



Tema 1. Introducción a la programación web

Programación web

Boni García
Curso 2017/2018

Índice

1. Introducción
2. Evolución histórica de la Web
3. Navegadores y servidores web
4. Evolución histórica de las aplicaciones web
5. Tecnologías del cliente
6. Tecnologías del servidor
7. Aplicaciones web en dispositivos móviles
8. Sistemas gestores de contenido
9. Lenguajes de marcado ligeros

Índice

1. **Introducción**
 - ¿Qué es la Web?
 - Página y sitio web
 - Aplicación web
 - **Desarrollo de aplicaciones web**
2. Evolución histórica de la Web
3. Navegadores y servidores web
4. Evolución histórica de las aplicaciones web
5. Tecnologías del cliente
6. Tecnologías del servidor
7. Aplicaciones web en dispositivos móviles
8. Sistemas gestores de contenido
9. Lenguajes de marcado ligeros

1. Introducción

¿Qué es la Web?

- La **Web** (*World Wide Web*) es un servicio de distribución de contenidos hipertexto accesibles vía Internet
- **Internet** es un conjunto descentralizado de redes de datos interconectadas que utilizan la familia de protocolos **TCP/IP** que interconecta cientos de millones de dispositivos (hosts o sistemas terminales) en todo el mundo

web

Del *ingl.*
web;
propiamente 'red, malla'.

1. f. *Inform.* Red informática.

página web

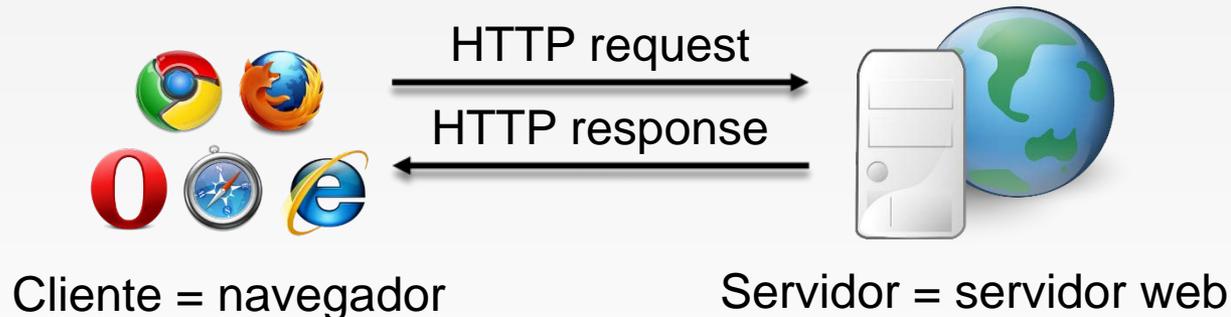
sitio web

Aunque la palabra *web* está en la RAE su definición no es muy precisa

1. Introducción

¿Qué es la Web?

- La Web está basada en una arquitectura **cliente-servidor**



- El protocolo de nivel de aplicación para comunicar clientes y servidores en la Web es **HTTP** (*HyperText Transfer Protocol*)
- Las páginas se identifican con un nombre único llamado dirección web o **URL** (*Uniform Resource Locator*)

1. Introducción

Página y sitio web

- Una **página web** es un documento electrónico escrito en **HTML** (*HyperText Markup Language*)
- Las páginas web están enlazadas a través de **hiperenlaces** (*links*)
- Mediante un navegador un usuario puede **navegar** a través de la web siguiendo los hiperenlace
- Un conjunto de páginas alojadas en el mismo servidor se suele conocer como **sitio web**
- En los comienzos de la web, todos los sitios web eran conjuntos de páginas web en forma de ficheros HTML

1. Introducción

Aplicación web

- Poco a poco los servidores comenzaron a ejecutar lógica para la generación de las propias páginas web u otras funciones (como por ejemplo el acceso a base de datos) naciendo así las llamadas aplicaciones web
- Una **aplicación web** es aquella aplicación que los usuarios pueden utilizar accediendo a un **servidor web** a través de **Internet** mediante un software que actúa de cliente llamado **navegador**
- Las aplicaciones web son muy populares:
 - El navegador web como cliente universal
 - Proporcionan independencia del sistema operativo
 - Facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales

1. Introducción

Desarrollo de aplicaciones web

- El impacto de la Web ha propiciado la aparición de diversas tecnologías para la creación de aplicaciones web. Hay dos enfoques principales:
 1. Creación de aplicaciones web con **tecnologías de desarrollo**:
 - **Lado cliente** (*frontend*). Tecnologías que permiten crear interfaces de usuario atractivos y permiten la comunicación con el servidor. Basadas en HTML, CSS y JavaScript
 - **Lado servidor** (*backend*). Tecnologías que permiten implementar la lógica de negocio así como la integración con otros servicios (típicamente con un servidor de base datos)
 2. Creación de aplicaciones web con **sistemas gestores de contenido** (CMS, *Content Management System*)
 3. Últimamente se ha popularizado el uso de **lenguajes de marcados ligeros** para la creación de sitios web estáticos

Índice

1. Introducción
2. Evolución histórica de la Web
 - Internet
 - La Web
 - Web 2.0
 - Redes sociales
3. Navegadores y servidores web
4. Evolución histórica de las aplicaciones web
5. Tecnologías del cliente
6. Tecnologías del servidor
7. Aplicaciones web en dispositivos móviles
8. Sistemas gestores de contenido
9. Lenguajes de marcado ligeros

2. Evolución histórica de la Web

Internet

- En los años 60 la agencia del Departamento de Defensa de EEUU llamada **DARPA** (*Defense Advanced Research Projects Agency*) creó la red Arpanet
- **Arpanet** era una red de datos de conmutación de paquetes no centralizada basada en TCP/IP
- En el año 83 nace oficialmente **Internet** al separarse Milnet (parte militar) de Arpanet



2. Evolución histórica de la Web

La Web

- En 1989 Tim Berners-Lee, físico del CERN (*Conseil Européen pour la Recherche Nucléaire*), propone un primer esbozo para la WWW (*World Wide Web*) para hacer accesible la gran cantidad de información de los proyectos del CERN
 - Berners-Lee y su equipo escriben el primer servidor web y el primer programa cliente
 - Berners-Lee y su equipo crearon el HTML, el HTTP y las URL
- En 1994 nace el W3C (*World Wide Web Consortium*)



2. Evolución histórica de la Web

La Web

- De 1993 a 1997 la Web es principalmente estática, y sus contenidos son de “sólo lectura”
- A parte de 1997 hay varios hitos tecnológicos:
 - En 1997 aparecen los blogs
 - En 1998 nace Google
 - En 2001 nace la Wikipedia
- Los usuarios tienen más facilidades para editar el contenido de las páginas web, naciendo así la Web “2.0”



2. Evolución histórica de la Web

Web 2.0

- Los usuarios toman el control de los contenidos
 - Web original (1.0): Web de lectura
 - Web 2.0: Web de lectura/escritura
- El término Web 2.0 fue acuñado por Dale Dougherty (confundador de O'Reilly) en 2004
- Algunas características de la Web 2.0 son:
 - Uso de tecnologías **wiki**: creación de contenidos de forma colaborativa
 - Uso de **folcsonomías** (uso de etiquetado en base a un espacio de nombres)
 - Uso de servicios de **redes sociales**



accessories apparel art
arts audio australia baby
bags beauty books
business california
canada car casual cheap
children clothes
clothing community
companies computer
computers
computersandinternet
consumerelectronics

2. Evolución histórica de la Web

Redes sociales

- Las redes sociales son aplicaciones web que permiten la creación de **comunidades** en las que se establece un intercambio de información:
 - Redes sociales generalistas para relaciones personales: Facebook, Google+
 - Redes sociales centradas en la agilidad en la comunicación: Twitter
 - Redes sociales centradas en las relaciones laborales: LinkedIn
 - Redes sociales centradas en vídeos: YouTube
 - Redes sociales centradas en imágenes: Flickr, Instagram



Índice

1. Introducción
2. Evolución histórica de la Web
3. Navegadores y servidores web
 - Navegadores web
 - Servidores web
 - Protocolo HTTP
 - Programación del lado del servidor
4. Evolución histórica de las aplicaciones web
5. Tecnologías del cliente
6. Tecnologías del servidor
7. Aplicaciones web en dispositivos móviles
8. Sistemas gestores de contenido
9. Lenguajes de marcado ligeros

3. Navegadores y servidores web

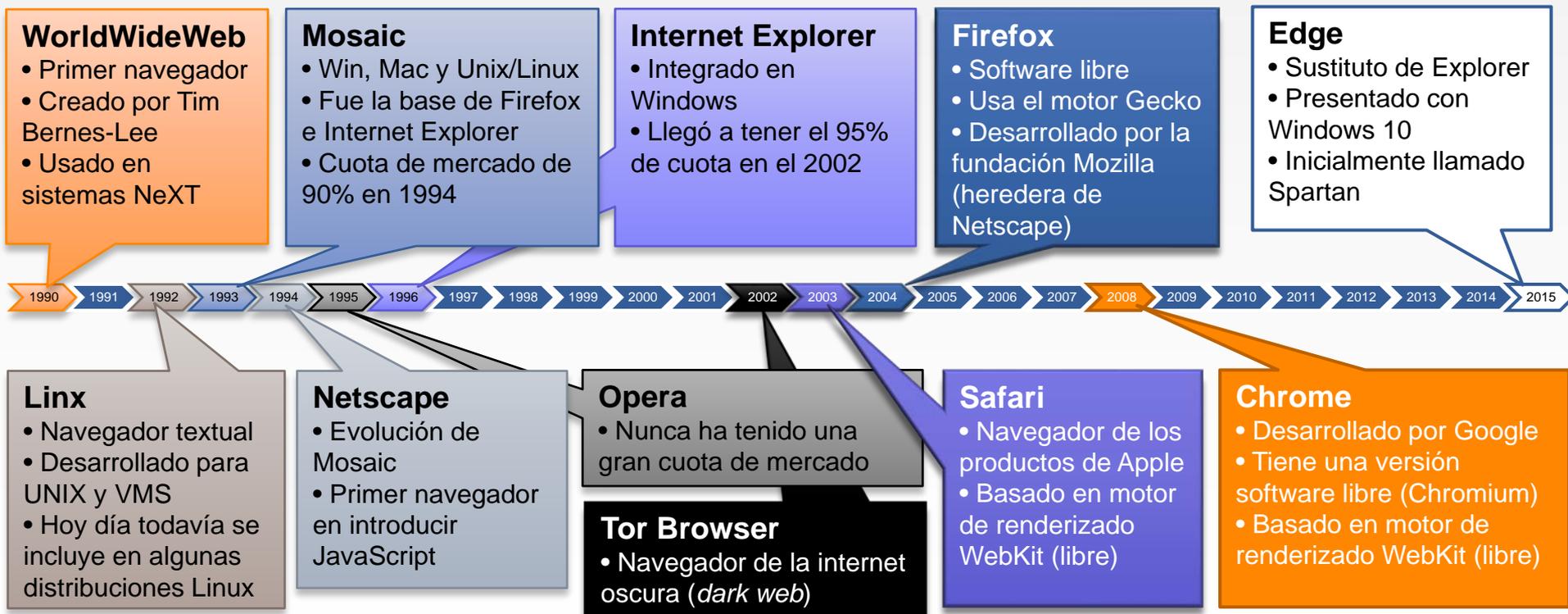
Navegadores web

- Un **navegador web** es una aplicación que se instala en el sistema que utiliza el usuario
- El usuario escribe una dirección web (**URL**, *Uniform Resource Locator*). La dirección contiene el nombre del servidor web y el nombre del recurso que se solicita
- El navegador hace una **petición** al servidor y solicita el recurso
- El navegador descarga el recurso y lo visualiza (o lo descarga si no puede hacerlo)
- Si el recurso es una página HTML, además de visualizar su contenido, descarga recursos adicionales como imágenes, estilos, etc. y los visualiza integrados en la página

3. Navegadores y servidores web

Navegadores web

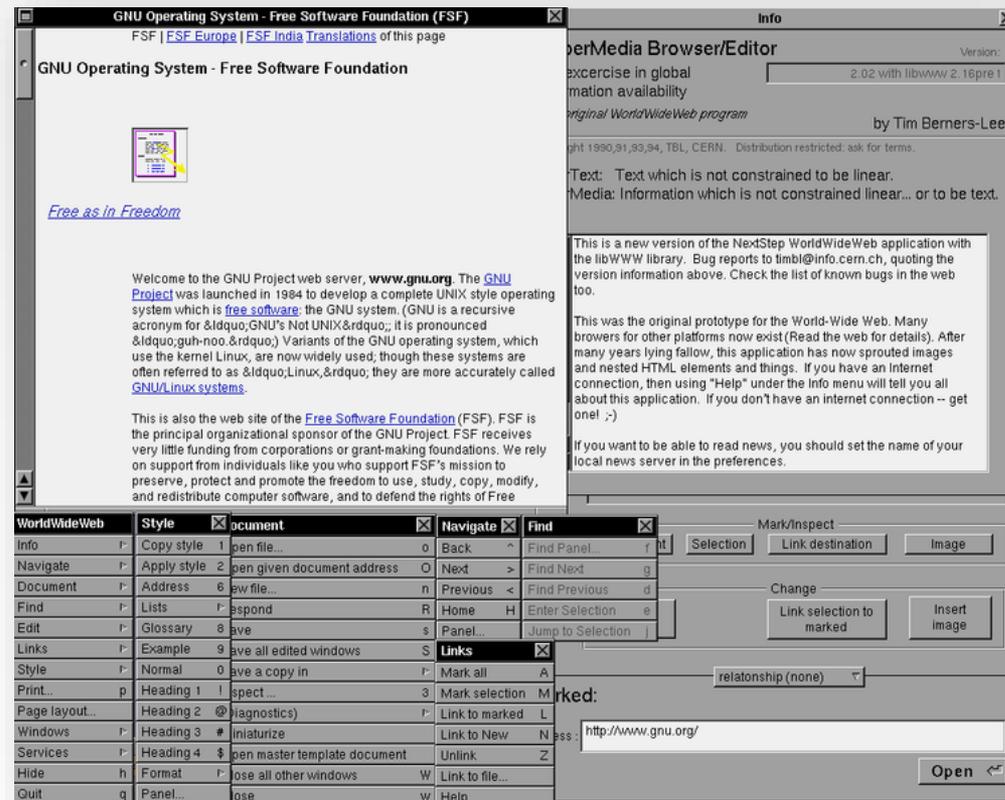
▪ Evolución de los navegadores web



3. Navegadores y servidores web

Navegadores web

- WordWideWeb (1990)



3. Navegadores y servidores web

Navegadores web

- Linx (1992)

```
# Portada - Wikipedia, la enciclopedia libre (p1 of 11)
#search copyright

Los resultados de la votación para el Consejo de Administración de
la Fundación Wikimedia ya están disponibles.

Portada
De Wikipedia, la enciclopedia libre

Saltar a navegación, búsqueda

¡Bienvenido a Wikipedia, la enciclopedia libre!
Edición en español de Wikipedia, iniciada en el 2001.
Quarto - Libro de visitas - Acceso WAP - Contacto

Explora Wikipedia

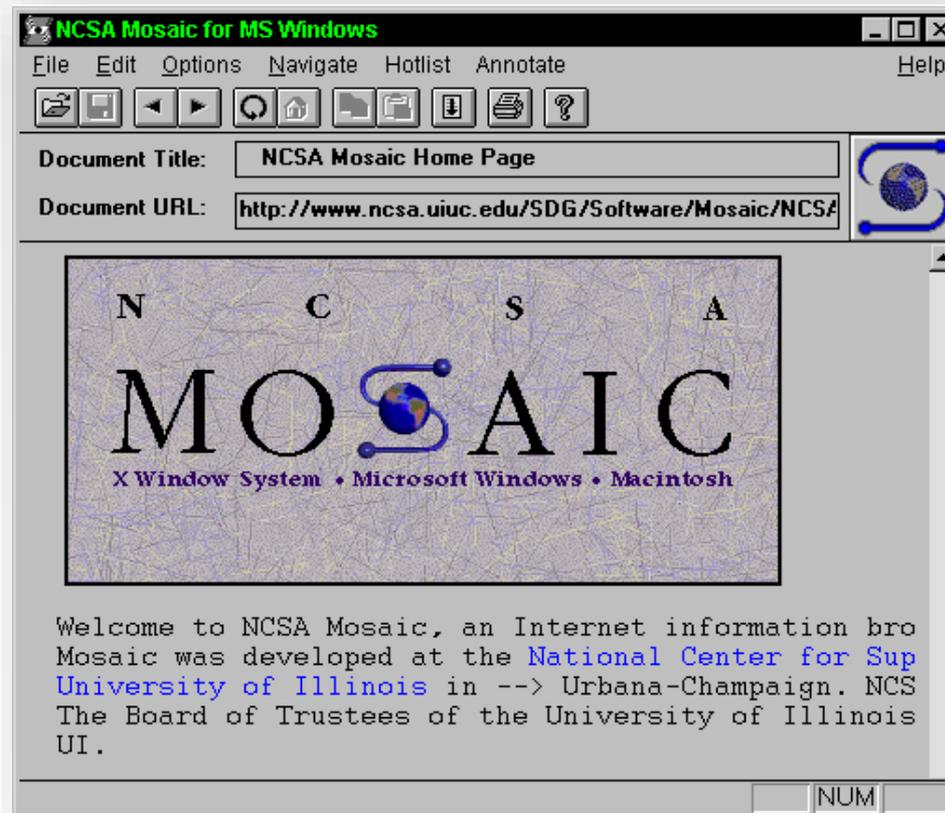
Exploración Exploración Clasificación Sistemas de clasificación
Búsqueda · índice alfabético · índice de categorías · Portales
temáticos · Todos los artículos · Acceso WAP Decimal Universal* ·
Unesco* (4 dígitos* · 6 dígitos*) · Listas (Acrónimos · Biografías ·
Países)

Buscar título exacto Buscar en el texto
En la columna de la izquierda de cada artículo hay una caja de
búsqueda equivalente a esta.
(NORMAL LINK) Use right-arrow or <return> to activate.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [deletel=history list
```

3. Navegadores y servidores web

Navegadores web

- Mosaic (1993)



3. Navegadores y servidores web

Navegadores web

- Netscape 0.9 (1994)



3. Navegadores y servidores web

Navegadores web

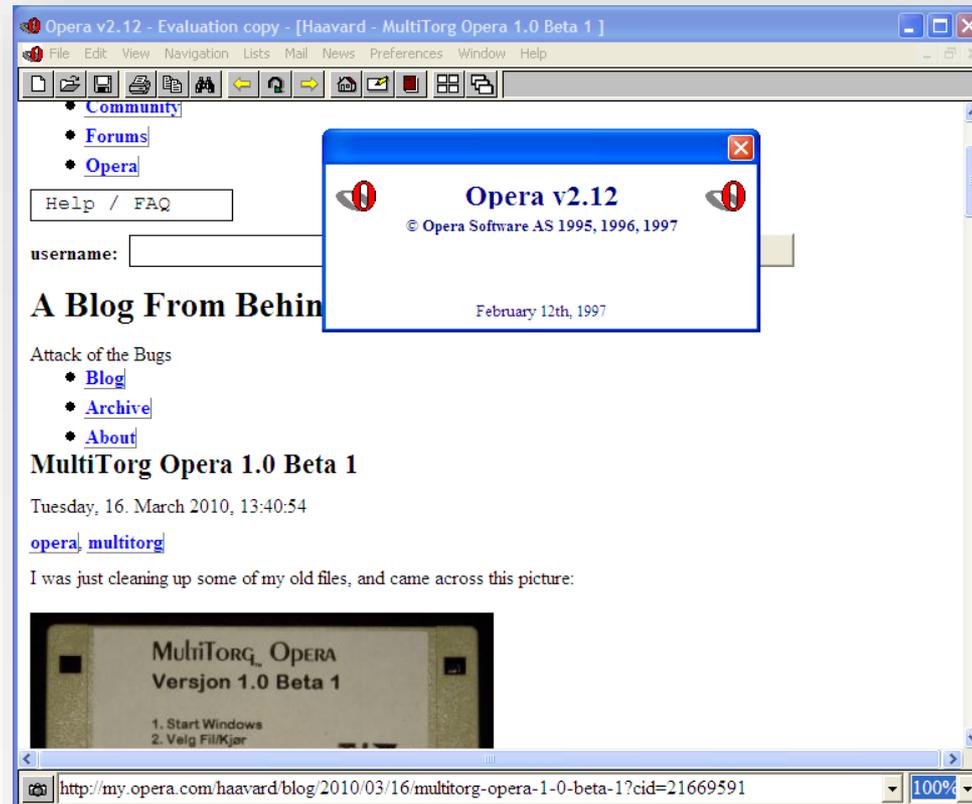
- Internet Explorer 1.0 (1995)



3. Navegadores y servidores web

Navegadores web

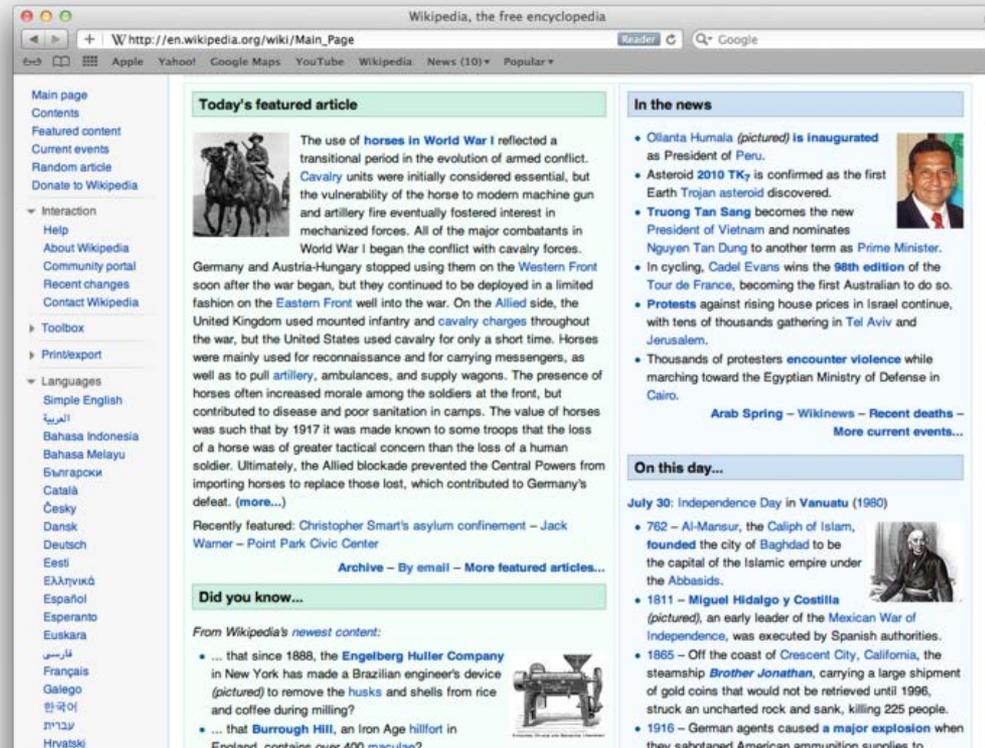
- Opera 2.12 (1997)



3. Navegadores y servidores web

Navegadores web

- Safari 5.1 (2011)



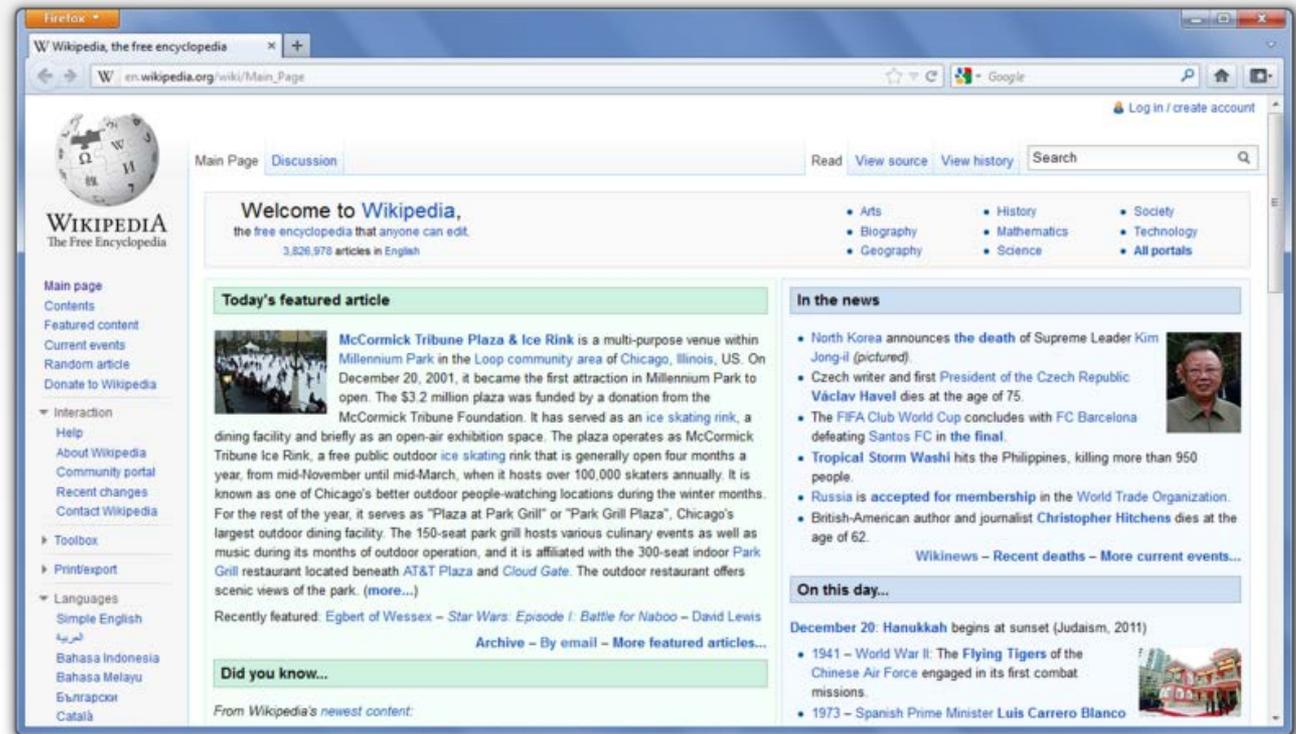
3. Navegadores y servidores web

Navegadores web

- Firefox 9 (2011)



mozilla
Firefox®



3. Navegadores y servidores web

Navegadores web

- Google Chrome 16 (2012)



The screenshot shows a Google Chrome browser window displaying the Wikipedia article for "Google Chrome". The browser's address bar shows the URL "en.wikipedia.org/wiki/Google_Chrome". The article content includes:

Google Chrome
From Wikipedia, the free encyclopedia

This article is about the web browser. For the operating system, see Google Chrome OS.

Google Chrome is a web browser developed by Google that uses the WebKit layout engine. It was first released as a beta version for Microsoft Windows on September 2, 2008, and the public stable release was on December 11, 2008. The name is derived from the graphical user interface frame, or "chrome", of web browsers. As of April 2011, Chrome was the third most widely used browser with 12% worldwide usage share of web browsers, according to Net Applications.^[1]

In September 2008, Google released a large portion of Chrome's source code, including its V8 JavaScript engine, as an open source project entitled Chromium.^{[2][3]} This move enabled third-party developers to study the underlying source code and to help convert the browser to the Mac OS X and Linux operating systems. Google also expressed hope that other browsers would adopt V8 to improve web application performance.^[4] The Google-authored portion of Chromium is released under the permissive BSD license,^[5] which allows portions to be incorporated into both open source and closed source software programs.^[6] Other portions of the source code are subject to a variety of open source licenses.^[7] Chromium implements the same feature set as Chrome, but lacks built-in automatic updates and Google branding, and most noticeably has a blue-colored logo in place of the multicolored Google logo.^[8]

Contents [hide]

- 1 History
 - 1.1 Announcement
 - 1.2 Public release
 - 1.3 Development
 - 1.3.1 Pre-releases
 - 1.3.2 Chromium
 - 1.4 Release history
- 2 Features
 - 2.1 Acid tests
 - 2.2 Web standards conformance tests
 - 2.3 Security

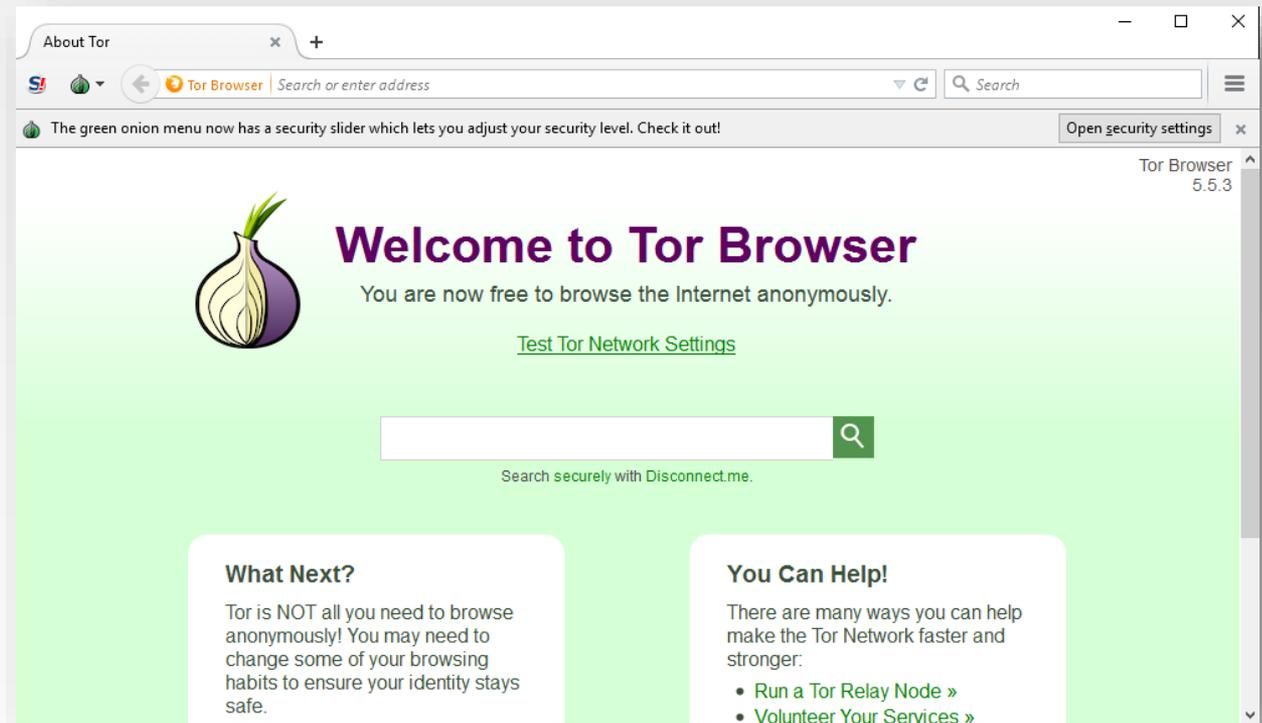
Google Chrome

Developer(s)	Google Inc.
Initial release	September 2, 2008; 2 years ago
Stable release	11.0.696.65 (May 6, 2011; 3 days ago) [v · i]
Preview release	12.0.742.30 (Dev) (May 6, 2011; 3 days ago) [v · i]
Written in	C++, Assembly, Python, JavaScript
Operating system	Linux Mac OS X (10.5 and later, Intel only) Windows (XP SP2 and later)
Engine	WebKit (based on KHTML)
Available in	50 languages
Development status	Active
Type	Web browser
License	Google Chrome Terms of Service g2[<i>note</i> 1] WebKit: BSD/LGPL; V8: BSD

3. Navegadores y servidores web

Navegadores web

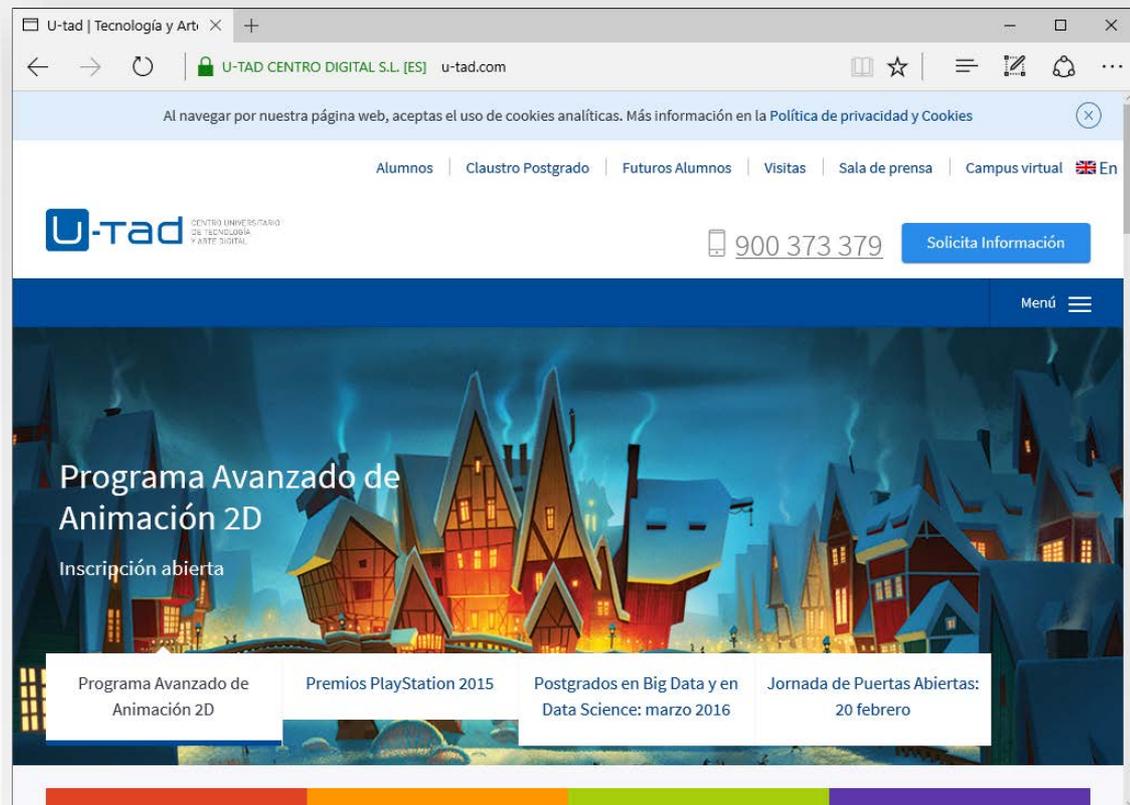
- Tor Browser (2015)



3. Navegadores y servidores web

Navegadores web

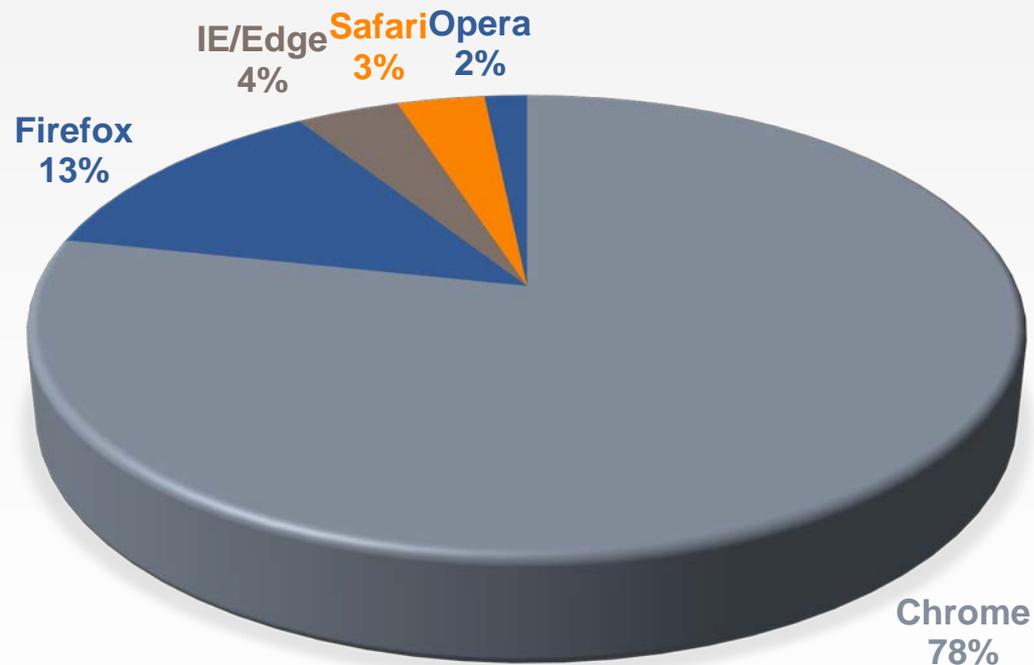
- Edge (2015)



3. Navegadores y servidores web

Navegadores web

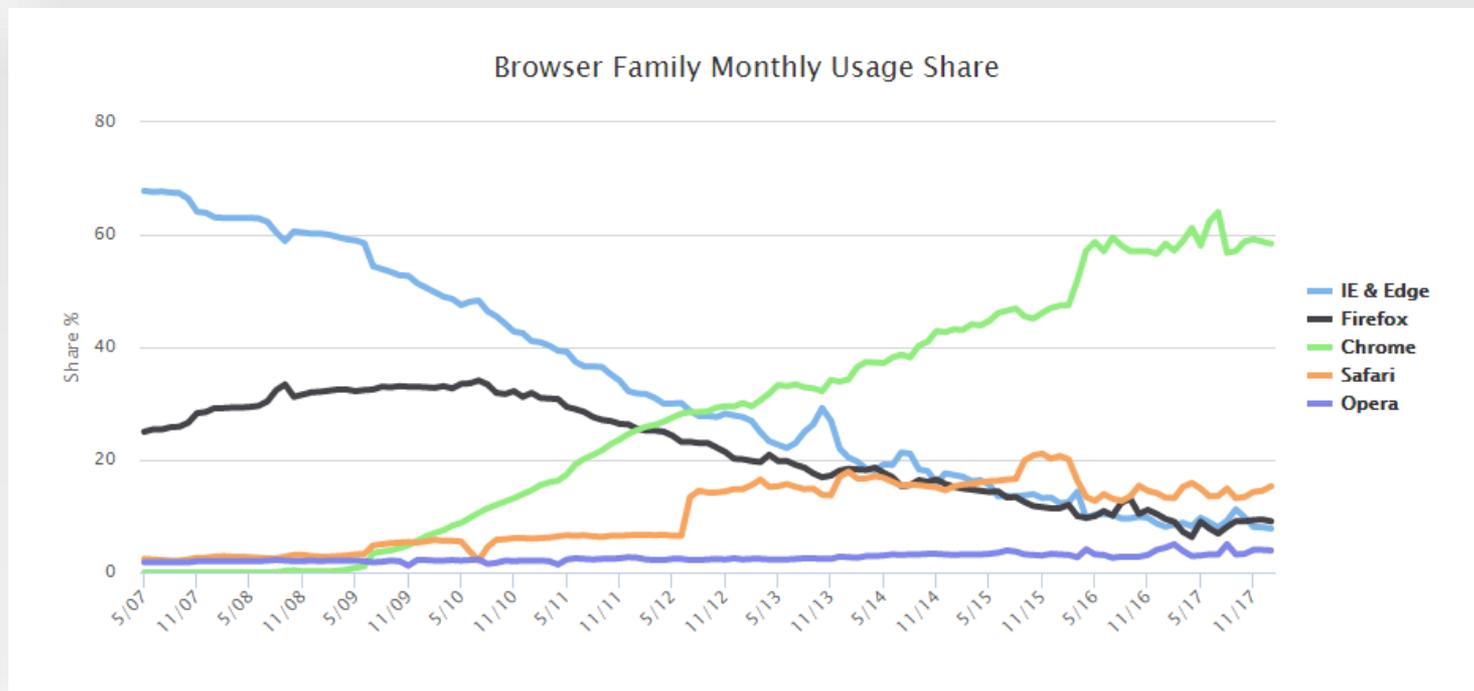
- Cuota de uso de navegadores (diciembre 2017)



http://www.w3schools.com/browsers/browsers_stats.asp

3. Navegadores y servidores web

Navegadores web



<http://www.w3counter.com/trends>

3. Navegadores y servidores web

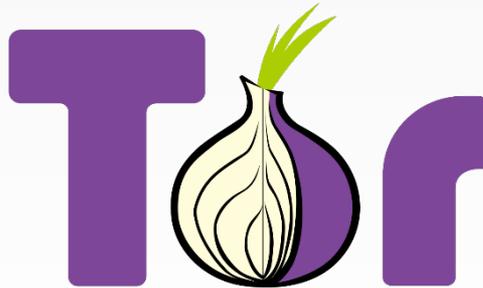
La web oscura

- La Internet oculta está compuesta por la web profunda (*deep web*) y la web oscura (*dark web*)
 - Las páginas que encontramos en la Internet oculta no son indexables por motores de búsqueda como Google, Bing, o Yahoo
 - Es el término opuesto a la Internet superficial (*surface web*)
- La **deep web** engloba a toda la información que no se puede acceder públicamente
 - Pueden tratarse de páginas convencionales protegidas por contraseña
 - Se calcula que el 90% del contenido web entra dentro de esta categoría
- La **dark web** es el contenido que sólo se puede acceder a través de redes oscuras (*dark net*)
 - Son redes que persiguen el anonimato entre los diferentes nodos
 - La más famosa es la red Tor

3. Navegadores y servidores web

La web oscura

- El proyecto Tor (*The Onion Router*) tienen como objetivo el desarrollo de la red que proporciona anonimato a nivel de red (IP)
- Para ello, el proyecto Tor proporciona diferentes tipo de software *open source* para los nodos, el navegador, etc.
- Tor Browser nos permite navegador en la web oscura (dominios .onion)

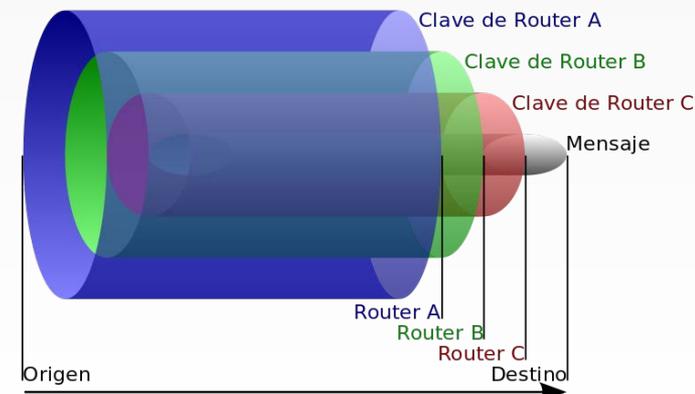


<https://www.torproject.org/>

3. Navegadores y servidores web

La web oscura

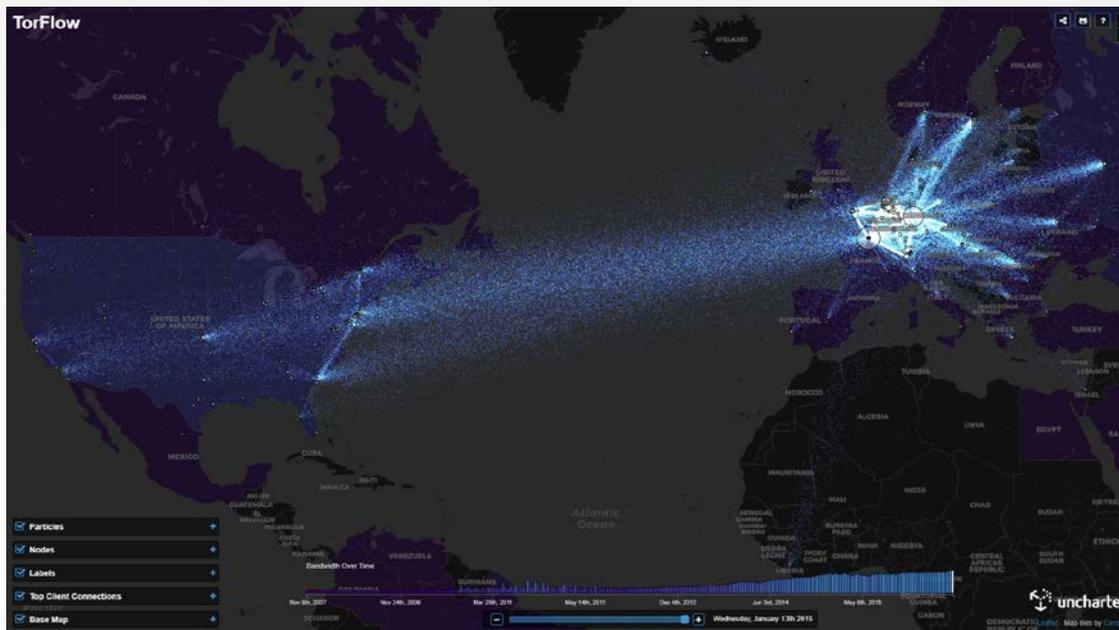
- La red Tor está formada por un número de nodos proporcionados por un conjunto de organizaciones e individuos. Esta red Tor implementa el enrutado cebolla (*onion routing*)
- Cuando un host quiere transmitir información a través de la red Tor, se generará un camino de forma aleatoria entre los nodos de la red hasta llegar al nodo de salida previo al destino
- El host origen utiliza cifrado asimétrico por capas (como una cebolla) de forma que los nodos sólo son capaces de descifrar los mensajes intermedios



3. Navegadores y servidores web

La web oscura

- Existen mapas que permiten visualizar el tráfico de la red Tor



<https://torflow.uncharted.software/>

3. Navegadores y servidores web

La web oscura

- La web oscura permite intercambiar información de manera anónima
 - Un de los servicios más popular es la “*Hidden wiki*”, que es un directorio de páginas alojadas en la web oscura (<http://zqktlwi4fecvo6ri.onion/>)
 - Otro servicios muy conocidos de la web oscura era “*Silk Road*”, un mercado negro de todo tipo de sustancias en el que se usa monedas virtuales descentralizadas como BitCoin. Fue cerrado por el FBI en octubre de 2013
- En definitiva la web oscura es un sitio con información muy diversa, mucha de ella ilegal. Algunos consejos si decidimos entrar:
 - Usar Tail (sistema operativo live basado en Debian) o similar
 - No acceder a nuestros servicios personales dentro de la web oscura
 - Extremar las medidas de seguridad

3. Navegadores y servidores web

Servidores web

- Un **servidor web** envía por HTTP los ficheros que tiene almacenados en su disco duro a los clientes que lo solicitan
- Puede servir cualquier tipo de fichero, aunque lo habitual son los ficheros que un navegador reconoce (html, jpg, png, pdf...)
- Cuando recibe una petición, devuelve el fichero del disco duro que se ajuste a la ruta indicada en la URLs

<http://www.miservidor.com/ruta/del/fichero/fichero.txt?clave=valor#fragmento>

Protocolo Nombre del servidor Ruta del recurso Nombre del recurso Consulta Anclaje

3. Navegadores y servidores web

Servidores web

- La mayoría de los servidores web permiten que en cada petición se ejecute un pequeño programa que genera **dinámicamente** el recurso que se envía al usuario (*server-side scripting*)
- A este contenido generado “al vuelo” se le denomina contenido dinámico, en contraposición al contenido estático obtenido del disco
- Es bastante habitual que el contenido dinámico se genere con la información de una **base de datos**
- Esta funcionalidad permite el desarrollo de completas **aplicaciones web**

3. Navegadores y servidores web

Servidores web

▪ Evolución de los servidores web:

CERN httpd

- Sistema NeXTSTEP



Internet Information Server (IIS)

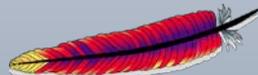
- Propietario de Microsoft, integrado en Windows y su base de datos MS SQL Server
- Es el segundo servidor más usado a (28% en enero de 2015)
- Permite escribir contenido dinámico
- <http://www.iis.net/>



1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004

Apache

- El servidor más popular (40% cuota en enero de 2015)
- Desarrollado por la Apache Software Foundation
- Licencia software libre Apache License
- Multiplataforma
- Permite escribir contenido dinámico
- <http://httpd.apache.org>



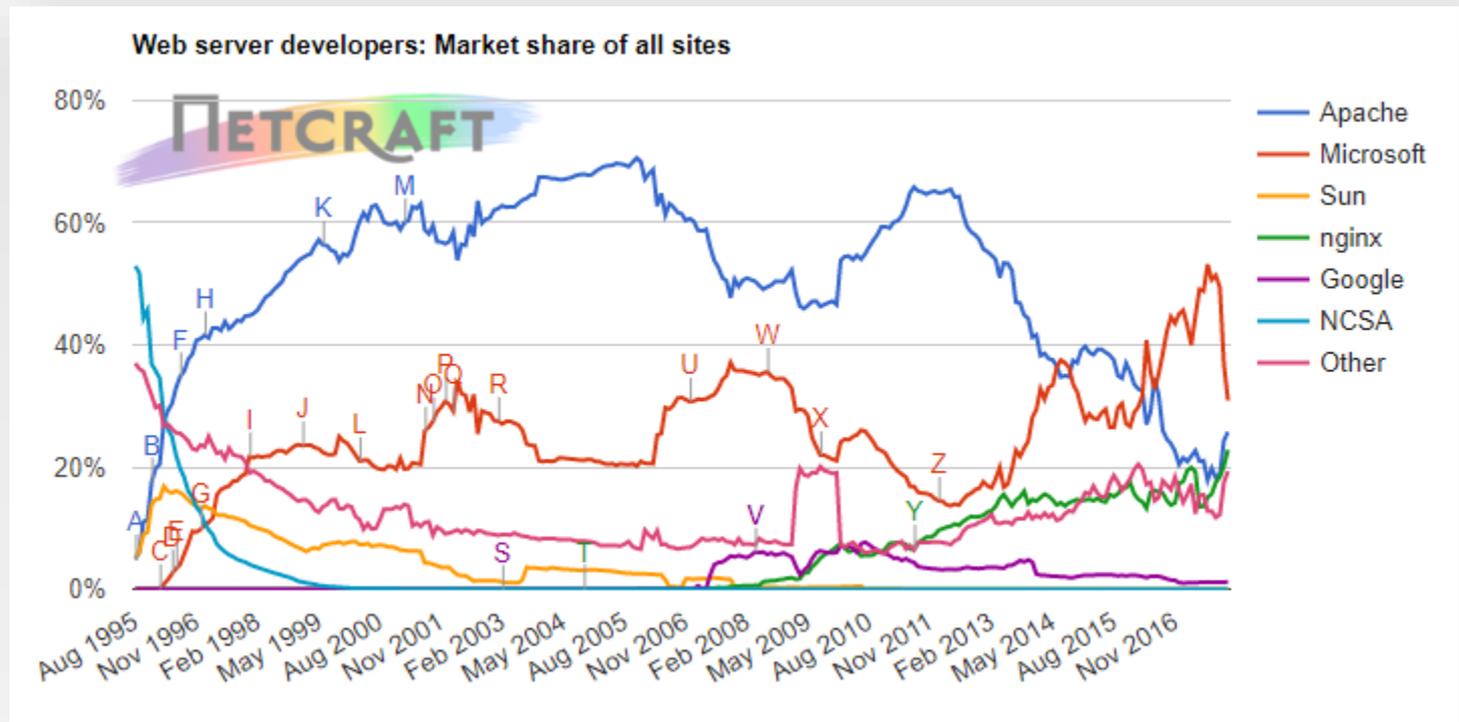
NginX

- Servidor multiplataforma y software libre (licencia tipo BSD)
- Es el tercer servidor más usado a (15% en enero de 2015)
- Es muy rápido y se usa como balanceador de carga
- <http://nginx.net/>

NGINX

3. Navegadores y servidores web

Servidores web

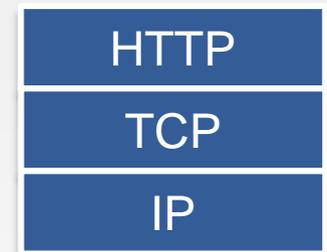


<https://news.netcraft.com/archives/2017/12/26/december-2017-web-server-survey.html>

3. Navegadores y servidores web

Protocolo HTTP

- *HyperText Transfer Protocol*, protocolo de aplicación basado en arquitectura cliente/servidor
- La versión más usada actualmente es HTTP 1.1
 - Desde 2015 ya está disponible HTTP 2.0
- Los servidores web escuchan por defecto en el puerto TCP 80 (*well-known port*)
- Todo elemento web está identificado por una **URL** (*Uniform Resource Locator*)
 - Desde 1994, en los estándares de la Internet (RFCs), el concepto de URL ha sido incorporado dentro del más general de URI (*Uniform Resource Identifier*)
- El cliente (navegador) realiza peticiones (***request***) que causan el envío de una respuesta (***response***) por parte del servidor web



3. Navegadores y servidores web

Protocolo HTTP

- Ejemplo *request-response*:

HTTP request

```
GET /indice.html HTTP/1.1
Host: www.ejemplo.com
User-Agent: Mozilla/4.0
Accept: text/html, image/gif,
image/jpeg
```

Respuesta

HTTP response

```
HTTP/1.1 200 OK
Date: Tue, 31 Dec 2011 23:59:59 GMT
Server: Apache/2.0.54 (Fedora)
Content-Type: text/html
Last-Modified: Mon, 30 Dec 2011 ...
Content-Length: 1221

<html>
    <body>
        <h1>Ejemplo de página</h1>
        . . .
    </body>
</html>
```

Cabeceras

CRLF

Cuerpo

Petición

Cabeceras

3. Navegadores y servidores web

Protocolo HTTP

- Los métodos en HTTP (algunas veces referido como "verbos") indican la acción que desea que se efectúe sobre el recurso identificado
- HTTP 1.1 ([RFC 2616](#)) define 8 métodos:



- Hay una extensión a HTTP 1.1 ([RFC 5789](#)) que define un nuevo método:



3. Navegadores y servidores web

Protocolo HTTP

- **GET:** Petición de un recurso determinado (URL)
 - **POST:** Envío de datos que serán procesados por un recurso (URL) } Métodos principales
 - **PUT:** Crea un recurso
 - **DELETE:** Borra un recurso } Métodos normalmente deshabilitados en los servidores web
 - **PATCH:** Solicita al servidor la modificación parcial de un recurso
- **HEAD:** Pide una respuesta idéntica a la que correspondería a una petición GET, pero sin el cuerpo de la respuesta. Esto es útil para conocer las cabeceras de la respuesta pero sin transportar todo el contenido.
- **TRACE:** Solicita al servidor que envíe de vuelta en un mensaje de respuesta con la petición enviada (servicio de *echo*). Se utiliza con fines de comprobación y diagnóstico.
- **OPTIONS:** Solicita al servidor los métodos admitidos para un determinado recurso. La respuesta se obtiene en la cabecera `Allow`:
- **CONNECT:** Se utiliza para indicar a un proxy web que establezca una conexión segura (TLS) con una máquina remota

3. Navegadores y servidores web

Protocolo HTTP

- Las respuestas pueden ser del tipo:
 - 1xx Respuesta informativa. Por ejemplo:
 - 101 Cambio de protocolo
 - 2xx Operación exitosa. Por ejemplo:
 - 200 OK
 - 3xx Redirección. Por ejemplo:
 - 304 No modificado (usado como respuesta en un GET condicional → recurso en caché)
 - 307 Redirección temporal
 - 4xx Error por parte del cliente. Por ejemplo:
 - 404 No encontrado
 - 5xx Error del servidor. Por ejemplo:
 - 500 Error interno

3. Navegadores y servidores web

Protocolo HTTP

- Algunas de las **cabeceras** más comunes en las HTTP request:

Cabecera	Descripción	Ejemplo
Accept	Determina el tipo de contenido o MIME	Accept: text/plain
Accept-Charset	Juego de caracteres de caracteres aceptable en la respuesta	Accept-Charset: utf-8
Connection	Tipo de conexión	Connection: keep-alive
Host	Nombre de dominio del servidor	Host: en.wikipedia.org
If-Modified-Since	Get condicional. Se usa para saber si un recurso ha cambiado desde una fecha determinada	If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT
User-Agent	Cadena que identifica al cliente	User-Agent: Mozilla/5.0 ...

3. Navegadores y servidores web

Protocolo HTTP

- Algunos de las tipos **MIME** más comunes:

Tipo	Extensión(es)	Descripción
text/plain	.txt	Texto plano
text/html	.html .htm	Página web
image/jpeg	.jpg .jpeg	Imagen JPEG
image/gif	.gif	Imagen GIF
image/png	.png	Imagen PNG
application/pdf	.pdf	Archivo PDF
audio/mpeg3	.mp3	Audio en formato MP3
video/mpeg	.mpg .mpeg	Vídeo en formato MPEG

3. Navegadores y servidores web

Protocolo HTTP

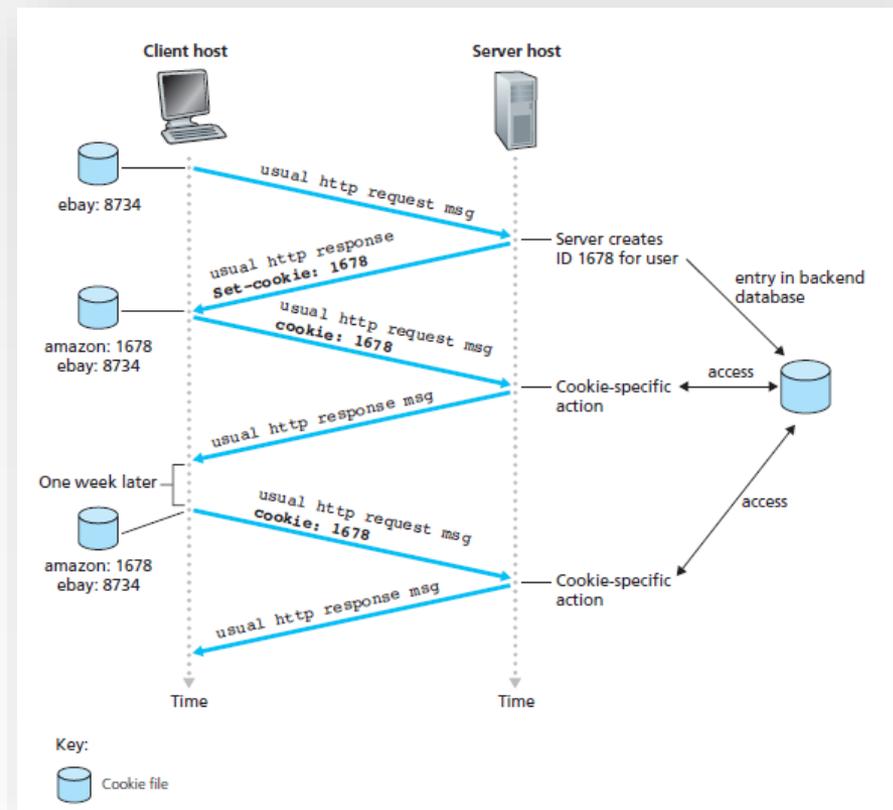
- Algunas de las **cabeceras** más comunes en las HTTP response:

Cabecera	Descripción	Ejemplo
Date	Fecha en que fue mandada la respuesta	Date: Tue, 15 Nov 1994 08:12:31 GMT
Server	Tipo de servidor	Server: Apache/2.4.1 (Unix)
Content-Type	Tipo MIME del cuerpo de respuesta	text/html; charset=UTF-8
Content-Length	Tamaño del cuerpo en bytes	Content-Length: 348
Last-Modified	Respuesta al GET condicional. Fecha en la que fue modificado el recurso	Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT

3. Navegadores y servidores web

Protocolo HTTP

- HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre clientes
- Para guardar información sobre el estado de los clientes, HTTP usa el mecanismo de **cookies**
- La tecnología de cookies tiene 4 componentes:
 - Cabecera `Set-cookie` en respuesta
 - Cabecera `Cookie` en petición
 - Fichero de cookie almacenado en el navegador
 - Base de datos de cookies en el servidor web



3. Navegadores y servidores web

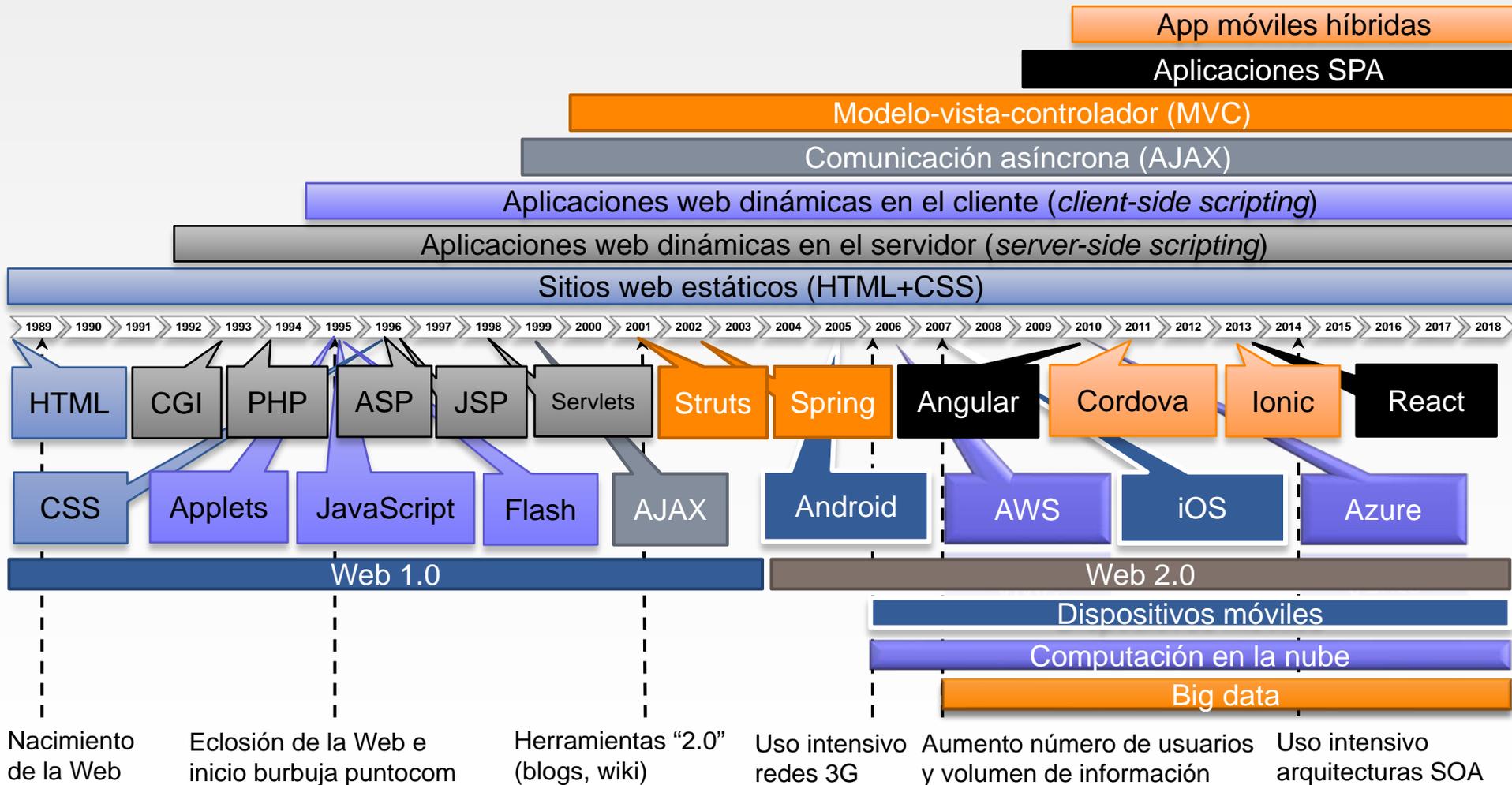
Protocolo HTTP

- En el año 2009 la Unión Europea publicó la [Directiva 2009/136/CE](#), en la que se regula el uso de las cookies en la Web
- En marzo de 2012, el [Real Decreto 13/2012](#) modificó el [artículo 22.2 de la Ley de la Sociedad de Servicios de la Información \(LSSICE\)](#) para adecuarse a la directiva europea
- Esta misma Ley dice que debes informar al usuario:
 - Debes contar con su consentimiento para usar cookies
 - Qué es una cookie y qué cookies usas una web
 - Cómo acceder a ellas en tu navegador

Índice

1. Introducción
2. Evolución histórica de la Web
3. Navegadores y servidores web
4. Evolución histórica de las aplicaciones web
 - Sitios web estáticos
 - Aplicaciones web dinámicas en el servidor
 - Aplicaciones web dinámicas en el cliente
 - Aplicaciones web dinámicas basadas en plantillas
 - Aplicaciones web dinámicas basadas en MVC
 - Comunicación asíncrona cliente-servidor
 - Aplicaciones móviles web e híbridas
 - Aplicaciones web SPA
 - Aplicaciones web en la nube
5. Tecnologías del cliente
6. Tecnologías del servidor
7. Aplicaciones web en dispositivos móviles
8. Sistemas gestores de contenido
9. Lenguajes de marcado ligeros

4. Evolución histórica de las aplicaciones web



4. Evolución histórica de las aplicaciones web

Sitios web estáticos

- Desde el inicio de la Web los dos pilares fundamentales han sido el protocolo HTTP y el lenguaje HTML
- Al principio los servidores web se limitaban a publicar un conjunto de páginas web en lenguaje HTML (**sitios web**)
- El navegador realiza peticiones HTTP al servidor, que va al disco duro, lee el fichero HTML y lo envía como respuesta
- La Web (1.0) popularizó el intercambio de información sobre Internet, pero era un modelo demasiado rígido (las páginas web eran estáticas)
- En 1996 surgen las hojas de estilo en cascada (CSS) que permiten dar estilos a las páginas web

4. Evolución histórica de las aplicaciones web

Aplicaciones web dinámicas en el servidor

- La popularidad de la web propició la creación de tecnologías dinámicas de servidor web (*server-side scripting*) naciendo las **aplicaciones web**
- La idea fundamental era crear páginas web dinámicas creadas con cada petición HTTP (la respuesta podía cambiar en función de la petición)
- La primera de estas tecnologías fue **CGI** (*Common Gateway Interface*) en 1993
- Le siguen otras tecnologías equivalentes, como los **servlets** en Java
- Surge la arquitectura de aplicación web en 3 capas:



4. Evolución histórica de las aplicaciones web

Aplicaciones web dinámicas en el cliente

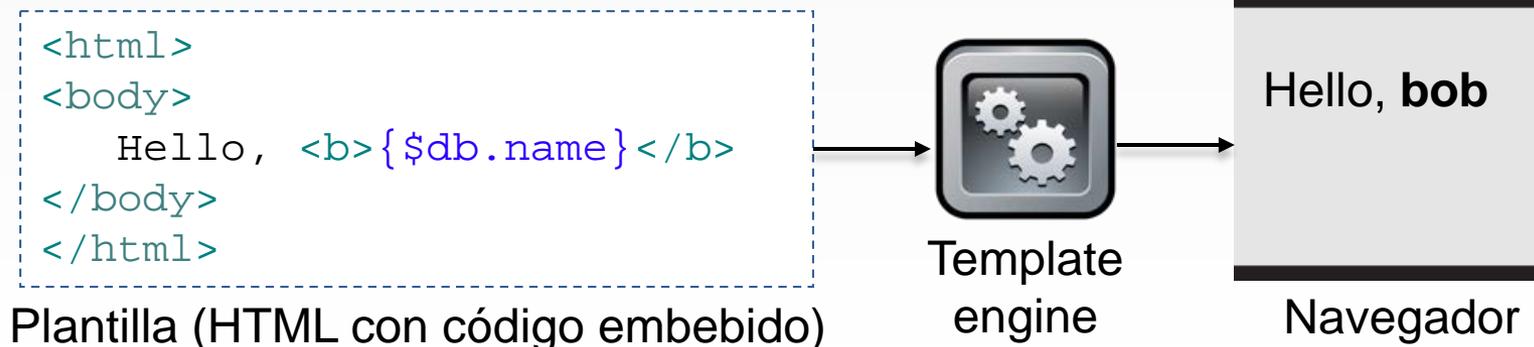
- En 1995 el navegador Netscape incorpora **JavaScript** por primera vez
- De esta forma las aplicaciones web se convierten en dinámicas también en el cliente:
 - Validación de formularios
 - Manipulación del contenido de páginas web
 - Menús desplegables y otros efectos
- También en 1995 nacen las **applets**, aplicaciones Java embebidas en páginas web y ejecutadas en el navegador
- En 1996 se desarrolla Adobe **Flash**, que fue una tecnología privativa capaz de dotar dinamismo en la parte cliente

} Ambas tecnologías están en desuso hoy día

4. Evolución histórica de las aplicaciones web

Aplicaciones web dinámicas basadas en plantillas

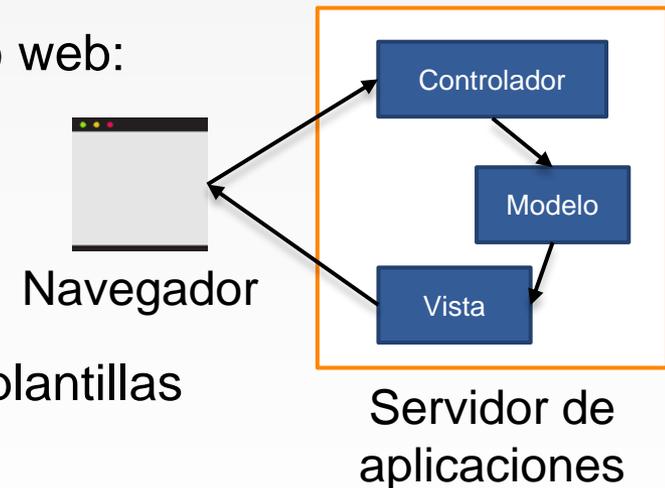
- Los sistemas basados en plantillas han sido (y son) ampliamente utilizados para la generación de páginas web de forma dinámica
- La primera tecnología de este tipo es **PHP** (1994)
- En 1996 le siguieron otras tecnologías similares:
 - **ASP** (*Active Server Pages*), basado en VBScript de Microsoft
 - **JSP** (*Java Server Pages*), basado en Java (originalmente de Sun Microsystems)



4. Evolución histórica de las aplicaciones web

Aplicaciones web dinámicas basadas en MVC

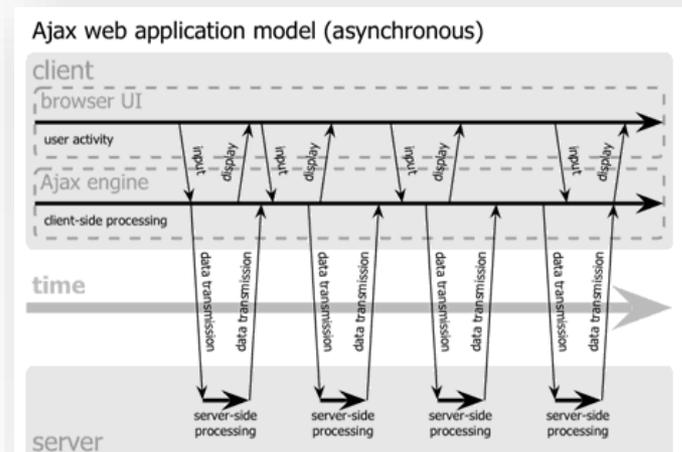
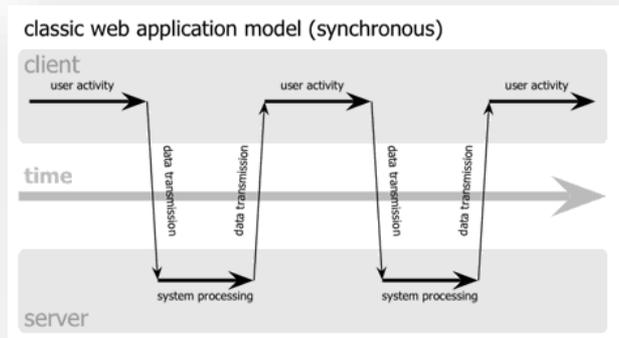
- Es un patrón arquitectónico que permite la implementación de interfaces de usuario basado en la separación de tres componentes:
 - Modelo: representa la información que maneja el sistema
 - Vista: implementa la presentación mediante una interfaz de usuario
 - Controlador: responde a eventos de usuario
- Existe multitud de frameworks MVC en el mundo web:
 - Struts, Spring MVC (Java)
 - ASP.NET MVC (.Net)
 - Ruby on Rails (Ruby)
 - Django (Python)
 - Grails (Groovy)
- Se suele usar en conjunción con un sistema de plantillas
 - Por ejemplo: Spring MVC y Thymeleaf



4. Evolución histórica de las aplicaciones web

Comunicación asíncrona cliente-servidor

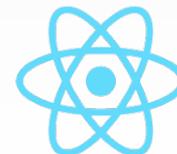
- En 1996 en navegador Internet Explorer introduce el objeto *XMLHttpRequest*
- Este objeto nos permite hacer peticiones en segundo plano (sin recargar la página) a un servidor web
- A esta técnica se la conoce como **AJAX** (*Asynchronous JavaScript And XML*)



4. Evolución histórica de las aplicaciones web

Aplicaciones web SPA

- Las aplicaciones que siguen la arquitectura **SPA** (*Single Page Application*) llevan la técnica AJAX al extremo ya que todo el contenido se carga con JavaScript en segundo plano
- Este tipo de aplicaciones se han hecho muy populares gracias a Gmail y Maps de Google
- Actualmente existen diferentes frameworks que han adoptado el paradigma SPA:
 - Ember.js: <https://www.emberjs.com/>
 - Angular: <https://angular.io/>
 - React: <https://reactjs.org/>



React

4. Evolución histórica de las aplicaciones web

Aplicaciones móviles web e híbridas

- En los primeros años del siglo XXI eclosionó el sector móvil:
 - Generalización de redes 3G
 - Comercialización de dispositivos móviles Android y iOS
- Hoy día existen diferentes tipos de dispositivos móviles: *smartphones*, *tablets*, y *wearable* (relojes, gafas)
- Debido a sus características (pequeña pantalla y control táctil) es recomendable que exista una versión adaptada de la aplicación web (***responsive web design***)
- El framework Apache **Cordova** permite usar tecnología web de lado cliente (HTML, CSS, JavaScript) para implementar **aplicaciones móviles híbridas**



4. Evolución histórica de las aplicaciones web

Aplicaciones web en la nube

- La llegada de la Web 2.0 junto con la generalización de depósitos móviles supuso un incremento notable del número de usuarios y volumen de información manejada por estos
- Esto a supuesto el nacimiento de nuevos paradigmas tecnológicos:
 - *Big Data*: Manejo de grandes volúmenes de datos (NoSQL, MapReduce)
 - *Cloud Computing*: Servicios con acceso remoto, elásticos, y con recursos compartidos
- Hoy día son cada vez más frecuentes los sistemas escalables y tolerantes a fallos basados en sistemas *cloud*
 - Servicios REST
 - Arquitectura orientada a servicios (SOA)
 - Arquitecturas de microservicios

Índice

1. Introducción
2. Evolución histórica de la Web
3. Navegadores y servidores web
4. Evolución histórica de las aplicaciones web
5. **Tecnologías del cliente**
 - Estándares web
 - HTML
 - CSS
 - JavaScript
6. Tecnologías del servidor
7. Aplicaciones web en dispositivos móviles
8. Sistemas gestores de contenido
9. Lenguajes de marcado ligeros

5. Tecnologías del cliente

Estándares web

- El cliente web por excelencia es el navegador web
- Existen un conjunto de estándares web, definidos por el **W3C**, que todo navegador debería implementar
- La familia de estándares W3C comprende diferentes tecnologías: HTML/CSS, Ajax, Servicios web, accesibilidad, etc.
- Para saber qué estándares soporta cada versión de cada navegador, se puede usar la web <http://caniuse.com/>



<http://www.w3.org>

5. Tecnologías del cliente

HTML

- La versión actual es HTML 5.1 (HTML 5.2 está en desarrollo)
- Ha supuesto una revolución para el dinamismo en el cliente porque ofrece muchas características avanzadas:
 - Multimedia: etiquetas vídeo, audio y canvas
 - Comunicación full-dúplex cliente-servidor: WebSockets
 - Concurrencia en JavaScript: Web Workers
 - Comunicación en tiempo real: WebRTC
 - Geolocalización
 - Arrastrar y soltar (*drag and drop*)



5. Tecnologías del cliente

CSS

- CSS (*Cascading Style Sheets*) es un lenguaje utilizado para dar estilo a contenido estructurado
- Se aplica principalmente a documentos HTML, pero también se puede usar con otros documentos como
 - SVG (*Scalable Vector Graphics*): Gráficos vectoriales
 - XML (*eXtensible Markup Language*): Lenguaje de marcado
- Con CSS se pueden especificar:
 - Colores: principal, de fondo, degradados, etc.
 - Tipografía: familia, tamaños, etc.
 - Layout: Disposición de los elementos en el documento
 - Efectos: sombras, esquinas redondeadas, etc.
- Su versión actual es CSS 3



5. Tecnologías del cliente

JavaScript

- Las páginas web se pueden dinamizar con JavaScript
- Se puede modificar la página y ejecutar código cuando se interactúa con ella mediante la API DOM (*Document Object Model*)
- JavaScript es un lenguaje de programación basado en el estándar **ECMAScript** de ECMA (otra organización diferente al W3C)
- Hay ligeras diferencias en la implementación de JavaScript de los navegadores, aunque actualmente todos son bastante compatibles entre sí
- Aunque algunos elementos de la sintaxis recuerden a Java, el lenguaje es muy diferente a Java. El nombre JavaScript se eligió al publicar el lenguaje en una época en la que Java estaba en auge y fue principalmente por marketing

<http://www.ecma-international.org/>

Índice

1. Introducción
2. Evolución histórica de la Web
3. Navegadores y servidores web
4. Evolución histórica de las aplicaciones web
5. Tecnologías del cliente
6. **Tecnologías del servidor**
 - **Java Enterprise Edition**
 - **PHP**
 - **ASP.NET**
 - **Node.js**
7. Aplicaciones web en dispositivos móviles
8. Sistemas gestores de contenido
9. Lenguajes de marcado ligeros

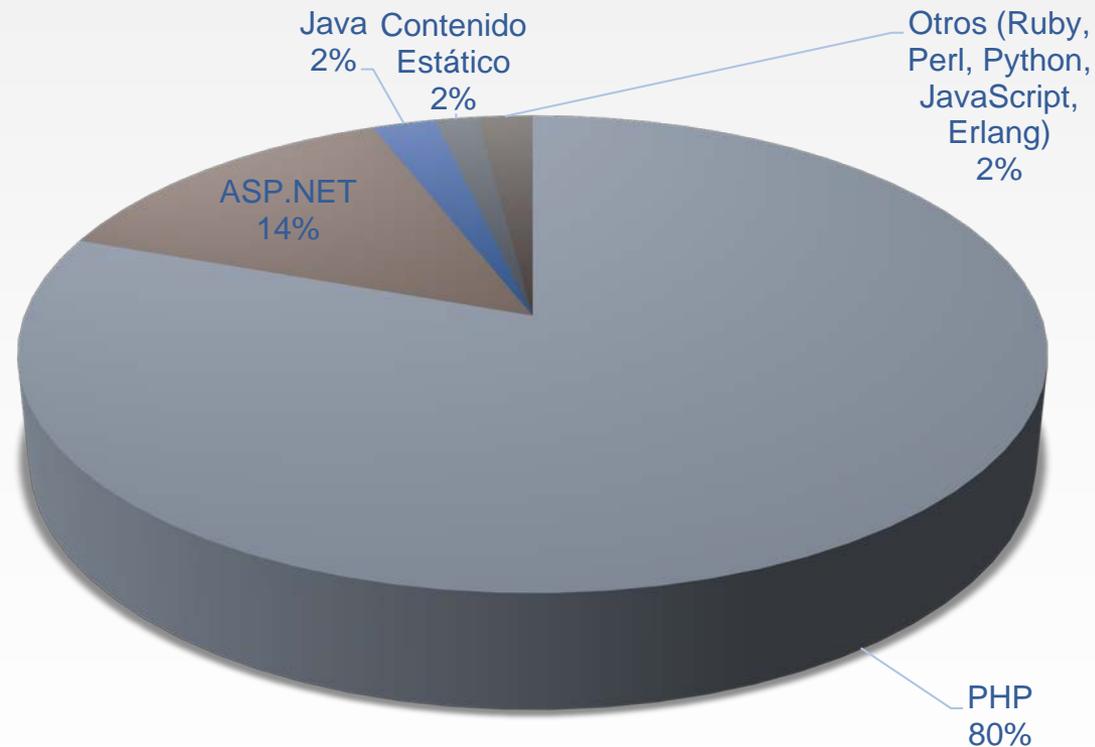
6. Tecnologías del servidor

- Los estándares son muy importantes en los navegadores web porque la web tiene que ser compatible con cualquier dispositivo
- En cambio los estándares no son necesarios en el servidor, porque cada organización desarrollará su aplicación en el servidor con la tecnología de su elección
- En el servidor, se pueden usar multitud de tecnologías:



6. Tecnologías del servidor

- Cuota de uso tecnologías del servidor (febrero 2018):



http://w3techs.com/technologies/overview/programming_language/all

6. Tecnologías del servidor

PHP

- Desarrollado en 1994 por Rasmus Lerdorf
- Fue una de las primeras tecnologías libres que se popularizaron para desarrollo web
- Se integra normalmente con Apache y MySQL en entornos Linux en un paquete llamado **LAMP**
- Es la tecnología de programación que más sitios activos tiene en Internet
 - Facebook es una muestra importante de la popularidad de PHP
 - CMSs como Drupal y Wordpress también están implementados en PHP



<http://www.php.net/>

6. Tecnologías del servidor

ASP.NET

- ASP.NET es un framework *open source* creado por Microsoft para el desarrollo de aplicaciones web dinámicas
- Está basado en ASP (*Active Server Pages*) original
- Integrada en la tecnología .NET de Microsoft (SQL Server, etc)
- Implementa un sistema de plantillas (*template engine*)
- Permite escribir código fuente en lenguaje C# (ficheros .cs) o Visual Basic (ficheros .vb)



ASP.NET

<http://www.asp.net/>

6. Tecnologías del servidor

Java Enterprise Edition

- Tecnología basada en **Java**
- Desarrollada por una coalición de empresas lideradas por Oracle, IBM, Red Hat, etc..
- Tecnología muy usada a nivel empresarial
- Existen muchos estándares e implementaciones pertenecientes a Java EE: Servlets, JSP, JDBC, JPA, JSF, EJBs...
- **Spring** es el framework de desarrollo de aplicaciones empresariales basado en tecnologías Java más popular



<http://www.oracle.com/javaee/>



<https://spring.io/>

6. Tecnologías del servidor

Node.js

- **Node.js** es un *entorno de ejecución* de aplicaciones **JavaScript** basada en V8 (el motor de JavaScript de Google)
- Podemos ver Node.js como una capa de software que permite la ejecución de aplicaciones desarrolladas con JavaScript
- El gestor de paquetes en Node.js se llama **NPM** (*Node.js Package Manager*)



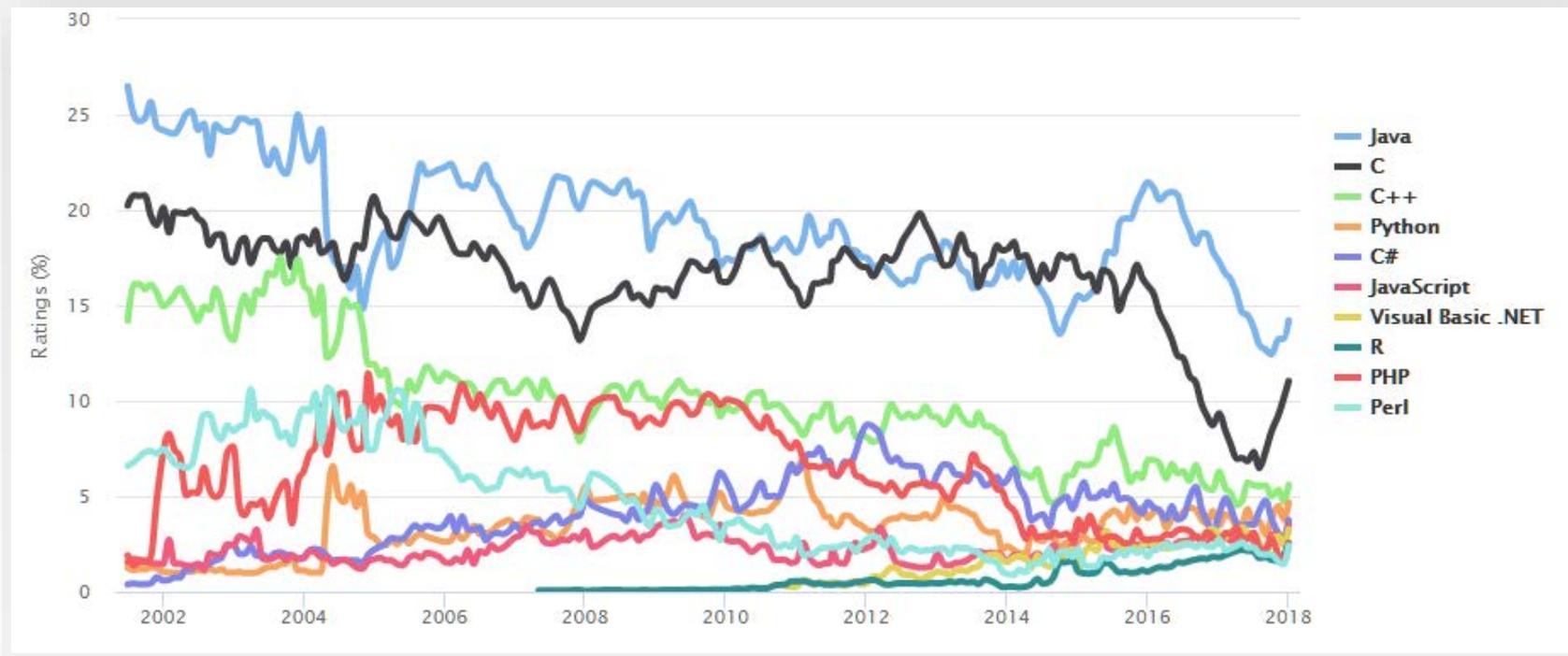
<https://nodejs.org/>



<https://www.npmjs.com/>

6. Tecnologías del servidor

- Índice TIOBE (febrero 2018):



<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

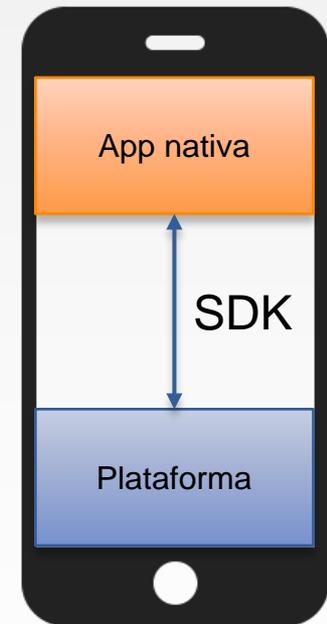
Índice

1. Introducción
2. Evolución histórica de la Web
3. Navegadores y servidores web
4. Evolución histórica de las aplicaciones web
5. Tecnologías del cliente
6. Tecnologías del servidor
7. **Aplicaciones web en dispositivos móviles**
 - Aplicaciones móviles nativas
 - Aplicaciones móviles híbridas
 - Aplicaciones móviles progresivas
8. Sistemas gestores de contenido
9. Lenguajes de marcado ligeros

7. Aplicaciones web en dispositivos móviles

Aplicaciones móviles nativas

- Las aplicaciones **nativas** de las plataformas móviles se implementan con diferentes tecnologías:
 - Android:** Java
 - iOS:** Objective-C

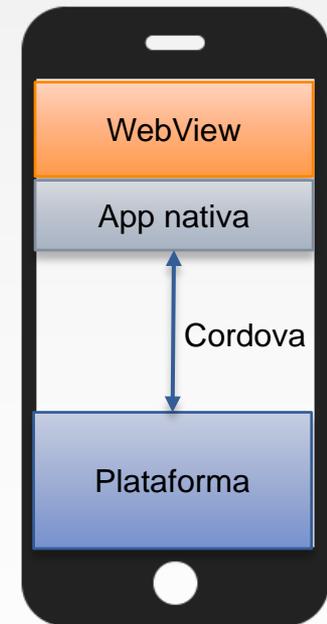


Aplicación móvil nativa

7. Aplicaciones web en dispositivos móviles

Aplicaciones móviles híbridas

- Las aplicaciones móviles **híbridas** permiten el uso de tecnologías web de lado cliente (HTML, CSS y JavaScript) para el diseño de interfaces de usuario
- Reducen el esfuerzo de desarrollo, ya que la misma aplicación se empaqueta para diferentes plataformas
- Requieren de un framework que realice el empaquetado de la aplicación: **Cordova**



Aplicación móvil híbrida

7. Aplicaciones web en dispositivos móviles

Aplicaciones móviles híbridas

- **Apache Cordova** es un framework *open source* que permite empaquetar aplicaciones implementadas con tecnologías web de lado cliente (HTML, CSS, JavaScript) como aplicaciones móviles
- **PhoneGap** es una distribución de Apache Cordova propiedad de Adobe



<https://cordova.apache.org/>



<http://phonegap.com/>

7. Aplicaciones web en dispositivos móviles

Aplicaciones móviles híbridas

- Las aplicaciones empaquetadas directamente con Cordova suelen ser poco usables (peor apariencia que las apps nativas)
- **Ionic** es un SDK (*Software Developer Kit*) *open source* para el desarrollo de aplicaciones móviles basadas en HTML, CSS y JavaScript
 - El objetivo de Ionic es conseguir desarrollar aplicaciones móviles de apariencia muy similar a las nativas, pero con tecnologías web
 - Ionic usa Cordova y Angular

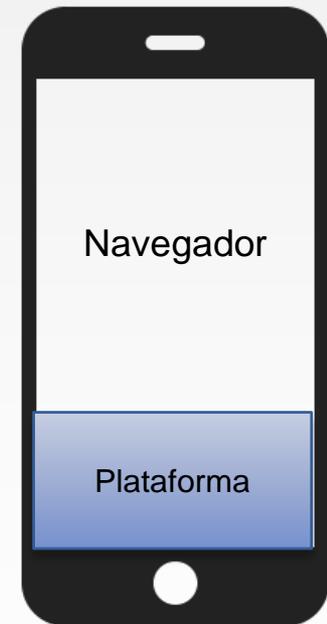


<http://ionic.io/>

7. Aplicaciones web en dispositivos móviles

Aplicaciones móviles progresivas

- Las aplicaciones móviles **progresivas** (PWA, *Progressive Web Apps*) son aplicaciones web que usan diferentes APIs web para ser consumidas como aplicaciones nativas
 - Instalación en la pantalla de inicio del móvil
 - Notificaciones push
 - Uso del almacenamiento local del dispositivo



Aplicación móvil progresiva

Índice

1. Introducción
2. Evolución histórica de la Web
3. Navegadores y servidores web
4. Evolución histórica de las aplicaciones web
5. Tecnologías del cliente
6. Tecnologías del servidor
7. Aplicaciones web en dispositivos móviles
8. **Sistemas gestores de contenido**
9. Lenguajes de marcado ligeros

8. Sistemas gestores de contenido

- Los sistemas gestores de contenido (**CMS**, **Content Management System**) son aplicaciones web genéricas que permiten la creación y administración de contenidos vía web
- El sistema permite manejar de manera independiente el contenido y el diseño, permite el cambio de diseño (con *templates* o *themes*)
- Los CMSs han evolucionado para convertirse en un nuevo modelo de desarrollo de aplicaciones web configurando y adaptando módulos con un interfaz web

8. Sistemas gestores de contenido

- Existen multitud de CMSs con enfoques y objetivos diferentes
- Ejemplos: Drupal (PHP), Joomla (PHP), Wordpress (PHP), Plone (JavaScript), Liferay (Java)...



- Existe CMS específicos para comercio electrónico (*eCommerce*), por ejemplo: Magento (PHP), PrestaShop (PHP), OsCommerce (PHP)...



Índice

1. Introducción
2. Evolución histórica de la Web
3. Navegadores y servidores web
4. Evolución histórica de las aplicaciones web
5. Tecnologías del cliente
6. Tecnologías del servidor
7. Aplicaciones web en dispositivos móviles
8. Sistemas gestores de contenido
9. **Lenguajes de marcado ligeros**

9. Lenguajes de marcado ligeros

- Un **lenguaje de marcado** permite añadir información semántica a un conjunto de datos mediante **etiquetas**. Por ejemplo:

HTML	XML
<pre> <!DOCTYPE html> <html lang="en"> <body> <h1>This is heading 1</h1> <p>This paragraph one.</p> <p>This paragraph two.</p> </body> </html> </pre>	<pre> <project> <dependencies> <dependency> <groupId>junit</groupId> <artifactId>junit</artifactId> <version>4.12</version> <scope>test</scope> </dependency> </dependencies> </project> </pre>

9. Lenguajes de marcado ligeros

- El problema que tienen los lenguajes de marcados es que habitualmente se hacen demasiado **complejos**
- Por esta razón, en los últimos años se está popularizando el uso de **lenguajes de marcado ligeros**
- El objetivo de estos lenguajes es ser más sencillos (mejor legibilidad por parte de las personas). Por ejemplo:

JSON	YAML
<pre>{ "name": "hello-world", "version": "1.0.0", "dependencies": { "cookie-parser": "^1.3.5", "minimist": "^1.1.1" } }</pre>	<pre>language: java sudo: true dist: trusty script: - mvn verify</pre>

9. Lenguajes de marcado ligeros

- Existen ciertos lenguajes de marcado ligeros que acaban generando HTML:

- Markdown
- MediaWiki
- reStructuredText
- AsciiDoc



reStructuredText

AsciiDoc

9. Lenguajes de marcado ligeros

■ Ejemplo de Markdown:

```
# Web Programming Examples

This project contains code examples for web
applications using Java in the server-side.

## List of technologies

The main technologies involved in these examples
are:

* Spring : Java Framework. These examples are
based on Spring Boot.
* Thymeleaf : Template engine for Spring MVC.
* JUnit : Unit testing framework.
* Selenium : Web testing framework.

## About

This is a project made by [Boni Garcia], Professor
at [U-tad]. Copyright &copy; 2017.

[U-tad]: http://www.u-tad.com/
[Boni Garcia]: http://bonigarcia.github.io/
```

Web Programming Examples

This project contains code examples for web applications using Java in the server-side.

List of technologies

The main technologies involved in these examples are:

- Spring : Java Framework. These examples are base on Spring Boot.
- Thymeleaf : Template engine for Spring MVC.
- JUnit : Unit testing framework.
- Selenium : Web testing framework.

About

This is a project made by [Boni Garcia](#), Professor at [U-tad](#). Copyright © 2017.

9. Lenguajes de marcado ligeros

■ Ejemplo de MediaWiki:

```
== Worlds in literature ==
[[File:VisualEditor - 2014-02 Metrics
deck.pdf|left|thumb]]
Creating a different world is a literary
device used by authors to illustrate ideas. By
placing the story in the setting of a
different world, the author can change the way
that things happen in the world. For example,
the author might imagine a world that has very
little water or a world that has very little
dry land. Deciding what the world looks like
and how the world works is called ''world-
building''. Thinking about the world helps the
author make good choices about what happens to
the characters in the story. Some authors
think about many details, such as what
[[languages]] the characters speak and what
the [[architecture]] is on the world.
```

Worlds in literature [\[edit \]](#)



Creating a different world is a literary device used by authors to illustrate ideas. By placing the story in the setting of a different world, the author can change the way that things happen in the world. For example, the author might imagine a world that has very little water or a world that has

very little dry land. Deciding what the world looks like and how the world works is called *world-building*. Thinking about the world helps the author make good choices about what happens to the characters in the story. Some authors think about many details, such as what [languages](#) the characters speak and what the [architecture](#) is on the world.

9. Lenguajes de marcado ligeros

▪ Ejemplo de reStructuredText:

```
Section Header
```

```
=====
```

```
Subsection Header
```

```
-----
```

```
This is an example.
```

```
- A bullet list item
```

```
- Second item
```

```
  - A sub item
```

```
1) An enumerated list item
```

```
2) Second item
```

```
.. code:: bash
```

```
    find . | grep java
```

Section Header

Subsection Header

This is an example.

- A bullet list item
- Second item
 - A sub item

1. An enumerated list item
2. Second item

```
find . | grep java
```

9. Lenguajes de marcado ligeros

■ Ejemplo de AsciiDoc:

```
Welcome to AsciiDocLIVE!  
-----  
  
AsciiDocLIVE is a *free online  
http://www.methods.co.nz/asciidoc/[AsciiDoc^]  
editor*.  
  
* Just type AsciiDoc source text into the *left*  
pane,  
* ...and the live preview appears in the *right*  
pane!  
  
What's AsciiDoc?  
~~~~~  
  
AsciiDoc is a human-readable text document format for  
writing notes, documentation, articles, books,  
ebooks, slideshows, web pages, man pages and blogs,  
and more. AsciiDoc files can be translated to many  
formats including HTML, PDF, EPUB, and man page.  
  
To learn more, visit the AsciiDoc home page at  
http://www.methods.co.nz/asciidoc/[^].
```

Welcome to AsciiDocLIVE!

AsciiDocLIVE is a **free online AsciiDoc editor**.

- Just type AsciiDoc source text into the **left** pane,
- ...and the live preview appears in the **right** panel!

What's AsciiDoc?

AsciiDoc is a human-readable text document format for writing notes, documentation, articles, books, ebooks, slideshows, web pages, man pages and blogs, and more. AsciiDoc files can be translated to many formats including HTML, PDF, EPUB, and man page.

To learn more, visit the AsciiDoc home page at
<http://www.methods.co.nz/asciidoc/>.

9. Lenguajes de marcado ligeros

- Últimamente se ha popularizado el uso de estos lenguajes para la generación de sitios web estáticos
- Herramientas útiles:
 - Jekyll: Convierte Markdown (entre otros formatos) a HTML
 - AsciiDoctor: Convierte AsciiDoc a HTML, PDF, EPUB
 - Sphinx: Convierte reStructuredText a HTML, PDF, EPUB
 - Mkdocs: Converte Markdown a HTML/CSS
- Alojamiento gratuitos de sitios web estáticos:
 - GitHub Pages
 - Readthedocs



AsciiDoctor



MkDocs

GitHub Pages



Read *the* Docs