Índice

[1 Introducción 2](#_Toc524473343)

[1.1 Internet, la Web y sus aplicaciones 2](#_Toc524473344)

[1.2 Evolución de la web según la forma de usarla 3](#_Toc524473345)

[2 Aspectos generales de arquitecturas web 5](#_Toc524473346)

[3 Evolución de los servicios web 6](#_Toc524473347)

[4 Tecnologías asociadas a las aplicaciones web. 7](#_Toc524473348)

[5 Tipos de aplicaciones web 8](#_Toc524473349)

[6 Arquitecturas web. Modelos 10](#_Toc524473350)

[7 Plataformas web libres y propietarias 12](#_Toc524473351)

[8 Escalabilidad 13](#_Toc524473352)

[8.1 Escalabilidad vertical 13](#_Toc524473353)

[8.2 Escalabilidad horizontal 13](#_Toc524473354)

[8.3 Clústers 14](#_Toc524473355)

# Introducción

## Internet, la Web y sus aplicaciones

**Red de ordenadores**

1959 – Paul Baran comenzó a trabajar en una red de comunicación segura capaz de sobrevivir a un ataque con armas nucleares, publicando sus resultados a partir de 1960. En su trabajo aparecían tres ideas claras:

* Red descentralizada con múltiples caminos entre dos puntos.
* División de mensajes completos en fragmentos que seguirían caminos distintos.
* Capacidad de responder automáticamente a sus propios fallos.

1962 – Joseph C.R. Licklider contempla el concepto de comunicar usuarios de distintos ordenadores. La agencia ARPA dependiente del Departamento de Defensa estadounidense.

1962 – En el MIT Leonard Kleinrock trabajaba en el concepto de almacenar y reenviar mensajes.

1965 – Donald Davies comienza a relacionar estos conceptos en el NPL (National Physical Laboratory) inglés.

DARPA, RAND, MIT y NPL se convirtieron en los cuatro primeros nodos experimentales de ARPANET. Utilizando el protocolo de conexión NCP (NetWare Core Protocol).

**Cronología de conexión:**

21-nov-1969: primera conexión de dos nodos.

5-dic-1969: toda la red estaba lista.

Mar-1970: se conectaba la costa este de Estados Unidos.

1971: 24 ordenadores conectados (universidades y centros de investigación).

1981: 213 ordenadores conectados.

1983: 500 ordenadores conectados. Con el fin de estandarizar se cambia al protocolo TCP/IP.

1989: Se integra con los protocolos OSI.

1990: En el CERN se desarrolla el lenguaje HTML, el primer servidor web y el primer cliente web.

**INTERNET**

Conjunto descentralizado de redes de comunicación interconectadas.

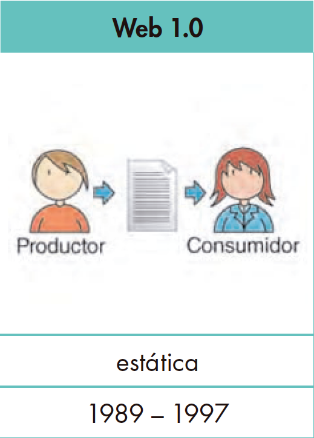
Utilizan familia de protocolos **TCP/IP** (http, ftp, pop3…).

**World Wide Web (WWW o Web)**

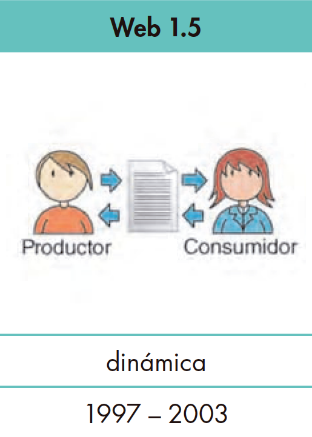
Servicio de más éxito de internet (http/**https**).

Aplicaciones web que sustituyen casi todos los protocolos TCP/IP

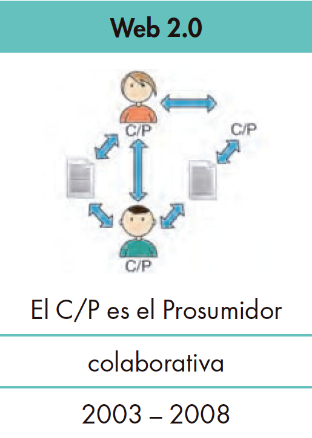
## Evolución de la web según la forma de usarla

**WEB 1.0**

Se concibió como una colección de documentos estáticos enlazados para su consulta o descarga, donde los usuarios no podían interactuar con ellos, teniendo que hacer un consumo pull, es decir, se iba a por el recurso siendo la actualización unidireccional.

**WEB 1.5**

De manera progresiva, se desarrollaron métodos que permitiesen la creación de páginas con contenidos dinámicos 1997. No tardaron en aparecer diversas arquitecturas y lenguajes de programación, tanto en la parte del cliente como en la parte del servidor, con lenguajes que permitían el desarrollo de páginas web dinámicas e interactivas. Se desarrollaron métodos que permitiesen la creación de páginas con contenidos dinámicos, de modo que estos se generasen en función de los parámetros de la petición, aparecieron los primeros CGI que se ejecutaban en el lado del servidor y lenguaje de programación que permitían incluir código en las páginas HTML.

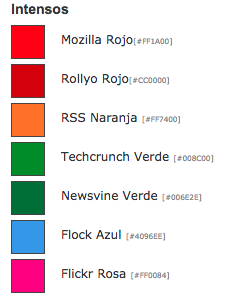
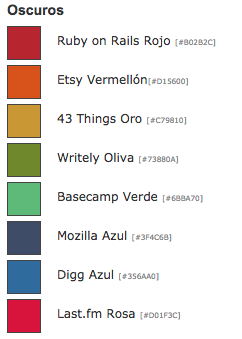
**WEB 2.0**

Estas tecnologías dieron paso a las aplicaciones web 2.0, donde los usuarios colaboran entre ellos y el consumo pasa a ser **push**, al usuario ahora le llegan los nuevos contenidos sin tener que ir a buscarlos, la actualización del contenido pasa a ser bidireccional. Los recursos son más fáciles de producir y consumir

¿Personalizable o colaborativa?

En un principio se llamó web 2.0 a un movimiento de cambio en la forma de desarrollar páginas web. El cambio era básicamente de **apariencia** y sus dos características clave fueron el cambio de paleta de colores a utilizar y la personalización de las páginas web.

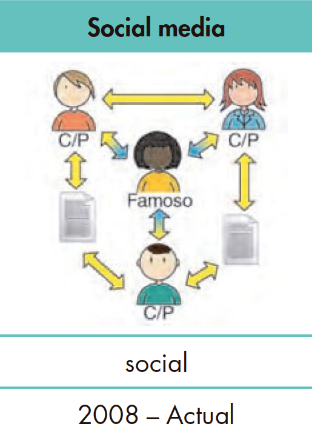
La **Web 2.0** tiene un estilo muy definido, colores brillantes, degradados, plateados y sombras.

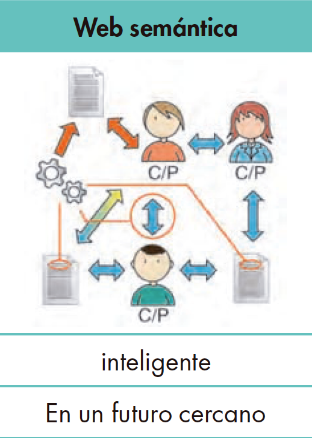
Entre las opciones de personalización la tendencia era tener una página de inicio en la decidir qué quiere ver, incluyendo opciones como el tiempo, RSS, cotización en bolsa.



Finalmente, este modelo de web no fue todo lo exitoso que se esperaba. Y con el nacimiento de los blogs la web se encaminó hacia la web colaborativa que conocemos. iGoogle ya no existe pero existen páginas web clon como por ejemplo: [iGoogle Portal](http://www.igoogleportal.com/portal/index.php#tab/3) en la que podemos ver cómo funcionaban estas webs personalizables. Existían páginas que nos explicaban cómo se utilizaba iGoogle: [Tutorial igoogle](http://www.mclibre.org/consultar/internet/google/google_personalizada.html)

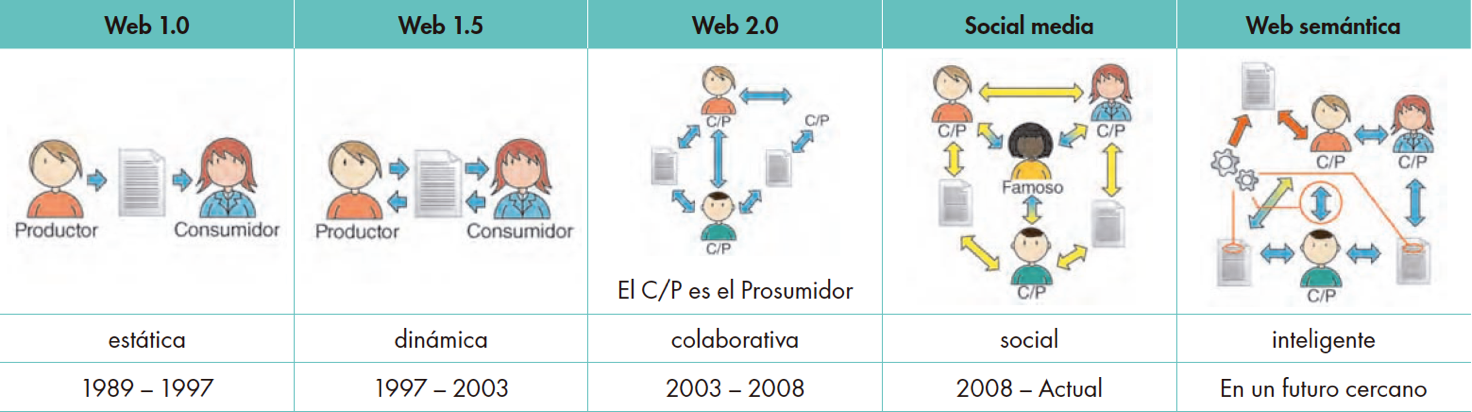
**Social Media**

La evolución de la Web gracias a la aparición de las redes sociales (**Facebook**, **Twitter**, **Instagram**…) y la posibilidad de disponer de internet en los dispositivos móviles (**SmartPhone**, **tablets**, **GPS**…) ha llevado a la aparición en 2008 de una nueva etapa llamada social media.

**Web semántica**

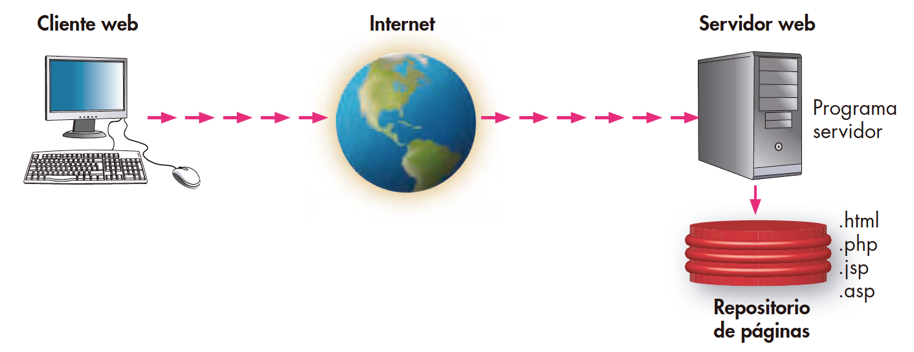
Lo próximo que nos viene en pocos años es la web semántica, que nos permitirá diferenciar lo que buscamos dependiendo del **contexto** en el que nos movamos. Esto será posible gracias a los metadatos y HTML5 (lenguaje de programación semántico).

**Resumen**



# Aspectos generales de arquitecturas web

La arquitectura World Wide Web (WWW) de Internet provee un modelo de programación sumamente poderoso y flexible. Las aplicaciones y los contenidos son presentados en formatos de datos estándar y son localizados por aplicaciones conocidas como "web browsers" (navegadores web), que envían peticiones de objetos a un servidor y éste responde con el dato codificado según un formato estándar.



Los estándares WWW especifican muchos de los mecanismos necesarios para construir un ambiente de aplicación de propósito general, por ejemplo:

* **Modelo estándar de nombres**: todos los servidores, así como el contenido de la WWW se denominan según un Localizador Uniforme de Recursos (Uniform Resource Locator: URL).
* **Contenido**: a todos los contenidos en la WWW se les especifica un determinado tipo permitiendo de esta forma que los browsers (navegadores) los interpreten correctamente.
* **Formatos de contenidos estándar**: todos los navegadores soportan un conjunto de formatos estándar, por ejemplo, HTML, JavaScript, etc.
* **Protocolos estándar:** éstos permiten que cualquier navegador pueda comunicarse con cualquier servidor web. El más comúnmente usado en WWW es HTML (Protocolo de Transporte de HiperTexto), que opera sobre el conjunto de protocolos TCP/IP.

Esta infraestructura permite a los usuarios acceder a una gran cantidad de aplicaciones y servicios de terceros. También permite a los desarrolladores crear aplicaciones y servicios para una gran comunidad de clientes.

Los aspectos generales a destacar en una arquitectura web son los siguientes:

* Escalabilidad.
* Separación de responsabilidades.
* Portabilidad.
* Utilización de componentes en los servicios de infraestructura.
* Gestión de las sesiones del usuario
* Aplicación de patrones de diseño.

**El hecho que haya estándares afecta tanto a los navegadores como a los desarrolladores web. ¿Cómo?**

De forma genérica podríamos decir que la arquitectura web es un modelo compuesto de tres capas:

* **Capa de Base de Datos,** donde estaría toda la documentación de la información que se pretende administrar mediante el servicio web y emplearía una plataforma del tipo MySQL, PostgreSQL, etc.
* En una segunda capa estarían los **servidores de aplicaciones web,** ejecutando aplicaciones de tipo Apache, Tomcat, Resin, etc.
* Una tercera capa estarían los **clientes del servicio web** al que accederán mediante un navegador web como Firefox, Internet Explorer, Opera, etc.

# Evolución de los servicios web

La evolución del uso de Servicios web en las organizaciones está fuertemente ligada al desarrollo de Internet como red prestadora de servicios. Entre los factores que han impulsado el uso de los servicios web se encuentran:

* El **contenido se está volviendo más dinámico**: Los sitios web actuales proporcionan contenidos "instantáneos". Un Servicio web debe ser capaz de combinar contenido proveniente de fuentes muy diferentes.
* El **ancho de banda es menos costoso**: Actualmente un Servicio web puede entregar tipos variables de contenidos como vídeo o audio. A medida que crezca el ancho de banda, los servicios web deben adaptarse a nuevos tipos de contenidos.
* El **almacenamiento es más barato**: Un Servicio web debe ser capaz de manejar cantidades masivas de datos, y debe poder hacerlo de forma inteligente.
* El **éxito de la computación extendida** se está volviendo más importante: con cientos de millones de dispositivos como teléfonos móviles, agendas electrónicas, etc. existentes actualmente, se está llegando a un momento en el cual las computadoras están dejando de ser el dispositivo más común en Internet.

A medida que las plataformas se hacen más diversas, los estándares se vuelven más importantes. Un servicio web no puede exigir que los usuarios ejecuten un navegador concreto. Los servicios web deben servir a todo tipo de dispositivos, plataformas y navegadores, entregando contenido sobre una amplia variedad de tipos de conexión.

Estos factores, unidos a los beneficios proporcionados por los servicios web en la organización y los buenos productos disponibles para su desarrollo, han hecho que su utilización se extienda sin mayores obstáculos.

En términos generales, cuando se empiezan a utilizar servicios web en una organización, estos se desarrollan e implementan como servicios simples, que poco a poco se van integrando hasta llegar a servicios web mucho más complejos.

En los orígenes del mundo web nos situábamos ante un entorno estático, con páginas en formato HTML que raramente sufrían modificaciones o actualizaciones y en las que apenas había interacción con el usuario.

La **Web 2.0** es la transición que se ha dado desde las aplicaciones tradicionales hacia aplicaciones que funcionan a través de la web y que están fuertemente enfocadas al usuario final. En este nuevo entorno existen una serie de nuevas tecnologías que, en general, tienen como objetivo:

* Transformar software de escritorio hacia la web.
* Separar hojas de estilo.
* Potenciar el trabajo colaborativo y la utilización de redes sociales.
* Dar control total a los usuarios en el manejo de su información.

# Tecnologías asociadas a las aplicaciones web.

Las aplicaciones web hoy en día funcionan usando características de la web estática y de la web dinámica. Como se ha visto anteriormente, en general el uso de una aplicación web conlleva un cliente que haga una petición de un recurso y un servidor que responda con ese recurso.

Esta visión es muy general debido a que hoy en día es habitual que tanto el cliente como el servidor tengan que ejecutar algún fragmento de código para el buen funcionamiento de la aplicación web. En este apartado se verán las tecnologías asociadas a las aplicaciones web que se ejecutarán tanto del lado del servidor como del cliente, especificando lo que corresponda en cada uno de los casos.

* **ASP (Active Server Pages):** Las "Páginas Activas" se ejecutan del lado del servidor, de este modo se forman los resultados que luego se mostrarán en el navegador de cada equipo cliente que ha realizado la solicitud. Existen versiones de ASP para Unix y Linux, a pesar de que fue una tecnología desarrollada por Microsoft para la creación dinámica de páginas web ofrecida junto a su servidor IIS.
* **CGI (Common Gateway Interface):** La "Interface Común de Entrada" es uno de los estándares más antiguos en Internet para trasladar información desde una página a un servidor web**.** Este estándares utilizado para bases de datos, motores de búsqueda, formularios, generadores de email automático, foros, comercio electrónico, rotadores y mapas de imágenes, juegos en línea, etc. Las rutinas de CGI son habitualmente escritas en lenguajes interpretados como Perl o por lenguajes compilados como C.
* **CSS (Cascading Style Sheets):** Las "Hojas de Estilo en Cascada" se usan para formatear las páginas web, crear el diseño gráfico de la misma. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas en las que se vincule esa CSS.
* **Java:** Java es un lenguaje eficiente y muy poderoso, que se caracteriza por:
  + Portabilidad: el mismo código (programa) se puede ejecutar en diferentes tipos de máquinas gracias a la Maquina Virtual de Java.
  + Generalmente las aplicaciones en Java son programas independientes, pero también pueden ejecutarse en el servidor (Servlets) o en el navegador web (Applets).
  + Se trata de un lenguaje desarrollado para la metodología de programación orientada a objetos.
* **JavaScript:** Lenguaje que se interpreta y se ejecuta en el cliente. Útil para realizar tareas de interactividad y animación gracias a los controladores de eventos. En las aplicaciones web suele preferirse JavaScript porque es aceptado por muchos más navegadores que VBScript (creado por Microsoft).
* **PHP (Php Hypertext Preprocessor):** Este lenguaje, como ASP, se ejecuta en el lado del servidor. PHP es similar a ASP y puede ser usado en circunstancias similares. Es muy eficiente, permitiendo el acceso a bases de datos empleando servidores como MySQL y, por lo tanto, suele utilizarse para crear páginas dinámicas complejas.
* **VBScript (Visual Basic Scripting):** La respuesta de Microsoft a JavaScript. VBScript es una buena herramienta para cualquier sitio destinado a ser mostrado exclusivamente en el navegador Microsoft Internet Explorer. El código en VBScript programarse para su ejecución en el lado del cliente o en el del servidor.

# Tipos de aplicaciones web

Cualquier proyecto que se quiera desarrollar en Internet, bien sea comercio electrónico, reservas de billetes de vuelo on-line, información meteorológica, registro de usuarios, simuladores de hipotecas, etc… conlleva el desarrollo de una aplicación web. En definitiva, una aplicación web es una plataforma orientada a automatizar los procesos de servicios que se quieran ofrecer a usuarios.

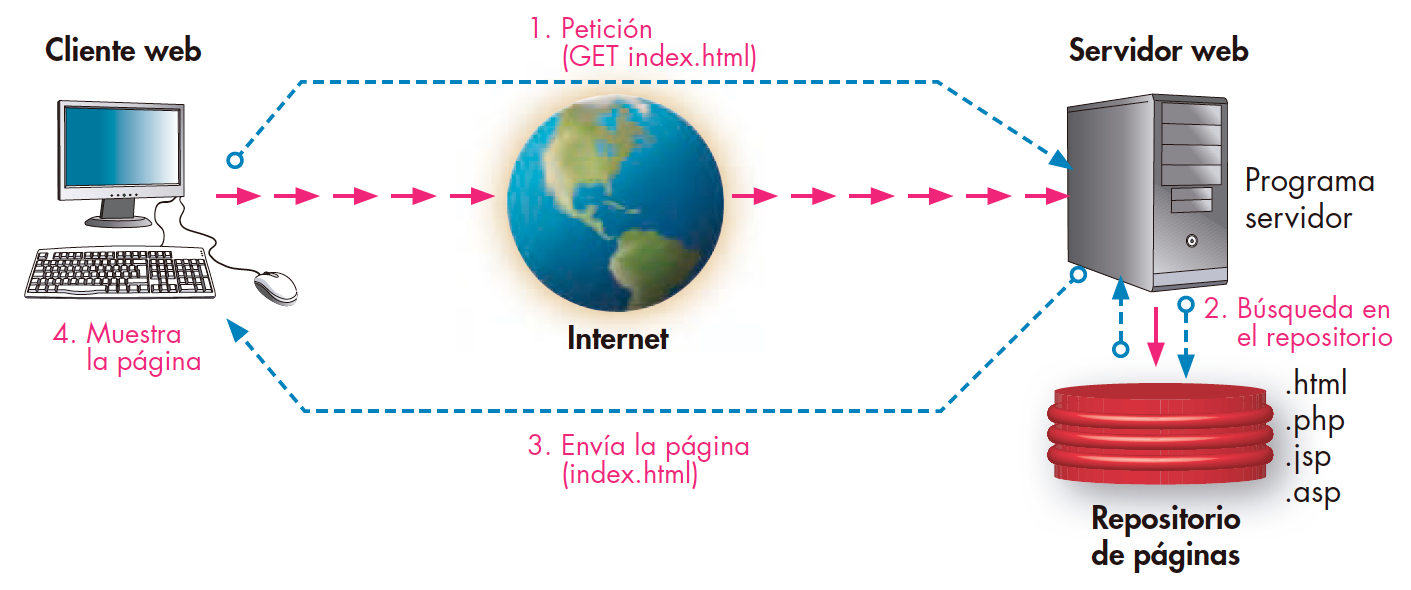


Establecer una clasificación de los tipos de aplicaciones web es una tarea compleja debido a la dificultad existente para poder establecer algún parámetro delimitador en función del cual establecer dicha clasificación junto con la innumerable cantidad de aplicaciones existentes en el actual entorno web 2.0.

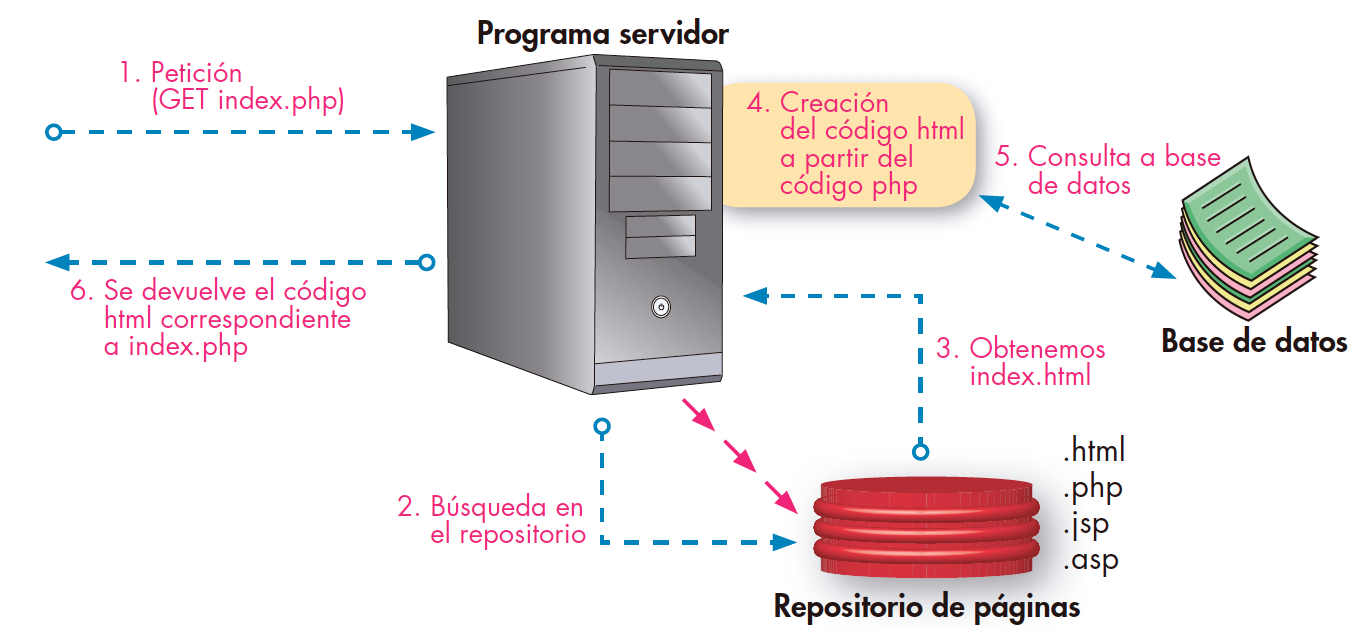
En función de cómo se presenta la aplicación web junto con el contenido que pretende mostrar, se ha establecido la siguiente clasificación:

* **Aplicación web Estática**. Aplicaciones web cuyo contenido rara vez cambia a lo largo del tiempo. Generalmente se desarrollan con el lenguaje HTML, para los estilos se usa CSS y en caso de necesitar animaciones se usa JavaScript. Pueden utilizarse también lenguajes de programación del lado del servidor.
* **Aplicación web Animada**. Se realizan con la tecnología FLASH, permite que una página web presente el contenido con ciertos efectos animados continuados. El uso de esta tecnología permite diseños más vanguardistas, modernos y creativos aunque van en contra de las bases que han creado la web.
* **Aplicación web Dinámica**. Aplicaciones web cuyo contenido varía con el tiempo. Se puede decir que son una ampliación de las aplicaciones web estáticas. Se usan las mismas tecnologías que en las aplicaciones web estáticas y además un lenguaje de programación del lado del servidor, generalmente PHP, ASP o JAVA. Este tipo de aplicaciones web necesitan de un sistema donde los datos puedan ser almacenados para su consulta, una base de datos.
* **Portal**. Aplicación web que en su página principal permite el acceso a múltiples secciones que, por lo general, son foros, chats, cuentas de correo, buscador, acceso registrado para obtener ciertas ventajas, las últimas noticias de actualidad, etc.
* **Tienda virtual o comercio electrónico**. Aplicación web que publica los productos de una tienda en Internet. Permite la compra on-line a través de diferentes medios de pago. Ofrece al administrador un panel de gestión para poder subir los productos, actualizarlos, eliminarlos, etc.
* **Aplicación web con "Gestor de Contenidos**". Se trata de una aplicación web cuyo contenido se actualiza a través de un panel de gestión por parte del administrador del sitio. Este panel de gestión suele ser muy intuitivo y fácil de usar. En aquellas páginas web que requieran una actualización constante, se suele incorporar este panel de gestión para que la web pueda controlarse día a día por parte del cliente.

**Backend y Frontend**



Estructura de una aplicación web estática



Estructura de una aplicación web dinámica

# Arquitecturas web. Modelos

Se puede establecer que la arquitectura de un sitio web comprende los **sistemas de organización y estructuración de los contenidos** junto con los **sistemas de recuperación de información y navegación** que provea el sitio web, con el objetivo de **servir de ayuda a los usuarios a encontrar y manejar la información**.

Centraremos el estudio de los modelos de arquitectura web relacionados, en función de cómo implementan cada una de las capas establecidas en una aplicación web:

* **Capa de presentación** es la encargada la presentación de la web, de la navegabilidad, validación de los datos de entrada, formateo de los datos de salida, etc… Se trata de la capa que se presenta al usuario.
* **Capa de negocio** es la que recibe las peticiones del usuario y desde donde se le envían las respuestas. En esta capa se verifican que las reglas establecidas se cumplen.
* **Capa de acceso a datos (persistencia)** es la formada por determinados gestores de datos que se encargan de almacenar, estructurar y recuperar los datos solicitados por la capa de negocio.

Así las arquitecturas existentes son las siguientes:

**Cliente – Servidor**

En este tipo de arquitectura en cada recurso que se ha de generar se agrupan las tres capas de la arquitectura. En ocasiones se suele separar el código en diferentes ficheros y como si fueran funciones, se llama a esos ficheros para poder crear el recurso, de manera que se obtiene código reutilizable.

**Multinivel**

En las arquitecturas multinivel, cada nivel se cómunica únicamente con los niveles adyacentes. Generalmente en las aplicaciones web se utiliza una arquitectura en tres niveles.

Hay que diferenciar entre multinivel y multicapa, en las dos cada nivel o capa se encarga de realizar una función diferenciada. En multinivel además cada nive se ejecuta en un hardware diferente sin embargo en multicapa, las capas se pueden ejecutar en el mismo hardware.

**Modelo Vista Controlador**

En la arquitectura MVC, del inglés Model-View-Controller, se separa el diseño e interacción del usuario de la lógica de negocio y de la representación de los datos. A priori, podría sonar igual que la arquitectura multinivel o multicapa, pero la característicaque difenrencia a MVC es cómo se comunican sus elementos.

* **Modelo:** Contiene los datos, la lógica de la aplicación y las funciones de esta.
* **Vista:** Contiene la interfaz de la aplicación, o lo que es lo mismo, la parte visible por el usuario y con la que interactúa.
* **Controlador:** Controla la interacción (entrada de datos) y convierte esta entrada en órdenes para el modelo y envía los datos del modelo obtenidos a las vistas para que estas puedan montar la interfaz.

**Cliente**

**Controlador**

**Modelo**

**Vista**

Petición

Respuesta

Consulta

Datos

Datos

HTML

# Plataformas web libres y propietarias

Una plataforma web es el **entorno de desarrollo** de software empleado para **diseñar y ejecutar** un sitio web. En términos generales, una plataforma web consta de cuatro componentes básicos:

1. **El sistema operativo** bajo el cual opera el equipo donde se hospedan las páginas web y que representa la base misma del funcionamiento del computador. Generalmente limita la elección de otros componentes.
2. **El servidor web** es el software que maneja las peticiones desde equipos remotos a través de Internet. En el caso de aplicaciones estáticas, el servidor web simplemente provee el archivo solicitado, el cual se muestra en el navegador. En el caso de aplicaciones dinámicas, el servidor web se encarga de pasar las solicitudes a otros programas que puedan gestionarlas adecuadamente.
3. **El gestor de bases de datos** se encarga de almacenar sistemáticamente un conjunto de registros de datos relacionados para ser usados posteriormente.
4. **Un lenguajes de programación** en el lado del servidor para realizar las tareas requeridas.

Diferentes combinaciones de los cuatro componentes señalados, basadas en las distintas opciones de software disponibles en el mercado, dan lugar a numerosas plataformas web, aunque, sin duda, hay dos que sobresalen del resto por su popularidad y difusión: **LAMP** y **WISA.**

La plataforma LAMP trabaja enteramente con componentes de software libre y no está sujeta a restricciones propietarias. El nombre LAMP surge de las iniciales de los componentes de software que la integran:

* **L**inux: Sistema operativo.
* **A**pache: Servidor web.
* **M**ySQL: Gestor de bases de datos.
* **P**HP: Lenguaje interpretado PHP, aunque a veces se sustituye por Perl o Python.

La plataforma **WISA** está basada en tecnologías desarrolladas por la compañía Microsoft; se trata, por lo tanto, de software propietario. La componen los siguientes elementos:

* **W**indows: Sistema operativo.
* **I**nternet Information Services: servidor web.
* **S**QL Server: gestor de bases de datos.
* **A**SP o ASP.NET: como lenguaje para scripting del lado del servidor.

Existen otras plataformas, como por ejemplo la configuración Windows-Apache-MySQL-PHP que se conoce como **WAMP.** **MAMP** cuando el sistema operativo es Mac. Si se usa MySQL y PHP sobre el servidor de windows se denomina **WIMP**. Estas plataformas son muy comunes pero solo como plataformas de desarrollo, por lo que su uso suele ser en local. De hecho una de las más conocidas y usadas es **XAMP** que está disponible para todas las plataformas, esto lo indica la X del nombre.

En esta asignatura se verá la instalación de diferentes de estas configuraciones, tanto para crear una plataforma de desarrollo local como para el paso final y objetivo de la asignatura que es el hecho de desplegar la aplicación web desarrollada.

# Escalabilidad

Las aplicaciones web se ejecutan en un entorno donde el número de clientes que solicitan el servicio puede variar en gran medida en función del momento. Es por ello que hay una característica de esencial importancia como es la escalabilidad.

En el entorno en que se ubican las aplicaciones web, uno de los principales factores que puede afectar al rendimiento de las mismas es el número de usuarios, ya que éste puede verse incrementado de forma vertiginosa en un periodo de tiempo relativamente corto. El éxito o el fracaso de un sitio web orientado al usuario común vendrá determinado, entre otros aspectos, por el dimensionamiento del sistema sobre el que se instala y soporta el software que sustenta dicho sitio. En consecuencia, uno de los requisitos fundamentales de una aplicación web es que sea completamente escalable sin que un aumento de los recursos dedicados a la misma suponga modificación alguna en su comportamiento o capacidades.

La escalabilidad de un sistema web puede ser:

* **Verticalmente**: de manera ascendente "upgrades" a cada nodo.
* **Horizontalmente**: consiste en aumentar el número de nodos.
* **Cluster:** consiste en crear agrupaciones de servidores.

## Escalabilidad vertical

Habitualmente, la separación lógica en capas se implementa de tal forma que se permita una separación física de las mismas. Interponiendo elementos conectores que actúen de middlewares es posible distribuir la aplicación de forma vertical (una máquina por cada capa del sistema), e incluso si esto no fuera suficiente, distribuyendo los elementos de una misma capa entre distintas máquinas servidoras.

## Escalabilidad horizontal

Se trata de clonar el sistema en otra máquina de características similares y balancear la carga de trabajo mediante un dispositivo externo. El balanceador de carga puede ser:

* **Balanceador Software:** Por ejemplo, habitualmente encontramos un servidor web apache junto con el módulo mod\_jk, que permite la redirección de las peticiones http que a tal efecto sean configuradas entre las distintas máquinas que forman la granja de servidores. Este tipo de balanceadores examinan el paquete http e identifican la sesión del usuario, guardando registro de cuál de las máquinas de la granja se está encargando de servir a dicha sesión. Este aspecto es importante, dado que nos permite trabajar (de cara al diseño de la aplicación) apoyándonos en el objeto sesión propio del usuario y almacenando información relativa a la sesión del mismo, puesto que tenemos la garantía de que todas las peticiones de una misma sesión http van a ser redireccionadas hacia la misma máquina.
* **Balanceador hardware**: Se trata de dispositivos que, respondiendo únicamente a algoritmos de reparto de carga (Round Robin, LRU, etc.), redireccionan una petición http del usuario a la máquina que, según dicho algoritmo, convenga que se haga cargo de la petición. Son mucho más rápidos que los anteriores, dado que se basan en conmutación de circuitos y no examinan ni interpretan el paquete http. Sin embargo, el no garantizar el mantenimiento de la misma sesión de usuario en la misma máquina, condiciona seriamente el diseño, dado que fuerza a que la información relativa a la sesión del usuario sea almacenada por el implementador del mismo, bien en cookies o bien en base de datos.
* **Balanceador hardware http**: Se trata de dispositivos hardware pero que examinan el paquete http y mantienen la relación usuario-máquina servidora. Mucho más rápidos que los balanceadores software, pero algo menos que los hardware, suponen hoy en día una de las soluciones más aceptadas en el mercado.

## Clústers

Con la aparición de los servidores de aplicaciones en clúster se abre una nueva capacidad de escalabilidad que, dependiendo de cómo se aplique, podría clasificarse como vertical u horizontal.

Un clúster de servidores de aplicaciones permite el despliegue de una aplicación web corriente, de forma que su carga de trabajo vaya a ser distribuida entre la granja de servidores que forman el clúster, de modo transparente al usuario y al administrador.

El clúster, mediante el mecanismo de replicación de sesión, garantiza que sea cual sea la máquina que sirva la petición http, tendrá acceso a la sesión del usuario (objeto HttpSession en java). Este tipo de sistemas, debido precisamente a la replicación de sesión, suele presentar problemas de rendimiento.