

---

# LABORATORIO TECNOLOGÍAS AUDIOVISUALES EN LA WEB

*José María Cañas Plaza*

*jmplaza@gsyc.es*



*Grado Sistemas Audiovisuales y Multimedia  
curso 2018-2019*

---

# Introducción a las aplicaciones web

## Contenido

- Un poco de historia, hasta hoy
- Arquitectura de aplicaciones web
- HTTP, MIME
- HTML, HTML5, XML, XHTML
- Tecnologías en el lado del cliente
- Tecnologías en el lado del servidor
- Glosario

## Un poco de historia, hasta hoy

- ARPANET, militar
- (1983) IP de 32 bits
- TCP:puerto
- (1992) Tim Berners-Lee: HTTP, HTML, navegador, servidor
- (1995) Netscape, Internet Explorer. Guerra de navegadores
- Explosión comercial. Se puede hacer dinero
- Crisis de las .com. Google
- (2004) Firefox, FaceBook
- (2005) YouTube, GoogleEarth
- (2009) Dropbox

## W3C

*The World Wide Web Consortium (W3C) is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards. Led by Web inventor Tim Berners-Lee and CEO Jeffrey Jaffe, W3C's mission is to lead the Web to its full potential.*

- <https://www.w3.org>
- Desarrolla y aprueba estándares (protocolos, guías de referencia...)
- Neutralidad frente a intereses comerciales
- Web for all, Web on everything
- Web for rich interaction
- Web of data and services

## Internet

- Internet: millones de ordenadores conectados. Conjunto de redes heterogéneas conectadas entre sí mediante el protocolo TCP/IP
- Web = vasta colección de documentos en Internet enlazados a través de hiperenlaces
- Los hiperenlaces permiten a los usuarios acceder a documentos situados en otros servidores Web, sin preocuparse de su ubicación

## Arquitectura TCP/IP

- Capa de enlace: de más bajo nivel, representa el medio físico encargado de enviar en última instancia los 0 y 1 que componen cada mensaje. Diversas tecnologías: Ethernet, ATM...
- Capa de red (IP). Esquema de direcciones, encaminamiento de los mensajes...
- Capa de transporte (TCP): Envía acuses de recibo, reagrupa el mensaje en destino, vuelve a mandar los paquetes perdidos o defectuosos...  
No garantiza tiempos de transmisión
- Capa de aplicación: programas que hacen uso de los servicios proporcionados por las capas inferiores, HTTP (HyperText Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol)...

## Web hoy

The top half of the image shows a Google Maps interface for 'URJC FUENLABRADA'. It includes a sidebar with options like 'GUARDAR', 'LUGARES cercanos', and 'COMPARTIR'. Below the map is a Facebook profile for 'Fabrizio Ferri-Benedetti' with a blacked-out profile picture. The bottom half shows a Spotify user's feed with posts like 'Tomorrow Is A Long Time - Nick Drake' and 'Desarrollado - Ons Wytars Ensemble, Unicorn Ensemble'.

The top section is the homepage of 'EL PAÍS' (El Periódico Global) dated 2 ENERO 2016. It features a large headline about the PSOE and includes sections for INTERNACIONAL, OPINIÓN, ESPAÑA, ECONOMÍA, CIENCIA, TECNOLOGÍA, CULTURA, ESTILO, DEPORTES, and TELEVISIÓN. The middle section shows the 'deportes' section of the website with a headline about Saudi Arabia executing 46 prisoners. The bottom section is a promotional banner for Netflix, featuring a man in a red jacket listening to music with earphones. Text on the banner includes 'NETFLIX', 'Podrás ver series y películas cuando quieras, donde quieras.', and 'Planes desde 7,99 € al mes.'

## Ejemplos exitosos

- Aplicaciones de correo electrónico (WebMail): GMail
- Maps
- Periódicos
- Teles, Radios, Netflix, Youtube
- Comercio electrónico, Amazon, tiendas, carrito de la compra
- Entradas, billetes de avión
- FaceBook, Instagram, redes sociales
- Aplicaciones corporativas de la URJC: foros, notas...
- Bancos

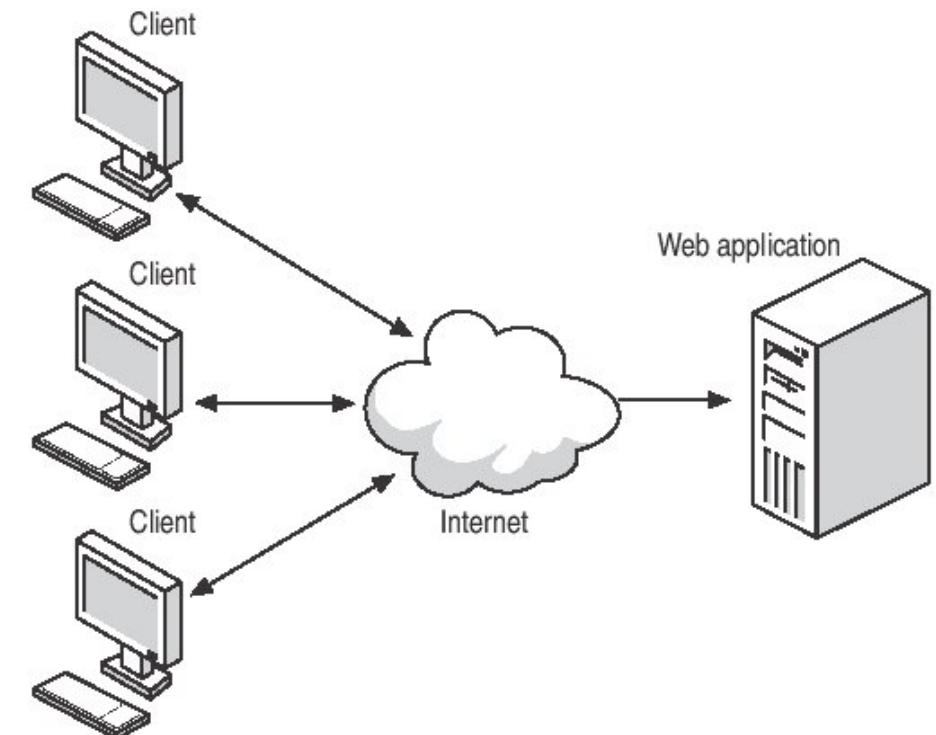
## Tendencias

- Capacidad de transmisión ha crecido
- De documentos (texto e imágenes) a aplicaciones (interacción)
- Multiplataforma, teléfonos móviles
- Aumento de rendimiento navegadores
- Multimedia, tiempo real
- Computación en la nube: AmazonWeb Services
- Almacenamiento en la nube

# Arquitectura de aplicaciones web

- World Wide Web
- Servidor, internet, navegador
- HTTP, HTML
- Puerto 80 por defecto
- Acceso a bases de datos
- URL  
protocolo:  
`lo://servidor:puerto/recurso`
- Información hipermedia: texto, imágenes, audio y video

proto-



- Quitar la magia, conocer las tripas para entender y crear
  - Máquinas conectan con otras máquinas
  - Software que hay que programar
- La web NO es internet
  - Se apoya en TCP/IP, como otras aplicaciones: correo electrónico, telnet, scp, ftp...
- Su uso está evolucionando. Al principio contenidos estáticos, actualmente contenidos dinámicos y mucho énfasis en los datos. Hoy día: software como servicio. Aplicaciones multimedia.
- Navegadores: Mosaic, Netscape, Internet Explorer, Firefox, Safari, Chrome, Opera
- El navegador se utiliza como interfaz de usuario para todas estas aplicaciones, y es estándard

- Servidores: Apache, Cherokee, lighttpd, Microsoft IIS
- Hay computación distribuida que NO es web
  - Objetos distribuidos: CORBA, ICE
  - RPC
- Eficiencia, compatibilidad entre plataformas, contenidos llamativos
- Seguridad: SSL. HTTPS
- Servicios Web no es lo mismo que Aplicaciones Web, nombre confuso.
- World Wide Web Consortium (W3C) publica estándares

## Funcionamiento básico

- El usuario solicita un documento tecleando su dirección en el navegador: <http://www.urjc.es>
- El cliente busca en el DNS cuál es la IP de www.urjc.es: 212.128.240.25
- Navegador y servidor web comienzan un diálogo a través del protocolo HTTP (protocolo de transferencia de hipertexto) GET /HTTP/1.0
- El servidor, si todo es correcto, devuelve el documento solicitado más información adicional
- El navegador mira el tipo de documento devuelto (MIME):
  - Si es “text/html” es un documento HTML, lo visualiza el propio navegador
  - Si es otro tipo de documento se ejecutará el programa que tenga asociado, o nos preguntará si queremos guardarlo

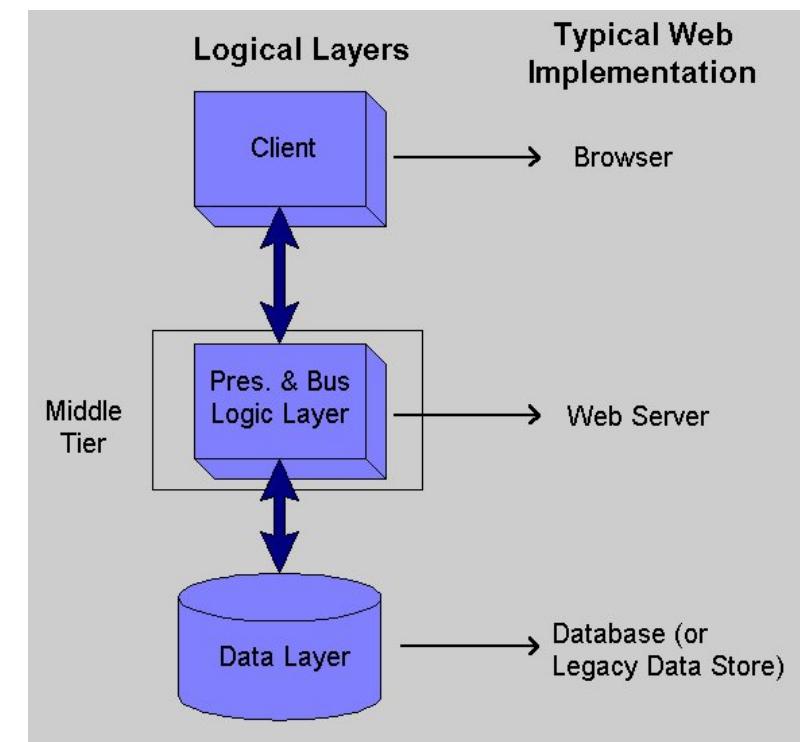


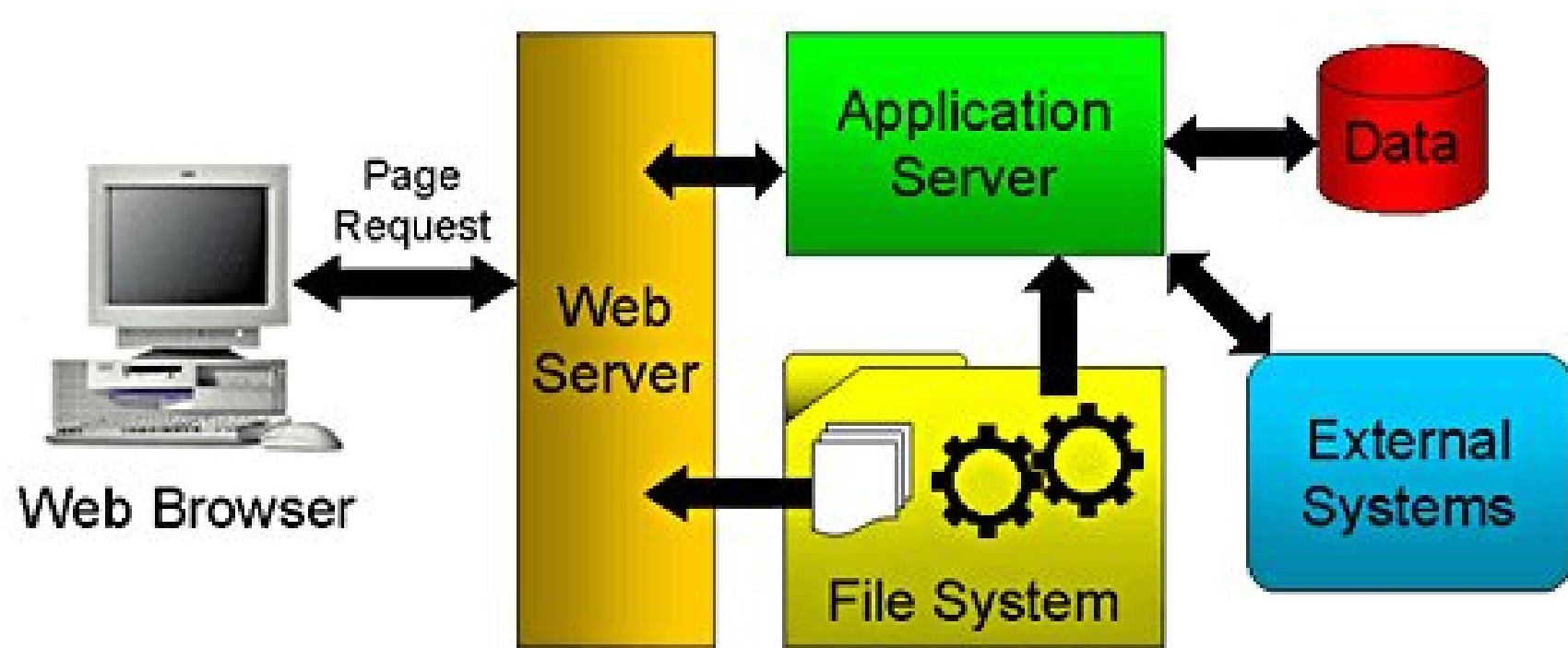
## Frente a aplicaciones tradicionales (de escritorio)

- (+) No hay que descargar o instalar nada en la máquina del cliente.  
No ocupa espacio en el disco duro del cliente
- (+) Actualización inmediata de la aplicación
- (+) Compatibilidad entre distintas plataformas, portabilidad
- (-) Menos funcionalidad que aplicaciones de escritorio
- (-) Disponibilidad depende de un tercero: la red
- Cliente, Servidor, permite poner cómputo en uno, en otro o ambos.

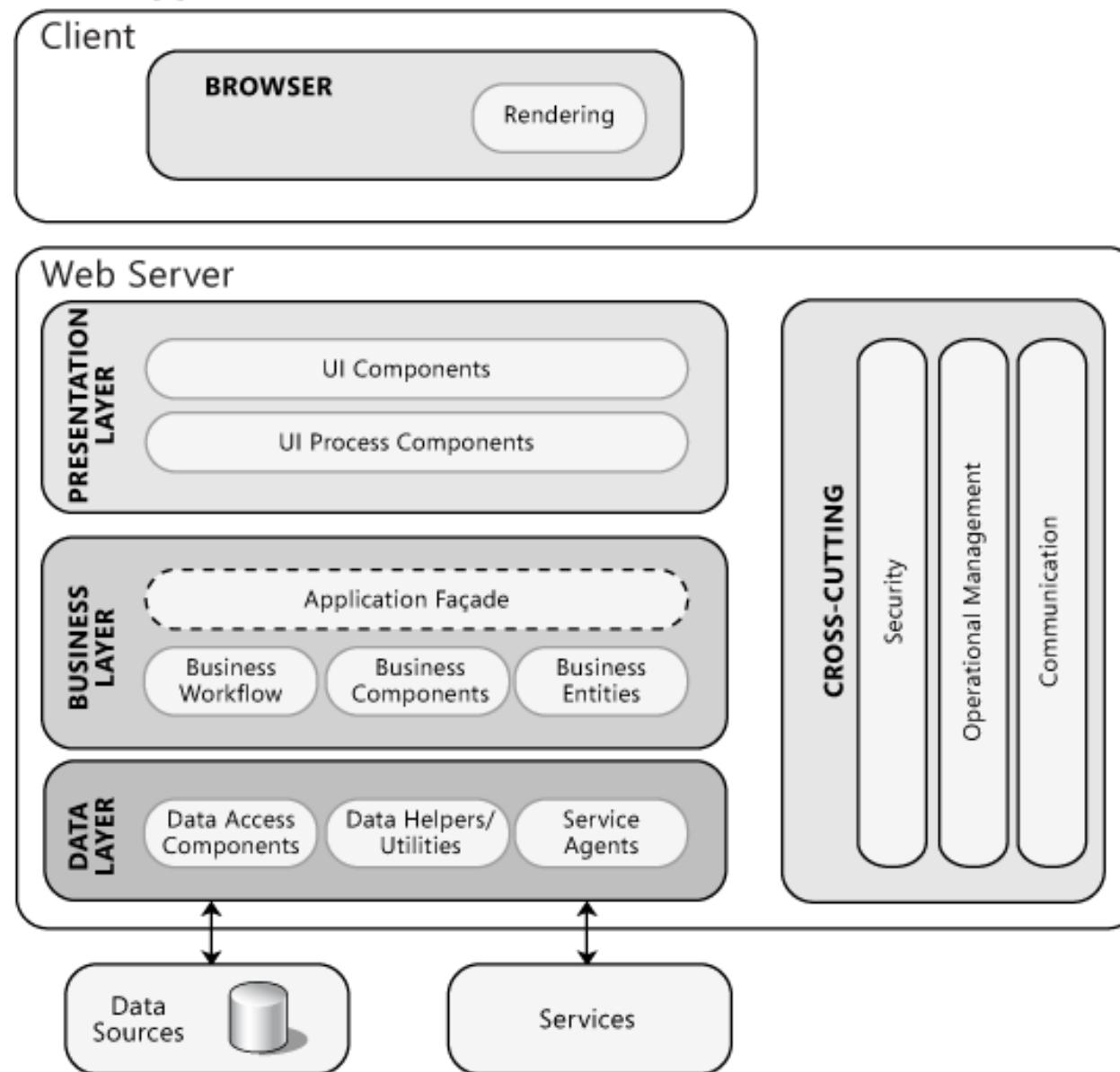
## Modelo de tres capas

- Capa de cliente
- Capa de presentación y negocio
- Capa de datos





## Web Application



## Servidor básico

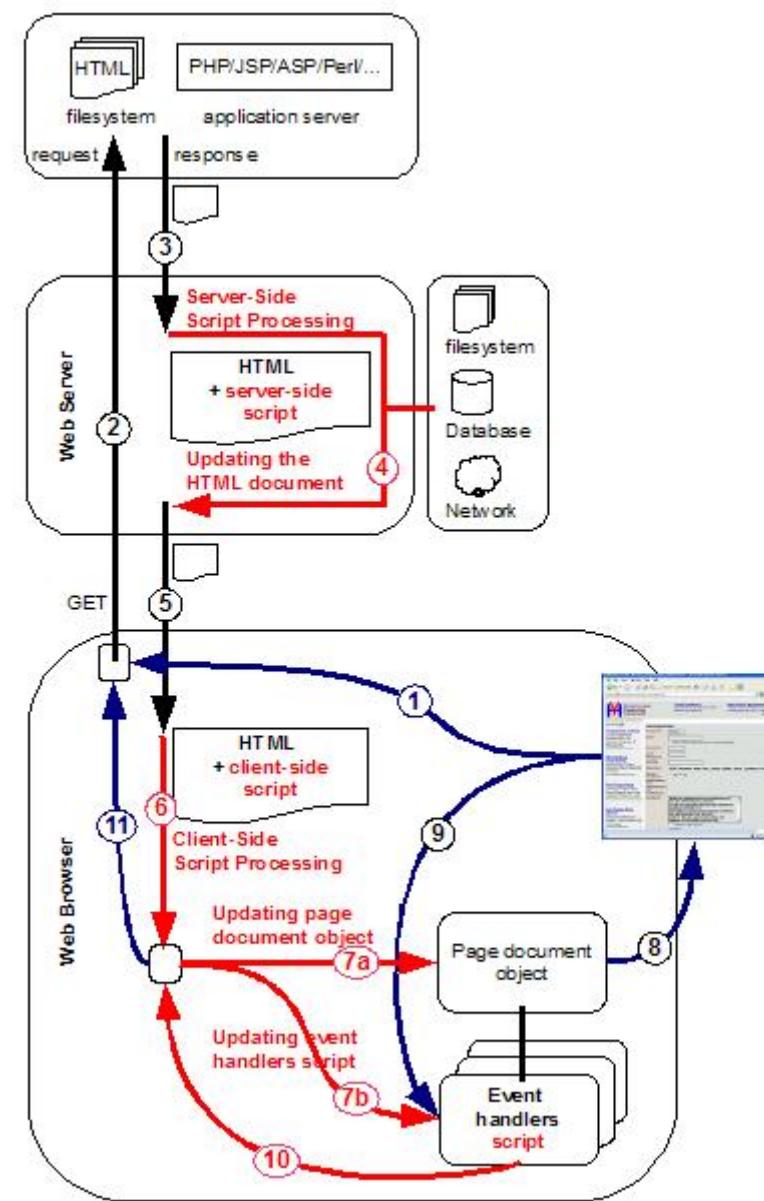
- Escuchar peticiones, puerto 80
- Atender petición, generar respuesta (lleva su tiempo)
- Enviar respuesta
- Volver a escuchar, bucle infinito
- Monohilo:
  - (-) atiende a pocas peticiones por segundo
  - (-) no aprovecha hardware paralelo

Una aplicación web puede ser utilizada por múltiples (miles) usuarios simultáneamente, lo que implica que puede suponer una carga importante de trabajo para el servidor a medida que aumentan el número de usuarios y de tareas demandadas.

- Servidores multihilo
- Servidores en varias (muchas) máquinas

## Contenidos estáticos vs dinámicos

- Objetivo histórico inicial: mostrar información guardada en ficheros estáticos.
- Contenidos dinámicos: creación de contenidos en tiempo de vuelo, atendiendo a interacción con usuario



# HTTP

- El protocolo HTTP (Hyper-Text Transfer Protocol) es un protocolo encargado de transmitir documentos a través de una conexión TCP/IP
- Anónimo por definición
- La capa externa de los servidores WEB es HTTP
- Es un protocolo sin estado
- Requiere que los datos sean transmitidos como un e-mail aunque los datos pueden no ser un e-mail propiamente dicho

## MIME

- son una serie de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario.
- muy usado en correo electrónico, SMTP
- Una parte importante del MIME está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos.
- Las extensiones de MIME van encaminadas a soportar:
  - Texto en conjuntos de caracteres distintos de US-ASCII
  - Adjuntos que no son de tipo texto
  - Cuerpos de mensajes con múltiples partes (multi-part)
  - Información de encabezados con conjuntos de caracteres distintos de ASCII

# HTML

- HTML (Hyper-Text markup Language) es un lenguaje de descripción de páginas derivado de otro mayor y más antiguo (SGML)
- HTML no es más que una serie de instrucciones que permiten especificar las características visuales de un documento
- No es un lenguaje de programación. No hay bucles, condicionales...
- Las páginas web son documentos.
- Incluyen texto y estilos, también código, imágenes, videos...
- Formularios como mecanismo de envío de datos al servidor

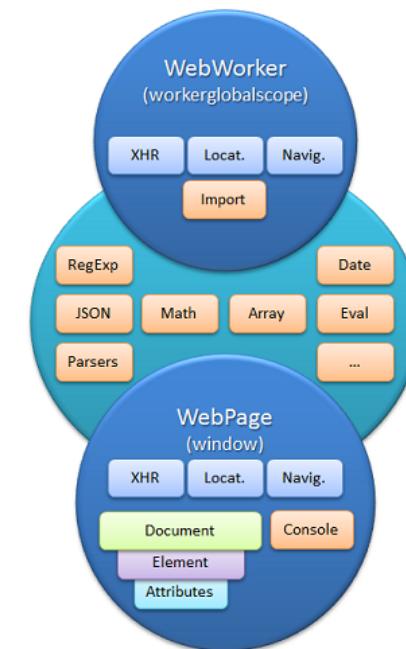
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Example</title>
5     <link rel="stylesheet" href="styl
6   </head>
7   <body>
8     <h1>
9       <a href="/">Header</a>
10    </h1>
11    <nav>
12      <a href="one/">One</a>
13      <a href="two/">Two</a>
14      <a href="three/">Three</a>
15    </nav>
```

- Se necesitan lenguajes de programación más potentes que permitan “mini” aplicaciones dentro de una web, como pueden ser:
  - Carro de la compra y comercio electrónico
  - Contador de visitas
  - Web Mail
  - Registro y entrada a un sistema
  - Transferencias y operaciones bancarias

## HTML5 y WebWorkers

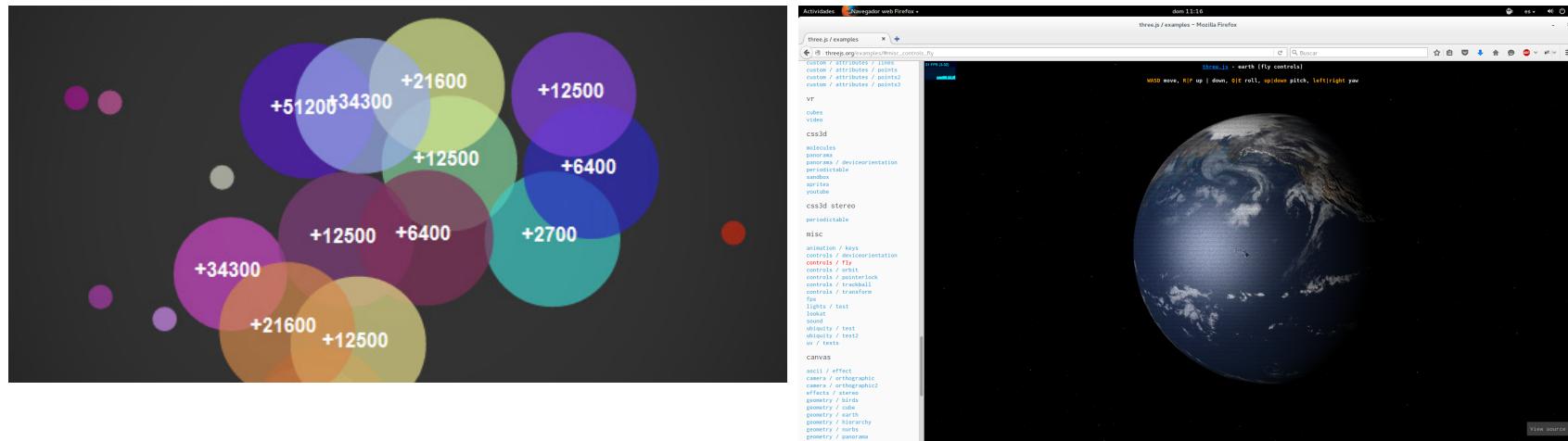
- **HTML5:** Establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos (Canvas, Video,...).
- **WebWorker:** Permite ejecutar scripts en segundo plano sin interferir con la interfaz de usuario.

```
<video src="..."></video>
```



## WebGL y Canvas en HTML5

- **WebGL:** Permite mostrar gráficos en 3D acelerados por hardware (GPU) en páginas web sin la necesidad de plug-ins.
- **Canvas:** Puede usarse para dibujar gráficos a través JavaScript. Por ejemplo, hacer composición de fotos, crear animaciones e incluso procesamiento de vídeo en tiempo real.



## XML

- Es un lenguaje (o Metalenguaje) de marcas creado para resolver las limitaciones de HTML.
- Los lenguajes de marcas se crearon ante la necesidad de utilizar un formato *estándar* para representar la información. HTML se definió para soportar un sistema de hipertexto distribuido, que fuera portable y fácil de aprender.
- HTML tiene un conjunto limitado de marcas. Resulta útil para formatear un documento pero no permite informar sobre la semántica de la información codificada en él, no indica lo que está representando, se preocupa principalmente de que algo se visualice con unas características determinadas.
- El lenguaje XML hace precisamente esto, describe el contenido de lo que etiqueta. La potencia de esta forma de trabajar radica en que se

está etiquetando e identificando el contenido, olvidando en un principio la forma de presentarlo.

- Tanto XML como HTML tienen su base en el SGML. El SGML (Standard Generalized Markup Language) es el estándar internacional para la definición de la estructura y el contenido de diferentes tipos de documentos electrónicos. Es un metalenguaje, un lenguaje utilizado para crear nuevos lenguajes de marcas.
- El problema es que SGML es complicado, difícil de aprender y bastante complejo. XML es una versión abreviada de SGML optimizada para su utilización en Internet.
- XHTML (eXtensible HyperText Markup Language) es básicamente HTML expresado como XML válido.

## Tecnologías en el lado del cliente

Son las páginas dinámicas que se procesan en el cliente. En estas páginas toda la carga de procesamiento de los efectos y funcionalidades la soporta el navegador. Usos típicos de las páginas de cliente son efectos especiales para webs como rollovers o control de ventanas, presentaciones en las que se pueden mover objetos por la página, control y validación de formularios, cálculos, etc.

- (+) Descargan de trabajo al servidor
- (+) Reducen el ancho de banda empleado en las comunicaciones, ya que el número de accesos entre cliente y servidor disminuyen.
- (+) Facilitan la validación interactiva de datos

- (+) Permiten utilizar recursos locales.
- (-) Seguridad, se ejecuta código que proviene de una fuente externa, hay que evitar que desencadenen acciones indeseadas.
- (-) Dependencia del cliente utilizado, el navegador tiene que estar preparado para interpretar dicho lenguaje. Puede suceder que cierta tecnología no funcione adecuadamente
- (-) Cuando se utiliza una Base de Datos en el servidor, el rendimiento disminuye ya que cada acceso a la misma supone ocupación en la red.
- El navegador ha ido creciendo en funcionalidad: mostrar texto e imágenes, lanzar aplicaciones externas, intérprete de lenguajes script, máquina virtual java...

- JavaScript
- AJAX
- VBScript
- Java Applet
- ActiveX
- Flash
- JQuery
- Flash

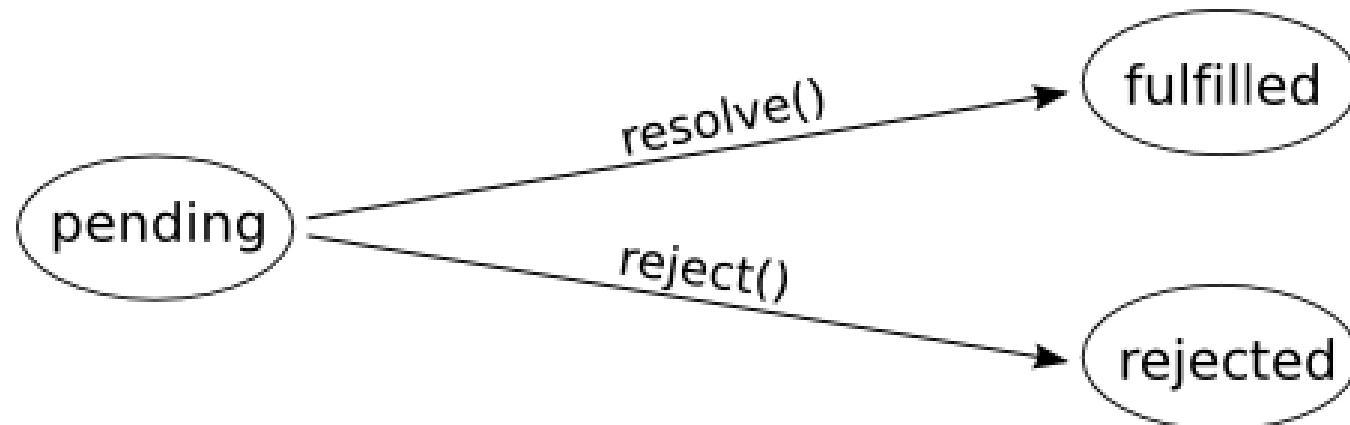


## JavaScript

- La utilización de lenguajes de guiones (como JavaScript), están muy extendidos actualmente. JavaScript es el estandard.
- Los scripts de una página web permiten la ejecución de código asociado a eventos, como pulsar un botón o seleccionar una casilla de verificación.
- Su aplicación más habitual es la validación de datos de un formulario antes de enviarlos al servidor (por ejemplo el chequeo de un NIF).
- Los lenguajes de guiones pueden actuar sobre el navegador a través de objetos integrados que representan al documento, a la ventana activa, a cada uno de los elementos de un formulario, etc..
- JavaScript (Netscape), Jscript (Microsoft), VBscript (IEexplorer), ECMAScript (basado en JavaScript).

## Javascript 6 y Promesas

- **JavaScript6 o ECMAScript 2015:** es la versión actual de la especificación del lenguaje ECMAScript conocida simplemente como “ES6”.
- **Promise:** Devuelve una promesa de tener un valor en algún momento en el futuro.

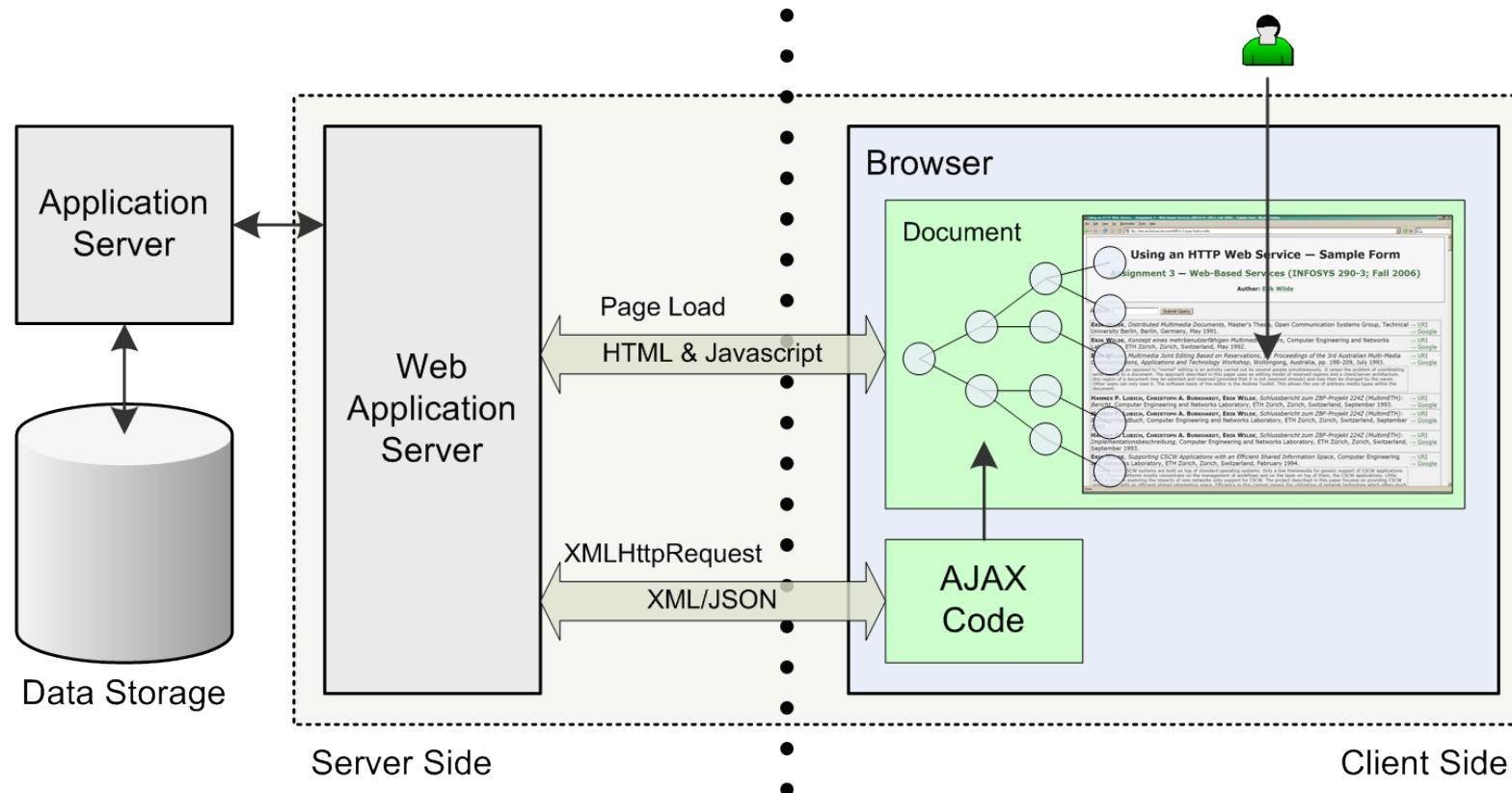


## JQuery y Bootstrap

- **JQuery**: es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos,...
- **Bootstrap**: conjunto de herramientas de software libre para diseño de sitios y aplicaciones web.

| JQuery   | Javascript  |
|--|---|
| <pre>\$(document).ready(function(){<br/>//code<br/>});</pre> | <pre>document.addEventListener("DOMContentLoaded",function(){<br/>//code<br/>});</pre>  |
| <pre>var divs = \$("div");</pre>                             | <pre>var divs = document.querySelectorAll ("div");</pre>  |
| <pre>var newDiv = \$("&lt;div&gt;");</pre>                   | <pre>var newDiv = document.createElement("div");</pre>  |
| <pre>\$(“a”).click (function({<br/>//code<br/>});</pre>      | <pre>[ ].forEach.call (document.querySelectorAll (“a”),function (el){<br/>el.addEventListener (“click”,function(){<br/>//code<br/>});<br/>});</pre> |





## Java Applet

- Una applet Java es una aplicación escrita en Java que se lanza desde una página web y es enviada por el servidor para su ejecución en el cliente.
- El encargado de la interpretación del código es el propio navegador que incorpora una máquina virtual.
- Un cliente solicita una página HTML a un servidor
- El servidor envía la página HTML sin el Applet
- El navegador descubre dentro del HTML la etiqueta <APPLET> y solicita al servidor que le envíe dicho APPLET
- El servidor envía el APPLET en forma de Código de bytes, que han de ser interpretados por el JVM (Java Virtual Machine) del cliente.
- La comunicación de applets Java con Bases de Datos se establece

mediante JDBC, que no es más que un mecanismo (API) que permite ejecutar instrucciones SQL sobre u base de datos y procesar los resultados obtenidos.

- (+) Portabilidad. Una applet se basa en la compilación de un código fuente (.Java) para generar un código intermedio (.class) que puede ser interpretado por una maquina virtual Java, disponible para la mayoría de plataformas existentes hoy en día.
- (+) Seguridad, que permite ejecutarlas en cualquier cliente de una forma segura, ya que se utiliza un módulo denominado "Security Manager" que comprueba de forma sistemática el acceso a los recursos y a las zonas de memoria utilizadas por la applet.
- (-) Requiere recursos considerables en el equipo cliente y su ejecución es más lenta debido a la interpretación del código intermedio.

## Tecnologías en el lado del servidor

- El código de la aplicación se ejecuta en el servidor y en el cliente sólo se recibe su salida en forma de páginas HTML, de este modo la compatibilidad esta garantizada.
- Este modelo es más adecuado cuando se utiliza una **Base de Datos** que ya reside en el servidor, ya que de esta manera se optimizan los accesos y se disminuye el ancho de banda utilizado.
- Son adecuadas cuando los procesos a ejecutar son operaciones críticas, que requieren unos chequeos de seguridad importantes, o cuando se necesita una potencia de cálculo que no se puede obtener en el cliente.
- Servidor ha ido creciendo en funcionalidad: entregar ficheros HTML, ejecutar aplicaciones externas, incorporar intérpretes de lenguajes script, máquina virtual de Java, servidor de aplicaciones...

- CGI
- PHP
- Servlets
- JSP
- ASP
- Servidores de aplicaciones
- Python, Django
- Ruby on Rails
- node.js

## CGI: Common Gateway Interface

- Antiguo. Creado inicialmente para gestionar formularios
- No es un lenguaje de programación.
- Es un mecanismo para que el servidor web pueda llamar a un programa externo.
- El programa CGI tiene acceso a cierta información sobre la conexión y a los valores del formulario
- El programa CGI puede estar desarrollado en cualquier lenguaje: C, perl, sh, python...
- Su salida es un HTML

- CGI es ineficiente: un proceso por cada petición, un intérprete por cada petición
- Para más eficiencia: Interfaces del propio navegador: ISAPI o NSAPI (pueden provocar caídas del servidor).
- Alternativa: empotrar el intérprete en el servidor web
- FastCGI, SCGI tratan de solucionar ineficiencia creando procesos en segundo plano

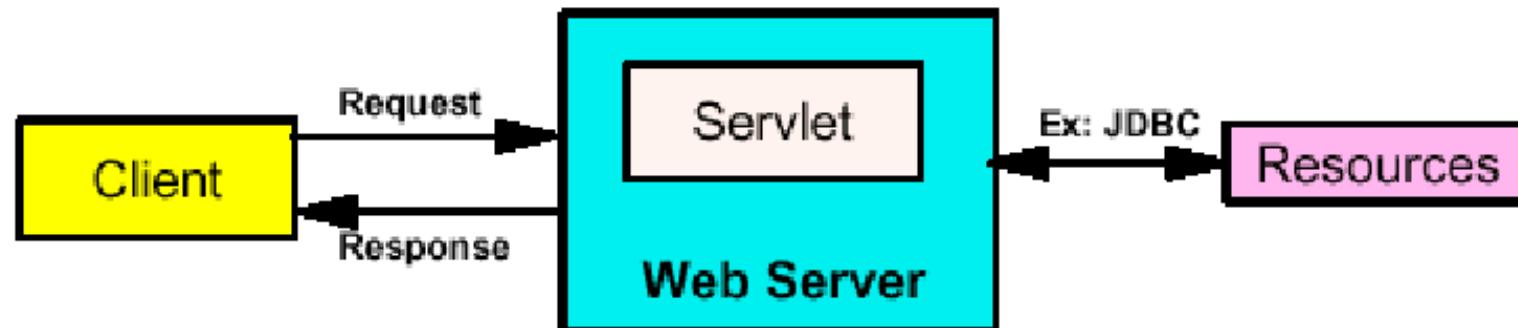
## PHP

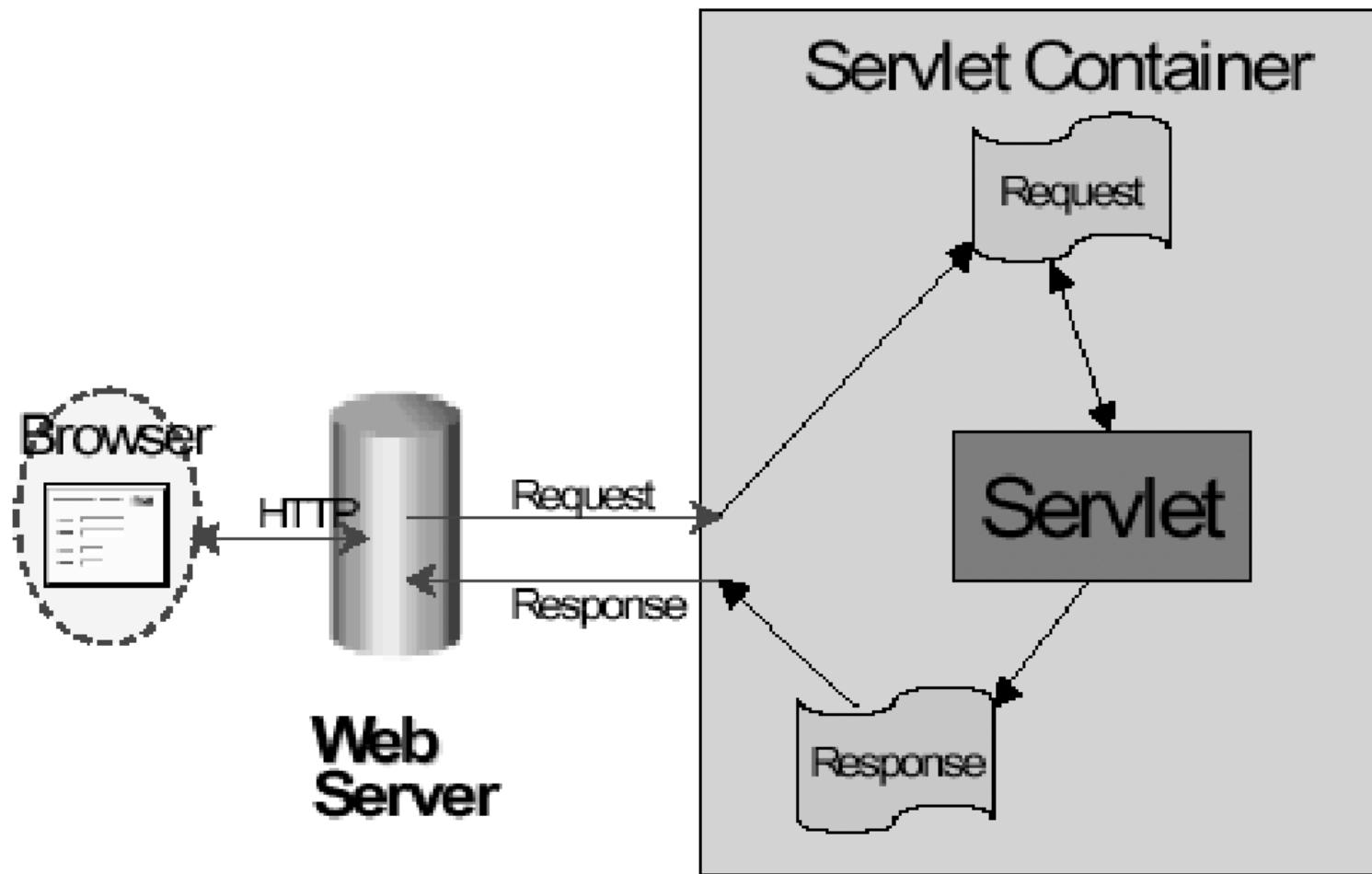
- Creado por Rasmus Lerdorf en 1994. Software libre.
- Lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.
- Código empotrado en HTML
- El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante (PHP interpreter)
- El cliente no ve el código PHP, sino el código HTML puro resultante de la ejecución
- Existen módulos para acceder fácilmente a bases de datos

## Servlets

- Son programas escritos en Java que corren en un servidor de aplicaciones con JVM
- Al ejecutarse en el servidor son aplicaciones sin GUI, base de hebras, no procesos.
- Similar al CGI, pero con los beneficios de Java.
- Capaces de mantener la conexión abierta y por lo tanto actualizar el contenido del navegador del cliente.

- El cliente hace una petición
- El servidor envía la información de la petición al Servlet
- El Servlet con esta información crea un contenido dinámico que entrega al servidor
- Este contenido es devuelto al cliente





## JSP: JavaServer Pages

- Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo
- Son documentos de texto que crean dinámicamente páginas web a partir de una petición de un cliente
- Un documento JSP contiene:
  - Plantillas de texto conteniendo formatos de presentación (HTML, XML)
  - Acciones dinámicas bien sea mediante código o mediante etiquetas JSP que permiten en acceso a directivas o a otros componentes
- El servidor web compila dinámicamente en un servlet la primera vez que es llamada.

## ASP: Active Server Pages

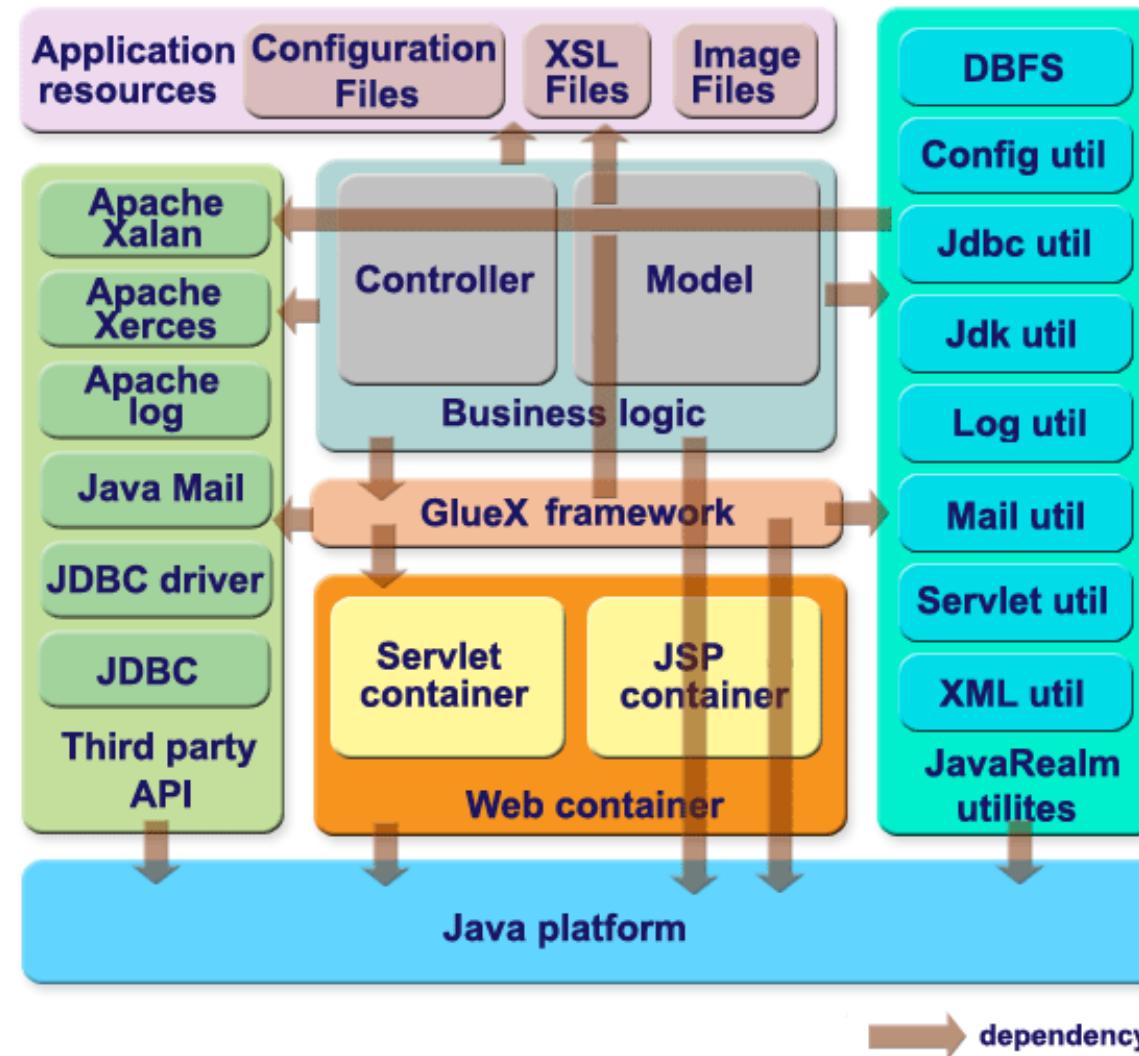
- Es código ejecutado en el servidor que devuelve HTML, por lo que queda garantizada su compatibilidad.
- ASP permite integrar componentes ActiveX (del lado del servidor), como acceso a base de datos, scripts..., lo que dota a la lógica de negocios de una potencia y flexibilidad enorme.
- ASP se presentó por primera vez con el IIS 3.0 de Microsoft.
- Una página ASP/JSP contiene HTML estático intercalado con scripts que se encargan de generar HTML de forma dinámica.
- (-) suele ser poco elegante

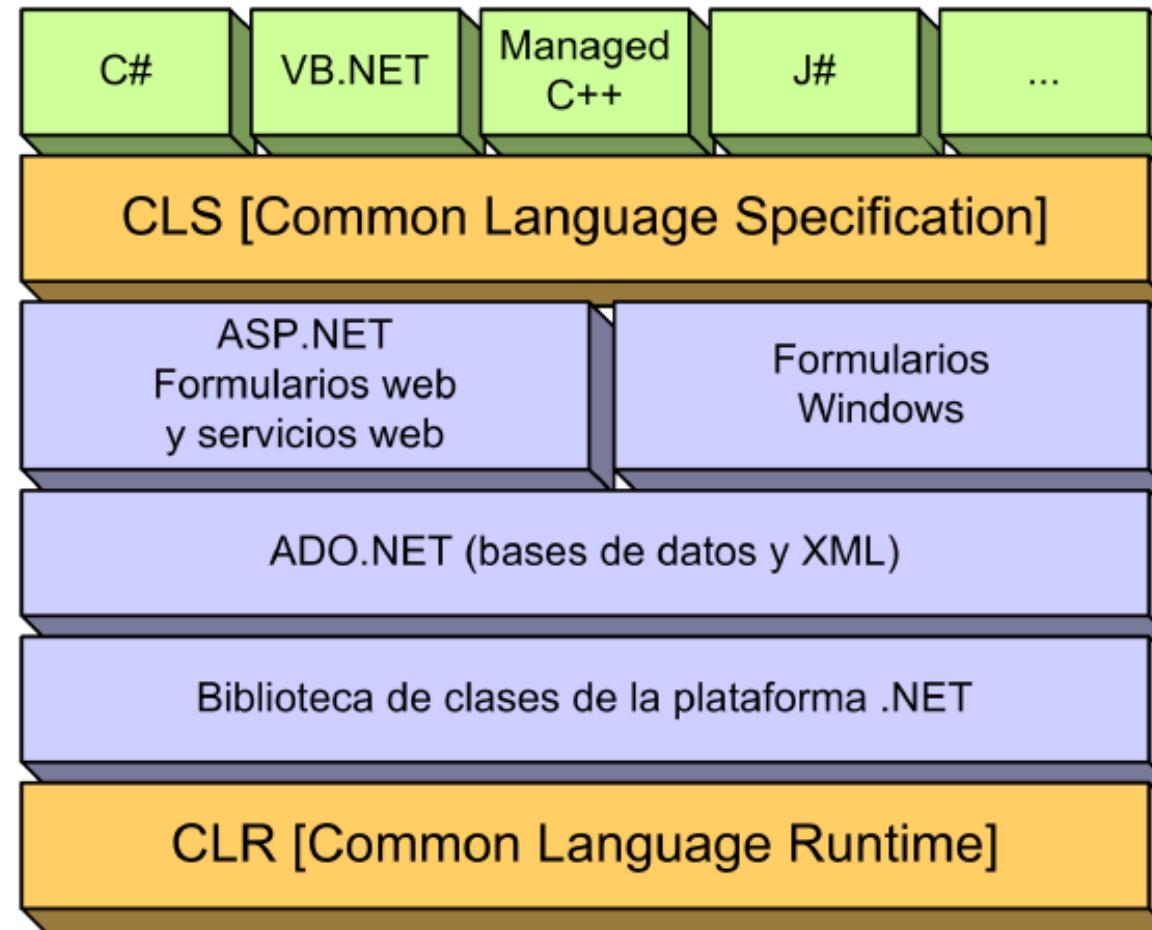
## Servidores de aplicaciones

Es un servidor web con capacidad de procesamiento, por lo que suele ser a la vez servidor web y servidor de lógica de negocio.

- Modulares
- Funcionalidad: seguridad, balanceo de carga, gestión de sistemas distribuidos, soporte a transacciones
- J2EE, entorno basado en Java
- .NET, entorno de Microsoft (incluye ASP.NET)

## Typical Web Application Component Architecture





## Python, Django

- Python: Lenguaje interpretado orientado a objetos
- Módulo Python en Apache
- Django: entorno Python para aplicaciones web en el lado del servidor
- Sigue el patrón Modelo-Vista-Controlador

# Glosario

CSS, Ajax, Flash, Html5, JavaScript, Servlet, Applet, Servicio Web, Servidor de aplicaciones, CGI, PHP, Perl, ASP, HTML, XML, Cookies, MIME, XHTML.

- CSS: Cascading Style Sheets
- CGI: Common Gateway Interface. Tecnologías del servidor
- SSL: Secure Socket Layer
- XML: eXtended Markup Language.
- XHTML (eXtensible HyperText Markup Language)
- URL: Uniform Resource Location
- JDBC: Java DataBase Connectivity.

- MIME: Multipurpose Internet Mail Extensions
- PHP: Hypertext Pre-processor
- J2EE: Java 2 platform Enterprise Edition
- IIS: Microsoft Internet Information Services web server
- HTML5
- WebRTC