

# Implantación de aplicaciones web

2º Grado Superior de ASIR

# UD1. Conceptos sobre aplicaciones web

# 1.1. Evolución de la web e Internet

- El éxito de Internet es algo ligado a la web, llegando incluso a que la mayoría de las personas no sea capaz de distinguirlas.
- Todo se realiza en la Web, incluso tareas que antes requerían de protocolos distintos de http.
- Ahora se realizan mediante servicios accesibles desde el protocolo http, usando un cliente de navegación web.
- En su comienzo, en Internet no había páginas web, se utilizaba para servicios como el correo electrónico, la transmisión de archivos o grupos de noticias.
- No había World Wide Web.

# 1.1. Perspectiva histórica

- Los servicios se realizaban mediante software cliente especializado, produciendo que el cliente tuviera muchas aplicaciones web, que no eran sencillas de usar.
- Para el correo electrónico hacía falta un cliente de correo, la transmisión de ficheros se realizaba mediante un software cliente FTP, los foros se leían a través de gestores de noticias y para chatear hacía falta una aplicación especial.
- Hoy en día estos servicios se pueden utilizar de esa forma hoy en día, como puede ser el software ThunderBird u Outlook. La mayoría de los usuarios realizan todo esto a través de: La Web.

# 1.1. Perspectiva histórica

---

- Las razones para el éxito de la web son:
  - Su manejo es sencillo.
  - Es muy visual.
  - Solo se necesita un único software.



# 1.1. Perspectiva histórica

- El cambio en las páginas web se debió a que los usuarios demandaron que diferentes servicios que inicialmente no tenían: se buscó que no mostraran únicamente un texto estático, sino que también se incluyeran imágenes, videos, animaciones,...
- Poco a poco se añadió el acceso a bases de datos, manejo de correo electrónico, transmisión de ficheros, compra y venta de productos, etc.
- Hoy en día desde la web se puede hacer cualquier tarea.

# 1.1. Perspectiva histórica

- El problema con este desarrollo de la web es el requerimiento de nuevas componentes (plugins) en el navegador.
- La máquina virtual de Java, el plugin de Flash o un intérprete veloz de JavaScript se han vuelto elementos indispensables para el desarrollo correcto de las páginas web, exigiéndose la instalación a los usuarios.
- Los plugins han pasado a ser un problema menos, debido a que los navegadores ya traen los mínimos necesarios para su funcionamiento. Contenidos hechos en Flash se van sustituyendo por tecnologías equivalentes compatibles con HTML5.

## 1.2. Historia de la Web

- La creación de aplicaciones web ha tenido numerosos cambios a lo largo de la historia de Internet.
  - **1989.** Tim Bernes Lee, intenta trasladar el hipertexto a los documentos científicos, mediante el cual es posible avanzar de un documento a otro mediante enlaces existentes en el propio texto. Teorizó la forma de transportar este tipo de documentos y sobre el lenguaje de marcas a utilizar.
  - **1990.** Aparece el primer navegador web, llamado WorldWideWeb, para realizar pruebas se creó para un ordenador Next.
  - **1991.** Tim Bernes acude a un grupo de discusión en Internet para discutir sobre cómo implementar el hipertexto de forma más conveniente.



## 1.2. Historia de la Web.

- **1992.** Pei-Yuan Wei crea ViolaWWW, considerado el verdadero primer navegador, era capaz de interpretar un lenguaje de script además del propio HTML.
- **1993.** El CERN anuncia que la web será libre para todo el mundo. Renuncia a sus posibles patentes.

La NCSA se interesa por la web y crea el primer navegador exitoso: Mosaic.

Lou Montulli desarrolla Lynx para los sistemas Unix, el primer navegador de texto en la web. Fue muy utilizado en los años posteriores a su creación, hasta la aparición de los navegadores gráficos.

Se empieza a hacer popular la etiqueta img, las imágenes empiezan a poblar la web.

Aparece el primer borrador de HTML gracias a Tim Bernes-Lee y Dan Conely.

## 1.2. Historia de la Web

- **1994.** Conferencia global sobre la web.

La IETF asigna un grupo de trabajo para estandarizar HTML.

El lenguaje HTML empieza a ser complicado debido a la aparición de numerosas etiquetas inventadas por cada entidad privada. Dan Connolly recopila las etiquetas más utilizadas y se crea el borrador de HTML 2.

Marc Andreessen y Jim Clark abandona la NCSA y fundan Mosaic Communications (futura Netscape).

Guido Van Rossum crea Python.

A finales del año se crea la World Wide Consortium.

Mosaic Communication se convierte en Netscape Communications y lanza el navegador Netscape Navigator.

A finales de año hay 10000 servidores web.

## 1.2. Historia de la web

- **1995.** Aparece CGI, Common Gateway Interface.

Aparece Windows 95.

Se crea el primer borrador de HTML 3, incluyendo tipos de letra y otras mejoras.

Microsoft crea Internet Explorer.

El World Wide Consortium queda como principal organismo de estandarización de HTML.

Hakom Wum Lie crea el navegador Opera.

A finales de año aparecen los primeros elementos de creación de hojas de estilo, raíz del lenguaje CSS.

Sun Microsystems crea el lenguaje Java.

## 1.2. Historia de la Web

- **1995.** Se lanza el servidor http de código abierto.

Los hermanos Allaire crean ColdFusion, un lenguaje de script sobre HTML que se ejecuta en el servidor que aloja las páginas web de modo que el cliente no necesita tener un software especial que reconozca esta tecnología.

Netscape desarrolla JavaScript.

Aparece la primera versión de MySQL.

Internet se empieza a hacer muy popular y la web se empieza a convertir en el servicio más habitual para acceder a Internet.

Se crea Geocities, antecesor de las redes sociales.

## 1.2. Historia de la Web

- **1996.** Se crea el HTML ERB. Es una reunión trimestral para ayudar en el estándar.

Se crea Yahoo!.

Rasmus Lerdorf crea el lenguaje script del lado del servidor PHP.

La empresa Macromedia lanza el software Flash.

La W3C estandariza CSS1 y su popularidad aumenta rápidamente.

Se lanza Hotmail.

- **1997.** Se produce el Escándalo Lewinsky.

Aparece la especificación estándar de HTML 3.2

Sun Microsystems crea Java Servlets y Microsoft crea el lenguaje ASP.

Versión 4 de Internet Explorer y Netscape Navigator.

## 1.2. Historia de la Web

- **1998.** La W3C publica los estándares de HTML 4.0 y CSS2.  
La combinación HTML, JavaScript y CSS se conoce desde este año como DHTML.  
Aparece XML 1.0 por parte de W3C.  
Netscape crea la fundación Mozilla para mejorar el código del navegador Netscape Navigator.  
Se funda la Web Standards Project, fundación encargada de promover el uso de estándares HTML.
- **1999.** Sun crea JSP y la plataforma de trabajo J2EE.  
El navegador Internet Explorer empieza a dominar el mercado.  
Se crea el test Acid 1.  
Aparece RSS, un formato de contenido basado en XML que permite la sindicación de contenidos.  
La W3C presenta HTML 4.01.  
Microsoft crea para Internet Explorer XMLHttpRequest.

## 1.2. Historia de la web

- **2000.** Aparece Wikipedia.

Aparece el estándar W3C XHTML LO, versión de HTML basado en XML que pretende derrocar a HTML.

ISO (organismo internacional de estándares) publica la norma ISO 15445.

Roy T. Fielding publica su tesis doctoral de la que aparece la idea de las webs de tipo REST, Representational State Transfer (Estado Representativo de Transferencia), o RESTful, que consiste en que las direcciones (URL) de las aplicaciones web en Internet representen el servicio al que dan acceso.

Se lanza Konqueror 2.0 para el sistema Linux con ventanas KDE navegador basado en el motor KHTML raíz del motor Webkit en el que se basarán los navegadores Safari y Chrome entre otros.

## 1.2. Historia de la Web

- **2001.** Aparece Explorer 6, integrado en el sistema operativo Windows XP.  
La W3C lanza el estándar XHTML 1.1.  
PHP y Flash son las tecnologías dominantes para crear aplicaciones web enriquecidas.  
Los sucesos del 11S provocan el colapso de Internet.  
Se produjo el estallido de la burbuja punto com. Se habla de redefinir la web, produciéndose el inicio de la Web 2.0.  
Se populariza el uso de blogs.  
Se crea Drupal, herramienta de gestión de contenidos que permite crear sitios colaborativos.  
Se crea Bit Torrent, siendo un protocolo para compartir archivos entre equipos sin servidor central.  
Napster es obligada a cerrar debido a varios juicios contra las leyes de autor que perdió.



## 1.2. Historia de la Web.

- **2002.** La fundación Mozilla recoge el código liberado por Netscape y crea su propio navegador FireBird.

Microsoft crea la plataforma .NET.

Apple crea una bifurcación del proyecto KHTML y lo llama Webkit.

Aparece MySpace.

- **2003.** Apple lanza al mercado Safari.

Se crea Wordpress, gestor de contenidos web CMS.

Chris Pederik crea la Web Developer Toolbar para Firefox.

Aparece la red social gráfica y virtual Second Life.

Se crean LinkedIn y Xing.

## 1.2. Evolución de la Web

- **2004.** Se comercializa el navegador Firefox.  
Se forma el WHATWG para conseguir un HTML versión 5.  
Aparece Facebook.  
Aparece Gmail.
- **2005.** AJAX se populariza pasando a ser de las tecnologías fundamentales para crear páginas web dinámicas.  
Se incorporan patrones MVC a la mayoría de las tecnologías de creación de aplicaciones web.  
Se lanza “Joomla!”.  
Se lanza el framework Ruby on Rails.  
Aparece el test Acid2.  
Aparece Google Maps y Youtube.

## 1.2. Evolución de la Web

- **2006.** Joe Hewitt lanza Firebug.  
John Resig crea jQuery.  
Tras el éxito de Mozilla Firefox, Microsoft lanza Internet Explorer 7.  
Aparece el sitio de microblogging, twitter.  
Google compra Youtube.
- **2007.** Google lanza Street View como complemento de Google Maps.  
Se crea Tumblr.  
Se crea SoundCloud.  
Apple comercializa el primer iPhone.  
Google presenta el sistema Android.

## 1.2. Evolución de la Web

- **2008.** Aparece Google Chrome recrudeciendo la guerra de los navegadores.  
Aparece el primer borrador de HTML 5.  
Aparece el test Acid 3.  
Aparece Spotify.  
Se crea Dropbox.  
El modelo de computación en la nube comienza a imperar.  
La empresa Nitobi desarrolla PhoneGap.
- **2009.** Satoshi Nakamoto Crea la criptomoneda digital Bitcoin.  
Se funda el sitio crowdfunding Kickstarter.  
Ryan Dahl crea Node.js  
Aparece MongoDB.  
Google lanza Angular.js.

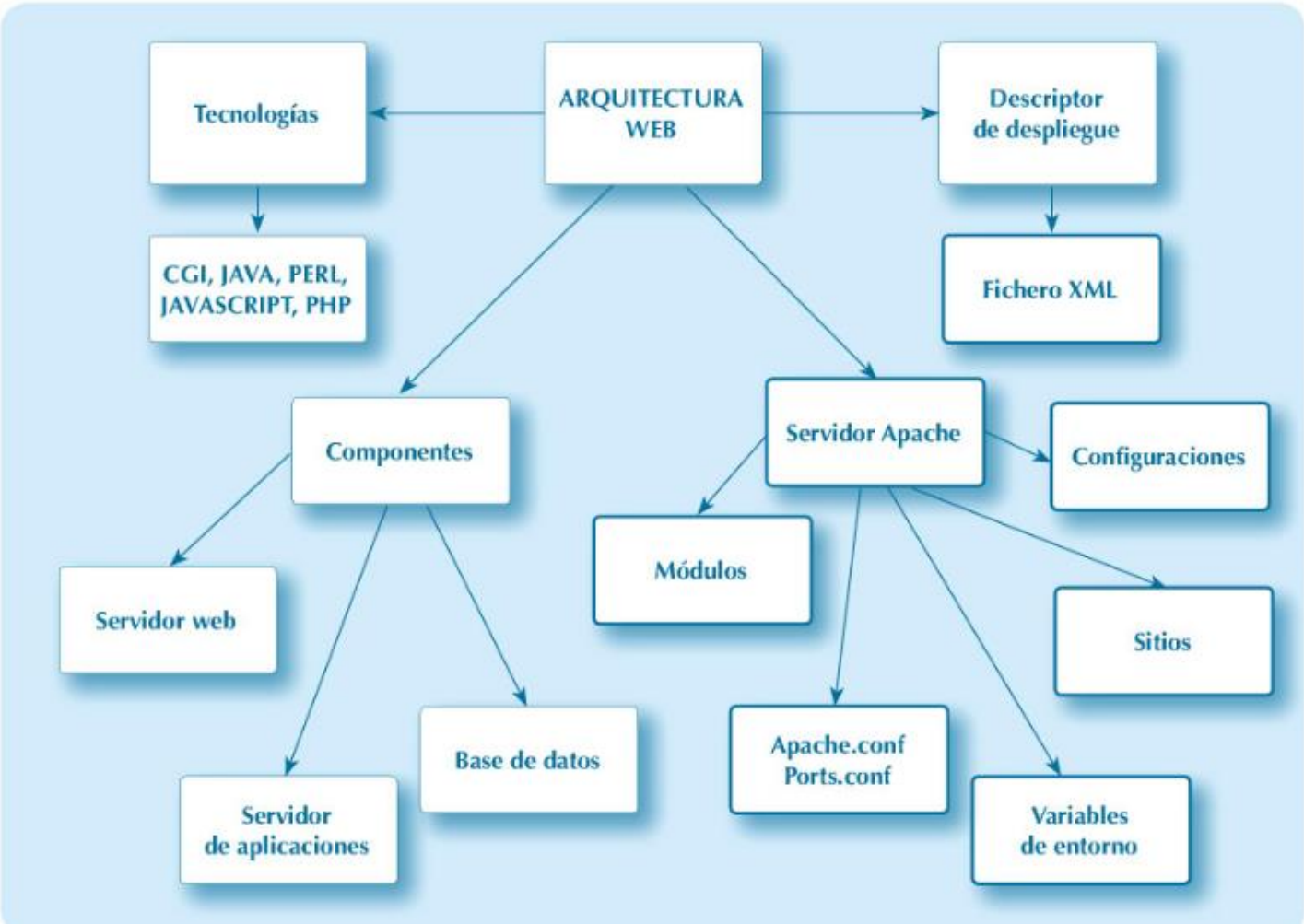
## 1.2. Evolución de la Web

- **2010.** Se produce el crecimiento del uso de los smartphones.  
Apple lanza el iPad,  
China tiene 460 millones de usuarios de Internet.  
Myanmar se queda sin acceso a Internet.  
Netflix empieza a dominar el mercado del video en streaming.
- **2011.** La W3C presenta la norma CSS 2.1.  
HTML 5 comienza a adoptarse por la mayoría de los desarrolladores web y  
Flash empieza a dejar de utilizarse.  
En China ya se navega más por dispositivos móviles que desde ordenadores de sobremesa.  
Se produce un aumento intenso en el uso de twitter y Facebook.

## 1.2. Evolución de la Web.

- **2012.** Los usuarios de Internet consiguen paralizar las leyes antipiratería en EEUU.  
Microsoft lanza Windows 8.
- **2013.** Yahoo! Compra Tumblr.  
Mozilla lanza el sistema operativo móvil FirefoxOS.  
Google lanza las Google Glasses.  
Las tecnologías relacionadas con el Big Data comienzan a proliferar.
- **2014.** Facebook compra Whatsapp.  
Apple presenta el reloj Apple Watch.  
Ali baba bate record de venta de acciones en Wall Street.  
Sony fue hackeada y la empresa se vio obligada a retrasar el estreno de la película The Interview debido a chantajes de hackers norcoreanos.

## 2. Aplicaciones Web



## 2. Aplicaciones Web

- ¿Qué es una aplicación web?

Una aplicación web es una aplicación creada para ser ejecutada por un navegador.

En la actualidad, todos los usuarios poseen y manejan navegadores para recorrer Internet y están muy acostumbrados a trabajar con ellos.

Las aplicaciones web se crean en HTML y sus tecnologías asociadas (CSS, JavaScript,...).

Ejemplos: Gmail, Google Maps, Office 365,...



## 2.1. Ventajas de las aplicaciones web

- Gran compatibilidad
- Requerimientos mínimos en cliente
- Facilidad de creación y mantenimiento
- Fácil manejo para el usuario
- Datos centralizados
- Costes reducidos en su implantación
- Accesibles desde diferentes máquinas y ubicaciones
- No hay instalación

## 2.2. Desventajas de las aplicaciones web

- Menos potentes
- Aprovechan peor el hardware
- Se requiere conectividad
- Las aplicaciones web son más difíciles de crear
- Delegación del control de nuestra información

## 1.4. Tipos de aplicaciones web

- ESPECÍFICAS

Son desarrolladas para gestionar un problema específico

- CMS

Sistemas de gestión de contenido. Son aplicaciones desarrolladas por un tercero que nos ofrecen una funcionalidad general. Ejemplos: blogs, portales, wiki,...

Los más conocidos son Wordpress, Joomla, Moodle...

### 3. La Web 1.0, La Web 2.0 y La Web 3.0

- Las primeras aplicaciones web eran poco más que un conjunto de páginas que servían información estática a los usuarios sin llegar a interaccionar con ellos prácticamente.
- La primera web constaba de páginas que eran simplemente un conjunto de textos e imágenes, junto con los hipervínculos que permitían saltar de unos contenidos a otros.
- Con el paso del tiempo se buscó la integración de nuevos contenidos como eran animaciones, videos, sonido... Aumentando la aparición de estas páginas tras el estallido de la burbuja punto com.

## 3.1. La Web 2.0

- Su primera denominación fue dada en 2004 por Jeff Bezos en una conferencia.
- Dentro de la Web 2.0 se puede decir que hay tres pilares que la forman:
  - Aplicaciones ricas de Internet.
  - SOA.
  - Web social

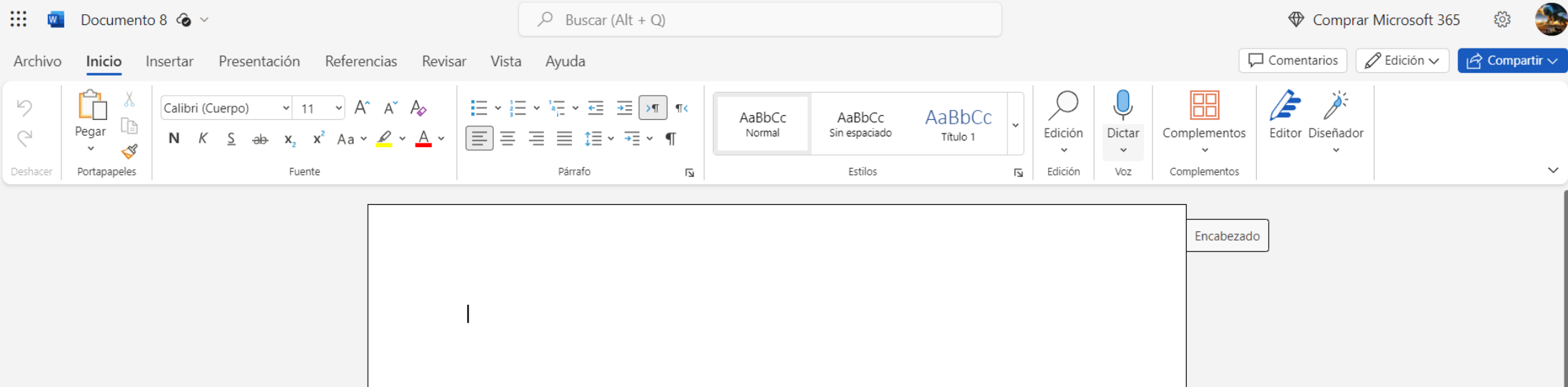
## 3.1. La Web 2.0

### **RIA: APLICACIONES RICAS DE INTERNET**

Son aquellas aplicaciones que ofrecen resultados que las hacen casi indistinguibles respecto a las aplicaciones de escritorio.

Las primeras aplicaciones web no permitían prácticamente la interacción usuario/ máquina.

Debido a la mejora en los navegadores, convirtiéndose en un software compilador de aplicaciones, se pueden traducir lenguajes que permiten la interactividad e incluso incluir medios más atractivos.



## 3.1. La Web 2.0

### **SOA: ARQUITECTURA ORIENTADA AL SERVICIO**

Otra clave en el desarrollo de las aplicaciones web 2.0 es el paradigma conocido como SOA, que agrupa el conjunto de tecnologías y técnicas que permiten diseñar aplicaciones como un conjunto de servicios que resuelven peticiones de los usuarios.

Cualquier componente software se diseña para ser utilizada como un servicio en la red, cuyo trabajo es recibir peticiones y entregar los resultados correspondientes a la misma



## 3.1. La Web 2.0

- Una aplicación web se puede modelar como si fuera un conjunto de piezas que se comunican entre sí.
- Esto es lo que ha dado lugar a un nuevo modelo de programación: Software as a Service (SaaS).
- La idea final es que las aplicaciones no estén pensadas para ser instaladas en el ordenador del cliente, sino que se coloca en un servidor de Internet al que accedan los clientes.

# 3.1. La Web 2.0

## **WEB SOCIAL**

Es la parte más entendible de la web 2.0. El usuario pasa a ser el centro de las aplicaciones web 2.0, se conecta con otros usuarios a través de la aplicación.

El usuario ha dejado de ser pasivo y los contenidos ya no pertenecen a una entidad del propietario, sino a los propios usuarios.

Los contenidos se escapan del control de la entidad. La empresa busca con esto publicitar su aplicación con el uso de usuarios activos y entusiastas que la publiciten.

Se convierte en imprescindible la conexión de las aplicaciones con los principales servicios de red social en internet.

## 3.2. La Web 3.0

- En la nueva guerra de navegadores entre Internet Explorer, Mozilla Firefox y Google Chrome principalmente ha dotado de una mayor potencia al lado del cliente, lo que producido un crecimiento de las prestaciones disponibles para los programadores front-end.
- Se han formado los considerados pilares de la web 3.0:
  - Web inteligente
  - Web semántica

## 3.2. La Web 3.0

- **Web inteligente**

Se refiere a la inteligencia artificial aplicada a la web; es decir, a la capacidad de que la propia web reaccione de forma inteligente ante los usuarios. Un ejemplo es los productos que nos muestra una aplicación porque ha detectado que nos gusta algo en concreto.

- La web inteligente utiliza las siguientes áreas de la ciencia de la programación:
  - **Procesadores de lenguaje natural (NLP)**
  - **Aprendizaje automático (Machine Learning)**
  - **Sistemas de recomendación**

## 3.2. La Web 3.0

### **Web Semántica**

Se trata de que los contenidos en la web dispongan de etiquetado o meta datos que permitan darles significado. Cuanto mejor es ese significado, mejor podremos analizar esos datos. El lenguaje XML y especialmente los sublenguajes RDF y OWL son los más empleados para dar este sentido semántico a los datos de las aplicaciones web. Para muchos analistas web 3.0 y web semántica son términos sinónimos.

## 4. Funcionamiento de una aplicación web

Al crear aplicaciones web se pueden utilizar diversos lenguajes y tecnologías. Hay dos estrategias:

- **Lenguajes y tecnologías en el lado del cliente:** se trata de los elementos que se incorporan junto al código HTML de una aplicación web y que necesitan ser interpretados en el navegador del usuario.
- **Lenguajes y tecnologías en el lado del servidor:** En este caso se trata de aplicaciones creadas con lenguajes y elementos que se interpretan en el servidor que aloja la aplicación.

## 4.1. Funcionamiento en el lado del cliente

- La página web, albergada en el servidor web, contiene elementos pertenecientes a otros lenguajes y tecnologías. Las principales son:
  - CSS. El lenguaje de las hojas de estilo. Permite modificar la apariencia de una página web.
  - JavaScript. Lenguaje de programación sencillo pero potente, cuyo código se puede incluir directamente dentro de una página web.
  - Flash. Componente que se suele incrustar mediante una etiqueta HTML de tipo object que hace referencia a un archivo de tipo SWF que es el que contiene el código Flash.

## 4.1. Funcionamiento en el lado del cliente

- Silverlight. Componente, semejante en comportamiento a Flash, que requiere también de un plugin.
- Applets de Java. Se incrusta mediante una etiqueta HTML de nombre applet y hace referencia a un archivo que contiene código Java.
- Generalmente el navegador del usuario ha de tener instalados varios plugins que le permitan interpretar dichas tecnologías.
- Se habla de tecnología web de cliente pesado cuando se refiere a una aplicación web que hace un uso intensivo de componentes del lado del cliente.
- En la actualidad triunfan las tecnologías que pertenecen al estándar HTML5, en especial JavaScript.

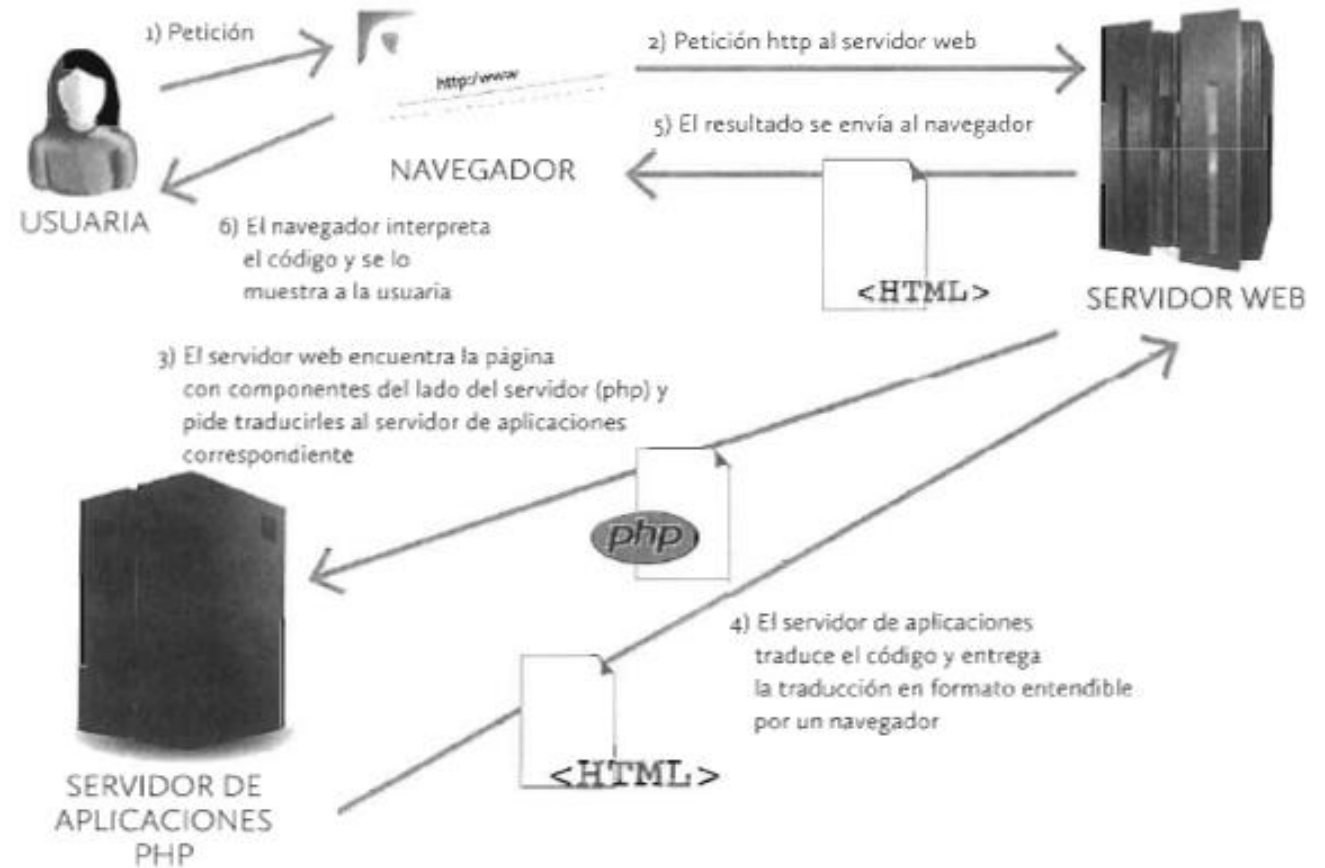


## 4.2. Funcionamiento en el lado del servidor

- En este paradigma, se crean páginas que contienen componentes realizados en lenguajes y tecnologías que deben de ser interpretadas por un servidor en Internet en lugar de en el navegador del usuario.
- Un usuario hace una petición a un recurso web, el servidor que contiene dicho recurso se da cuenta de que contiene elementos a interpretar en el lado del servidor y pide que al servidor de aplicaciones que traduzca esos elementos.
- El servidor de aplicaciones enviará el resultado al servidor web en un formato traducible en el lado cliente

## 4.2. Funcionamiento en el lado del servidor

- La ventaja de este modo de trabajo es que el navegador puede ser más ligero y la parte dura o pesada se la lleva el servidor web.



## 4.2. Funcionamiento en el lado del servidor

- Hoy en día ambos métodos se mezclan, de modo que los desarrolladores web crean aplicaciones que contienen elementos en el lado del servidor (back-end) y en el lado cliente (front-end).
- Por ejemplo, se pueden crear páginas PHP que incluyen código JavaScript; lo que implica proceso en el lado del servidor (traducir el PHP) y en el del cliente (traducir JavaScript)

# 5. Creación de Aplicaciones Web

## **SERVIDORES WEB**

Un servidor web es una máquina o software capaz de interpretar peticiones web realizadas con el protocolo HTTP o HTTPS y de devolver el resultado de la petición.

El navegador es el que pide al servidor web el recurso que desea el usuario, así como de recibirlo, o bien una notificación de error si dicho recurso no existe. La mayoría de las peticiones http resultan en la entrega de un documento HTML.

## 5.1. Servidores de aplicaciones web

- Los servidores web solo tienen la capacidad de resolver peticiones http. Pero no se molestan en descifrar el código de los documentos que entregan.
- Un 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 que le pidió dicha traducción.
- Además, el código del lado del servidor puede contener instrucciones que impliquen acceder a otros servidores con lo que el proceso puede ser muy complejo.

## 5.1. Servidores de aplicaciones web

- Los servidores de aplicaciones trabajan en conjunto con los servidores web para que el proceso se haga de forma transparente al usuario.
- En la práctica, los servidores de aplicaciones son simplemente módulos software que se añaden al servidor web, de modo que los términos servidor web y servidor de aplicaciones web se confunden.
- Hay que tener muy claro su diferencia clave: uno se encarga de las peticiones http y otro de traducir lenguajes de servidor.

## 5.2. Arquitecturas de tres niveles

- Las aplicaciones web utilizan lo que se conoce como arquitectura de tres niveles. Estas capas son:
  - **Capa de presentación.** Maneja la parte de la aplicación web que ve el usuario. Es decir, se encarga de la forma de presentar la información al usuario. Consta del código del lado del cliente (HTML, JavaScript, CSS...) que le llega al navegador, aunque ese código haya sido generado originalmente por una tecnología del lado del servidor.
  - **Capa lógica.** Se encarga de gestionar el funcionamiento de la aplicación. En ella se encuentran los documentos escritos en un lenguaje que se debe de interpretar en el lado del servidor y cuyo resultado se enviará al servidor web para que este, a su vez, lo envíe al cliente. La programación en esta capa divide al código en tres niveles: el modelo, el controlador y la vista.

## 5.2. Arquitectura de tres niveles

- **Capa de negocio.** Es la que contiene la información empresarial. Esta información debe de quedar oculta a todo usuario que no tenga 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.

El servidor de aplicaciones hace peticiones a estos servidores para obtener los recursos de la empresa que se requieren para cumplir la petición http original. El proceso de acceso queda oculto a estos recursos añadiendo mayor seguridad al proceso.

Todo esto es el que involucra la creación de aplicaciones web



## 5.3. Programación BACK-END y Programación FRONT-END

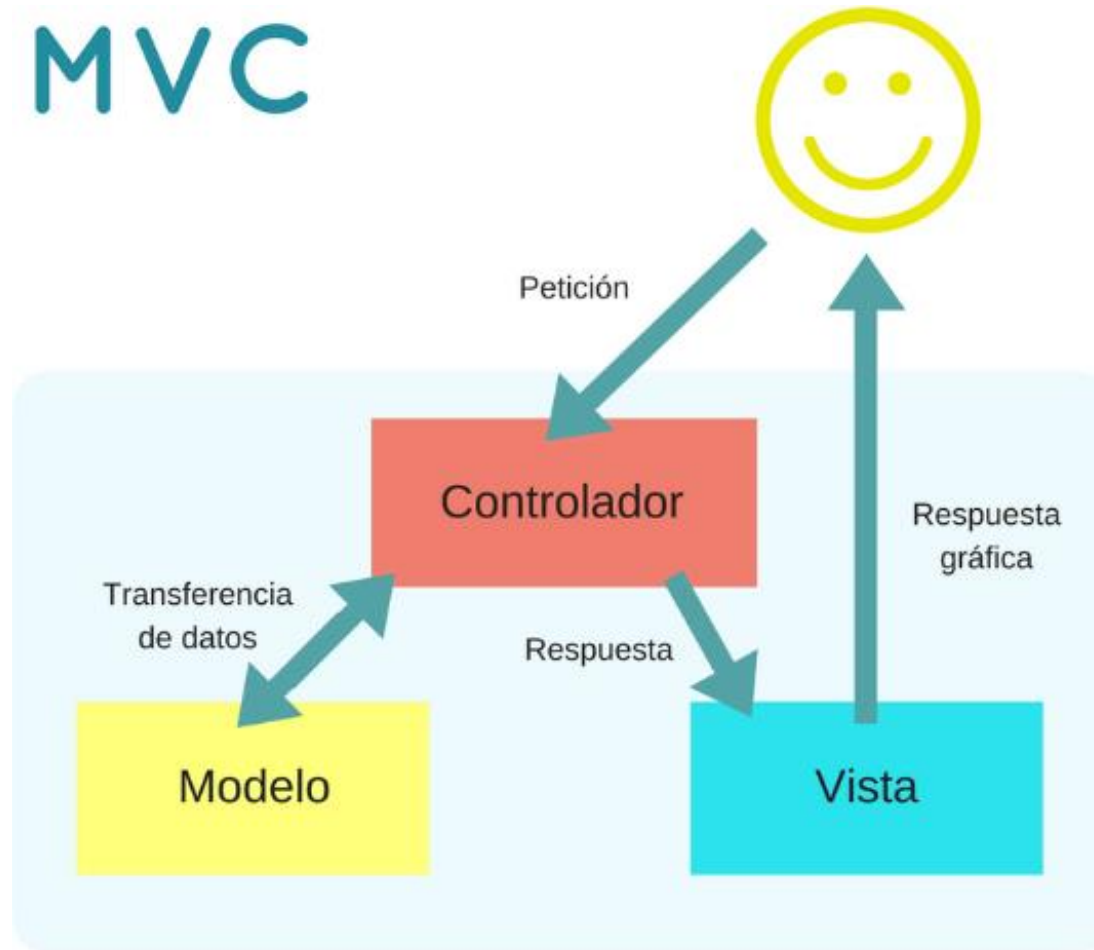
- Existe, además, una división entre las tareas de las personas encargadas del desarrollo de aplicaciones web.
  - **Front-end.** Es la parte del desarrollo que se encarga de producir la apariencia final de la aplicación web que verá el usuario. Forman parte de esto los diseñadores de las maquetas de la aplicación y las personas encargadas del HTML y CSS de la aplicación.
  - **Back-end.** En esta parte se encargan de realizar la parte de la aplicación que queda oculta al usuario. Es la parte de la aplicación encargada de que funcione debidamente la aplicación.

Los objetivos son: acceso rápido a datos, comunicación eficiente con el navegador, control de seguridad, gestor de errores, etc.

## 5.4. Paradigma MVC

- EL modelo MVC de creación de aplicaciones ha tenido una espectacular aceptación. Esta técnica separa la programación de la capa lógica en otras tres capas:
  - Modelo. Contiene el código que se encarga de asociar la información que procede la capa de negocio a su formato entendible por el lenguaje y la tecnología.
  - Vista. Genera la presentación de cara al usuario. Es la encargada de definir la interfaz de usuario.
  - Controlador. Capa encargada de manejar las peticiones del usuario y de comunicar las capas anteriores para que se obtengan lo necesario de dicha petición.

## 5.4. Paradigma MVC



# 6. Tecnologías para crear aplicaciones web

## 6.1 CGI

Es el llamado Interfaz de Pasarela Común y fue la primera tecnología que permitía crear aplicaciones en el lado del servidor.

La idea es crear una aplicación en un servidor en base a un formato especial que permita comunicar dicha aplicación con el servidor web.

Cuando a un servidor web le llega una petición http que contiene implícita una llamada a un a interfaz CGI, el servidor web pide a la aplicación correspondiente que resuelva dicha petición, que desde su origen contendrá los parámetros necesarios para su correcta finalización, y le envíe el resultado.

## 6.1. CGI

- El funcionamiento es el siguiente:
  - 1) El cliente realiza una petición que provocará una solicitud CGI. Lo más habitual es el envío de datos de un formulario par a su proceso; eso suele provocar una petición http de tipo GET o POST.
  - 2) El servidor web recibe la petición y detecta la necesidad de lanzar un a aplicación CGI a la que se pasa como parámetros los datos recibidos por GET o POST.
  - 3) A través de CGI el servidor de aplicaciones accede a los parámetros necesarios y se ejecuta la aplicación con esos datos.
  - 4) El resultado se comunica al servidor web indicando el formato de respuesta y este envía el resultado al cliente a través de una respuesta http.

## 6.2. Lenguajes de programación habituales

Lógicamente hay lenguajes más apropiados que otros para crear aplicaciones de servidor.

Hay lenguajes que tienen especial facilidad para crear aplicaciones web gracias sobre todo a frameworks de trabajo. Comparado con los lenguajes de script son más potentes, pero también más difíciles de aprender.

## 6.2. Lenguajes de programación habituales

- **Perl.** Es el lenguaje clásico para crear aplicaciones de tipo CGI. Es muy poderoso para manejar ficheros y expresiones regulares.
- **Python.** Es un lenguaje muy fácil y potente que además dispone de numerosas facilidades para crear aplicaciones de red y gráficas. Es el lenguaje que más crece actualmente para crear aplicaciones en el lado del servidor.
- **Ruby .** Es también un lenguaje fácil de aprender y que tiene bastante potencia. Es similar a Python. Se hizo muy popular gracias al marco de trabajo Ruby on Rails que facilita la creación completa de una aplicación web usando un patrón MVC.

## 6.2. Lenguajes de programación habituales

- **Java.** Es muy potente y versátil, pero más difícil que los anteriores de aprender. Sus posibilidades para programar en el lado del servidor son muy extensas.
- **C.** Ideado por Microsoft para desbancar el éxito de java, es muy popular para programar aplicaciones utilizando servidores .NET.
- **JavaScript.** El lenguaje clásico del lado del cliente está tomando mucha fuerza para programar en el lado del servidor, gracias a la aparición del servidor de JavaScript node.js, y a los numerosísimos marcos de trabajo especializados como Angular.js o backbone.js.



## 6.3. Lenguajes de script de servidor

- Este tipo de lenguajes se basan en la incrustación del código HTML de una página web. Todos ellos indican mediante una apertura de etiqueta especial, un bloque de código que no es HTML, sino código perteneciente al lenguaje del servidor.
- Además, la página en sí tendrá una extensión especial, gracias a la cual el servidor web detecta que el archivo es de un tipo que requiere la intervención de un servidor de aplicaciones.

## 6.3. Lenguajes de script de servidor

Los principales lenguajes de script son:

- **PHP (Personal Home Pages).** Es el más utilizado. El código PHP se basa en lenguaje C y en Perl. Es un lenguaje fácil de aprender, pero poco valorado por los programadores más formales por su falta de rigidez en las normas de escritura y poca potencia para hacer aplicaciones muy complejas.
- **ASP (Active Server Pages).** Tecnología de Microsoft similar a la anterior. Está pensada para utilizar en servidores web de Microsoft. Actualmente se conoce como ASP.net, ya que se utiliza en la plataforma .NET de Microsoft.

## 6.3.Lenguajes de script de servidor

- **JSP (Java Server Pages).** Lenguaje de script del lado del servidor basado en java. Es la forma de script utilizada cuando una aplicación web se ha programado en java.
- **ColdFusion.** Otro lenguaje de scripts, esta vez propiedad de Adobe. Es el más sencillo de todos, pero requiere servidores especiales. Está en claro desuso.

## 6.4. Plataformas de desarrollo de servicios web empresariales.

Las plataformas de desarrollo incluyen:

- Una estructura de directorios rígida en la que guardar cada tipo de archivo.
- Una serie de archivos de configuración que permitirán configurar la aplicación como un servicio http que sepa cómo atender cada petición que le llegue.
- Un conjunto de programas realizado en uno o más lenguaje de programación completos y potentes (Java, C# ...)
- Un conjunto de documentos ya preparados para enviar al cliente; se les conoce como archivos estáticos porque no requieren la traducción del servidor de aplicaciones.

## 6.4. Plataformas para el desarrollo de servicios web empresariales

- Una serie de archivos de script de servidor (JSP, ASP.net...)
- Componentes reutilizables por los elementos anteriores (como los Enterprise Java Bean , por ejemplo, de java).
- Librerías diversas (de acceso a bases de datos, de uso de módulos especiales para los programas...)
- Facilidad para incorporar patrones y marcos de trabajo compatibles con el paradigma MVC.

## 6.4. Plataformas para el desarrollo de servicios web empresariales

Las plataformas más populares son:

- J2EE (Java 2 Enterprise Edition). Nombre que se le da a la plataforma de creación de aplicaciones web empresariales de Java. Está formada principalmente por el propio lenguaje Java, EJB (Enterprise Java Beans, componentes reutilizables empresariales), servlets y JSP.
- .NET. Plataforma de Microsoft que permite crear aplicaciones y servicios web, haciendo especial énfasis en el transporte de datos mediante XML

## 6.5. Frameworks MVC

Frameworks puede traducirse como estructura, por lo que en lo que a nosotros nos afecta, sería un marco de trabajo. MVC son las siglas de Modelo-Vista-Controlador, un paradigma de programación que separa la misma en tres visiones diferentes.

Todos los frameworks imponen, o por lo menos recomiendan, una estructura concreta de trabajo que separa los diferentes elementos de la aplicación.

## 6.5. Frameworks MVC

Los frameworks MVC más populares son:

- **Ruby on Rails.** Es un marco de trabajo muy exitoso por la facilidad que tiene de programar y sus buenos resultados visuales. El lenguaje en el que se basa es Ruby.
- **Apache Struts.** El marco de trabajo más famoso para la creación de aplicaciones J2EE. Muy preparado para utilizar con servidores web Apache.
- **Spring.** Otro marco para trabajar en J2EE con bastante éxito.
- **Django.** Escrita en Python y pensada para utilizar ese lenguaje.



## 6.5. Paradigma MVC

- **Zend.** Framework escrito para PHP.
- **Angular.js.** Marco de trabajo para JavaScript creado por Google. Es el más utilizado para la creación de aplicaciones web SPA de página simple que permiten que la página vaya mutando para ir cargando solo los elementos de la misma que se requieren.
- **Backbone.js.** Marco ligero de programación de aplicaciones web usando JavaScript como único lenguaje.

## 6.5. Paradigma MVC

Actividad: TECNOLOGÍAS PARA CREAR APLICACIONES WEB

Entramos en la dirección:

[https://w3techs.com/technologies/overview/programming\\_language](https://w3techs.com/technologies/overview/programming_language)

Observa los diferentes apartados y realiza un informe sobre las conclusiones que tomas sobre los lenguajes utilizados en la web.

# 7. Aplicaciones en la nube. Cloud Computing

Consiste en ofrecer cualquier prestación que se pueda programar en un sistema informático. Es accesible desde cualquier dispositivo con conexión a Internet ya que distribuye una serie de enorme de servidores.

Requirió un cambio drástico en la forma de crear aplicaciones, de modo que las mismas puedan atender a una demanda ingente de peticiones de servicio.

El usuario puede conectarse desde cualquier dispositivo y recibir servicio de forma eficiente, sin que tenga que instalar aplicaciones en su ordenador.

## 7. Aplicaciones en la nube. Cloud computing.

- Empresas como Google, Facebook o Amazon fueron las primeras que tuvieron que atender a gran velocidad a las peticiones de millones de usuarios.
- Esta idea permite que se utilice Internet como la base de trabajo, sustituyendo así al propio ordenador personal, el cual puede ser de menor potencia al delegar en la nube el proceso de las tareas y el almacenamiento de la información.

# 7. Aplicaciones en la nube. Cloud computing

Algunos ejemplos de servidores en la nube son:

- Servicios de almacenamiento en la nube.
- Aplicaciones de ofimática web.
- Copias de seguridad en línea.
- Calendarios.
- Contactos.
- Sistemas operativos web.
- Redes sociales.
- Bibliotecas multimedia.
- Marcadores en línea.

# 7.1. Modelos de cloud computing

Los servicios en la nube se suelen dividir en tres tipos:

- **SaaS**, acrónimo de *Software as a Service* que hace referencia a las aplicaciones creadas, de forma que se accede a ellas como un servicio en la red. Se accede a ella mediante el protocolo http.

En definitiva, un servicio es de tipo SaaS, cuando el servicio parece ser una aplicación normal que podríamos tener instalada en nuestro equipo.

- **PaaS**, acrónimo de *Platform as a Service*. Se trata de un servicio que proporciona una plataforma preparada para desarrollar aplicaciones.

La ventaja es que los desarrolladores pueden trabajar desde diferentes estaciones de trabajo debido a que el entorno de trabajo y la propia aplicación están en la nube.

# 7.1. Modelos de cloud computing

- IaaS, Infrastructure as a Service. Servicio en la nube que nos permite virtualizar hardware de todo tipo: un servidor, una red de ordenadores, un disco duro...

La ventaja es que dispondremos de máquinas que no tendremos físicamente instaladas, y alas que accederemos desde un navegador o una aplicación móvil.

A veces esta división se realiza para tener en cuenta las necesidades de una organización. Un servicio SaaS soluciona una necesidad muy concreta. Un PaaS permite solucionar problemas más complejos. Finalmente, IaaS es aún más completa ya que puede atender y personalizar todas las necesidades computacionales de una organización.

## 7.2. Implementación de nubes para el cloud computing

Cuando implementamos un servicio en la nube, realmente lo que tenemos es una aplicación que se distribuye en una nube de servidores ofreciendo el servicio que deseamos implementar.

Hay tres tipos de nubes para implementar un servicio:

- Nubes públicas.
- Nubes privadas.
- Nubes híbridas.



## 7.2. Implementación de nubes para el cloud computing

- **Nubes públicas.** Son nubes ofrecidas por empresas externas.. Ofrecen facilidad para implementarlos ya que toda la gestión es externa; el problema es que se delega, en terceros, información que puede ser crítica; además de que la personalización es muy limitada, al depender de las posibilidades que nos otorga los propietarios de la nube.

## 7.2. Implementación de nubes para el cloud computing

- **Nubes privadas.** Son implementadas por la propia empresa que necesita la nube. Requiere una inversión más grande, pero nos permite personalizar absolutamente el servicio, así como ser nosotros quienes gestionamos nuestros datos sin intervención de terceros. Esta es la forma adecuada si necesitamos servicios en la nube que utilicen información crítica o estratégica.

## 7.2. Implementación de nubes para el cloud computing

- **Nubes híbridas.** Son una mezcla de las anteriores, parte de nuestro negocio se delega a una nube pública y parte lo gestionamos en nuestra nube privada . Intenta aunar lo mejor de ambas opciones.

## 8. Aplicaciones web y aplicaciones móviles

- El éxito de los smartphones ha permitido al usuario lograr que desde el móvil se pueda acceder a los servicios disponibles en Internet. Las aplicaciones presentes en los móviles tienen sencillez de uso y acceso directo para resolver problemas complejos o, al menos, muy interesantes para el usuario.
- El éxito de las aplicaciones ha hecho que las aplicaciones web palidezcan debido, principalmente, a la facilidad para instalar una app en el móvil.

## 8. Aplicaciones web y aplicaciones móviles

- Hay varias similitudes entre aplicaciones móviles y aplicaciones web. Las aplicaciones que un smartphone pueden ejecutar son de dos tipos:
  - Web apps. Son aplicaciones web pensadas para ser utilizadas en un dispositivo móvil; consideran el tamaño, resolución y orientación habitual de un móvil para construir la aplicación.  
Se crean fundamentalmente en HTML5, CSS y JavaScript.
  - Apps nativas. Son aplicaciones programadas en los lenguajes nativos de los dispositivos móviles: Java, C++, Swift, etc. Si queremos que nuestra app nativa sea compatible con todas las plataformas debemos programarlo en cada lenguaje de cada una de ellas. Es más costoso al obligar a hacer varias versiones.

## 8. Aplicaciones web y aplicaciones móviles

La principal ventaja de las apps nativas es que el hardware de la máquina está en nuestro dispositivo ya que el lenguaje permite el acceso al GPS, la luz LED del móvil, la cámara, etc.

Otra ventaja es que se publican en un market, es decir, una tienda online que permite buscar e instalar la aplicación en el dispositivo.

El manejo de las apps nativas es mucho más sencillo para el usuario, pero es mucho más complejo a la hora de programarse, por lo que el desarrollador debe esforzarse más.