
UD1. Introducción a las aplicaciones web

IAW - 2º ASIR

Objetivos de la unidad

- ★ Diferencia entre aplicación web y de escritorio.
- ★ Funcionamiento de las aplicaciones web.
- ★ Generaciones de aplicaciones web.
- ★ Arquitectura de una aplicación web.
- ★ Tecnologías usadas en las aplicaciones web del lado del cliente y del servidor.

Índice de contenidos

1. Conceptos generales en aplicaciones web.
2. Generaciones de aplicaciones web.
3. Funcionamiento de las aplicaciones web.
4. Creación de aplicaciones web.

1. Conceptos generales en aplicaciones web

¿Qué es una aplicación web?

- Se trata de una aplicación creada para ser ejecutada en un navegador.
- Normalmente una aplicación web es menos potente que una aplicación de escritorio al no poder optimizar su código para la máquina que se está ejecutando.
- Depende de la habilidad y potencia del navegador.
- Las aplicaciones web se suelen crear en HTML y sus tecnologías asociadas como CSS, Javascript, PHP, etc...



1. Conceptos generales en aplicaciones web

Ventajas de las aplicación web

- ✓ Gran compatibilidad.
- ✓ Requerimientos “mínimos” en el cliente.
- ✓ Fáciles de manejar por el usuario.
- ✓ Facilidad de mantenimiento.
- ✓ Datos centralizados.
- ✓ No es necesario instalarla en el dispositivo.
- ✓ Muy accesibles.
- ✓ Se permite el uso de pc o clientes ligeros para su utilización.



1. Conceptos generales en aplicaciones web

Desventajas de una aplicación web

- ✗ Menos potentes que las aplicaciones de escritorio.
- ✗ Aprovechan peor el hardware.
- ✗ Se requiere conectividad.
- ✗ Las aplicaciones web son más difíciles de implementar y depurar.
- ✗ Delegación del control de nuestra información.



2. Generaciones de aplicaciones web

- La **Web 1.0** es la original, el principio, el primer contacto que existen con las páginas web. En ellas básicamente nos limitamos a **consumir contenido sin más** actualización o interacción.
- La **Web 2.0** fue la primera gran evolución. La conocida **web social** nos ha permitido intercambiar información entre usuarios a través de blogs o las populares redes sociales que hoy usan millones de usuarios en todo el mundo.
- La **Web 3.0 (web semántica)** es un salto diferencial en la forma en que accedemos a la información. Los **buscadores** son clave pero no por sus mejores algoritmos, mayor indexación de información u otros extras, sino porque obtenemos **información más personalizada**, descartando información que para cada uno de nosotros será irrelevante.

2. Generaciones de aplicaciones web

- La **Web 4.0** es el próximo gran avance y se centrará en ofrecer un **comportamiento más inteligente**, más predictivo, de modo que podamos con sólo realizar una afirmación poner en marcha un conjunto de acciones que tendrán como resultado aquello que pedimos o decimos.



3. Funcionamiento de las aplicaciones web

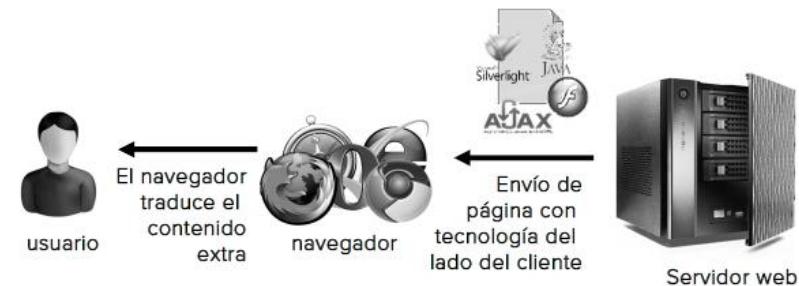
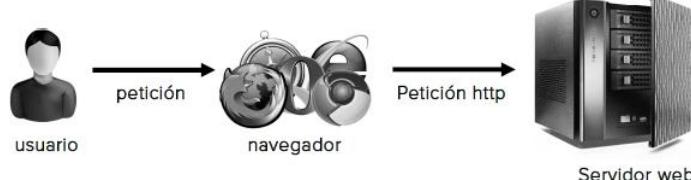
Dos estrategias diferentes

- **LADO CLIENTE (FRONT-END)**: Son elementos que se incorporan al código HTML de una aplicación web (como CSS y Javascript) y deben ser interpretados por el navegador.
- **LADO SERVIDOR (BACK-END)**: Son aplicaciones que se alojan y se ejecutan en el servidor, empleando otros lenguajes de programación y no estando condicionados por los navegadores.

[*Diferencia entre interpretar y compilar*](#)

3. Funcionamiento de las aplicaciones web

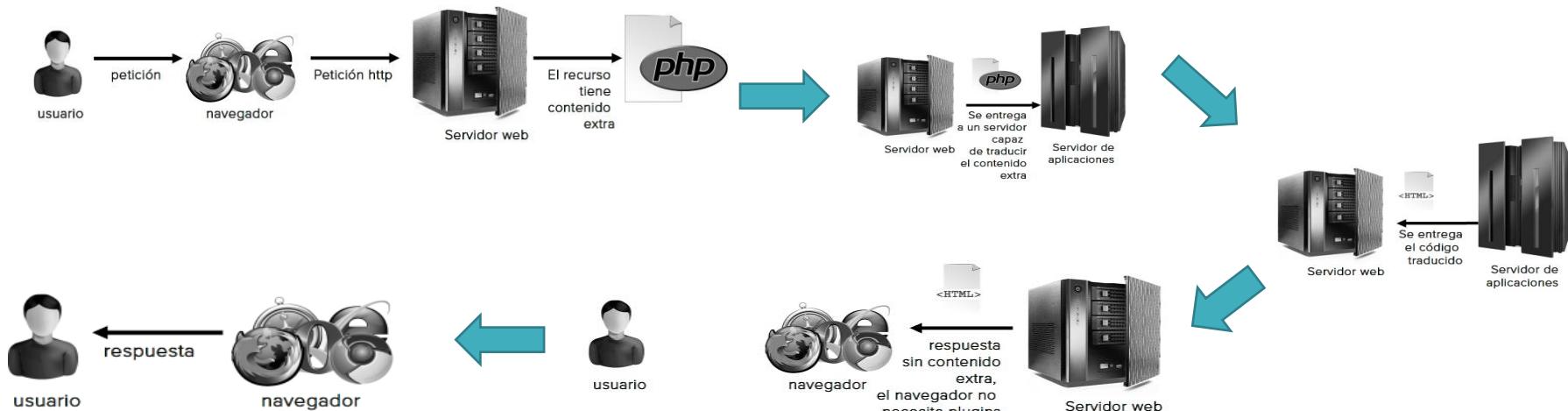
Funcionamiento en el lado cliente



- **HTML.** Lenguaje base para las páginas devueltas por un servidor web ante una petición HTTP.
- **CSS.** Complemento del lenguaje HTML para dotar de estilos a las páginas web.
- **JavaScript.** Aporta interactividad a las páginas web. Cada vez hay más módulos que amplían la potencia y funcionalidad de las mismas.

3. Funcionamiento de las aplicaciones web

Funcionamiento en el lado servidor

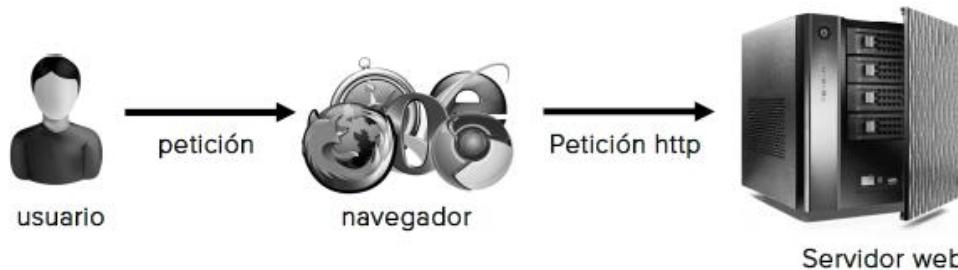


- Cuando un usuario realiza una **petición** a un recurso web, el servidor entiende que tiene elementos a **interpretar** en el lado del servidor y solicita al servidor de aplicaciones su ejecución para posteriormente enviar los **resultados** al cliente en forma documento HTML.

4. Creación de una aplicación web

Servidor web

- Es un software capaz de interpretar las **peticiones** web realizadas con el protocolo **HTTP/HTTPS** y devolver el **resultado** de la petición. Normalmente es el navegador el que pide al servidor web el recurso que desea el usuario. La mayoría de las respuestas a estas peticiones son páginas en HTML.



- El servidor web **no** se molesta en **descifrar** el código de los documentos que entregan. Esta tarea la dejan en manos del **cliente** que hizo la petición, es decir, la realiza el propio navegador.

4. Creación de una aplicación web

Servidor de aplicaciones web

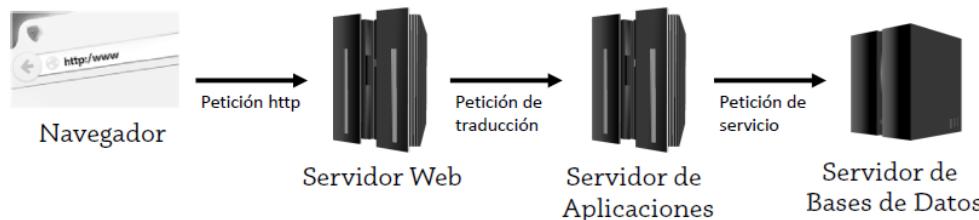
- El servidor de aplicaciones es el encargado de **traducir** las instrucciones hechas por el **lenguaje del lado del servidor** y entregar el **resultado** de la misma al servidor web que hizo la solicitud. Además este código puede contener instrucciones que impliquen a un **tercero** (por ejemplo un servidor de BBDD).
- En la práctica los servidores de aplicaciones son simplemente unos **módulos de software que se añaden al servidor web**. Dependiendo de la tecnología de servidor utilizada, tendremos servidores de tipo PHP, .Net, Java, etc.



4. Creación de una aplicación web

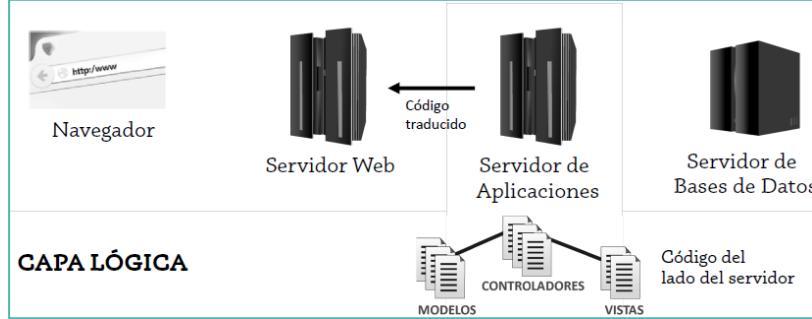
Arquitectura de tres capas

- **CAPA DE PRESENTACIÓN.** Se encarga de manejar, formatear y presentar la información al usuario. Consta del código de programación del lado del cliente (HTML, JavaScript, CSS, etc...)
- **CAPA LÓGICA.** Contiene los documentos escritos en el lenguaje que debe interpretarse en el lado del servidor y cuyo resultado se enviará al servidor web.
- **CAPA DE NEGOCIO.** Es la que contiene la información y debe protegerla de accesos no autorizados. En esta capa fundamentalmente se encuentra el SGBD. El servidor de aplicaciones realiza peticiones al SGBD para responder a las necesidades solicitadas por el usuario en su petición, pero desde la capa de presentación no se pueden acceder a esos recursos de datos.



4. Creación de una aplicación web

Arquitectura de tres capas



4. Creación de una aplicación web

Programación back-end y front-end

- **FRONT-END.** Es la programación de la interfaz de usuario (capa de presentación). Por una parte suelen ser diseñadores de maquetas y por otra parte estarían los programadores encargados de pasar esos diseños a páginas HTML y CSS. Su objetivo es conseguir que la aplicación sea atractiva al usuario y fácil de usar.
- **BACK-END.** Se incluye en este punto la programación de las capas lógica y de negocio. Es la parte encargada de que la aplicación funcione debidamente. Su objetivo es el acceso rápido a datos, gestión de errores, seguridad, eficiencia, etc...

CGI fue la primera tecnología que permitió crear aplicaciones web en el lado servidor.

4. Creación de una aplicación web

Lenguajes de programación para aplicaciones web del lado de servidor

- **Perl.** Lenguaje clásico para crear aplicaciones web tipo CGI. Poderoso para manejar ficheros y expresiones regulares.
- **Python.** Lenguaje fácil y potente que además dispone de numerosas facilidades para crear aplicaciones de red y gráficas. (Gran auge en los últimos tiempos en el mercado laboral).
- **Ruby.** Parecido al anterior, pero con mucho menos uso.
- **Java.** Lenguaje más popular, potente y versátil, pero más difícil de aprender.
- **C#.** Muy utilizado en servidores de aplicaciones .NET.

4. Creación de una aplicación web

Lenguajes de script de servidor

Se basan en la incrustación del código dentro de la página HTML, incluyendo marcas o etiquetas del tipo `<?php` o `<%jsp` o haciendo referencia a ficheros del servidor de aplicaciones .php o .jsp.

- **PHP.** El más utilizado. Se basa en lenguaje C y en Perl. Fácil de aprender.
- **JSP.** Lenguaje basado en Java.
- **ASP.** Tecnología de Microsoft similar a PHP pero para servidores de aplicaciones IIS. La última versión ASP.net