

# Introducción a las aplicaciones Web

## Tecnologías de la Web

---

### Aplicaciones Web/Sistemas Web



Juan Pavón Mestras  
Dep. Ingeniería del Software e Inteligencia Artificial  
Facultad de Informática  
Universidad Complutense Madrid

Material bajo licencia Creative Commons



## Web - Conceptos básicos

---

- **Lenguajes de marcado**
  - Texto + etiquetas (marcas, en inglés, *tags*)
  - Las etiquetas proporcionan información adicional acerca de la estructura del texto o su presentación
- En la Web:
  - **HTML** (HyperText Markup Language)
    - **Hipertexto:** Habilidad de navegar desde un documento a otro a través de conexiones (*hyperlinks* o *links*)
  - **XHTML** (eXtensible HyperText Markup Language)
    - HTML expresado como XML válido
    - Extiende HTML 4.0 combinando
      - HTML diseñado para mostrar datos
      - XML diseñado para describir datos
    - Puede incluir otros lenguajes (SVG, SMIL, MathML, etc.)
  - **RDF** (Resource Description Framework)
    - Framework para metadatos en la Web
  - **WSDL** (Web Services Description Language)

- Lenguaje con el que se definen páginas Web
  - Permite describir el contenido de una página, incluyendo texto y otros elementos (imágenes, videos, pequeñas aplicaciones, etc.)
- Texto plano que incluye etiquetas que sirven para indicar un tratamiento especial
  - Meta-información: `<meta name="keywords" content="Curso Web, UCM"/>`
  - Estructura del documento: `<head><body><h1>`
  - Formato: `<b>Texto en negrita</b>`
  - Enlaces a otros objetos: `<a href="http://www.ucm.es/">`
  - Formularios: `<form>`
  - Scripts: `<script>`
- Versiones actuales:
  - HTML 4.01 (W3C, 1999)
    - Estandarizado también por ISO en 2000
  - HTML 5 (W3C, 17/12/2012)
    - Para 2014 se definirán los tests de interoperabilidad
  - XHTML 1.1 (2001), actualmente definiéndose XHTML 5

## Web - Conceptos básicos

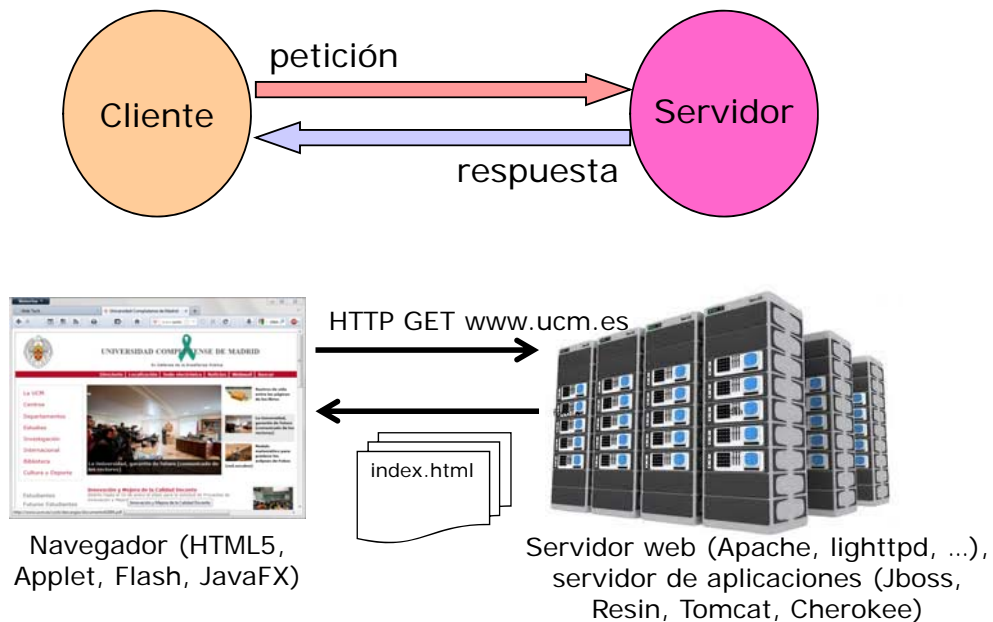
---

- **Identificadores de Recursos**
  - UniformResourceLocator (**URL**)
    - Descriptor del acceso (cómo encontrar) un recurso
    - esquema://autoridad/ruta/archivo?solicitud
      - Esquema: http, https, ftp, mailto, ldap, file, ...
      - Autoridad: www.ucm.es  
`https://cv3.sim.ucm.es/portal/help/main?help=sakai.siteinfo`
        - Se puede incluir también el puerto (por defecto 80 para http y 443 para https)  
`https://cv3.sim.ucm.es:443/portal/help/main?help=sakai.siteinfo`
  - UniformResourceIdentifier (**URI**)
    - Identificador inequívoco de un recurso
      - Definido en RFC 2396
    - Más completo que URL, permite identificar una parte del recurso principal, determinada por el "fragmento"
    - esquema://máquina/directorio/archivo?solicitud#fragmento  
`http://www.sans.org/top25-software-errors/#cat1`

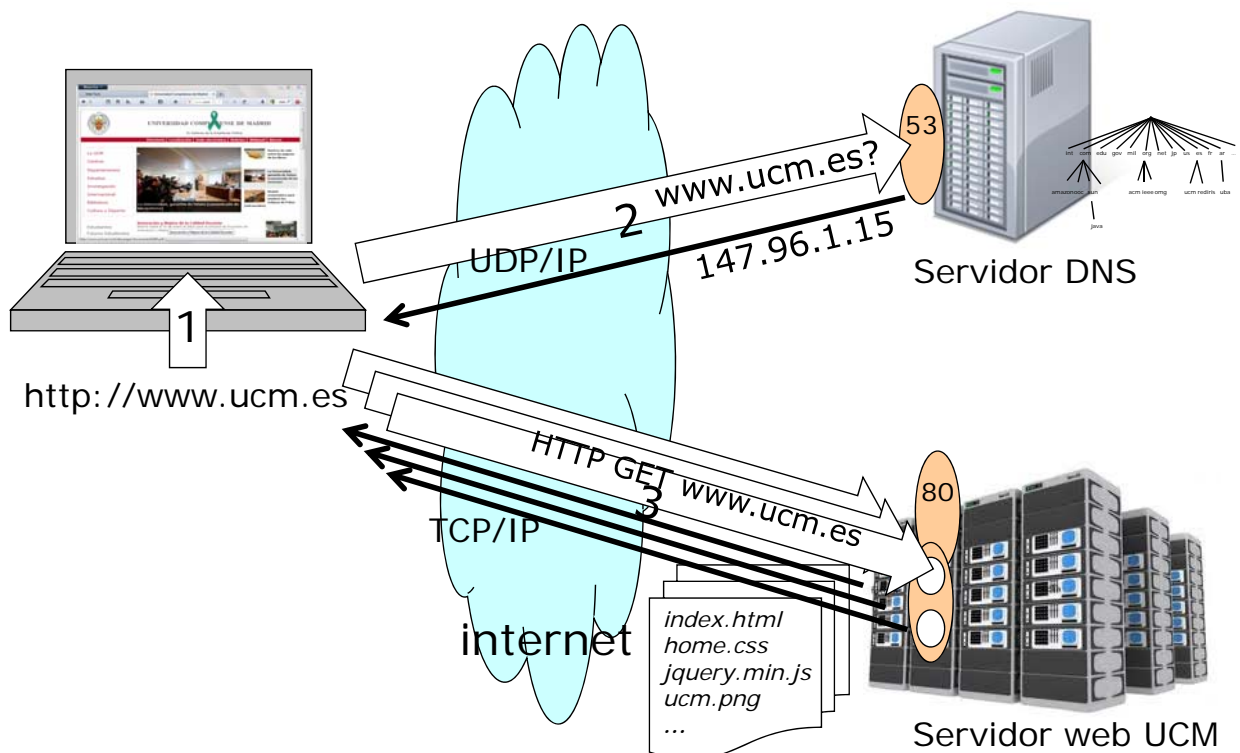
# Web - Conceptos básicos

## ■ Modelo **cliente-servidor**

- Un cliente demanda servicios o recursos a un servidor a través de una interfaz, usando un protocolo

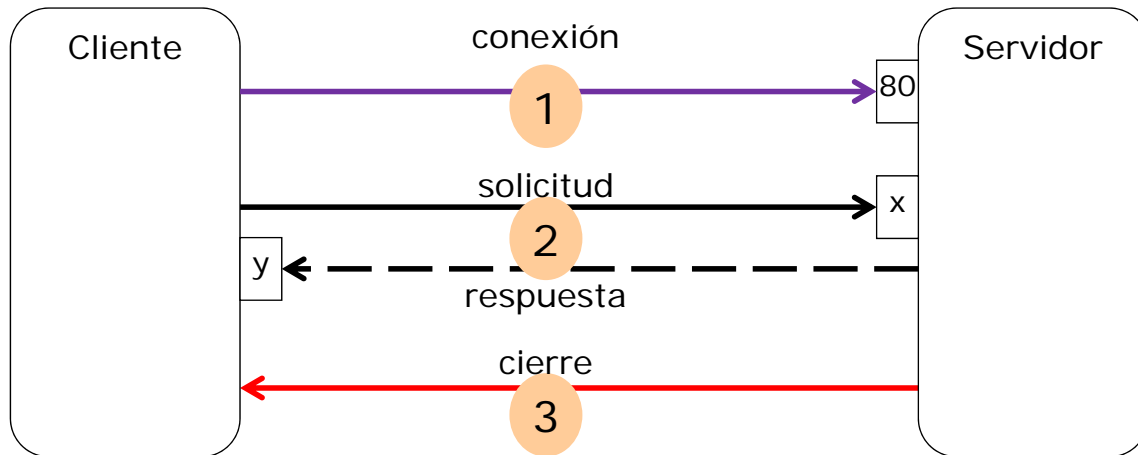


## Funcionamiento



## Protocolo HTTP

- Protocolo de aplicación para transferencia de hipertexto
  - Funciona sobre TCP/IP
- Permite a un navegador (cliente, *user agent*) solicitar una página a un servidor y que éste la envíe
- Basado en el envío de comandos y respuestas en texto ASCII



## Protocolo HTTP

- Tipos de mensajes
  - HTTP/1.0
    - GET
      - Solicitud de un recurso
    - POST
      - Envío de datos al servidor para que los procese (p.ej. con un script CGI)
      - Ejemplo: datos de un formulario
    - HEAD
      - Como el GET pero pide al servidor que solo envíe la cabecera de la respuesta (esto es, que responda sin enviar el objeto requerido)
      - Para comprobar características de un recurso sin descargarlo
  - HTTP/1.1
    - GET, POST, HEAD
    - PUT
      - Sube archivos en el cuerpo de la solicitud
    - DELETE
      - Borra el archivo especificado en el campo URL
    - Otros: OPTIONS, TRACE and CONNECT

# Protocolo HTTP

---

## ■ Mensajes HTTP

- Línea inicial
- 0..n líneas de cabecera
- Línea en blanco (CRLF)
- Cuerpo de mensaje opcional (un fichero, solicitud de datos, datos resultado de una solicitud)

<línea inicial, diferente para solicitud o respuesta>

Cabecera1: valor1

Cabecera2: valor2

Cabecera3: valor3

<opcional - cuerpo de mensaje, contenido de fichero o datos de query;  
puede tener cualquier cantidad de líneas, incluso datos binarios \$&\*%@!^\$@>

# Protocolo HTTP

---

## ■ Mensajes HTTP - Línea inicial

- Solicitud (request)
  - **GET** /path/to/file/index.html HTTP/1.0
- Respuesta
  - HTTP/1.0 200 OK
  - HTTP/1.0 404 Not Found
- Códigos de estado típicos
  - **200 OK** Solicitud exitosa – el recurso resultante sigue en el cuerpo
  - **400 Bad Request** El servidor no entiende la solicitud
  - **404 Not Found** El recurso solicitado no existe en el servidor
  - **301 Moved Permanently**
  - **302 Moved Temporarily**
  - **303 See Other** (*solo en HTTP 1.1*) El recurso se ha movido a otro URL (indicado en la cabecera **Location:**), y debe ser automáticamente recuperado por el cliente (redirección)
  - **500 Server Error** Error inesperado del servidor

## Protocolo HTTP

---

- Mensajes HTTP – Líneas de cabecera
  - Proporcionan información de la solicitud o respuesta
  - Estructura: **Nombre-cabecera: valor**
  - Nombre de cabecera
    - HTTP 1.0 define 16 (ninguno obligatorio)
    - HTTP 1.1 define 46, y requiere al menos una (**Host:**)
  - Por *net-politeness* (cortesía en la red) el cliente pone las siguientes:
    - **From:** dirección email o programa solicitante
    - **User-Agent:** identifica el programa que hace la petición con la forma: "**nombre-programa/x.xx**"  
Host:www.ucm.es  
User-Agent:Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:11.0) Gecko/20120313 Firefox/11.0
  - El servidor suele poner las siguientes:
    - **Server:** identifica el software del servidor "**Program-name/x.xx**".
    - **Last-Modified:** fecha de modificación del recurso (se usa para gestioanr las cachés). En Greenwich Mean Time  
Content-Type:text/html; charset=iso-8859-1  
Date:Fri, 14 Dec 2012 09:23:36 GMT  
Server:Apache

## Protocolo HTTP - Ejemplo

---

- Para recuperar el fichero en el URL  
http://www.dominio.com/path/fichero.html
- Solicitar una conexión en el puerto 80 del host *www.dominio.com*
- Enviar a través del nuevo socket el siguiente texto:  
GET /path/fichero.html HTTP/1.0  
From: usuario@ucm.es  
User-Agent: HTTPTool/1.0  
[línea en blanco CRLF]
- La respuesta del servidor llegará por el mismo socket:  
HTTP/1.0 200 OK  
Date: Wed, 12 Dec 2012 12:09:34 GMT  
Content-Type: text/html  
Content-Length: 1354  
  
<html>  
<body>  
<h1>Bienvenido al curso de la Web</h1>  
(resto del contenido)  
.  
.  
</body>  
</html>
- Tras enviar la respuesta el servidor cierra el socket

# Seguridad con HTTPS

- **HTTPS** (Hypertext Transfer Protocol Secure)
  - Permite que la información sensible (datos de usuario, *passwords*, pagos, etc.) no pueda ser interceptada durante la transferencia de datos
  - La información viaja por un canal cifrado sobre SSL/TLS
    - TLS (*Transport Layer Security*) es el protocolo estandarizado por IETF (RFC 2246), basado en SSL (*Secure Sockets Layer*)
      - Última versión: TLS 1.2 (RFC 5246, 2008)
    - Usa
      - Criptografía asimétrica (sistema de clave pública) para acordar el protocolo y claves de sesión
      - Criptografía simétrica para el intercambio confidencial de información
      - Códigos de autenticación de mensajes para la integridad de los mensajes
    - Requiere que el servidor tenga un certificado digital
      - Normalmente debe estar firmado por una *Autoridad de certificación* para que el navegador web lo acepte
  - Para utilizar HTTPS el servidor recibe las conexiones en el puerto 443 (por defecto)
  - La URL indica el uso de este protocolo: `https://...`

## Protocolo HTTP – Ejercicios

- Experimentar el protocolo HTTP usando telnet (en Unix)
  - `telnet www.dominio.com 80`

```
$ telnet www.google.es 80
Trying 212.106.221.27...
Connected to www.google.es.
Escape character is '^]'.
HEAD / HTTP/1.1
Host: www.google.es
HTTP/1.1 200 OK
Date: Fri, 01 Mar 2013 15:54:35 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: PREF=ID=8730c30ca4e6210e:FF=0:TM=1362153275:LM=1362153275:S=1r3Lm912j2Rdd2mS;
expires=Sun, 01-Mar-2015 15:54:35 GMT; path=/; domain=.google.es
Set-Cookie: NID=67=e839boWwCqTrzsUkPjpafLY_5vLV0YBWhX7bWejx8RvmRy1UJDbLCKFEh0jwpu0LtySsKu-
AYG0z5rvinJoYwzjLXbDgE4nHmXQIrsVJwrT2WUyeQH1mITNowDsJJ_oo; expires=Sat, 31-Aug-2013 15:54:35 GMT;
path=/; domain=.google.es; HttpOnly
P3P: CP="This is not a P3P policy! See
http://www.google.com/support/accounts/bin/answer.py?hl=en&answer=151657 for more info."
Server: gws
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

Connection closed by foreign host.
$
```

## Protocolo HTTP – Ejercicios

---

- Escribir un programa Java sencillo que implemente un cliente de HTTP
  - Usa las clases del paquete `java.net` y `java.io`
  - Otra alternativa más directa es usar la clase `JEditorPane` que puede interpretar código HTML

## Cookies

---

- HTTP es un protocolo SIN ESTADO
  - No se guarda información de la sesión/historia pasada
  - (Esto simplifica el protocolo)
- Uso de “cookies”
  - Un *cookie* es un *string* que se pasa en una cabecera HTTP y que el navegador puede guardar en un pequeño fichero de texto
    - En archivos temporales del navegador correspondiente
  - El *cookie* se reenvía luego al servidor HTTP con cada petición del cliente a ese servidor
  - Los *cookies* **no** pueden capturar información del cliente
    - Sólo recuerdan información proporcionada por el usuario al servidor (es el servidor quien los crea)
  - Usos
    - Guardar las preferencias del usuario
    - Reconocimiento de usuarios
    - Ayuda a recoger datos usados por aplicaciones de compra electrónica
      - El cookie puede guardar un identificador que permite al servidor acceder a todos los datos almacenados en su base de datos

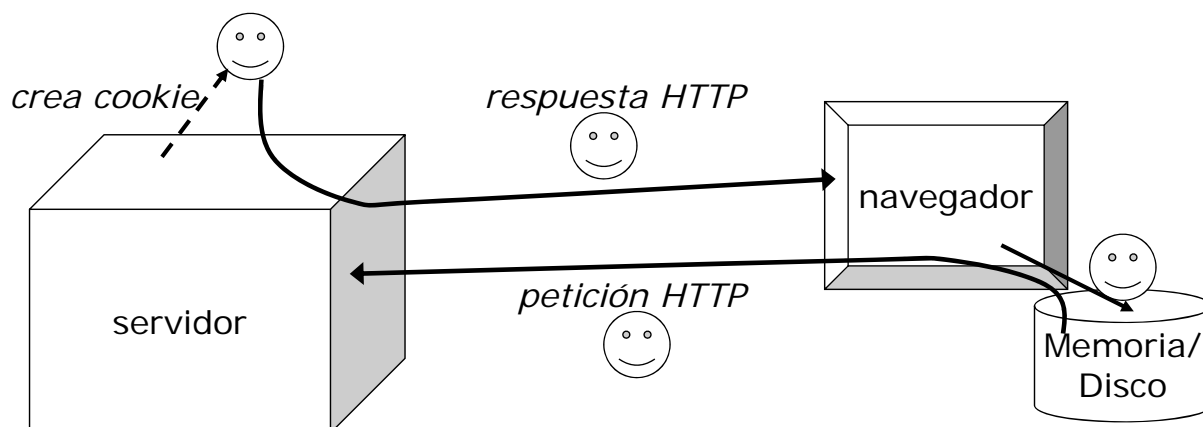


# Cookies

- Atributos
  - Par (Nombre, Valor)
  - Comentario (se puede presentar al usuario)
    - Interesante para explicar para qué se usa el cookie (política del sitio web)
  - Especificación de las páginas y dominios a los que se puede enviar el *cookie*
  - Fecha y hora de expiración
    - Permite controlar por ejemplo el tiempo máximo de una sesión antes de volver a pedir login
  - Requiere o no una página segura
  - Versión
- Tamaño máximo: 4Kbytes (Normalmente ocupan alrededor de 100 bytes)
- Seguridad
  - Los *cookies* sólo pueden ir al dominio especificado
  - No conviene poner información sensible en el *cookie*, mejor utilizar un identificador en el *cookie* que sirva de clave de acceso en la base de datos del servidor

# Cookies

- Funcionamiento de los Cookies



## Programación de cookies con Java

---

- Creación y envío de un cookie
  - Por un *servlet* como respuesta a una petición HTTP
  - clase *javax.servlet.http.Cookie*

```
// 1. crea el cookie
Cookie cookie = new Cookie("nombre", "valor");
```
  - El cookie se envía como parte de una respuesta HTTP

```
// 2. envía el Cookie en un HttpServletResponse
public void doGet (HttpServletRequest, request,
    HttpServletResponse response) throws IOException
{
    response.addCookie (cookie));
}
```

## Programación de cookies con Java

---

- Envío de un identificador único
  - La clase *java.rmi.server.UID* permite crear identificadores que son únicos dentro de la máquina en que se generan

```
String uid = new java.rmi.server.UID().toString();
```
  - Para enviarlo por HTTP hace falta codificarlo con el método *encode()* de *java.net.URLEncoder*, que convierte el string al formato MIME denominado *x-www-form-urlencoded*
    - 'a' - 'z', 'A' - 'Z', y '0' - '9' no se modifican.
    - El carácter de espacio se transforma en '+'.
    - Los demás caracteres se convierten en string de 3-caracteres "%xy", donde xy es la representación hexadecimal con dos dígitos de los 8-bits del carácter

```
Cookie cookie = new Cookie("uid", java.net.URLEncoder.encode(uid));
// ...
response.addCookie (cookie));
```

## Programación de cookies con Java

---

### ■ Recepción de cookies

- Con una respuesta HTTP se pueden recibir varios cookies asociados

```
public void doGet (HttpServletRequest, request,
    HttpServletResponse response) throws IOException
{
    Cookie[] cookies = request.getCookies();
    if ( cookies != null )
        for (int i=0; i<cookies.length; i++) {
            Cookie unCookie = cookies[i];
            System.out.println(" nombre: " + unCookie.getName()
                + ", valor: " + unCookie.getValue());
        }
}
```

## Programación de cookies con Java

---

### ■ Configuración de cookies

- La clase Cookie ofrece operaciones para configurar los atributos de cada cookie

- Tiempo de expiración

```
cookie.setMaxAge (numero_segundos);
// si 0, entonces el cookie expira inmediatamente
// si negativo, entonces el cookie expira al apagar el navegador
```

- Comentario

```
cookie.setComment ("comentario");
```

- Dominio

```
cookie.setDomain ("patron_de_dominio");
```

- Valor (para modificar el que se le dio al crearlo)

```
cookie.setValue ("nuevovalor");
```

## Alternativas a los cookies

---

- Campos ocultos de los formularios
  - Obliga a procesar cada petición de página con el mecanismo de enviar formulario: anticuado (ver HTML dinámico)
- Añadir información de estado al final del URL mediante query string del URL
  - `http://maquina/pagina.html?parametro=valor`
  - Poco seguro

## Ejercicios sobre cookies

---

Los cookies se pueden programar también con otros lenguajes. Mira el código de las siguientes páginas:

- Para PHP:  
<http://www.mclibre.org/consultar/php/ejercicios/cookies/cookies.html>
- Para Javascript:  
<http://www.iec.csic.es/criptonomicon/cookies/ejemplos.html>

# Tecnologías de programación de aplicaciones Web

---

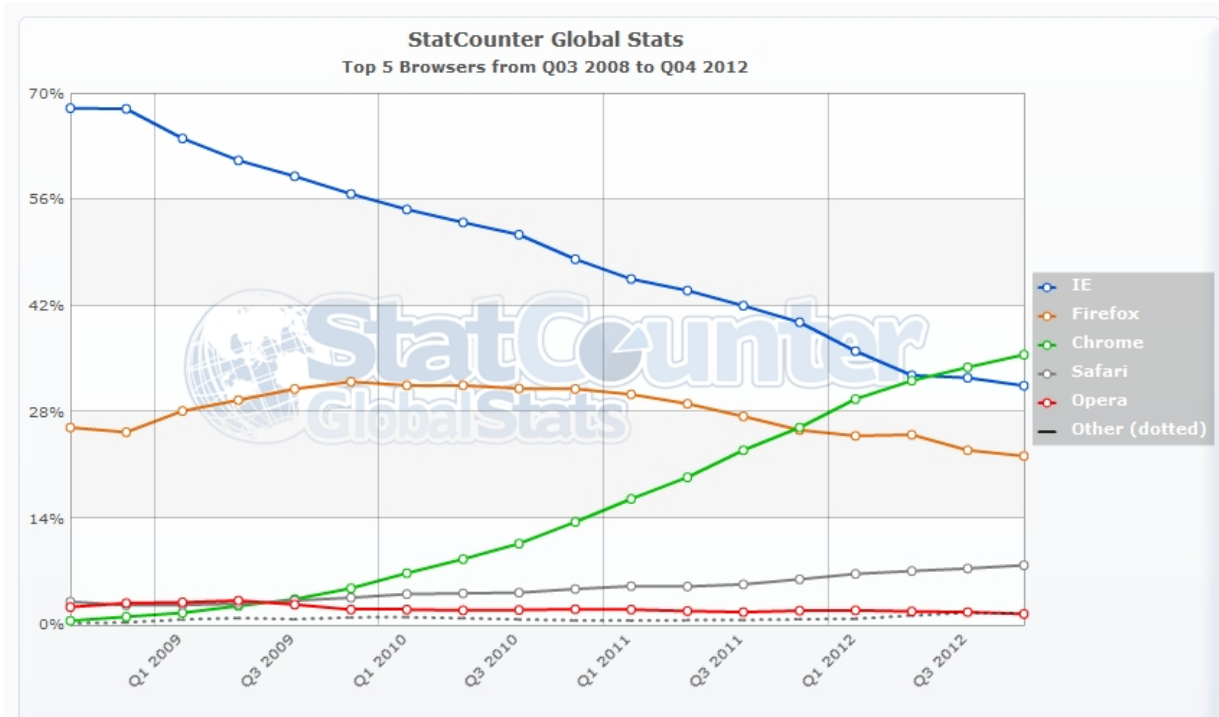
- En el cliente:
  - Navegadores
  - Lenguajes de programación
- En el servidor:
  - Servidores
  - Lenguajes de programación
  - Gestores de contenidos
- Estándares en la web
  - Protocolos y Lenguajes
  - Accesibilidad

## Navegadores (*browsers*)

---

- Permiten acceder a la web y visualizar en modo gráfico documentos HTML (XHTML)
  - Procesan también otros tipos de objetos: imágenes, sonidos, videos, scripts, etc.
  - Pueden arrancar aplicaciones que traten los ficheros recibidos
- Aceptan la instalación de plugins (módulos con funcionalidad extra)
  - Para procesar ciertos tipos de documentos (p.ej, PDF)
  - Presentaciones flash
- Más populares:
  - WorldWideWeb (Tim Berners-Lee, 1991)
  - Mosaic (NCSA, 1993)
  - Mozilla (1998)
    - Netscape Navigator (2000): Introduce JavaScript en la v2 (1995)
    - Firefox (2002)
  - Microsoft Internet Explorer (1995): solo en Windows
  - Safari (2003): en Mac OS X [Apple llegó bien tarde a la Web...]
  - Chrome (2008): multiplataforma
  - Opera (1996): el que más tecnologías soporta, multiplataforma

## Navegadores (*browsers*)



Información de <http://gs.statcounter.com/>

## Generación dinámica de información en el cliente

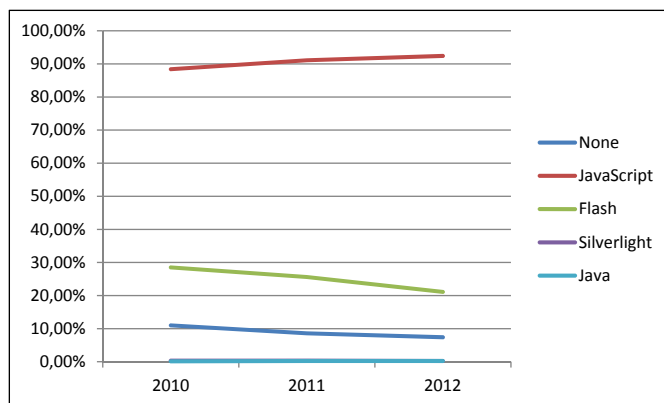
- En una página HTML se pueden incrustar elementos computacionales y scripts:
  - `<object>`
    - El navegador reconoce el tipo de elemento y lo ejecuta (requiere normalmente un *plugin*)
    - Por ejemplo, un applet (`<applet>` está *deprecated* desde HTML 4.0)

```
<OBJECT
  classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
  width="200" height="200">
  <PARAM name="code" value="Applet1.class">
</OBJECT>
```
  - `<script>`
    - DHTML (Dynamic HTML): HTML donde se ejecuta un código de un lenguaje de script (como JavaScript)
    - Ejemplo:

```
<script type="text/javascript">
  alert("Hola Mundo");
</script>
```
    - Se puede combinar con eventos de navegación y formularios

```
<p onclick="alert('Hola de nuevo')">Un texto.</p>
```

# Lenguajes de programación en el cliente



**W<sup>3</sup>Techs**

Web Technology Surveys

<http://w3techs.com>

	2010	2011	2012
None	11,00%	8,60%	7,40%
JavaScript	88,40%	91,10%	92,40%
Flash	28,50%	25,60%	21,10%
Silverlight	0,30%	0,30%	0,20%
Java	0,10%	0,20%	0,20%

## Tecnologías de script

### ■ JavaScript

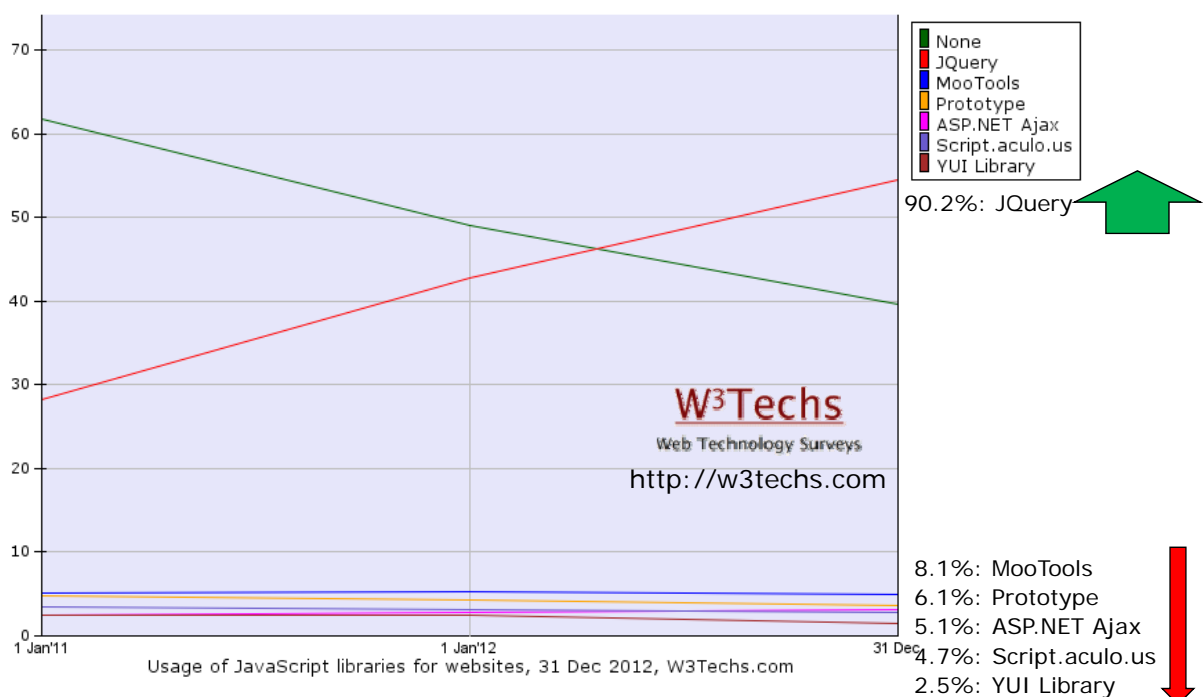
- Lenguaje de script cuyo código se puede insertar en páginas HTML
  - Lenguaje de programación interpretado (no compilado)
    - Las instrucciones las analiza y procesa el navegador cuando deben ser ejecutadas
  - Embebido en el código HTML
    - Etiquetas `<script type="text/javascript">` código JavaScript `</script>`
  - Orientado a objetos basado en prototipos (clonación de objetos)
  - Débilmente tipado (no requiere declarar el tipo de las variables)
  - Dinámico (permite cambiar la definición de miembros (atributos, funciones) de los objetos en tiempo de ejecución)
- ECMAScript (ECMA-262, 1992): versión estandarizada por ECMA (*European Computer Manufacturer's Association*)
- Permite al navegador alterar los contenidos presentes en una página Web
  - Agregar o eliminar contenidos
  - Modificar estilos visuales
  - Modificar textos del contenido

## Tecnologías de script

### ■ **AJAX** (*Asynchronous JavaScript And XML*)

- Creación de aplicaciones Web interactivas
- Las aplicaciones se ejecutan en el cliente
  - Pueden comunicarse asíncronamente con el servidor
  - El contenido de las páginas se actualiza sin necesidad de volver a cargarlas => Mayor dinamismo e interactividad
- Comprende varias tecnologías:
  - **XHTML** y **CSS**: Presentación basada en estándares
  - **DOM**: Interacción y manipulación dinámica de la presentación
  - **XML**, **XSLT** y **JSON**: Intercambio y manipulación de información
  - **XMLHttpRequest**: Intercambio asíncrono de información
  - **JavaScript**: Unión del resto de tecnologías
- En AJAX el cliente hace una petición al servidor por medio del objeto XMLHttpRequest
  - El servidor procesa la petición y devuelve una respuesta en XML en lugar de una página (X)HTML
  - El propio objeto XMLHttpRequest procesa dicha respuesta y actualiza únicamente las secciones necesarias de la página, evitando tener que recargarla por completo

## Librerías JavaScript





## Ejercicios – Tecnologías en el cliente

---

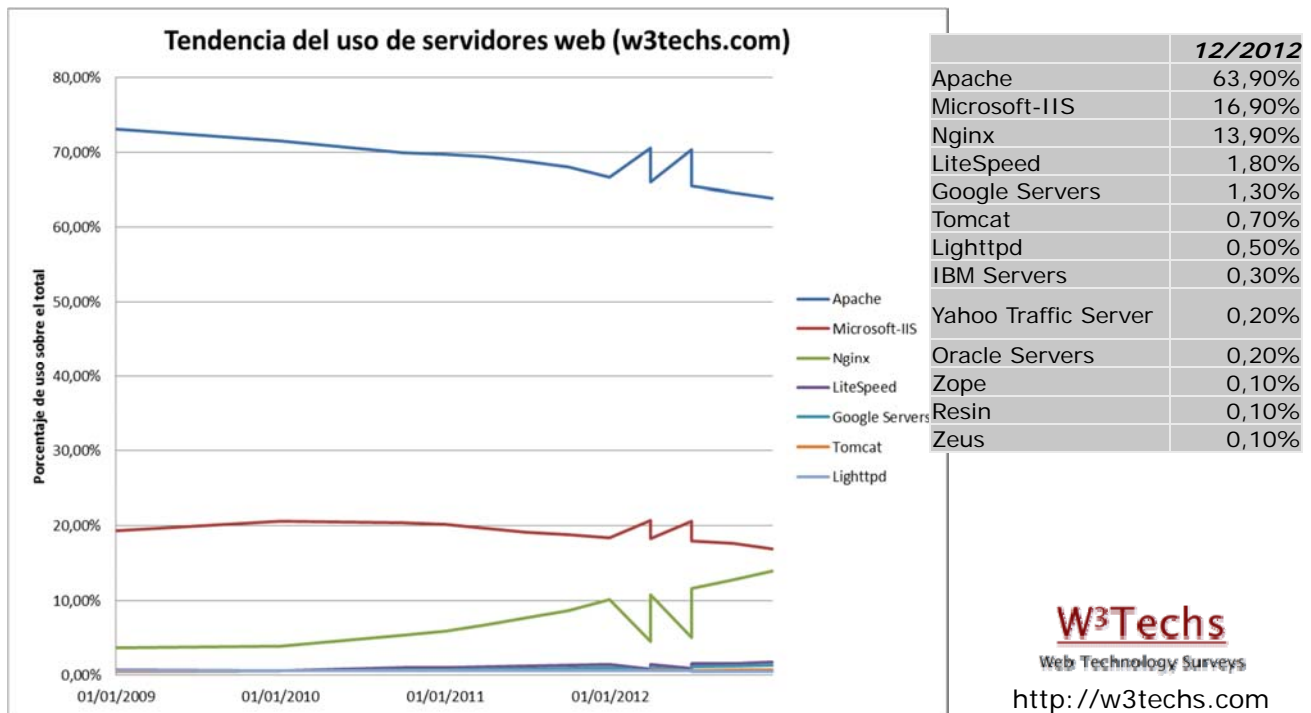
- Considera los diez sitios web que más utilices e identifica las tecnologías que usan en el lado del cliente
  - Puedes utilizar herramientas para el desarrollador del navegador
    - Firefox: Web Developer
    - Chrome: Herramientas->Herramientas para desarrolladores
    - MS Internet Explorer: Herramientas de desarrollo
  - Utiliza el foro de la asignatura para comentar con tus compañeros tus descubrimientos

## Servidores Web

---

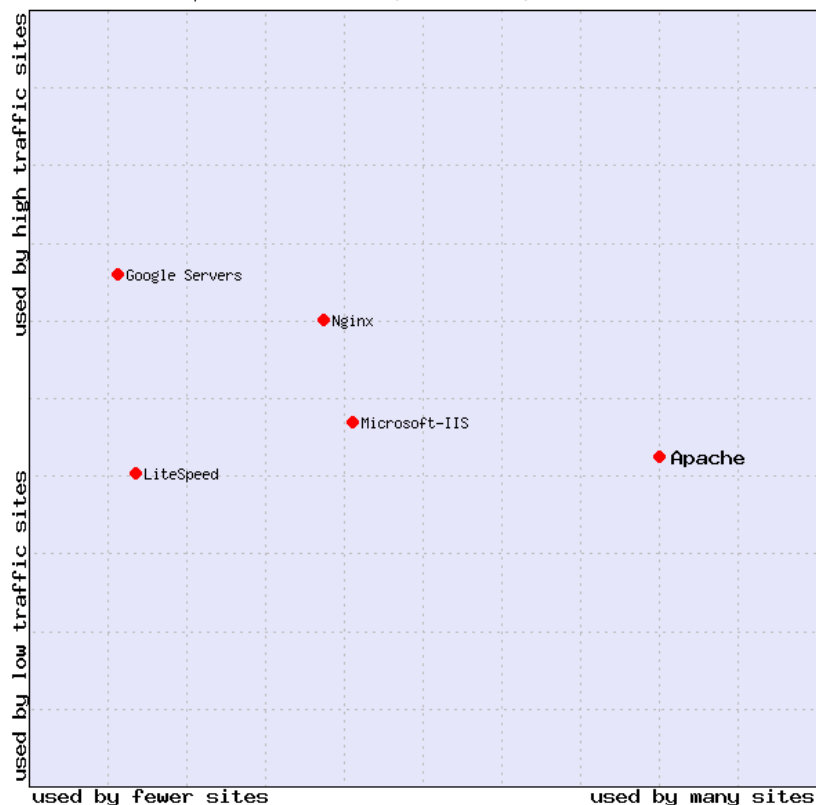
- Procesan mensajes HTTP de clientes y devuelven mensajes con la información solicitada
  - Estados
  - Datos
  - Códigos de error
- Todas las operaciones pueden adjuntar objetos o recursos Web descritos por su URL
  - Documentos HTML
  - Ficheros multimedia
  - Aplicaciones CGI
- Ejemplos de servidores Web
  - Apache (apache.org)
    - El más estándar en linux (también funciona en windows)
  - Internet Information Server (IIS)
    - Solo para windows, basado en la tecnología .NET
  - Nginx (nginx.org)
    - Muy ligero y escalable, aunque menos versátil que Apache

## Servidores Web



## Servidores Web

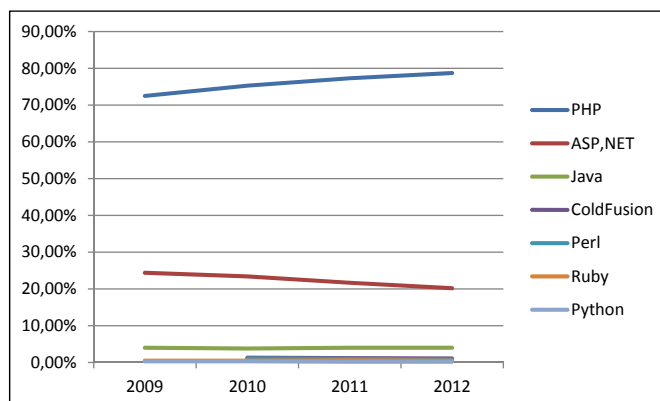
Apache Market Position, 27 Dec 2012, W3Techs.com



## Generación dinámica de información en el servidor

- **CGI** (*Common Gateway Interface*)
  - Una de las primeras formas de crear contenido dinámico
  - Estándar que define un mecanismo de comunicación para que un cliente solicite información a un servidor web
    - El servidor pasa la solicitud a un programa externo y la salida que éste genere (un objeto MIME) se manda como respuesta al cliente
    - Las aplicaciones que se ejecutan en el servidor se denominan CGI's
  - El programa CGI se arranca, se ejecuta, devuelve el resultado y acaba
    - Esto es poco eficiente
- **Código incrustado en HTML**
  - El servidor reconoce ciertas etiquetas y ejecuta el código que contienen
  - El programa tiene acceso a componentes del servidor
  - Lenguajes habituales:
    - **PHP**
    - **ASP** (Active Server Pages)
    - **JSP** (Java Server Pages)
- **J2EE** (Java Enterprise Edition): servlets

## Lenguajes de programación en el servidor



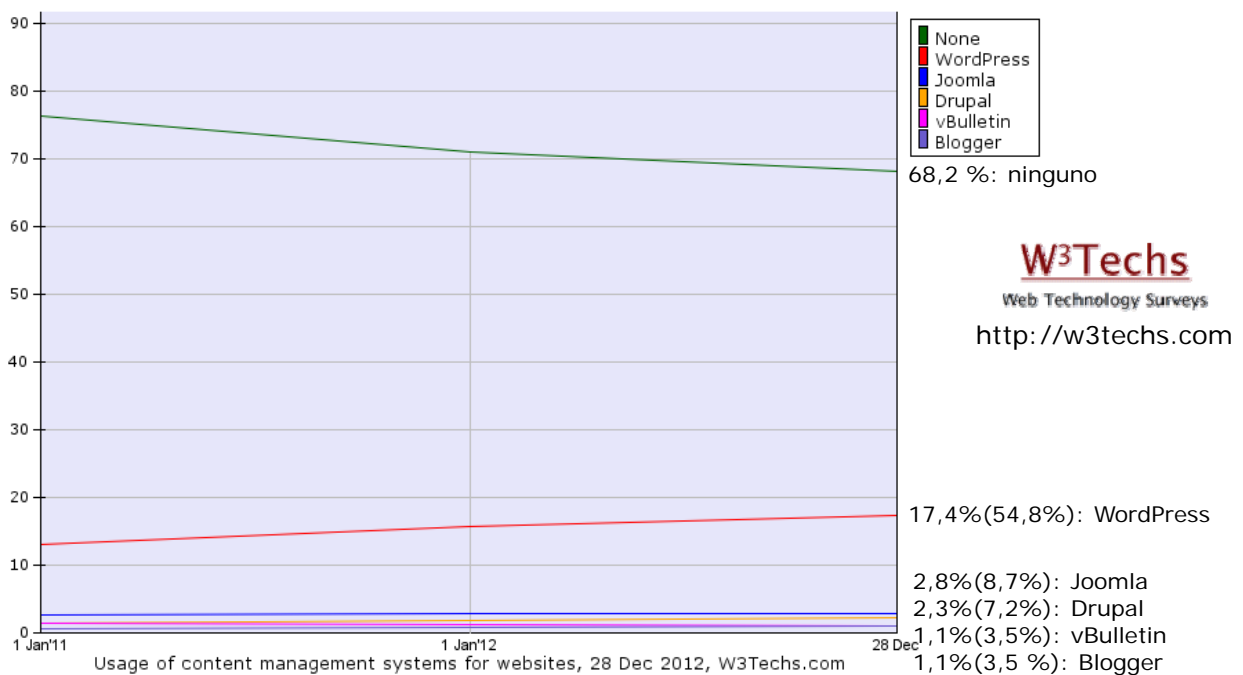
**W<sup>3</sup>Techs**  
Web Technology Surveys  
<http://w3techs.com>

	2009	2010	2011	2012
PHP	72,50%	75,30%	77,30%	78,70%
ASP,NET	24,40%	23,40%	21,70%	20,20%
Java	4,00%	3,80%	4,00%	4,00%
ColdFusion		1,30%	1,20%	1,10%
Perl		1,10%	1,00%	0,80%
Ruby	0,50%	0,50%	0,60%	0,50%
Python	0,30%	0,30%	0,20%	0,20%

## Gestores de contenidos

- *Content Management Systems (CMS)*
  - Frameworks para la creación y administración de contenidos de sitios web
  - Permiten la edición de los contenidos por varias personas con distintos roles (administrador, editor, participante, etc.)
    - Facilita el control de los contenidos en un sitio colaborativo
  - Separa la presentación de la información de su gestión en las bases de datos asociadas
    - Suelen ofrecer plantillas de presentación para facilitar la edición del sitio con formatos predefinidos
    - Facilita la actualización del sitio
  - Implementados generalmente con tecnologías estándar (PHP, MySQL, etc.)
- *Aplicaciones*
  - Blogs, Foros, Wikis
  - Plataformas de enseñanza on-line
  - Publicaciones digitales
  - Difusión de contenido multimedia

## Gestores de contenidos



## Ejercicios – Tecnologías en el servidor

---

- Identificar las tecnologías más **utilizadas** en el servidor y anotarlas en el campus virtual
- Identificar las tecnologías más **recientes** en el servidor y anotarlas en el campus virtual
  - Indicar referencias a las mismas

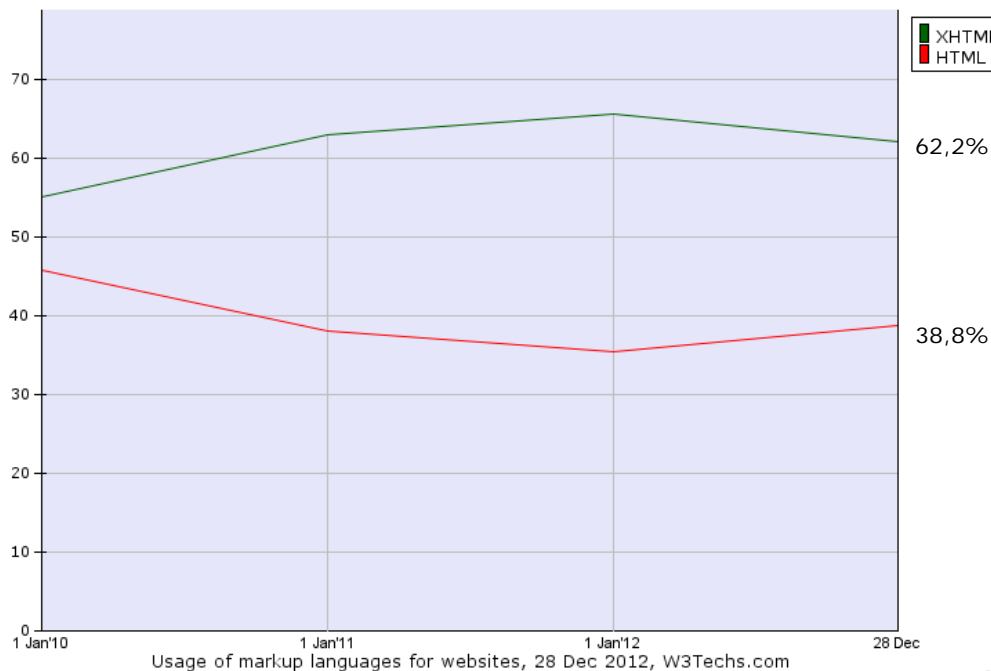
*¿Cómo seleccionar qué tecnología utilizar?*

## Estándares de la Web

---

- Los estándares de internet los define el IETF (*Internet Engineering Task Force*)
  - <http://www.ietf.org/>
  - Utiliza el mecanismo de *Request for Comments* (RFC)
    - Documentos que contienen las descripciones técnicas necesarias para el funcionamiento de todos los servicios de Internet
    - También hay RFC de carácter simplemente informativo
    - Ejemplo: <http://www.arrakis.es/~pjleon/rfc-es/rfc/rfc2223-es.txt>
- Desde 1995 el W3C (*World Wide Web Consortium*) define los estándares de la Web
  - <http://www.w3.org>
    - <http://www.w3c.es/> sitio español
  - Más de 400 organizaciones

## Uso de HTML/XHTML



**W<sup>3</sup>Techs**  
Web Technology Surveys  
<http://w3techs.com>

## XML

- **XML** (eXtensible Markup Language )
  - Especificación de carácter genérico derivada del Estándar SGML (*Standard Generalized Markup Language*) que permite definir lenguajes de marcado
  - Es un metalenguaje: sirve para definir lenguajes de marcado
  - Y para **describir, almacenar e intercambiar datos** en la Web
    - Estándar para el intercambio de información estructurada entre diferentes aplicaciones y plataformas de un modo sencillo, seguro y fiable
    - Permite guardar la información en un formato independiente del documento final que recibe el usuario
  - Es extensible a través de la definición de nuevas etiquetas
  - Los documentos XML pueden ser validados contra una **DTD** (Definición de Tipo de Documento)
  - El análisis de un documento XML es un proceso estandarizado, lo que permite utilizar cualquier analizador, evitando de este modo errores y optimizando el desarrollo de aplicaciones

## XML – lenguajes derivados

---

- **XSL** (EXtensible Stylesheet Language)
  - Familia de lenguajes para definición de presentación o formato de documentos XML:
    - **XSLT**: Para transformar la información en el formato final más apropiado para el usuario
    - **XSL-FO**: Permite describir la forma en que se presentan los componentes de un documento XML
    - **XPath**: Permite identificar de forma inequívoca cualquier elemento o atributo de un documento XML
- **Xlink** (XML Linking Language)
  - Permite definir de forma estándar hipervínculos en archivos XML
- **XPointer y Xfragments**
  - Lenguajes para apuntar a partes de un archivo XML
- **Xquery**
  - Lenguaje de consulta similar a SQL para colecciones de datos XML
- **XSchema**
  - Lenguaje de esquema empleado para describir la estructura y contenido adecuados de los elementos incluidos en documentos XML

## XML – lenguajes derivados

---

- **CSS** (Cascading Style Sheets)
  - Lenguaje de hojas de estilos que permite controlar la presentación de documentos (X)HTML y XML
- **Xforms**
  - Lenguaje de etiquetado de formularios Web
  - Permite comprobar automáticamente los datos introducidos por el usuario
  - Envía los datos de los formularios como XML
- **MathML**
  - Visualización de ecuaciones matemáticas
- **VoiceXML**
  - Lenguaje de hojas de estilos que permite controlar la presentación de documentos (X)HTML y XML
- **ebXML**
  - Negocios electrónicos (e-business)

## XML – Estándares de seguridad

---

Garantizar la integridad, confidencialidad y autenticidad de los datos que fluyen por la Web

- **XML-Encryption**

- Lenguaje de cifrado de documentos para asegurar su confidencialidad
- Permite cifrar partes de documentos (XML o no)

- **XML-Signature**

- Firma digital

- **XKMS** (XML Key Management Specification)

- Protocolo XML para distribuir y registrar claves públicas
- Protocolo XML para verificar claves públicas y certificados

- **XACML** (eXtensible Access Control Markup Language)

- Define esquemas y espacios de nombres para especificar políticas de autorización y control de acceso a objetos basados en XML

- **XrML** (eXtensible rights Markup Language)

- Define condiciones y derechos de acceso a recursos digitales

- **SAML** (Security Assertion Markup Language)

- Intercambio de información de autorización y autenticación

## JSON

---

- **JavaScript Object Notation**

- Lenguaje textual (legible por humanos) para intercambio de datos
- Descrito en RFC 4627 (2006)
- Se suele utilizar en aplicaciones AJAX, como alternativa a XML
  - JSON es más simple y legible que XML, pero no es extensible
- Se basa en JavaScript para representar estructuras de datos basadas en varios tipos básicos
  - Number (float de doble precisión de JavaScript)
  - String (secuencia de caracteres Unicode entre comillas)
  - Boolean (*true* o *false*)
  - Array (secuencia ordenada de valores de cualquier tipo separados por comas, entre corchetes)
  - Object (colección no ordenada de pares clave:valor or key:value, separados por comas, entre llaves)
  - null
- Es independiente del lenguaje (hay parsers para muchos lenguajes)

- **JSON Schema**

- Define la estructura de datos JSON (lo que XSD para XML)



## Tecnologías multimedia

---

- **PNG** (*Portable Network Graphics*)
  - Estandarizado por W3C (1996) y por ISO ((ISO/IEC 15948:2003)
  - Mucho mejor que **GIF** (*Graphics Interchange Format*):
    - GIF está patentado por Unisys y necesita licencia para su empleo, PNG es libre: *PNG's Not GIF*
    - Mejor formato de compresión (5-25% mejor) sin perder calidad
    - Colores más ricos y precisos (16,7 millones vs. 256 en GIF)
    - Soporta canales alfa y hasta 256 niveles de transparencia (GIF solo 1)
    - Los archivos gráficos en formato PNG pueden ser indexados por los motores de búsqueda, debido a la inclusión de metainformación
    - PNG ofrece un modo de compresión progresivo (entrelazado de dos dimensiones) que facilita el reconocimiento de la imagen en el inicio de su descarga
  - No permite imágenes animadas (GIF sí)
    - Pero hay la variante animada MNG (*Multiple-image Network Graphics*)
  - **JPEG** (*Joint Photographic Experts Group*) permite utilizar ratios de compresión muy altos pero en detrimento de la calidad de la imagen

## Tecnologías multimedia

---

- **SVG** (*Scalable Vector Graphics*)
  - Descripción de gráficos vectoriales en dos dimensiones,
    - Con formato XML
  - Recomendación del W3C (2001)
    - Implementado en casi todos los navegadores actuales
  - Define tres tipos de objetos gráficos:
    - Formas gráficas vectoriales (líneas, curvas, áreas)
    - Texto
    - Imágenes de mapa de bits/digitales
  - Permite crear gráficos dinámicos y animados
    - mediante ECMAScript o SMIL
    - Ejemplo: Juego de tetris  
[http://www.croczilla.com/bits\\_and\\_pieces/svg/samples/svgtetris/svgtetris.svg](http://www.croczilla.com/bits_and_pieces/svg/samples/svgtetris/svgtetris.svg)

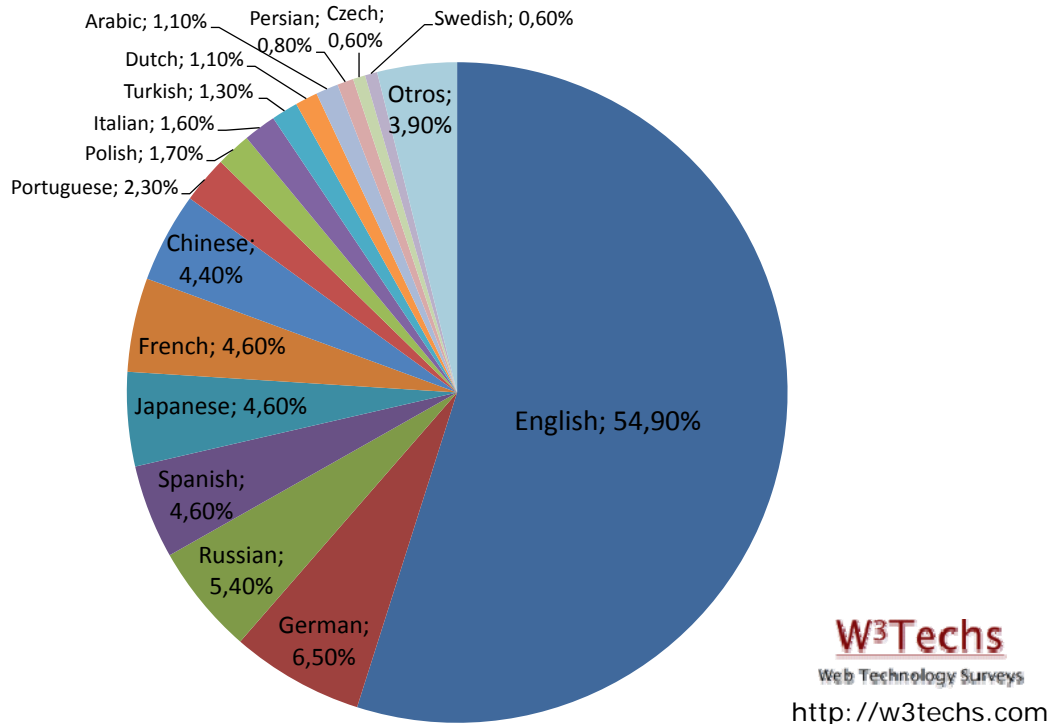
- **SMIL** (*Synchronized Multimedia Integration Language*)
  - Basado en XML. Actualmente en versión 3
  - Creación de presentaciones audiovisuales interactivas
    - Permite especificar la composición y secuencia de los eventos de una presentación multimedia
    - Integra audio y video con imágenes, texto y otros medios
    - Permite la interacción con el usuario
  - Se puede integrar con otros lenguajes de la familia XML
  - Permite incluir vínculos en presentaciones multimedia

## Internacionalización

---

- La Web funciona en todos los países y todos los idiomas
  - Los sitios Web deben diseñarse para adaptarse automáticamente a cualquier idioma y región sin necesidad de cambiar el código
- Unicode/ISO 10646
  - Estándar universal para codificar texto multi-lenguaje
  - Mantenido por UTC (*Unicode Technical Committee*)
  - Define tres formas de codificación: UTF-8, UTF-16 y UTF-32
- Para indicar el conjunto de caracteres que se utiliza
  - En XHTML, dentro de <HEAD>, con una etiqueta <META>:  
`<meta http-equiv="Content-Type" content="text/html; charset= utf-8" />`
  - En XML:  
`<?xml version="1.0" encoding="UTF-8" ?>`
- Para indicar el idioma en una parte del contenido de una página
  - En HTML: `lang="es"`
    - Se recomienda indicar el idioma del documento antes del <HEAD>
    - Y cada vez que se cambie de idioma a lo largo del texto de la página
  - En XML: `xml:lang="es"`

## Idiomas utilizados en la Web



## Accesibilidad

- **Web Accessibility Initiative (WAI)** del W3C
  - <http://www.w3.org/WAI/>
  - Guías y herramientas para facilitar la accesibilidad de los sitios Web
- **WCAG: Web Content Accessibility Guidelines 2.0** (W3C, 2008)
  - Define 14 pautas de soluciones de diseño accesible y que definen unos puntos de verificación que ayudan a detectar posibles errores, asignados a unos niveles de prioridad:
    - Prioridad 1: puntos que si no se cumplen, ciertos grupos de usuarios no podrían acceder a la información del sitio Web
      - Indicar siempre el cambio de idioma (`lang="en"`, `lang="fr"`, `lang="es"`)
      - Utilizar el atributo `alt` para incorporar texto equivalente a una imagen
    - Prioridad 2: puntos que si no se cumplen, sería muy difícil acceder a la información para ciertos grupos de usuarios
    - Prioridad 3: puntos que si no se cumplen, algunos usuarios experimentarían ciertas dificultades para acceder a la información
  - Niveles de conformidad
    - A: Satisface todos los puntos de verificación de prioridad 1
    - AA: ídem prioridad 1 y 2
    - AAA: ídem prioridad 1, 2 y 3

## 1 Perceivable

- 1.1 Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.
- 1.2 Provide alternatives for time-based media.
- 1.3 Create content that can be presented in different ways (for example simpler layout) without losing information or structure.
- 1.4 Make it easier for users to see and hear content including separating foreground from background.

## 2 Operable

- 2.1 Make all functionality available from a keyboard.
- 2.2 Provide users enough time to read and use content.
- 2.3 Do not design content in a way that is known to cause seizures.
- 2.4 Provide ways to help users navigate, find content, and determine where they are.

## 3 Understandable

- 3.1 Make text content readable and understandable.
- 3.2 Make Web pages appear and operate in predictable ways.
- 3.3 Help users avoid and correct mistakes.

## 4 Robust

- 4.1 Maximize compatibility with current and future user agents, including assistive technologies.

# Web 2.0

---

- Al principio los usuarios de la Web eran pasivos: solo podían recuperar información
- Web 2.0 es un término que define una forma de usar la web por usuarios y desarrolladores
  - La Web como medio de colaboración e interacción entre usuarios
  - Los usuarios como creadores de contenidos
  - Comunidades virtuales
  - Experiencia de usuario más rica
- Ejemplos:
  - Blogs: personal blogs, microblogging, corporate, etc.
  - Wikis
  - Compartición de videos y música
  - Redes sociales
  - Aplicaciones web: webmail, groupware, comercio electrónico, etc.
  - Servicios web: aplicaciones como composición de servicios Web
  - *Mashups*: aplicaciones que combinan y agregan datos de distintas fuentes para presentarlos de manera más conveniente: traducción automática, servicios de geolocalización, sistemas de pago electrónico, agregadores de noticias, etc.

## Web 2.0

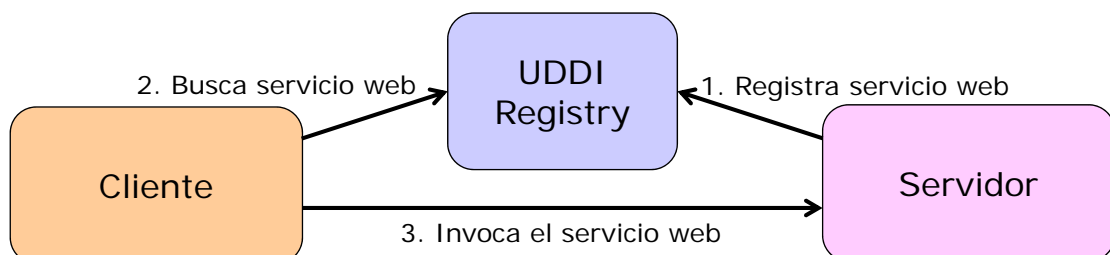
---

- Conceptos de la Web 2.0:
  - **RIA** (*Rich Internet application*): experiencia del escritorio en el navegador
    - Ajax y Flash
  - **WOA** (*Web-oriented architecture*): composición de aplicaciones Web
    - Servicios Web, mash-ups
  - **Web Social**: mayor interacción del usuario final que pasa a ser parte integral de la Web
    - Blogs, Wikis, RSS (*Rich Site Summary* o *Really Simple Syndication*)

## Servicios Web

---

- Una forma de implementar el modelo cliente-servidor en la web
  - Las aplicaciones definen los servicios que proporcionan con **WSDL** (*Web Service Description Language*, basado en XML)
  - Y los declaran en un registro **UDDI** (*Universal Description, Discovery and Integration*)
  - Los clientes pueden buscar localizar el servicio apropiado en el servidor UDDI
  - E invocar remotamente los servicios web
    - Todas las invocaciones se hacen con el protocolo **SOAP** (*Simple Object Access Protocol*), basado en XML



## Web semántica

---

- Movimiento colaborativo liderado por W3C con el propósito de que se incluya contenido semántico en las páginas Web
  - Para facilitar el procesado automático de la información en la Web
  - Una nueva Web con agentes inteligentes que traten toda la información por nosotros
  - La Web inicial era para ser visualizada por personas, no por máquinas
- **RDF** (*Resource Description Framework*)
  - Lenguaje para describir los recursos de la Web con meta-datos
  - RDF Schema: vocabulario para describir propiedades y clases de recursos basados en RDF
- **OWL** (*Web Ontology Language*)
  - Una ontología define un conjunto de términos en un área de conocimiento, y sus relaciones
- Para la sintaxis se puede utilizar **XML** aunque hay alternativas, como **Turtle** (*Terse RDF Triple Language*)
  - Turtle representa la información como triples (sujeto, predicado, objeto)  
`ex:person "Miguel de Cervantes" ex:relation "author" ex:book "La Galatea" .`

## Web móvil

---

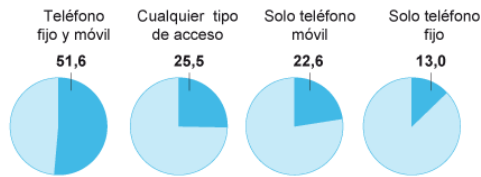
- Smartphones y tablets están cambiando el paradigma
  - Las aplicaciones en los dispositivos móviles hacen cada vez un uso más intensivo de servicios en la Web
  - A través de widgets y apps
    - Especialización en el acceso a los servicios
- Mobile Web Initiative (MWI)
  - <http://www.w3.org/Mobile/>

# Web móvil

## LOS 'SMARTPHONES' DISPARAN INTERNET MÓVIL

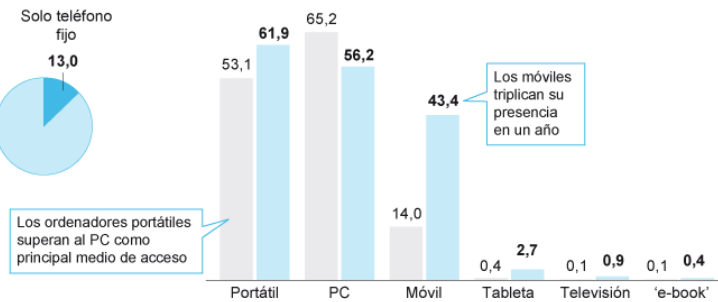
### USUARIOS CONECTADOS PERMANENTEMENTE

Porcentaje de internautas, según el tipo de acceso



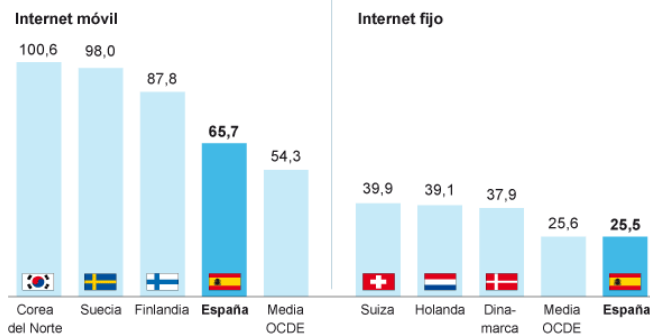
### TIPO DE DISPOSITIVO DE ACCESO A LA RED

En % 2011 2012



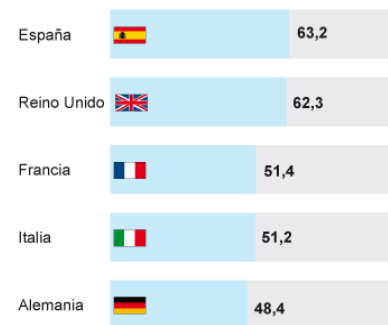
### PENETRACIÓN DE LA BANDA ANCHA EN EL MUNDO

Número de líneas por cada 100 habitantes



### USUARIOS DE 'SMARTPHONES'

En % del total de teléfonos móviles

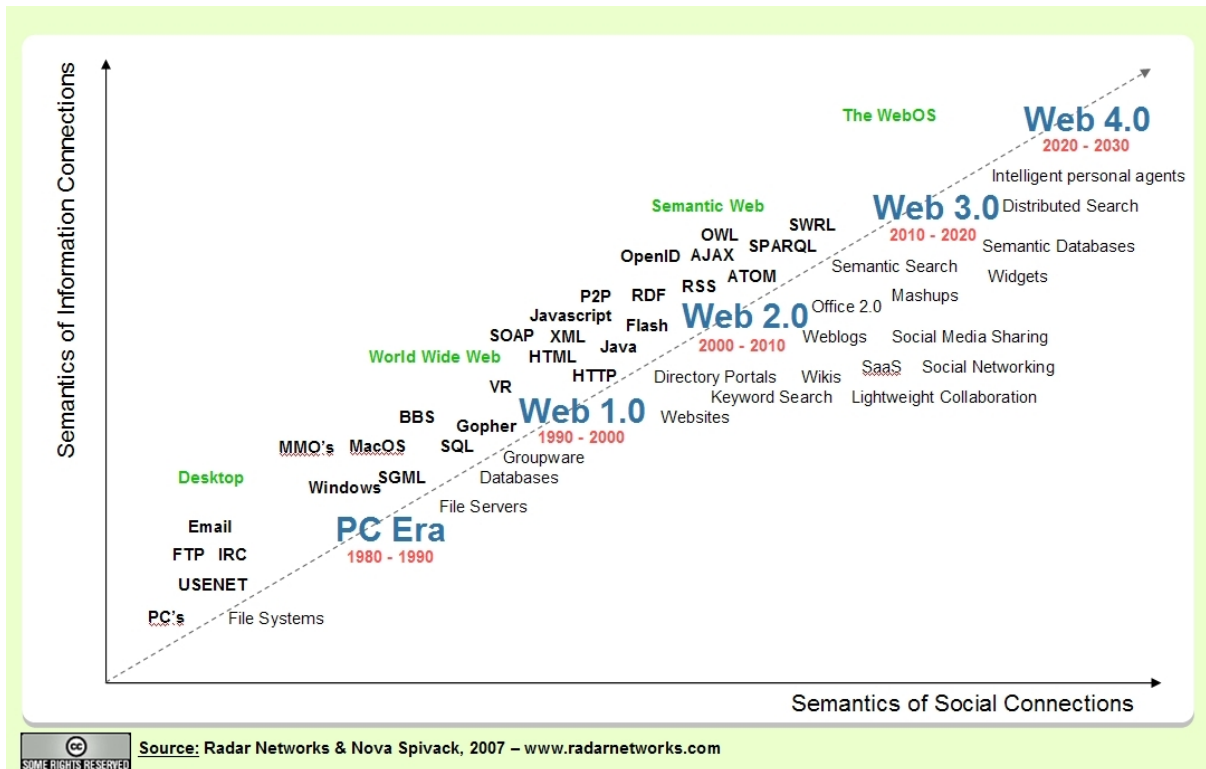


El País 10 ENE 2013, Fuente: 'La sociedad de la información en España 2012', Fundación Telefónica.

# Web ubicua

- Web ubicua y universal
  - Nuevas aplicaciones
    - Conectar una cámara de fotos a una impresora cercana
    - Usar el móvil para dar una presentación con un proyector inalámbrico
  - Descubrimiento y coordinación dinámica de los recursos
    - UPnP (Universal Plug and Play)
    - Jini
    - WSD (Web services for devices)
- Inteligencia ambiental
- Internet de las cosas
- Agentes inteligentes

# Evolución de la Web



## Resumen: Pila tecnológica del W3C



<http://www.w3.org/2004/10/RecsFigure.png>



## Bibliografía

---

- Sobre este tema hay información abundante en internet, proponed en el campus virtual
- Algunos tutoriales interesantes:
  - James Marshall. *HTTP Made Really Easy. A Practical Guide to Writing Clients and Servers*. <http://www.jmarshall.com/easy/http/>
  - Adam Shostack. An Overview of SHTTP. <http://www.homeport.org/adam/shttp.html>
  - Cursos variados sobre temas relacionados con la web, con licencia *Creative Commons*: <http://www.mclibre.org/consultar.html>
  - Curso online de la Univ. Standford: <http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=WebApplications>
- Estándares
  - Definition of URL/URI syntax, RFC 2396. <http://www.cis.ohiostate.edu/htbin/rfc/rfc2396.html>
  - HTTP 1.0, RFC 1945. <http://www.cis.ohio-state.edu/htbin/rfc/rfc1945.html>
  - HTTP 1.1, RFC 2068. <http://www.cis.ohio-state.edu/htbin/rfc/rfc2068.html>
  - HTTP State Management Mechanism, RFC 2109. <http://www.cis.ohiostate.edu/htbin/rfc/rfc2109.html>

**Ejercicio: contribuir a crear una lista de estándares en el campus virtual**