

# DHTML

Oscar González

Es raro encontrar hoy en día a una persona relacionada con el mundo del ocio o de la empresa que no sepa todavía qué es Internet. En un breve espacio de tiempo, la red de redes ha experimentado un crecimiento exponencial que todavía parece no tener freno. Para los millones de personas que diariamente se conectan para navegar y conseguir información, el término HTML ha pasado a formar parte de su lenguaje cotidiano. ¿Qué navegante no ha pensado alguna vez en hacer su propia página personal, en la que poder mostrar sus gustos y aficiones?

En otra vertiente más profesional, el lenguaje HTML se ha convertido en un estándar de facto en el mundo de la programación, siendo rara la empresa de software que no haya realizado una migración o ampliación de sus productos de cara a la integración con las famosas páginas Web. Muchos programas, entre los que se incluyen las más conocidas "suites" de ofimática, ya ofrecen posibilidades de exportación de datos a páginas Web, haciéndonos posible la tarea de compartir nuestros documentos, ya sean ficheros de texto, bases de datos u hojas de cálculo, con el resto de usuarios de Internet.

Pero el lenguaje HTML (Hypertext Mark-Up Language), siempre ha adolecido de una gran desventaja, y es el carácter estático de las páginas que produce.

El desarrollador de Internet puede crear mediante el lenguaje HTML cualquier tipo de página, desde una sencilla página de texto plano, hasta la más colorida que podamos imaginar. Pero ambas tienen una característica en común, y es que una vez visualizadas, no son capaces de realizar ningún cambio. Para que cambie el contenido que visualizamos en nuestro navegador nos vemos obligados a pulsar algún enlace para solicitar una nueva página al servidor.

Para paliar esta situación, han aparecido en el mercado una serie de tecnologías alternativas, como pueden ser



Figura 1. "Una página DHTML de Internet puede resultar mucho más vistosa y atraer a más usuarios"

GIFs animados, Java, ActiveX o animaciones Flash que pretenden llevar algo de “vidilla” a las páginas HTML. El problema de todas estas tecnologías suele ser la incompatibilidad entre los distintos navegadores y sus múltiples versiones, y la falta de disponibilidad inmediata, siendo necesaria la mayor parte de las veces una instalación previa del software o “plugin” añadido que vamos a utilizar. Otro de los problemas es que estas tecnologías tienen cada una su propio método de desarrollo, diferente al de HTML, y por tanto necesitan una inversión en tiempo y conocimientos que a veces nos resulta complicado evaluar si va a ser rentable o duradera.

Por tanto se hace necesaria una tecnología que implementen los navegadores “de fábrica”, y que no suponga la necesidad de adoptar una nueva forma de trabajo totalmente distinta a la utilizada con HTML. Para ello aparece en el mundo Web un concepto llamado DHTML (Dynamic HTML), que engloba a una serie de técnicas que nos van a permitir crear páginas activas, que soporten una interacción real con el usuario, y que nos ofrezcan la posibilidad de realizar vistosas animaciones, cambios de texto, elementos de navegación, etc.

Decimos que DHTML se trata de un concepto porque en realidad no existe nada que se pueda calificar con este término, sino que existen distintas tecnologías que entre todas forman lo que se podría llamar DHTML. Es decir, que ninguno de las organizaciones de creación de estándares de Internet (el más importante es el World Wide Web Consortium, W3C) ha definido el estándar de DHTML, sino que por el contrario, se han definido otra serie de estándares, como el del lenguaje ECMA (proveniente de Javascript), las hojas de estilo en cascada (CSS) o el modelo de objetos de documento (DOM). Es la conjunción de estos estándares la que forma lo que se viene llamando DHTML, ya que todos están orientados a construir contenidos dinámicos a partir de una base HTML.

## CASCADING STYLE SHEETS. UN NUEVO ESTÁNDAR DE ESTILO

El World Wide Web Consortium publicó en 1996 la especificación del lenguaje CSS (Cascading Style Sheets), que reúne las bases de un lenguaje utilizado para definir las distintas características de formato de los distintos elementos que constituyen una página Web. Tradicionalmente, a través del lenguaje HTML se podía especificar el formato de una serie de elementos individuales

contenidos en nuestra página (por ejemplo, un párrafo de texto) mediante una etiqueta o “Tag”, que definía el tipo de letra, tamaño y color del mismo. El siguiente código HTML mostraría en pantalla la palabra “Hola” en un tamaño “7” (el más grande), con un tipo de letra “Arial” y en color negro:

```
<FONT FACE="Arial" SIZE="7" COLOR="black">Hola</FONT>
```

Como vemos, la palabra a formatear se encierra entre dos etiquetas, una de comienzo y otra de finalización, que definen exactamente cómo queremos ver nuestra frase en pantalla. Esta filosofía de trabajo se aplica de igual forma a todos los elementos de una página Web, como pueden ser imágenes, tablas, fondos, listas, etc.

Pero en el día a día de la creación de páginas Web, surge una complicación de esta forma de orientar el diseño de una página, y es la cantidad de trabajo que hay que invertir en realizar cambios sobre nuestras páginas. Supongamos que estamos realizando un proyecto para una empresa, y que después de tener más de cien páginas hechas, nuestro cliente nos dice que prefiere un tipo de letra más grande y en otra gama de colores. La cantidad de cambios que tendremos que realizar entonces será enorme.

El estándar CSS1 viene a solucionar este problema de la siguiente manera. Mediante CSS, ahora podemos definir una serie de “estilos”, que aplicaremos a cualquier tipo de elemento de nuestra página web, de forma que el navegador utilizará el estilo adecuado con cada elemento de nuestra página. De esta manera, si organizamos nuestro código para tener una única “Hoja de Estilos” de la que dependan el resto de las páginas, con únicamente cambiar dicha hoja de estilos, el resto de páginas también se verán afectadas por los cambios de forma automática. En nuestro caso de estudio anterior, el esfuerzo de cambiarle el tipo de letra a todos los ficheros que componen nuestro proyecto sería mínimo, ya que únicamente tendríamos que retocar la hoja de estilo correspondiente.

Pero CSS no se queda únicamente ahí. También define una serie de convenciones para manejar la posición y tamaño de los distintos elementos HTML dentro de nuestra página, lo que nos permite saltarnos viejas limitaciones, y obtener efectos que antiguamente eran imposibles, como por ejemplo la superposición de elementos tales como texto o imágenes, definición exacta de espacios, posiciones y márgenes, gestión de listas, bordes, etc.

Una de las preguntas que entonces surgen es: Si CSS es un lenguaje de modelado, superior a HTML pero estático al fin y al cabo, ¿dónde está o aparece el contenido dinámico? La respuesta es que el contenido dinámico de nuestras páginas no lo implementa CSS, sino que se incluye gracias a la acción de otro estándar de Internet, que no es otro que el lenguaje JavaScript.

## JAVASCRIPT. PROGRAMACIÓN EN LA PARTE CLIENTE

Javascript es uno de los lenguajes más controvertidos de todo el mundo de la informática. Debido a la gran importancia que tiene dentro del software de navegación, y al ser este el mercado con más expansión del mundo, los dos grandes “monstruos” del mercado de los navegadores de Internet (Microsoft y Netscape), han intentado llevar dicho lenguaje a su propio territorio.

Originariamente Javascript fue creado por Netscape, y tenía el objetivo de ofrecer al programador la posibilidad de incluir cierta interacción en la parte “cliente”, como un tímido acercamiento a lo que sería DHTML. Mediante Javascript, el navegador Netscape Navigator era capaz de realizar pequeñas tareas como abrir nuevas ventanas de navegación, comprobar datos de formularios HTML, o sacar mensajes de aviso al usuario, pero

realmente sus prestaciones no se podían equiparar a las que tiene actualmente.

Acto seguido, Microsoft realizó su propia implementación del lenguaje para su navegador Internet Explorer, y la llamó JScript (en clara referencia al nombre de su competidor). JScript es realmente muy similar a Javascript, pero entre ambos navegadores existían (y existen) pequeñas diferencias que hacen la vida bastante incómoda al resto del mundo. Adicionalmente Microsoft sacó otro lenguaje de script paralelo a Javascript, basado fuertemente en Visual Basic, y que se llamó VBScript. Este lenguaje sigue actualmente en uso, aunque Netscape no ha realizado ninguna implementación del mismo en su navegador. Posteriormente, debido a los conflictos que estas dos versiones creaban en toda la comunidad de Internet, el grupo de estandarización ECMA creó el ECMAScript, que intenta dar un estándar final a dicho lenguaje de script.

Es por tanto este lenguaje el que integra el carácter dinámico en nuestras páginas, y el que se encarga de manejar los diversos elementos HTML y CSS para crear todo tipo de efectos. Javascript (en adelante nos referiremos a este lenguaje por este nombre) es un lenguaje de sintaxis muy similar a la del popular lenguaje Java, y por tanto similar a otros lenguajes como C o C++. Las diferencias obvias son la inexistencia de clases, herencia, polimorfismo y el resto de características o ventajas que se derivan de un lenguaje orientado a objetos. Javascript es básicamente un lenguaje de script (Script se viene traduciendo en múltiples publicaciones con el término “Guión”), orientado a eventos, lo que quiere decir que es un lenguaje que se interpreta y ejecuta en el mismo momento de la carga de la página, y que no necesita de una compilación previa como la mayoría de los lenguajes de programación.

El que un lenguaje de script esté orientado a eventos significa que básicamente lo vamos a utilizar como respuesta a eventos que se producen en el entorno en donde se ejecuta, en este caso en un navegador de Internet. Así por ejemplo un evento puede producirse, o “dispararse”, cuando el usuario pulsa un determinado botón, o cuando el ratón pasa por encima de cierta imagen. Estos eventos son los llamados eventos de usuario, y son los que se producen a partir de una



Figura 2. “En el sitio web de Netscape se encuentra toda la documentación sobre el lenguaje Javascript”

interacción del usuario con cualquier elemento de nuestra página Web o con el navegador mismo.

También existen otra serie de eventos que no dependen del usuario, y que podríamos denominar como eventos de sistema. Dichos eventos nos sirven para controlar diversos “momentos” importantes en nuestro medio, como el momento en que la página HTML solicitada por el navegador ha terminado de descargarse, o bien cuando se produce un evento de temporizador cada cierto intervalo de tiempo.

En estos casos, el sistema analiza nuestra página, y construye una lista de los eventos que tienen alguna función pendiente de ejecutarse. Como es lógico, podemos querer utilizar sólo unos pocos eventos de entre todos los que tenemos disponibles, y no estamos obligados a escribir código para todos y cada uno de ellos, ni siquiera a declararlos dentro de nuestra página.

Cuando el sistema captura un evento y encuentra una función específica dedicada al mismo, construye un objeto del tipo “Event” y lo pone a nuestra disposición. Dicho objeto viene con una serie de propiedades que podemos consultar en caso de que necesitemos especificar algún dato referente a dicho evento. Veamos el siguiente ejemplo:

```
<HTML>
<HEAD></HEAD>

<BODY>
<CENTER>
```

```
<FORM>
<INPUT TYPE="button" NAME="Pulsame" VALUE="Pulsame"
onClick="alert('Evento: ' + event.type + '(' +
event.screenX + ',' + event.screenY + ');')>
</FORM>
</CENTER>
</BODY>
</HTML>
```

En este ejemplo, vemos cómo hemos construido un simple formulario, con un botón que contiene el código que maneja el evento “onClick”, que se ejecuta siempre que se pulsa el botón en cuestión. En dicho código, mostramos un mensaje al usuario informándole de una propiedad en concreto (el tipo de evento) del objeto “event” que hemos recibido, y de las coordenadas de pantalla en donde el usuario ha hecho clic.

Llegados a este punto es necesario recordar que Javascript es un lenguaje sensible al uso de mayúsculas y minúsculas, por tanto, la propiedad “screenX” es reconocida como una propiedad válida del objeto “event”, no siendo así con “ScreenX” o “SCREENX”, que devolverán un valor indefinido (“undefined”).

## HTML DINÁMICO. COMENZANDO A TRABAJAR

Ya hemos visto los dos elementos básicos de lo que llamamos HTML dinámico, que son las Hojas de Estilo en Cascada (CSS) y el lenguaje Javascript.

Pero hasta ahora no hemos hecho nada que sea realmente espectacular, o por lo menos radicalmente distinto a lo que podemos llegar a hacer con simple HTML. Vamos a ver un sencillo ejemplo, y verá como las cosas comienzan a cambiar. La siguiente página muestra en principio un texto en pantalla. Hasta aquí todo normal y sin sorpresas. Pero enseguida nos daremos cuenta de que algo pasa (ver Figura 4)...

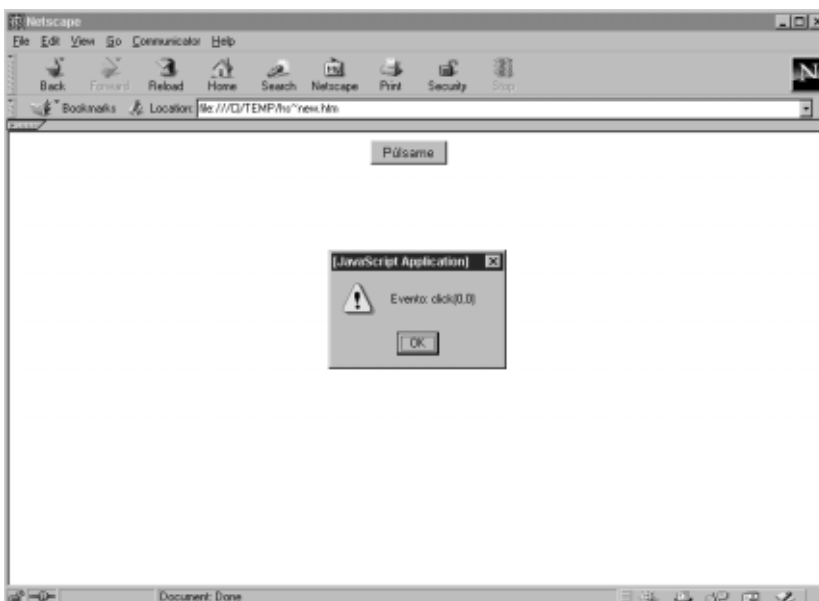


Figura 3. “Utilizamos Javascript para detectar cuándo el usuario ha realizado determinadas acciones”

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function Efecto0 {
    Mueveme.style.filter="Wave(Add=0,
Freq=10, lightStrength=10, Phase=0, Strength=4)";
}

function Efecto20 {
    Mueveme.style.filter="";
}
</SCRIPT>
</HEAD>

<BODY>
<DIV id="Mueveme" style="width:100; font-size=36pt;"
onMouseOver="Efecto0;"
onMouseOut="Efecto20;">
Hola. Estoy a punto de cambiar.
</DIV>
</BODY>
</HTML>
```

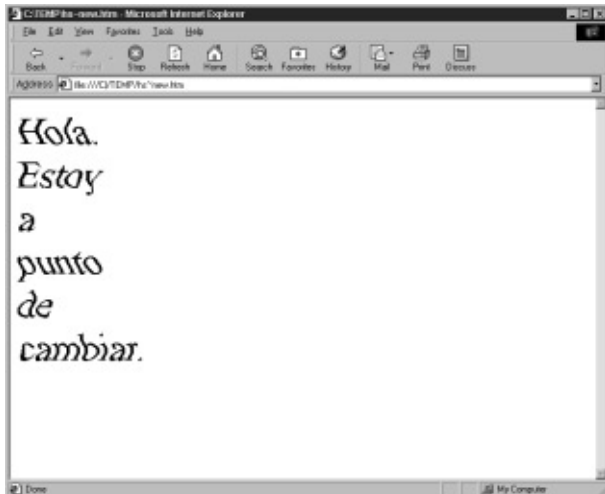


Figura 4. "Resultado de aplicar un efecto dinámico de Internet Explorer 4"

Este ejemplo funciona únicamente con Internet Explorer, y pese a que no es recomendable utilizarlo en sitios web que estén orientados al gran público (más de la mitad de los usuarios utilizan Netscape y por tanto no lo verían) lo incluimos por su vistosidad y sencillez. Este navegador implementa como vemos una propiedad llamada filter, que utiliza para aplicar un filtro sobre toda la capa definida por la etiqueta "DIV". Pruebe usted mismo a ejecutar este código cambiando el texto de saludo, e inclu-

so incluyendo alguna imagen dentro del Tag DIV. Los resultados que se pueden obtener con un poco de imaginación son bastante interesantes, y sobretodo muy diferentes de las posibilidades que teníamos a nuestra disposición con HTML.

El rango de efectos que podemos crear con DHTML es inmenso. Desde transacciones de texto, ocultar y mostrar elementos HTML, utilizar efectos de transformación, efectos de luces, mover objetos por delante y por detrás de otros, etc.

Veamos un nuevo ejemplo:

```
<HTML>
<HEAD></HEAD>

<BODY>
<P>
Este párrafo es un párrafo estático y no cambiará nunca.
</P>
<P id="P1" style="font-family: Arial; color:black;
background-color: white"
onMouseOver="Cambio10;"
onMouseOut="Cambio20;">
Este párrafo sin embargo cambiará en cuanto el ratón pase
por encima de él...
</P>

<SCRIPT language="Javascript">
function Cambio10 {
    P1.style.backgroundColor="black"
    P1.style.color="white"
    P1.style.fontFamily="Cursive"
}

function Cambio20 {
    P1.style.backgroundColor="white"
    P1.style.color="black"
    P1.style.fontFamily="Arial"
}
</SCRIPT>
</BODY>
</HTML>
```

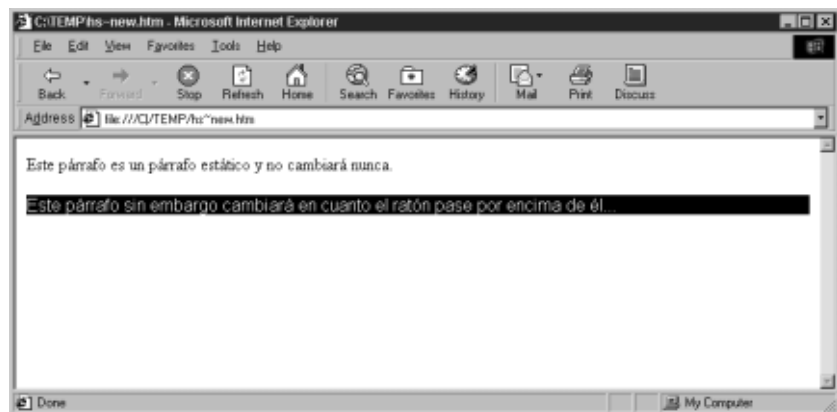


Figura 5. "Cambio dinámico del aspecto de un texto HTML"

Como se puede ver en la figura 5, esta página cambia uno de sus párrafos cuando el ratón pasa por encima de él (cuando se dispara el evento “onMouseOver”), y restaura el formato del mismo párrafo cuando el ratón abandona la zona del mismo (cuando se dispara el evento “onMouseOut”). Este ejemplo muestra cómo podemos acceder a las distintas características de un párrafo (color del texto, color del fondo, tipo de letra, etc.), y cambiarlas a nuestro antojo, sin tener que realizar en ninguno de los casos ninguna llamada al servidor.

## DHTML EN LA PRÁCTICA

Desde un punto de vista del día a día de una empresa de programación, tanto lo que llamamos DHTML, como el sistema CSS y Javascript abarcan dos grandes ámbitos de acción que no siempre están bien avenidos.

Uno es el mundo del diseño, y el otro el de la programación de aplicaciones de Intranet.

Para los diseñadores, DHTML es una magnífica oportunidad de añadir movimiento e interacción a sus diseños. Los sites que se pueden llegar a construir mediante DHTML son en ocasiones mucho más impacantes de lo que lo son los realizados con simple HTML. Hay que tener en cuenta que las posibilidades de movimiento e interacción pueden aplicarse a lo que es la misma navegación del site, y por tanto no sólo la presentación sino la manejabilidad del mismo se ven incrementadas en calidad.

Para este colectivo puede ser en ocasiones algo difícil el cambio, ya que tradicionalmente no hacía falta saber “programación” para diseñar páginas Web, y con un buen editor de imágenes y un editor de HTML se podían construir sitios web estupendos.

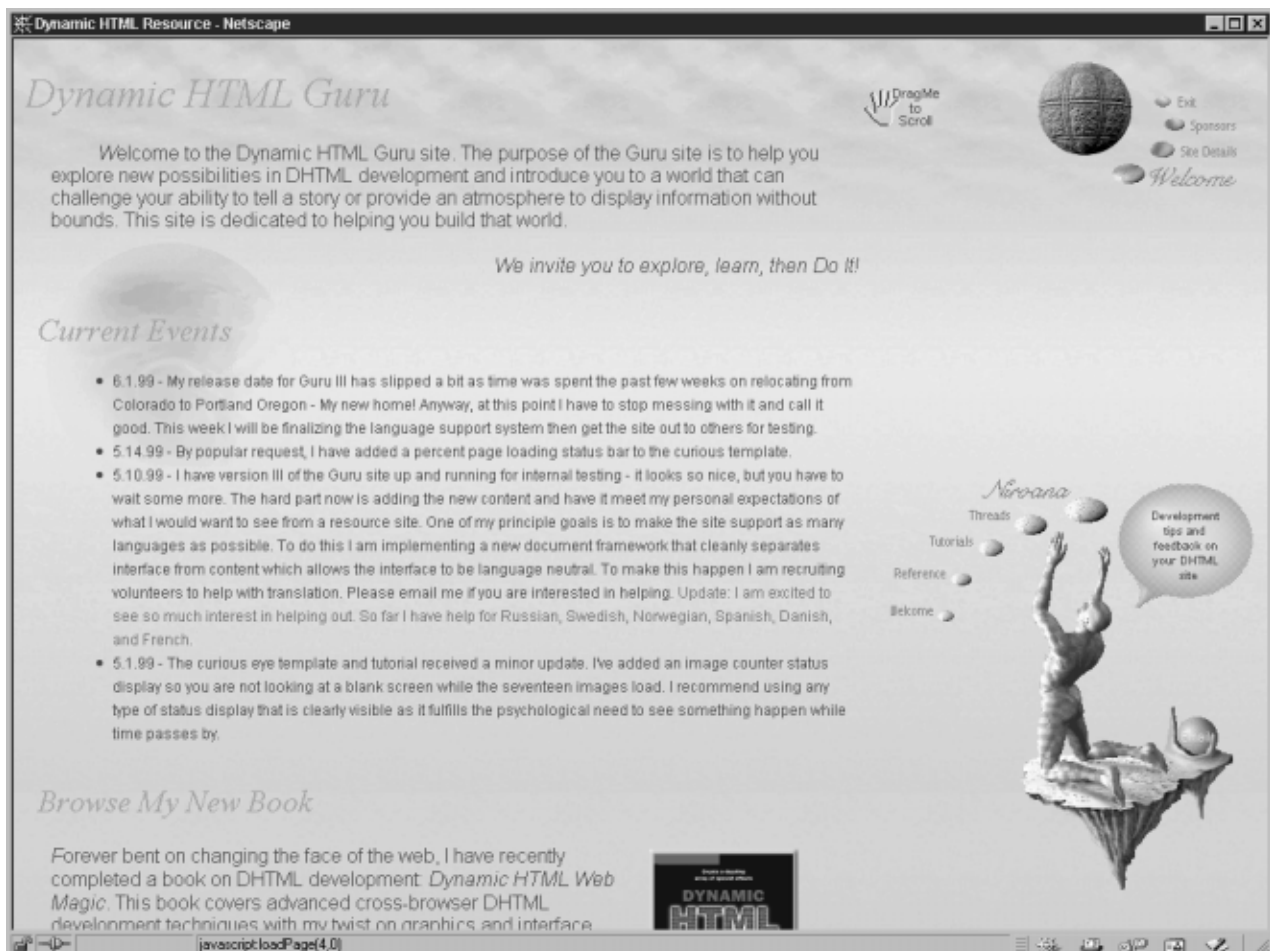


Figura 6. “En Internet se pueden encontrar múltiples páginas con tutoriales sobre DHTML”

Con la llegada de DHTML y Javascript los requerimientos suben, y ya no sólo hace falta tener buen gusto, sino que también se hacen necesarios ciertos conocimientos del lenguaje Javascript.

En estos casos, nuestro diseñador tiene dos opciones. La primera es aprender los rudimentos de la programación Javascript, que por otro lado es bastante sencilla y asequible. La segunda opción es la de utilizar alguna herramienta de autor para la creación de DHTML como la conocida "DreamWeaver" de Macromedia.

Estas herramientas permiten crear páginas HTML con CSS y diversos efectos Javascript de una manera totalmente visual y sin escribir ni una sola línea de código.

Estas herramientas están proliferando de manera espectacular, y seguramente todavía nos queden por ver bastantes maravillas en este terreno.

El otro colectivo al que nos referíamos, el de los programadores, pueden ver a DHTML de una forma totalmente opuesta. En este caso, la dificultad de aprender el lenguaje no es tal, ya que cualquier programador no debe tener ningún problema en aprender Javascript si es que no lo conoce ya.

Sin embargo, la duda que surge es si de verdad DHTML puede aportar algo a aplicaciones mas "serias", y en caso afirmativo, qué cosas aporta y qué dificultades añadidas comporta.

Una de las características de las aplicaciones de Intranet es el gran uso de acceso a base de datos que realizan. Por lo tanto, casi todas las páginas de una aplicación de Intranet hacen uso de su base de datos. En estos casos, DHTML puede ayudar no tanto desde la parte del diseño o de conseguir efectos visuales, sino más bien desde la parte de ayudar a la aplicación a tomar decisiones o seleccionar datos antes de tener que solicitar una nueva página.

Supongamos que estamos desarrollando una aplicación de Intranet para una inmobiliaria, en la cual cada uno de sus comerciales puede conectarse a la aplicación para solicitar datos sobre un inmueble, o bien para añadir inmuebles nuevos. Una de las aplicaciones más claras en este caso sería el de utilizar DHTML para comprobar los datos que el usuario introduce, descargando al servidor de esta tarea. Igualmente podríamos utilizar DHTML para construir una página dinámica que pudiese mostrar los datos de los inmuebles seleccionados descargando al servidor de la tarea de tener que "servir" cada página de cada inmueble de forma individual. Hay que tener en cuenta que el uso de DHTML en estos casos serviría para descargar al servidor de peticiones, lo cual sabemos que es uno de los puntos cruciales de cualquier aplicación o servidor, el número de peticiones simultáneas que puede atender sin mostrar problemas de rendimiento.

En definitiva, el uso y utilidad final de DHTML está aún en entredicho debido a la inseguridad que produce en la comunidad informática no disponer de un estándar claro, o mejor dicho, que este estándar esté disponible pero que no sea cumplido por las grandes empresas implicadas en esta área. Si usted está interesado en implantar el uso de DHTML en su empresa o equipo de trabajo, la mejor opción que puede tomar actualmente es la de decidirse por un navegador y desarrollar únicamente para dicha aplicación, sin preocuparse por los problemas de compatibilidad. En este caso, Internet Explorer le ofrece una implementación de DHTML muy superior a la de Netscape, aunque con el punto en contra de no utilizar características DHTML estándar, y por lo tanto no estar asegurada su compatibilidad futura con otros programas de la misma o de diferentes compañías. Antes de nada, analice sus necesidades, y evalúe qué le conviene mas. Esperemos que en breve se solucionen tantos problemas de compatibilidad, y volvamos a una Internet de estándares fiables.