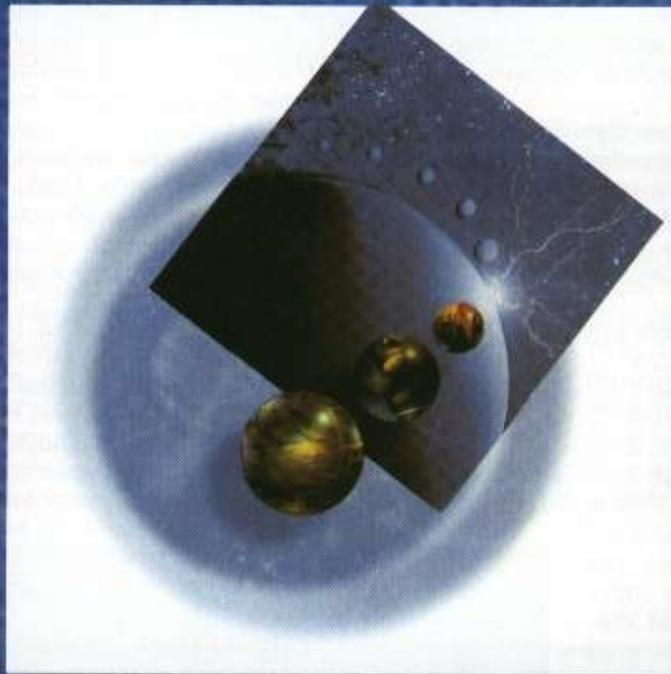


CURSO IBM

PROGRAMAR

es fácil



4

IBM **WebSphere
Studio**

Programación en Internet

Multimedia Ediciones, S.A.

Conceptos de programación (4)

Las palabras del lenguaje



Al programar, como ya sabemos, usaremos un lenguaje determinado y muy concreto, adecuado a las necesidades y limitaciones del ordenador. Vamos a realizar un examen cuidadoso del típico juego de instrucciones disponibles en cualquier lenguaje y de cómo nos ayudan a resolver los problemas de la programación.

Como ya mencionamos anteriormente, en esta sección del curso hemos de prestar atención al cómo de la programación; por ello, las instrucciones y los ejemplos que se desarrollarán a continuación irán expresadas en pseudocódigo, concepto que ya hemos repasado y que, como sabemos, no es "código de verdad", ejecutable, aunque nos permite diseñar y planificar cómo vamos a crear ese auténtico código. Por su interés actual, vamos a presentar los conceptos en una forma

que resultará especialmente aplicable a los lenguajes C, C++ y Java.

LAS VARIABLES

El concepto de variable es muy sencillo, y a la vez resulta absolutamente fundamental a la hora de programar. Una variable no es más que un segmento de la memoria del ordenador en el cual podemos guardar un valor.

Existen diversos tipos de variables, y cada lenguaje emplea una aproximación propia a este concepto. En este

momento, lo que más nos importa para nuestros fines es saber que las variables tienen un tipo, y que deben ser declaradas y creadas para poder utilizarlas. Recordaremos de las temibles clases de matemáticas del "cole" que los números eran de diversos tipos: enteros, naturales, racionales, etc. Con las variables ocurre lo mismo: tenemos variables de tipo entero (que pueden, como indica su nombre, guardar números enteros), de coma flotante (para guardar números con decimales), etc. Los detalles sobre qué tipos existen y para qué sirven varían de un lenguaje a otro, por lo que deberemos estudiar cada caso en concreto. Por ejemplo, en la mayoría de ellos existen, entre otros, los tipos int (variables enteras) y float (coma flotante). Los tamaños de estas variables, así como los valores máximo o mínimo que pueden almacenar, dependen del lenguaje (y a veces del compilador o del intérprete utilizado). Pero no sólo de números viven los programas. Otros tipos de variables muy usados son el carácter (una letra o símbolo único) y la cadena de caracteres; ésta consta de una serie de caracteres de texto, generalmente encerrados entre comillas, corchetes, etc. (también dependiendo del lenguaje). Por ejemplo:

"Hola, esto es una cadena"



FIG 1. Una variable es un segmento de la memoria en el que se almacena un dato determinado que es usado en un programa.



FIG 2. Igual que los antiguos magos invocaban a los espíritus por su nombre, para crear una variable, lo primero que hemos de hacer es darle un nombre.

Nuevamente, los detalles exactos sobre las cadenas y su utilización dependen del lenguaje. Además de estos tipos de datos básicos, algunos lenguajes permiten crear otros tipos de datos según las necesidades del programador. A estos tipos se les conoce como tipos de datos compuestos o de usuario, y constituyen por sí mismos todo un mundo de posibilidades. En el caso de Java, por ejemplo, disponemos de una serie de datos compuestos, como las estructuras, uniones y las vitales clases. Volveremos sobre estos conceptos más en detalle, ya que son de una importancia capital.

Antes de poder usar una variable hemos de crearla; al igual que los magos antiguos, según decíamos, invocaban a los espíritus por su nombre, para crear una variable lo primero que hemos de hacer es darle un nombre. Las reglas que rigen los nombres válidos de una variable son, una vez más, dependientes del lenguaje concreto con el que estemos trabajando. Así, en el caso de Java, C y C++ los nombres de variable pueden ser una colección de letras (no valen los acentos), números y algunos símbolos especiales (como el guión bajo "_"). El nombre, que no puede empezar por un dígito,

```
//Hello.java
import java.awt.*;
import java.applet.*;

public class Hello extends Applet
{
    public void paint(Graphics g)
    {
        for (int x = 0 ; x < size().height ; x++)
        {
            g.drawString("Hello World!", 10, x*15);
        }
    }
}
```

FIG 3. Declarar una variable permite al compilador asignarle un espacio en la memoria.

puede tener la longitud que se quiera (aunque los compiladores suelen limitar los nombres a un tamaño máximo). Las mayúsculas y las minúsculas se consideran diferentes, por lo que los nombres "nombre1" y "Nombre1" son dos variables distintas (mucho cuidado al escribir, ya que esta es otra fuente de errores comunes).

DECLARÁNDOSE...

En algunos lenguajes (como Basic) basta con utilizar un nombre de variable para que el sistema la cree automáticamente; en otros lenguajes, como Java, es obligatorio declarar la variable antes de poder utilizarla por vez primera (no hacerlo supondrá un error al compilar o ejecutar el programa). Declarar una variable es escribir una orden en la que informamos al ordenador de que existe una variable, indicamos de qué tipo es y cuál es su nombre, junto con cualquier otra información que pueda ser necesaria (por ejemplo, el tipo de almacenamiento de la variable, si se guarda siempre o sólo durante un tiempo, si puede ser modificada desde una parte del programa y desde otra no, etc.).

Declarar una variable permite al compilador asignarle un espacio en la memoria, hacer optimizaciones o crear las instrucciones necesarias para acceder a la misma de una forma muy eficaz. Pero informar al compilador de que existe una variable no es suficiente: hay que crearla, esto es, hacer la reserva de una porción de memoria para poder guardar el valor en ella.

En ciertos casos, cuando una variable única es compartida por varios programas, cada programa por separado ha de saber que existe dicha variable, aunque no hay que reservar espacio para ella en todos ellos, ya que algún programa ya lo ha creado. Lo que se hace en este caso es que la variable se declara en cada programa (el compilador sabe de su existencia) y se crea sólo en uno de ellos. La diferencia entre creación y declaración puede parecer sutil, pero es de vital importancia especialmente en el campo de la programación orientada a objetos, donde hasta se distingue con un nombre especial la declaración de la *instanciación* o creación de una varia-

CON COMENTARIOS...

Uno de los elementos a considerar a la hora de confeccionar un programa son los comentarios introducidos por el programador durante la creación del mismo. Los comentarios que no son instrucciones de programa, son ignorados por el ordenador y no tienen ningún impacto sobre el programa (salvo, tal vez, aumentar algo el tamaño del código fuente). Son simples notas introducidas que explican, por ejemplo, lo que hace cada línea, el fundamento del algoritmo en el que se basa el software, notas explican que los cambios hechos sobre el programa original, quién los hizo y a qué se debieron, etc. Todos estos detalles son fundamentales en una buena programación, ya que permiten entender qué hace el programa y cómo funciona. Con frecuencia, en programas profesionales las líneas de comentario suelen ser mayores y más numerosas que las del código propiamente dicho, lo que permite apreciar la importancia de este apartado. La forma de escribir comentarios varía de un lenguaje a otro. Por ejemplo, en Java se interpreta como comentario (esto es, el compilador ignora) las líneas que comiencen con dos barras seguidas, o bien todo el texto que hay desde dos barras seguidas hasta el final de la línea:

```
//Esto es un comentario
Circulo = ValorDePI * 2 *
Radio; // Calcula longitud de
circunferencia
```

Se admite también otro tipo de formatos para los comentarios en Java, al igual que en C y C++; como es el delimitado por las combinaciones `/* */` y `/** */`. Estos comentarios

pueden ser de más de una línea ya que se interpretará como comentario desde la aparición del primer símbolo hasta el segundo. Por ejemplo, todas las formas siguientes de comentario son correctas:

```
/* Este es un comentario
de varias líneas */
Circulo = ValorDePI * 2 * Radio; /* Pi por dos
por R*/
Circulo = /* Pi por dos por R*/ ValorDePI * 2 *
Radio;
```

Con frecuencia se prescinde de unos buenos comentarios con el afán de escribir el código más rápidamente; al fin y al cabo, si lo escribimos nosotros, ya sabemos cómo funciona y siempre podremos añadirlos más tarde, con todo terminado. ¿Cierto? No, falso. La memoria humana no puede retener los miles de detalles involucrados en la creación de un programa de una mínima complejidad, y es casi seguro que no recordaremos algunos detalles que pueden ser cruciales cuando, más tarde, vayamos a comentarlos. Además, si transcurren semanas o meses desde que escribimos el código, seguro que casi ni recordamos cómo funcionaba en un principio, lo que supondrá un esfuerzo adicional para comprenderlo. En definitiva, sobre todo si los programas son complejos y van a ser reutilizados y modificados a lo largo del tiempo, es crucial que incluyan unos comentarios cuidadosos y tan detallados como sea posible.

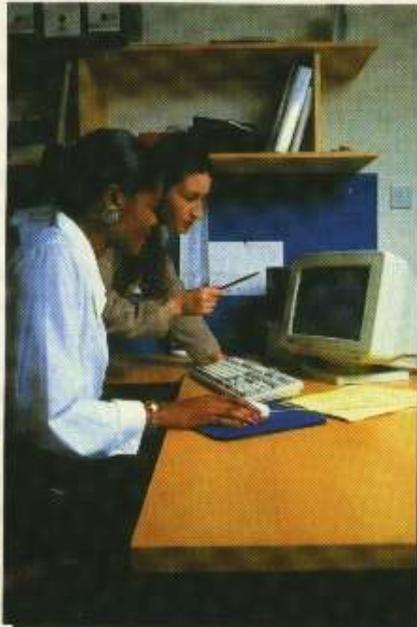


FIG 4. Comentar el código es fundamental para poder usar, mantener o ampliar programas, tanto más cuanto más grandes y complejos sean.

ble. Por usar un ejemplo sencillo, declarar una variable es como dibujar el plano de una casa: cualquiera que lo vea sabrá cómo es esa casa, la disposición de las habitaciones, superficies, etc. Pero es difícil irse a vivir a un plano de papel, sobre todo cuando llueve: hace falta una casa de verdad. Construir esa casa (eso sí, siguiendo el plano original) es instanciar o crear esa casa. Lo bueno es que podemos crear más casas a partir de un único plano, esto es, una variable puede ser

instanciada (creada) cuantas veces queramos, y guardar en ella valores diferentes aunque siempre de un mismo tipo. Volveremos sobre este particular más adelante, ya que, aunque parezca intrascendente, el concepto es fundamental para la teoría de programación orientada a objetos.

PRINCIPIO Y FIN

Un buen punto de partida es el principio. En efecto, muchos lenguajes de programación disponen de instruccio-

nes que determinan cuál es la primera línea del programa, esto es, desde dónde empieza a ejecutarse. Igualmente, suele existir una instrucción que provoca el fin del programa (y el retorno del control al sistema operativo). Llamaremos a estas instrucciones BEGIN y END, respectivamente (los usuarios con algo de experiencia en Pascal sabrán reconocer inmediatamente estas instrucciones). Así, un programa típico dispondrá de una secuencia de inicio y de una o varias secuencias de fin:

BEGIN

...

instrucciones del programa

...

END

Fijémonos en que sólo existirá una instrucción de inicio, ya que el programa comienza en un único sitio, pero que el mismo programa puede terminar por varias causas y, por tanto, tener varias secuencias de fin. Los programas en estos lenguajes empiezan ejecutando la función main (volveremos más adelante sobre el concepto de función), y terminan cuando ésta acaba o bien el programa encuentra una instrucción exit en un punto cualquiera. En general, un programa consistirá en una serie de instrucciones individuales, que se ejecutan en el orden en que

que dan como resultado algo concreto en términos de funcionalidad del programa (por ejemplo, guardan un valor en una variable, imprimen un resultado, etc.). Así, una serie de sentencias de programa podrían ser algo así como:

Imprimir "Hola, pulse una tecla";
Esperar una tecla;
Borrar pantalla;
 ...
Otras instrucciones del programa;

El olvidar la correcta escritura de los terminadores es la causa más común de errores durante el desarrollo de un programa.

Por fortuna, estos errores suelen ser detectados por el intérprete o compilador; algunos de estos incluso nos advierten de ello y sugieren dónde habría que situar el terminador que falta. Esto se debe a que es muy fácil leer una línea de programa y no prestar atención a la existencia de los terminadores; los seres humanos tenemos una gran habilidad para leer lo que queremos leer y no lo que está escrito en realidad...

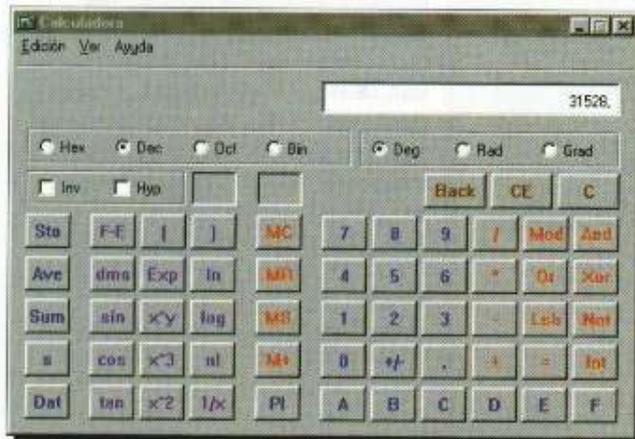


FIG 5. Los operadores más comunes son los aritméticos y matemáticos, aunque existen otros como, por ejemplo, los lógicos, los de relación, etc.

están escritas (o en el determinado por la lógica del programa, como veremos más adelante). Para que el ordenador sepa cuándo termina una instrucción y, así, poder ejecutarla (o, en el caso de un compilador, dejar de analizarla y generar el comando máquina que la ejecuta en la práctica), al final de cada sentencia, se pone lo que se conoce como un terminador, esto es, un símbolo que indica que la instrucción acaba ahí. Este símbolo es el "punto y coma" (";") en la mayoría de los lenguajes. Una instrucción terminada en punto y coma se convierte en lo que conocemos como una sentencia, es decir, una secuencia de instrucciones

APRENDER A EXPRESARSE

Una expresión es una secuencia de instrucciones que dan como resultado un valor (que puede ser numérico o de cualquier otro tipo). Las expresiones más sencillas son las constantes, esto es, un determinado valor; por ejemplo:

1
3.14159
"Hola"

Otra expresión sencilla se forma con el nombre de una variable; el resultado es el valor almacenado dentro de dicha variable, y tendrá el tipo de la misma (por lo que se le aplicarán las

restricciones y condiciones particulares para ese tipo):

Variable1
ValorDePI
x

Estas expresiones sencillas se pueden combinar entre sí mediante operadores. Los operadores más frecuentes son los aritméticos: suma (+), resta (-), multiplicación (*) y división (/). Algunos lenguajes incorporan otros, como el operador *módulo* (el resto de la división entre dos números enteros, y que se representa por el símbolo %) o los que permiten calcular funciones matemáticas (como funciones trigonométricas, logaritmos o exponenciación, etc.). Así, por ejemplo, expresiones compuestas pueden ser las siguientes:

x + 1
ValorDePI * 2 * Radio

Se pueden hacer varias cosas con el resultado de una expresión. Lo más obvio es, por ejemplo, asignarle el valor calculado a una variable; la asignación de valores a una variable se realiza de diversas maneras según el lenguaje. La forma más frecuente es emplear el operador de asignación "igual" (=):

Circulo = ValorDePI * 2 * Radio;

Se calcula el valor de ValorDePI * 2 * Radio y se guarda en resultado en la variable Circulo. Nótese el punto y coma final, que transforma todo esto en una sentencia.

Por último, otra necesidad frecuente es la de agrupar sentencias para formar un bloque de sentencias y/o expresiones.

Un bloque no es más que un conjunto de instrucciones que se ejecutan solidariamente, como si fuesen una, en el seno del programa (por ejemplo, mostrar un mensaje en una ventana, esperar la pulsación de una tecla y cerrar dicha ventana). En Java y los lenguajes C y C++, se emplean las llaves "{" y "}" para encerrar las instrucciones que forman un bloque.

Soluciones prácticas (4)

AAScripTer: Código de JavaScript fácil

Muchos usuarios y empresas se han lanzado de cabeza a la Red, dándose a conocer gracias a sus páginas Web. Pero para que una página brille con luz propia, se hace necesario ir más allá del código HTML; hay que entrar de lleno en la programación de código JavaScript.

En efecto, JavaScript es un lenguaje para Internet que está causando furor gracias a su potencia y a su relativa facilidad de programación. No obstante, no podemos olvidar que JavaScript es un lenguaje y que, como tal, es necesario poseer ciertos conocimientos en el momento de embarcarse en su programación.

Si no desea aprender JavaScript o de momento no tiene tiempo para

meterse de lleno en el apasionante mundo de la programación en este lenguaje, existen herramientas que le podrán ayudar.

Se trata de programas que contienen muchas utilidades creadas en código de JavaScript. Estas herramientas son exactamente lo que parecen: permiten acceder a código utilizable en una página Web sin apenas la intervención del usuario. De esta forma, es

posible dejar un poco de lado la, en ocasiones, tediosa y lenta tarea de programar código para centrarse en otros temas relacionados con el diseño de una página Web. El código generado es freeware, por lo que se puede usar sin problemas.

En estas líneas vamos a presentar AAScripTer, una de estas aplicaciones, y vamos a ver la forma de obtener el máximo partido de ella.

AASCRIPter

La primera vez que ejecute AAScripTer se asombrará ante la sencillez de su interfaz. Es simplemente una pequeña ventana que muestra unos cuantos botones y un cuadro de lista desplegable. No obstante, no deje que su espartano aspecto le llame a engaño. Ahorrará mucho tiempo usando el código que recoge el programa.

Su forma de operar es muy sencilla. Para empezar, escoja una de las categorías del cuadro de lista desplegable; se incluyen las siguientes:

Status Bar Effects (efectos para la barra de estado), **Text Effects** (efectos de texto), **Image and Sound Effects** (efectos de imagen y sonido), **Links Effects** (efectos de enlaces), **Mouse Cursors** (cursores de ratón), **Random Effects** (efectos aleatorios), **Forms and Form Related** (formularios y relacionados con los formularios), **Math & Arithmetic Related** (matemá-

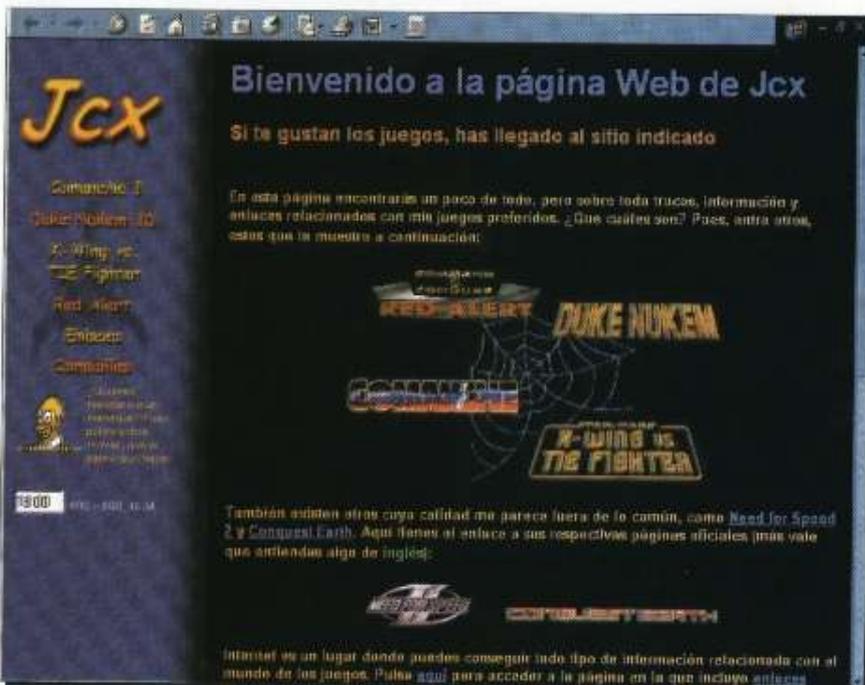


FIG. 1 El diseño de una página Web realmente atractiva exige algo más que conocer con una cierta profundidad el lenguaje HTML.

FREWARE

La herramienta **AAScripter** es, como podrá comprobar a lo largo de estas líneas, una aplicación que destaca por su utilidad y su sencillez de manejo. Sin embargo, **AAScripter** también llama poderosamente la atención en otro apartado: el código generado con ella se puede utilizar como si usted mismo lo hubiese tecleado, sin pagar nada por ello. Si desea obtener más información o estar al tanto de la aparición de nuevas versiones de esta utilidad, no dude en visitar la página Web del autor, **All Almassawi**, en <http://www.cyberiapc.com/aa/index.htm>.

Para instalar el programa, bastará con que ejecute el archivo **AAScripter.exe** que se encuentra en la carpeta **Utilidades** del CD-ROM que acompaña a esta entrega.



FIG. 2 Visite la página Web de All Almassawi, en <http://www.cyberiapc.com/aa/index.htm>, para comprobar si hay una nueva versión de **AAScripter**.

tics y relacionados con la aritmética), **Clocks and Dates** (relojes y fechas), **Security** (seguridad), **Search Engines** (motores de búsqueda), **Browser Window** (ventana de exploración), **Menus & Navigation Systems** (sistemas de menús y navegación), **Redirection** (redireccionamiento), **Dis-**

play User Information (mostrar la información del usuario) y **Other** (otros). Como ve, dispone de numerosos tipos de categorías y, dentro de cada una de ellas, de varias utilidades.

En la creación de este ejemplo vamos a escoger la categoría **Image and Sound Effects**; se mostrarán 31 utili-

dades programadas en código de JavaScript. De entre ellas, y siempre a modo de ejemplo (recuerde que puede cambiar sus elecciones en cualquier momento sin que este texto deje de tener sentido, ya que las operaciones se explican tanto en general como en particular), vamos a escoger el llamado **Drop Down Image Selector 2** (se podría traducir como "cuadro de lista desplegable de selección de imágenes"). Para ello basta con hacer doble clic sobre su nombre.

En la pantalla se mostrará ahora una nueva ventana de la aplicación. En la parte superior encontrará una explicación de qué se consigue hacer con el código escogido. Podrá ampliar la información usando el botón **Click here for more instructions**. En definitiva, con este código se puede generar un cuadro de lista desplegable capaz de mostrar imágenes según la elección que realice el usuario de entre las opciones del cuadro; además, haciendo clic sobre las imágenes se podrán visitar otras páginas Web.

Seguramente ya se habrá fijado en la sección **Cut-and-Paste Code** (código para cortar y pegar). Aquí se encuentra el código que debe usar en su página Web. En el caso de esta utilidad de JavaScript, deberá usar dos fragmentos de código en su página Web: uno en la cabecera y otro en el cuerpo del archivo HTML.

PÁGINA WEB

Para poder insertar los dos fragmentos del código que nos está proporcionando **AAScripter** en este momento es necesario crear una página Web, que en este ejemplo será extremadamente sencilla. Para programar en HTML no se necesitan programas avanzados de edición **WYSIWYG** (*What You See Is What You Get*, lo que ve es lo que obtiene), aunque estos ahorran una enorme cantidad de tiempo y esfuerzo, como estará viendo en el curso dedicado a **IBM WebSphere Studio**. No obstante, y dado que no necesitamos más que se muestre el resultado del código de JavaScript, vamos a crear la página Web con la ayuda del **Bloc de notas**.

Abra el **Bloc de notas** y escriba, separados por una línea en blanco, los siguientes comandos:

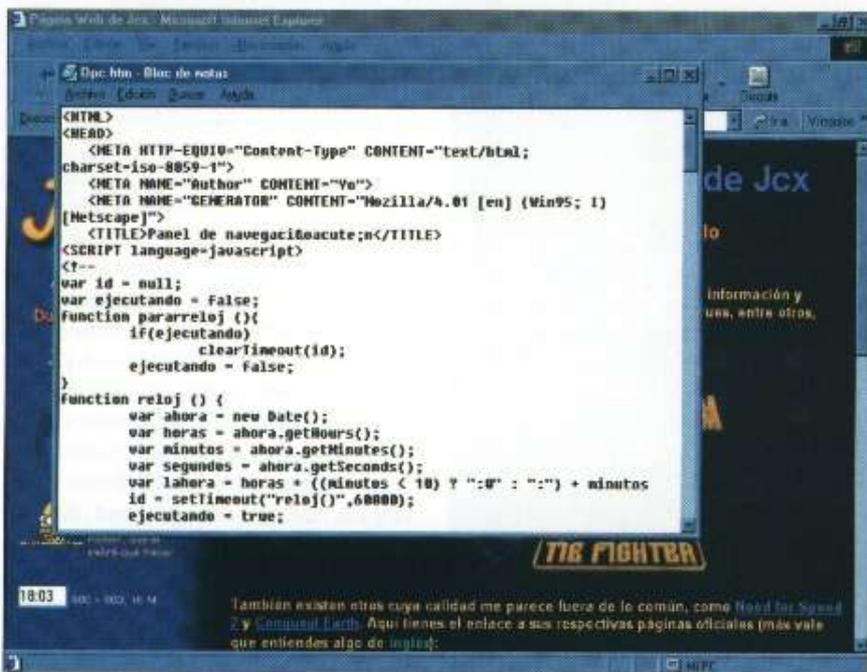


FIG. 3 Si no dispone de tiempo para aprender a programar código de JavaScript, use el código que incluye **AAScripter**.

```
<HEAD>
</HEAD>
<BODY>
</BODY>
```

Las etiquetas <HEAD> (apertura de sección) y </HEAD> (cierre de sección) delimitan la cabecera de la página. <BODY> y </BODY> indican que el código que se encuentra entre ellas se refiere al cuerpo de la página. Ahora comprenderá a qué se refiere AAScripTer cuando muestra un mensaje especificando que una parte del código se debe colocar entre <HEAD> y </HEAD> y otro entre <BODY> y </BODY>.

Ponga de nuevo en primer plano el programa AAScripTer, pulse dentro del cuadro de texto que contiene el primer fragmento de código, asegúrese de que

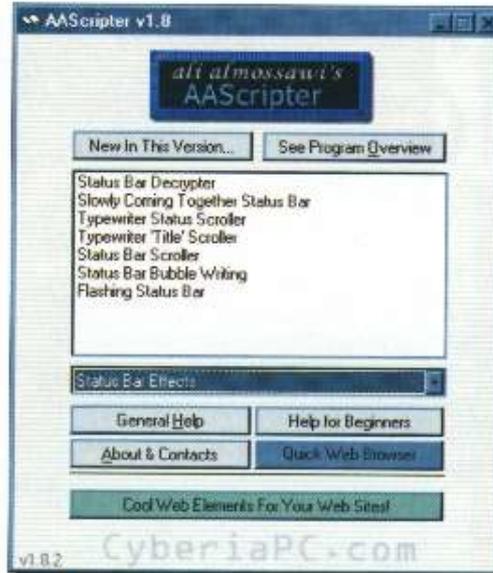


FIG. 4 La interfaz de la aplicación es realmente sencilla, pero no se deje engañar por su aparente austeridad.

el cursor está delante de la primera palabra del código (pulse, por ejemplo, la combinación de teclas **Control + Inicio**), seleccione todo el texto hasta el final (con **Control + Mayús + Fin**), copie el contenido al Portapapeles de Windows (con **Control + C**), muestre en primer plano la ventana del Bloc de notas, sitúe el punto de inserción de texto entre las etiquetas <HEAD> y </HEAD>, y pegue el contenido del Portapapeles (con **Control + V**).

Si examina el código, aunque no conozca el lenguaje, descubrirá que existen tres datos que debe cambiar, ya que no pertenecen al lenguaje de programación, sino que son constantes que ha definido el autor del código. Debe editar esta parte:

CÓDIGO PARA LA CABECERA

A continuación se puede ver el código generado para la cabecera de nuestra página Web.

```
<script language="javascript">
<!--

/*Combo Box Image Selector:
By Website Abstraction (www.wsabstract.com)
Over 200+ free JavaScript here!
*/

function linkrotate(which){
var mylinks=new Array()
//add in more links if you want (ie:mylinks[3]=...)
mylinks[0]="http://www.abcnews.com"
mylinks[1]="http://www.cnn.com"
mylinks[2]="http://www.geocities.com"

window.location = mylinks[which]
}

function showimage()
{
if (!document.images)
return
document.images.pictures.src=
document.mygallery.picture.options
[document.mygallery.picture.selectedIndex].value
}
-->
</script>
```

```
mylinks[0]="http://
www.abcnews.com"
mylinks[1]="http://
www.cnn.com"
mylinks[2]="http://
www.geocities.com"
```

Concretamente, debe cambiar los enlaces. Si lo desea, puede ampliar o reducir el número de enlaces; en su caso pueden existir tan sólo dos enlaces, o cuatro, o los que estime oportunos. En el ejemplo vamos a cambiarlos por enlaces a las páginas *playa.htm*, *montana.htm* y *ciudad.htm*, de esta forma:

```
mylinks[0]="playa.htm"
mylinks[1]="montana.htm"
mylinks[2]="ciudad.htm"
```

Fíjese en que en el código se especifica que van a existir tres páginas Web en la misma carpeta donde resida la página Web en la que estamos insertando el código de JavaScript. Todos los archivos HTML finales de ejemplo, así como las imágenes que definiremos más adelante, se encuentran en la carpeta **UtilidadesJavaScript** del CD-ROM que acompaña a esta unidad.

Pegue ahora la segunda porción de código de AAScripTer entre las etiquetas <BODY> y </BODY>, tal y como se ha explicado para el caso del primer fragmento. No olvide guardar la página

na Web en su disco duro. Si se fija en este nuevo código, los parámetros que ha definido el creador del mismo y que no forman parte del lenguaje son cuatro líneas concretas:

```
<option selected
value="me.gif">Picture of
me</option>
<option
value="myaunt.gif">Picture
of my aunt</option>
<option
value="brother.gif">Picture
of my brother</option>
onmouseover="window.
status="";return true">Fotografía
de una playa</option>
<option
value="montana.jpg">Fotografía
de una montaña</option>
<option
value="ciudad.jpg">Fotografía
de una ciudad</option>
onmouseover="window.
status="";return true">
<tr>
<td width="100%"><form name="mygallery"><p><select
name="picture" size="1" onChange="showimage()">
<option selected value="me.gif">Picture of me</option>
<option value="myaunt.gif">Picture of my aunt</option>
<option value="brother.gif">Picture of my brother</option>
</select></p>
</form>
</td>
</tr>
<tr>
<td width="100%"><p align="center"><a
href="javascript:linkrotate(document.mygallery.
picture.selectedIndex)" onmouseover="window.status="";return
true"></a></td>
</tr>
</table>
```

Web. Use AAScripter a discreción y aproveche el tiempo que ha ahorrado. Por último, nos permitimos recordarle una vez más, que en la carpeta \Utilidades\JavaScript del CD-ROM, encontrará el ejemplo completo así como los archivos complementarios.



FIG. 6 Aunque en la pantalla principal existe una corta descripción de la función del código, puede hacer clic sobre "Click here for more instructions" para obtener más información.

IBM WebSphere Studio (4)

Formularios en IBM WebSphere Studio

Continuamos con la creación del sitio Web de ejemplo "El Espacio" con IBM WebSphere Studio. En esta ocasión vamos a abordar la creación de una nueva página del sitio en la que, además de encontrar enlaces a otras páginas relacionadas con el mismo tema, el visitante pueda enviarnos todo tipo de información gracias a los formularios.

La capacidad de los visitantes de ponerse en contacto con los creadores de un sitio Web es una de las características más interesantes de la World Wide Web. A la hora de crear un sitio Web hemos de tener en cuenta que, además de proporcionar la información requerida a los usuarios, en multitud de ocasiones necesitaremos que sean ellos quienes nos la envíen a nosotros, ya sea en forma de encuestas, opiniones o simplemente preguntas.

Por supuesto, ya hemos visto en anteriores unidades la manera adecuada para crear un enlace a una dirección de correo electrónico mediante el cual se podría llevar a cabo esta comunicación, aunque esto puede provocar algunos problemas de forma, ya que el usuario del que necesitamos la información puede no enviar toda la requerida al tener que redactarla por sí mismo, o simplemente puede no saber cómo hacerlo.

Frente a esto, el uso de formularios en nuestras páginas nos permite organizar y clasificar la información que el visitante deba introducir, pidiéndole los datos necesarios de manera separada, como su nombre y apellidos, correo electrónico, teléfono, etc. A la vez, los formularios permitirán que la información nos llegue de una manera muy esquematizada y organizada, facilitando la automatización del tratamiento de los datos. Como puede ver, las ventajas frente al envío de correo electrónico directo por parte del visitante son evidentes.

Llamamos formulario a toda página Web o parte de ella que contenga alguno de los siguientes objetos:

- Campos y áreas de texto y de contraseña.
- Grupos de opciones.
- Casillas de verificación.
- Listas desplegables y de selección.
- Cuadros de diálogo.
- Botones de acción.
- Campos ocultos.

Además de, por lo general, dos botones: uno para enviar la información y otro para restaurar los campos a sus valores originales en caso de que el usuario desee comenzar de nuevo a escribirlos. De esta manera, al hacer clic sobre el botón de envío los datos nos serán enviados de forma ordenada, reenviándonos a una dirección de correo electrónico, aunque lo más co-

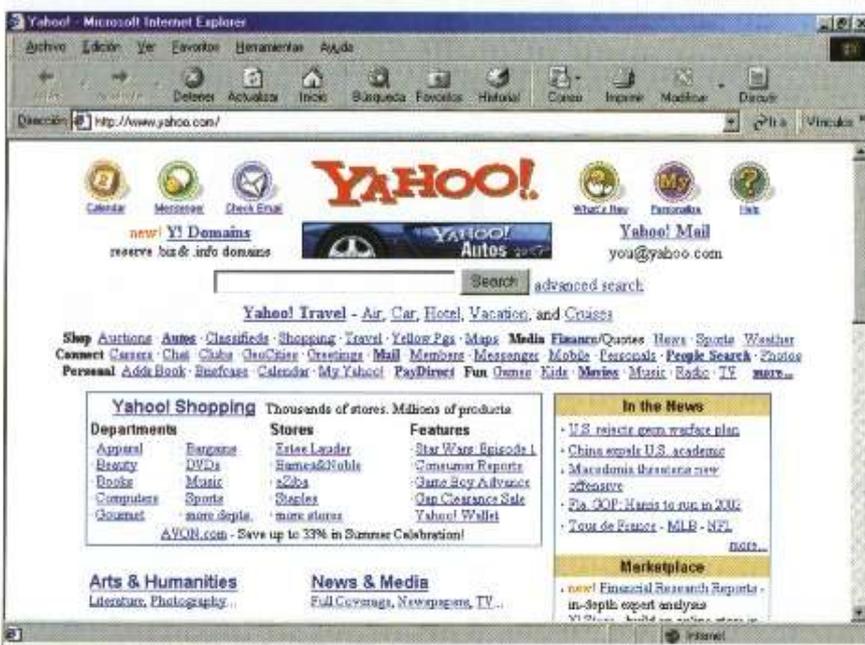


FIG. 1 Con formularios se pueden crear motores de búsqueda avanzada al llamar a programas CGI.



FIG. 2 El primer paso será crear una página en blanco llamada enlaces .html.

mún es que antes se realice un procesamiento previo por parte de un programa residente en el servidor.

Dichos programas reciben el nombre de CGI, esto es, *Common Gateway Interface*, y están escritos en cualquier tipo de lenguaje, siendo su única limitación el que sean soportados por el servidor que los contiene, ya sean estos lenguajes compilados o interpretados. La característica principal de un CGI es que éste se ejecuta directamente en el servidor Web, nunca en el ordenador del visitante de la página que lo llama, al contrario que JavaScript, por ejemplo, que se ejecuta en la máquina local del visitante.

A la vez, no es posible que dicho visitante interactúe con el CGI, al tra-

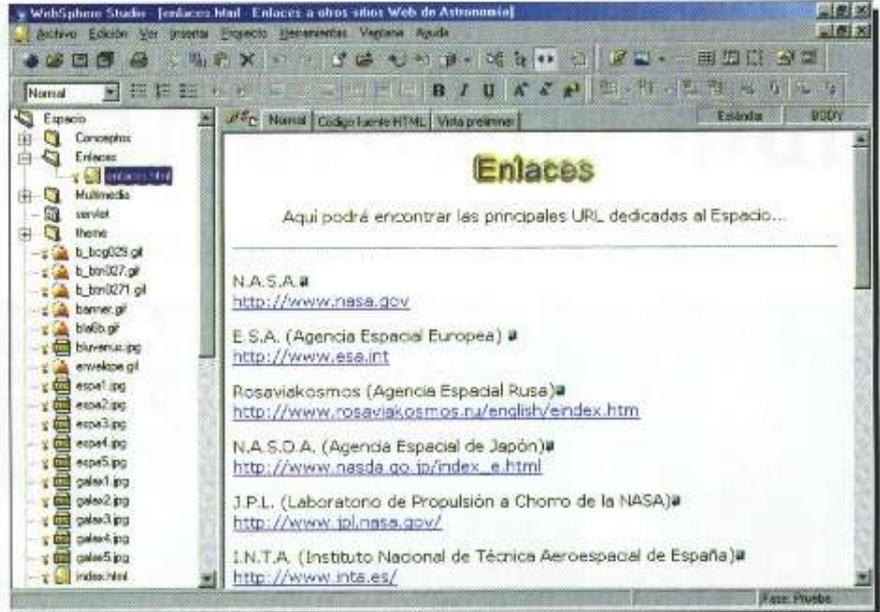


FIG. 3 Introduciremos los enlaces bajo una breve descripción de cada uno de ellos.

tarse de una comunicación entre el código de la página y el CGI residente en el servidor. Los lenguajes más usados para su desarrollo son Perl y C. Perl es un lenguaje interpretado, muy usado en la creación de CGI por las facilidades que aporta en cuanto a tratamiento de textos, mientras que C y C++ también están muy extendidos al ser más potentes que Perl, aunque su mayor desventaja es la necesidad de compilarlos para que puedan ser

usados, además de su complejidad. Los CGI, además, no sólo sirven para procesar datos de formularios, sino que también pueden manejar bases de datos, con lo que se pueden crear potentes motores de búsqueda desde nuestra propia página Web.

En este momento la programación de CGI se aleja de nuestros objetivos; nos basta con saber que los datos de los formularios que incluimos en nuestras páginas se envían a uno de estos programas que, además, es en la mayoría de las ocasiones proporcionado por el propio proveedor de servicios de hospedaje de la página. Para nuestro ejemplo nos bastará con enviar el resultado del formulario a una dirección de correo electrónico mediante **mailto**, con lo que dispondremos de una serie de mensajes con la información requerida al visitante cuyo tratamiento podremos automatizar con cualquiera de los lenguajes que se verán a lo largo del curso.

FORMULARIOS EN HTML

La principal etiqueta en lenguaje HTML para crear un formulario es **<FORM>**, siendo ésta la encargada de definir su comienzo. Dentro de esta etiqueta se puede insertar código para los campos del formulario y otros objetos diversos (como texto, imágenes, enlaces, etc.) para finalmente acabar con la etiqueta **</FORM>**. Además, dentro de **<FORM>** es necesario definir una serie de parámetros:

■ **ACTION:** Sirve para especificar la URL donde se entregarán los datos del formulario, generalmente un programa CGI, aunque es posible enviar directamente datos a una dirección de correo electrónico.

■ **METHOD:** En este caso se puede elegir entre GET y POST. GET dirige los campos del formulario añadiendo el símbolo "?" como separador, mientras que POST se emplea cuando se quieren enviar los formularios por correo o se adjuntan archivos con éste.

■ **ENCTYPE:** Indica la forma en que se codificarán los datos a la hora de enviarlos. Para texto son válidos los tipos "application/x-www-form-urlencoded" y "text/plain", pero en caso de querer añadir archivos adjuntos es necesario usar el tipo "multipart/form-data".

UNA PÁGINA DE ENLACES

Antes de comenzar a crear nuestro formulario debemos preparar la página en la que estará contenido. En nuestro caso, y siguiendo con el ejemplo de "El Espacio", completaremos una página de enlaces a otros sitios en la que añadiremos nuestro formulario como medio de que los visitantes puedan

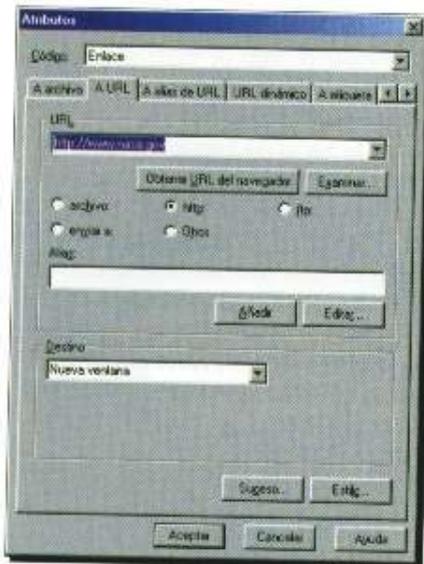


FIG. 4 Desde la ventana atributos asignaremos la URL de destino del enlace.

hacemos llegar las direcciones de sitios que consideren interesantes.

Abra, pues, IBM WebSphere Studio y, dentro de la carpeta **Enlaces**, abra el archivo **enlaces.html**. Como siempre, estará en blanco, por lo que, una vez más, borre el texto seleccionado y cambie el título (recuerde que esto se realizaba haciendo clic en **Atributos de BODY**) por el de **Enlaces a otros sitios Web de Astronomía**. Añada un logotipo manteniendo el formato usado en an-

teriores unidades y con un tamaño de letra 20; baje un poco el cursor con un retomo de carro. Escriba *Aquí podrá encontrar las principales URL dedicadas al Espacio...* y centre el texto. Inserte una línea horizontal y un retomo de carro, e introduzca el siguiente texto:

N.A.S.A.
<http://www.nasa.gov>
E.S.A. (Agencia Espacial Europea)
<http://www.esa.int>
Rosaviakosmos (Agencia Espacial Rusa)
<http://www.rosaviakosmos.ru/english/eindex.htm>
N.A.S.D.A. (Agencia Espacial de Japón)
http://www.nasda.go.jp/index_e.html
J.P.L. (Laboratorio de Propulsión a Chorro de la NASA)
<http://www.jpl.nasa.gov/>
I.N.T.A. (Instituto Nacional de Técnica Aeroespacial de España)
<http://www.inta.es/>
N.R.A.O. (Observatorio Nacional de Radio Astronomía de EE.UU.)
<http://www.nrao.edu/>

Al final de esta lista, añada una segunda línea horizontal para delimitar el área de enlaces. Ahora observe que, aunque haya introducido el texto, necesita introducir los enlaces asociados a cada URL para que el visitante pueda navegar por ellas desde nuestra propia página. Seleccione con el ratón la primera URL (<http://www.nasa.gov>) y haga clic con el botón derecho del ratón sobre la selección. En el menú contextual que aparece, haga clic sobre **Insertar enlace** y, en la ventana **Atributos** del enlace a realizar, vaya a la pestaña **A URL**. Allí, en el apartado **URL**, intro-

duzca www.nasa.gov, teniendo cuidado de no borrar el texto **http://** que ya aparece en dicho apartado. Si no lo estaba ya, marque la casilla **http:**.

Como no queremos que el usuario deje atrás nuestro sitio Web al pasar a uno nuevo, vamos a hacer que los enlaces se abran en una nueva ventana del navegador que el visitante esté utilizando. Para ello, en el apartado **Destino**, seleccione **Nueva ventana** y finalmente haga clic sobre el botón **Aceptar**.

Al volver a la ventana principal del editor verá que el texto aparece como un enlace (subrayado en azul). Si hace clic sobre el enlace, aparecerá una barra flotante llamada **URL**, indicando la dirección a la que llama y el **Destino**, en el que debe aparecer **Nueva ventana**. Puede cerrarla haciendo clic en el botón en forma de aspa que aparece en la esquina superior derecha de la barra. De la misma forma que ha creado este primer enlace, complete los siguientes asegurándose de que la opción **Nueva ventana** esté seleccionada como destino.

En este momento vamos a añadir a la página una sección con un formulario que sirva a los visitantes para enviarnos nuevos enlaces que hayan encontrado en la Web y que quieran compartir. Dicho formulario contendrá una serie de casillas de verificación, cajas

CAMPOS DE TEXTO

En HTML un campo de texto se puede especificar mediante la etiqueta **<INPUT TYPE="text" NAME="Identificador">**, siendo posible usar los parámetros **SIZE** y **MAXLENGTH** para definir su tamaño, a la vez que con **VALUE** podemos incluir un texto por omisión. Veamos un pequeño ejemplo:

```
<INPUT TYPE="text"
NAME="Nombre"
SIZE="20"
MAXLENGTH="15"
VALUE="Escriba su nombre">
```

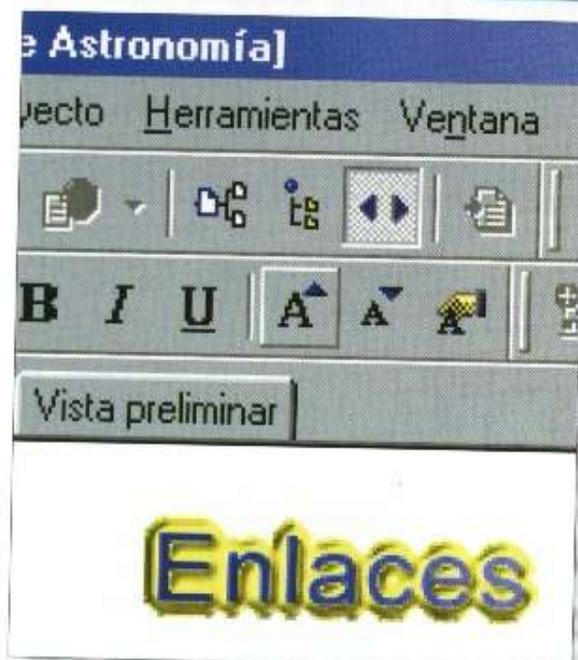


FIG. 5 Gracias al botón Aumentar tamaño del font podremos ampliar el texto en la página.

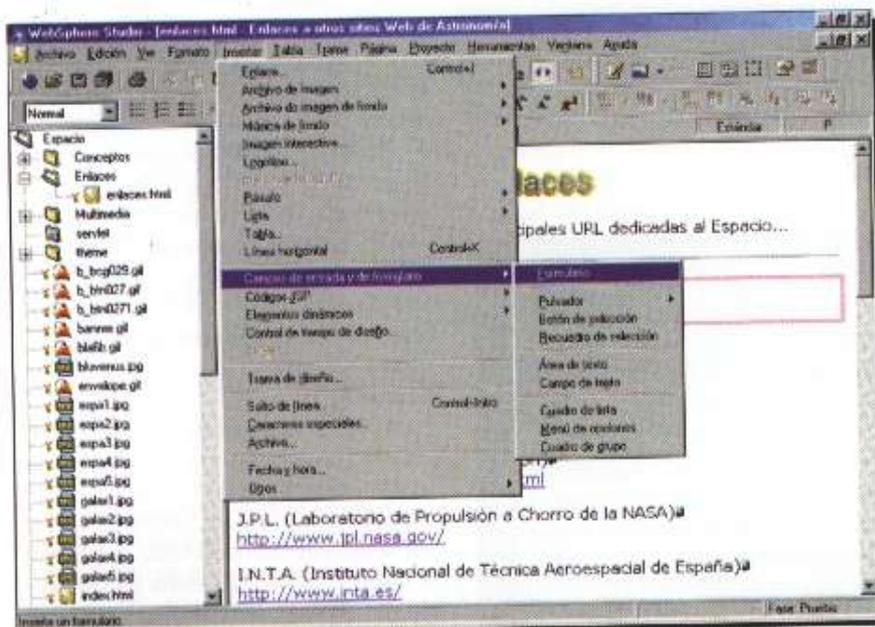


FIG. 6 Desde Formulario podremos crear el marco donde irán incluidos todos los demás objetos.

de texto y opciones que el usuario tendrá que rellenar, pulsando finalmente un botón para enviárnoslas. De esta manera, y como ya comentamos anteriormente, podremos disponer finalmente de una serie de archivos que nos llegarán por correo electrónico y que tendrán el mismo formato, por lo que realizar una sencilla aplicación para tratarlos no entrañará demasiados problemas. Para comenzar, y debajo de la última línea horizontal introducida, escriba *Más enlaces*, centrando el texto y usando el botón **Aumentar tamaño del font** para crear un subtítulo. Inmediatamente debajo, y con un tamaño y un formato normales, escriba lo siguiente:

Si lo desea, puede hacernos llegar más direcciones de sitios Web relacionados con el mundo de la Astronomía usando el siguiente formulario. Tras su análisis, dichas direcciones serán publicadas en esta página junto con su nombre.

Coloque el cursor inmediatamente debajo de este texto (recuerde que debe usar la tecla **Flecha abajo** en vez de la tecla **Enter**) y estará en condiciones de comenzar el diseño del formulario. Introduzca un retorno de ca-

rrero para separar el formulario del texto y abra el menú **Insertar**, para más tarde hacer clic sobre la opción **Formulario** contenida dentro del apartado **Campos de entrada y de formulario**.

Observe que aparecen un nuevo cuadro y una nueva barra de herramientas flotante en la pantalla. Dicho cuadro representa el área que contendrá los objetos que componen el formulario, mientras que la nueva barra contiene las herramientas necesarias para

crear dichos objetos. Sitúe el cursor dentro del cuadro de formulario y escriba *Nombre* con un retorno de carro detrás. Seguidamente, haga clic, dentro de la barra de herramientas **Campos de entrada y de formulario**, en el icono **Insertar campo de texto**. En la ventana que aparece, y dentro de la pestaña **Campo de texto**, escriba como **Nombre** el texto *NOMVIS*. Como **Tipo de entrada** debe seleccionar **Texto**, mientras que en **Valor inicial** debe introducir *Escriba aquí su nombre*. Haga clic sobre el botón **Aceptar** y volverá a la ventana del editor, donde habrá aparecido una caja de texto en el lugar apropiado. Vuelva a introducir otro retorno de carro para separar las siguientes etiquetas.

Siguiendo el mismo procedimiento, cree otros dos campos de texto, precedidos respectivamente por los textos *Apellidos* y *Dirección URL*, a la vez que como **Nombre** introduce *APEVIS* y *URLVIS* en las ventanas correspondientes. No olvide añadir dentro de **Valor inicial** las etiquetas *Escriba aquí sus apellidos* y *Escriba aquí la URL que desea que conozcamos* para que el visitante sepa para qué sirve cada campo.

Debajo de estos tres campos de texto escriba *Comentario* y haga clic en el botón **Insertar área de texto** de la barra **Campos de entrada y de for-**

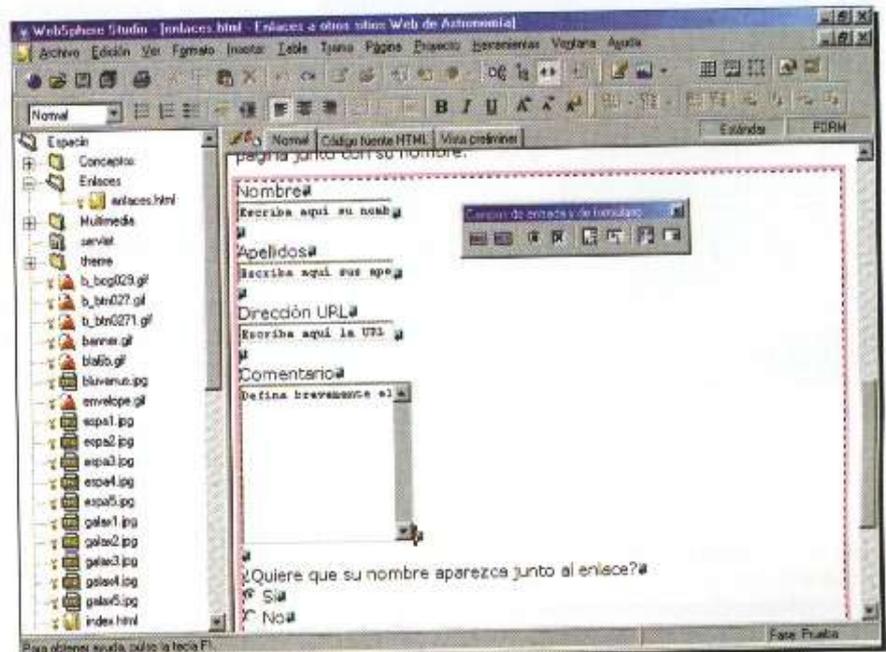


FIG. 7 Aspecto del formulario en modo de diseño.

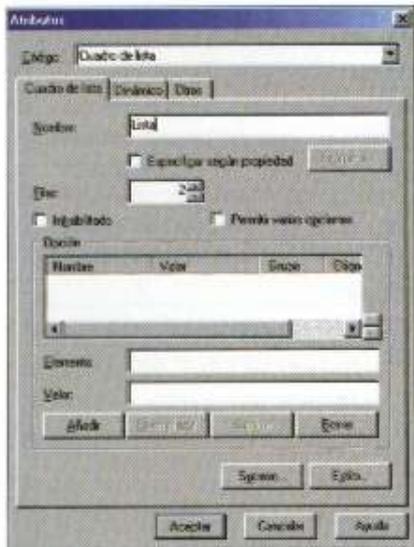


FIG. 8 Ventana de Atributos al insertar un cuadro de lista.

mulario. En la ventana emergente, de forma similar a los casos anteriores, introduzca *COMENT* como **Nombre** y aumente el número de filas a **10**. Como

ÁREAS DE TEXTO

Para crear un área de texto en código HTML es necesario usar la etiqueta `<TEXTAREA>`. Entre esta etiqueta y la que la finaliza, `</TEXTAREA>`, se puede escribir el texto que se desea mostrar inicialmente, o bien dejar el espacio en blanco si no queremos que se muestre nada. Con el parámetro `WRAP` se consigue que las líneas más largas que el área de texto se recorten, usando los valores "Physical", "Virtual" u "Off". Con `ROWS` y `COLS` podemos definir el número de filas y columnas respectivamente en el área de texto. Veamos un ejemplo de código:

```
<PROGRAMA
<TEXTAREA
NAME="Comentarios"
ROWS="4" COLS="60"
WRAP="Physical">Escriba
sus comentarios</
TEXTAREA>
```

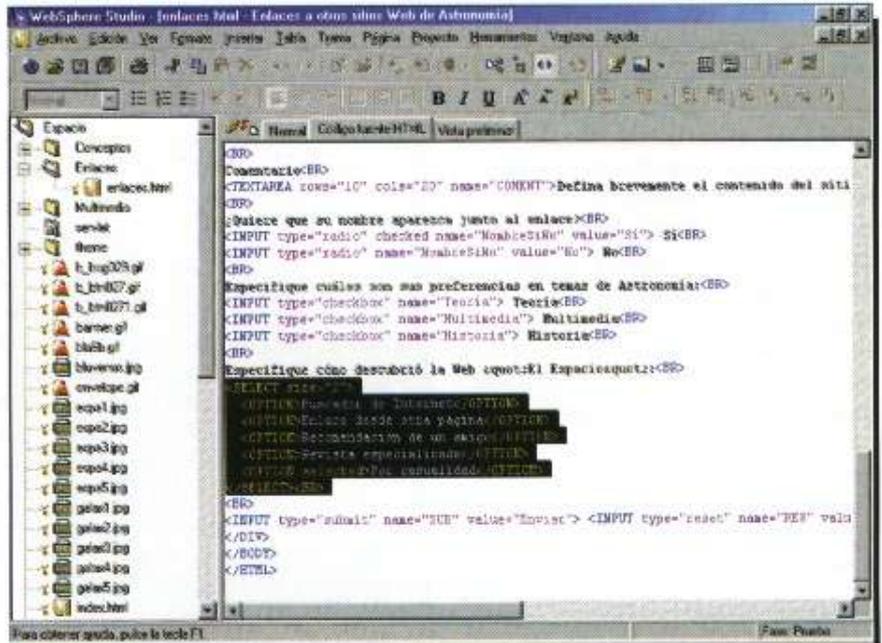


FIG. 9 Aspecto del código HTML del formulario, con la zona del cuadro de lista destacada.

Valor inicial escriba *Defina brevemente el contenido del sitio que nos envía* y haga clic en el botón **Aceptar**, con lo que podrá observar que ha aparecido una nueva clase de objeto, en la que el visitante podrá escribir cantidades mayores de texto, en este caso un comentario acerca del sitio Web del que nos envía su URL.

Escriba ahora el texto *¿Quiere que su nombre aparezca junto al enlace?*, volviendo a introducir un retorno de carro al final. Haga clic ahora sobre el botón **Insertar botón de selección**.

CASILLAS DE VERIFICACIÓN

Para insertar una casilla de verificación en código HTML se debe usar la etiqueta `<INPUT TYPE="checkbox" NAME="Nombre">`. El parámetro `CHECKED="CHECKED"` nos permitirá que la casilla aparezca marcada inicialmente. Vea este ejemplo:

```
<INPUT
TYPE="CHECKBOX"
NAME="Casilla1"
VALUE="No"
CHECKED="CHECKED">
```

En la nueva ventana, dentro de **Nombre de grupo** introduzca *NombreSiNo*. En **Valor** debe escribir **Sí**, mientras que en **Estado inicial** debe elegir la opción **Seleccionado**. Haga clic en **Aceptar** y verá que aparece un botón seleccionado. Escriba a su lado **Sí**. Repita esta operación para insertar, debajo de éste, otro botón de selección que tenga el valor **No** y que aparecerá sin seleccionar. Para conseguir esto, y permitir que sólo uno de ellos pueda ser elegido en un momento determinado, al crear el segundo botón y dentro de la ventana **Atributos** especifique como **Nombre de grupo** el valor *NombreSiNo* (debe aparecer al abrir la lista desplegable de la derecha). Pruebe ahora la página haciendo clic en la pestaña **Vista preliminar**. Observe que si hace clic sobre **No** la selección desaparecerá a su vez de **Sí**. Guarde todos los cambios antes de proseguir.

Continúe ahora con la creación del formulario. Introduzca el texto *Especifique cuáles son sus preferencias en temas de Astronomía:* y seleccione un nuevo tipo de objeto haciendo clic sobre el icono **Insertar recuadro de selección**. Introduzca como **Nombre Teoría**, haga clic en el botón **Aceptar** y recuerde escribir al lado del recuadro de selección el texto *Teoría*. Repita esta operación para crear los recuadros *Mul-*

timedia e Historia. Vuelva a situar el cursor con un retorno de carro y escriba el texto *Especifique cómo descubrió la Web "El Espacio"*. Haga clic sobre el botón **Insertar cuadro de lista** y pasará a una ventana algo más compleja que las anteriores, donde podrá definir una lista de selección. En esta lista vamos a introducir una serie de opciones de las que el usuario podrá elegir una. Como **Nombre**, dentro de la ficha **Cuadro de lista**, introduzca *Lista1*. En el campo **Filas** podremos elegir la altura que deseamos que muestre la lista en la pantalla; si introducimos más opciones que altura tenga la lista, el cuadro mostrará automáticamente los correspondientes botones para acceder a todas las opciones. De

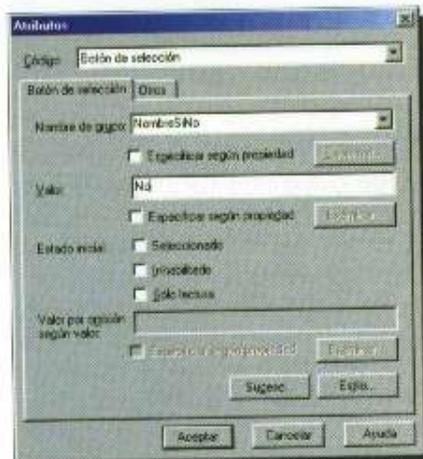


FIG. 10 Si no elegimos el nombre de grupo correcto el botón de opción no funcionará.

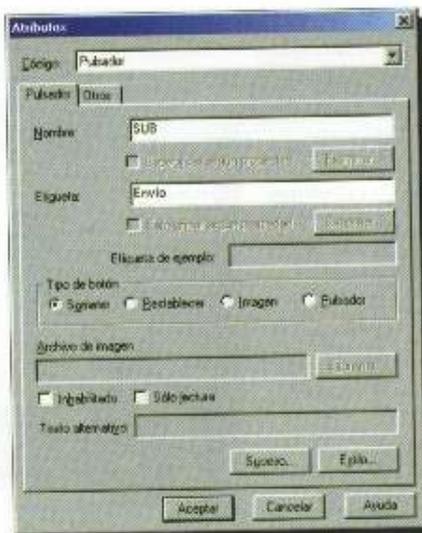


FIG. 11 El botón Enviar (SUB) será imprescindible para poder recibir los datos del visitante de la página.

esta forma, introduzca 4 en el campo **Filas**, y pase al recuadro **Opción**, donde tendrá que definir una a una las opciones que incluirá su lista. Observe que existe una casilla llamada **Permitir varias opciones**, que permite que el visitante pueda dar varias respuestas. En nuestro caso sólo queremos una respuesta, por lo que dejaremos sin seleccionar dicha casilla. Siguiendo con el recuadro **Opción**, introduzca en **Elemento** el texto *Buscador de Internet* y haga clic sobre el botón **Añadir**. Observe que ha aparecido una nueva opción en la lista con dicho **Elemento** como nombre. De la misma manera, y teniendo cuidado con el orden dado así

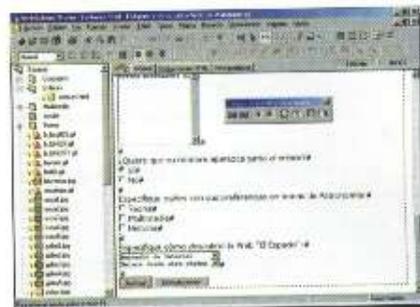


FIG. 12 El formulario ya está creado. Sólo falta configurar el destino.

como de no olvidar hacer clic sobre el botón **Añadir**, introduzca las siguientes opciones:

**Enlace desde otra página
Recomendación de un amigo
Revista especializada
Por casualidad**

Cuando haya terminado, haga clic sobre el botón **Aceptar** para volver a la ventana del editor y probar el formulario haciendo clic en la pestaña **Vista preliminar**. Guarde los cambios efectuados.

BOTONES EN LOS FORMULARIOS

Llegados a este punto tendremos un formulario con los objetos que necesitamos para que los visitantes de la página nos envíen, si lo desean, información sobre su sitio Web. Sin embargo, aún necesitamos un botón que nos envíe los datos e incluso otro que permita borrar lo que el usuario ha introducido y devolver los objetos a su valor inicial. IBM WebSphere Studio genera automáticamente el código para dichos botones, aunque nosotros debemos introducir los parámetros necesarios referentes al destino de los datos que aparezcan en la página. Veamos cómo hacer esto.

En primer lugar, sitúe el cursor al final del formulario, y haga clic en la barra **Campos de entrada y de formulario**, sobre el botón **Insertar botón de envío**. Aparecerá de nuevo la ventana **Atributos**, esta vez mostrando las propiedades del **pulsador**. En **Nombre** introduzca *SUB*, mientras que en **Etiqueta** debe escribir *Enviar*. Observe la diferencia: **Nombre** especifica el nombre interno que tiene el bo-

BOTONES DE OPCIÓN

Usaremos la etiqueta `<INPUT TYPE="radio" NAME="Grupo">` para insertar un botón de opción en código fuente HTML. "Grupo" será compartido por todos los botones que, valga la redundancia, pertenezcan al mismo grupo, esto es, sólo un botón de opción de cada grupo podrá estar marcado en un momento dado. Si queremos que uno de ellos aparezca marcado por omisión basto con añadir el parámetro `CHECKED="CHECKED"`. Con `VALUE="valor"` se puede indicar el valor asociado a cada botón dentro del grupo. Observe con detenimiento el siguiente ejemplo:

```
<P><INPUT TYPE="radio" NAME="Op1" VALUE="1"
CHECKED="CHECKED">Valor 1</P>
<P><INPUT TYPE="radio" NAME="Op2" VALUE="2">Valor
2</P>
<P><INPUT TYPE="radio" NAME="Op3" VALUE="3">Valor
3</P>
```

LISTAS DE SELECCIÓN

En HTML se pueden crear listas desplegables y listas de varios elementos mostrados a la vez con la etiqueta `<SELECT NAME="identificador">`. Para diferenciar entre lista desplegable y lista múltiple se utiliza el parámetro `SIZE`, con el cual se puede indicar la lista de elementos que es posible mostrar simultáneamente en la pantalla. Como una lista desplegable únicamente puede mostrar un elemento, el parámetro será definido como `SIZE="1"`, mientras que para una lista múltiple se emplearán valores superiores. Si desea permitir la selección de varios elementos a la vez dentro de una misma lista debe emplear el parámetro `MULTIPLE="MULTIPLE"`. Para definir las opciones de la lista se utilizará la etiqueta `<OPTION>`, usando tantas etiquetas como elementos deseemos que tenga la lista. Si desea que una opción aparezca seleccionada de antemano, use el parámetro `SELECTED="SELECTED"`. Observe el siguiente ejemplo:

```
<SELECT NAME="Lista1" SIZE="4" MULTIPLE="MULTIPLE">
  <OPTION VALUE="Elemento1"
    SELECTED="SELECTED">Elemento 1</OPTION>
  <OPTION VALUE="Elemento2">Elemento 2</OPTION>
  <OPTION VALUE="Elemento3">Elemento 3</OPTION>
  <OPTION VALUE="Elemento4">Elemento 4</OPTION>
  <OPTION VALUE="Elemento5">Elemento 5</OPTION>
</SELECT>
```

tón, mientras que **Etiqueta** hace referencia al texto que se muestra en la pantalla. En el recuadro **Tipo de botón** seleccione **Someter**, si no lo estaba ya. Observe que también puede añadir una imagen al botón (apartado **Archivo de imagen**), aunque en este caso ignoraremos esta posibilidad. Haga clic sobre **Aceptar**.

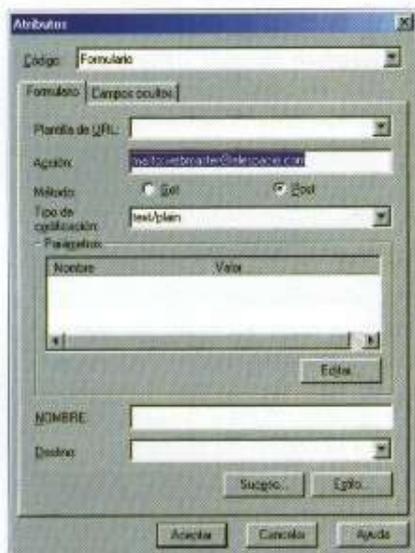


FIG. 13 El formulario se enviará mediante correo electrónico.

El siguiente paso es insertar el botón que permita restaurar los objetos a sus valores iniciales. Junto al botón que acaba de insertar (que ahora debe

mostrar la etiqueta **Enviar**), y separado por un espacio en blanco, haga clic sobre **Insertar botón de restablecimiento** en **Campos de entrada y de formulario**. Como **Nombre** escriba **RES**, y como **Etiqueta**, **Restablecer**. Observe que en **Tipo de botón** aparece seleccionado **Restablecer**; si no lo estuviera debe marcarlo antes de volver a hacer clic sobre el botón **Aceptar**. Desde este momento aparecerán los dos botones en sus lugares respectivos. Si prueba ahora la página en modo **Vista preliminar** podrá comprobar que si introduce datos nuevos y hace clic sobre el botón **Restablecer** todos los campos y objetos volverán a mostrar sus valores iniciales. Esto es importante cuando un usuario se ha equivocado en varios datos introducidos y desea empezar desde cero: es mucho más sencillo y rápido hacer clic en un botón que marcar el texto, borrarlo y ajustar las casillas a sus valores iniciales.

EL ENVÍO DEL FORMULARIO

Como dijimos al principio de esta unidad, el fin de un formulario es servir de interfaz entre los datos que introduce el visitante y la persona o sistema que los debe recibir. Así, al disponer

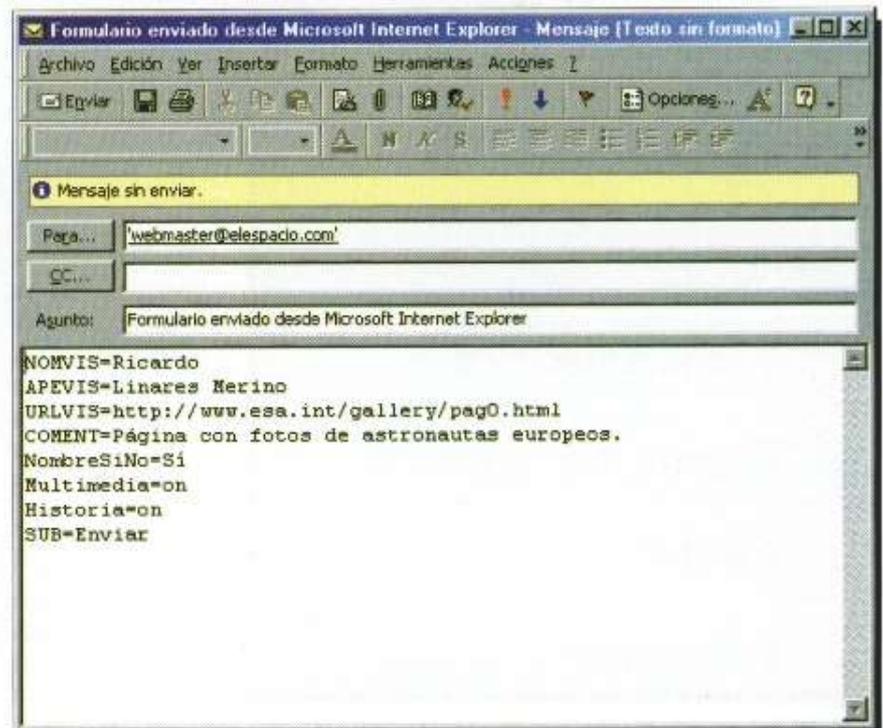


FIG. 14 Así recibiremos los datos en nuestro programa de correo electrónico.

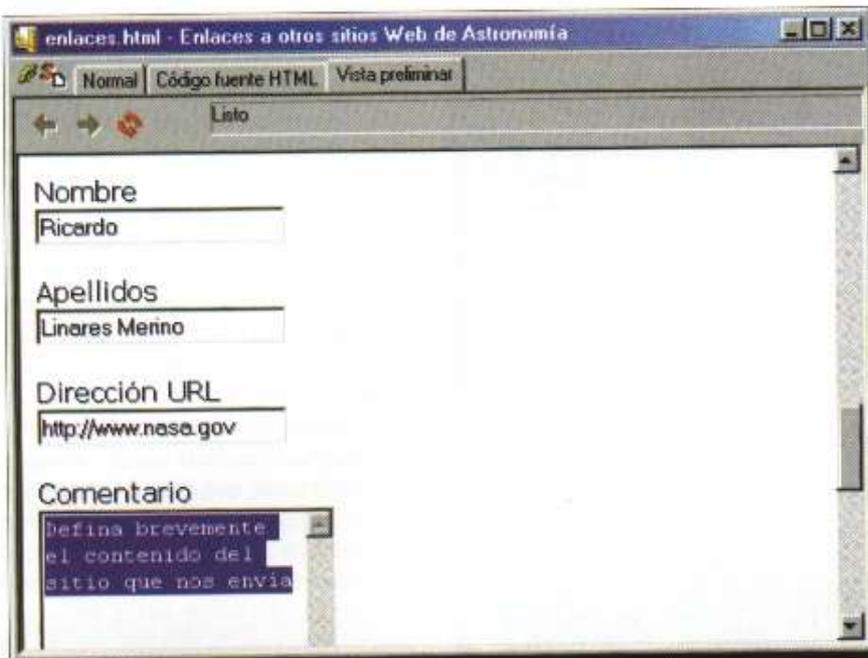


FIG. 15 Al probar la página podremos comprobar si funciona correctamente.

de datos ordenados por campos, el proceso de automatización se acelera y se simplifica. No obstante, dichos datos deben ser procesados para que su formato se ajuste a nuestras necesidades. De esto y de muchas más cosas se encargan los CGI, que ya comentamos anteriormente.

No obstante, y dado que los CGI suelen ser proporcionados por el proveedor de servicios de Internet que alberga el sitio Web, nosotros vamos a suponer un caso en que los datos del

formulario nos llegarán directamente a nuestra cuenta de correo desde la cuenta del visitante, que tendrá que disponer de un programa de correo electrónico en su sistema para poder enviarnos el formulario (requisito que se cumple prácticamente siempre, por otra parte). La operación, pues, se completará tras pulsar el botón **Enviar**, momento en el cual el visitante recibirá un mensaje de advertencia de su sistema antes de lanzarse su aplicación de correo electrónico para enviar automáticamente los datos del formulario a nuestra propia cuenta. De esta forma y si el método de envío es **GET**, recibiremos tantos mensajes como visitantes hayan rellenado el formulario y pulsado el botón **Enviar**. Cada uno de estos mensajes contendrá un archivo, **POSTDATA.ATT**, en formato ASCII que albergará los datos divididos mediante separadores, en este caso por el símbolo de interrogación "?". En caso de que

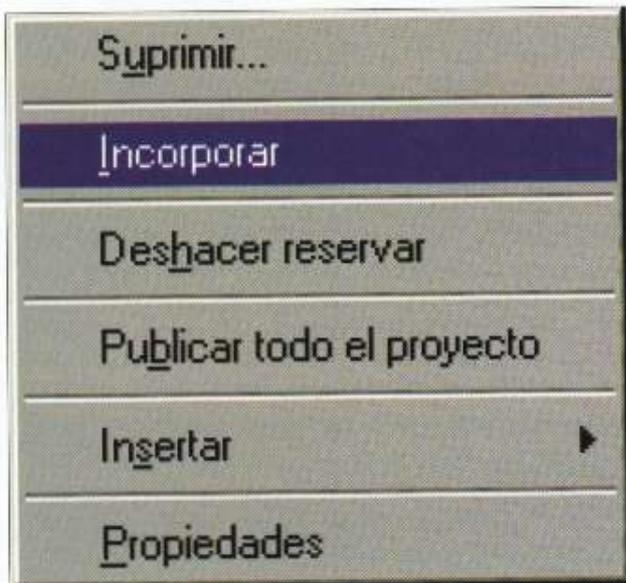


FIG. 16 Seleccione la opción Incorporar para abortar la reserva de archivos por parte de IBM WebSphere Studio.

el método elegido sea **POST**, recibiremos directamente los datos del formulario en un mensaje de correo electrónico, organizados con el formato **objeto= valor**. Por comodidad, elegiremos la segunda opción en el momento oportuno.

Para definir el destino al que serán enviados los datos del formulario, haga clic con el botón derecho en cualquier zona en blanco dentro del formulario y, en el menú contextual que aparece, seleccione la opción **Atributos**. Se abrirá la ventana **Atributos** vinculada al

BOTONES DE ENVÍO Y DE RESTABLECIMIENTO

Los botones de envío (tipo **submit**) son definidos en **HTML** simplemente mediante la etiqueta **<INPUT TYPE="submit" NAME="identificador">**. Si deseamos cambiar el texto que se muestra dentro del botón se puede utilizar el parámetro **VALUE="texto"**. Puede incluso añadir un archivo gráfico para crear el botón con el parámetro **SRC="archivo"**, aunque debe cambiar el **TYPE** a **"image"**. Observe este ejemplo:

```
<INPUT TYPE="submit"
NAME="Enviar"
VALUE="Enviar">
```

Esto crea un botón con el nombre "Enviar". Para que en vez de botón aparezca una imagen, podría usar:

```
<INPUT TYPE="image"
NAME="EnviarGraf"
SRC="archivos\
botonenvio.gif">
```

Para crear un botón de restablecimiento el procedimiento es muy similar, sólo hay que cambiar **TYPE** a **"reset"**, como en el siguiente ejemplo:

```
<INPUT TYPE="reset"
NAME="Restablecer"
VALUE="Restablecer">
```

OTROS CAMPOS DE FORMULARIOS

Además de lo visto, existen otros tipos de campos que se pueden insertar en un formulario. En primer lugar disponemos de campos de contraseña o passwords que son muy similares a los campos de texto, con la diferencia de que los caracteres introducidos son cambiados automáticamente por asteriscos cuando el visitante escribe en la caja apropiada. Para introducir un campo de contraseña debe usar la etiqueta `<INPUT TYPE="password" NAME="identificador">`. Además, puede emplear los campos `SIZE="longitud"` para definir la longitud en caracteres que tendrá el campo y `MAXLENGTH="máximo"` para indicar la longitud máxima del texto que se tecleará en el interior de la caja.

```
<INPUT TYPE="password" NAME="contraseña" SIZE="12" MAXLENGTH="10">
```

Además, puede ser excepcionalmente útil añadir campos ocultos en nuestros formularios, generalmente para definir constantes que hay que pasarle al programa CGI o valores predefinidos que no deben mostrarse en la pantalla. Para crear un campo oculto, usaremos la etiqueta `<INPUT TYPE="hidden" NAME="identificador" VALUE="valor">`. En este tipo de etiqueta se introduce el valor que queremos que esté oculto dentro del parámetro VALUE.

```
<INPUT TYPE="hidden" NAME="NombreArchivo" VALUE="paginaprinc.html">
```

Finalmente, otra característica importante que podemos añadir es la capacidad de "subir" archivos (file upload), que, en contra de lo que pueda parecer, se consigue de forma muy sencilla. Para insertar un campo de este tipo basta con introducir la etiqueta `<INPUT TYPE="file" NAME="identificador">`. Como en otros campos, puede usar los parámetros `SIZE` y `MAXLENGTH` para ajustar los caracteres que muestre la caja de texto en la pantalla.

```
<INPUT TYPE="file" NAME="Archivo" SIZE="30" MAXLENGTH="40">
```

código **Formulario**. Dentro de **Acción** puede introducir el lugar donde serán enviados los datos del formulario o la acción a llevar a cabo por el programa CGI. En nuestro caso, llamaremos mediante **mailto** a nuestra dirección de correo. Escriba, pues, dentro de **Acción** el comando `mailto:webmaster@elespacio.com`. Observe que dicha dirección es ficticia, por lo que a la hora de probar la página ésta no llegará a enviar la información requerida. Si desea probar el resultado final, puede sustituir la línea por `mailto:<dirección>`, cambiando "dirección" por su propia dirección de e-mail.

Seguidamente, seleccione **Post** como **Método**, ya que vamos a enviar los datos por correo electrónico. Abra la lista desplegable **Tipo de codificación** y seleccione la opción **text/plain**. Deje las demás opciones como están y haga clic sobre el botón **Aceptar** para

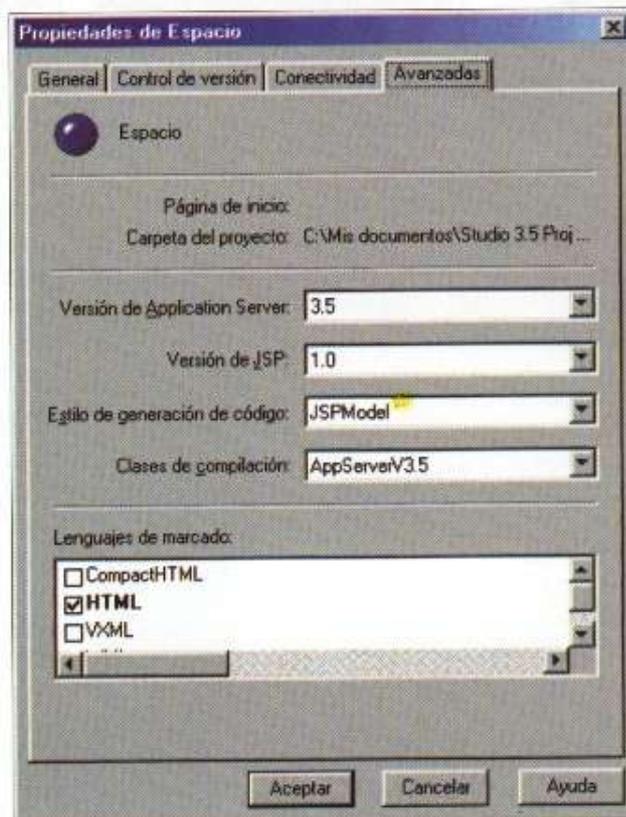


FIG. 17 Desde la ventana **Propiedades** se podrán ajustar algunos parámetros avanzados.

volver a la ventana principal del editor. Guarde todos los cambios usando el botón **Guardar todo** y pruebe la página creada, abriendo el menú **Herramientas | Navegador WWW** y seleccionando su navegador de Internet predeterminado, por ejemplo, **Internet Explorer**. Aparecerá la página en cuestión, donde debe introducir algunos valores en el formulario para finalmente hacer clic sobre el botón **Enviar**.

En ese momento, y si dispone de un programa de correo electrónico convenientemente configurado en su sistema, el formulario será enviado a la dirección indicada con **mailto**.

Este proceso puede variar enormemente en función de la aplicación que esté usando (Outlook Express, Netscape, etc.), pero el resultado será siempre un mensaje como el siguiente:

NOMVIS=Ricardo
APEVIS=Linares Merino
URLVIS=http://www.esa.int/gallery/pag0.html
COMENT=Página con fotos de astronautas europeos.
NombreSiNo=Si
Multimedia=on
Historia=on
SUB=Enviar

Observe que cada línea muestra en primer lugar el campo **Nombre** que introdujo en cada ventana de atributos correspondiente, a continuación el separador de campos, en este caso el símbolo "=", y finalmente el valor que el usuario haya introducido en cada opción, tomando en algunos casos el valor que hayamos fijado nosotros (los botones de opción **NombreSiNo** fueron definidos con dos valores posibles, **Si** o **No**) y en otros simplemente se nos informará de que están activados (las casillas de verificación **Multimedia** e **Historia** muestran como valor **on**).

De esta forma, al disponer de archivos formateados de la misma manera en nuestro ordenador resultará casi trivial escribir un programa que pueda analizar los mensajes recibidos e incorporarlos, por ejemplo, a una base de datos que podamos utilizar posteriormente.

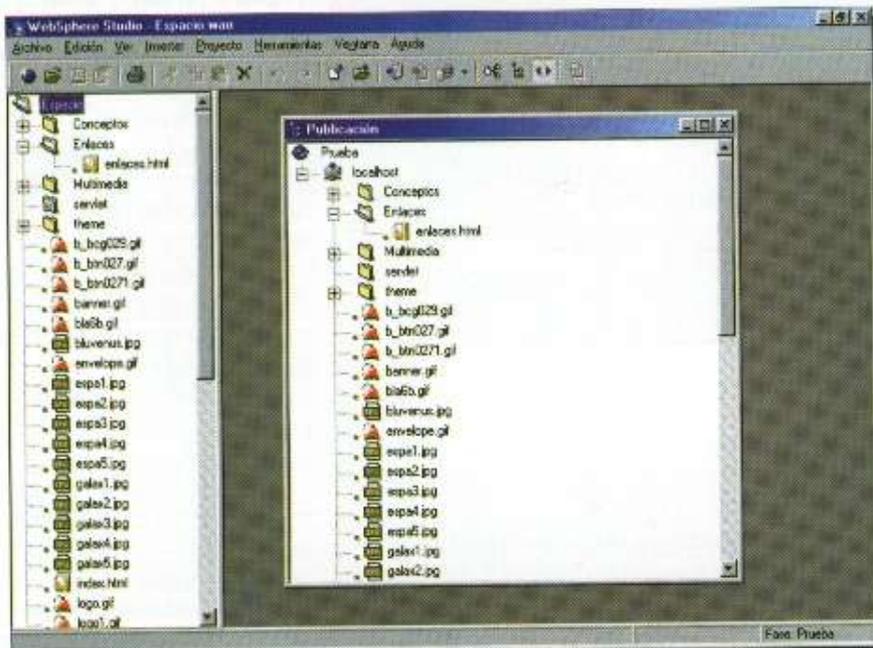


FIG. 19 Desde la Vista de publicación se procederá a publicar el sitio Web.

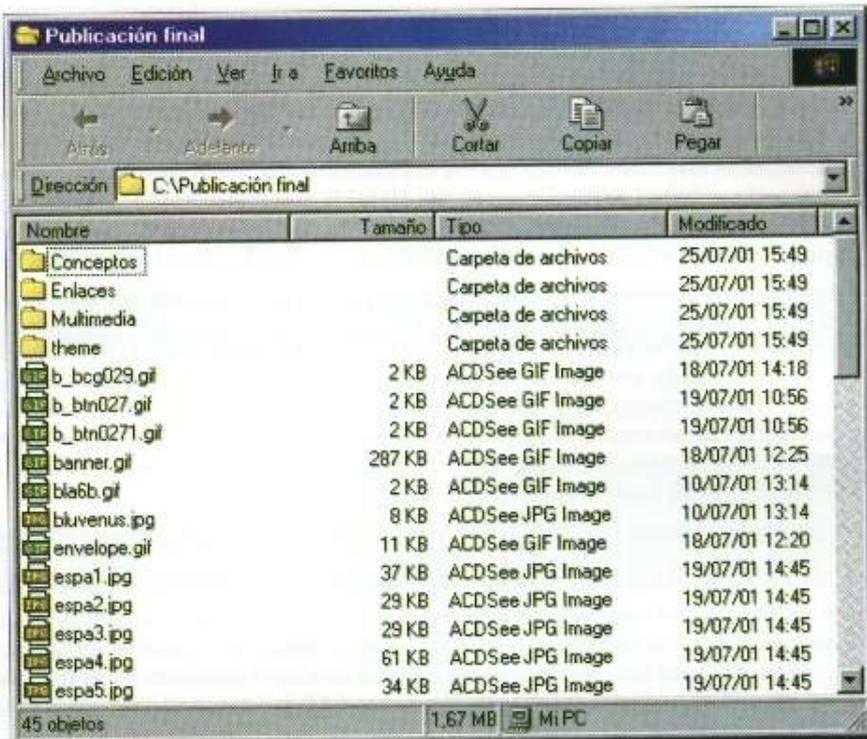


FIG. 18 Debe crear una carpeta donde IBM WebSphere Studio publicará los archivos.

ÚLTIMOS DETALLES

A lo largo de estas cuatro entregas dedicadas a IBM WebSphere Studio se ha ido desarrollando el sitio Web de ejemplo "El Espacio", que ya está prácticamente completado. Ha llegado, pues, el momento de comprobar que todo funcione y que el sitio no contenga

enlaces rotos o errores en cuanto a código. Una cuidadosa revisión de todos los objetos introducidos ahorrará mucho tiempo cuando en el futuro llegue la hora de, una vez publicado el sitio, proceder a su mantenimiento. Guarde en primer lugar todos los cambios y cierre las ventanas de edición que pudiera tener abiertas para evitar realizar modificaciones no deseadas en sus archivos. Una vez hecho todo esto, vaya a la ventana **Vista de objetos** y haga clic con el botón derecho del ratón sobre el archivo principal del pro-

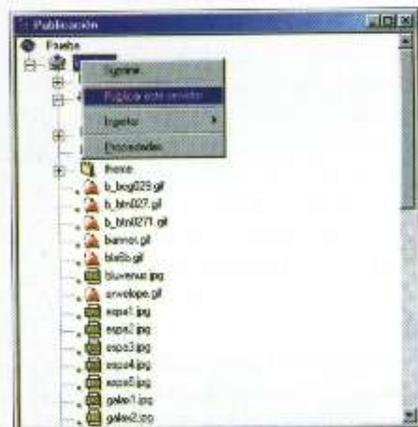


FIG. 20 Haga clic aquí para publicar el sitio.

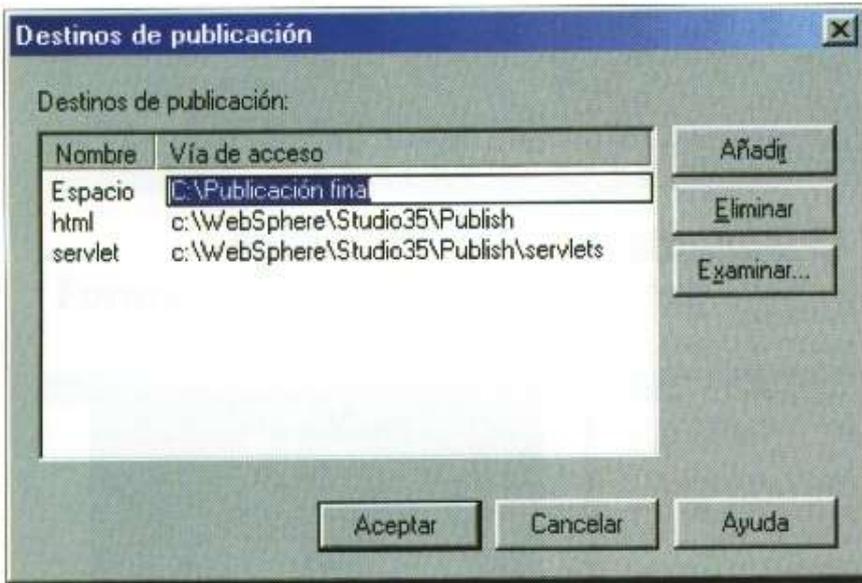


FIG. 21 Introduzca aquí el nombre de la carpeta donde publicará los archivos.

yecto, **Espacio**, que verá representado por un icono en forma de carpeta abierta.

En el menú contextual que aparece, seleccione la opción **Incorporar**. Tras unos momentos todos los archivos que estuvieran reservados pasarán a estar incorporados al proyecto principal. Puede comprobar esto fijándose en que los objetos que tenían una casilla roja de verificación junto a su icono carecen ahora de ella.

Con la incorporación de los archivos se persigue que se pueda publicar el proyecto en su totalidad y que todos los demás usuarios tengan acceso a ellos (recuerde que con IBM WebSphere Studio varios usuarios pueden trabajar en un mismo proyecto, con lo que si se tiene un archivo reservado para un usuario, otro no podrá modificarlo, aunque sí acceder a él).

Una vez hecho esto, vuelva a hacer clic con el botón derecho del ratón sobre **Espacio** y seleccione la opción **Propiedades**, con lo que pasará a la ventana **Propiedades de Espacio** en la que podrá conocer algunas características sobre el sitio Web que ha creado, como, por ejemplo, el número de **Carpetas** y **Archivos** que contiene en

sus respectivos apartados, o la ruta donde se encuentra el **Archivo de proyecto**.

En el apartado **Comentarios** puede añadir un pequeño texto para que otros usuarios que abran el proyecto puedan tener información sobre éste. Puede escribir, por ejemplo:

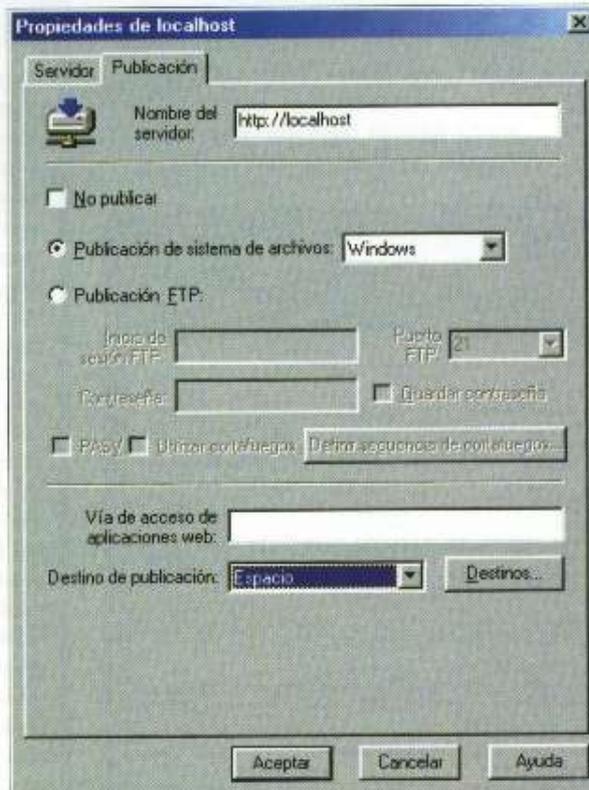


FIG. 22 El destino de la publicación debe ser definido como "Espacio".

"El Espacio", sitio Web de Astronomía
Creado por RLM-LH
Contiene páginas con frames

A su vez, dentro de la ficha **Conectividad** puede definir un proxy si trabaja con Internet desde una red, así como un cortafuegos como medida de seguridad ante posibles intrusiones en el sistema. Finalmente, en la ficha **Avanzadas** puede cambiar los números de versiones usadas y modelos de compilación de los programas basados en JavaScript que añada al sitio, así como ver y modificar una lista de los **Lenguajes de marcado** que soportará su sitio. Haga clic sobre **Aceptar** cuando haya finalizado para volver a IBM WebSphere Studio.

Ha llegado el momento de publicar una vez más el sitio Web, pero esta vez como prueba de producción final. Abra el Explorador de Windows y cree en su disco duro una nueva carpeta donde publicar los archivos, por ejemplo *Publicación final*. Vuelva a IBM WebSphere Studio y abra la **Vista de publicación**.

Haga clic con el botón derecho sobre el servidor virtual de su sistema, es decir, **localhost**, y seleccione la opción **Propiedades**. En la ventana **Propiedades de localhost** haga clic sobre el botón **Destinos...** y, en la nueva ventana, compruebe que no haya ninguna entrada bajo **Nombre** llamada **Espacio**. Si así fuera, selecciónela y haga clic sobre **Eliminar**. A continuación, haga clic sobre **Añadir** e introduzca *Espacio* como nombre. Seleccione el apartado en blanco **Vía de acceso** y haga clic sobre el botón **Examinar...** En la nueva ventana debe seleccionar la carpeta que ha creado (**Publicación final**). Haga clic sobre **Aceptar** en todas las ventanas para volver a **Propiedades de localhost**, donde debe seleccionar **Espacio** en la lista desplegable **Destino de publicación**. Vuelva a hacer clic sobre **Aceptar** para volver al modo **Vista de publicación**. Haga clic ahora sobre

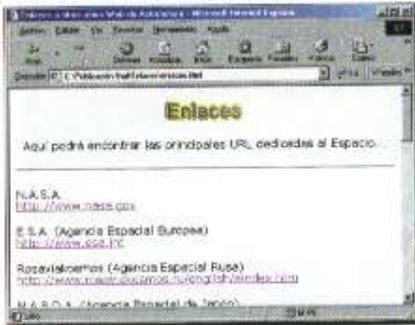


FIG. 23 El último paso será comprobar el sitio Web desde Internet Explorer.

localhost y seleccione **Publicar este servidor**. Aparecerá una ventana en la que se le pedirá confirmación para la creación de carpetas en su sistema. Marque la casilla **No volver a mostrar este diálogo** y haga clic sobre el botón **Sí**. Observe que al finalizar la publicación los archivos que aparecen en la **Vista de objetos** muestran un pequeño círculo verde junto a su icono, señal de que han sido publicados con éxito. Vuelva al Explorador de Windows y compruebe, dentro de la carpeta **Publicación final**, que el sitio Web funciona en su totalidad haciendo doble clic sobre el archivo **index.html**. Como de costumbre, debe prestar especial atención a todos los enlaces, comprobando esta vez que el formulario creado funciona correctamente y envía la información requerida, por ejemplo a su dirección de correo electrónico.

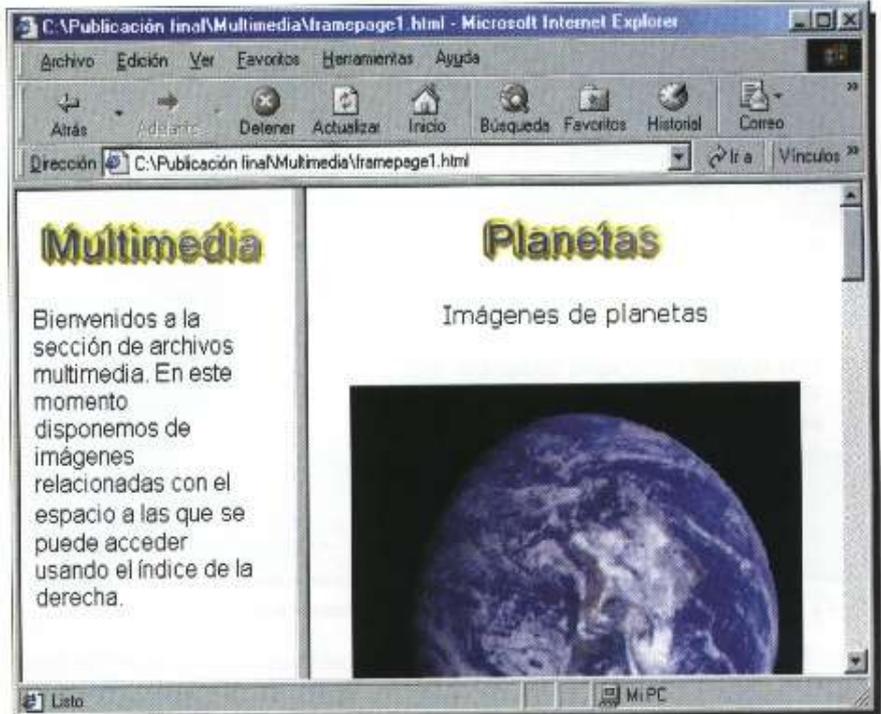


FIG. 24 Como de costumbre, hay que tener cuidado y comprobar especialmente las páginas con frames.

Debe además comprobar el código fuente de la página desde IBM WebSphere Studio, aprovechando para practicar lo explicado en los cuadros de este número dedicados a formularios. El sitio Web ha sido completado. En la próxima entrega veremos cómo añadir toda la potencia de Java a "El Espacio" de forma sencilla y con la ayu-

da de los asistentes que incluye IBM WebSphere Studio.

RESUMEN

En esta entrega de IBM WebSphere Studio hemos visto cómo podemos añadir más interactividad a nuestro sitio gracias a los formularios, mediante los cuales se pueden incorporar objetos tales como botones, casillas de verificación o cuadros de texto a una página Web. Además, hemos finalizado la creación de las páginas principales de nuestro sitio Web de ejemplo "El Espacio" añadiendo una página de enlaces a otros sitios Web con un formulario para incluir nuevos sitios remitidos por los visitantes. Además hemos completado la publicación final de la fase de prueba del proyecto, comprobando que todas las partes del sitio funcionan desde un navegador de Internet, en nuestro caso Microsoft Internet Explorer.

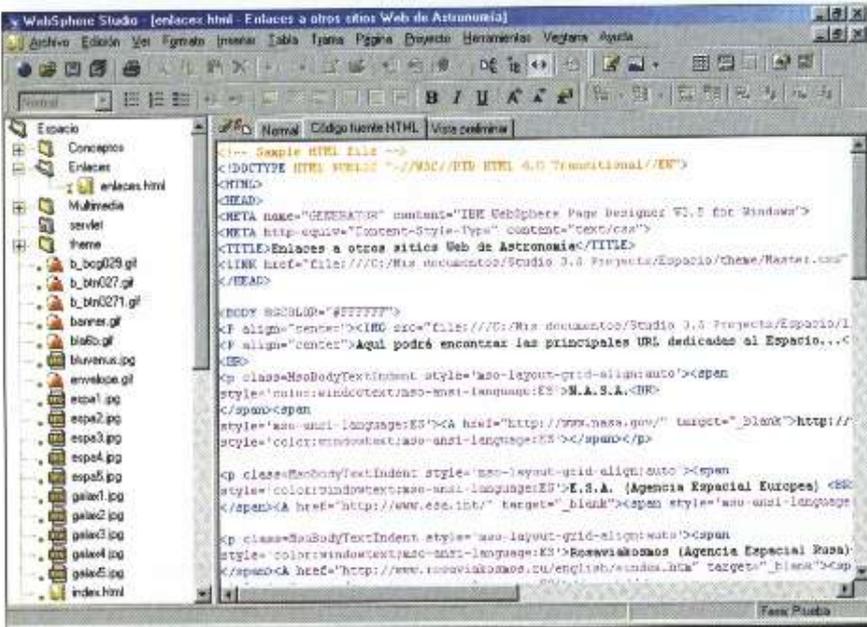


FIG. 25 Los archivos publicados aparecen con un círculo verde al lado del icono.