



Departamento de Lenguajes y
Sistemas Informáticos



Universitat d'Alacant
Universidad de Alicante

Qué es una aplicación web

Programación en Internet
Curso 2007-2008

Programación en Internet – Curso 2007-2008

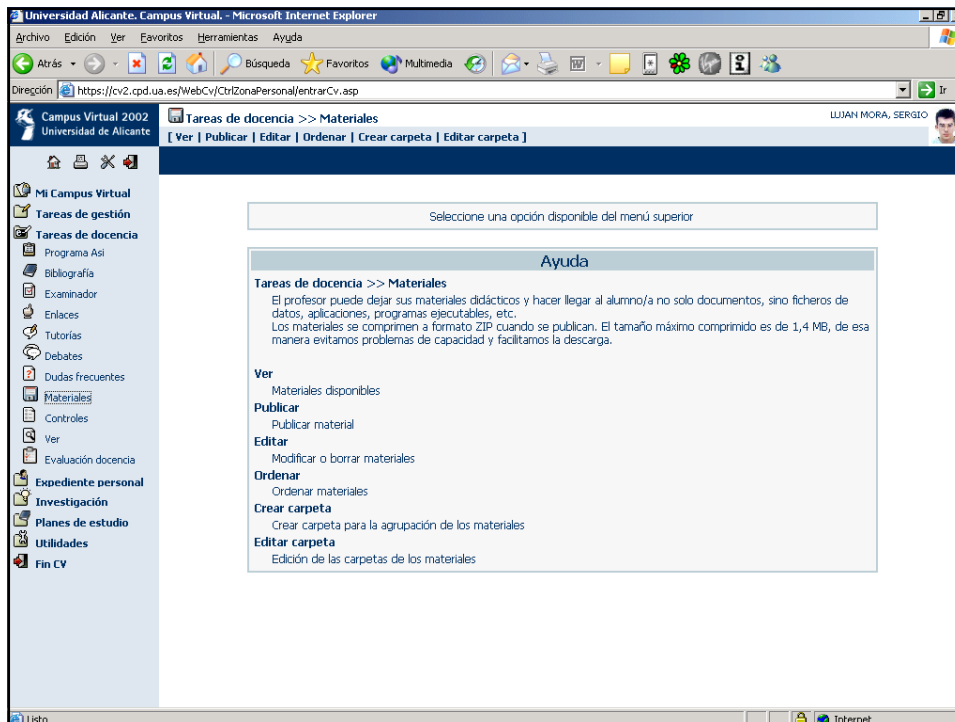
Índice

- Introducción
- Cliente
- Servidor
- Transferencia páginas web
- Ejemplo de aplicación web
- Entornos web
- Ventajas y desventajas
- Arquitectura de una aplicación web
- Metodología de diseño

Programación en Internet – Curso 2007-2008

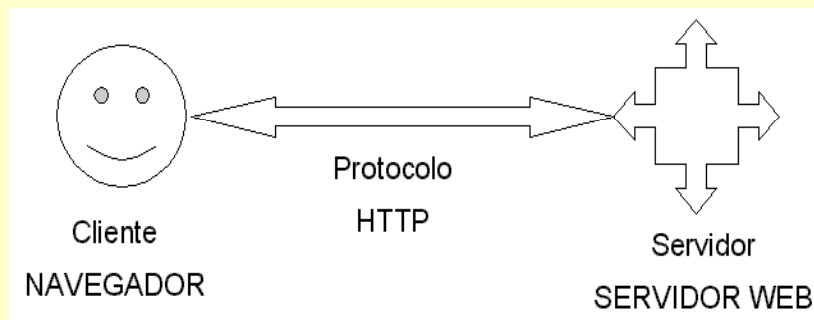
Introducción (1)

- *Web-based (web-enabled) application:*
 - Aplicación informática que se ejecuta en entorno web
 - Aplicación cliente/servidor: el cliente, el servidor y el protocolo ya están definidos (implementados)
 - Cliente: navegador
 - Servidor: servidor web
 - Comunicación: protocolo HTTP



Introducción (y 2)

- Protocolo: *HyperText Transfer Protocol* (HTTP)
- HTTP → TCP/IP: no está orientado a conexión
- Se sitúa en el nivel 7 (aplicación) del OSI-RM (*Open System Interconnection-Reference Model*)



Cliente

- Gestiona las peticiones del usuario y la recepción de las páginas que provienen del servidor
- Interpreta los documentos HTML y sus recursos. Las tecnologías más empleadas son:
 - *HyperText Markup Language* (HTML)
 - *Cascading Style Sheets* (CSS), DHTML
 - Lenguaje de script (JavaScript, VBScript, etc.)
 - ActiveX
 - Applets en Java
 - Plug-ins: Macromedia Flash, Autodesk MapGuide, ...
 - *Virtual Reality Modeling Language* (VRML)

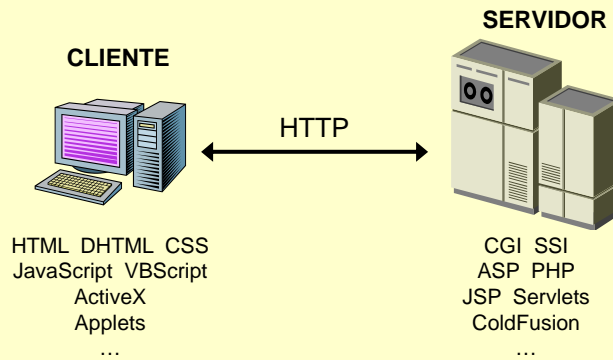
Servidor (1)

- Programa residente que espera peticiones: demonio (*daemon*) en Unix y servicio en servidores de Microsoft
- En la aplicación del servidor hay:
 - Páginas estáticas (documentos HTML)
 - Recursos multimedia (imágenes y documentos adicionales del sitio web)
 - *Scripts* o programas de servidor que al ser invocados se ejecutan y dan como resultado una página HTML generada (pueden acceder a una BD)

Servidor (y 2)

- Tecnologías de servidor:
 - CGI: complejo y poco eficiente
 - SSI: estándar de “macros” de servidor web
 - ASP (Microsoft): Windows, Windows NT
 - JSP y Servlets (Sun Microsystems): Windows, algunos Unix
 - PHP (PHP.net): código fuente, binarios para Win32 y algunos Unix
 - ColdFusion (Macromedia/allaire): Windows, Windows NT, Linux, Solaris, HP-UX
- Las más modernas tienen prestaciones que facilitan el desarrollo de aplicaciones

Cliente/Servidor



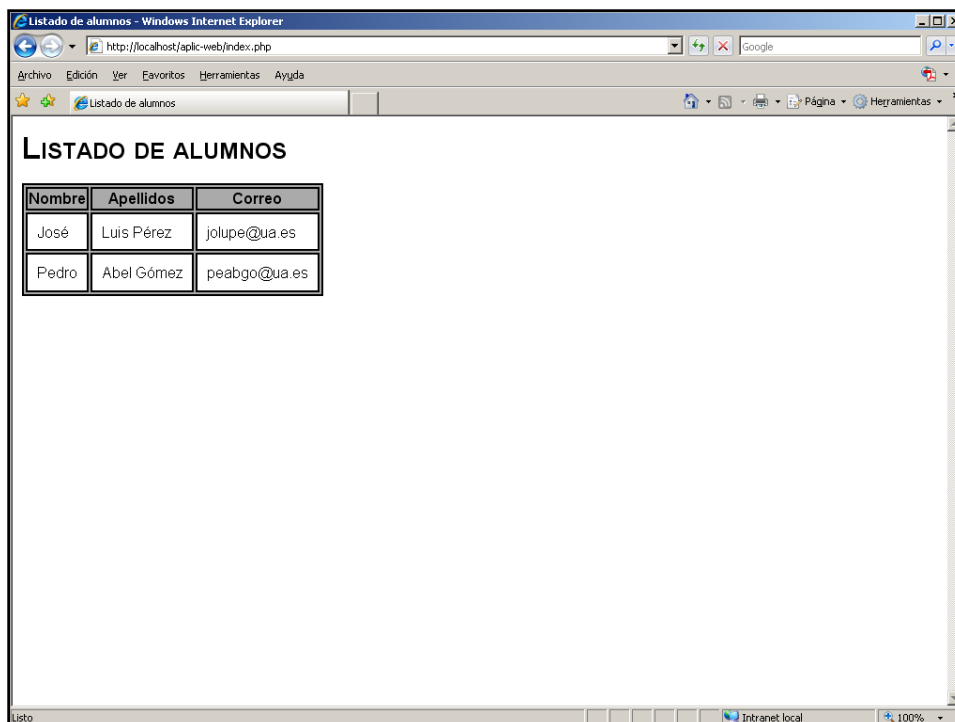
Transferencia páginas web

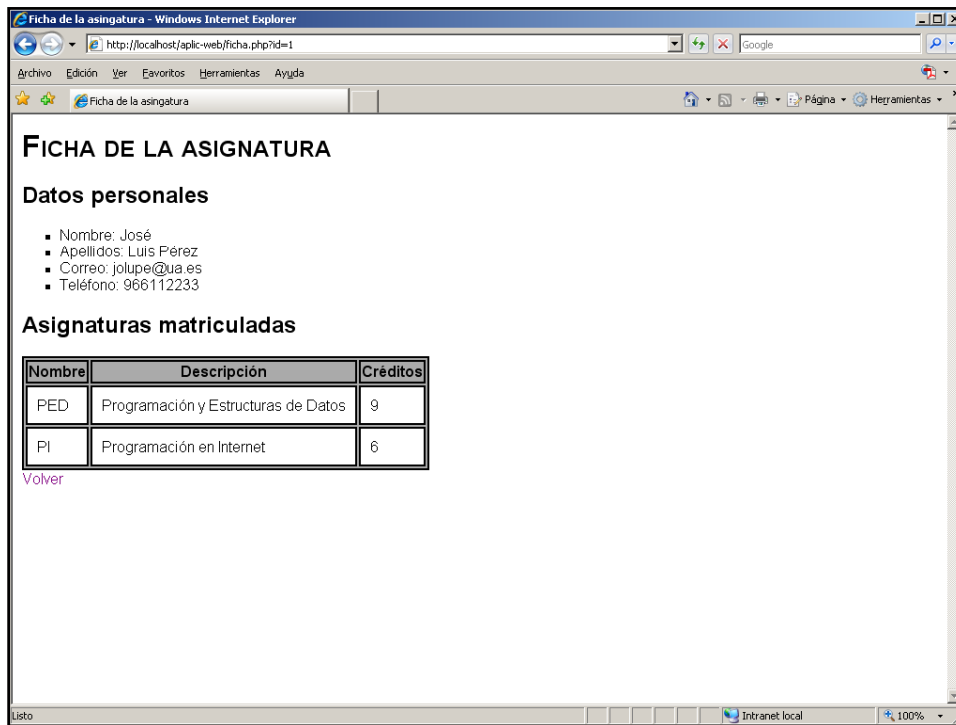
- Envío de la URL
- Apertura de la conexión con el servidor
 - Siempre se abre una conexión nueva con cada petición del cliente (excepto con HTTP 1.1 que permite *Keep-Alive Connections*)
- Solicitud de la página o recurso
- Envío (o código de error) por parte del servidor
- Interpretación del documento HTML y petición de otros objetos a los que hace referencia
- Cierre conexión una vez que hayan llegado todos los recursos y documentos

Ejemplo de aplicación web

- Aplicación web sencilla basa en PHP
- ¿Qué es PHP?

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.





```

Programación en Internet – Curso 2007-2008

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Listado de alumnos</title>
<link rel="stylesheet" type="text/css" href="estilos.css" />
</head>
<body>
<h1>Listado de alumnos</h1>
<table>
<tr>
<th>Nombre</th><th>Apellidos</th><th>Correo</th>
</tr>

```

Programación en Internet – Curso 2007-2008

```
<?
$bd = "aplic-web";

// resource odbc_connect ( string $dsn, string $user, string $password [, int $cursor_type] )
$con = odbc_connect($bd, "", "");

// resource odbc_exec ( resource $connection_id, string $query_string [, int $flags] )
$result = odbc_exec($con, "Select * From Alumnos");

// array odbc_fetch_array ( resource $result [, int $rownumber] )
while(($fila = odbc_fetch_array($result)))
{
    $id = $fila["IdAlumnos"];
    echo "<tr onclick=\"window.location.href=ficha.php?id=$id;\">";
    echo "<td>" . $fila["Nombre"] . "</td>";
    echo "<td>" . $fila["Apellidos"] . "</td>";
    echo "<td>" . $fila["Correo"] . "</td>";
    echo "</tr>\n";
}
?>
</table></body></html>
```

Programación en Internet – Curso 2007-2008

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ficha de la asignatura</title>
<link rel="stylesheet" type="text/css" href="estilos.css" />
</head>
<body>
<h1>Ficha de la asignatura</h1>
<h2>Datos personales</h2>
```


Programación en Internet – Curso 2007-2008

```

</ul>
<?
    $bd = "aplic-web";

    // resource odbc_connect ( string $dsn, string $user, string $password [, int $cursor_type] )
    $con = odbc_connect($bd, "", "");

    // resource odbc_exec ( resource $connection_id, string $query_string [, int $flags] )
    $result = odbc_exec($con, "Select * From Alumnos Where IdAlumnos = " . $_GET["id"]);

    // array odbc_fetch_array ( resource $result [, int $rownumber] )
    while(($fila = odbc_fetch_array($result)))
    {
        echo "<li>Nombre: " . $fila["Nombre"] . "</li>";
        echo "<li>Apellidos: " . $fila["Apellidos"] . "</li>";
        echo "<li>Correo: " . $fila["Correo"] . "</li>";
        echo "<li>Teléfono: " . $fila["Telefono"] . "</li>";
    }
?>
</ul>

```

Programación en Internet – Curso 2007-2008

```

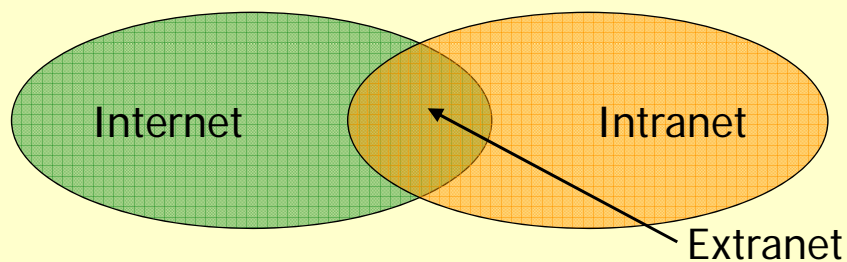
<h2>Asignaturas matriculadas</h2>
<table>
<tr>
<th>Nombre</th><th>Descripción</th><th>Créditos</th>
</tr>
<?
    // resource odbc_exec ( resource $connection_id, string $query_string [, int $flags] )
    $result = odbc_exec($con, "Select * From Asignaturas, Matriculas Where Asignatura =
        IdAsignaturas And Alumno = " . $_GET["id"]);

    // array odbc_fetch_array ( resource $result [, int $rownumber] )
    while(($fila = odbc_fetch_array($result)))
    {
        echo "<tr>";
        echo "<td>" . $fila["Nombre"] . "</td>";
        echo "<td>" . $fila["Descripcion"] . "</td>";
        echo "<td>" . $fila["Creditos"] . "</td>";
        echo "</tr>\n";
    }
?>
</table>
<div><a href="index.php">Volver</a></div></body></html>

```

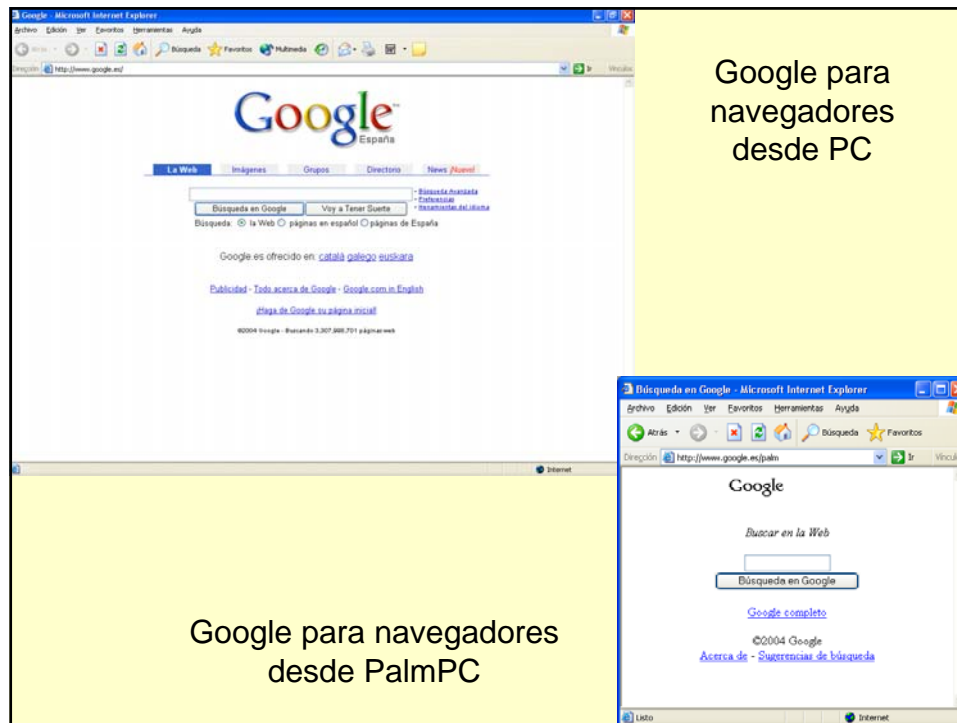
Entornos web (1)

- Internet: aplicación abierta y accesible desde Internet
- Intranet: la aplicación se ejecuta en una red local con TCP/IP y servicios de Internet
- Extranet: aplicación accesible desde Internet pero con restricciones (seguridad, VPN, etc.)



Entornos web (y 2)

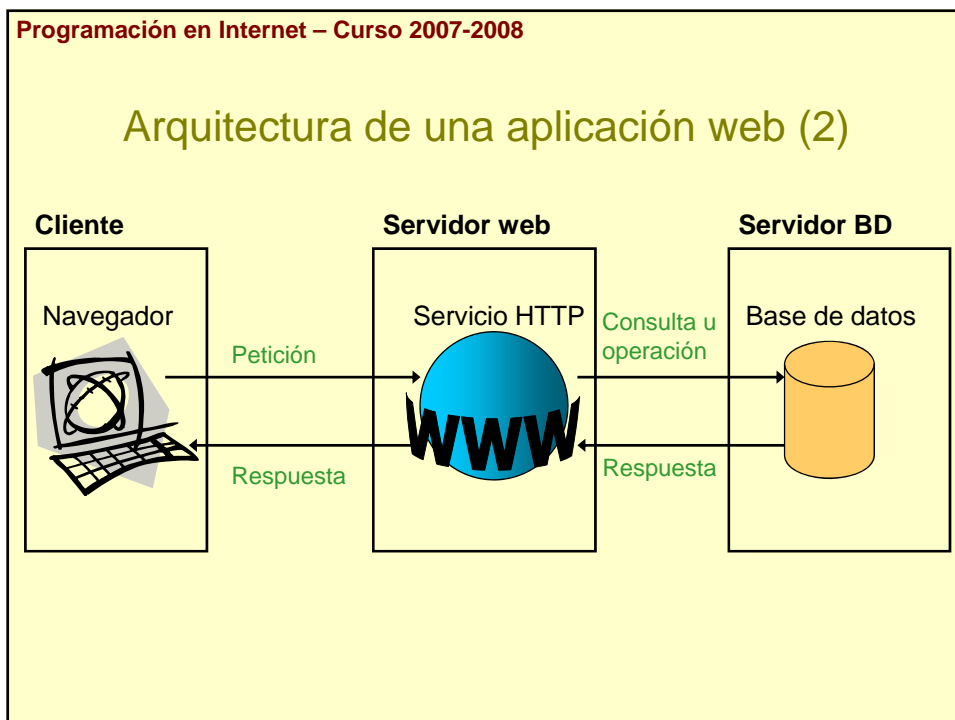
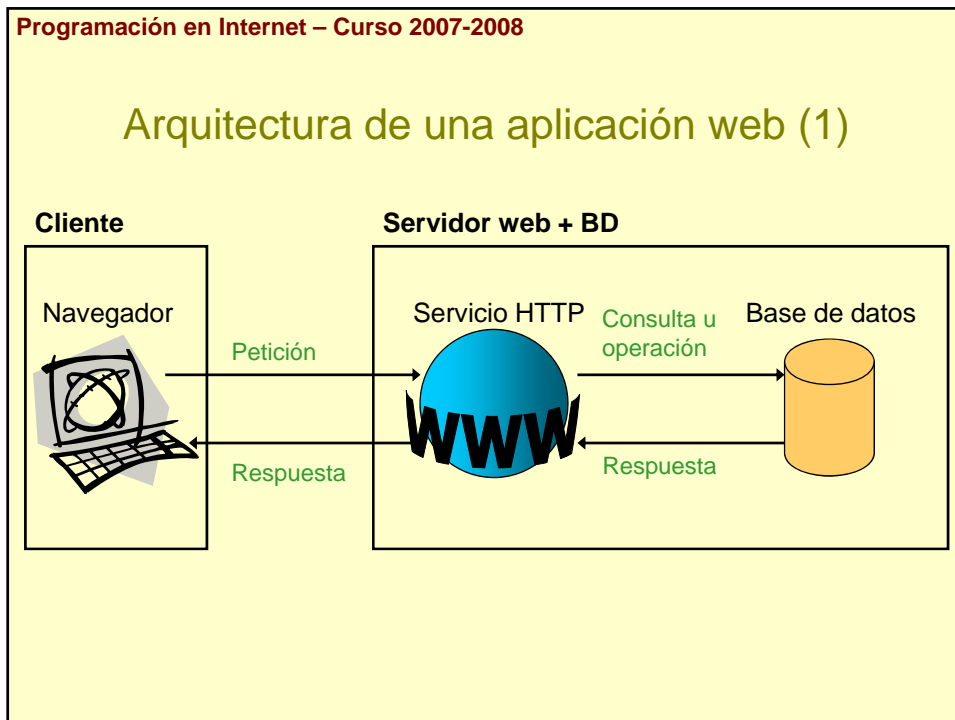
- Aplicaciones web:
 - *Multi-channel*: el usuario puede trabajar con la misma aplicación a través de distintos canales (ordenador, PDA, teléfono móvil, web TV, etc.)
 - Cada plataforma o tecnología es un canal de comunicación con la aplicación web
 - *Cross-channel*: los efectos producidos en una aplicación a través de un canal pueden ser percibidos por el usuario a través de otros canales

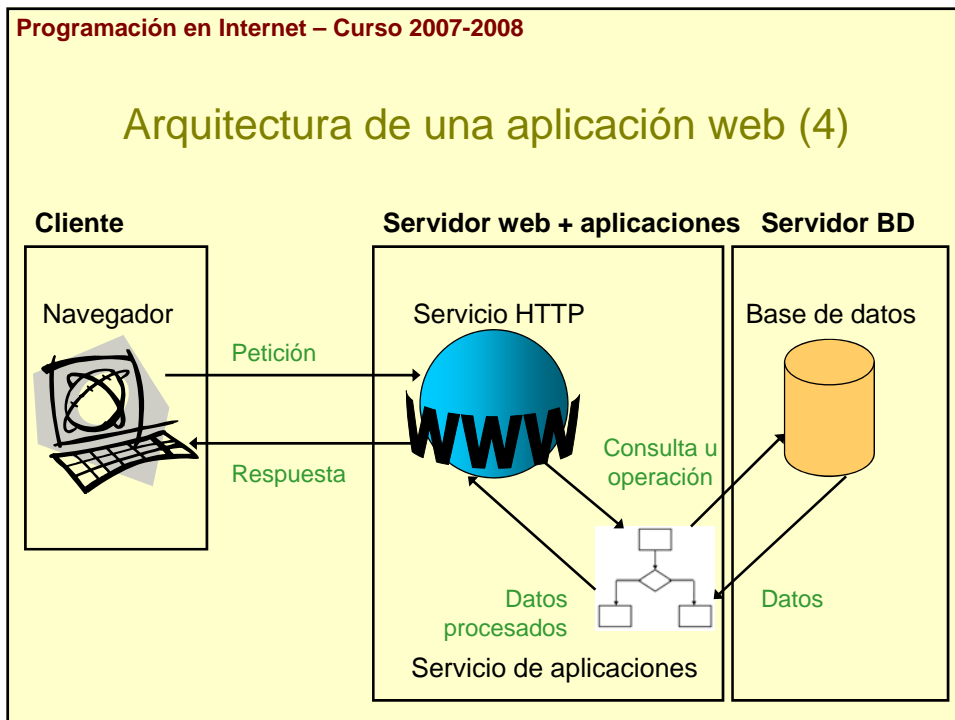
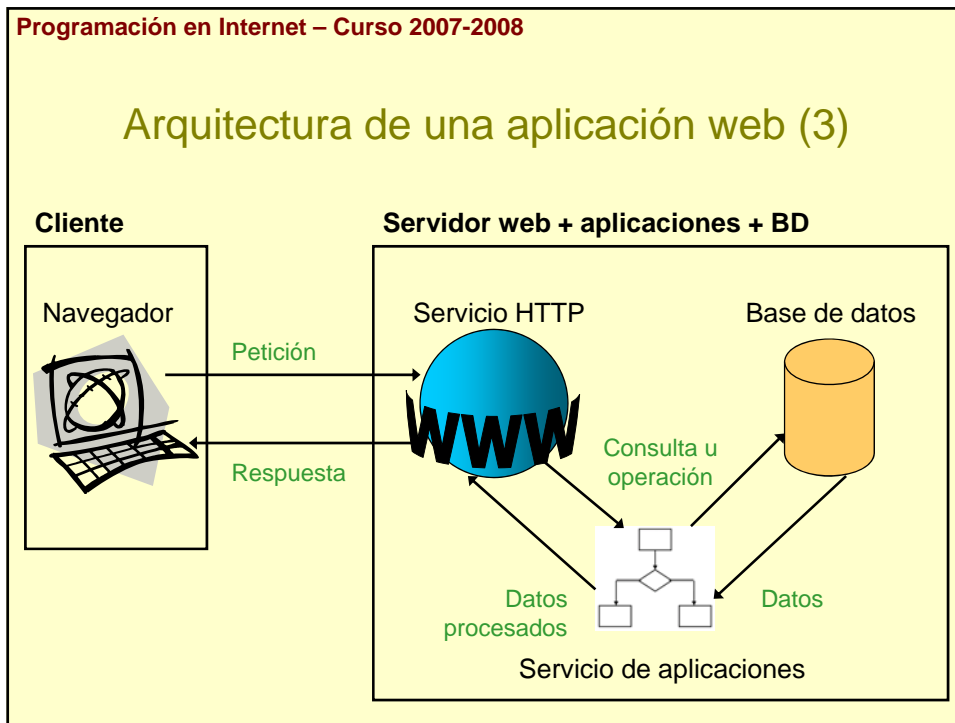


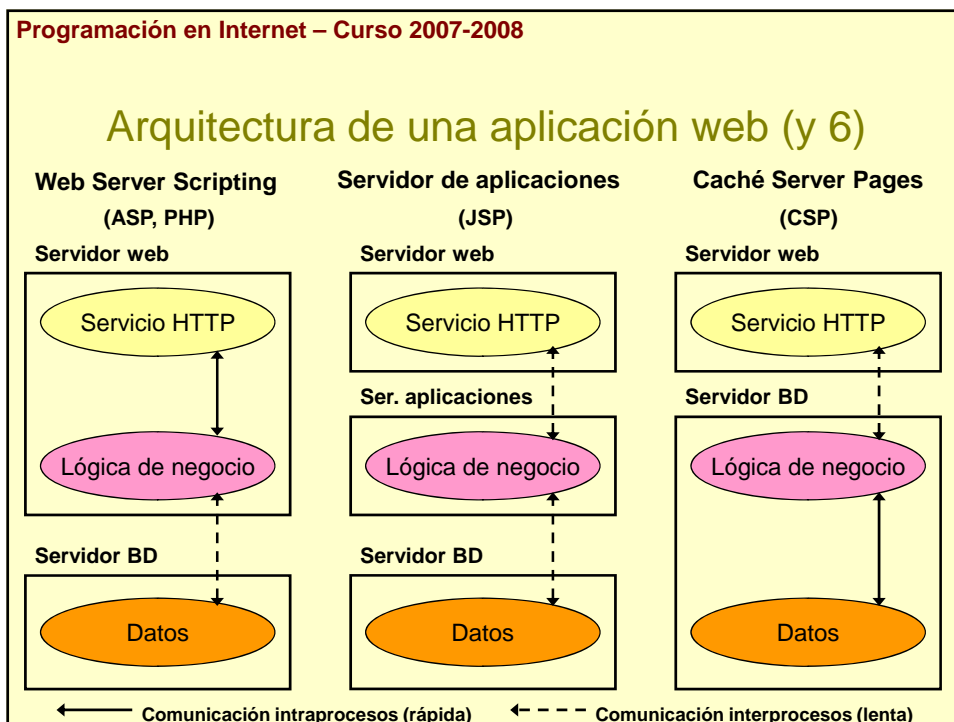
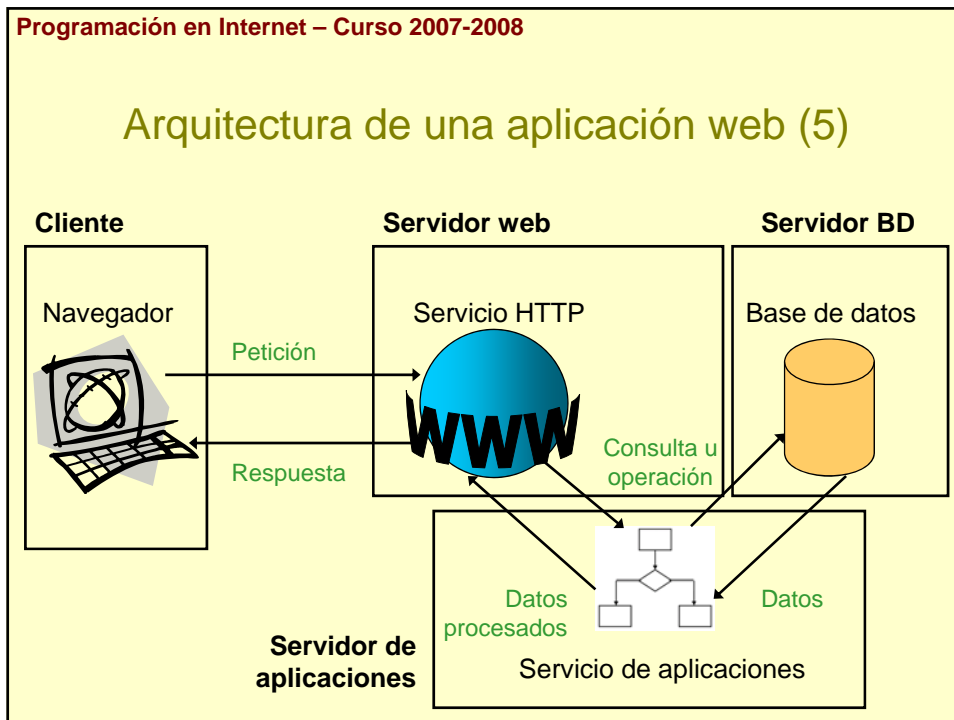
Programación en Internet – Curso 2007-2008

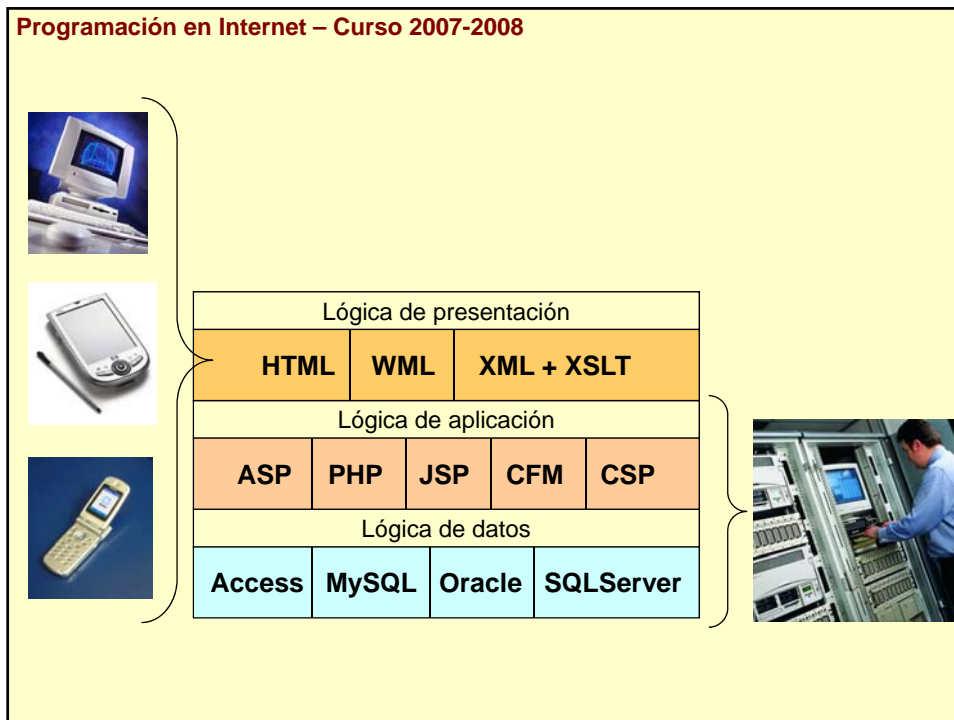
Ventajas y desventajas

- Estandarización: TCP/IP, HTTP, HTML, ...
- Teletrabajo y movilidad: extranets
- Reducción coste instalación y actualización clientes, gestión de versiones
- Entorno del cliente controlado: navegador, versión, fabricante, etc.
- Integración Internet e intranet (aprendizaje)
- Independencia de plataforma (GUI)
- Desventaja: versatilidad y potencia (limitaciones de HTML) y necesidad de buenas comunicaciones









- Programación en Internet – Curso 2007-2008**
- ## Metodología de diseño (1)
1. Estudio de requisitos: contenido, objetivo, etc.
 2. Elección de la tecnología, arquitectura del sitio web, proveedor del servidor y motor de la base de datos
 3. Diseño de la estructura lógica o mapa de navegación
 4. Diseño de la estructura física
 5. Creación de los contenidos
 6. Diseño gráfico y ergonómico: colores, montaje, tipografía, etc.

Programación en Internet – Curso 2007-2008

Metodología de diseño (y 2)

7. Diseño de las páginas estáticas y elementos multimedia
8. Desarrollo de los *scripts* y páginas dinámicas
9. Verificación y pruebas: versiones, accesos, navegación, pruebas de carga, etc.
10. Puesta en marcha