

Técnicas avanzadas de diseño web

Elaborado por:

Equipo de tutores

EDITORIAL ELEARNING

ISBN: 978-84-17172-64-0

No está permitida la reproducción total o parcial de esta obra bajo cualquiera de sus formas gráficas o audiovisuales sin la autorización previa y por escrito de los titulares del depósito legal.

índice

Técnicas avanzadas de diseño web

UD1

Evolución del diseño web

1.1.	<i>La evolución del diseño de páginas web</i>	7
1.2.	<i>Diferencia entre una página estática y una dinámica</i>	11
1.3.	<i>Páginas web con conexión a bases de datos</i>	14
1.4.	<i>Objetivos de las páginas web</i>	15
1.4.1.	<i>Tipos de sitios web</i>	15
1.4.2.	<i>Introducción a la interfaces gráficas</i> <i>Características deseables de todo sitio web</i>	18
1.5.	<i>Estructura básica de una página web</i>	20
1.6.	<i>Introducción a los lenguajes básicos de diseño web</i>	22
1.6.1.	<i>HTML</i>	22
1.6.2.	<i>Introducción a las Hojas de Estilo o CSS</i>	23
1.6.3.	<i>Introducción a los lenguajes de script</i>	24
1.7.	<i>Presente y futuro de la tecnología web</i>	25
1.7.1.	<i>Presente: la web 2.0</i>	25
1.7.2.	<i>El futuro de la tecnología web: la web 3.0</i>	29
1.8.	<i>Introducción al desarrollo web. Software de diseño y edición</i>	30
	<i>Lo más importante</i>	33
	<i>Autoevaluación UD1</i>	35

UD2

Hojas de estilo (CSS)

2.1.	<i>Concepto de Hojas de Estilo o CSS</i>	39
2.2.	<i>¿Cómo se aplican las Hojas de Estilo?</i>	40
2.2.1.	<i>Selectores y declaraciones</i>	41
2.2.2.	<i>Las clases</i>	42
2.2.3.	<i>Parámetros para definir estilos</i>	44

2.3.	<i>La estructura de cajas</i>	46
2.4.	<i>Ejemplo práctico</i>	49
2.5.	<i>Templates</i>	51
	<i>Lo más importante</i>	53
	<i>Autoevaluación UD2</i>	55

UD3 *JavaScript*

3.1.	<i>¿Qué es JavaScript?</i>	59
3.2.	<i>El código JavaScript</i>	62
3.2.1.	<i>Sentencia y declaraciones</i>	62
3.2.2.	<i>Variables</i>	66
3.2.3.	<i>Objetos</i>	67
3.2.4.	<i>Operadores</i>	70
3.2.5.	<i>Arrays</i>	72
3.3.	<i>Ejemplo práctico: acceso a una página mediante contraseña</i>	73
	<i>Lo más importante</i>	77
	<i>Autoevaluación UD3</i>	79

UD4 *HTML dinámico o DHTML*

4.1.	<i>Introducción a DHTML</i>	83
4.2.	<i>Definición de capa (layer)</i>	89
4.3.	<i>Propiedades de las capas</i>	89
4.4.	<i>Ejemplos prácticos</i>	92
4.4.1.	<i>Efectos del atributo clip</i>	92
4.4.2.	<i>Mostrar/Ocultar un elemento</i>	94
	<i>Lo más importante</i>	95
	<i>Autoevaluación UD4</i>	97

UD5

XML

5.1.	<i>¿Cómo nace XML?</i>	103
5.2.	<i>¿Qué es XML?</i>	103
5.3.	<i>Sintaxis de XML</i>	105
5.4.	<i>Validación de XML</i>	106
5.5.	<i>Atributos y comentarios en XML</i>	107
5.6.	<i>Secciones CDATA e identificación del lenguaje</i>	108
5.7.	<i>XML en el servidor</i>	110
5.8.	<i>¿XML es el sustituto de HTML?</i>	112
	<i>Lo más importante</i>	113
	<i>Autoevaluación UD5</i>	115

UD6

Usabilidad y accesibilidad

6.1.	<i>Usabilidad</i>	121
6.1.1.	<i>¿Qué es la usabilidad?</i>	121
6.1.2.	<i>Importancia de la usabilidad</i>	121
6.1.3.	<i>Metodología para la usabilidad</i>	122
6.2.	<i>Accesibilidad</i>	125
6.2.1.	<i>¿Qué es y a quién va dirigida la accesibilidad web?</i>	125
6.2.2.	<i>¿Por qué un sitio web debe ser accesible?</i>	126
6.2.3.	<i>Recomendaciones WAI</i>	127
6.2.4.	<i>¿Qué son los Niveles WAI?</i>	128
6.2.5.	<i>¿Cómo construir con accesibilidad?</i>	129
6.2.6.	<i>Nuevas tendencias</i>	130
	<i>Lo más importante</i>	133
	<i>Autoevaluación UD6</i>	137

UD7

Desarrollo web avanzado

7.1.	<i>¿Qué es una aplicación web?</i>	141
7.2.	<i>Estructura de una aplicación web</i>	141
7.3.	<i>Lenguajes para el desarrollo de aplicaciones web.</i>	
	<i>Lenguajes del lado del servidor</i>	143
7.3.1.	<i>Lenguaje PHP</i>	144
7.3.2.	<i>Lenguaje JSP</i>	150
7.3.3.	<i>ASP/ASP.NET</i>	152
7.4.	<i>Introducción a AJAX</i>	153
7.5.	<i>Gestores de contenidos: joomla!, moodle, phpnuke,</i>	158
7.6.	<i>Ejemplos de aplicaciones web: CRM, e-commerce, Foros,</i>	162
	<i>Lo más importante</i>	163
	<i>Autoevaluación UD7</i>	165

UD8

Publicación y promoción de páginas web

8.1.	<i>¿Dónde publicar?</i>	169
8.2.	<i>¿Cómo transferir las páginas al servidor?</i>	171
	8.2.1. <i>Transferir archivos por FTP</i>	171
	8.2.2. <i>Acceso con formulario web</i>	172
8.3.	<i>Protección de directorios web</i>	173
8.4.	<i>Promoción del sitio web</i>	174
	8.4.1. <i>Publicidad y Banners</i>	175
	8.4.2. <i>Alta y posicionamiento en buscadores. Metatags</i>	177
8.5.	<i>Actualizar las páginas web</i>	182
	<i>Lo más importante</i>	185
	<i>Autoevaluación UD8</i>	187

UD1

Evolución del diseño web

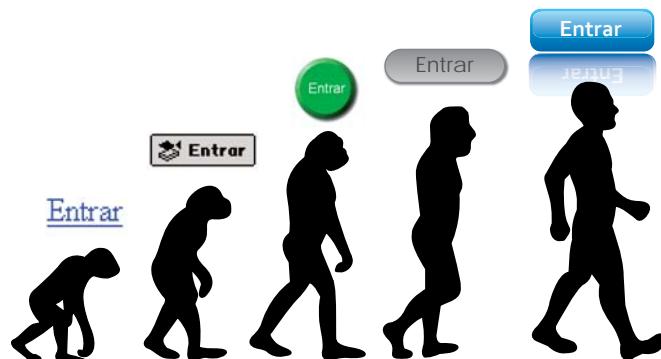
- 1.1. La evolución del diseño de páginas web
- 1.2. Diferencia entre una página estática y una dinámica
- 1.3. Páginas web con conexión a bases de datos
- 1.4. Objetivos de las páginas web
 - 1.4.1. Tipos de sitios web
 - 1.4.2. Introducción a la interfaces gráficas. Características deseables de todo sitio web
- 1.5. Estructura básica de una página web
- 1.6. Introducción a los lenguajes básicos de diseño web
 - 1.6.1. HTML
 - 1.6.2. Introducción a las Hojas de Estilo o CSS
 - 1.6.3. Introducción a los lenguajes de script
- 1.7. Presente y futuro de la tecnología web
 - 1.7.1. Presente: la web 2.0
 - 1.7.2. El futuro de la tecnología web: la web 3.0
- 1.8. Introducción al desarrollo web. Software de diseño y edición

InterNet

1.1. La evolución del diseño de páginas web

La evolución del diseño web a lo largo de estos años ha sido muy interesante. En 1993, el National Centre for Supercomputing desarrolló el primer navegador libre llamado "Mosaic". Al año ya había unos dos millones de usuarios de este programa. Este navegador era capaz de mostrar texto y gráficos, pero tenía muchas limitaciones. No obstante, era suficiente para las páginas web de aquella fecha, las cuales eran muy lineales y funcionales para los científicos que querían compartir información. Poco tiempo más tarde aparecieron Internet Explorer y Netscape Navigator. La tecnología de estos primeros navegadores limitaba la capacidad de mostrar información gráfica necesaria para la comunicación visual. Ésta y otras restricciones influenciaron mucho al diseño de páginas web, que tenía que vérselas con problemas como las conexiones lentas con módem, los monitores monocromo y la incapacidad de los proveedores de Internet de transferir datos rápidamente.

Las páginas web de los primeros tiempos de vida de Internet eran tan simples como un documento de texto o un simple folleto en el que el desarrollo más complicado consistía en usar efectos en las fuentes y colores. El diseño solía realizarse de arriba a abajo y de izquierda a derecha, intercalando secuencias de imágenes y textos, con muchos saltos de línea y otros elementos separadores como las barras horizontales y las viñetas de lista. Se construían en lenguaje HTML. No obstante y aunque este tipo de páginas generalmente funcionaba bien, los programadores se encontraron con problemas como la superposición de colores o efectos de textos demasiado resaltados y sobredimensionados.



Para hacernos una idea, en la siguiente URL se encuentra una herramienta con la cual se puede observar el aspecto de cualquier página web (en caso de estar disponibles) desde su nacimiento, es decir, las diferentes actualizaciones que haya sufrido. Tan sólo hay que introducir la dirección de la página en el campo de texto:

<http://www.archive.org/web/web.php>

Con el paso del tiempo, el desarrollo del comercio electrónico y las necesidades comunicativas, las páginas web comenzaron a tener estructuras más complejas

y otros lenguajes de programación y metodologías entraron en juego, como JavaScript, que permitía crear pequeños programas muy útiles en la web. La gente empezó a darse cuenta del poder de la world wide web y de los sitios, surgiendo así el “la burbuja de Internet” acompañada del convencimiento de los desarrolladores de que había que cambiar la apariencia de las páginas por completo, hacerlas visualmente convincentes con el fin de que fueran más atractivas a los ojos del usuario.

En 1994 fue creado el consorcio W3C para establecer objetivos y estándares en el desarrollo del lenguaje HTML. Desde entonces, este lenguaje ha tenido varias versiones hasta la actual 5.0. Desde la primera, W3C y Nestcape han trabajado para alcanzar mejores prestaciones capaces de ofrecer contenido dinámico a través de la red.

Por aquel entonces surgió una nueva generación de páginas web cuyas principales características de diseño eran los iconos que reemplazaban a palabras, fondos de pantalla que se formaban a partir de mosaicos, botones con bisel y relieve, banners que sustituían a las cabeceras y el uso de desplegables. Por otra parte, estos diseños tendían a ser muy recargados con tecnologías que no respetaban el objetivo de la página. Introducían muchos iconos llamativos, gifs animados y colores primarios. Pero aún así, había diseñadores que conseguían utilizar el código HTML de maneras innovadoras consiguiendo un efecto más deseado. Así surgió la estructura de tablas cuando introdujeron la etiqueta <table>, más pensada para introducir datos estadísticos que para el propio diseño. Los desarrolladores empezaron a utilizar el código de las tablas para colocar en sus columnas el texto y los gráficos tal y como lo hacen los maquetadores de las revistas. De esta manera se conseguía un mejor control del posicionamiento de los elementos y se solucionaban muchos problemas estéticos.

Estas páginas aún se estaban adaptando a una tecnología constantemente cambiante, como por ejemplo, el paso de los monitores de 640x480 píxeles a los de 800x600, lo cual era una pesadilla para los diseñadores, que además, necesitaban saber si los monitores de los usuarios eran de 8 o de 24 bits para tratar la profundidad de color de los gráficos.

Otro problema era que Microsoft y Nescapte tenían cada uno sus propias reglas en sus navegadores, con lo que una página web no se veía igual en Internet Explorer que en Netscape Navigator. Durante estos años, Microsoft y Nestcape han estado compitiendo para quedarse con el dominio del mercado de navegadores y de tecnología web, pero gracias a esta competencia han surgido muchos avances en la tecnología web.

Más tarde llegó un período en el que los desarrolladores habían aprendido de todos estos errores del pasado y comenzaron a realizar diseños agradables que permitían al visitante focalizar su atención en los principales productos y servicios que ofertaban las páginas. Los websites llegaron a convertirse en el “punto de

información” de los negocios o de las empresas. Pero en este momento, los usuarios ya estaban alcanzando un nivel alto de su curva de experiencia de navegación en el que empezaron a detectar las páginas de mala calidad que simplemente servían para mostrar información de poca utilidad.

Observando este problema, la mayoría de los desarrolladores se adentraron en la interactividad, el lenguaje DHTML (HTML dinámico) y los clips Flash para impresionar a los visitantes. Estructurar los distintos elementos de la página utilizando tablas no estaba mal, pero era un método muy limitado, así que tenían que solventarlo escribiendo gran cantidad de código e implementando estilos, lo cual era un trabajo duro y complicado. Para deshacerse de este problema, introdujeron el lenguaje CSS (hojas de estilo en cascada), el cual se hizo muy popular en poco tiempo entre los desarrolladores. Lo más importante de este lenguaje es que no está basado en tablas, sino que introduce la etiqueta <div> para asignar los estilos.

Estas páginas tenían como objetivo atraer a la audiencia y la comunicación con ella en relación con el propósito del sitio. Este propósito podría ser informar, recabar información o vender en el espacio de tiempo más corto posible. Para ello, había que dar mayor importancia al diseño del proyecto para crear una página que hiciera que el usuario pudiera ser consciente instantáneamente de a quién pertenece la página, qué ofrece, dónde encontrarlo y por qué el sitio existe. El usuario podía saber qué se ofrecía en la página y además cómo conseguirlo usando una navegación intuitiva.

Esta filosofía tendía a ser similar a la de las estrategias de marketing, es decir, la atracción de audiencia en una fracción de segundo, la retención de la atención, la inducción al deseo de informarse y la llamada a la acción. Los especialistas en marketing empezaron a “subirse al carro”, con lo cual se generó una enorme oferta de servicios de campañas publicitarias en la red. El diseño de una web de esta generación solía tener una página “splash” o una “intro” en flash para atraer la atención, una página “túnel” para guiar al usuario hacia una única oferta y una página “núcleo” que proveía de información sobre el total del contenido del sitio, además de un sistema de navegación que mostraba cómo obtener qué y dónde (en esta sección era donde se incitaba al deseo de conocer más). También solían tener una página de salida que fomentaba la llamada a la acción, como un formulario electrónico que había que llenar para participar en una oferta.

En este punto los programadores habían alcanzado un nivel de madurez en el uso de la tecnología muy altos. Aquí surgieron la optimización de las páginas para ser encontradas y posicionadas por los motores de búsqueda, y los layouts, la última tendencia en el campo del diseño web de esos días. Los layouts permiten estructurar los elementos de la página y se utilizan para dirigir la atención del usuario al principal contenido de ésta, no del diseño. Motivados por esta nueva técnica, empezó a mantenerse el fondo (background) con un diseño muy simple y a dejar la parte exterior del layout en blanco para evitar la distracción.

Una vez que se fue estandarizando la comunicación entre las distintas plataformas (PC, Mainframe, Mac, etc.), también empezaron a proliferar lenguajes de programación avanzados que permitían prestaciones de las que hasta entonces no disponían las páginas web. Se trata de PHP, C# y la tecnología ASP entre otros. Esto permitía acciones sumamente importantes como conectar la página con una base de datos, algo primordial en las tiendas online para crear catálogos de productos, por ejemplo.

El diseño web comenzó a centrarse en la integración de los elementos multimedia como el sonido, los clips de video, la animación, las bases de datos, los mundos 3D y el vrml (virtual reality modeling language). Pero los diseñadores eran conscientes de que el principal obstáculo de la utilización de estos elementos era la velocidad de descarga. Los navegadores, principalmente Netscape y Microsoft Explorer, ya estaban incrementando su tecnología en relación con el contenido multimedia debido a la demanda de éste.

El objetivo del diseño web de esta generación consistía en crear un sistema de navegación que permitiera a los usuarios encontrar lo que estaban buscando lo más rápido posible, lo cual se hace muy necesario cuando el sitio contiene grandes cantidades de información y de contenidos. El usuario debía, por ejemplo, poder reconocer para qué sirvía un formulario visualizando simplemente el botón de envío del mismo. Es por tanto que los diseñadores debieron darse cuenta de que tenían la responsabilidad de presentar el contenido y navegación necesarios para el propósito del sitio. Este trabajo era similar a crear una imagen corporativa a imprimir; qué color y qué símbolos son los apropiados para comunicar visualmente la misión o aquello que el cliente esté mirando sea capaz de lograrlo.

En definitiva, las características de esta generación de páginas web eran:

- ↶ El uso apropiado del lenguaje HTML utilizando solamente sus etiquetas estructurales en el documento y el uso de CSS.
- ↶ Respeto a la usabilidad.
- ↶ Motores de búsqueda dentro de la página para encontrar contenidos de la misma.
- ↶ Facilidad de actualización.
- ↶ Automaticidad a la hora de modificar el aspecto de la página enlazando el código a archivos CSS.
- ↶ Seguimiento de los estándares.

Durante el primer lustro de este siglo, las oportunidades surgían en la red fácilmente para aquellos que supieron reconocerlas. La tecnología ya permitía desarrollar estrategias innovadoras que, aunque en principio podían parecer ideas triviales, resultaron ser la lanzadera de muchas compañías que supieron cómo utilizarlas. Sin embargo, el cambio más notable que experimentaron las páginas web de principios de siglo no se refiere expresamente a la tecnología, aunque esté

íntimamente relacionado con ella. De repente y gracias al desarrollo de ciertas herramientas, los usuarios se dieron cuenta de que podían hacer muchas más cosas en Internet que simplemente mirar las páginas, también podían participar. Empezaron a aparecer los blogs, las redes sociales, los sistemas de mensajería cada vez eran más sofisticados y cualquier usuario sin mucha experiencia en informática ya podía subir su propia página web. Nació así la web 2.0., cuyos patrones de diseño se caracterizan porque el usuario puede manejar los aspectos visuales y las metáforas de presentación con tecnologías como AJAX y con un conjunto de lenguajes y técnicas de programación que no eran novedosos pero que, al ser combinados, facilitan el desarrollo de interfaces ricas y adaptables. El diseño de las páginas se vuelve elástico.

En el futuro, el diseñador profesional de páginas web no tendrá que preocuparse tanto por el aspecto de la página como por las prestaciones que ésta ofrezca. Al igual que ocurrió con el diseño industrial, la publicidad, la fabricación de coches, el video VHS y muchas otras más, poco a poco se tenderá a hacer las cosas más sencillas, más fáciles de usar y más adaptadas a las necesidades del usuario final. Nos adentramos en una era en la que la tecnología, incluidas las soluciones interactivas como las páginas web, se va a adaptar al usuario y no al revés. Esto supone un cambio fundamental en la metodología de diseño utilizada. En el sector se llama “User-Centered Design (UCD)” o diseño centrado en el usuario, que consiste en conducir todos los esfuerzos a satisfacer los intereses del destinatario final al que va dirigido el producto del esfuerzo creativo.

Es preciso considerar que vivimos en una era en la que la información es poder y está cada vez más en manos del usuario.

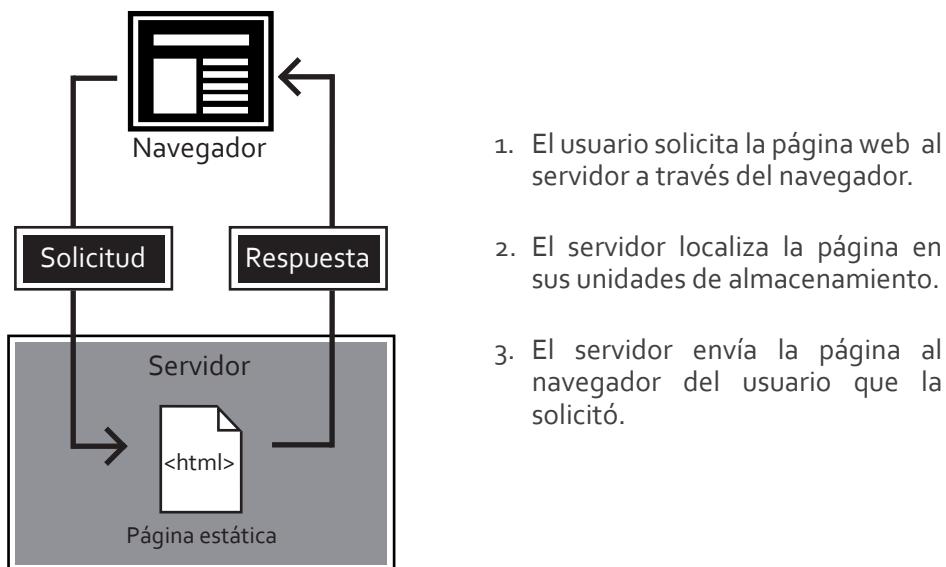
1.2. Diferencia entre una página estática y una dinámica

Una página web es una fuente de información adaptada para la World Wide Web que se visualiza a través de un navegador de Internet o cliente web. Está compuesta por multitud de archivos que pueden ser de imagen, de video y por supuesto los archivos del código fuente. Éste debe estar escrito como mínimo en lenguaje HTML y debe indicar en su código dónde colocar cada elemento, texto, imagen o video, es decir, la estructura de la página.

En la web se pueden crear dos tipos de páginas: estáticas y dinámicas. Las páginas estáticas son aquellas que se construyen con lenguaje HTML, pueden presentar textos estáticos acompañados de imágenes o archivos multimedia (sonido o video) y a parte de los enlaces y el envío de formularios, no permiten interactividad con el usuario. Son fáciles de crear, aunque ofrecen pocas ventajas a los visitantes.

Un sitio web estático está formado por un conjunto de páginas y de archivos HTML relacionados entre sí y que están alojados en un servidor web.

El servidor web suministra las páginas web como respuesta a las peticiones de los navegadores, las cuales se generan cuando el usuario hace clic en un enlace, elige un marcador en un navegador o introduce una URL en la barra de direcciones del navegador. El contenido final de una página web estática lo decide el diseñador y éste no varía cuando se solicita la página. El diseñador escribe las líneas de código HTML de la página antes de subirla al servidor. El código HTML no cambia una vez colocado en el servidor y por ello, este tipo de páginas se denomina página estática.

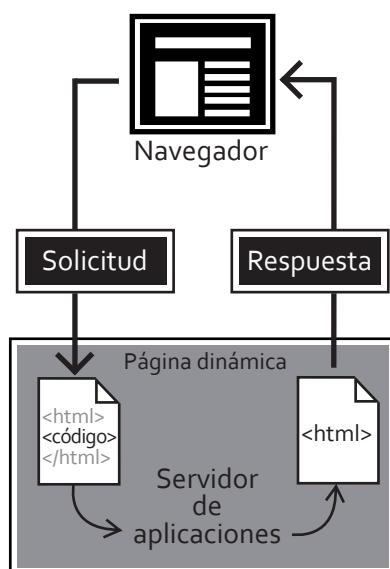


Las páginas dinámicas contienen elementos que permiten al usuario interactuar con ellas, o bien, presentan efectos especiales. Para crearlas, aparte de HTML, se necesitará otro tipo de lenguajes que se describirán más adelante. Sin embargo no hay que olvidar la importancia del HTML, ya que éste es la base del desarrollo web. Cuando se construye una página dinámica, el código correspondiente a los demás lenguajes (código que permite crear los efectos especiales o interactividad) de programación se incluye embebido dentro del HTML y se suele denominar “script”, aunque no todo programa interpretado es considerado un script.

Las páginas dinámicas pueden ejecutarse en el cliente (navegador) o en el servidor. Cuando una página HTML contiene script de cliente, el navegador se encarga de interpretarlo y ejecutarlo. Que se ejecuten en uno u otro dependerá del uso que se le vaya a dar a la página. Por ejemplo, usos típicos de las páginas de cliente serían: efectos especiales para webs como rollovers o control de ventanas, cálculos, presentaciones en las que el texto se desplace por la página, etc. Las páginas dinámicas de cliente se escriben básicamente en dos lenguajes de programación: Javascript y VBScript, que se verán más adelante.

Este tipo de páginas tienen la desventaja de que son muy dependientes del navegador en el que se ejecutan y teniendo en cuenta que cada uno tiene sus propias características, no todas las páginas se visualizarán igual en todos los navegadores. Como ventajas se pueden citar que al ejecutarse en el navegador descargan de trabajo al servidor, que su respuesta ante la petición de los usuarios es muy rápida y que permiten la utilización de recursos de la máquina local.

Las páginas dinámicas de servidor son ejecutadas por el servidor. El archivo HTML de la página web llega al servidor y éste ejecuta los script contenidos en él devolviendo al cliente sólo código HTML. Las páginas dinámicas de servidor se usan en aplicaciones web tales como: foros, chats, ... y más concretamente en aquellas aplicaciones que requieren el acceso a una base de datos situada en el servidor, como por ejemplo un banco.



1. El usuario solicita la página web dinámica al servidor a través del navegador.
2. El servidor localiza la página en sus unidades de almacenamiento y la envía al servidor de aplicaciones.
3. El servidor de aplicaciones busca las instrucciones en la página y construye el resultado final.
4. El servidor de aplicaciones envía la página final al navegador del usuario.

Las principales tecnologías utilizadas del lado del servidor son: ASP, JSP, CGI, PHP.

↳ Ventajas de las páginas dinámicas de servidor:

- El cliente no puede ver los script.
- Los script son independientes del navegador del usuario.

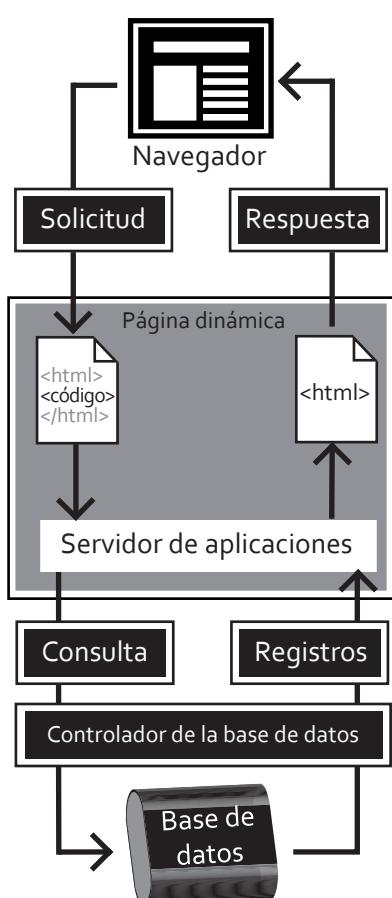
↳ Desventajas:

- Necesidad de disponer de un servidor más potente y con más capacidades que el necesario para las páginas dinámicas de cliente.

- Debido al alto tiempo de procesamiento individual, estos servidores soportarán menos usuarios concurrentes.

1.3. Páginas web con conexión a bases de datos

Actualmente resulta indispensable, sobre todo para las páginas web relacionadas con el comercio electrónico, conectar la interfaz con una base de datos para mostrar contenidos que se puedan someter a diferentes criterios de búsqueda, es el caso de los catálogos de las tiendas online.



Para ello es necesario que la página esté alojada en un servidor de aplicaciones que permita trabajar con recursos del lado del servidor, en este caso las bases de datos. Por ejemplo, una página dinámica puede ordenar al servidor de aplicaciones que filtre datos de una base de datos y los inserte en el código HTML de la página.

El hecho de introducir diferentes criterios de búsqueda en una base de datos y solicitar que los muestre, es lo que se suele llamar “consulta”. Estos criterios de búsqueda están definidos en un lenguaje de programación para bases de datos denominado SQL (Structured Query Language, lenguaje de consulta estructurado). La consulta SQL se ejecuta en los scripts o etiquetas del lado del servidor de la página.

Un servidor de aplicaciones no se puede comunicar directamente con una base de datos porque sus formatos no son compatibles. El servidor de aplicaciones sólo se puede comunicar con la base de datos por medio de un controlador que actúe de intermediario, un software que actúa como un intérprete entre el servidor de aplicaciones y la base de datos.

Cuando el controlador establece la comunicación, la consulta se ejecuta en la base de datos y se origina un conjunto de registros. Un conjunto de registros es un grupo de datos extraídos de una o varias tablas de una base de datos. El conjunto de registros se devuelve al servidor de aplicaciones, que los emplea para mostrarlos en la página. Algunos ejemplos de consulta de base de datos sencilla escrita en SQL serían:

- ↳ Consulta para mostrar todos los campos de una tabla: `SELECT * FROM NOMBRE_TABLA`
- ↳ Consulta para mostrar el valor máximo de un campo de la tabla: `SELECT MAX (CAMPO_VALOR) FROM TABLA_TOTALES`
- ↳ Consulta para mostrar, por ejemplo, el nombre y el teléfono de la tabla clientes: `SELECT Nombre, Teléfono FROM Clientes`

Se puede utilizar prácticamente cualquier base de datos con su aplicación web, siempre y cuando se haya instalado el controlador de base de datos correcto en el servidor.

Si el objetivo es desarrollar pequeñas aplicaciones en la página, se puede utilizar una base de datos basada en archivos como las de Microsoft Access. En cambio, si el objetivo es desarrollar aplicaciones empresariales críticas, lo aconsejable es utilizar una base de datos basada en servidor, como las que permite crear Microsoft SQL Server, Oracle Data Base o MySQL. Si la base de datos está guardada en un sistema distinto del servidor Web, hay que asegurarse de disponer de una conexión rápida entre ambos sistemas para que la aplicación web pueda funcionar de forma rápida y eficiente.

1.4. Objetivos de las páginas web

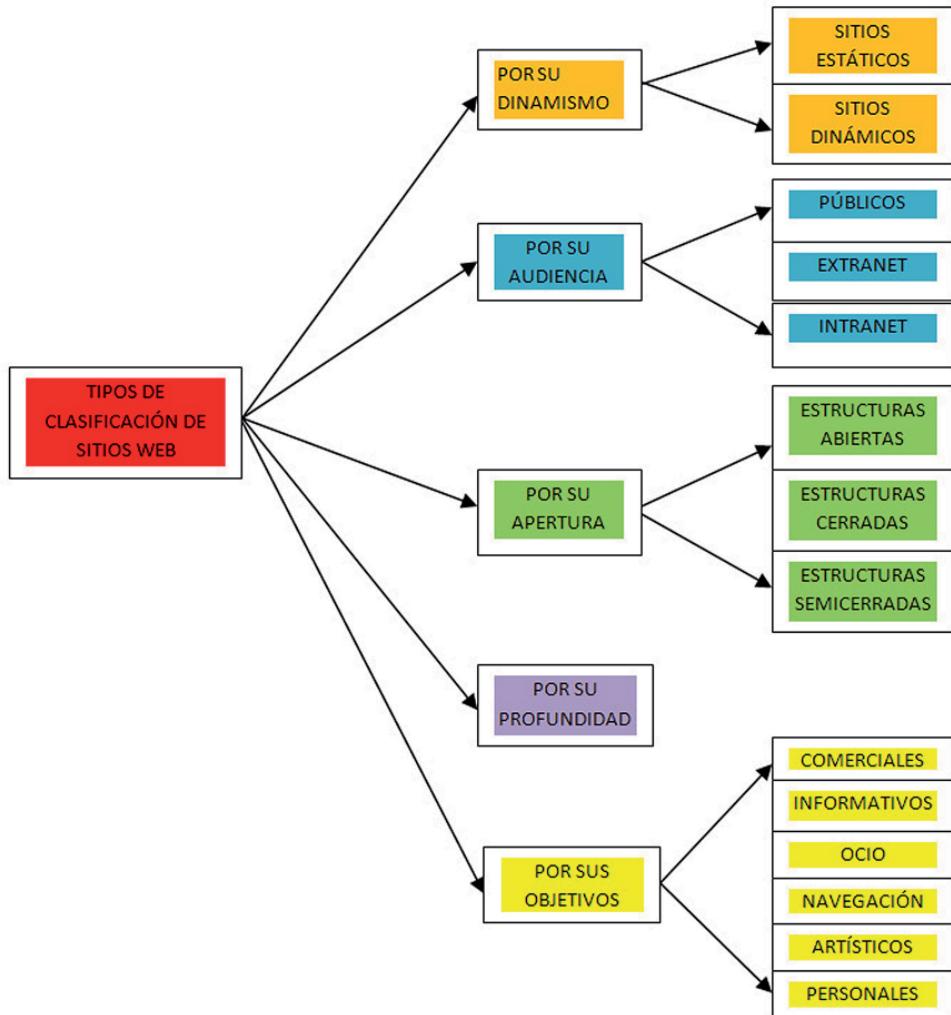
1.4.1. Tipos de sitios web

Recuerda

Un **sitio web** o **website** es un conjunto de páginas web (archivos) comunes a un dominio o subdominio de Internet.

Es importante, no confundir sitio web con página web, ya que ésta última es sólo un archivo y forma parte de un sitio web. Cuando se solicita una página en el navegador insertando su dirección (URL), por ejemplo: www.grupovertice.com, en realidad se está haciendo referencia a un sitio web, que tiene una página inicial y que es lo primero que se visualiza, pudiendo existir en ella hipervínculos al resto de páginas del sitio.

Existe una amplia gama de sitios web clasificados según su contenido o el uso para el que haya sido creado. El siguiente esquema muestra los tipos de clasificación de sitios web:



- ↪ **Sitios estáticos:** como se explicó anteriormente, los usuarios no pueden modificar el texto de la página, encargándose exclusivamente los diseñadores de su contenido.
- ↪ **Sitios dinámicos:** permiten interactividad con los usuarios, personalizando éstos la información que se les ofrece.
- ↪ **Sitios públicos:** se tratan de sitios web dirigidos al público en general, sin restricciones de acceso en principio.

- ↪ **Extranet:** este tipo de sitios limitan el tipo de usuarios que pueden acceder, por ejemplo los proveedores de una empresa o los clientes.
- ↪ **Intranet:** son sitios de acceso restringido a una empresa u organización. Normalmente suelen publicarse en redes privadas.
- ↪ **Estructura abierta:** los usuarios pueden acceder a cualquier página del sitio web, ya que todas ellas disponen de su dirección.
- ↪ **Estructura cerrada:** son sitios en los que los usuarios no pueden acceder a todas las páginas que lo forman. Por ejemplo, las páginas que solicitan un registro para acceder a ellas.
- ↪ **Estructura semicerrada:** los usuarios tienen que pasar por puntos específicos para poder acceder a las secciones más importantes.
- ↪ **Por su profundidad:** este tipo de clasificación está basada en el número de enlaces que el usuario debe pulsar para llegar al contenido. Se aconseja que el número de enlaces no sea mayor de tres.
- ↪ **Sitios comerciales:** tienen una finalidad económica. Los usuarios que visitan este tipo de sitios suelen ser clientes, empleados, inversores, los sectores de la competencia, etc. Los sitios comerciales se dividen a su vez en:
 - Corporativos: informan sobre la empresa.
 - Promocionales: informan sobre los productos.
- ↪ **Sitios informativos:** distribuyen información a los usuarios que dependerán del tipo de contenido que ofrezcan.
- ↪ **Sitios de ocio:** este tipo de sitios son un caso especial, suelen seguir reglas propias en su diseño. Su finalidad suele ser económica.
- ↪ **Sitios de navegación:** ayudan al usuario a encontrar lo que busca en internet. Dentro de este tipo de sitios se encuentran los portales, sitios web que proporcionan un punto de inicio, entrada o portal a otros recursos en Internet o una intranet.
- ↪ **Sitios artísticos:** muestran la expresión artística del diseñador y las únicas normas por las que se rige son las que el propio diseñador o diseñadores dicten.

- ↪ **Sitios personales:** mantenido por una persona o un pequeño grupo (como por ejemplo familia), contiene información o cualquier contenido que la persona quiera incluir. En este grupo se incluyen los blogs.

Antes de diseñar un sitio web es muy importante encuadrar, entre los objetivos del mismo, el tipo de website del que se va a tratar, puesto que esto nos ayudará a poner en práctica estrategias y metodologías propias de cada tipo que ayudarán a posicionar la página y a diseñar su estructura adecuadamente.

1.4.2. Introducción a la interfaces gráficas. Características deseables de todo sitio web

El diseño de la interfaz de un sitio web necesita ser capaz de comunicar claramente lo que éste ofrece y cómo los usuarios pueden hacer uso de él. Diseñar websites efectivos requiere tener en cuenta consideraciones técnicas (tiempo de descarga y accesibilidad), estructurales (organización y flujo de trabajo) y visuales (personalización).

Los elementos que forman una interfaz gráfica de usuario (GUI) son objetos que se configuran a gusto del diseñador, que van a responder sobre determinados eventos que pueden ocurrir durante la ejecución de la aplicación y deben ser programados por el equipo de desarrollo. Ejemplos de estos objetos son:

- ↪ Contenedores (Panel, Form, etc.).
- ↪ Botones (Button).
- ↪ Cajas de texto (TextField).
- ↪ Casillas de verificación (CheckBox).
- ↪ Etiquetas (Label).
- ↪ Imágenes (Image).
- ↪ Listas desplegables (ComboBox).
- ↪ Etc.

En el siguiente enlace se puede encontrar información para profundizar sobre las aplicaciones gráficas en Java y las posibilidades que se presentan desde el punto de vista del diseño de interfaces: