

# Introducción a las aplicaciones web

Vaciado de respuestas en Google Docs

**1 Describe el esquema de funcionamiento de un servicio web. Componentes involucrados. Inserta una imagen que lo describa.**

Una visión simplificada del uso de la web se muestra en el siguiente diagrama:

- Un cliente HTTP como un navegador web utiliza el modelo cliente-servidor. El cliente abre una conexión y envía una petición HTTP a un servidor HTTP,
- El mensaje de respuesta HTTP del servidor o bien por lo general contiene un mensaje de error o el documento solicitado o recurso.
- Después, el servidor cierra la conexión, por lo tanto, se trata de una conexión **sin estado** en que se cierra cada solicitud después de su finalización. Ver el tutorial HTTP para los ejemplos.

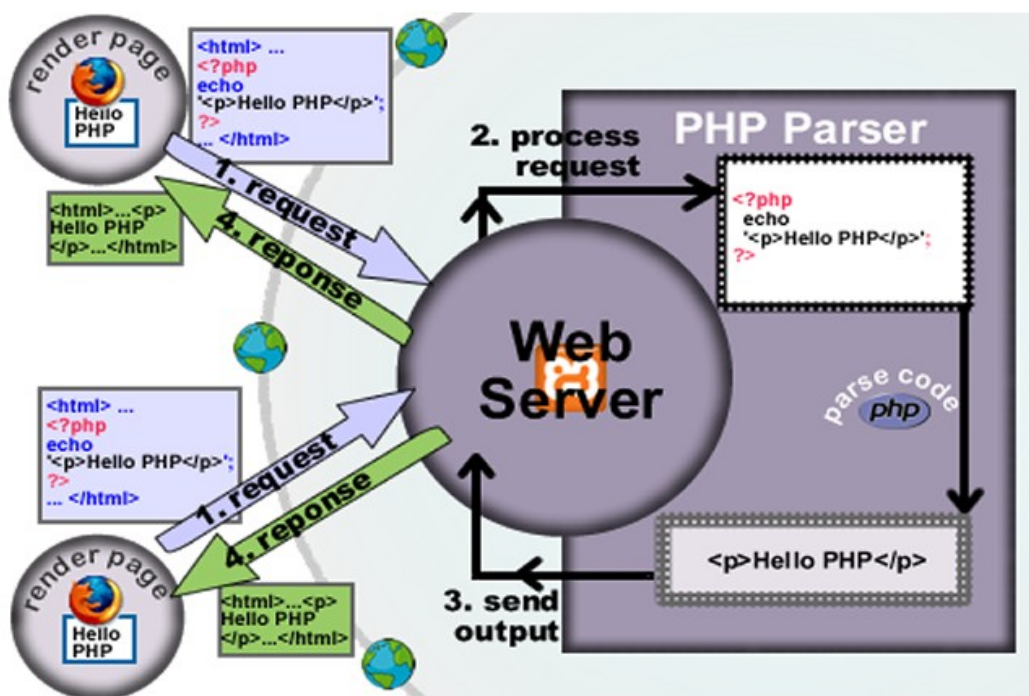


Esto implica que un cliente (por ejemplo, un programa navegador) envía una solicitud para un programa de servidor para un documento particular o de recursos.

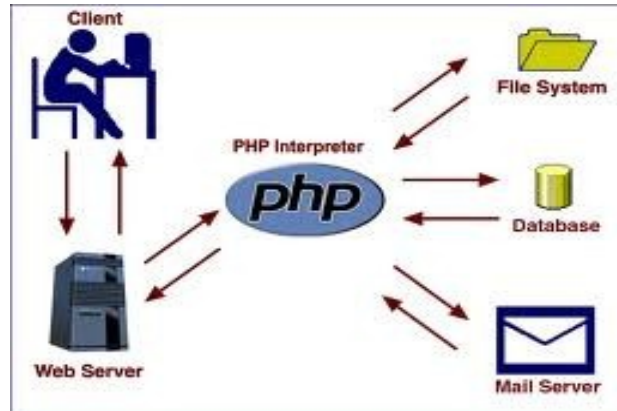
**Webs estáticas y dinámicas:** Si el sitio web sólo almacena archivos, entonces no se necesita programa lado del servidor, esto se conoce como un **sitio web estático**. Sin embargo, si queremos que el servidor responda de forma **dinámica** en función de los parámetros enviados en la solicitud, entonces tenemos que tener un programa en el servidor que es capaz de analizar la solicitud y responder en consecuencia.

De forma más detallada, si el contenido es dinámico y se emplea un lenguaje de script de servidor y además se accede a una base de datos, el funcionamiento es el siguiente:

- El navegador del cliente solicita el documento que requiere ejecución de código en el servidor (por ejemplo en PHP)
- Llega la solicitud del servidor y el servidor localiza el documento, lanza el intérprete del lenguaje y ejecuta todo su código.
- Si dentro del script se accede a una base de datos para consultar o almacenar datos, el interprete del lenguaje de programación establece la conexión con un gestor de base de datos y ejecuta la consulta
- Una vez ejecutado el código se genera el resultado en HTML y lo devuelve al servidor para que lo transfiera al cliente.
- El servidor transfiere el resultado en HTML y es mostrado en el navegador del cliente.



*Esquema de funcionamiento de cliente que accede a servidor y se solicita documento en PHP*



*Esquema funcionamiento de cliente que accede a servidor, se solicita documento en PHP y este conecta con diferentes servicios de servidor (base de datos, correo, sistema de archivos)*

## 2 ¿Qué protocolo de comunicaciones está involucrado en el tráfico web? ¿Qué tipos de mensaje intercambian cliente y servidor?

HTTP (Hyper Text Transport Protocol) es el protocolo utilizado en Internet para la comunicación entre un cliente y un servidor web.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las [cookies](#), que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.

Una [transacción](#) HTTP está formada por un encabezado seguido, opcionalmente, por una línea en blanco y algún dato. El encabezado especificará cosas como la acción requerida del servidor, o el tipo de dato retornado, o el código de estado.

Veamos un ejemplo de un diálogo HTTP

Para obtener un recurso con el [URL](#) `http://www.example.com/index.html`

- Se abre una conexión al host `www.example.com`, puerto 80 que es el puerto por defecto para HTTP.
- Se envía un mensaje en el estilo siguiente:

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: nombre-cliente
[Línea en blanco]
```

- La respuesta del servidor está formada por encabezados seguidos del recurso solicitado, en el caso de una página web:

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221

<html>
<body>
<h1>Página principal de tuHost</h1>
(Contenido)
.
.
.
</body>
</html>
```

**Nota:** Un **localizador de recursos uniforme**, más comúnmente denominado **URL** (sigla en inglés de *uniform resource locator*), es una secuencia de caracteres que se usa para nombrar recursos en Internet para su localización o identificación

El formato general de un URL es:

*esquema://máquina/directorio/archivo*

También pueden añadirse otros datos:

*esquema://usuario:contraseña@máquina:puerto/directorio/archivo*

Por ejemplo: *http://www.iesharia.es/*

### 3 Describe como funciona un ataque distribuido de denegación (DdoS) de servicio por inundación Syn (Syn Flood)

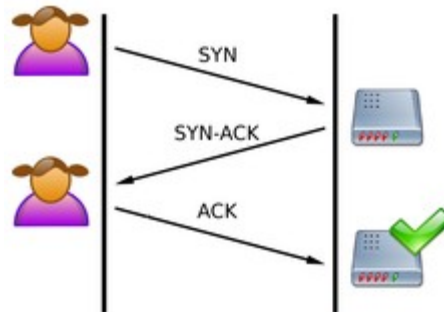
En seguridad informática, un **ataque de denegación de servicio**, también llamado ataque **DoS** (de las siglas en inglés *Denial of Service*), es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea **inaccesible a los usuarios legítimos**. Normalmente provoca la pérdida de la conectividad de la red por el **consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales** del sistema de la víctima.

Se genera mediante la saturación de los puertos con flujo de información, haciendo que el servidor se sobrecargue y no pueda seguir prestando servicios, por eso se le denomina "denegación", pues hace que el servidor no dé abasto a la cantidad de solicitudes.

Una ampliación del ataque Dos es el llamado **ataque distribuido de denegación de servicio**, también llamado ataque **DDoS** (de las siglas en inglés *Distributed Denial of Service*) el cual lleva a cabo generando un gran flujo de información desde **varios puntos de conexión**.

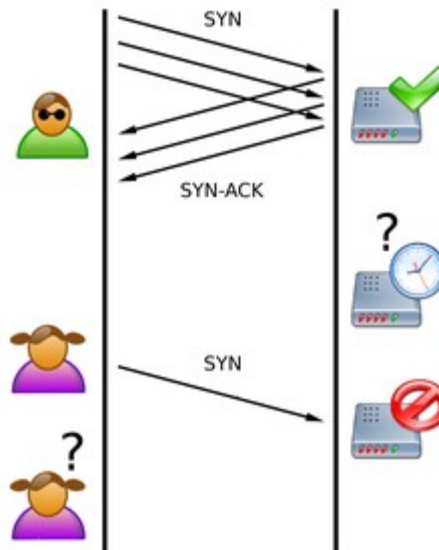
### DoS de tipo SynFlood

Cualquier sistema que proporcione servicios basados en TCP (ftp, http, etc.) es susceptible a este tipo de ataques. Para comprender el modo de funcionamiento de este ataque se ha de entender el **mecanismo por el cual se establece una conexión TCP**:



- Una vez que ha enviado un mensaje SYN
- y el servidor ha respondido con un mensaje SYN-ACK,
- el cliente responde con mensaje ACK.

En un ataque DoS de tipo Syn Flood



- Una vez que ha enviado un mensaje SYN
- y el servidor ha respondido con un mensaje SYN-ACK,
- el cliente no responde con mensaje ACK.

De esta forma se crea una **conexión "medio abierta"** . El servidor tiene en memoria las estructuras de datos necesarias para describir todas las conexiones pendientes. Esta estructura de datos tiene un tamaño finito, por lo que puede ser desbordada intencionadamente creando múltiples conexiones "medio abiertas" y una vez que esta desbordada el sistema no es capaz de aceptar nuevas conexiones.

Crear conexiones "medio abiertas" se consigue fácilmente con IP spoofing (Consiste en sustituir la dirección IP de origen por otra IP falsa), de manera que una vez que el paquete SYN llega al servidor, resulta imposible saber cual es el verdadero origen del paquete.

Normalmente hay un tiempo asociado a las conexiones pendientes, así las conexiones "medio abiertas" caducarán pasado un tiempo. No obstante, el atacante puede seguir enviando paquetes SYN (con dirección origen falsificada) más rápidamente que el servidor pueda cerrarlas debido a que hayan caducado. El problema se agrava si los atacantes son muchos distribuidos en diferentes equipos, los cuales incluso pueden ser ordenadores **zombie** infectados por algún tipo de malware que permita a un atacante coordinarlos para realizar un ataque al mismo sitio

#### **4 ¿Qué son los lenguajes de marcas?¿para qué se utilizan?**

Un **lenguaje de marcado** o **lenguaje de marcas** es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. El lenguaje de marcas más extendido es el HTML

Los lenguajes de marcas se suelen utilizar para codificar documentos. Las marcas son códigos que indican a un programa como debe tratar su contenido; si por ejemplo queremos que un texto aparezca con un determinado formato, dicho texto debe ir delimitado por la correspondiente marca que indique cómo debe ser mostrado.

#### **5 El lenguaje HTML:**

- **¿Qué es?**

El lenguaje más popular de marcas es el HTML (Hyper Text Markup Language). HTML es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web.

La versión actual de HTML es el HTML 4.01 incluye mecanismos para utilizar hojas de estilo, ejecución de scripts, capas, tablas mejoradas y otras ventajas.

HTML se escribe mediante etiquetas rodeadas entre los signos de menor que y mayor que (<, >) y algunas tienen atributos que pueden tomar algún valor.

- **Extensiones por defecto de los archivos**

Por convención, los archivos en formato HTML utilizan la extensión **.htm** o **.html**. La extensión le permite al SO operativo saber como tiene que interpretar o ejecutar el archivo.

- **En que secciones se divide un documento en HTML y para qué es cada una**

Las páginas HTML se dividen en dos partes: la cabecera y el cuerpo.

La cabecera incluye información sobre la propia página, como por ejemplo su título y su idioma.

El cuerpo de la página incluye todos sus contenidos, como párrafos de texto e imágenes.

Las etiquetas `<html>` y `</html>` delimitan la totalidad del documento web. El encabezado se delimita entre las etiquetas `<head>` y `</head>` y el cuerpo del documento entre `<body>` y `</body>`. La estructura básica de una página es:

```
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

- **Qué etiquetas más comunes se pueden encontrar en la sección de encabezado**

Las etiquetas más comunes y su utilidad son:

**Para el encabezado:**

Etiquetas	Uso
<code>&lt;title&gt;&lt;/title&gt;</code>	título de la página es el que aparecerá en la parte superior de la ventana del navegador
<code>&lt;meta&gt;</code>	Los metadatos contienen información oculta acerca del documento a través de atributos. Son utilizadas por los navegadores para clasificar la página. La información pueda ser extraída de forma automática por los principales robots, bases de datos e indexadores automáticos de los buscadores
<code>&lt;!doctype&gt;</code>	<i>declaración del tipo de documento</i> especifica la definición del tipo de documento (DTD) que se usa en el documento
<code>&lt;style&gt;&lt;/style&gt;</code>	Hojas de estilo a utilizar

<link>	Permite especificar hojas de estilo externas (vinculadas). Permite cambiar la presentación si tener que revisar el documento fuente en HTML
<adress></adress>	Permite a los autores especificar información de contacto

**Nota:** El SEO o **posicionamiento en buscadores**, **posicionamiento web** u **Optimizador de motores de búsqueda (SEO** por sus siglas en inglés, de *Search Engine Optimizer*) es el proceso de mejorar la visibilidad de un sitio web en los diferentes [buscadores](#), como [Google](#), [Yahoo!](#) o [Bing](#) de manera orgánica, es decir sin pagarle dinero al buscador para tener acceso a una posición destacada en los resultados.

El uso adecuado de las etiquetas de metadescripción <meta> en el encabezado de los documentos web es relevante a la hora de aparecer en los resultados de los buscadores.

Por tanto, una buena metadescripción del documento es importante para aparecer en los primeros resultados de las búsquedas

## 6. HTML versión 5. Qué novedades importantes incorpora frente a HTML 4. En que estado de desarrollo se encuentra.

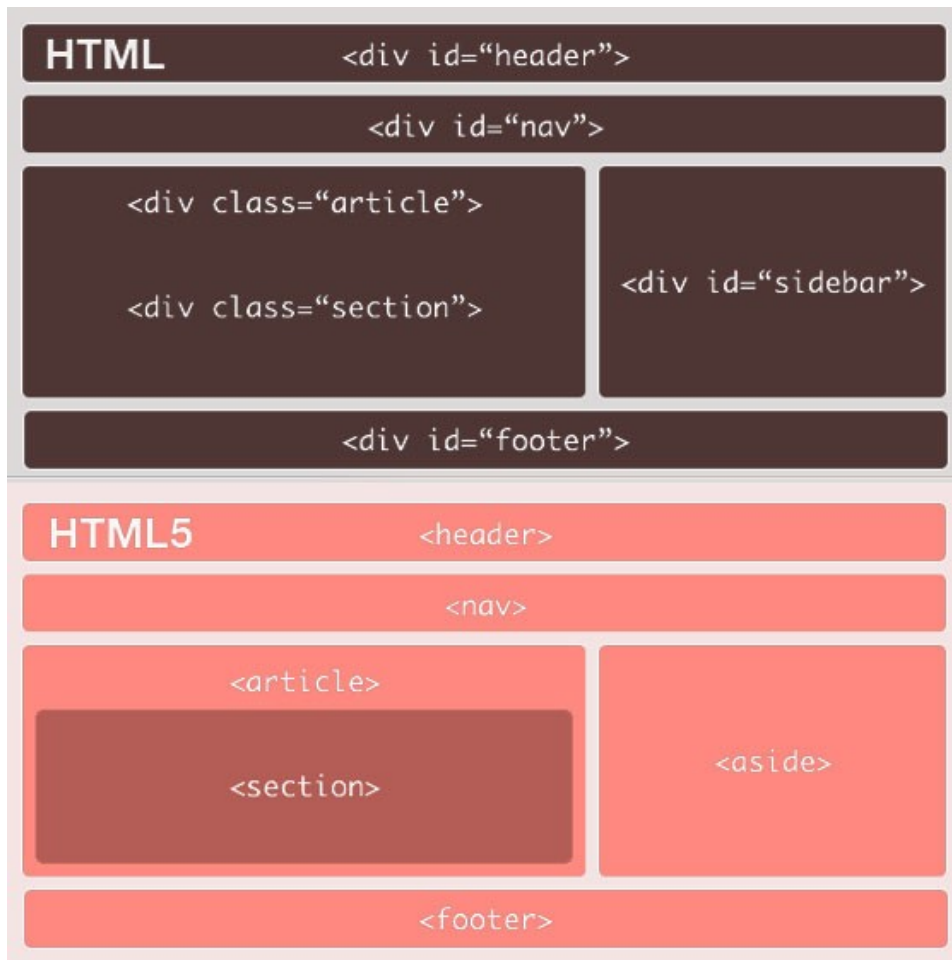
**HTML 5** (*HyperText Markup Language*, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web. A día de hoy (septiembre de 2011) todavía se encuentra en modo experimental, aunque ya es usado por múltiples desarrolladores y sitios web y soportado por las últimas versiones de los navegadores.

**HTML5** nos permite una mayor interacción entre nuestras páginas web y contenido multimedia (vídeo, audio, entre otros) así como una mayor facilidad a la hora de codificar nuestro diseño básico.

Incorpora:

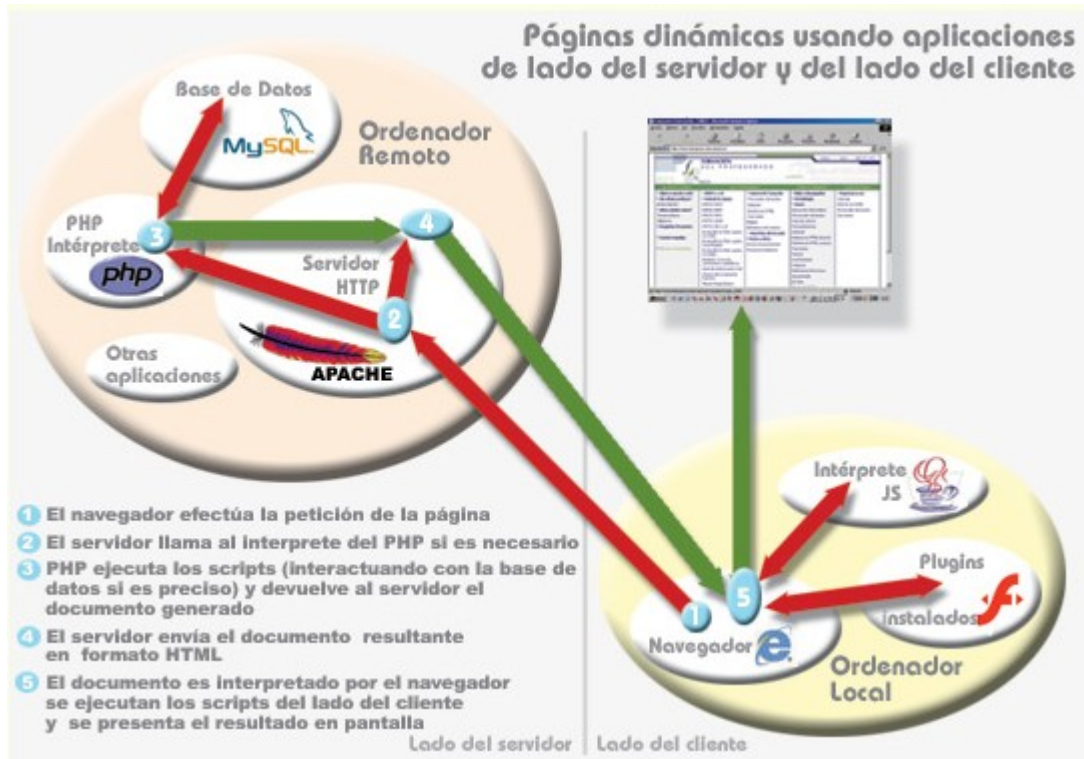
- Etiquetas para mejorar el diseño. **<header>**, **<footer>**, **<nav>**, **<section>**, **<article>** que están destinadas a remplazar la necesidad de tener una <div> para cada parte de la página, y en cambio, tener etiquetas específicas para ello.





- La nueva etiqueta **<video>** para insertar un reproductor de video, mejorando el reproductor antiguo utilizado por la etiqueta `<embed>` y evitándonos insertar el código de `<object>`, así como eliminar la necesidad del Flash Player para reproducir videos (lo que nos lleva a un ahorro en la cantidad de memoria utilizada).
- Una nueva tag **<audio>** para insertar audio en nuestro sitio web, remplazando la vieja etiqueta `<embed>` con las mismas cualidades de la etiqueta anterior.
- Una etiqueta **<canvas>** para manejo de gráficos en internet, sea para dibujar vectores o hacer animaciones.
- Etiquetas para manejar grandes cantidades de datos Datagrid, Details, menu y Command. Permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en cliente.
- Mejoras en los formularios: nuevos tipos de datos (eMail, number, url, datetime ...) y facilidades para validar el contenido sin Javascript.
- Visores: MathML (fórmulas matemáticas) y SVG (gráficos vectoriales).
- En general se deja abierto a poder interpretar otros lenguajes XML. Drag & Drop. Nueva funcionalidad para arrastrar objetos como imágenes.

## 7. Lenguajes de script de cliente y servidor



- **Para qué se utilizan y en qué se diferencian**

Se utilizan para añadir funcionalidad a los contenidos web visualizados en el navegador. Cuando desde el navegador solicitamos un archivo HTML alojado en un servidor este es enviado e interpretado por nuestro navegador (el cliente). Por tanto podemos hablar de lenguajes de lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor cuyo resultado es enviado al cliente en un formato comprensible para él. Y por otro lado, los lenguajes de lado cliente son aquellos que pueden ser directamente interpretados y renderizados por el navegador y no necesitan un preprocesamiento en el servidor.

- **Lenguajes de script de cliente**

- VISUAL BASIC SCRIPT
- JAVASCRIPT
- Applets de Java
- FLASH
- Lenguajes de script de servidor:
  - CGI
  - Perl
  - PHP

- ASP
  - JSP
- **Qué etiquetas se utilizan para insertarlo en un archivo web**

Inserción de javascript:

```
<script language=javascript>
--Código javascript --
</script>
```

Inserción de php

```
<?php
--Código PHP--
?>
```

## 8. Hojas de estilo

- **Qué son**

Las **hojas de estilo** son conjuntos de instrucciones, a veces en forma de archivo anexo, que se asocian a los archivos de texto y se ocupan de los aspectos de formato y de presentación de los contenidos: tipo, fuente y tamaño de letras, justificación del texto, colores y fondos, etc. Las hojas de estilo permiten liberar la composición del texto de los aspectos visuales y favorecen que se estructure y anote mediante códigos que permiten un tratamiento más eficaz de los contenidos.

- **Por qué aparecieron**

Las hojas de estilo representan un avance importante para los diseñadores de páginas web, al darles un mayor rango de posibilidades para mejorar la apariencia de sus páginas. En los entornos científicos en que la Web fue concebida, la gente estaba más preocupada por el contenido de sus páginas que por su presentación. A medida que la Web era descubierta por un espectro mayor de personas de distintas procedencias, las limitaciones del HTML se convirtieron en fuente de continua frustración, y los autores se vieron forzados a superar las limitaciones estilísticas del HTML.

- **Qué etiqueta se utiliza**

Se pueden insertar los estilos en el propio documento HTML especificado entre las etiquetas `<style type="text/css">` `</style>` en el encabezado del documento.

- **Cómo se especifica la hoja de estilo a aplicar si esta se encuentra en otro fichero**

Si queremos que los estilos se apliquen desde un fichero diferente, se llama a ese fichero; desde el documento HTML utilizando:

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

En el encabezado del documento en HTML.

## 9 Herramientas de diseño web

- **Qué son . Nombra algunas y sus características más importantes**

El diseño de páginas web trata básicamente de realizar un documento con información hiperenlazado con otros documentos y asignarle una presentación para diferentes dispositivos de salida (en una pantalla de [computador](#), en [papel](#), en un [teléfono móvil](#), etc). Las herramientas de diseño web nos facilitan esta tarea.

Las fases de un desarrollo web, así como los lenguajes de programación usados, son muy extensas y variadas, y por ello necesitamos herramientas específicas para cada una de ellas.

### Fases de desarrollo de una web

Para elegir las herramientas a utilizar, antes debemos identificar las fases del proceso que forman el ciclo de vida de un desarrollo web:

- **Diseño:**  
el diseño consiste en crear esbozos de la web final mediante una herramienta gráfica, como Photoshop, GIMP o Inkscape.
- **Maquetación HTML/CSS:**  
la maquetación consiste en convertir los esbozos creados en la fase anterior en plantillas HTML, su respectiva hoja de estilos, y las imágenes usadas. Podemos utilizar Inkscape, Adobe Fireworks o Photoshop como editores vectoriales para crear iconos, banners y demás dibujos.
- **Programación cliente:**  
la programación cliente consiste básicamente en Javascript. Podemos utilizar en esta fase Dreamweaver, Aptana o Amaya. También permiten maquetar.
- **Programación servidor:**  
en esta fase, que se desarrolla junto con la anterior, crearemos la aplicación web en un lenguaje de servidor, como puede ser PHP, ASP .NET, Python, Perl, etc. Herramientas: Zend Studio, Eclipse con plugin de PHP
- **Depuración:**  
esta fase enlaza la anterior con la siguiente, y es donde haremos las pruebas unitarias, aserciones, trazas, etc. Zend Platform, Firefox con las extensiones firebug y web developer.
- **Pruebas en local:**

en nuestro servidor local haremos todas las pruebas posibles. Servidores web como Apache,

- **Subir ficheros al hosting:**

una vez nuestra web esté completada y bien testada en nuestro servidor local (desarrollo), la subiremos al servidor del hosting elegido (producción). Dependiendo del hosting, podremos usar FTP, SFTP (SSH), WebDAV, o incluso Subversion. Dreamweaver y Zend Studio incluyen clientes de FTP y de SFTP (SSH)

- **Pruebas en hosting:**

realizaremos las últimas pruebas en el servidor del hosting para comprobar que el cambio de servidor no ha afectado a nada. Para evitar problemas, nuestro servidor local debe tener exactamente la misma configuración que el servidor del hosting.

## **10. Bases de datos**

- **Qué son**

Es el conjunto de informaciones almacenadas en un soporte legible por ordenador y organizadas internamente por registros (formado por todos los campos referidos a una entidad u objeto almacenado) y campos (cada uno de los elementos que componen un registro). Permite recuperar cualquier clase de información: referencias, documentos textuales, imágenes, datos estadísticos, etc.

Es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente

- **Para qué se utilizan en el desarrollo web.**

La solución a la limitación de las páginas web estáticas fueron los lenguajes de programación de script en el lado del servidor y las bases de datos. La información mostrada puede variar en función de la interacción del cliente en el navegador. La mayoría de páginas que hay actualmente en Internet son dinámicas y utilizan bases de datos.

- **Gestores de bases de datos**

- MySQL
- PostgreSQL
- Firebird
- SQLite
- Microsoft SQL Server