segunda, con el tipo de caracteres que se usarán; y la tercera, con el remitente, a dónde responder y el programa que lo genera). En **[@mail]** colocamos todas las variables que enviaremos, y en **[header]** indicamos a qué web debe direccionar el formulario una vez realizado el proceso.

PASO | 45 Para terminar, entre la variable **[\$para]** y la primera variable **[Scabecera]**

escribimos aquellas que se deben tomar del formulario, incluyendo una variable **[`$asunto`]** (ésta imprimirá el asunto en nuestro correo) y las variables **[`$mensaje`]**, que dependiendo de la propiedad **[`name`]** de cada etiqueta **[`input`]** o **[`textarea`]**, recolectarán la información del formulario.

PASO | 46 Para crear los cuadros que contienen las diferentes formas de contacto, en Photoshop marcamos y guardamos las imágenes correspondientes a cada recuadro. En este caso, las nombramos como **contact_icon1.png**, **contact_icon2.png**, etcétera.

PASO | 47 Ahora, vamos a crear la estructura de un solo cuadro, la cual duplicaremos para que se adapte a los diferentes contenidos que le incluyamos.



47 | RECUADROS CONTACTO

```

.address_box {
background-color: #333333;
border-bottom: 3px solid #444444;
float: left;
margin-top: 20px;
padding-bottom: 10px;
padding-top: 10px;
width: 700px;
}

.address_box img { float: left; margin-left: 10px; margin-right: 10px; }

.address_box h3 {
color: #444444;
display: block;
font-size: 12px;
font-weight: lighter;
margin-bottom: 7px;
}

.address_box p { color: #ffffff; font-size: 12px; }

.address_box a { color: #ffffff; text-decoration: none; }

.address_box a:hover { color: #444444; }

.address_box iframe { border: 1px solid #ffffff; margin-bottom: 10px; margin-top: 10px; }

.address_box a:link { font-size: 12px; margin-left: 10px; }

```

```

<div class="address_box">

<h3>Dirección</h3>
<p>Rogee Saenz Peña 1425</p>
</div>
<div class="social_box">
<div class="social_box">

<h3>Twitter</h3>
<p>Por favor envíame los comentarios de manera rápida por los trabajos...</p>
<div class="social_box">

</div>
</div>
<div class="social_box">


```

48 | ESTILOS CONTACTOS

 Dirección
Rogee Saenz Peña 1425

 E-mail
artedigital@artedigital.com

 MSN
artedigital@artedigital.com

 Teléfono
(011) 4444-4444

49 | FORMAS DE CONTACTO

```

<div class="address_box">
</img>
<h3>Teléfono</h3>
<p>(+5411) 5555-5555</p>
</div>

```

PASO | 48 A continuación, creamos las reglas CSS tal como vemos en la imagen.

PASO | 49 Duplicamos el código HTML del primer cuadro y agregamos los contenidos específicos de cada uno.

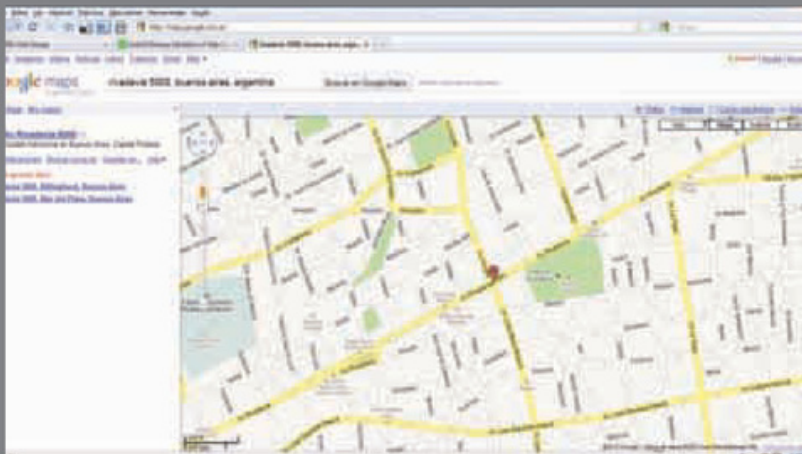


PÁGINA DE CONTACTO

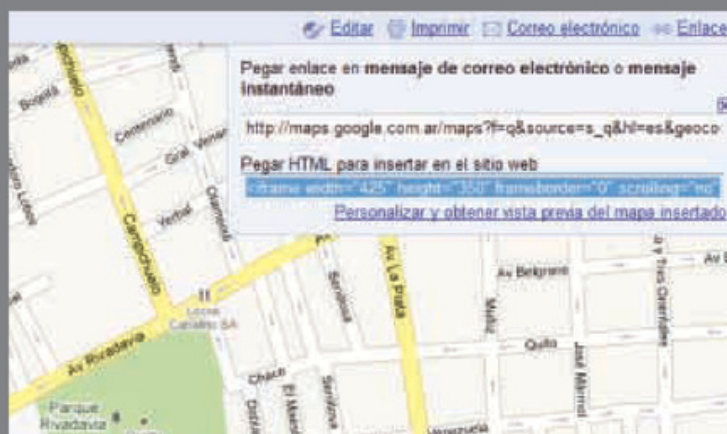
PASO A PASO



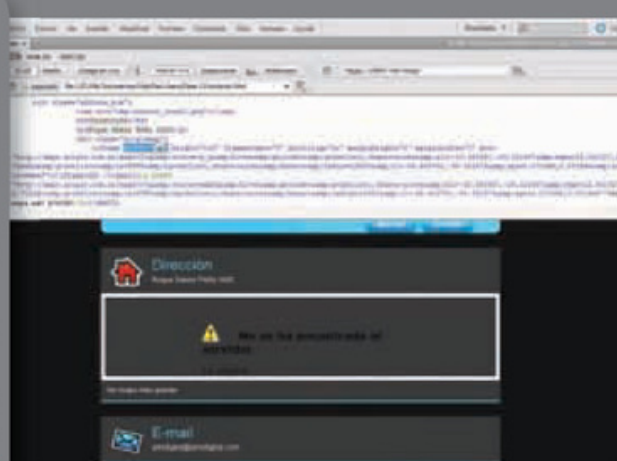
50 CONTENEDOR MAPA



INGRESAR DIRECCIÓN **51**



52 ENLACE AL MAPA



ANCHO DEL MAPA **53**

PASO | 50 Para agregar el mapa de Google Maps, lo primero que haremos será crear un contenedor dentro del primer div `[address_box]` insertado.

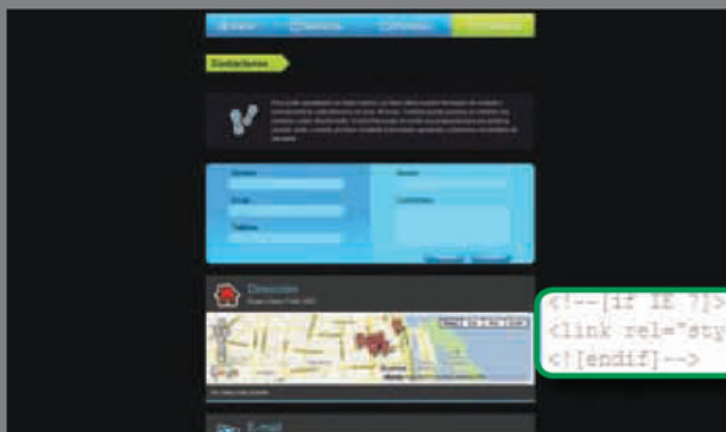
PASO | 51 Nos dirigimos al sitio de Google Maps correspondiente a nuestro país (en este caso, www.maps.google.com.ar) e introducimos la dirección que deseamos ubicar en el mapa.

PASO | 52 Hacemos clic en `[Enlace]` y, en el desplegable que aparece, seleccionamos el código ubicado debajo del título `[Pegar HTML para insertar en el sitio web]`. Luego, pegamos el código dentro del primer div `[address_box]`.

PASO | 53 En el código que pegamos, buscamos el atributo `[width]` dentro de la etiqueta `[iframe]` y cambiamos su valor a `[685]`.

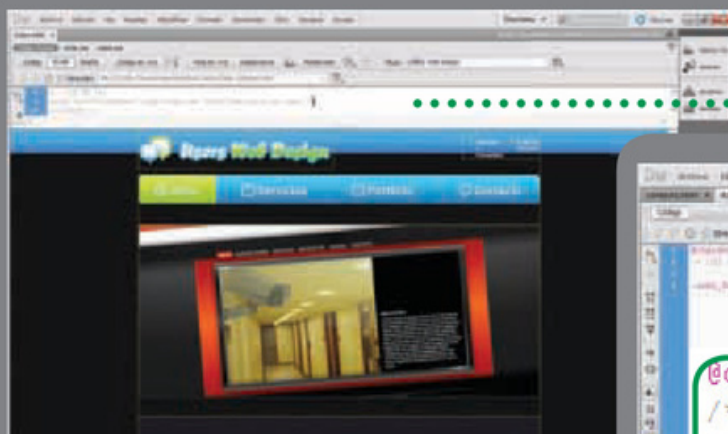
PASO | 54 Ahora, revisamos la página en el navegador para verificar si funciona de manera correcta.

PASO | 55 Una vez que hemos terminado el maquetado, tendremos que verificar si el sitio se ve bien en diversos navegadores. En este caso, en Firefox logramos un buen resultado, pero encontramos inconvenientes con Internet Explorer 7. Para solucionarlo, dentro del header de nuestras páginas HTML colocamos el código siguiente:

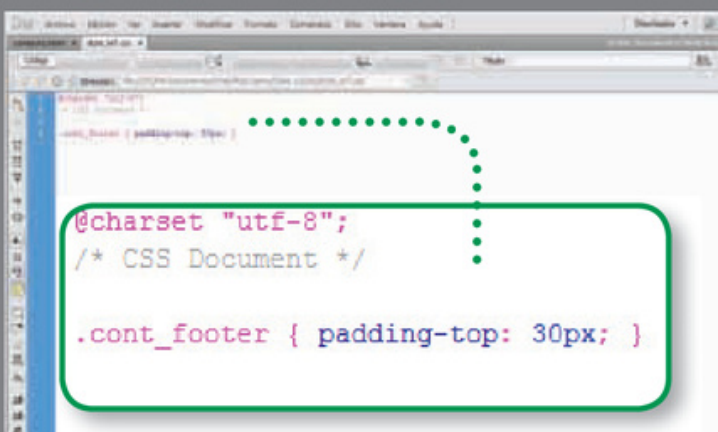


54 | PÁGINA DE CONTACTO

```
<!--[if IE 7]>
<link rel="stylesheet" type="text/css" href="css/style_ie7.css" />
<![endif]-->
```



55 | SOLUCIÓN IE7



56 | ESTILOS IE7

```
<!--[if IE 7]>
<link rel="stylesheet" type="text/css"
href="css/style_ie7.css" />
<![endif]-->
<!--[if IE 8]>
<link rel="stylesheet" type="text/css"
href="css/style_ie8.css" />
<![endif]-->
```

```
.cont_footer {
    padding-top: 30px;
}
```

PASO | 56 Creamos un nuevo archivo CSS llamado **style_ie7.css** y, dentro de él, incluimos las reglas que aparecen en la imagen. Como vemos, sólo tenemos que modificar una propiedad de una sola regla para que el maquetado se vea igual en Internet Explorer 7.

RESUMEN

Hasta aquí, hemos terminado con el maquetado del sitio. Si bien podríamos aplicar algunas técnicas para simplificar su actualización en el futuro, tenemos la base para poder seguir investigando y potenciar el sitio por nuestra cuenta. Para investigar el funcionamiento del slide de la página principal, podemos descargar los archivos fuente que se encuentran en el sitio de internet (www.redusers.com/cursos/webdesign).

ACTIVIDADES



PRÁCTICAS

01 Cree una página de servicios.

02 Incluya una frase destacada en la página de servicios y cree un cuadro destacado para ello.

03 Realice una lista con los servicios.

04 Cree una página portfolio.

05 Genere un submenú de navegación en la página de portfolio.

06 Escriba las reglas del submenú de navegación, anteriormente creado, en **CSS**.

07 Cree las fichas de proyectos con: título, imagen y una línea de texto explicativa.

08 Defina las fichas de proyectos como listas.

09 Convierta en bloque los elementos internos de la lista.

10 Defina las propiedades **CSS** para los elementos internos de la lista.

11 Cree una página de detalle de cada uno de los proyectos.

12 Cree un **botón volver** para regresar al sitio principal.

13 Genere una página de contacto.

14 Inserte un formulario de contacto en la página recientemente creada.

15 Agregue el mapa de **googlemaps** con la ubicación del cliente.





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, éste debe encontrarse alojado en un servidor. Éste 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 éste 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.

EMISOR



RECEPTOR



EMITE LA REPETICIÓN

PROCESA Y ENVÍA LA RESPUESTA

El **cliente** es el equipo que solicita la información. El **servidor** es el encargado de dar respuesta a esa petición.



Los clientes, a través del navegador web, se comunican con el servidor para visualizar las páginas de los sitios.



Un servidor proxy se ubica entre los clientes y el servidor web, para filtrar la información.

SERVIDOR PROXY

- **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 del 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 del 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	0001	0011	0001	0011
0101	0110	0101	0110	0101	0110
0001	0011	0001	0011	0001	0011
0101	0110	0101	0110	0101	0110
0001	0011	0001	0011	0001	0011
0101	0110	0101	0110	0101	0110



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 utilicemos, 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*),



En una red local, cada equipo debe tener su propia dirección IP, única en la red.

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,

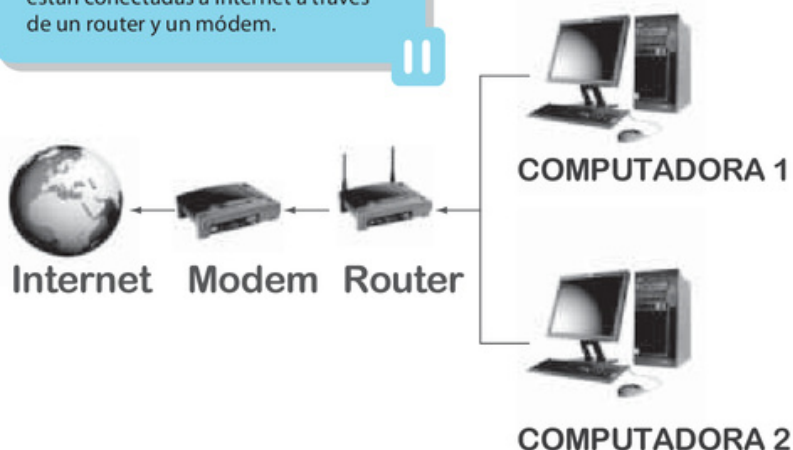
separadas por un punto (por ejemplo, 64.233.163.104). Podríamos decir que si Internet fuese una autopista, las direcciones IP serían las patentes de los autos que circulan por ella. Es posible distinguir entre diferentes tipos de direcciones IP:

► **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,

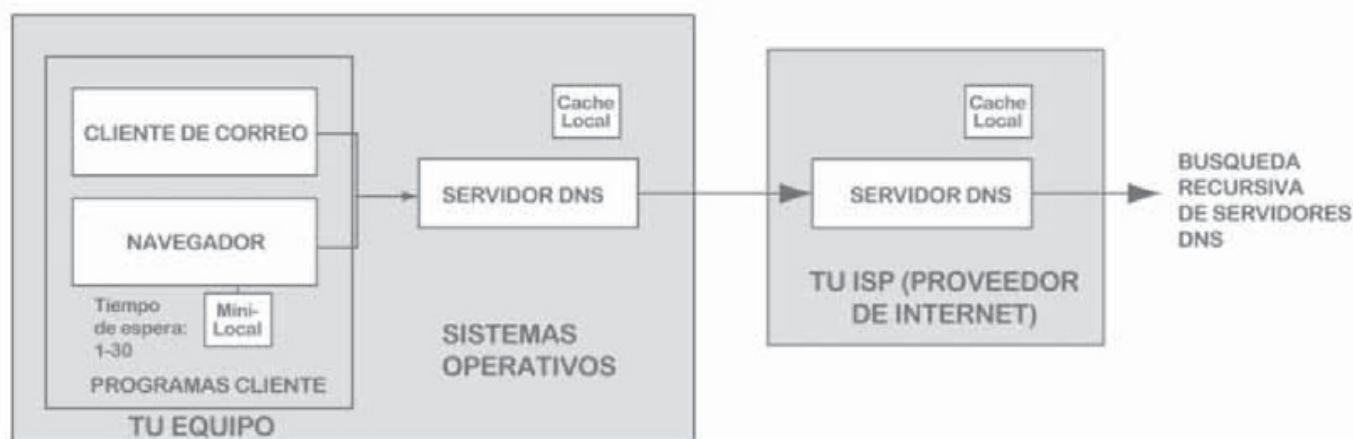
llamada **Top level domain** (dominio de nivel superior). Esto corresponde al **.com** en **www.google.com**, o al **.org** en **www.wikipedia.org**. Cada etiqueta hacia la izquierda se conoce como **subdominio**. Cada uno de ellos debe tener al menos un carácter y un máximo de 63, sólo alfanuméricos, 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*). Ésta 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 | ▶ Éste 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 | ▶ Éste 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 ésta 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. ◀



ELEGIR UN NOMBRE DE DOMINIO

Llegó el momento de elegir un nombre para nuestro sitio. Aunque ésta 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 sólo 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', 'TRANSACTIONS', 'PERSONALS', etc. Below that, a search bar contains 'disenoweb'. The main section is titled 'Congratulations! The following domains are available:'. It shows a grid of domain suggestions with their respective prices. For example, 'disenoweb.com' is listed for \$15.49/yr. There are also filters for different domain types like .com, .net, .org, etc.

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 eslógan.

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 adulto.

► 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, ésta 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 conflictos con otros sitios.



DOMINIOS REGIONALES

En muchos casos, se opta por elegir sólo el dominio del país propio (por ejemplo, **.com.ar**) para localizar más las búsquedas. Sin embargo, éste 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 sólo a esa nación. Como vemos, lo más aconsejable es adquirir los dos dominios.

► 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.

YAHOO! MUJER



En **http://ar.mujer.yahoo.com** encontramos un ejemplo de subdominio del portal de Yahoo!

► 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. ◀

Domain-Generator.net

Domain Combination Checker

Domain Name Generator | Domain Tester | Domain Combination Checker | Business Name Generator | Bulk Domain

Asix by Google:
Studio Ammos
Hosting
EU Shared, Dedicated,
VPS Hosting As of:
€0.79/mo, Plesk,
Domains, ...

Com.Au Domain
Names \$15yr
Australian Com.Au
Domain Names No Set
Fees, Fast and Free
DNS
Start Here

Pizzeria Interactiva
Pizzerias que
revolucionan la forma
de enseñar en el aula.
Ingresar
www.interactiva.com.ar

Domain Name
Get tons of

Report

Domain	Status
turismo-turismo.com	Test ALL Available
argentina-argentina.com	Registered
argentina-turismo.com	Registered
argentina.com	Registered
argentinaargentina.com	Registered
argentinaturismo.com	Registered
turismo-argentina.com	Registered
turismo.com	Registered
turismoargentina.com	Registered
turismoturismo.com	Registered

Back to Domain Name Main Page

Domain
Generator

Word 1: argentina
Word 2: turismo
Word 3:
Word 4:
Word 5:

Domain Generator crea combinaciones de dominios con las palabras ingresadas (**www.domaingenerator.net**).

¿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

PASO A PASO

UN DOMINIO

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



INICIAR REGISTRO 02

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

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:

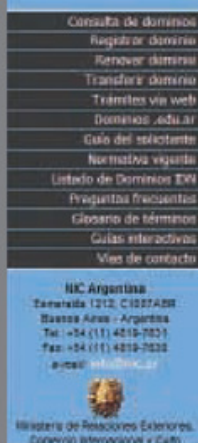
velezp@gmail.com

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 solicitante manifiesta conocer y aceptar las reglas, procedimientos e instrucciones.

03 ELEGIR DOMINIO



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 velezp@gmail.com

Actividad Diseño de Páginas Web

Seleccione el tipo de entidad que corresponde

Registrante, persona física y residente en Argentina

Registrante, persona jurídica y residente en Argentina

Registrante y residente en el extranjero

continuar

04 DARNOS DE ALTA



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

numero 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 ésta 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.

06 | RESERVAR O DELEGAR

FINALIZAR EL REGISTRO | 08

07 | PROVEEDOR ISP

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 éstos 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.toweb.com) brinda hosting pago con distintas variantes y opciones, y permite elegir si el servidor tendrá sistema **Linux** o **Windows**.



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.

que Windows está orientado a programadores de ASP, ASP.NET, bases de datos Access y MySQL, y a usuarios que publiquen su web mediante las extensiones de FrontPage.

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-

lidad de usar **dominios** y **subdominios**. Los primeros son las direcciones posibles de nuestra página web (por ejemplo, www.mipagina.com). Los segundos son las secciones que podemos hacer para que el usuario acceda a nuestra web de manera más directa (por ejemplo, <http://ventas.mipagina.com> o <http://info.mipagina.com>). Según el servicio que contratemos, los subdominios pueden ser ilimitados o tener un número determinado. Es importante conocer las **tecnologías** provistas por el servidor y verificar que el servicio brinde soporte a aquellas que

pueden servirnos en el desarrollo del sitio, como PHP, ASP, Perl, CGIs, 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.

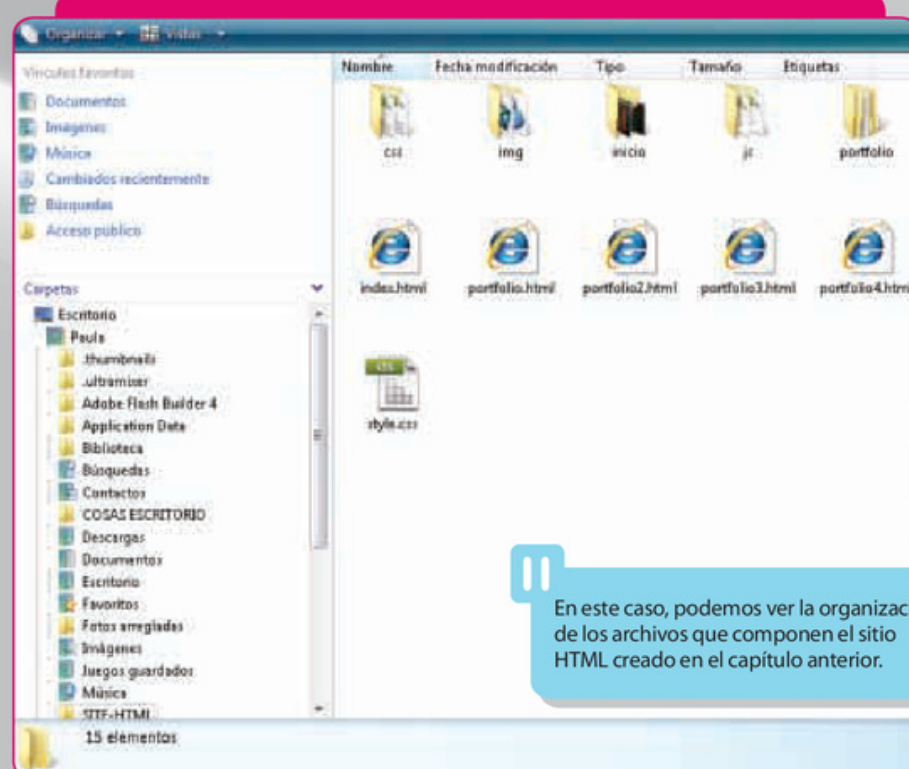
FTP (FILE TRANSFER PROTOCOL)

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, éstos 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.

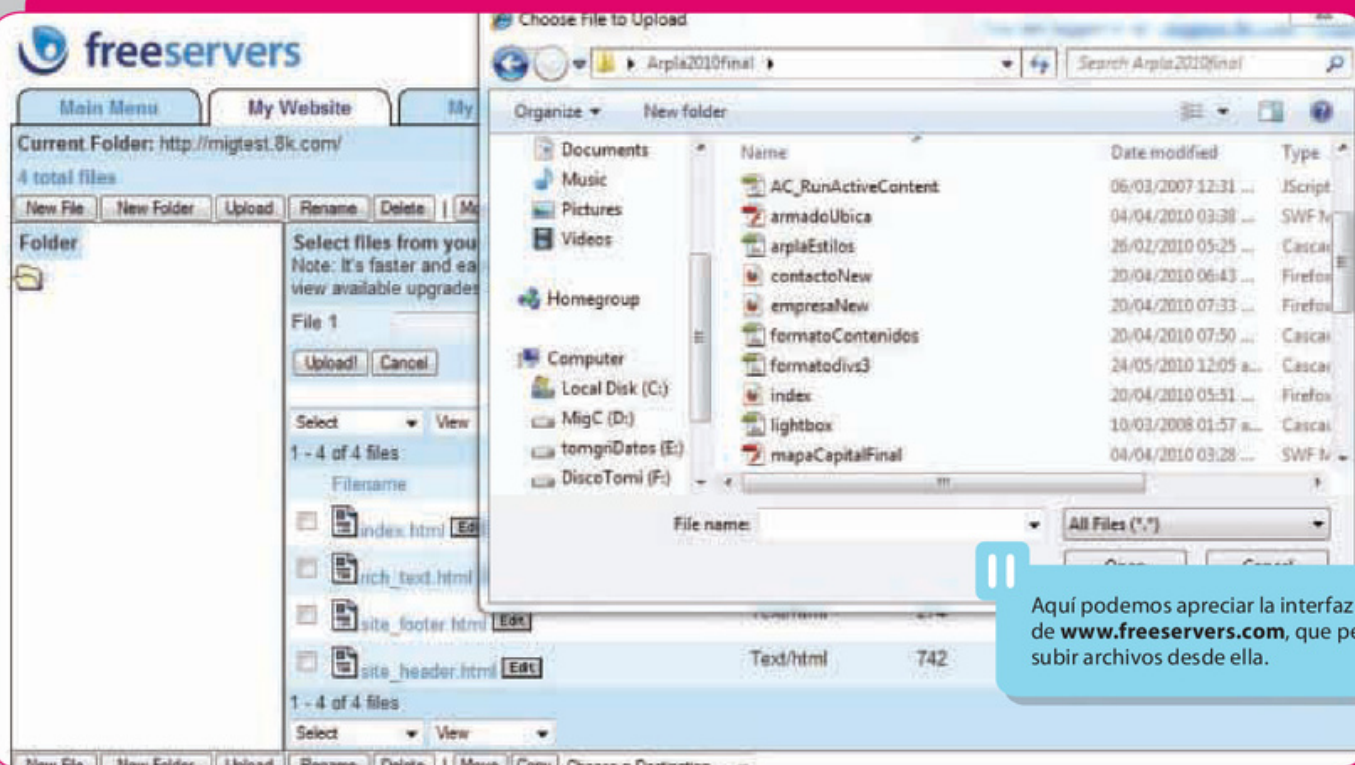
	FTP CLIENT	INTERFAZ WEB
USO IDEAL	▶ Grandes volúmenes de archivos.	▶ Pocos archivos por sitio.
PESO DE ARCHIVOS	▶ Archivos de pesos variados.	▶ Hasta un limitado peso en KB.
DIRECTORIOS	▶ Visualización simultánea de los directorios.	▶ Visualización de un directorio por vez.
TRANSFERENCIA	▶ Posibilidad 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.**



Aquí podemos apreciar la interfaz web de **www.freeservers.com**, que permite subir archivos desde ella.

SUBIR UN SITIO AL PASO A PASO

Llegó el momento de subir los archivos a Internet para que los usuarios puedan verlos.

FileZilla The free FTP solution

Home

FileZilla

Features
Screenshots
Download
Documentation

FileZilla Server

Download

General

Contact

Client Download

The latest stable version of FileZilla Client is 3.3.4.1

Please select the file appropriate for your platform below

Windows

FileZilla_3.3.4.1_win32-setup.exe (recommend)

FileZilla_3.3.4.1_win32.zip

Windows XP, Vista and Windows 7 are supported

01 | DESCARGA Y DATOS

Gestor de sitios

Seleccione el sitio:

Mis sitios

Hosting gratuito

General

Servidor: ftp.domenoweb.com

Puerto:

Server Type: FTP - File Transfer Protocol

Login Type: Normal

Usuario: pveloz

Contraseña: *****

Cuenta:

Comentarios:

Nuevo sitio

Nueva carpeta

Nuevo servidor

Renombrar

Borrar

Copiar

Conectar

Aceptar

Cancelar

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. Éstos 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

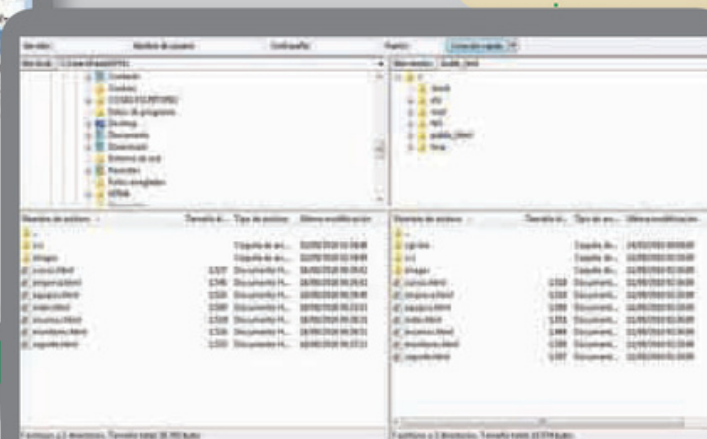
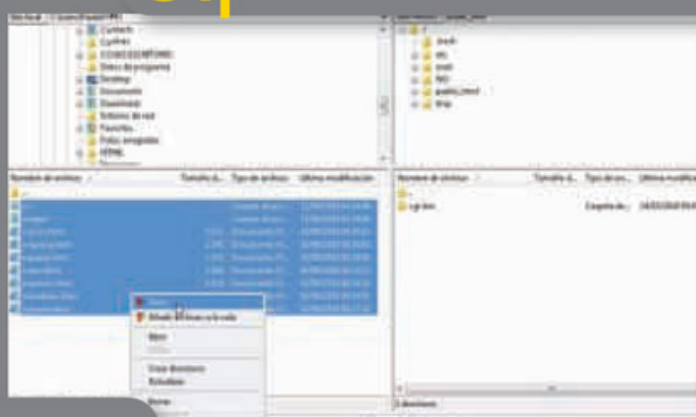
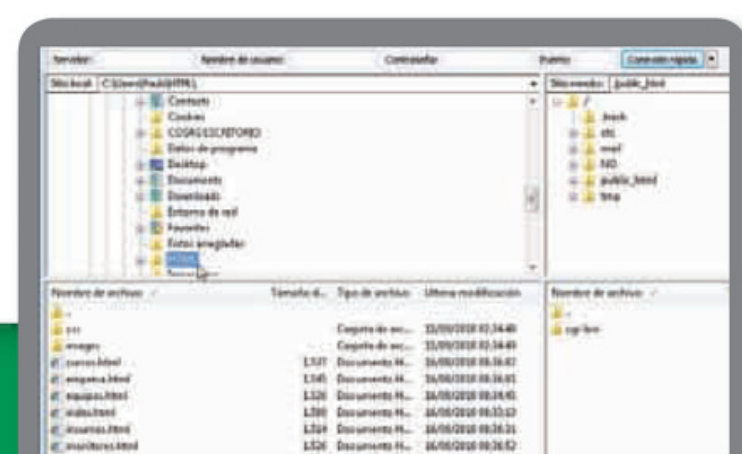
SUBIR ARCHIVOS | 05

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



FINALIZAR Y VERIFICAR | 06

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.

ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 ¿Qué tipo de servidores existen?

02 ¿Cuál es la función del receptor?

03 ¿En qué consiste el modelo cliente/servidor?

04 ¿De dónde vienen los textos e imágenes cuando navegamos?

05 ¿Qué son los protocolos?

06 ¿Cuáles son los protocolos más conocidos en Internet?

07 ¿Cuáles son los diferentes tipos de direcciones IP?

08 ¿Cómo está formado un dominio?

09 ¿Cuáles son los tipos de dominios?

10 ¿Qué es la ICANN?

11 ¿Cómo registrar un dominio?

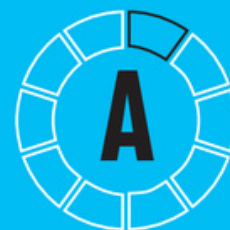
12 ¿Cuáles son las consideraciones al registrar un hosting pago?

13 ¿Qué es el FTP?

14 ¿Cómo podemos subir archivos mediante una interfaz web?

15 ¿Cuáles son las principales diferencias entre subir los archivos por FTP y mediante la interfaz web?





Vida laboral

- 
- ▶ LA SALIDA LABORAL
 - ▶ WORKFLOW DE TRABAJO
 - ▶ ¿EMPREENDEDOR, EMPLEADO O FREELANCE?
 - ▶ ¿CÓMO INICIARSE?
 - ▶ CONSEGUIR CLIENTES ONLINE
 - ▶ EMPREENDEDOR, ¿YO?
 - ▶ EL CLIENTE Y SUS NECESIDADES
 - ▶ PLANIFICAR UN PORTFOLIO EFECTIVO
 - ▶ OPORTUNIDAD EN LAS IDEAS
 - ▶ DETERMINAR EL PRECIO DEL TRABAJO
 - ▶ MODELO DE NEGOCIO



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. Éste puede ser dividido 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 sólo 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. Éste 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 empre-

sas especializadas que ya están establecidas en el mundo del desarrollo web hace que tengamos la posibilidad de participar en proyectos para grandes marcas. Como éstos 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 por 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

Éste 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 laborales.

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 sólo 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 EMPREENDEDOR

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 éstos en general no serán fruto de la labor directa del emprendedor, ya que no serán hechos por sus propias manos. En cambio, éste 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. Éste 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 dedicar 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 éstos 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. Ésta es una excelente manera de darnos a conocer e, incluso, de descubrir nuevos proyectos en los que podremos participar.

PRESUPUESTAR CORRECTAMENTE

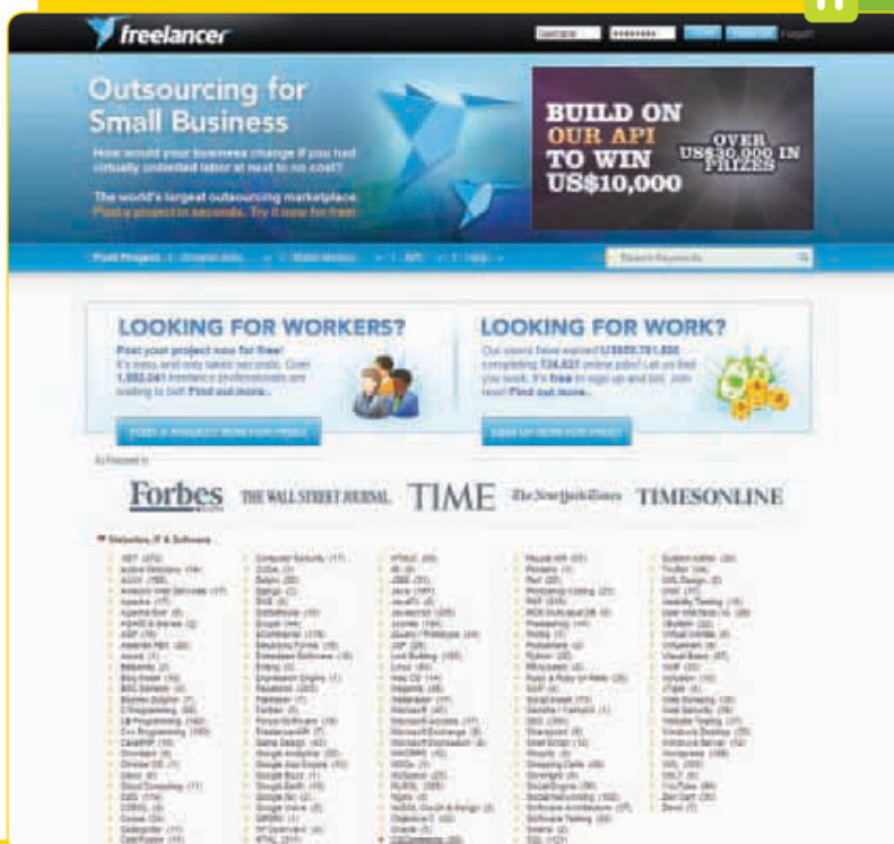
Una de las mayores dudas que tienen los profesionales freelance está relacionada con lo que deben cobrar por su trabajo. La realidad es que debemos hacer valer la calidad de lo que realizamos, ofreciendo un precio razonable y justo. No necesariamente vamos a tener clientes si cobramos poco, ya que lo 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: 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 sólo 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 sólo 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 sólo 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. Afirmo ser utilizado por las empresas más importantes de cada nación en la que está presente. Por su parte, la bolsa de empleo argentina **Búmeran** (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. ◀



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. Ésta es la visión que nos impulsa a seguir, la que nos dice en dónde estamos, dónde queremos estar y cómo haremos para llegar ahí. ◀

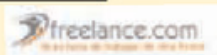


EMPLEADO VS. FREELANCE

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.

FREE LANCE

Veamos el camino que debe recorrer el profesional del trabajo independiente.



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.





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.



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, sólo 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 sólo 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 sólo 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 CSS**. 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

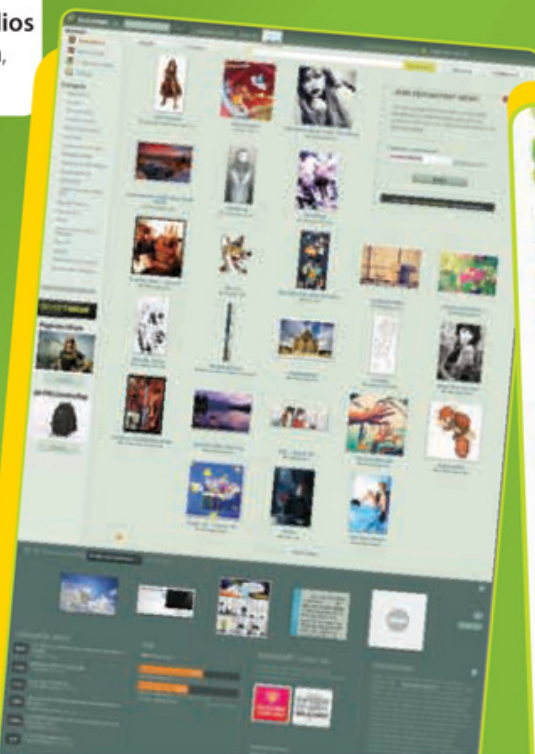
Ya vimos la importancia que tiene la creación de un portfolio online para un desarrollador web. En esta ocasión, analizaremos las estrategias para poseer uno interesante y atractivo destinado a los potenciales clientes.

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 sólo 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 sólo 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 pos 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, sólo 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 contenido

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 éste 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 CSS. 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 éstos 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? Éstas 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 insueme 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.
- ▶ Ahorra 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 sólo 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?". Sólo 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 sólo 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.

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

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.



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 las capacidades de los mismos?

► **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 prospectos 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 ésta 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 aquél 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 sólo 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. ◀



ACTIVIDADES



TEST DE AUTOEVALUACIÓN

01 Mencione algunos ejemplos de áreas de trabajo en que se puede especializar.

02 ¿Cuáles son las diferencias entre trabajar de manera freelance y hacerlo en relación de dependencia?

03 ¿Cuáles son las recomendaciones de equipamiento para un diseñador freelance?

04 ¿Cuáles son las diferentes etapas de nuestro trabajo?

05 ¿Qué sitios se recomiendan para los profesionales independientes?

06 ¿Cuáles son algunas de las desventajas del trabajo freelance?

07 Además del trabajo de creación, programación y diseño ¿qué otras tareas debe encarar el diseñador freelance?

08 ¿Cuáles son las recomendaciones más importantes para el trabajador freelance?

09 ¿De qué manera podemos encontrar nuevos clientes?

10 ¿Cuáles situaciones del contexto nos pueden conducir a una buena idea?

11 ¿Cuáles son las principales áreas de una empresa?

12 ¿Cómo se generan las ganancias económicas?

13 ¿Cómo diferenciarse del resto de los profesionales?

14 ¿Cómo podemos mejorar nuestro emprendimiento?

15 ¿Cómo organizarse para brindar los servicios?

CURSO VISUAL Y PRÁCTICO



Servicios al lector



► ÍNDICE TEMÁTICO

► CATÁLOGO



ÍNDICE TEMÁTICO

B

Background-attachment	155
Background-color	154
Background-image	154
Background-position	155
Background-repeat	154
Bordes	150/151
Botón file	204
Botón image	204
Botón radio	203
Botón reset	204
Botón submit	204
Brújulas	020
Buscadores	014/015

C

Caja de selección o validación	204
Campo hidden	204
Captcha	199
Checkbox	204
Colores plenos	152
Coming soon page	030/031/032/033
Concatenar	215
CSS Sprites	189/190/191/192/193/194/195

D

Diagrama del sitio	045
Domínios genéricos	268
Domínios regionales	271

E

Emprendedor	300/301
Error404	034/035/036/037
Estructura en árbol	045
Estructuras de Control	214
Etiqueta input	202
Etiqueta label	207
Etiqueta legend	207
Etiqueta para texto largo	205
Etiqueta radio	203
Etiquetas fieldset	207

F

Firefox	015/093
Flash	023/024
Fondos compuestos por varios fondos	153
Formulario básico	198
Formulario de contacto	198
Formularios avanzados	198
Formularios de compra	199
Formularios de login	199
Formularios de registro	199
Formularios de suscripción	198
Fotogramas	024
Freelance	292/293/294/295/296/297/298/299
FTP	278/280/281
FutureSplash Animator	024/025



G

GIF	180/181
Google	014/015/041/043/093
Google Chrome	015/093
Gradientes	152
Grilla	020/021

I

ICANN	269
Includes	214
Interfaz web	279
Internet Explorer	015/093

J

JPG	180/181
-----	---------

L

Layout en Photoshop	046/047/048/049/050/051/132/147
Layout líquido	147
Lenguajes de cliente	213
Lenguajes de servidor	213
Lista de opciones	206
Listas anidadas	163
Listas con imágenes	164
Listas en menú de navegación	165
Listas estándar	164
Listas no ordenadas	163
Listas ordenadas	162
Logo	039/045/072
Longdesc	179

M

Mapa de imágenes	184/185/186/187
Margen	148
Menú de navegación horizontal	169
Menú de navegación vertical	169
Menús con listas	163
Modelo cliente/servidor	262
Motor de armado	026

N

Name	179
Navegación con el mouse	038
Navegación sin clics	038
Navegación vertical	038
Navegadores	014/015/093
Netscape Navigator	015

O

Opera	015/093
-------	---------

P

Padding	149
Patrones	153
Personalizar listas	166/167
PNG	180
Protocolos	267

R

Registro de un dominio	273/274/275/276/277/278
Relleno	149

S

Safari	015/093
--------	---------

T

Target	021
Texto oculto	202
Tim Berners-Lee	023
Tipos de dominios	268
Tipos de servidores	262/263
Type button	205

V

Variables definidas	213
Variables entre páginas	214
Vectores	214

W

W3C	022/023/024/071/077/082
Wordpress	017/026/179/303
Workflow	288

X

Xform	201
-------	-----

Y

Yahoo	014/015
YouTube	017/191

Z

Zonas calientes	021
-----------------	-----



CLAVES PARA COMPRAR UN LIBRO DE COMPUTACIÓN

1 SOBRE EL AUTOR Y LA EDITORIAL

Revise que haya un cuadro "sobre el autor", en el que se informe sobre su experiencia en el tema. En cuanto a la editorial, es conveniente que sea especializada en computación.

2 PRESTE ATENCIÓN AL DISEÑO

Compruebe que el libro tenga guías visuales, explicaciones paso a paso, recuadros con información adicional y gran cantidad de pantallas. Su lectura será más ágil y atractiva que la de un libro de puro texto.

3 COMPARE PRECIOS

Suele haber grandes diferencias de precio entre libros del mismo tema; si no tiene el valor en tapa, pregunte y compare.

4 ¿TIENE VALORES AGREGADOS?

Desde un sitio exclusivo en la Red, un Servicio de Atención al Lector, la posibilidad de leer el sumario en la Web para evaluar con tranquilidad la compra, y hasta la presencia de adecuados índices temáticos, todo suma al valor de un buen libro.

5 VERIFIQUE EL IDIOMA

No solo el del texto; también revise que las pantallas incluidas en el libro estén en el mismo idioma del programa que usted utiliza.



usershop.redusers.com

VISITE NUESTRO SITIO WEB

» Vea información más detallada sobre cada libro de este catálogo.

» Obtenga un capítulo gratuito para evaluar la posible compra de un ejemplar.

» Conozca qué opinaron otros lectores.

» Compre los libros sin moverse de su casa y con importantes descuentos.

» Publique su comentario sobre el libro que leyó.

» Manténgase informado acerca de las últimas novedades y los próximos lanzamientos.

**TAMBIÉN PUEDE CONSEGUIR NUESTROS LIBROS EN KIOSCOS O
PUESTOS DE PERIÓDICOS, LIBRERÍAS, CADENAS COMERCIALES,
SUPERMERCADOS Y CASAS DE COMPUTACIÓN.**



LLEGAMOS A TODO EL MUNDO VÍA  Y 

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

 usershop.redusers.com  usershop@redusers.com  + 54 (011) 4110-8700



Creación de sitios Web Versión 2.0

Esta obra está pensada para aprender a diseñar sitios web, sin tener conocimientos previos, a través de la última versión de la herramienta más difundida y profesional de la actualidad: Adobe CS5 Web Premium (Dreamweaver, Photoshop, Illustrator).

→ COLECCIÓN: MANUALES USERS
→ 320 páginas / 978-987-1773-99-2



Java

Este libro fue concebido para ayudar a quienes buscan aprender a programar en Java, como así también para aquellos que conocen el lenguaje, pero quieren profundizar sus conocimientos, a través de la aplicación de buenas prácticas y las mejores técnicas.

→ COLECCIÓN: MANUALES USERS
→ 320 páginas / 978-987-1773-97-8



C# Avanzado

Este libro abre nuevas perspectivas para la programación de aplicaciones con C#, al llevar al lector a recorrer temas más avanzados del lenguaje y framework .NET, con la ayuda del autor, un catedrático experto en el tema, autor de múltiples Manuales Users.

→ COLECCIÓN: MANUALES USERS
→ 320 páginas / 978-987-1773-96-1



Administrador de Redes Windows

En esta obra presentamos los fundamentos y las prácticas necesarias para la instalación y puesta en marcha de una red corporativa. El libro está centrado en WSBS 2011, un sistema operativo para servidores pensado para quienes necesitan llevar adelante soluciones de networking a bajo costo.

→ COLECCIÓN: MANUALES USERS
→ 320 páginas / 978-987-1773-80-0



HTML 5

En esta obra presentamos un nuevo paradigma de Internet que cambia de manera sustancial al diseño y desarrollo web. A lo largo de sus capítulos, aprenderemos sobre los estándares web que existen, y conoceremos las diferencias entre las versiones anteriores y la actual del lenguaje.

→ COLECCIÓN: MANUALES USERS
→ 320 páginas / 978-987-1773-79-4



Visual Basic

Este libro está escrito para aquellos usuarios que quieren aprender a programar en VB.NET. Desde el IDE de programación hasta el desarrollo de aplicaciones del mundo real en Visual Studio 2010, todo está contemplado para conocer en profundidad el lenguaje.

→ COLECCIÓN: MANUALES USERS
→ 352 páginas / 978-987-1773-57-2

¡Léalo antes Gratis!

En nuestro sitio, obtenga GRATIS un capítulo del libro de su elección antes de comprarlo.



Microcontroladores

Este manual es ideal para todos aquellos que quieran iniciarse en la programación de microcontroladores. A través de esta obra, podrán aprender sobre los microcontroladores PIC 16F y 18F, hasta llegar a conectar los dispositivos de forma inalámbrica, entre muchos otros proyectos.

→ COLECCIÓN: MANUALES USERS
→ 320 páginas / 978-987-1773-56-5



Programador .NET

Este libro está dirigido a todos aquellos que quieran iniciarse en el desarrollo bajo lenguajes Microsoft. A través de los capítulos del manual, aprenderemos sobre POO, la programación con tecnologías .NET y de qué manera se desenvuelven con otras tecnologías existentes.

→ COLECCIÓN: MANUALES USERS
→ 352 páginas / 978-987-1773-26-8



Photoshop: proyectos y secretos

En esta obra aprenderemos a utilizar Photoshop, desde la original mirada de la autora. Con el foco puesto en la comunicación visual, a lo largo del libro adquiriremos conocimientos teóricos y prácticos, para conocer los efectos y herramientas que ofrece el programa.

→ COLECCIÓN: MANUALES USERS
→ 320 páginas / 978-987-1773-25-1



WordPress

Este manual está dirigido a todos aquellos que quieran presentar sus contenidos o los de sus clientes a través de WordPress. En sus páginas el autor nos enseñará desde cómo llevar adelante la administración del blog hasta las posibilidades de interacción con las redes sociales.

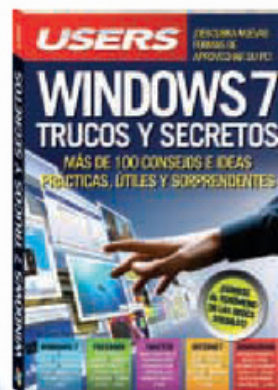
→ COLECCIÓN: MANUALES USERS
→ 352 páginas / 978-987-1773-18-3



Administrador de servidores

Este libro es la puerta de acceso para ingresar en el apasionante mundo de los servidores. Aprenderemos desde los primeros pasos sobre la instalación, configuración, seguridad y virtualización, hasta tener el control de los servidores en la palma de nuestras manos.

→ COLECCIÓN: MANUALES USERS
→ 352 páginas / ISBN 978-987-1773-19-0



Windows 7: Trucos y secretos

Este libro está dirigido a todos aquellos que quieran sacar el máximo provecho de Windows 7, las redes sociales y los dispositivos ultraportátiles del momento. A lo largo de sus páginas, el lector podrá adentrarse en estas tecnologías mediante trucos inéditos y consejos asombrosos.

→ COLECCIÓN: MANUALES USERS
→ 352 páginas / ISBN 978-987-1773-17-6



Desarrollo PHP + MySQL

Este libro presenta la fusión de dos de las herramientas más populares para el desarrollo de aplicaciones web de la actualidad: PHP y MySQL. En sus páginas, el autor nos enseñará las funciones del lenguaje para aplicar lo aprendido en nuestros propios desarrollos.

→ COLECCIÓN: MANUALES USERS
→ 432 páginas / ISBN 978-987-1773-16-9



Excel 2010

Este manual resulta ideal tanto para quienes se inician en el uso de Excel, como para los usuarios que quieran conocer las nuevas herramientas que ofrece la versión 2010. La autora nos enseñará desde cómo ingresar datos hasta la forma de imprimir ahorrando papel y tiempo.

→ COLECCIÓN: MANUALES USERS
→ 352 páginas / ISBN 978-987-1773-15-2



Técnico Hardware

Esta obra es una completa guía para aprender a llevar adelante un correcto diagnóstico y determinar la solución más adecuada para los problemas de hardware. En sus páginas veremos todas las técnicas necesarias para implementar las soluciones de los profesionales.

→ COLECCIÓN: MANUALES USERS
→ 320 páginas / ISBN 978-987-1773-14-5



PHP Avanzado

Este libro brinda todas las herramientas necesarias para acercar al trabajo diario del desarrollador los avances más importantes incorporados en PHP 6. En sus páginas, repasaremos todas las técnicas actuales para potenciar el desarrollo de sitios web.

→ COLECCIÓN: MANUALES USERS
→ 400 páginas / ISBN 978-987-1773-07-7



AutoCAD

Este manual nos presenta un recorrido exhaustivo por el programa más difundido en dibujo asistido por computadora a nivel mundial, en su versión 2010. En sus páginas, aprenderemos desde cómo trabajar con dibujos predeterminados hasta la realización de objetos 3D.

→ COLECCIÓN: MANUALES USERS
→ 384 páginas / ISBN 978-987-1773-06-0



Windows 7 Avanzado

Esta obra nos presenta un recorrido exhaustivo que nos permitirá acceder a un nuevo nivel de complejidad en el uso de Windows 7. Todas las herramientas son desarrolladas con el objetivo de acompañar al lector en el camino a ser un usuario experto.

→ COLECCIÓN: MANUALES USERS
→ 352 páginas / ISBN 978-987-1773-08-4



El contenido de este libro fue publicado en el curso en fascículos Web Design

CONTENIDO

1 | DISEÑO Y TECNOLOGÍA

Principios para un buen diseño / Estructura y diseño de la página

2 | HERRAMIENTAS DE TRABAJO

Adobe CS5 Web Premium / Photoshop, Illustrator y Dreamweaver

3 | XHTML Y CSS: BÁSICO

¿Qué es HTML? / Estructura / Etiquetas y atributos / CSS3

4 | XHTML Y CSS: TEXTO

Estilos CSS para texto / Párrafos y títulos

5 | ENLACES

Hipertextos / Enlaces relativos y avanzados

6 | MAQUETADO

Tableless / Posicionamiento de elementos /

7 | LISTAS

Definición / Ordenadas y no ordenadas / Personalización

8 | IMÁGENES

Optimización / Mapa de imágenes / CSS sprites

9 | FORMULARIOS

Elementos / Crear un formulario completo / PHP: fundamentos

10 | PROYECTO REAL I

Presentación del sitio / Layout / Menús / Footer

11 | PROYECTO REAL II

Las páginas de servicio / Portfolio / Página de contacto

12 | SERVIDORES Y TECNOLOGÍAS WEB

APÉNDICE A | VIDA LABORAL

CREACIÓN DE SITIOS WEB V2.0

Esta obra está pensada para aprender a diseñar sitios web sin tener conocimientos previos, a través de la última versión de la herramienta más difundida y profesional de la actualidad: Adobe CS5 Web Premium.

En cada uno de los capítulos, adquiriremos las herramientas y técnicas necesarias para crear un sitio, desde las bases mismas de los lenguajes clave: HTML y CSS. Además, aprenderemos tanto los fundamentos del diseño como las últimas tendencias en el armado de páginas. De esta manera es que contamos en nuestras manos con una guía práctica para crear sitios web con Dreamweaver, con la ayuda imprescindible de Illustrator y Photoshop.

Todos los procedimientos son expuestos de forma visual y práctica, con fotografías a todo color, tablas, infografías 3D y la teoría necesaria para comprender en profundidad cada tema presentado.

Al finalizar la obra, entenderemos todos los aspectos de la creación de sitios web, y con el desarrollo de un caso real, podremos llevarlos a la práctica. Un libro sin desperdicio, que nos permitirá ampliar nuestro horizonte laboral, al mismo tiempo que nos ayudará a actualizarnos con lo último en tecnología web.



RedUSERS.com

En este sitio 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

NIVEL

EXPERTO

AVANZADO

INTERMEDIO

PRINCIPIANTE

WEBSITES DEVELOPMENT 2.0

This manual is a complete guide to learn how to make a website under XHTML and CSS. The reader is going to learn fundamental aspects of design and will get into the last trends in web site development to become a professional web designer.



ISBN 978-987-1773-99-2



9 789871 773992 >