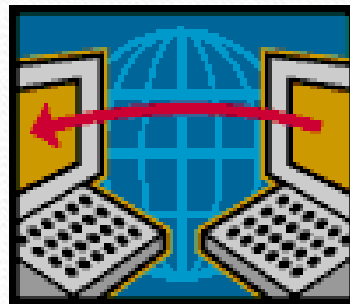


U.T. 1: Arquitecturas y herramientas de programación en entorno cliente/servidor.



Contenidos

- ❑ Modelos de programación en entornos cliente / servidor.
 - ❑ Generación dinámica de páginas web.
 - ❑ Lenguajes de programación en entorno servidor.
 - ❑ Integración con los servidores web.
 - ❑ Herramientas de programación.
-

La web

☐ ***World Wide Web:***

- ☐ Universo de información interconectada, accesible a través de internet.
- ☐ Propuesta por Tim Berners-Lee (1989).
- ☐ Ha tenido mayor difusión que otros servicios contemporáneos (Archie, Gopher, WAIS), gracias sobre todo a uno de sus elementos: el HTML.
 - ☐ Hipertexto.
 - ☐ Hipermedia.

Arquitectura cliente/servidor

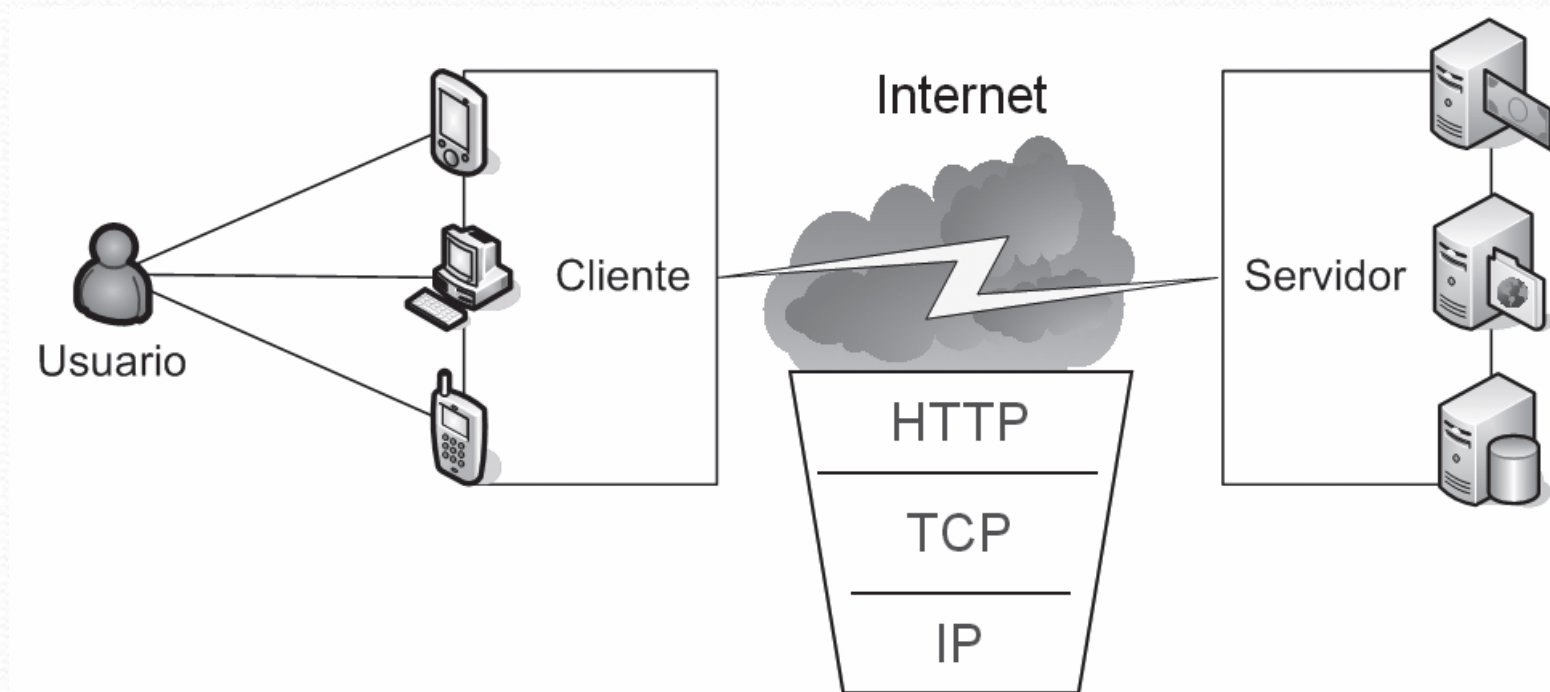
El funcionamiento de la Web es posible gracias a la coexistencia de una serie de componentes hardware y software:

- ☐ hubs, repetidores, puentes, pasarelas, enrutadores, etc.
 - ☐ protocolos de comunicaciones (TCP, IP, **HTTP**, FTP, SMTP, etc.)
 - ☐ sistema de nombres de dominio (DNS) para la búsqueda y utilización de recursos => **URL** del recurso solicitado.
 - ☐ software específico para consumir los recursos
-

Arquitectura cliente/servidor

- ❑ La arquitectura cliente/servidor está basada en la idea de **servicio**.
 - ❑ El **cliente** es un componente consumidor de servicios.
 - ❑ El **servidor** es un proceso proveedor de dichos servicios.
 - ❑ La comunicación entre ambos componentes se lleva a cabo a través del **intercambio de mensajes**.
 - ❑ Normalmente el cliente, a través de un navegador, inicia el intercambio de información, solicitando datos al servidor.
 - ❑ El servidor responde enviando uno o más flujos de datos al cliente.
-

Arquitectura cliente/servidor



Arquitectura cliente/servidor: Clientes

- ❑ Originan el tráfico web.
 - ❑ Envían las peticiones y reciben las respuestas.
 - ❑ Dos clases de clientes web: navegadores y robots.
 - ❑ Los navegadores (Netscape, IE, Chrome, Opera, FireFox, Safari).
 - ❑ Las peticiones están dirigidas por el usuario.
 - ❑ Repiten peticiones al mismo objeto cuando navegan por un site.
 - ❑ Utilizan caches de memoria y disco.
 - ❑ Robots (Motores de búsqueda)
 - ❑ Las peticiones son automatizadas.
-

Arquitectura cliente/servidor:

Funciones de los navegadores

- ☐ Construyen y envían la petición HTTP.
- ☐ Reciben, interpretan y presentan la respuesta.
- ☐ Proporcionan el interfaz para conectarse y utilizar otros servicios: mail, news, ftp, etc.
 - ☐ El protocolo por defecto es http.
- ☐ Caché local.
 - ☐ Sirve recursos guardados en la caché sin conectarse al servidor.
- ☐ Manejo de las Cookies.

Arquitectura cliente/servidor: Servidores

Su cometido básico es proveer de contenido estático a un cliente que ha realizado una petición a través de un navegador.



El servidor es un lugar donde almacenar y administrar los recursos HTML (DHTML).

Arquitectura cliente/servidor: Servidores

Las peticiones al servidor contienen una dirección de tipo URL (Localizador Uniforme de Recurso) formada por:

- ❑ la referencia a un cierto protocolo (HTTP, FTP, etc.)
- ❑ la dirección IP o nombre de dominio del servidor
- ❑ la descripción del recurso en forma de **ruta** al objeto que queremos descargar.
- ❑ opcionalmente se puede incluir el puerto por el que el servidor escucha las peticiones del cliente.

Arquitectura cliente/servidor: Servidores

- ❑ Programa que contesta y genera la respuesta HTTP a las peticiones de recursos web por parte del cliente.
 - ❑ Involucra múltiples servidores, scripts, bases de datos, ..
 - ❑ Trabajo básico:
 1. Se conecta con el cliente.
 2. Recibe el mensaje HTTP de la petición (GET, HEAD, POST)
 3. Procesa el mensaje HTTP.
 4. Localiza y envía el resultado (en forma de mensaje HTTP)
 - ❑ Los servidores de altas prestaciones, además:
 - ❑ Tratan múltiples peticiones:
 - ❑ hilos para manejar cada conexión.
 - ❑ Generan dinámicamente contenido: ASP, PHP, JSP
-

Arquitectura cliente/servidor: Servidores

□ Los más populares son :

- Apache
- Microsoft IIS (Internet Information Services)
- NGINX
- Lighttpd
- Sun Java System Web Server

Ejercicio: Investiga las características y diferencias de cada uno de éstos servidores web: plataforma en la que se ejecutan, lenguajes de programación que interpretan, propietario, cuota de mercado en 2014, etc.

Arquitectura cliente/servidor: Servidores

El intercambio de información cliente/servidor puede requerir también:

- ❑ Controlar el acceso a recursos restringidos:
 - ❑ Autenticación.
 - ❑ Piden al usuario que se identifique (login y password)
 - ❑ La información se incluye en la cabecera del mensaje.
 - ❑ Autorización.
 - ❑ Comprobar en la lista de acceso si el usuario está autorizado.

Aplicaciones web

- ❑ Una aplicación web es proporcionada por un servidor web y utilizada por usuarios que se conectan desde cualquier punto vía clientes web (navegadores).
 - ❑ Definición:
 - ❑ Son aplicaciones basadas en el modelo Cliente/Servidor gestionadas por servidores web, y que utilizan como interfaz páginas web.
 - ❑ La colección de páginas son en una buena parte dinámicas (ASP, PHP, etc.), y están agrupadas lógicamente para dar un servicio al usuario.
 - ❑ El acceso a las páginas está agrupado también en el tiempo (sesión).
 - ❑ Ejemplos: venta de libros, reserva de billetes, etc.
-

Arquitectura de las aplicaciones web

- ❑ Las funcionalidades en la arquitectura cliente/servidor de la web se agrupan en diferentes capas.
- ❑ Cada capa se centra en la gestión de un aspecto determinado del sistema web.
- ❑ El modelo más extendido divide la arquitectura cliente/servidor en tres capas:
 - ❑ Capa de presentación
 - ❑ Capa de negocio
 - ❑ Capa de persistencia o de datos

Arquitectura de las aplicaciones web

□ Capa de presentación

- Es la capa que ve el usuario
 - Presenta la interfaz gráfica del recurso solicitado
 - Sirve para recoger la interacción con el usuario
 - Suele estar situada en el cliente
-

Arquitectura de las aplicaciones web

☐ Capa de negocio

- ☐ Gestiona y conoce las funcionalidades que esperamos del sistema web.
 - ☐ Recibe las peticiones del usuario, las procesa y envía las respuestas apropiadas.
 - ☐ Suele estar programada tanto en el cliente como en el servidor.
-

Arquitectura de las aplicaciones web

- ❑ Capa de persistencia o de datos

- ❑ Encargada del almacenamiento y acceso a los datos
- ❑ Está formada por uno o más gestores de bases de datos que administran y resuelven las solicitudes de almacenamiento y recuperación de información desde la capa de negocio.

Generación de páginas web

- ❑ Al observar la capacidad de las aplicaciones web de comunicarse con los usuarios, las podemos clasificar en:
 - ❑ Aplicaciones web estáticas
 - ❑ Aplicaciones web dinámicas
 - ❑ Aplicaciones web interactivas
-

Generación de páginas web

□ Aplicaciones web estáticas

- El usuario recibe una página web desde el servidor, que no conlleva ningún tipo de acción, ni en la propia página, ni genera ninguna respuesta por parte del servidor.
 - Utilizan HTML para la organización visual de la información.
-

Generación de páginas web

□ Aplicaciones web estáticas

```
001  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
002  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es-ES" lang="es-ES">
003      <head>
004          <title>Título de la página</title>
005          <meta http-equiv="content-type" content="text/html; charset=utf-8">
006      </head>
007      <body>
008          <p>Ha accedido a las 9:20</p>
009      </body>
010  </html>
```

Generación de páginas web

□ Aplicaciones web dinámicas

- La interacción del cliente con la página web recibida desde el servidor produce algún cambio en la visualización de la misma (cambio de formato, ocultación de partes de la página, comienzo de animaciones, aparición de elementos nuevos, ...).
 - Incluyen DHTML, Flash, CSS, JavaScript, ...
-

Generación dinámica de páginas web

□ Aplicaciones web dinámicas

```
001 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
002 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es-ES" lang="es-ES">
003     <head>
004         <title>Título de la página</title>
005         <meta http-equiv="content-type" content="text/html; charset=utf-8">
006     </head>
007     <body>
008         <p>Ha accedido a las <?php echo date('G:i');?></p>
009     </body>
010 </html>
```

Generación dinámica de páginas web

□ Aplicaciones web dinámicas

```
001 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
002 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es-ES" lang="es-ES">
003     <head>
004         <title>Título de la página</title>
005         <meta http-equiv="content-type" content="text/html; charset=utf-8">
006     </head>
007     <body>
008         <p>Ha accedido a las <script type='text/javascript'>var hora=new
    Date();document.write(hora.getHours()+':'+hora.getMinutes());</script></p>
009     </body>
010 </html>
```


Generación dinámica de páginas web

☐ Aplicaciones web interactivas

- ☐ La interacción del cliente con la página web recibida desde el servidor hace que se genere un diálogo entre ambos.
 - ☐ Son las aplicaciones que más se utilizan en Internet actualmente.
 - ☐ En el lado cliente encontramos HTML, controles ActiveX, Flash, applets, AJAX, etc.
 - ☐ En el lado servidor se utilizan lenguajes embebidos en código HTML como PHP, ASP, JSP, enlaces a ejecutables CGI, servlets, ASP.net.
-

Lenguajes de programación en entorno servidor

Son aquellos cuyo código, precompilado o interpretado, es ejecutado en el servidor por un software específico para dicho código.

Existen múltiples alternativas a la hora de ejecutar código en el servidor:

- ❑ Lenguajes de scripting (PHP, ASP, JSP, ...)
 - ❑ Enlaces a código ejecutable (CGI, JSP, EJB, ...)
 - ❑ Estrategias híbridas que utilizan tanto enlaces a código ejecutable como código embebido en la propia página web (Web Forms de ASP.net).
-

Lenguajes de scripting

- ❑ Código que se intercala con el código HTML de una aplicación web.
 - ❑ El código está formado por instrucciones escritas en multitud de lenguajes de programación que son ejecutadas por un servidor para proporcionar dinamismo a la página web.
 - ❑ El servidor web debe tener instalado un módulo o programa que le permita interpretar el lenguaje de programación del código embebido en la página web.
-

Lenguajes de scripting

❑ PHP (Hypertext Processor)

- ❑ gratuidad, código abierto, portable en diferentes plataformas
- ❑ lenguaje interpretado
- ❑ construcciones orientadas a objetos
- ❑ lo soportan la mayoría de los servidores web

❑ ASP (Active Server Pages)

- ❑ tecnología propietaria de código cerrado de Microsoft.
 - ❑ diseñado para ejecutarse especialmente sobre IIS (Internet Information Server)
 - ❑ actualmente este lenguaje ha dejado paso a la versión ASP.net
-

Lenguajes de scripting

❑ Perl

- ❑ inicialmente se utilizó para manipular cadenas de caracteres.
- ❑ primeros lenguajes para la programación web.
- ❑ código abierto
- ❑ basado en programación estructurada como C

❑ Python

- ❑ portable en diferentes plataformas y gratuito
 - ❑ orientado a objetos
 - ❑ lenguaje interpretado
-

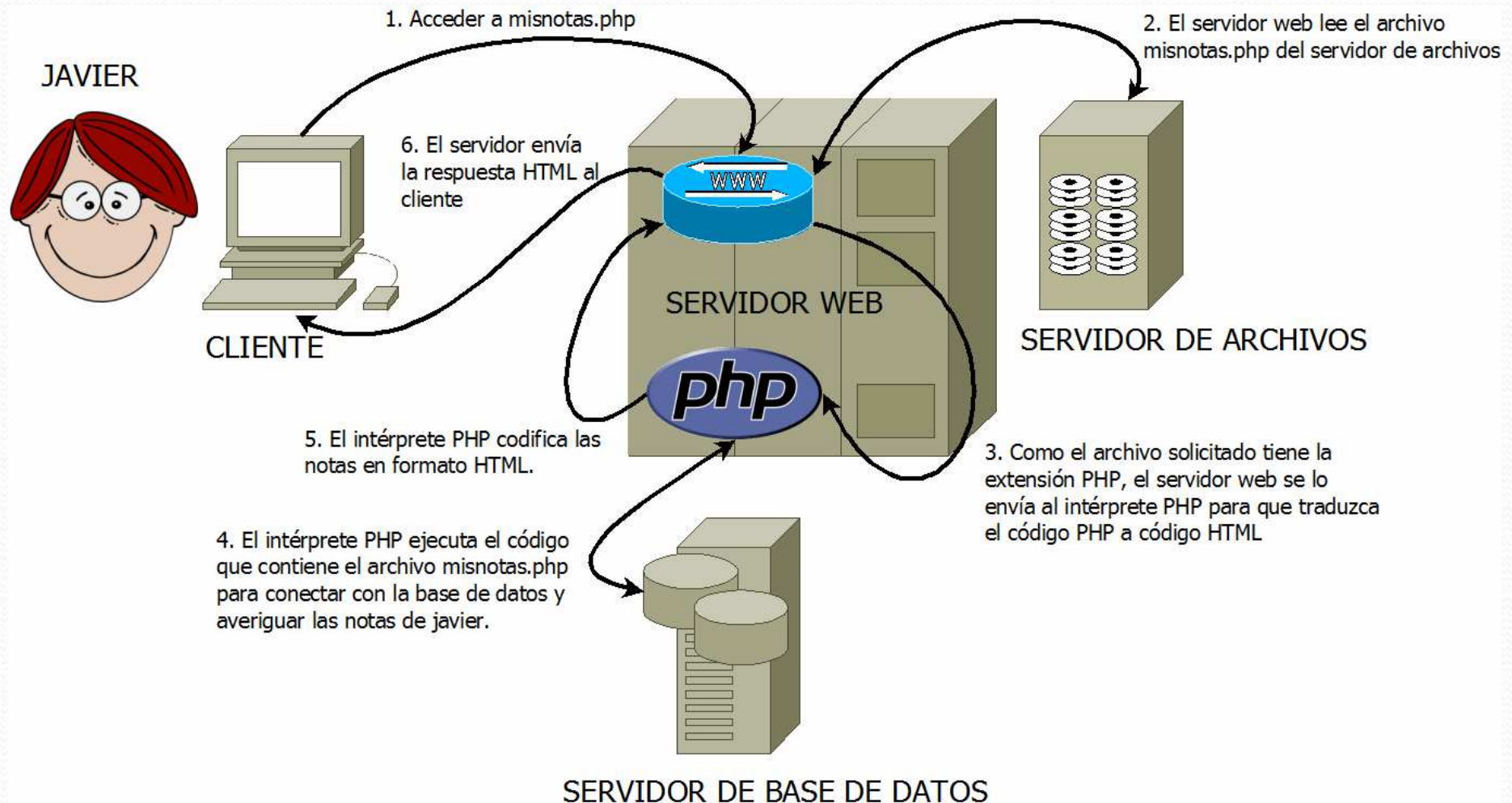
Lenguajes de scripting

❑ JSP (Java Server Pages)

- ❑ porciones de código java intercalado con HTML estático
 - ❑ el código java embebido se denomina servlet
 - ❑ el servlet se carga en la memoria del servidor web al ser ejecutado la primera vez, para mejorar su ejecución en posteriores llamadas a dicha página web.
-

Lenguajes de scripting

El proceso que se realiza a la hora de visitar una página PHP sería:



Aplicaciones CGI y derivados

- ❑ CGI (Common Gateway Interface) es la forma más simple de crear páginas web dinámicas.
 - ❑ Un programa independiente al servidor web devuelve como resultado el contenido que debe visualizar el cliente (la página web resultante) a partir de ciertos parámetros de entrada.
 - ❑ La ubicación del programa a ejecutar se indica en la URL que forma la petición HTTP del cliente.
-

Aplicaciones híbridas

- ❑ Tecnología intermedia entre los lenguajes de scripting y los programas CGI.
 - ❑ Plataforma de Microsoft.Net Framework con el lenguaje ASP.Net.
 - ❑ ASP.Net es una tecnología totalmente orientada a objetos, que permite la creación de páginas web dinámicas utilizando lenguajes como VB.Net, C#, Jscript.Net.
 - ❑ Las páginas ASP.Net están contenidas en ficheros.ASPX que son los que el cliente solicita a través de una URL al servidor.
-

Herramientas de programación

Los instrumentos involucrados en el desarrollo web se pueden clasificar en:

- ☐ Navegadores: Internet Explorer, Google Chrome, Mozilla firefox, etc.
 - ☐ Editores de documentos
 - ☐ Entornos de programación: Eclipse, NetBeans, CodeIgniter, Dreamweaver, FrontPage,...
 - ☐ Herramientas para la creación y administración de bases de datos
 - ☐ Herramientas para el tratamiento de imágenes.
-

Herramientas de programación

El servidor web debe contar con el software necesario para responder a las peticiones del cliente y ejecutar el código embebido dentro del recurso solicitado.

Existen paquetes de libre distribución que incluyen todo el software necesario para el desarrollo web:

- ❑ appServer : www.appservnetwork.com
 - ❑ xampp: www.apachefriends.org
 - ❑ wamp: www.wampserver.com
 - ❑ lamp: www.apachefriends.org
-

Herramientas de programación

Todos los paquetes de instalación de un entorno para el desarrollo de aplicaciones web suelen contener el siguiente software:

- ☐ Apache HTTP Server
- ☐ MySQL
- ☐ PHP
- ☐ PhpMyAdmin

Ejercicio: Investiga qué versión de cada una de estas herramientas se instalarán en el paquete LAMP de Ubuntu 14.04 LTS.

Programa de prueba

Almacenar el siguiente fichero en el directorio:

C:\xampp\htdocs\prueba.php ó
/var/www/html/prueba.php

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

Solicitarlo a través del navegador desde la URL:

<http://localhost/prueba.php>
