

DESARROLLO WEB EN ENTORNO SERVIDOR

UT 1: Introducción a la programación
Web



UT 1: INTRODUCCIÓN A LA PROGRAMACIÓN WEB

1.Introducción.

2.Conceptos básicos de la Web.

3.Evolución de la web.

4.Arquitectura cliente-servidor.

1.Arquitectura de dos niveles.

2.Arquitectura de tres niveles.

5.Aplicaciones Web

6.PHP

7.Instalación de un servidor de aplicaciones web

1. INTRODUCCIÓN

Internet y web son conceptos que se confunden.

Internet: red de redes formada por dispositivos interconectados. Es un conjunto de redes conectadas mediante el protocolo TCP/IP que permite compartir recursos e intercambiar información.

Web: Es uno de los servicios que proporciona Internet. También conocido como WWW (World Wide Web).

Otros servicios:

- E-mail
- FTP
- Telnet

UT 1: INTRODUCCIÓN A LA PROGRAMACIÓN WEB

- 1.Introducción.
- 2.Conceptos básicos de la Web.**
- 3.Evolución de la web.
- 4.Arquitectura cliente-servidor.
 - 1.Arquitectura de dos niveles.
 - 2.Arquitectura de tres niveles.
- 5.Aplicaciones Web
- 6.PHP
- 7.Instalación de un servidor de aplicaciones web

2. CONCEPTOS BÁSICOS DE LA WEB

Hay 3 conceptos fundamentales relacionados con la Web:

- HTTP
- Hipertexto
- URL



2. CONCEPTOS BÁSICOS DE LA WEB

La Web fue creada en 1989 por Tim Berners Lee.

La Web es una forma de organizar la información usando:

- **Internet** como medio físico de comunicación
- Protocolo **HTTP**

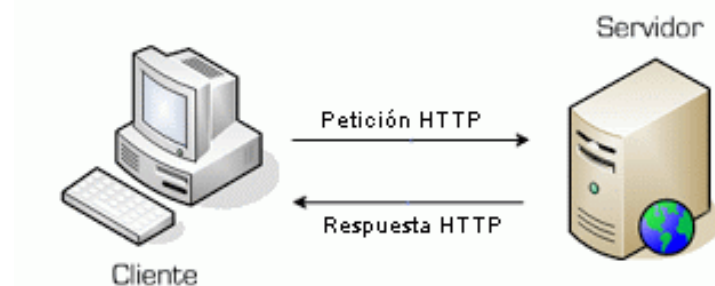
WEB = INTERNET + HTTP

2. CONCEPTOS BÁSICOS DE LA WEB

HTTP (Hypertext Transference Protocol)

Es el protocolo de transferencia de hipertexto que los navegadores utilizan para realizar peticiones a los servidores web y para recibir respuestas de ellos.

Es el protocolo que se utiliza para ver páginas web.



2. CONCEPTOS BÁSICOS DE LA WEB

HTTP (Hypertext Transference Protocol)

Los métodos de petición de HTTP son GET, POST, PUT, HEAD...

El protocolo HTTP está basado en el protocolo TCP.

2. CONCEPTOS BÁSICOS DE LA WEB

HIPERTEXTO

Los **hipertextos** se crean a través de lenguaje HTML (Hypertext Markup Language o Lenguaje de marcas de hipertexto).

Estos archivos son interpretados por el navegador.

El protocolo HTTP permite solicitar documentos que contengan hipertexto.

2. CONCEPTOS BÁSICOS DE LA WEB

URL (Uniform Resource Locator O Localizador de Recursos Uniforme)

Es una dirección que apunta a un único recurso o documento de la Web.

Una URL está formada por:

- Protocolo: <http://>
- Nombre de host: www.ejemplo.com
- Puerto: Donde está escuchando el servidor, que contiene el documento (si no se pone nada el 80)
- Documento: ruta y nombre del archivo

UT 1: INTRODUCCIÓN A LA PROGRAMACIÓN WEB

- 1.Introducción.
- 2.Conceptos básicos de la Web.
- 3.Evolución de la web.**
- 4.Arquitectura cliente-servidor.
 - 1.Arquitectura de dos niveles.
 - 2.Arquitectura de tres niveles.
- 5.Aplicaciones Web
- 6.PHP
- 7.Instalación de un servidor de aplicaciones web

3. EVOLUCIÓN DE LA WEB

WEB 1.0

- Páginas estáticas escritas en HTML
- los usuarios no podían interactuar
- Contenido y diseño junto en el código HTML
- No se usaban bases de datos
- La interactividad se limita a la consulta de datos estáticos.

 **WEB 1.0** {HTML, PORTALS}



3. EVOLUCIÓN DE LA WEB

WEB 2.0

- Uso de base de datos.
- Uso de hojas de estilo.
- Uso de lenguajes de programación.
- Los usuarios pueden crear su propio contenido, son autores.
- Los usuarios pueden interactuar entre ellos.
- Aparecen las redes sociales.
- Aparece la arquitectura orientada al servicio (SOA): los componentes softwares son creados para usarse como un servicio en la red.

Ej: Facebook, YouTube, Wikipedia

3. EVOLUCIÓN DE LA WEB

WEB 1.0 vs WEB 2.0



3. EVOLUCIÓN DE LA WEB

WEB 1.0 vs WEB 2.0



3. EVOLUCIÓN DE LA WEB

WEB 3.0

- Web como una gran base de datos distribuida accesible desde diferentes dispositivos.
- Consolidación de la **web semántica** (red internet inteligente que entiende y relaciona los que el usuario busca)
- La web es una inmensa base de datos accesible desde diferentes dispositivos.
- Web inteligente: aprende de la interacción de los usuarios.

[Video resumen](#)

3. EVOLUCIÓN DE LA WEB

WEB 4.0

- Web ubicua cuyo objetivo es que las personas y las cosas se comuniquen entre sí para tomar decisiones.
- Con este modelo de Web podremos hacer una consulta del tipo: “Quiero que un taxi venga a buscarme” y que tu móvil se comunique automáticamente con la compañía de taxis más cercana, sin intervención directa del usuario.
- Conectados las 24 horas.

3. EVOLUCIÓN DE LA WEB

WEB 4.0



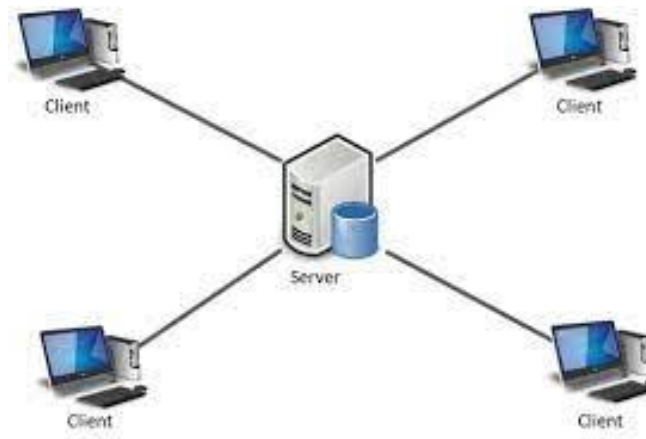
UT 1: INTRODUCCIÓN A LA PROGRAMACIÓN WEB

- 1.Introducción.
- 2.Conceptos básicos de la Web.
- 3.Evolución de la web.
- 4.Arquitectura cliente-servidor.**
 - 1.Arquitectura de dos niveles.
 - 2.Arquitectura de tres niveles.
- 5.Aplicaciones Web
- 6.PHP
- 7.Instalación de un servidor de aplicaciones web

4. ARQUITECTURA CLIENTE-SERVIDOR

En la arquitectura de cliente-servidor es un modelo de diseño de software formada por dos componentes:

- Los **clientes** o programas: representan entidades que necesitan servicios.
- Los **servidores** o programas: proporcionan servicios. Dan respuestas a las peticiones de los clientes.



4. ARQUITECTURA CLIENTE-SERVIDOR

Los servidores y los clientes son objetos separados desde un punto de vista lógico y que se comunican a través de una red de comunicaciones para realizar una o varias tareas de forma conjunta.

Esto puede ir en una sola computadora o sistemas distribuidos.

4. ARQUITECTURA CLIENTE-SERVIDOR

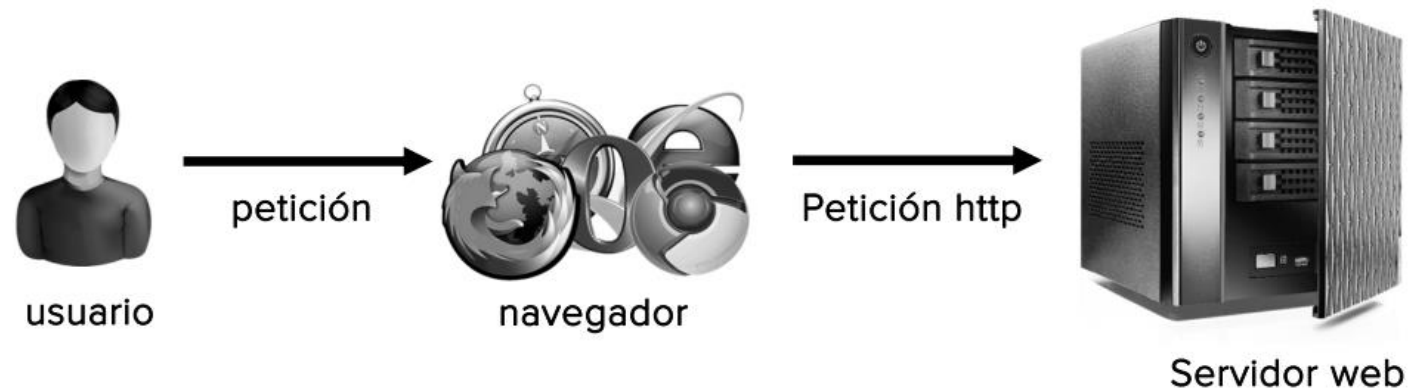
FUNCIONAMIENTO DEL LADO CLIENTE

En modo cliente, la página entregada por el servidor web que la alberga, contiene (además del código HTML, XML O XHTML) elementos pertenecientes a otros lenguajes y tecnologías. Las principales tecnologías son:

- CSS
- JavaScript
- Applets de Java
- Flash (En desuso)

4. ARQUITECTURA CLIENTE-SERVIDOR

FUNCIONAMIENTO DEL LADO CLIENTE



4. ARQUITECTURA CLIENTE-SERVIDOR

FUNCIONAMIENTO DEL LADO CLIENTE

Resumen:

- El cliente realiza una petición de recurso
- Un servidor web entrega el recurso al ordenador del cliente
- El navegador del cliente traduce el código que recibe
- El usuario ve el resultado

4. ARQUITECTURA CLIENTE-SERVIDOR

FUNCIONAMIENTO DEL LADO SERVIDOR

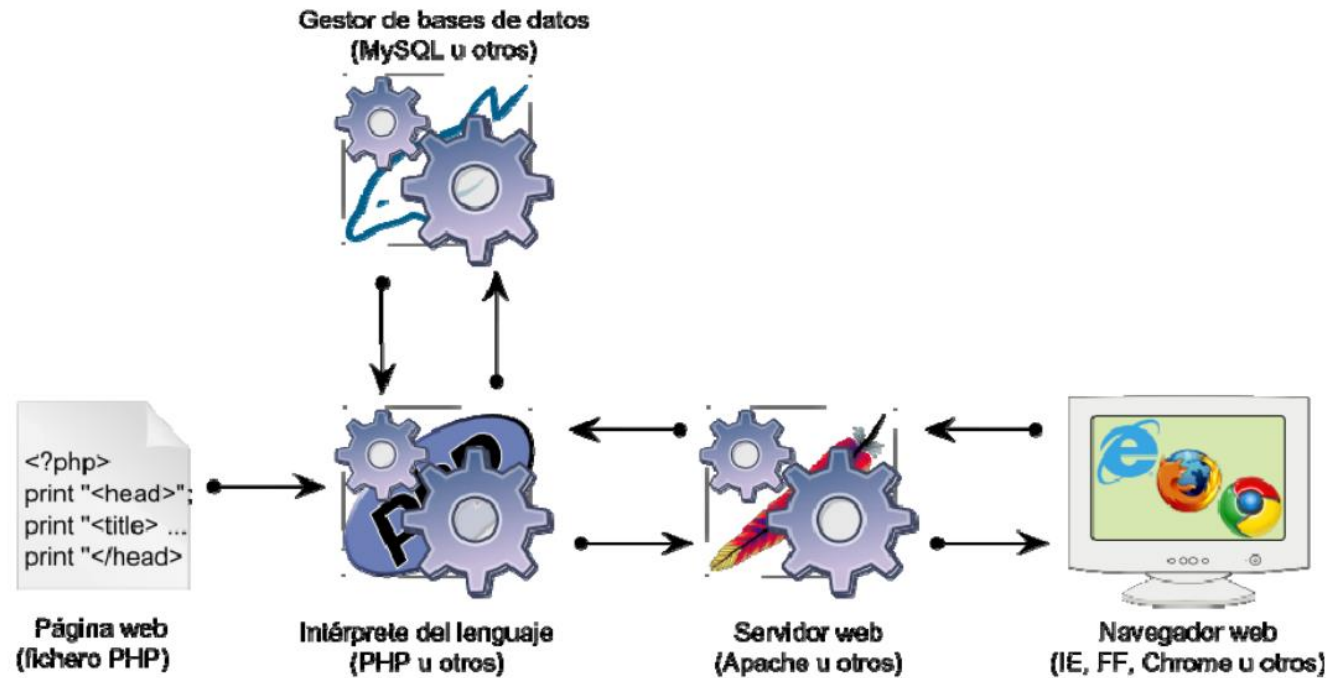
Cuando un usuari@ hace una petición a un recurso web, el servidor que contiene dicho recurso se da cuenta de que contiene elementos a interpretar en el lado del servidor y pide al servidor de aplicaciones adecuado que traduzca esos elementos antes de enviar el resultado al navegador.

El servidor de aplicaciones, que suele ser realmente un módulo software, enviará el resultado al servidor web en un formato traducible en el lado del cliente; es decir, un documento HTML.

El servidor web finalmente enviará al usuario este resultado. La ventaja de este modo de trabajo es que el navegador puede ser más ligero (se la llama tecnología de cliente ligero) y la parte dura o pesada se la lleva el servidor web.

4. ARQUITECTURA CLIENTE-SERVIDOR

FUNCIONAMIENTO DEL LADO SERVIDOR



4. ARQUITECTURA CLIENTE-SERVIDOR

FUNCIONAMIENTO DEL LADO SERVIDOR

- El cliente realiza una petición de recurso
- Un servidor web encuentra el recurso y se encuentra con código especial
- Pide ayuda a otro servidor o servicio para traducir ese código
- El servidor entrega el código traducido al navegador
- El navegador interpreta lo que se le entrega
- El usuario ve el resultado

4. ARQUITECTURA CLIENTE-SERVIDOR

Algunos lenguajes de programación en el lado servidor

- PHP
- Java
- Python
- `javaScript` (`node.js`)

4. ARQUITECTURA CLIENTE-SERVIDOR

Se puede realizar una clasificación de las aplicaciones servidores atendiendo a:

- Protocolo de transporte que utilice : UDP (no confiable) o TCP (confiable)
- Iterativos o Concurrentes: Atienden a los clientes de uno en uno, o varios a la vez.

UT 1: INTRODUCCIÓN A LA PROGRAMACIÓN WEB

- 1.Introducción.
- 2.Conceptos básicos de la Web.
- 3.Evolución de la web.
- 4.Arquitectura cliente-servidor.**
 - 1.Arquitectura de dos niveles.**
 - 2.Arquitectura de tres niveles.
- 5.Aplicaciones Web
- 6.PHP
- 7.Instalación de un servidor de aplicaciones web

4.1. ARQUITECTURA DE DOS NIVELES

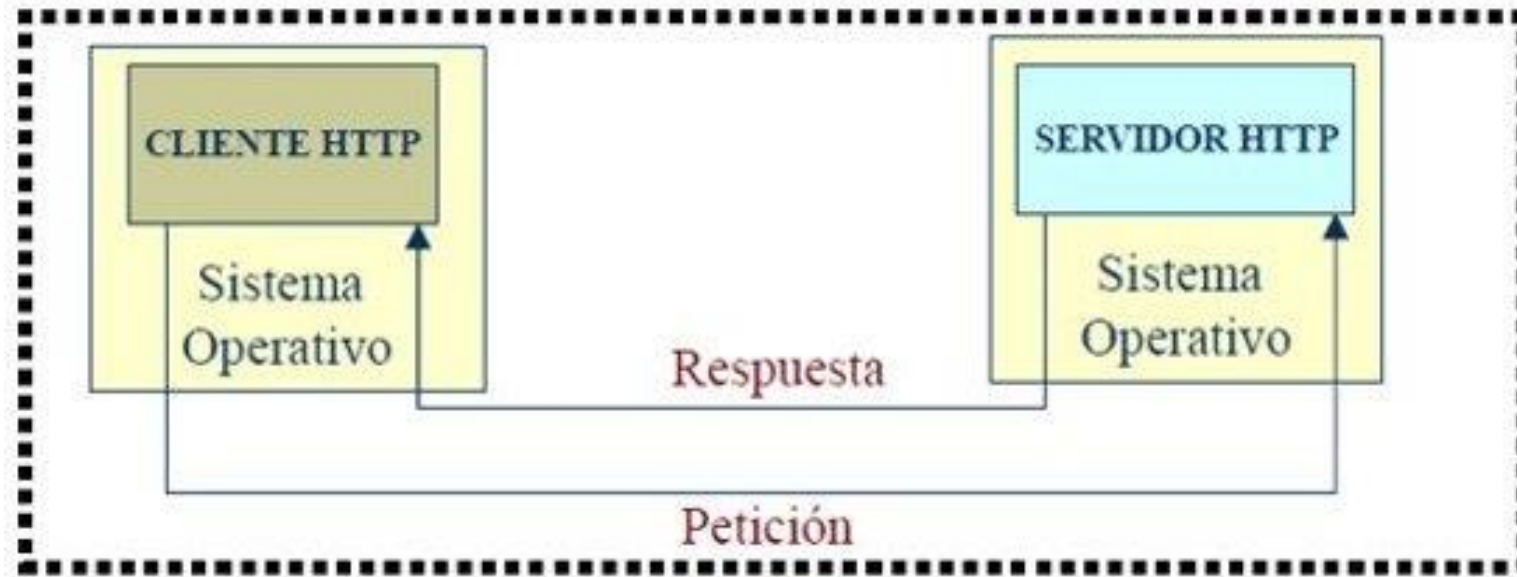
Sistema con dos elementos:

- Cliente
- Servidor

Un cliente hace una petición de un servicio y recibe la respuesta a dicha petición.
Un servidor recibe y procesa la petición, y devuelve la respuesta solicitada.

Si hay una gran cantidad de procesos o son muy complejos, este modelo no es eficaz. Se produce una congestión en la red. Este problema se llama *cliente pesado*.

4.1. ARQUITECTURA DE DOS NIVELES



UT 1: INTRODUCCIÓN A LA PROGRAMACIÓN WEB

- 1.Introducción.
- 2.Conceptos básicos de la Web.
- 3.Evolución de la web.
- 4.Arquitectura cliente-servidor.**
 - 1.Arquitectura de dos niveles.
 - 2.Arquitectura de tres niveles.**
- 5.Aplicaciones Web
- 6.PHP
- 7.Instalación de un servidor de aplicaciones web



4.2. ARQUITECTURA DE TRES NIVELES

Este modelo intenta solucionar los problemas de sobrecarga equilibrando las tareas que tiene que soportar el servidor.

Se reparten las funciones en tres capas:

- Capa de presentación (Cliente): Es el ordenador que a partir de la interfaz de usuario realiza las peticiones al servidor.
- Capa de negocio (Servidor de aplicaciones)
- Capa de datos (Servidor de datos)



4.2. ARQUITECTURA DE TRES NIVELES

- **Capa de presentación:** Maneja la parte de la aplicación web que ve el usuario. Es decir, se encarga de la forma de presentar la información al usuario.
Consta del código del lado del cliente (HTML, JavaScript, CSS, Flash...) que le llega al navegador, aunque ese código haya sido generado originalmente por una tecnología del lado del servidor.

Se encarga de la interfaz de usuario y la interacción con el usuario

4.2. ARQUITECTURA DE TRES NIVELES

- **Capa de negocio:** Es la encargada de gestionar el funcionamiento de la aplicación.

En ella se encuentran los documentos escritos en un lenguaje que se debe de interpretar en el lado del servidor (por ello, esta capa está relacionada con el servidor de aplicaciones) y cuyo resultado se enviará al servidor web para que este, a su vez, lo envíe al cliente que hizo la petición.

4.2. ARQUITECTURA DE TRES NIVELES

- **Capa de datos:** En esta capa, fundamentalmente, se encuentra el sistema gestor de bases de datos (SGBD) de la empresa, es decir, la capa que se conecta y accede a los datos.

4.2. ARQUITECTURA DE TRES NIVELES

VENTAJAS

- Separación de responsabilidades: La arquitectura de tres capas separa claramente las responsabilidades y funciones en una aplicación, lo que facilita la comprensión, el mantenimiento y la modificación del código.
- Reutilización de código: Los componentes de lógica de negocio y acceso a datos pueden utilizarse en diferentes partes de la aplicación o incluso en aplicaciones diferentes.
- Escalabilidad: La arquitectura de tres capas permite la escalabilidad de la aplicación de manera más efectiva. Puedes escalar cada capa de forma independiente según las necesidades, lo que facilita la gestión del rendimiento a medida que la aplicación crece.

4.2. ARQUITECTURA DE TRES NIVELES

VENTAJAS

- **Mantenibilidad:** Debido a la clara separación de responsabilidades y la modularidad, es más fácil realizar cambios, actualizaciones y correcciones de errores en una aplicación de tres capas sin afectar otras partes del sistema.
- **Seguridad:** Puedes aplicar medidas de seguridad en cada capa para proteger la aplicación y los datos. La capa de presentación se encarga de la autenticación y la autorización, la capa de lógica de negocio implementa reglas de seguridad y la capa de datos puede tener medidas de seguridad como cifrado de datos.

4.2 ARQUITECTURA DE TRES NIVELES

PATRÓN MODELO-VISTA-CONTROLADOR (MVC)

No confundir la arquitectura cliente-servidor de 3 niveles con el MVC.

El patrón MVC es un diseño de software que se utiliza principalmente en aplicaciones de interfaz de usuario, como aplicaciones web y de escritorio.

4.2 ARQUITECTURA DE TRES NIVELES

PATRÓN MODELO-VISTA-CONTROLADOR (MVC)

Divide la aplicación en tres componentes principales:

- El Modelo representa la lógica de negocio y los datos de la aplicación. Es responsable de la gestión de datos y operaciones relacionadas con los datos.
- La Vista es responsable de la presentación de los datos y la interfaz de usuario. Muestra la información al usuario y refleja los cambios en el Modelo.
- El Controlador actúa como intermediario entre el Modelo y la Vista, gestionando las interacciones del usuario y actualizando el Modelo y la Vista según sea necesario.

4.2 ARQUITECTURA DE TRES NIVELES

PATRÓN MODELO-VISTA-CONTROLADOR (MVC)

Framework basados en el MVC:

- Angular: basado en una variante del modelo MVC.
- Symphony
- Backbone.js

4.2 ARQUITECTURA DE TRES NIVELES

DIFERENCIA ENTRE LA ARQUITECTURA DE TRES NIVELES Y EL MVC

El Modelo-Vista-Controlador (MVC) y la arquitectura cliente-servidor de tres capas son dos conceptos diferentes, aunque comparten algunos principios de diseño. Ambos enfoques se utilizan para organizar y estructurar aplicaciones de software, pero tienen propósitos y enfoques ligeramente diferentes.

MVC se utiliza principalmente en aplicaciones de interfaz de usuario, como aplicaciones web y de escritorio, donde la interacción con el usuario es un componente clave.

La arquitectura cliente-servidor de tres capas es un enfoque más general para organizar aplicaciones de software, ya sean aplicaciones de interfaz de usuario o servicios web.

4.2 ARQUITECTURA DE TRES NIVELES

DIFERENCIA ENTRE LA ARQUITECTURA DE TRES NIVELES Y EL MVC

En resumen, la principal diferencia radica en el enfoque y el alcance: MVC es un patrón de diseño específico para la organización de la interfaz de usuario en aplicaciones interactivas, mientras que la arquitectura cliente-servidor de tres capas es un enfoque más general para organizar aplicaciones de software en capas distintas, independientemente de si tienen una interfaz de usuario o no.

UT 1: INTRODUCCIÓN A LA PROGRAMACIÓN WEB

- 1.Introducción.
- 2.Conceptos básicos de la Web.
- 3.Evolución de la web.
- 4.Arquitectura cliente-servidor.
 - 1.Arquitectura de dos niveles.
 - 2.Arquitectura de tres niveles.
- 5.Aplicaciones Web**
- 6.PHP
- 7.Instalación de un servidor de aplicaciones web

5. APLICACIONES WEB

Herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una Intranet a través de un navegador.

Lenguajes de programación

- Lado cliente (Frontend): javaScript, HTML, CSS
- Lado servidor (Backend): PHP, Java, Python, javaScript (node.js)

5. APLICACIONES WEB

DESARROLLO DE APLICACIONES WEB

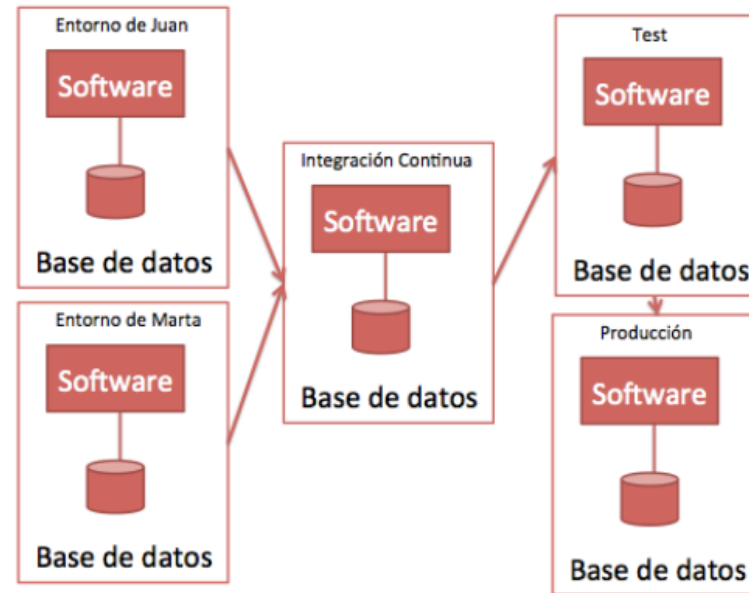
- Específicas: Son desarrolladas para gestionar un problema específico. En la actualidad en el desarrollo de aplicaciones web se utilizan framework: symfony, django, RoR,...
- CMS: Sistema de gestión de contenido. Son aplicaciones desarrolladas por un tercero. Ejemplos: blogs, portales, wiki, comercio electrónico, ... Las más conocidas: wordpress, joomla, mediawiki, prestashop,...

5. APLICACIONES WEB

- La puesta en marcha de un proyecto de software comprende en algo más que desarrollar el código.
- El software es algo vivo, sigue cambiando una vez se ha entregado; se sigue desarrollando y generando nuevas versiones.
- Poner en producción una aplicación es algo crítico, genera incertidumbre.
- Es recomendable automatizar todas las tareas posibles y establecer políticas de calidad y control claras son fundamentales para que el proceso de dar servicio de calidad a los clientes (implementar cambios) sea un proceso frecuente, sencillo y seguro.

5. APLICACIONES WEB

Entornos



- Desarrollo
- Test
- Producción

5. APLICACIONES WEB

DIFERENTES MANERAS DE DESARROLLO WEB

- Estática: Mediante páginas html estáticas, en las que el usuario no puede interactuar (enviar información), solo puede leer.
- Html Dinámico: Se añade funcionalidad a la página web en el lado del cliente (navegador), programas que son ejecutados por el navegador. El usuario puede ver el código. Ejemplo: Javascript
- Aplicaciones en el servidor: Se generan las páginas al realizar la petición al servidor como consecuencia de la ejecución de programas (asp, php, servlet, etc..). Se permite el acceso a BD. Uso de XML para presentar la información. PHP, ASP.NET, Servlets, CGI.

5. APLICACIONES WEB

SOFTWARE

El software de desarrollo que se puede usar va desde editores de texto a entornos de desarrollo más avanzados, también conocido como IDE (Integrated development environment):

- Visual Studio Code: libre distribución, muchos plugins de ayuda, sirve para diversos lenguajes y multiplataforma.
- Sublime: libre distribución
- Aptana: Entorno profesional para desarrollo web, libre distribución
- DreamWeaver: Similar a Aptana, con diseñador visual.
- Eclipse: Entorno de desarrollo multiplataforma.
- DevPHP: Entorno muy básico

UT 1: INTRODUCCIÓN A LA PROGRAMACIÓN WEB

- 1.Introducción.
- 2.Conceptos básicos de la Web.
- 3.Evolución de la web.
- 4.Arquitectura cliente-servidor.
 - 1.Arquitectura de dos niveles.
 - 2.Arquitectura de tres niveles.
- 5.Aplicaciones Web
- 6.PHP**
- 7.Instalación de un servidor de aplicaciones web

6. PHP

PHP (Hypertext Pre-processor)

- Lenguaje interpretado apareció en 1994
- ¿Porqué PHP?
 - Es un lenguaje potente y “fácil” de aprender, de libre distribución
 - Permite acceso a base de datos.
 - Muchas aplicaciones web se desarrollan en PHP: Gestores de contenidos, blogs, tiendas online, foros, etc..



6. PHP

- Sintaxis similar a C.
- Orientado a objetos (no puro)
- Es necesario tener instalado un interprete que entienda y ejecute las instrucciones
- Página oficial www.php.net
- Documentación <http://www.php.net/download-docs.php>
- Más información en <http://es.wikipedia.org/wiki/PHP>
- Fichero de configuración php.ini

6. PHP

Es un lenguaje de scripting.

Los lenguajes de scripting son un tipo específico de lenguajes informáticos que se pueden utilizar para dar instrucciones a otro software, como un navegador web, un servidor o una aplicación independiente.

Los lenguajes de scripting se caracterizan por ser interpretados en lugar de compilados, lo que significa que el código se ejecuta línea por línea por un intérprete en tiempo de ejecución en lugar de convertirse previamente en código máquina.

6. PHP

Muchos de los lenguajes de programación más populares hoy en día son lenguajes de script.

Ejemplo: JavaScript, PHP, Ruby y Python.

Ejemplo de lenguaje de programación: Java.

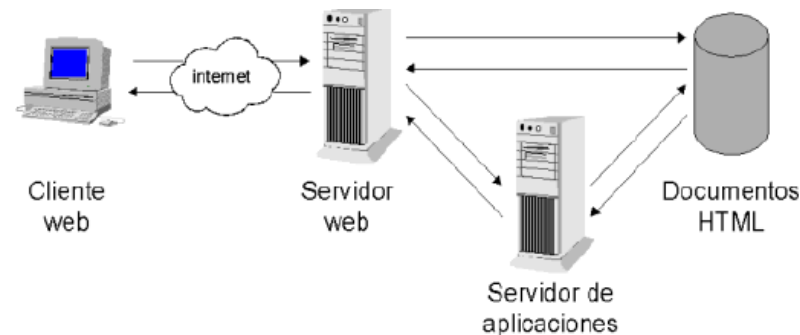
6. PHP

Lenguajes de scripting	Lenguajes de programación
Específicos de plataforma	Independientes de la plataforma (multiplataforma)
(Mayoritariamente) interpretado	Compilado
Más lento en tiempo de ejecución	Más rápido en tiempo de ejecución
Menos intensivo en código	Más intensivo en código
Crea aplicaciones como parte de un stack	Crea aplicaciones independientes

6. PHP

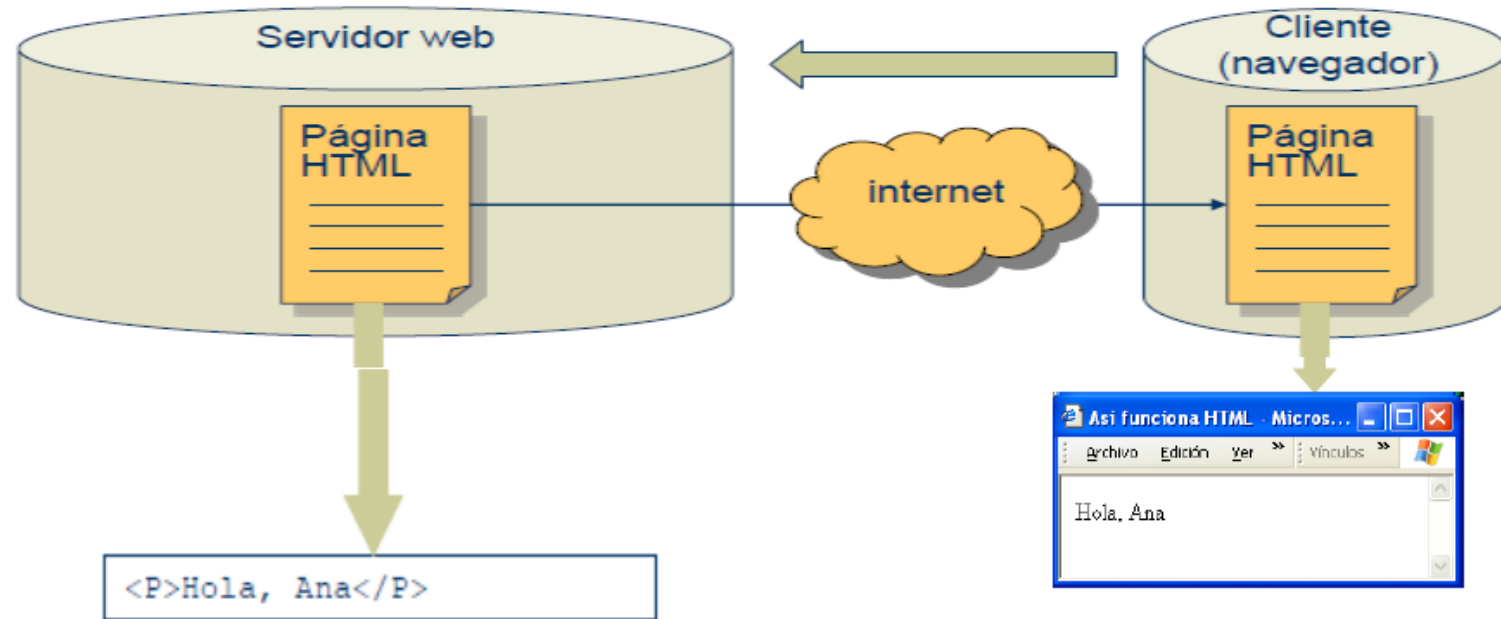
¿CÓMO FUNCIONA PHP?

- PHP es un lenguaje de script del lado del servidor. Otros lenguajes similares son ASP, JSP o ColdFusion
- Los scripts PHP están incrustados en los documentos HTML y el servidor los interpreta y ejecuta antes de servir las páginas al cliente
- El cliente no ve el código PHP sino los resultados que produce



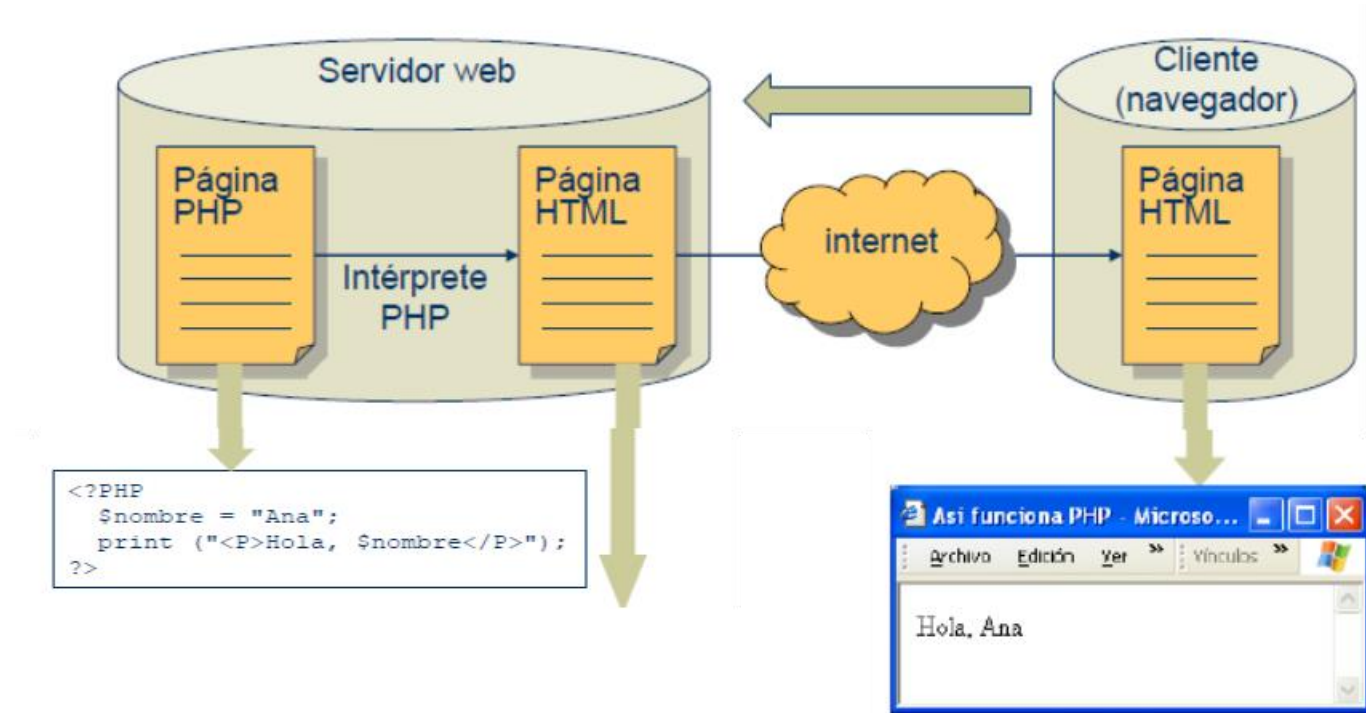
6. PHP

¿CÓMO FUNCIONA PHP? EJEMPLO SIN PHP



6. PHP

¿CÓMO FUNCIONA PHP?



6. PHP

¿CÓMO FUNCIONA PHP?

- Se escribe el código fuente de la página web en lenguaje PHP.
- Se guarda este código en el servidor web.
- El navegador de un usuario solicita al servidor la página codificada y guardada en los dos pasos anteriores.
- El servidor interpreta el código PHP.
- El servidor envía el resultado del código PHP al navegador del usuario, que ve en su pantalla la página en formato HTML.

6. PHP

Dos formas de trabajar con el lenguaje PHP:

- Scripts en maquina local: En este caso solo es necesario tener instalado el intérprete PHP
- Scripts para desarrollo web: Será necesario:
 - Servidor web: Apache
 - MySql: Sistema gestor de base de datos: MySql, Oracle...
 - Intérprete del lenguaje que lleva el mismo nombre: PHP
 - Herramienta para administrar la base de datos: Phpmyadmin, SQL Developer...

6. PHP

Documentación

Manuales en www.apache.org, www.php.net, www.mysql.com

Tutoriales, scripts, artículos

www.phpbuilder.com

www.codewalkers.com

www.devshed.com

www.tutorialfind.com

www.melonfire.com

UT 1: INTRODUCCIÓN A LA PROGRAMACIÓN WEB

- 1.Introducción.
- 2.Conceptos básicos de la Web.
- 3.Evolución de la web.
- 4.Arquitectura cliente-servidor.
 - 1.Arquitectura de dos niveles.
 - 2.Arquitectura de tres niveles.
- 5.Aplicaciones Web
- 6.PHP
- 7.Instalación de un servidor de aplicaciones web**

7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

Para que un ordenador tenga la capacidad de mostrar páginas web, es necesario tener instalado un servidor web.

Diferentes servidores web:

- Apache: Servidor web gratuito, multiplataforma, modular y robusto.
Limitaciones en el rendimiento
- Nginx: Servidor web gratuito, multiplataforma con grandes prestaciones sobretodo en atender peticiones concurrentes.
- ISS: Servidor web propietario, orientado a Windows.

7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

Vamos a ver la instalación de los siguientes componentes

- Servidor Web: Apache
- Servidor de Base de datos: MySql
- Interprete PHP: PHP
- Administrador de base de datos: PHPMyAdmin



7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

SERVIDOR WEB: APACHE

- Servidor Web de código abierto
- Multiplataforma
- Fácilmente integrable con Mysql y php
- Modular
- Página oficial: <http://www.apache.org/>
- Servidor <http://httpd.apache.org/>
- Fichero de configuración httpd.conf



7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

MYSQL

- Servidor de base de datos de código abierto
- SGBD, distribuido con licencia GNU
- Página oficial <http://www.mysql.com/>



7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

PHPMYADMIN

- Herramienta web para administrar bases de datos mysql
- Escrita en PHP
- Página oficial http://www.phpmyadmin.net/home_page/
- Documentación <http://localhost/phpmyadmin/Documentation.html>



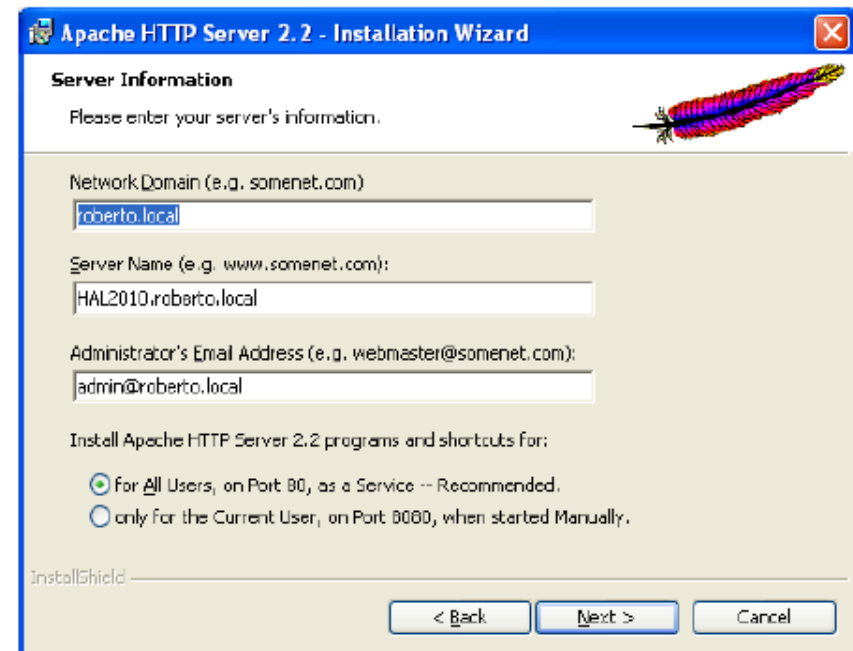
7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

GUÍA DE INSTALACIÓN EN WINDOWS

1ºPaso: Instalar Servidor Web Apache

- Es importante descargar la versión de Apache que nos recomiendan desde la página oficial de PHP para Windows.

<https://windows.php.net/download/>



7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

GUÍA DE INSTALACIÓN EN WINDOWS

2º PASO: Interprete PHP

- <https://www.php.net/downloads.php>
- Ver requerimientos previos
- El archivo de configuración es php.ini. (subida de archivos vía HTTP, sesiones, directorios...)
- ¿Dónde se encuentra php.ini?

```
<?php  
    phpinfo();  
?>
```

7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

GUÍA DE INSTALACIÓN EN WINDOWS

- Lista de directivas de configuración de PHP:
<https://www.php.net/manual/es/ini.list.php>
- Ejemplo de configuración de php.ini

```
register_globals=off
short_open_tag=on
extension_dir="c:\directoriophp\ext"
extension=php_mysql.dll(y todas las que se quieran)
upload_tmp_dir="C:\Servidor\PHP\uploads"
session.save_path= "C:\Servidor\PHP\sessions"
```


7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

GUÍA DE INSTALACIÓN EN WINDOWS

3º PASO: Configurar el Servidor Apache

Archivo de configuración de Apache es httpd.conf

Ejemplo:

Alias /ejemplo /srv/www/ejemplo

<Directory "/srv/www/ejemplo">

Options Indexes Includes SymLinksIfOwnerMatch

AllowOverride all

Require local

Require all granted

</Directory>

7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

GUÍA DE INSTALACIÓN EN WINDOWS

4º PASO: Instalación de PhpMyAdmin

- Extraer el fichero en una carpeta del servidor web
- Archivo de configuración config.inc.php
 - \$cfg['PmaAbsoluteUri'] = '<http://localhost/phpmyadmin/>';
 - \$cfg['blowfish_secret'] = 'aquí puedes poner lo que quieras'; Semilla para la encriptación de contraseñas al usar la autenticación con cookies
- Habilitar la extensión mbstring.dll en php.ini (extension=mbstring.dll)

7. INSTALACIÓN DE UN SERVIDOR DE APLICACIONES WEB

La instalación se puede hacer de forma manual o mediante soluciones integradas que incluyen todo el software necesario para montar un servidor web con soporte mysql y phpmyadmin:

- XAMPP: Incluye además un servidor ftp. Disponible para Windows y Linux
- AppServ: Similar a Xampp
- Wamp

VAMOS A VER COMO SE INSTALA XAMPP EN
WINDOWS