

TEMA

1

2ºDAW



Introducción DWES

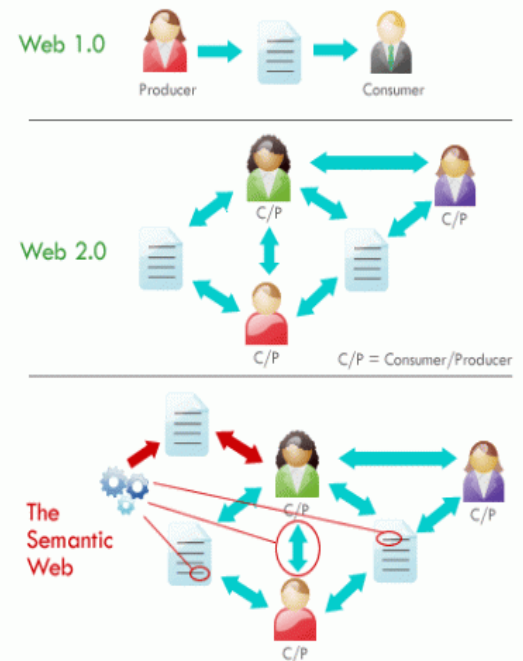


1

Características de la programación Web

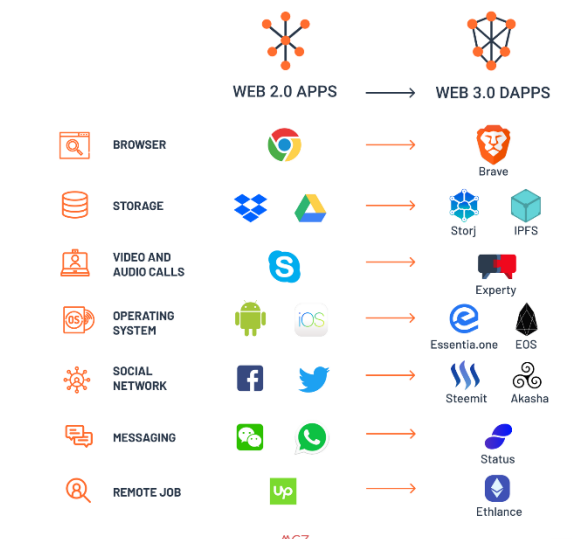
HISTORIA Y FUNDAMENTOS WEB

- Inicialmente la web era simplemente una colección de **páginas estáticas (WEB 1.0)**, documentos, etc., que podían consultarse o descargarse.
- El siguiente paso fue la inclusión de un método para confeccionar páginas **dinámicas (WEB 2.0)** conocido como CGI (common gateway interface).
- Se empiezan a desarrollar alternativas a los CGI para solucionar el problema de rendimiento que presentan. Surgen sistemas de ejecución de módulos más integrados en el servidor y lenguajes de programación interpretados (Java, PHP, ASP) que permitían incluir código en las páginas HTML.
- Estas tecnologías dieron paso a las aplicaciones web, instaladas en un servidor y a las que se accede mediante un navegador.
- El desarrollo de Internet, junto con su capacidad de almacenamiento y ejecución de programas, ha dado lugar a lo que se conoce como **computación en la nube. Ejemplo Google Apps**



Una vez aclarada la definición de web 2.0, trataremos de explicar aquí **hacia dónde evoluciona el concepto Web 3.0**. Si buscáis en internet veréis el concepto de web 3.0 unido a conceptos tales como

- ✓ **Contenidos semánticos:** Se llevará a cabo un proceso para conseguir un perfeccionamiento real de las búsquedas por internet, en cubrir la necesidad de que una búsqueda termine siendo un proceso más usable y más humano.
- ✓ **Búsquedas de lenguaje natural:** A día de hoy se trabaja en la posibilidad de construir sobre la web una base de conocimiento con las preferencias de los usuarios.
- ✓ **Contenidos accesibles sin navegación**
- ✓ **Tecnologías de inteligencia artificial**



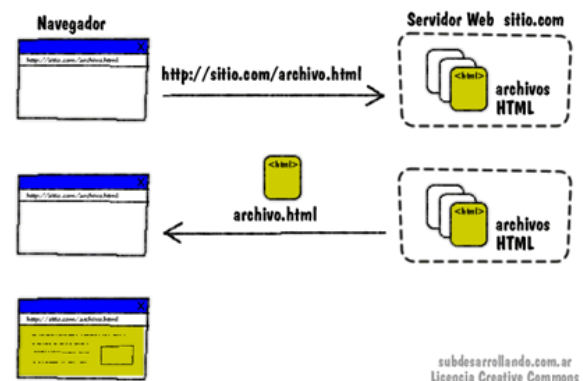
+ PÁGINAS ESTÁTICAS O DINÁMICAS

Una **página estática (WEB 1.0)** está compuesta por código HTML/XHTML+ CSS (hoja de estilo). Cuando el archivo es requerido por el navegador, el servidor lo envía y punto. Todos los visitantes verán el mismo contenido. Una página Web estática es aquella que es básicamente informativa, el programador es el único que puede modificar su contenido. **NO se utilizan bases de datos ni se requiere programación de servidor.** No permiten grandes libertades o funcionalidades más allá de los enlaces.

Su principal ventaja es que su costo es mucho menor que las páginas dinámicas, teniendo un diseño elegante, moderno, funcional.

El proceso es el que se refleja en la siguiente figura. Los pasos son los siguientes:

1. **Tu ordenador solicita a un servidor web una página** con extensión .htm, .html o .xhtml.
2. **El servidor busca esa página** en un almacén de páginas (cada una suele ser un fichero).
3. Si **el servidor encuentra esa página**, la recupera.
4. Y por último **se la envía al navegador** para que éste pueda mostrar su contenido.



Este es un ejemplo típico de una **comunicación cliente-servidor**. El cliente es el que hace la petición e inicia la comunicación, y el servidor es el que recibe la petición y la atiende. En nuestro caso, el navegador es el cliente web.

Las páginas web estáticas pueden visualizarse sin que intervenga un servidor web

Una **página Web dinámica** es aquella que puede interactuar con el visitante y/o administrador Web, pudiéndose modificar el contenido de la página. Ejemplos de esto son: cuando el usuario puede escribir un comentario, escoger los productos y ponerlos en un carrito de compras, subir archivos o fotografías, etc. En este caso **SI se utilizan bases de datos y se requiere programación Web. El lenguaje de servidor utilizado puede ser alguno de los siguientes: PHP, ASP.NET o JSP (Java).**

Sus ventajas y posibilidades son infinitas, desde una página informativa hasta una potente herramienta de trabajo rica en contenido dinámico. La web 2.0 es lo que se conoce como webs dinámicas.

Dentro de las páginas web dinámicas, es muy importante distinguir **dos tipos**:

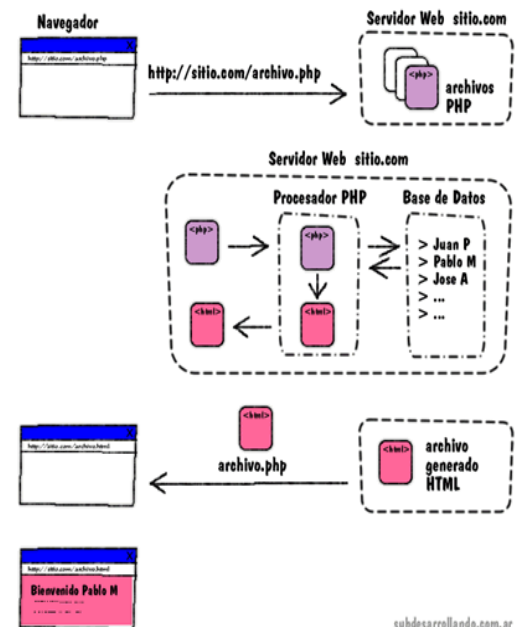
1. Aquellas que incluyen código que ejecuta el navegador. En estas páginas el código ejecutable, normalmente en lenguaje **JavaScript**, se incluye dentro del HTML (o XHTML) y se descarga junto con la página. Cuando el navegador muestra la página en pantalla, ejecuta el código que la acompaña. Este código puede incorporar múltiples funcionalidades que pueden ir desde mostrar animaciones hasta cambiar totalmente la apariencia y el contenido de la página.

En este módulo no vamos a ver **JavaScript**, salvo cuando éste se relaciona con la programación web del lado del servidor.

2. Como ya sabes, hay muchas páginas en Internet que no tienen extensión .htm, .html o .xhtml. Muchas de estas páginas tienen extensiones como **.php**, **.asp**, **.jsp**, **.cgi** o **.aspx**. En éstas, el contenido que se descarga al navegador es similar al de una página web estática: HTML (o XHTML). Lo que cambia es la forma en que se obtiene ese contenido. Al contrario de lo que vimos hasta ahora, esas páginas no están almacenadas en el servidor; más concretamente, el contenido que se almacena no es el mismo que después se envía al navegador. El HTML de estas páginas se forma como resultado de la ejecución de un programa, y esa ejecución tiene lugar en el servidor web (aunque no necesariamente por ese mismo servidor).

El esquema de funcionamiento de una página web dinámica es el siguiente:

1. El cliente web (navegador) de tu ordenador solicita a un servidor web una página web.
1. El **servidor** busca esa página y la recupera.
2. En el caso de que se trate de una página web dinámica, es decir, que su contenido deba ejecutarse para obtener el HTML que se devolverá, el servidor web contacta con el **módulo responsable de ejecutar el código** y se lo envía.
5. Como parte del proceso de ejecución, puede ser necesario obtener información de algún **repositorio** (Cualquier almacén de información digital, normalmente una base de datos), como por ejemplo consultar registros almacenados en una base de datos.
6. El **resultado de la ejecución será una página en formato HTML**, similar a cualquier otra página web no dinámica.
7. El **servidor web envía el resultado obtenido al navegador**, que la procesa y muestra en pantalla.



subdesarrollando.com.ar
Licencia Creative Commons

Obviamente, el navegador no envía esa misma página a todos los usuarios, sino que la genera de forma dinámica en función de quién sea el usuario que se conecte. Para generarla ejecuta un programa que obtiene los datos de tu usuario (tus contactos, la lista de mensajes recibidos) y con ellos compone la página web que recibes desde el servidor web.

Un servidor web o **servidor HTTP** esta siempre a la escucha a través del **puerto 80 (no seguro) o 443(seguro)** esperando recibir peticiones desde la conexión cliente Web (normalmente su navegador). El servidor Web utilizan el protocolo HTTP/HTTPS (Hyper Text Transfer Protocol) para comunicarse a través de una conexión TCP.

APLICACIONES WEB

Las aplicaciones web emplean páginas web dinámicas para crear aplicaciones que se ejecuten en un servidor web y se muestren en un navegador. Puedes encontrar aplicaciones web para realizar múltiples tareas. Unas de las primeras en aparecer fueron las que viste antes, los clientes de correo, que te permiten consultar los mensajes de correo recibidos y enviar los tuyos propios utilizando un navegador.

Hoy en día existen aplicaciones web para multitud de tareas como procesadores de texto, gestión de tareas, o edición y almacenamiento de imágenes. Estas aplicaciones tienen ciertas ventajas e inconvenientes si las comparas con las aplicaciones tradicionales que se ejecutan sobre el sistema operativo de la propia máquina.

Ventajas de las aplicaciones web:

- ✓ No es necesario instalarlas en aquellos equipos en que se vayan a utilizar. Se instalan y se ejecutan solamente en un equipo, en el servidor, y esto es suficiente para que se puedan utilizar de forma simultánea desde muchos equipos.
- ✓ Como solo se encuentran instaladas en un equipo, es muy sencillo gestionarlas (hacer copias de seguridad de sus datos, corregir errores, actualizarlas).
- ✓ Se pueden utilizar en todos aquellos sistemas que dispongan de un navegador web, independientemente de sus características (no es necesario un equipo potente) o de su sistema operativo.
- ✓ Se pueden utilizar desde cualquier lugar en el que dispongamos de conexión con el servidor. En muchos casos esto hace posible que se pueda acceder a las aplicaciones desde sistemas no convencionales, como por ejemplo teléfonos móviles.

Inconvenientes de las aplicaciones web:

- La interface de usuario de las aplicaciones web es la página que se muestra en el navegador. Esto restringe las características del interface a aquellas de una página web.
- Dependemos de una conexión con el servidor para poder utilizarlas. Si nos falla la conexión, no podremos acceder a la aplicación web.
- La información que se muestra en el navegador debe transmitirse desde el servidor. Esto hace que cierto tipo de aplicaciones no sean adecuadas para su implementación como aplicación web (por ejemplo, las aplicaciones que manejan contenido multimedia, como las de edición de vídeo).

Hoy en día muchas aplicaciones web utilizan las ventajas que les ofrece la generación de **páginas dinámicas**. La gran mayoría de su contenido está almacenado en una base de datos. Aplicaciones como son los **gestores de contenidos (CMS)**: Wordpress, Drupal, Joomla! y otras muchas ofrecen dos partes bien diferenciadas:

- ✚ **Una parte externa o front-end**, que es el conjunto de páginas que ven la gran mayoría de usuarios que las usan (usuarios externos).
- ✚ **Una parte interna o back-end**, que es otro conjunto de páginas dinámicas que utilizan las personas que producen el contenido y las que administran la aplicación web (usuarios internos) para crear contenido, organizarlo, decidir la apariencia externa, etc.



Los CMS son programas, al que accedemos a través de cualquier **navegador**, que desde un panel de control (interfaz) nos permite modificar las bases de datos donde se controlan los contenidos de una página web, así podemos añadir y editar contenido de forma más sencilla, sin tener que utilizar la programación. Usar un gestor de contenidos nos permitirá trabajar el diseño y el contenido de forma independiente. Esto te permitirá personalizar el diseño de tu web sin necesidad de tener muchas nociones de diseño y de desarrollo web. Los CMS más conocidos suelen funcionar con plantillas prediseñadas que se instalan en él y se adaptan al contenido que tu hayas creado o vayas a crear. Cerca del 40% páginas web están realizadas con un CMS



© W3Techs.com	usage	change since 1 February 2018	market share	change since 1 February 2018
1. WordPress	30.2%	+0.8%	60.3%	+0.3%
2. Joomla	3.1%		6.3%	-0.1%
3. Drupal	2.2%		4.4%	-0.2%
4. Magento	1.2%		2.3%	-0.2%
5. Shopify	1.0%	+0.1%	2.0%	+0.1%

2

Tecnologías para programación web del lado del servidor

En este bloque **vas a aprender las distintas tecnologías** que se pueden utilizar para **programar aplicaciones** que se ejecuten en un **servidor web**, y cómo se relacionan unas con otras. Verás las ventajas e inconvenientes de utilizar cada una, y qué lenguajes de programación deberás aprender para utilizarlas.

Los **componentes principales** con los que debes contar para ejecutar aplicaciones web en un servidor son los siguientes:

- ✓ Un **servidor web** para recibir las peticiones de los clientes web (normalmente navegadores) y enviarles la página que solicitan (una vez generada puesto que hablamos de páginas web dinámicas).
El servidor web debe conocer el procedimiento a seguir para generar la página web: quemódulo se encargará de la ejecución del código y cómo se debe comunicar con él.
- ✓ El **módulo encargado de ejecutar el código** o programa y generar la página web resultante. Este módulo debe integrarse de alguna forma con el servidor web, y dependerá del lenguaje y tecnología que utilicemos para programar la aplicación web.
- ✓ Una **aplicación de base de datos**,
- ✓ El **lenguaje de programación** que utilizarás para desarrollar las aplicaciones.

Además de los componentes a utilizar, también es importante decidir cómo vas a **organizar el código** de la aplicación. Muchas de las arquitecturas que se usan en la programación de aplicaciones web te ayudan a estructurar el código de las aplicaciones en **capas o niveles**. El motivo de dividir en capas el diseño de una aplicación es que se puedan **separar las funciones lógicas** de la misma, de tal forma que sea posible ejecutar cada una en un servidor distinto (en caso de que sea necesario).

Cada capa puede ocuparse de una o varias de las funciones anteriores. Por ejemplo, en las aplicaciones de **3 capas** nos podemos encontrar con:

- ✓ Una **capa cliente**, que es donde programarás todo lo relacionado con el interface de usuario, esto es, la parte visible de la aplicación con la que interactuará el usuario.
- ✓ Una **capa intermedia** donde deberás programar la funcionalidad de tu aplicación.
- ✓ Una **capa de acceso a datos**, que se tendrá que encargar de almacenar la información de la aplicación en una base de datos y recuperarla cuando sea necesario.

ARQUITECTURAS O PLATAFORMAS




La primera elección que harás antes de comenzar a programar una aplicación web es la arquitectura que vas a utilizar. Hoy en día, puedes elegir entre:

- ✓ **Java EE (Enterprise Edition), que antes también se conocía como J2EE**. Es una plataforma orientada a la programación de aplicaciones en lenguaje Java. Está apoyada por grandes empresas como Sun y Oracle, que mantienen Java, o IBM.

Dentro de esta arquitectura existen distintas tecnologías como las **páginas JSP** y los **servlets**, ambos orientados a la generación dinámica de páginas web, o **los EJB**, componentes que normalmente aportan la lógica de la aplicación web.

- ✓ **CGI/Perl.** Es la combinación de dos componentes: Perl, un potente lenguaje de código libre creado originalmente para la administración de servidores, y CGI, un estándar para permitir al servidor web ejecutar programas genéricos, escritos en cualquier lenguaje. El principal inconveniente de esta combinación es que CGI es lento.
- ✓ **ASP Net es la arquitectura comercial propuesta por Microsoft** para el desarrollo de aplicaciones. Es la parte de la plataforma .Net destinada a la generación de páginas web dinámicas. Proviene de la evolución de la anterior tecnología de Microsoft, ASP. El lenguaje de programación puede ser **Visual Basic.Net o C#**. La arquitectura utiliza el servidor web de Microsoft, IIS, y puede obtener información de varios gestores de bases de datos entre los que se incluye, como no, **Microsoft SQL Server**. Una gran ventaja es que tiene su propio entorno de desarrollo, **Visual Studio**, aunque hay otras opciones disponibles. La mayor es que se trata de una plataforma comercial de código propietario.
- ✓ **AMP.(Son las siglas de Servidor web Apache, MySQL/MariaDB y PHP).** Todos los componentes de esta arquitectura son de código libre (open source). Es una plataforma de programación que permite desarrollar aplicaciones de tamaño pequeño o mediano con un aprendizaje sencillo.

Existen paquetes de aplicaciones que nos permiten instalar todas las herramientas necesarias para la arquitectura AMP

Para Linux	
Para MAC	 <p>MAMP</p>
Para Windows	

✚ HERRAMIENTAS DE PROGRAMACIÓN

Una de las grandes preguntas que todo desarrollador se hace al iniciar su vida profesional es, **¿qué es mejor un IDE o un editor de código/texto?**

A simplemente vista, tanto el IDE como el editor de texto pueden resultar muy similares, pues en ambos es posible escribir código y ejecutarlo. Sin embargo y como dice el dicho “el diablo esta en los detalles”.



En definición, un **IDE es Integrated Development Environment (entornos de desarrollo integrado)**, lo que significa que los IDE's tiene muchas más herramientas que se integran en mismo programa. Solo por poner algunos ejemplos, los IDEs nos permite

- Resaltado de texto.
- Completado automático.
- Navegación en el código.
- Comprobación de errores al editar.
- Acceso a la BD
- Generación automática de código.
- Ejecución y depuración.
- Gestión de versiones.

Por otra parte, los **editores de texto** en un inicio eran solo programas que permitían ver el código de una forma agradable y realizar algunas acciones muy simples, como abrir terminales para ejecutar el programa. Sin embargo, los editores de texto modernos, han estado evolucionando muy rápido, agregando capacidades que solo los IDE's tenían.

IDE	
Editores de texto/código	

En este módulo vamos a instalar **IDE Netsbeans** y como **editor de código (“vitaminado”): Sublime Text + plugin.**

Plugins que vamos a instalar son

Package control: Es necesario para instalar los plugin <https://packagecontrol.io/installation>

Emmet: Genera código automáticamente (Ej html:5)

Side Bar Enhancement: Opciones extras

Color Picker: Colores HTML

BracketHighlighter, Highlight y Highlighter: Señaladores de código

Sublime Linter, SublimeLinter-PHP y sublimeLinter-PHPCS: Errores de código tendrás que configurar en **Preference>Package Settings> SublimeLinter>Settings-user**

EJERCICIOS

- 1) Explica el funcionamiento de un servidor Web. ¿Cuáles son las diferencias entre páginas estáticas y dinámicas?
- 2) ¿Qué son las DAPPS (WEB 3.0)?
- 3) ¿Qué diferencia a la web 1.0, web 2.0 y la futuras webs?
- 4) Una de las novedades de la web 2.0 es el crowdfunding. ¿En qué consiste?
- 5) Lee el artículo Comparación entre navegadores web y realiza una conclusión general y determina cuál puede ser el mejor navegador.
- 6) ¿Qué son las hojas de estilo o CSS? ¿Qué extensión tienen sus archivos?
- 7) ¿Por qué se utilizan los CMS? ¿Cuál es el más usado? ¿Busca algunos por Internet de un tema de interés?
- 8) ¿Qué es LAMP? ¿Y XAMPP?
- 9) Si quieres montar una tienda virtual para ayudar a un amigo. ¿Qué herramientas utilizarías? ¿Cuál es su página Web de la herramienta utilizada?
- 10) ¿Qué significa organizar el código de una aplicación en 3 capas?
- 11) ¿Qué prefieres para programar un IDE o editor de código? Justifica tu respuesta
- 12) ¿Cuáles son las arquitecturas de programación mas utilizada en programación del lado del servidor?
- 13) ¿Qué es Drupal?