

Unidad 1. Arquitecturas y tecnologías web

¿Qué es una aplicación web?

- Aplicación informática creada para ser visualizada por un navegador.
- Se trata de una aplicación cliente-servidor:
- Cliente: Los navegadores (contienen intérpretes para lenguajes incrustados en las aplicaciones web (especialmente javascript), como es el caso del motor V8 del navegador Chrome).
- Servidor: Servidor web
- Comunicación: Protocolo HTTP

Ejemplos

- Página de un periódico
- Gmail
- Google maps
- Office 365
- Google Street view
- erMaker: www.jorgesanchez.net/ermaker
- RIA=Aplicaciones de Internet enriquecidas

Ventajas de las aplicaciones web sobre las aplicaciones informáticas de escritorio

- Gran compatibilidad
 - Funcionan en todo tipo de sistemas. Lo único que se requiere es un navegador. En contraposición con las aplicaciones de escritorio creadas para un S.O.
- Requerimientos mínimos en el cliente
 - El único requisito es el navegador (y plugin). En las aplicaciones de escritorio se exige un S.O. RAM, etc..
- Fáciles de crear
- Fáciles de manejar para el usuario
- Facilidad para el mantenimiento. Parches y mejoras instantáneas.
- etc..

Unidad 1

Funcionamiento de una aplicación web

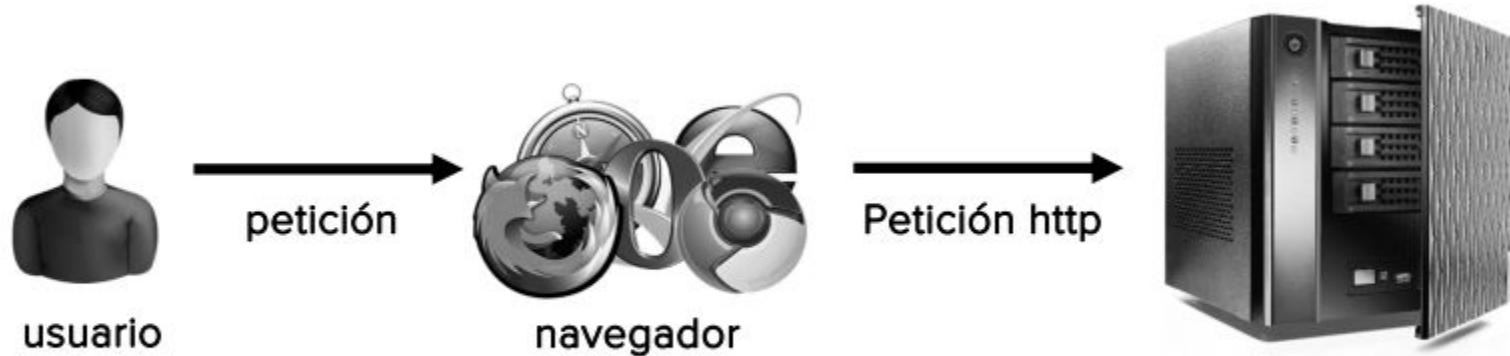
[1.4] Funcionamiento de una aplicación web

- Dos estrategias
- En el lado del cliente (front-end)
 - Elementos que se incorporan junto al lenguaje HTML y que son interpretados por el navegador
- En el lado del servidor (back-end)
 - Aplicaciones creadas con lenguajes y elementos que se interpretan en el lado del servidor que aloja la aplicación.

[1.4.1] Funcionamiento en el lado del cliente

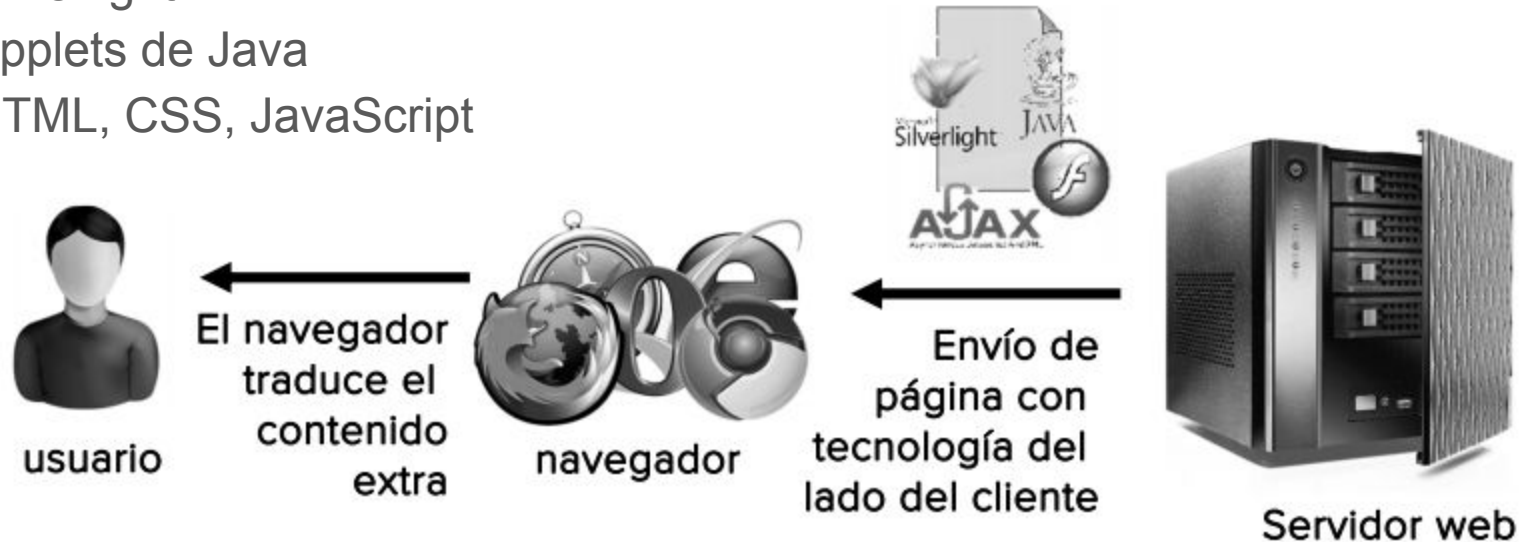
HTML y elementos pertenecientes a otras tecnologías y lenguajes

- CSS, JavaScript, Flash (Adobe), Silverlight (Microsoft), Applets de Java que requieren de un plugin llamado máquina virtual de Java (JVM) en el navegador



[1.4.1] Funcionamiento en el lado del cliente

- Tecnologías del lado del cliente
- Flash
- Silverlight
- Applets de Java
- HTML, CSS, JavaScript



[1.4.1] Funcionamiento en el lado del cliente

- Resumen del lado del cliente
- 1. El cliente realiza una petición de recurso
- 2. Un servidor web entrega el recurso al ordenador del cliente
- 3. El navegador del cliente traduce el código que recibe
- 4. El usuario ve el resultado

[1.4.2] Funcionamiento en el lado del servidor

- Tecnologías y lenguajes del lado del servidor
- Cuando un usuario/a hace una petición a un recurso web, el servidor que contiene dicho recurso se da cuenta de que contiene elementos que interpretar en el lado del servidor y pide al **servidor de aplicaciones** adecuado que traduzca esos elementos antes de enviar el resultado al navegador.



[1.4.2] Funcionamiento en el lado del servidor



[1.4.2] Funcionamiento en el lado del servidor

- El servidor de aplicaciones enviará el resultado al servidor web en un formato traducible por el cliente, es decir un documento HTML. El servidor web finalmente enviará al usuario este resultado.



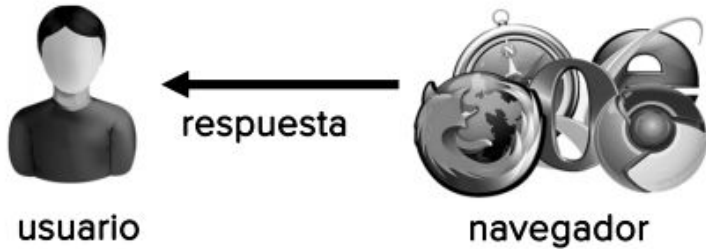
[1.4.2] Funcionamiento en el lado del servidor

- Hoy en día ambos métodos se mezclan de tal manera que los desarrolladores web crean aplicaciones que contienen back-end y front-end.
- Ejemplo: páginas PHP (servidor) que incluyen código JavaScript (cliente)



[1.4.2] Funcionamiento en el lado del servidor

- Los desarrolladores web crean aplicaciones que contienen back-end y front-end. Ejemplo: páginas PHP (servidor, back-end) que incluyen código JavaScript (cliente, front-end- navegador)
- La idea es crear aplicaciones web más ricas, funcionales e interactivas.



Servidor web



Servidor de aplicaciones

[1.4] Funcionamiento de una aplicación web

- Resumen del lado del cliente
 - 1. El cliente realiza una petición de recurso
 - 2. Un servidor web encuentra el recurso y se encuentra con código especial
 - 3. Pide ayuda a otro servidor o servicio para traducir ese código
 - 4. El servidor entrega el código traducido al navegador
 - 5. El navegador interpreta lo que se le entrega
 - 6. El usuario ve el resultado
- NORMALMENTE... Se usan ambas estrategias:
 - Hay contenidos del lado del servidor
 - Hay contenidos del lado del cliente

Unidad 1

Creación de aplicaciones web

[1.5.1] Servidores web

Servidor web: máquina o software capaz de interpretar peticiones web realizadas con el protocolo HTTP o HTTPS y de devolver el resultado de la petición, que suele propiciar la entrega de un recurso alojado en el servidor.

A los servidores web se les llama también servidores HTTP

La mayoría de peticiones HTTP resultan en la entrega de un documento HTML

[1.5.2] Servidores de aplicaciones web

Servidor de aplicaciones, es el encargado de **traducir** instrucciones hechas en **lenguajes del lado del servidor** y entregar el resultado de esa traducción al servidor web (servidor HTTP) que le pidió dicha traducción.

El código del lado del servidor puede contener instrucciones que impliquen acceder a otros servidores (acceso a bases de datos).

Los servidores de aplicaciones trabajan en conjunto con los servidores web para que el proceso se haga transparente para el usuario.

En la práctica los servidores de aplicaciones son módulos software que se añaden al servidor web.

[1.5.2] Servidores de aplicaciones web

Dependiendo de la tecnología del servidor utilizada tendremos diferentes servidores de aplicaciones.

Servidores PHP

Servidores .NET

Servidores Java

etc.

[1.5.3] Arquitectura a tres niveles

Las aplicaciones web actuales utilizan lo que se conoce como arquitectura a tres niveles (*three-tier architecture*). Estas capas son

Capa de presentación

Capa lógica

Capa de acceso a datos

[1.5.3] Arquitectura a tres niveles

Capa de presentación:

Se encarga de la forma de presentar la información al usuario. Maneja la parte de la aplicación web que ve el usuario

Consta del código del lado del cliente (HTML, JavaScript, CSS, Flash..) que le llega al navegador, aunque ese código haya sido generado por una tecnología del lado del servidor.

es la encargada de la navegabilidad, validación de los datos de entrada, formateo de los datos de salida, presentación de la web, etc.; se trata de la capa que se presenta al usuario.

[1.5.3] Arquitectura a tres niveles

Capa lógica: Es la encargada de gestionar el funcionamiento de la aplicación. Recibe las peticiones del usuario y se le envían las respuestas al usuario; en esta capa se verifican que las reglas establecidas se cumplen.

Documentos escritos en un lenguaje que se debe interpretar en el lado del servidor (por ello esta capa está relacionada con el servidor de aplicaciones) cuyo resultado se enviará el servidor web, para que éste, a su vez, lo envíe al cliente que hizo la petición.

La programación en esta capa divide el código en tres partes **MVC**: el **modelo**, el **controlador** y la **vista**. Cada parte se encarga de una acción concreta de la aplicación y así se facilita el mantenimiento de la misma.

[1.5.3] Arquitectura a tres niveles

Capa de acceso a datos

Es la que contiene la información empresarial. Requerimiento: que quede oculta a cualquier persona sin autorización.

En esta capa se encuentra el **sistema gestor de bases de datos (SGBD)** de la empresa, además de otros servidores que proporcionan otros recursos empresariales (servidores de video, audio, certificados..)

Los gestores de datos se encargan de almacenar, estructurar y recuperar los datos solicitados por la capa lógica.

[1.5.3] Arquitectura a tres niveles

Capa de acceso a datos

El servidor de aplicaciones hace peticiones a estos servidores para obtener los recursos de la empresa para cumplir la petición HTTP original. El proceso de acceso a estos recursos queda oculto totalmente al navegador, lo que añade mayor seguridad al proceso.

Los servidores de esta capa entregarán los recursos solicitados al servidor de aplicaciones será el encargado de situarle de forma adecuada en el resultado que viajará hasta el navegador del usuario.

[1.5.3] Arquitectura a tres niveles

Las aplicaciones web actuales utilizan lo que se conoce como arquitectura a tres niveles (*three-tier architecture*). Estas capas son

Todo este mecanismo de trabajo es el que involucra la creación de aplicaciones web.

Los servidores web actuales actúan de servidor de aplicaciones, una vez que se les instala el software pertinente. Por ello cuando hablamos de servidores web también hablamos de servidores de aplicaciones web.

[1.5.4] Programación Back-End y Front-End

División de tareas de las personas encargadas del desarrollo de aplicaciones web en base a lo cerca o lejos que su tarea está respecto al usuario

- Front-End. Parte del desarrollo encargado de producir la apariencia final de la aplicación que verá el usuario.
- Back-End. Parte del desarrollo encargado de realizar la parte de la aplicación que queda oculta al usuario.

[1.5.4] Programación Back-End y Front-End

- Front-End:
- Parte del desarrollo encargado de producir la apariencia final de la aplicación que verá el usuario. En cierto modo es la interfaz del usuario
- Son las personas que diseñan las maquetas o mockups de la aplicación, usando software como **Adobe Photoshop** u otras herramientas más especializadas para la producción de **wireframes** (diseños de líneas que simulan el aspecto final de la aplicación) y **mockups** (maquetas con una mínima funcionalidad)
- También las personas encargadas de HTML, CSS y JavaScript.
- Se encarga del funcionamiento de la capa de presentación. Intenta conseguir una buena experiencia de usuario.

[1.5.4] Programación Back-End y Front-End

- Back-End:
- Parte del desarrollo encargado de realizar la parte de la aplicación que queda oculta al usuario.
- El funcionamiento de la capa lógica y de base de datos. Usa los lenguajes de Servidor y los de Base de Datos.
- Es la parte de la aplicación que se encarga de que funcione debidamente la aplicación.

[1.5.4] Programación Back-End y Front-End

- Back-End:
- Objetivos:
 - Acceso rápido a datos
 - Comunicación eficiente con el navegador
 - Control de la seguridad,
 - Estructurar el código para un fácil mantenimiento
 - Gestión de errores, etc..

[1.5.4.1] Modelo Full-Stack

- Arquitectura por niveles, división en back-end y front-end → Pila de lenguajes → Programadores especialistas back-end y front-end
 - Front-End: HTML, JavaScript y CSS
 - Back-End: por ejemplo PHP, SQL
- Full-Stack: nuevo paradigma de programación en el que el mismo lenguaje, JavaScript, se utiliza en todas las capas.
 - Debido a la aparición de Node.js (Servidor JavaScript muy utilizado como servidor de aplicaciones web) y a la aparición de ciertas bases de datos NoSQL (como MongoDB)

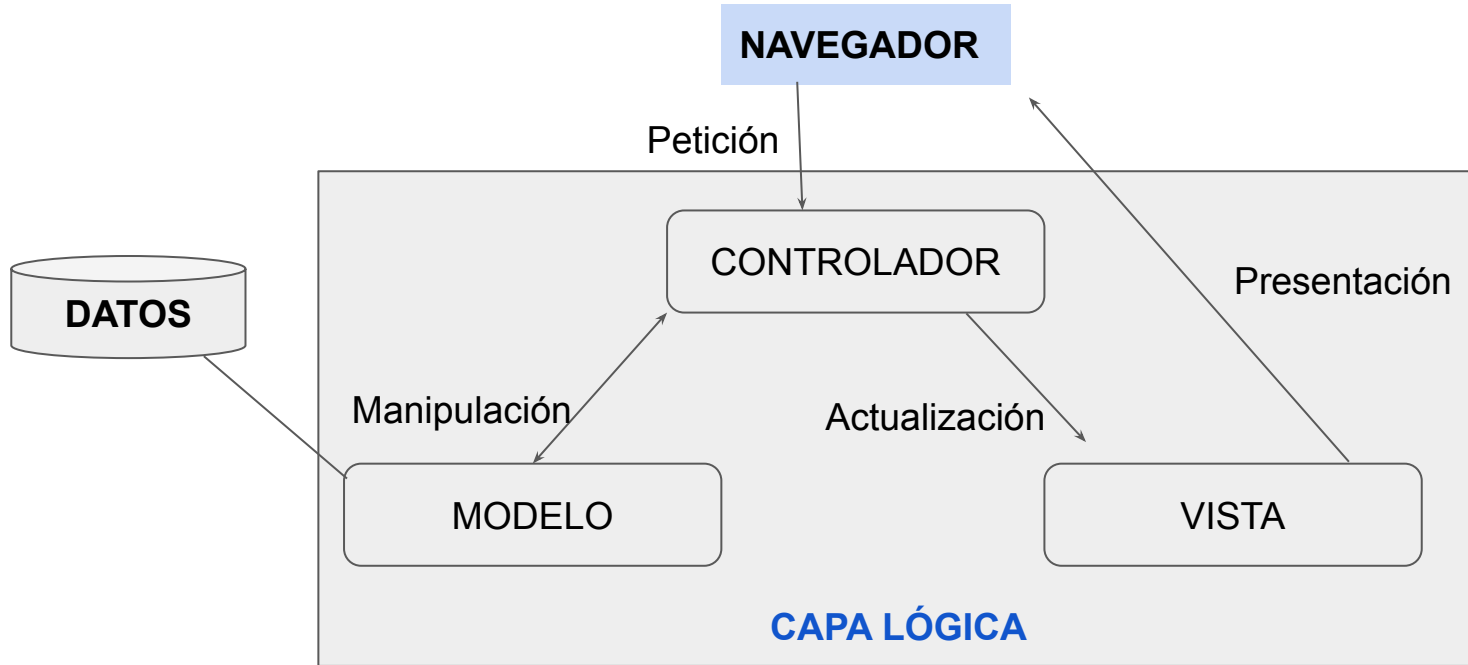
[1.5.4.1] Modelo Full-Satck

- Modelo Full-Stack. Un experto en el lenguaje puede cambiar de de área o nivel y no se desmotiva porque su especialización le relega siempre a la misma tarea. Sus detractores argumentan que los lenguajes especializados están más adaptados a la tarea que realizan y, por lo tanto, la realizan mejor.

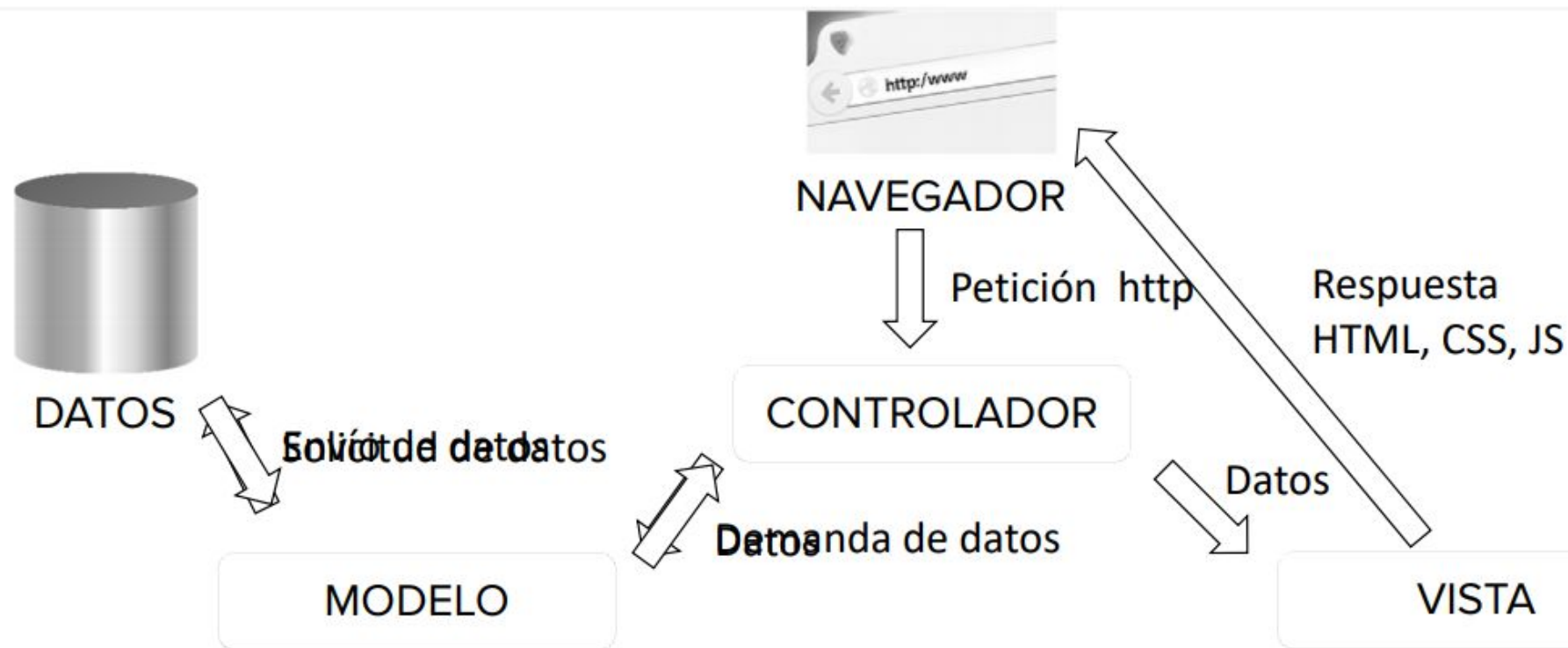
[1.5.5] Paradigma MVC

- Se aplica en la capa lógica.
 - Recuerda! La capa lógica gestiona el funcionamiento de la aplicación. Lado del servidor de aplicaciones (lenguajes de servidor). El resultado lo envía al servidor web (HTTP) y éste lo envía al cliente, al navegador web.
- Paradigma MVC, Separa la programación de la capa lógica en otras tres capas:
 - Modelo. Asocia los datos en la capa lógica con datos de la capa de negocio (SGBD).
 - Vista. Es la encargada de definir la interfaz de usuario. Genera la presentación de cara al usuario.
 - Controlador. Capa encargada de manejar las peticiones del usuario y de comunicar las capas anteriores para que obtengan lo necesario de dicha petición. Capa que captura los eventos de usuario (clic de ratón,..) actuando en consecuencia con las otras capas. Captura eventos, solicita (modelo) los datos y los envía a la capa Vista para que genere el resultado final que llega al usuario.. El controlador es un mediador entre el modelo y la vista.

[1.5.5] Capa Lógica: Paradigma MVC



Paradigma MVC



[1.5.5] Paradigma MVC

- Ventaja de este modelo: División en tareas concretas, facilita el trabajo en equipo y el mantenimiento.

Unidad 1. Arquitecturas y tecnologías web.

Busca información sobre tecnologías utilizadas en el desarrollo de aplicaciones web y describe brevemente las tecnologías más utilizadas

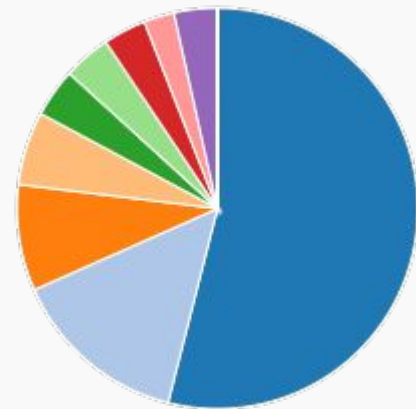
- Clientes (navegadores)
- Servidores
 - Servidores Web
 - Servidores de aplicaciones
 - Sistemas gestores de Bases de Datos
- Desarrollo/Lenguajes
 - Cliente
 - Servidor
 - Frameworks

Unidad 1. Arquitecturas y tecnologías web.

- Clientes (navegadores)

- Un **navegador web** (en inglés, **web browser**) es un [software](#), [aplicación](#) o [programa](#) que permite el acceso a la [Web](#), interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser visualizados.
- La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Además, permite visitar páginas web y hacer actividades en ella, es decir, enlazar un sitio con otro, imprimir, enviar y recibir correo, entre otras funcionalidades más.

Cuota de uso de navegadores web
(junio de 2017, según
StatCounter)¹



Google Chrome (54,00 %)	Safari (14,39 %)
UC Browser (8,57 %)	Mozilla Firefox (5,93 %)
Internet Explorer (3,93 %)	Opera (3,83 %)
Samsung Internet (móvil) (3,47 %)	Android (2,45 %)
Otros (3,49 %)	

Unidad 1. Arquitecturas y tecnologías web.

- Servidores

- Servidores Web

- Un **servidor web** o **servidor HTTP** es un [programa informático](#) que procesa una [aplicación del lado del servidor](#), realizando conexiones con el cliente y generando o cediendo una respuesta en cualquier lenguaje o [Aplicación del lado del cliente](#). El código recibido por el cliente es renderizado por un [navegador web](#). Para la transmisión de todos estos datos suele utilizarse algún [protocolo](#). Generalmente se usa el protocolo [HTTP](#) para estas comunicaciones, perteneciente a la capa de aplicación del [modelo OSI](#).
 - https://es.wikipedia.org/wiki/Servidor_web

Algunos servidores web importantes son:

- [Nginx](#)
- [Apache](#)
- [Internet Information Services](#) (IIS)

Otros servidores, más simples pero más rápidos, son:

- [lighttpd](#)
- [thttpd](#)

Unidad 1. Arquitecturas y tecnologías web.

- Servidores

- Servidores de Aplicaciones

- Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negociación y de acceso a los datos de las aplicaciones.

- Servidores de aplicación Java EE

- Entre los servidores de aplicación Java EE privativos : [WebLogic](#) de [Oracle](#) , [JBoss Enterprise Application Platform](#) de [Red Hat](#) y [WebSphere](#) de [IBM](#). [EAServer](#) de [Sybase Inc.](#) es también conocido por ofrecer soporte a otros lenguajes diferentes a Java, como [PowerBuilder](#). Entre los servidores de aplicaciones libres se encuentran [JOnAS](#) del consorcio [ObjectWeb](#), [Wildfly](#) versión comunitaria de JBoss [Red Hat](#), [Geronimo](#) de [Apache](#), [TomEE](#) de [Apache](#), [Resin Java Application Server](#) de [Caucho Technology](#), [Blazix](#) de [Desiderata Software](#), [Enhydra Server](#) de [Enhydra.org](#) y [GlassFish](#) de [Oracle](#).

- Internet Information Server

- productos no-J2EE. Por ejemplo, con el aumento de la popularidad de [.NET](#), [Microsoft](#) califica a su producto [Internet Information Server](#) como un servidor de aplicaciones. Adicionalmente, se pueden encontrar servidores de aplicación de código abierto y comerciales de otros proveedores; algunos ejemplos son [Base4 Server](#) y [Zope](#).

Unidad 1. Arquitecturas y tecnologías web.

- Servidores

- Sistemas gestores de Bases de Datos

- Las bases de datos más populares y utilizadas para el desarrollo de aplicaciones web son las **bases de datos relacionales**. Hay muchas bases de datos relacionales tanto comerciales como de software libre como MySQL, Derby, Oracle, MS SQL Server, PostgreSQL.
 - Actualmente el sistema gestor de bases de datos más utilizado es MySQL, es multiplataforma y está desarrollado en C, tiene licencia de código abierto GPL y consta de una herramienta interactiva para hacer consultas y crear bases de datos, en su conjunto se ha hecho muy popular en el desarrollo web.
 - se está haciendo hueco una nueva familia de **bases de datos denominadas NoSQL**, entre las que se encuentran Cassandra, mongoDB, riak, redis.

Unidad 1. Arquitecturas y tecnologías web.

- Desarrollo/Lenguajes

- Cliente. Se trata de los elementos que se incorporan junto con el código HTML y que necesitan ser interpretados en el navegador del usuario
 - Los lenguajes de programación del lado cliente se usan para su integración en páginas web. Un código escrito en un lenguaje de script se incorpora directamente dentro de un código HTML y se ejecuta interpretado.
 - Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje [javascript](#) y [java](#), aunque pueden añadirse más lenguajes mediante el uso de [plugins](#).
 - CSS, javascript, flash, silverlight, applets de java

Unidad 1. Arquitecturas y tecnologías web.

- Desarrollo/Lenguajes

- Servidor. Se trata de aplicaciones creadas con lenguajes y elementos que se interpretan en el servidor que aloja la aplicación.
 - el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.
 - Las aplicaciones del lado del servidor están escritas mediante algún lenguaje de programación: Perl
 - Python
 - Ruby
 - Java
 - C#
 - JavaScript

Unidad 1. Arquitecturas y tecnologías web.

- Desarrollo/Lenguajes

- Frameworks,

- Un Framework es un marco de trabajo, se puede traducir como estructura. Recomendamos una estructura concreta de trabajo que separe los diferentes elementos de la aplicación (MVC, archivos de configuración, etc.)
 - Ruby on Rails, Apache Struts (aplicaciones J2EE), Spring (J2EE), Django (Python), Zend (PHP), Angular.js (javascript, creado por Google), Backbone.js (javascript)

[1.6.5] Frameworks MVC

- Marco completo de trabajo (basado en MVC), semejante a las plataformas empresariales
- En inglés **framework** se puede traducir como estructura. En el sentido que nos ocupa, un framework sería un marco de trabajo.
- MVC (modelo-vista-controlador), paradigma de programación de aplicaciones que separa la misma en tres visiones diferentes.
- Todos los frameworks (al igual que las plataformas de desarrollo) imponen, o al menos recomiendan, una estructura concreta de trabajo que separa los diferentes elementos de la aplicación; no solo el modelo, la vista y el controlador, sino los archivos de configuración, elementos estáticos, desvío o enrutamiento de las peticiones HTTP...

[1.6.5] Frameworks MVC

Los frameworks MVC más populares actualmente (I)

- Ruby on Rails, programación fácil y buenos resultados visuales. Se puede ejecutar en casi cualquier servidor web, basta con instalar el componente correspondiente. El lenguaje en el que se basa es Ruby.
- Apache Struts, para la creación de aplicaciones J2EE. Muy preparado para utilizar con servidores web Apache
- Spring, para la creación de aplicaciones J2EE. Tiene una versión para aplicaciones .NET

[1.6.5] Frameworks MVC

Los frameworks MVC más populares actualmente (II)

- Django. Escrita en Python y pensada para utilizar en ese lenguaje
- Zend. Escrito para PHP
- Angular.js Marco de trabajo para JavaScript creado por Google
 - Aplicaciones web SPA de página única (**SPA, Single Page Applications**) permite que la página vaya mutando para ir cambiando solo los elementos de la misma que se requieren. Su paradigma es una variante de MVC conocida como **MVVM (Modelo-Vista-Vista-Modelo)**. Es el marco que más crece.
- Backbone.js Marco de trabajo que utiliza JavaScript como único lenguaje. Es uno de los primeros que se crearon para JavaScript.

Estado actual de uso

- <http://w3techs.com/technologies>
 - Lado servidor
 - Lado cliente

Unidad 1

[1.3] La WEB 1.0, la WEB 2.0 y la WEB 3.0

[1.3] La WEB 1.0, la WEB 2.0 y la WEB 3.0

- Las primeras páginas web eran estáticas.
 - Texto, imágenes e hipervínculos
- Poco a poco se integraron animaciones, vídeo, sonido, juegos, aplicaciones ofimáticas..

La WEB 2.0

- 3 pilares
- RIA: APLICACIONES RICAS DE INTERNET
 - Aplicaciones web que ofrecen resultados que las hacen casi indistinguibles de las aplicaciones de escritorio.
 - Google Docs, Office365, Google Maps
- SOA: ARQUITECTURA ORIENTADA AL SERVICIO
 - Modelo de programación: **Software as a Service (Software como Servicio o SaaS)**
- WEB SOCIAL.

La WEB 2.0

- SOA: ARQUITECTURA ORIENTADA AL SERVICIO
 - Conjunto de tecnologías y técnicas que permiten diseñar aplicaciones como un conjunto de servicios que resuelven peticiones de los usuarios.
 - Se crean pequeños elementos software reutilizables de forma independiente respecto al lenguaje con el que fueron creados.
 - Cualquier componente se diseña para ser utilizado como un servicio web.
 - Ejemplo: software de opiniones de los usuarios que se reutilizan, empresas que utilizan los servicios de google maps
 - Modelo de programación: **Software as a Service (Software como Servicio o SaaS)**

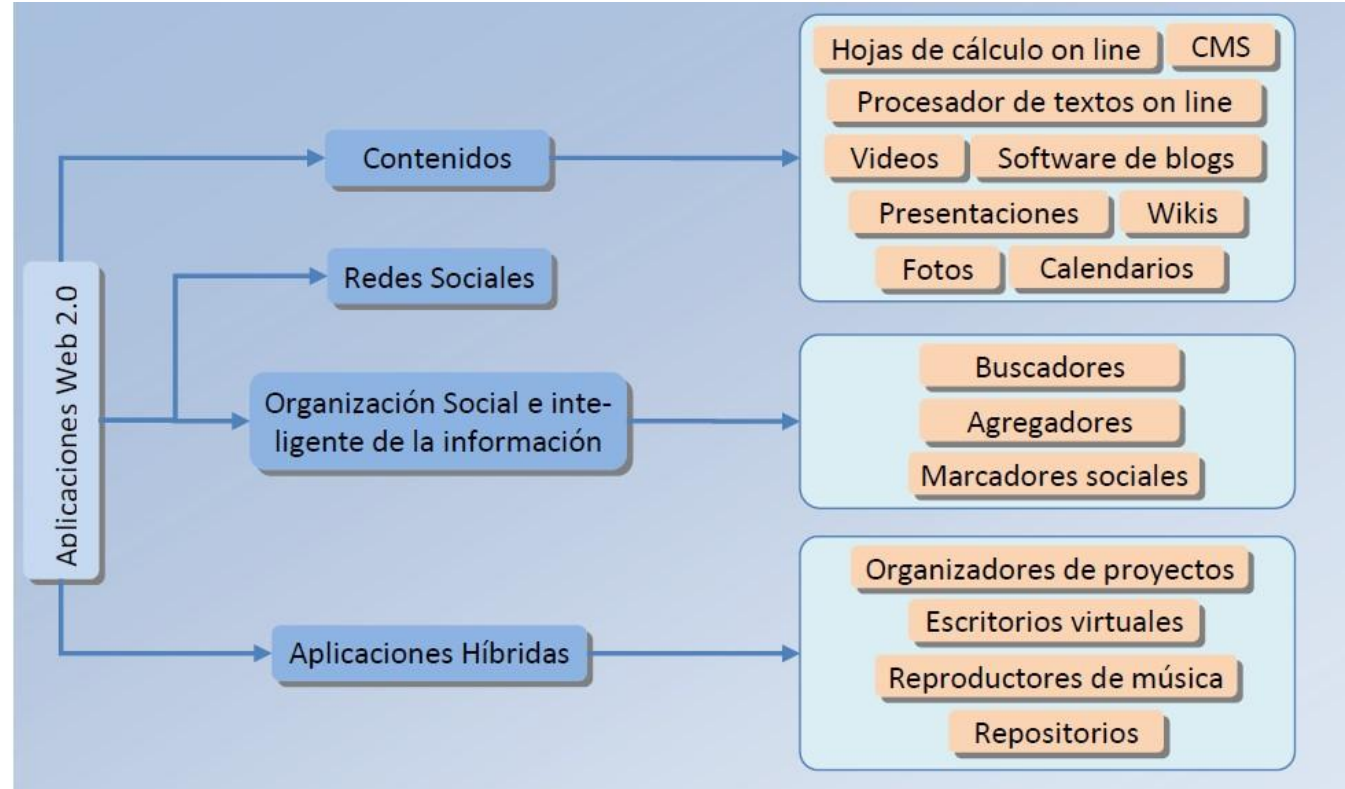
La WEB 2.0

- WEB SOCIAL.

- El usuario pasa a ser el centro de de las aplicaciones web 2.0: Twitter, facebook, Instagram
- Usuario activo, los contenidos ya no pertenecen solo a la entidad propietaria del sitio web, sino a los propios usuarios. El contenido escapa al control de de la entidad
- Pasa a ser imprescindible la conexión de las aplicaciones con los principales servicios de red social en Internet.
- 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.

La WEB 2.0

- WEB SOCIAL.



La WEB 3.0

- Año 2010 aproximadamente
- Guerra de navegadores, se dota de mayor potencia y prestaciones al lado del cliente
- Programadores front-end
- Pilares de la web 3.0
 - Web Inteligente. Inteligencia artificial aplicada a la web
 - Web Semántica. Los contenidos en la web disponen de etiquetado o meta datos que permiten darles significado. Cuanto mejor es ese significado mejor podemos analizar los datos.
Lenguaje XML y sublenguajes RDF y OWL

La WEB 4.0

- 2016
- La Web 4.0 permite adelantarse a situaciones cotidianas, como ver que llegas tarde al trabajo y enviar un mensaje avisando de lo que sucede, tomar una ruta alternativa a la habitual porque está colapsada, etc. Si tu smartwatch detecta que el ritmo cardíaco es elevado y tienes registrado que sufres problemas de corazón, avisará a la asistencia sanitaria y aparecerá una ambulancia.

La WEB 4.0

- 2016
- La web 4.0 ofrece soluciones a partir de toda la información que le damos y que existe en la web. Para lograrlo, se fundamenta en cuatro pilares:
 - La comprensión del lenguaje natural hablado, escrito y tecnologías (de voz a texto y viceversa)
 - Nuevos sistemas de comunicación máquina a máquina (M2M)
 - Uso de la información del contexto. Por ejemplo, ubicación que aporta el GPS, ritmo cardíaco que registra el smartwatch, etc.
 - Nuevo modelo de interacción con el usuario.