

# Introducción al desarrollo web dinámico

Desarrollo web en entorno servidor

# Contenidos

- Qué es una aplicación web y su arquitectura
- Páginas estáticas y páginas dinámicas
- Lenguaje de script de cliente y lenguaje de script de servidor
- Historia de PHP
- Software necesario para desarrollar una aplicación web PHP
- Instalación XAMPP y NetBeans
- Despliegue de una aplicación web PHP y ejecución

# ¿Qué es una aplicación web?

- **Aplicación web** – En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.
- En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.
- Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web e instalar software a miles de usuarios potenciales.

# Aplicación web. Estructura de las APW.

- Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres-capas.
- En su forma más común, el navegador web ofrece la primera capa, y un motor capaz de usar alguna tecnología web dinámica, por ejemplo: PHP, Java Servlets o ASP, ASP.NET, CGI, ColdFusion, embPerl, Python o Ruby on Rails que constituye la capa intermedia. Por último, una base de datos constituye la tercera y última capa.
- El navegador web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario.

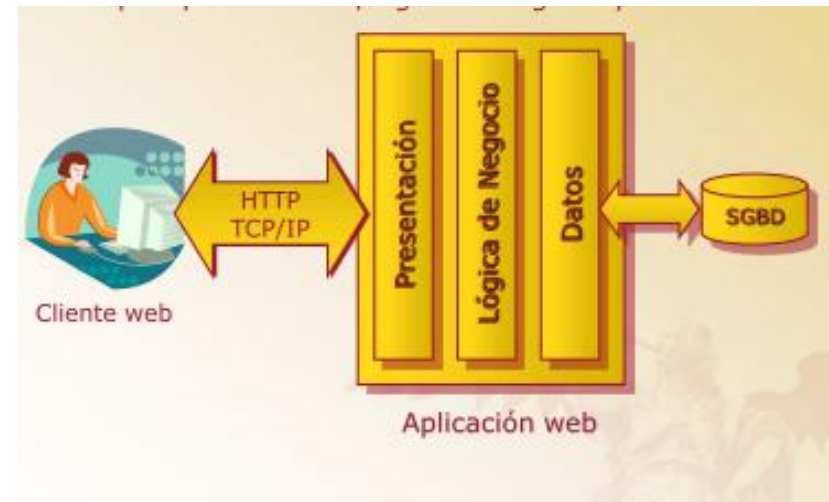
# Estructura de una aplicación web.

- Aunque con variantes, una aplicación web (en la parte del servidor) está normalmente estructurada como una aplicación de tres/n-capas

- *capa de presentación*

- genera la interfaz de usuario con la información proporcionada por la capa de lógica de negocio

- *capa de lógica de negocio* – incluye alguna tecnología web dinámica (PHP, Java servlets, ASP.NET, ...)



- implementa la lógica de la aplicación como respuesta a las peticiones del usuario. Accede a la capa de datos

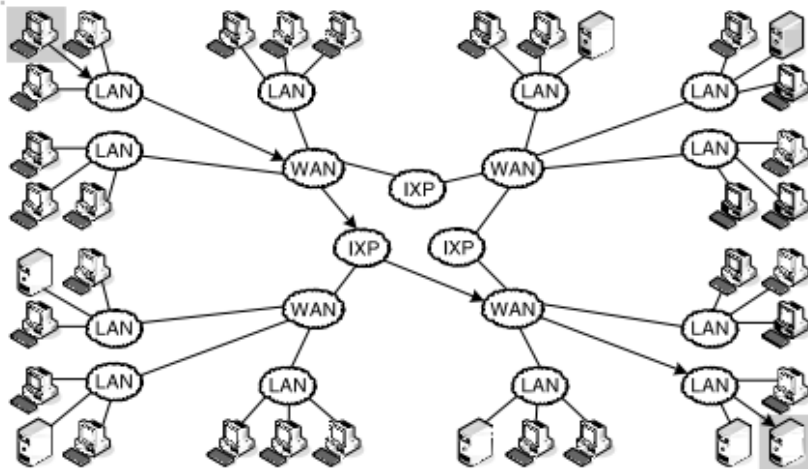
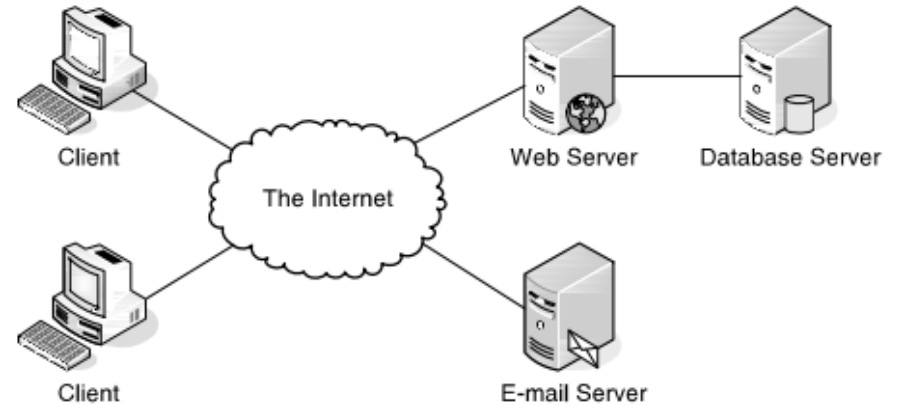
- *capa de datos* – permite el acceso a los datos a través de un SGBD

# Aplicación web. Estructura de las APW.

- Una aplicación web si funciona siguiendo un modelo *cliente - servidor*
  - el servidor comparte recursos (ficheros, páginas web, bases de datos, email, ...)
  - el servidor web es un servidor que comparte sitios web
  - el navegador web es el cliente software utilizado para acceder al servidor web
  - la comunicación entre los clientes y servidores web se realiza a través de la red Internet (www) y el protocolo HTTP

# Aplicación web. Modelo cliente-servidor.

Arquitectura cliente/servidor de una aplicación web



- LAN – red de área local – conecta ordenadores en un espacio reducido
- WAN – redes de área amplia ( a través de routers de conectan diferentes LAN)
- Internet – múltiples redes WAN conectadas

Arquitectura de Internet

# Servidores y clientes.

- **Servidor** — nos referimos a un host (ordenador remoto) que
  - ofrece uno o varios servicios (páginas web – www, correo, .....)
  - para atender a cada servicio necesita el software (programa) adecuado (software para servir páginas web, software para servir correo, software servidor de BD, ...)
  - ejemplos – servidor de páginas web (Apache, IIS...)
- **Cliente** - el software que
  - realiza peticiones de un servicio a un servidor
  - ejemplos - cliente que solicita página web – el navegador (Mozilla Firefox), cliente de correo (Outlook Express) , cliente FTP (FileZilla), ... Existen varios tipos de clientes, dependiendo de la cantidad de tareas que realice el cliente en comparación con el servidor.

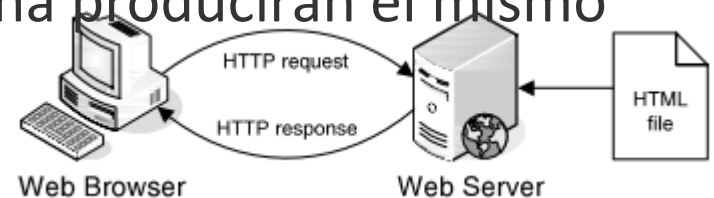


# Servidores y clientes.

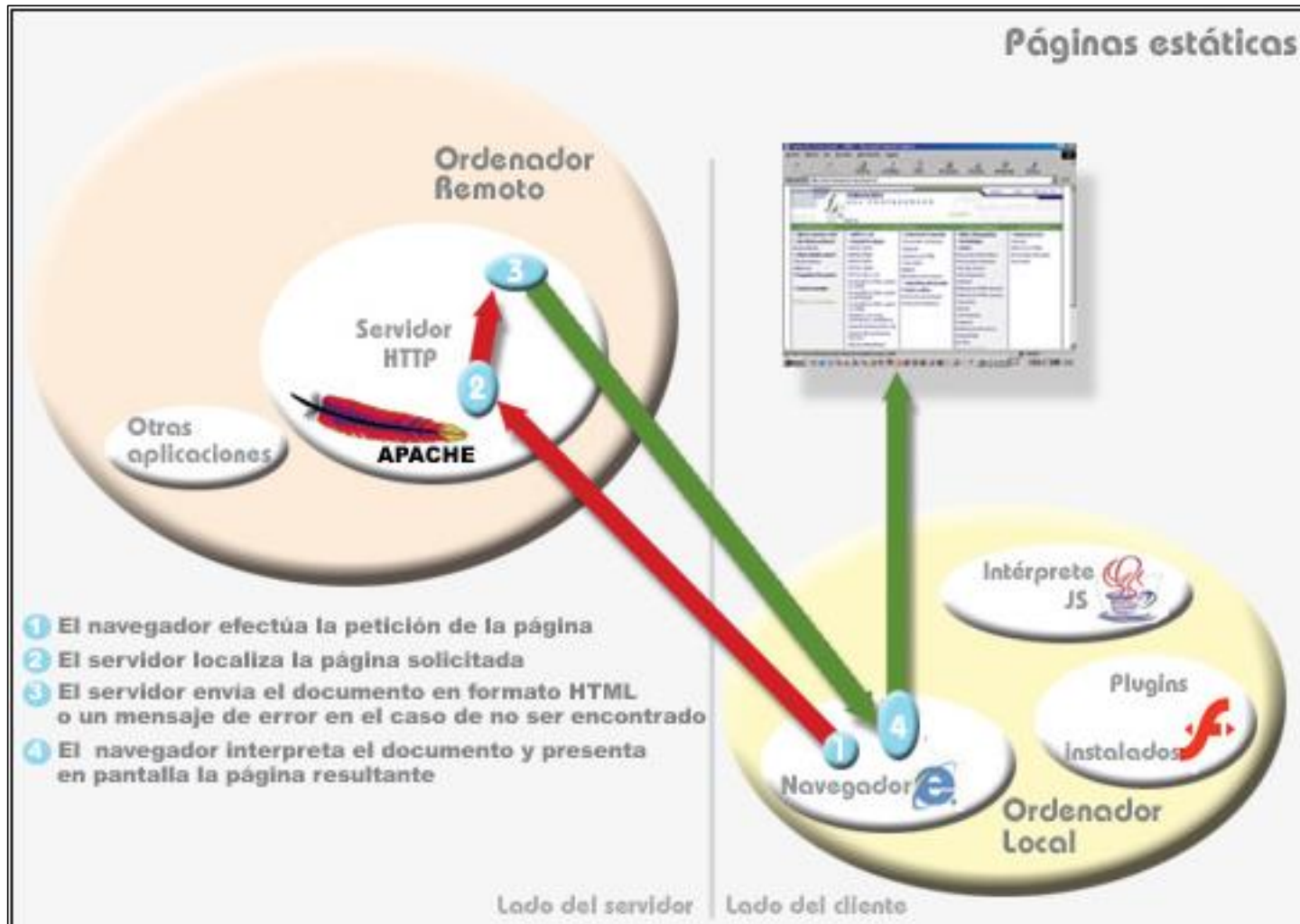
- Situación habitual
  - software de cliente instalado en una máquina (nuestra máquina local)
  - software de servidor servidor instalado en una máquina distinta
  - comunicación a través de Internet
- Servidor y cliente pueden estar en la misma máquina
  - servidor es *localhost* (127.0.0.1)

# Contenido estático y dinámico

- **Página web estática** – documento HTML (.html) cuyo contenido no cambia en respuesta a las entradas del usuario.
  - cuando se solicita desde un cliente una página web estática se introduce su URL en el navegador
  - el navegador envía una petición HTTP al servidor
  - el servidor, una vez localizada la página, enviará una respuesta HTTP que incluye la página al navegador
  - éste interpreta el código HTML y lo visualiza
  - sucesivas peticiones de la misma página producirán el mismo resultado

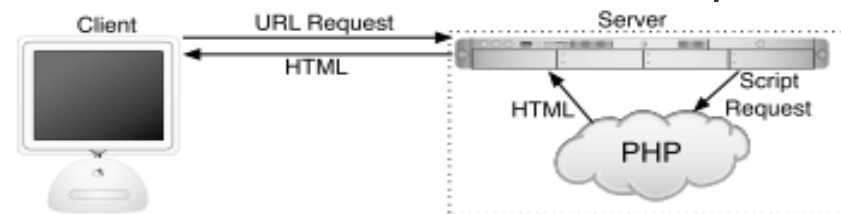


# Contenido estático y dinámico



# Contenido estático y dinámico

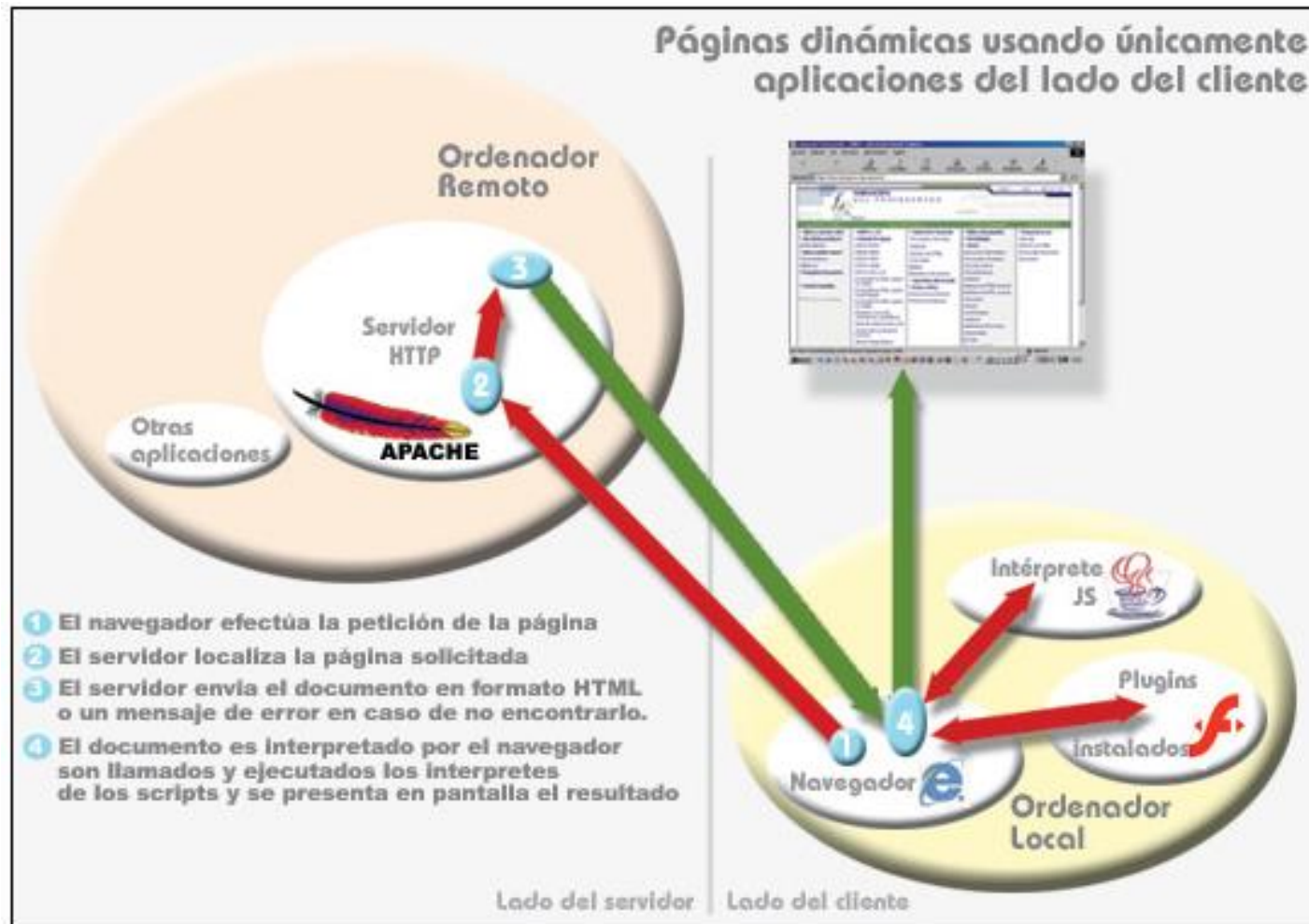
- **Página web dinámica** – página web que se genera por un script o programa. La página cambia de acuerdo a la información que ha sido enviada por el navegador
- Las páginas web dinámicas entran en una de estas dos categorías:
  - *procesadas en el lado del servidor* (por un script o lenguaje de servidor – script PHP)
    - el usuario introduce información (habitualmente un formulario)
    - la información se transmite al servidor web a través de una petición HTTP



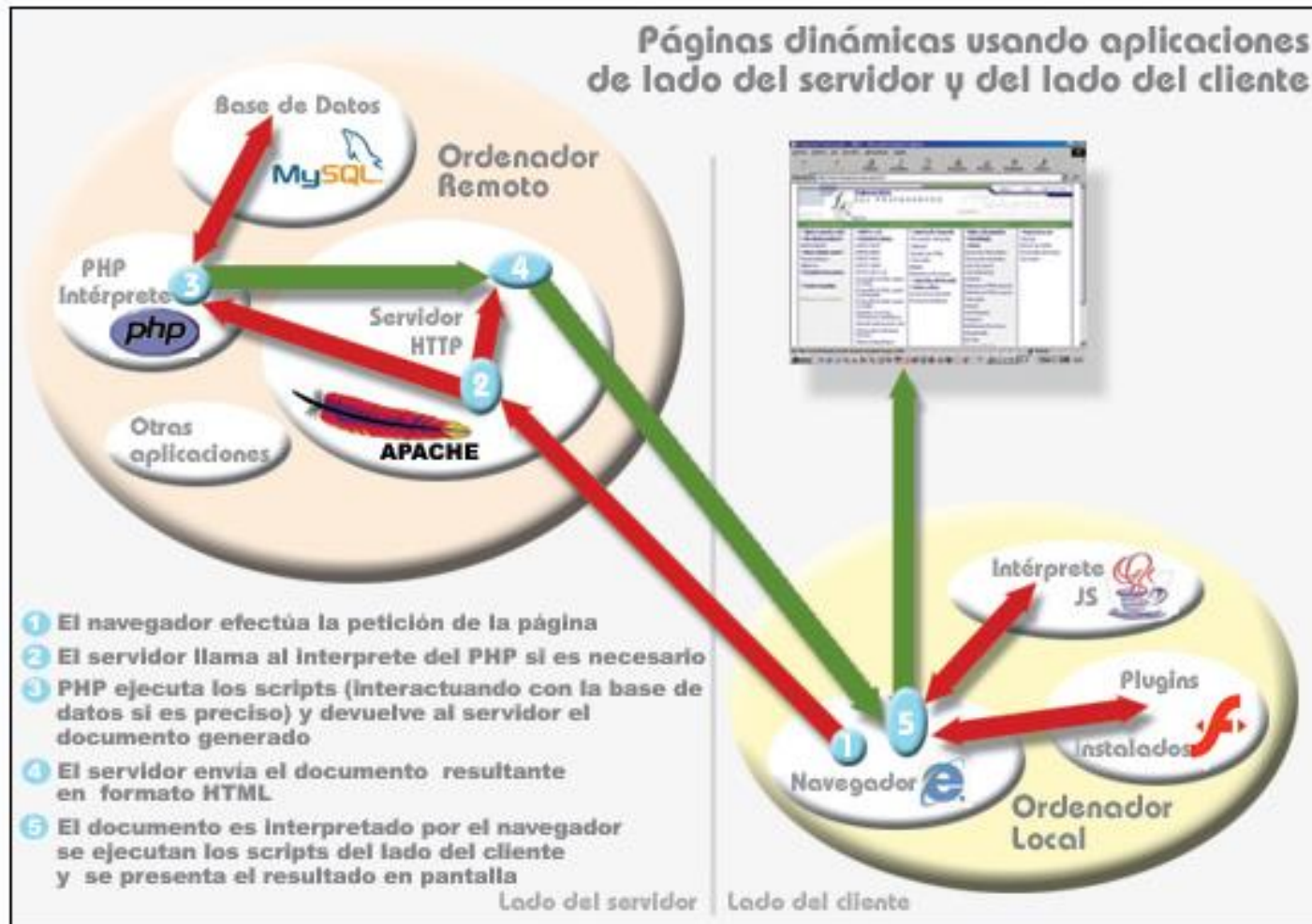
# Contenido estático y dinámico

- el servidor ejecuta un programa basado en la información enviada junto con la petición y genera una página HTML
- esta página HTML se envía al navegador para que la visualice
- *procesadas en el lado del cliente* (por un script o lenguaje de cliente – script JavaScript , tecnología AJAX.....)
  - el navegador solicita una página HTML al servidor
  - éste la envía como respuesta HTTP
  - la página incluye un programa que se ejecuta cuando el navegador visualice la pagina.
  - en el lado del cliente (navegador) se produce la interacción con el usuario. Parte de la página se ve modificada.

# Contenido estático y dinámico



# Contenido estático y dinámico



# Contenido estático y dinámico

- El proceso en el lado del cliente es más rápido (las entradas del usuario y las respuestas no se envían al servidor), reduce la carga del servidor web
- Hoy por hoy, la mayoría de aplicaciones web requieren, sin embargo, proceso en el servidor para poder acceder a una BD
- Muchas aplicaciones web combinan el proceso del lado del cliente y el proceso del lado del servidor



# Lenguajes de script de cliente

- Un *script* es un conjunto de instrucciones interpretadas por un programa
- Un *lenguaje de script* es un lenguaje de programación para escribir scripts. El código de un script se inserta dentro de un documento HTML.
- Un *script de cliente* se ejecuta en el navegador del cliente al cargar la página, o cuando sucede algo especial como puede ser el pulsar sobre un enlace.
- Lenguaje de script más utilizados: JavaScript, applets de Java, Flash.

# Lenguajes de script de cliente

```
<html>
  <body>

    <script type="text/javascript">
      var nombre;
      nombre="Pepe";
      document.write(nombre);
      document.write("<br />");
      nombre="Juan";
      document.write(nombre);
    </script>

    <p>Ejemplo de script de cliente.</p>

  </body>
</html>
```

Código JavaScript, interpretado  
por el navegador  
(el intérprete que  
incluye el navegador)

Pepe  
Juan  
  
Ejemplo de script de cliente.

# Lenguajes de script de cliente

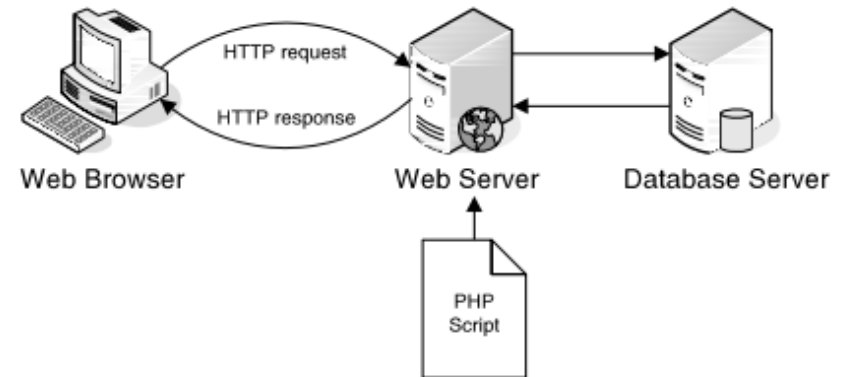
- ¿Qué se puede hacer con un script de cliente?
  - Validar formularios antes de enviarlos al servidor, reaccionar ante ciertos eventos (al pasar el ratón por un elemento de la página que cambie de color, cargar una página al hacer click en un botón, ), obtener información sobre el tipo del navegador, efectuar cálculos, crear animaciones, almacenar ...
- Limitaciones de los scripts de cliente
  - problemas con la compatibilidad de los navegadores, visibilidad del código, no acceso a una base de datos, ...

# Lenguajes de script de servidor

- Permiten escribir scripts de servidor. Un *script de servidor* se ejecuta en el servidor web
- Lenguajes de servidor más utilizados: PHP (.php), JSP (.jsp), ASP.NET (.aspx) , Perl (.pl), Python (.py)
- Cuando un servidor web recibe una petición para generar una página web dinámica:
  - utiliza la extensión del fichero solicitado para determinar el programa (script) que ha de ejecutar
  - si la extensión es **.php** el servidor llama al *intérprete* PHP (que está ejecutándose en el servidor web) para que ejecute el script

# Lenguajes de script de servidor

- el script puede utilizar los datos recibidos desde el cliente para,
- acceder al servidor de BD y recoger datos que utilizará para generar la página HTML
- cuando el intérprete PHP la ha generado se la pasa al servidor web que,
- la envía como respuesta al navegador para que la visualice



# Lenguajes de script de servidor

```
<html>
  <body>

    <h4>Ejemplo de script en el lado del cliente</h4>

    <script type="text/javascript">
      document.write("<p>" + Date() + "</p>");
    </script>

    <h4>Ejemplo de script en el lado del servidor</h4>
    <?php
      echo "Fecha actual ".date("m.d.y");
    ?>
  </body>
</html>
```

Código PHP, ejecutado por el intérprete del lenguaje PHP que está en el servidor

**Ejemplo de script en el lado del cliente**

Tue Aug 30 2011 20:58:54 GMT+0200 (Hora de verano romance)

**Ejemplo de script en el lado del servidor**

Fecha actual 08.30.11

# Lenguajes de script de servidor

- Ventajas de los scripts de servidor
  - cómo se ejecutan en el servidor no dependen de ningún tipo de navegador
  - permiten el acceso a BD, modificar ficheros del servidor, ...
  - código no visible para los usuarios
- Desventajas
  - más lenta su ejecución que la del lado del cliente

# URL y protocolo HTTP

- **URL – Uniform Resource Locator** – dirección para identificar de forma inequívoca a un recurso en la web (**URI**)
  - La dirección que escribimos en la barra de direcciones de un navegador cuando queremos solicitar una página  
<http://www.w3schools.com/html/default.asp>
- **HTTP** – el protocolo utilizado para la comunicación entre un servidor web y un cliente web (el navegador)
  - Proporciona un mecanismo básico de *petición/respuesta*. El cliente web envía una petición HTTP solicitando un recurso, el servidor web envía una respuesta HTTP devolviendo el recurso



# Software que necesitaremos para desarrollar una aplicación web PHP

- 4 componentes software para desarrollar nuestra aplicación web PHP
  - el **navegador** (nuestro cliente web) – Firefox, Chrome, Explorer, Safari...
  - el **servidor web** -
    - **Apache** – open source – puede ejecutarse en la mayoría de los SO. Soporta muchos lenguajes de script de servidor y puede interactuar con muchos servidores de bases de datos
  - **lenguaje de servidor PHP** - (.php)
    - Necesitaremos el **intérprete PHP** para ejecutar los scripts
  - el **servidor de bases de datos**
    - **MySQL** – servidor de BD open-source disponible para la mayoría de SO

# Desarrollar una aplicación web PHP – IDE (entorno de desarrollo)

- Además necesitaremos un editor para nuestras páginas PHP
  - utilizaremos NetBeans que es un conocido IDE para desarrollar en Java, C/C++, Ruby y también en PHP
    - desarrollado en Java
    - fácilmente extensible
    - open source
  - con NetBeans podremos
    - editar,
    - ejecutar y
    - depurar el código



# El lenguaje PHP

- PHP – acrónimo de PHP: Hypertext PreProcessor
  - lenguaje para crear scripts interpretados en el servidor , creando contenido que se envía al cliente
  - Tecnología *open source*
  - Típicamente utilizado con el servidor web Apache aunque disponible también para IIS
  - extensión **.php**

# El lenguaje PHP. Un poco de historia.

Año	Versión	Características
1995	1.0	Originalmente era un conjunto de scripts en Perl muy simples creados por Rasmus Lerdorf Personal Home Page
1998	3.0	Primera versión parecida al PHP actual, Versión renombrada, PHP: Hypertext Preprocessor rescritura completa por parte de Andi Gutmans y Zeev Suraski. Incluía una sintaxis mucho más fuerte y soporte mínimo para OO.
2000	4.0	Introdujo mejoras de eficiencia, modularidad y soporte para más bases de datos y librerías de terceros (Zend Engine)
2005	5.0	Nuevo modelo de objetos, extensión PDO, mejoras de eficiencia

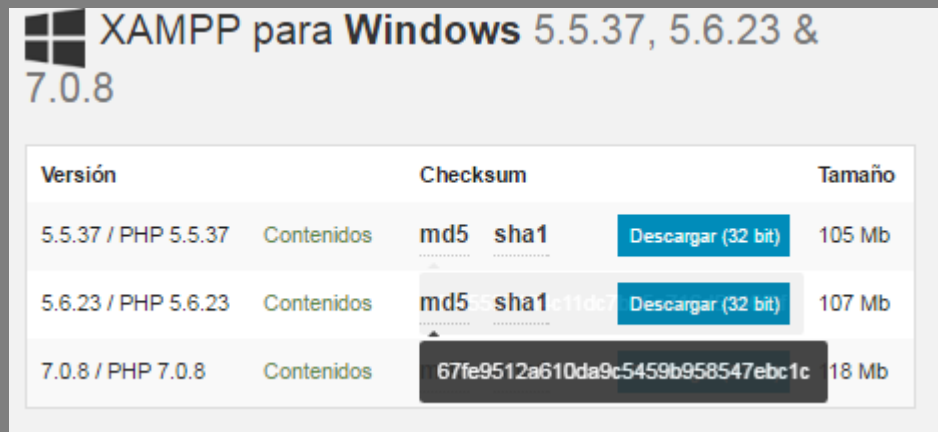
# Instalación XAMPP y NetBeans

# ¿Qué es XAMPP?

- Paquete de software gratuito, open source y multiplataforma que simplifica la instalación del Apache, MySQL y PHP
- Incluye:
  - Servidor web Apache
  - Servidor de bases de datos MySQL
  - Intérprete de PHP
  - Intérprete de Perl

# Descarga de XAMPP para Windows

- <https://www.apachefriends.org/es/download.html>
- Descargamos XAMPP



Versión	Checksum	Tamaño
5.5.37 / PHP 5.5.37	Contenidos md5 sha1 .....	105 Mb
5.6.23 / PHP 5.6.23	Contenidos md5 sha1c11dc7 .....	107 Mb
7.0.8 / PHP 7.0.8	Contenidos 67fe9512a610da9c5459b958547ebc1c	18 Mb

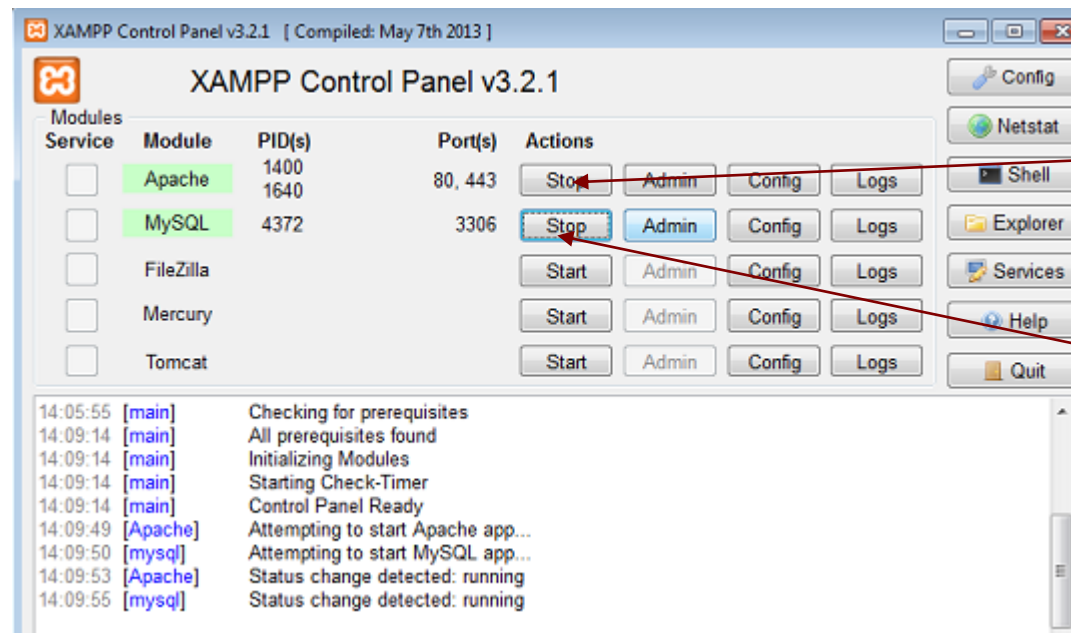
# Instalación de xampp para Windows

- Ejecutar el fichero
- Instalar xampp en la carpeta que aparece por defecto [c:\xampp](#)
- Para crear accesos directos en el escritorio y en el menú Inicio dejar marcadas las casillas de verificación.
- Nos permite marcar si queremos que al arrancar el SO se inicien los servicios. Los dejamos sin marcar.
- Terminar.

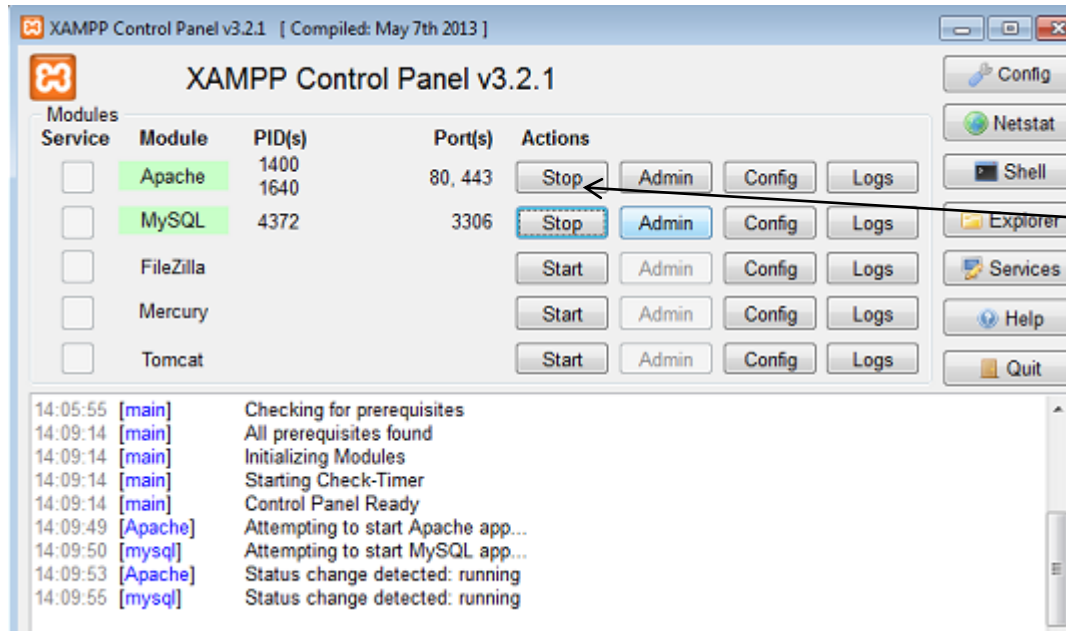


# Arrancar los servicios Apache y MySQL desde xampp

- Doble click en el icono de XAMPP del escritorio par abrir el Panel de control de XAMPP



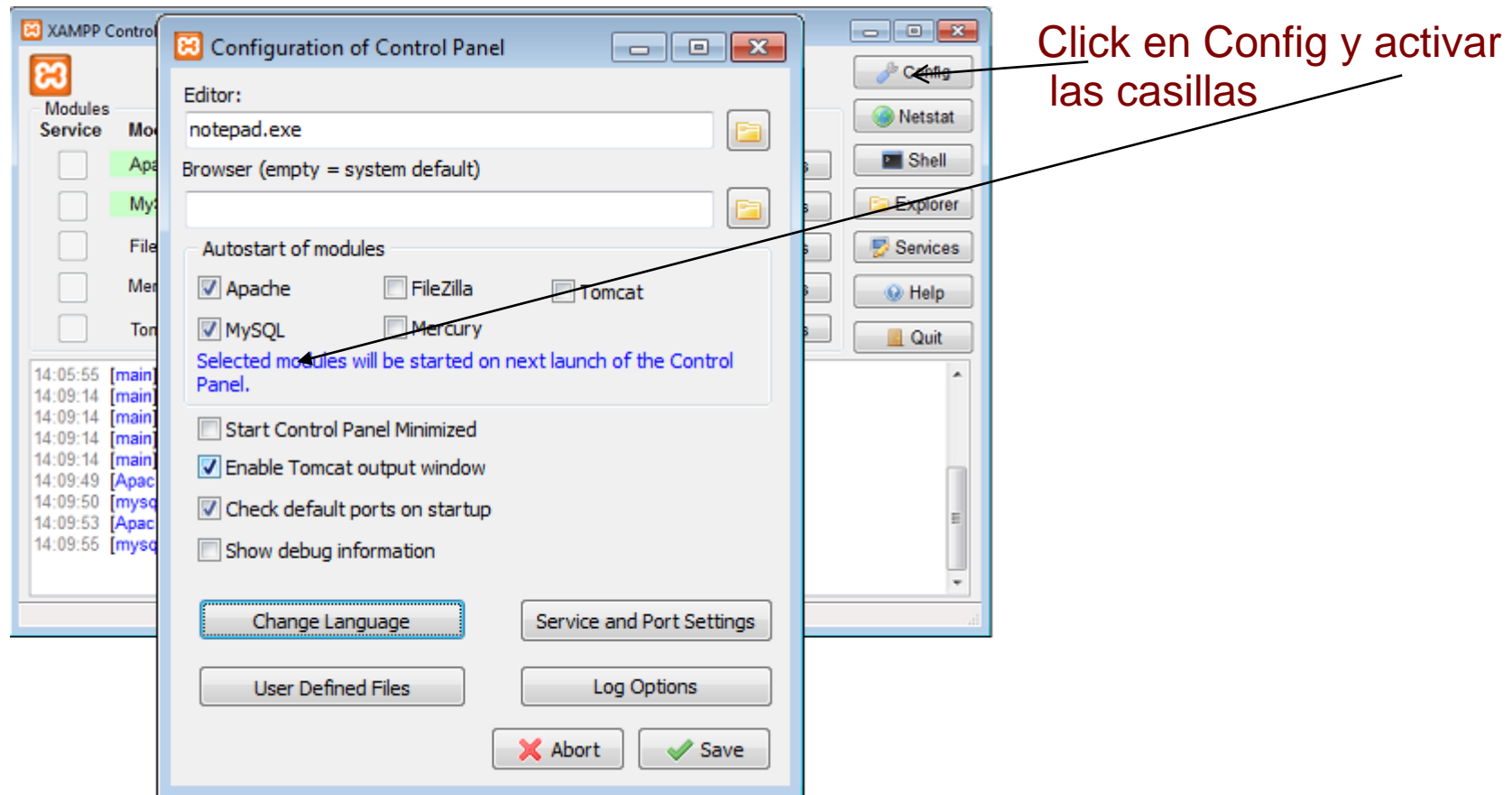
# Arrancar los servicios Apache y MySQL desde xampp



Iniciamos / paramos los servicios a través de *start / stop*

# Arrancar los servicios Apache y MySQL automáticamente desde xampp

- Si queremos que arranquen los servicios al iniciar Windows automáticamente (Nosotros NO)



# Probar la instalación

Tecleamos en el navegador <http://localhost>



# Configuración de Apache (en xampp)

## – Fichero de configuración httpd.conf

- El fichero de configuración de Apache es el archivo **httpd.conf**, un archivo de texto sin formato.
- En XAMPP 1.7.4 el archivo *httpd.conf* se encuentra en el directorio: **C:\xampp\apache\conf\**
- En *httpd.conf* las líneas comentadas empiezan por el carácter almohadilla (#). Estas no tienen efecto
- Las líneas descomentadas tienen efecto. Se pueden modificar. Es recomendable comentar los cambios y guardar una copia de **httpd.conf** por seguridad.
- Si se cambia la configuración, hay que reiniciar el servidor.

# httpd.conf – Algunas directivas de configuración de Apache

- **DocumentRoot** - especifica el directorio donde se encuentran las páginas web. Directorio raíz del servidor.
  - Cuando en la barra de dirección del navegador se escriba una dirección que comience por `http://localhost/`, el servidor buscará los archivos a partir de ese directorio.

```
181 # symbolic links and aliases may be
182 #
183 DocumentRoot "C:/xampp/htdocs"
184
185 #
```

# httpd.conf – Algunas directivas de configuración de Apache

- **Listen** - definir una IP y/o puerto que “escuchara” nuestro servidor, de forma que al teclear esa IP y puerto en nuestro navegador veamos el contenido web
  - usará la ip por defecto, y el puerto 80, también por defecto, por tanto funcionará si llamamos a **http://127.0.0.1**
- **ServerName** - definir el nombre del servidor y un puerto determinado
  - además de utilizar la IP, podremos de esta forma utilizar **http://localhost:80** o bien solo **http://localhost** ya que el puerto 80 lo toma por defecto.

```
16 #Listen 0.0.0.0:80
17 Listen 80
18
```

```
174 # If your host doesn't have
175 #
176 ServerName localhost:80
177
```

# httpd.conf – Algunas directivas de configuración de Apache

- **DirectoryIndex** - como responde el servidor cuando se solicita un directorio (qué pagina por defecto muestra)

```
244 <IfModule dir_module>
245     DirectoryIndex index.php index.pl index.cgi index.asp index.shtml index.html index.htm \
246                   default.php default.pl default.cgi default.asp default.shtml default.html default.htm
247                   home.php home.pl home.cgi home.asp home.shtml home.html home.htm
248 </IfModule>
```



# Configuración de PHP (en xampp) – El fichero php.ini

- El archivo de configuración de PHP es el archivo **php.ini**, un archivo de texto sin formato.
- En php.ini las líneas comentadas empiezan por el carácter punto y coma (;).
- En XAMPP 1.7.4 (Windows) el archivo php.ini se encuentra en el directorio **C:\xampp\php\php.ini**.

# php.ini – Algunas directivas de configuración de PHP

## ■ **short\_open\_tag**

- **short\_open\_tag = On** - Se admite el uso de etiquetas cortas `<? ... ?>` en vez de `<?php ... ?>`
- Aunque se recomienda no utilizar las etiquetas cortas dejaremos la directiva a on para evitar problemas con las páginas del panel de control de xampp

## ■ **error\_reporting** -

- especifica el tipo de errores notificados por PHP.
- **error\_reporting = E\_ALL | E\_STRICT** ; Valor recomendado
  - así se muestran todos los errores

```
514 ; Production Value: E_ALL & ~E_DEPRECATED
515 ; http://php.net/error-reporting
516 error_reporting = E_ALL | E_STRICT
517
```

<http://php.net/manual/es/ini.core.php>

# php.ini – Algunas directivas de configuración de PHP

- **error\_reporting -**

- En producción, se suelen omitir los mensajes de error por motivos de seguridad
- En desarrollo es conveniente mostrar el mayor número de errores y avisos para corregirlos.

```
966 ;extension=php_mysqlnd.dll
967 extension=php_mysql.dll
968 ;extension=php_oci8.dll
```

- **extensión mysqli**

- activada este curso para trabajar con MySQL

- **date.timezone**

- establece la zona horaria del lugar en que está ubicado el servidor para todas las funciones de fecha y hora.

# php.ini – Algunas directivas de configuración de PHP

- **session.save\_path**

- lugar del servidor en el que se almacena información temporal relativa a las sesiones

```
1475 ; where MODE is the octal representati  
1476 ; does not overwrite the process's uma  
1477 ; http://php.net/session.save-path  
1478 session.save_path = "C:\xampp\tmp"  
1479
```

# Descargar e instalar kit de desarrollo de Java (JDK)

- Para la instalación de NetBeans se necesita tener instalado el JDK de Java
  - software que provee herramientas de desarrollo para la creación de programas en java
  - NetBeans está desarrollado en Java, por eso necesita el JDK
- <http://www.oracle.com/technetwork/java/javase/downloads/index.html>



- Descargamos si no tenemos (Nos avisa NetBeans) y lo ejecutamos

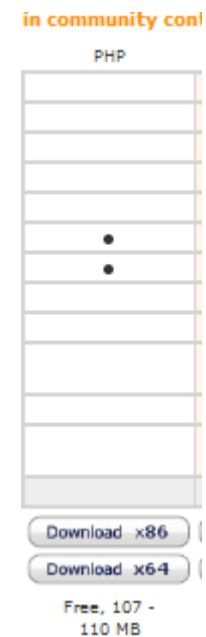
# Descargar e instalar NetBeans para PHP

- Descargaremos la versión NetBeans para PHP
- <http://netbeans.org/downloads/index.html>
- Elegimos idioma español / Plataforma Windows



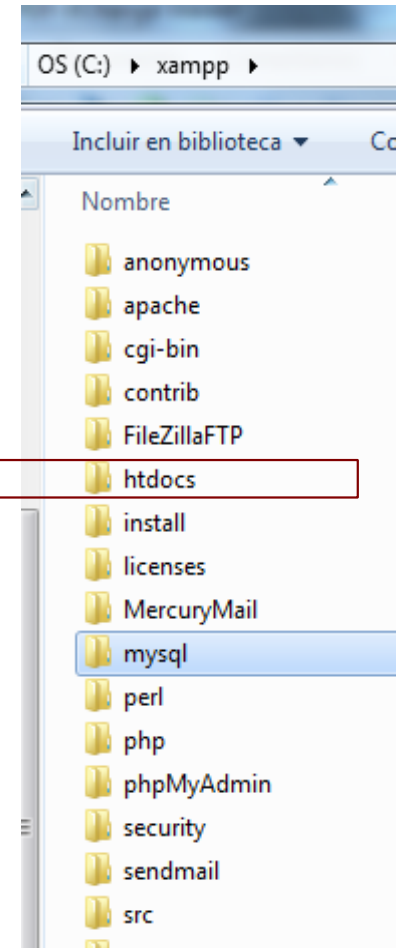
Idioma del IDE: Español Plataforma: Windows

- Doble click en el fichero *netbeans-xxxxxx-php-windows* para instalar
- Aceptamos todas las opciones recomendadas



# Despliegue de una aplicación PHP(importante)

- Estructura de directorios de XAMPP
- *Desplegar* una aplicación web es hacerla accesible desde un navegador para poder ejecutarla
- Para ejecutar una aplicación PHP en un servidor local (**localhost**) tenemos que guardar todos los directorios y ficheros de la aplicación en el directorio raíz (*Document root*) del servidor Apache que está en **C:\xampp\htdocs**



# Despliegue de una aplicación PHP

Figura A.(directorios)

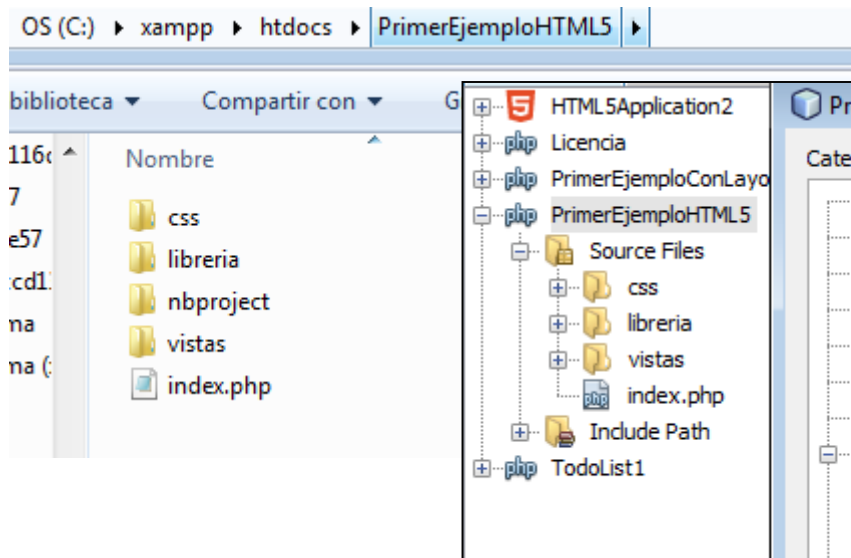
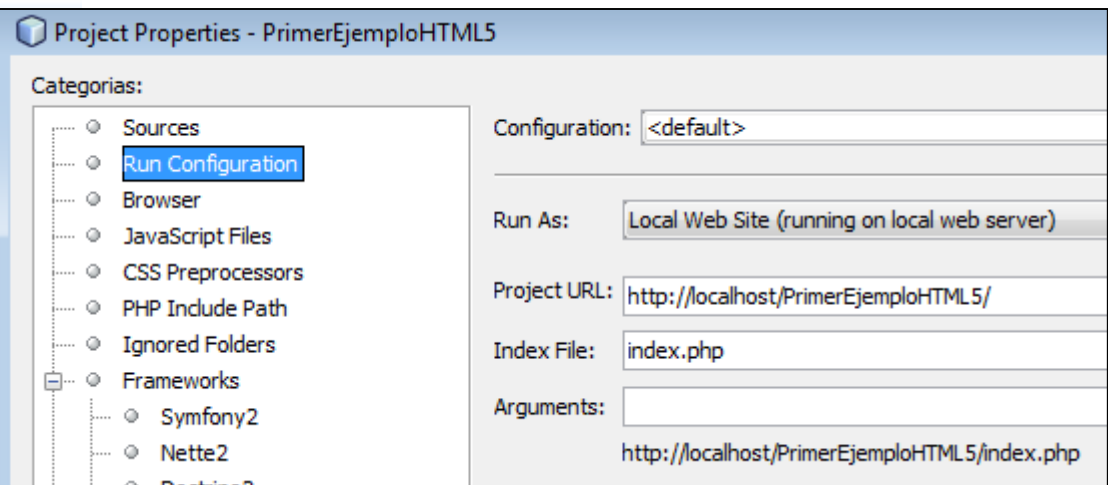


Figura B. (NetBeans)



**htdocs** – directorio raíz del servidor,  
directorio raíz para todas las  
aplicaciones

**PrimerEjemploHTML5**– directorio raíz  
de la aplicación

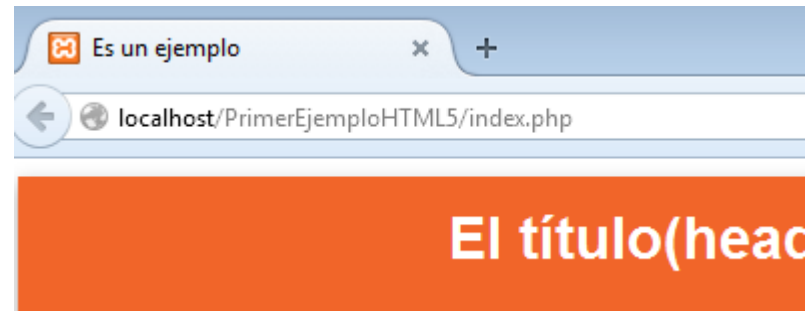
Opción **Run Configuration** en  
Propiedades del proyecto



# Ejecutar una aplicación PHP en un servidor local

- Introducir la URL (Uniform Resource Locator) de la página solicitada en a barra de direcciones del navegador

<http://localhost/PrimerEjemploHTML5>



- Si la primera página no es index.php debemos incluirla en la URL  
<http://localhost/PrimerEjemploHTML5/primera.php>

# Ejecutar una aplicación PHP en un servidor de Internet

- Para desplegar una aplicación web PHP en un servidor de Internet subiremos, a través de FTP, los directorios y ficheros de la aplicación desde nuestro servidor local al directorio raíz del servidor Internet .
- El servidor Internet deberá ejecutar Apache y PHP
- Introducimos la URL de la página solicitada en la barra de direcciones del navegador:

<http://docentes.educacion.navarra.es/carocena/PrimerEjemploHTML5/>

<http://localhost/ejerciciosPHP/PrimerEjemploHTML5>

- Las aplicaciones web se desarrollan y testean en un servidor local y luego se suben a un servidor Internet

# Componentes de una URL HTTP

- <http://docentes.educacion.navarra.es/carocena/PrimerEjemploTML5/>

protocolo

nombre dominio  
*localhost* en el caso  
del servidor local

camino (path) en el  
que reside el fichero  
en el servidor

nombre del  
fichero

- Si no se especifica fichero, por defecto, el servidor Apache busca alguno de los siguientes, *index.htm*, *index.html*, *index.php*
- Si se omite fichero y no encuentra uno por defecto, Apache muestra la lista de directorios y ficheros en esa ruta

# Ejemplo

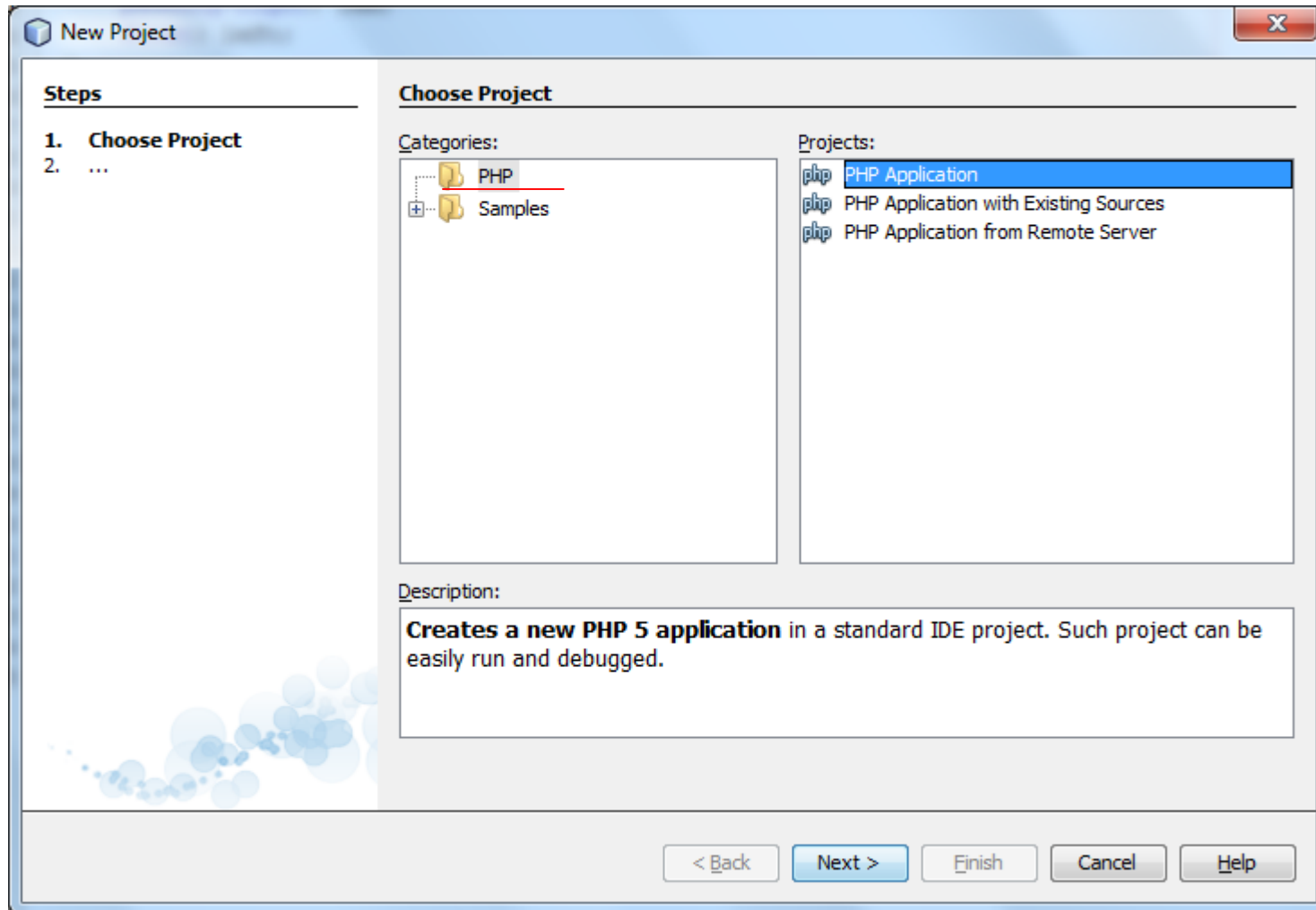
- Si cambiamos el proyecto a otra carpeta (**¡¡CUIDADO!!**) cambia su URL.
- Por ejemplo vamos a crear una carpeta llamada **ejerciciosPHP** dentro de **C:\xampp\htdocs**.
- Debemos cambiar la ruta en **Run Configuration del proyecto** a la URL:

<http://localhost/ejerciciosPHP/PrimerEjemploHTML5/>

- y pulsamos Ejecutar el proyecto.
- O ponemos directamente la dirección en el navegador.

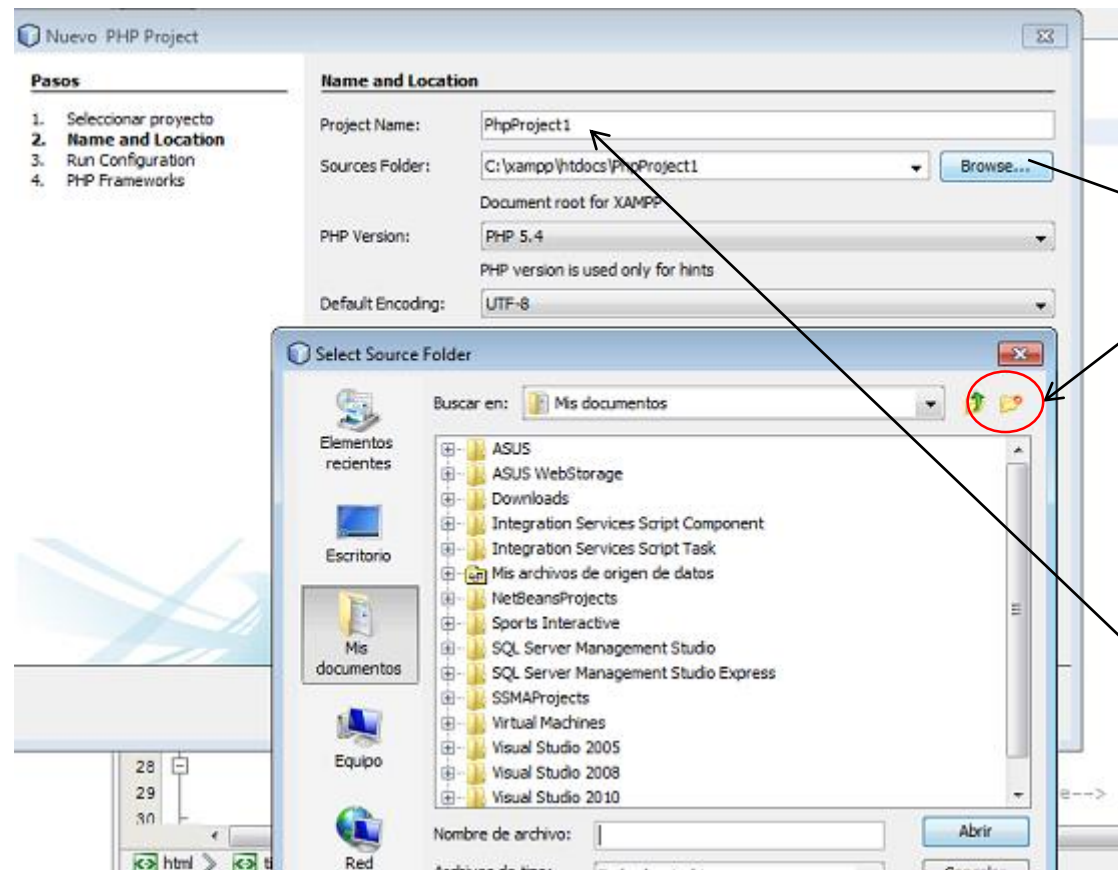
# Crear un proyecto nuevo en NetBeans

## File / New project



# Crear un proyecto nuevo en NetBeans

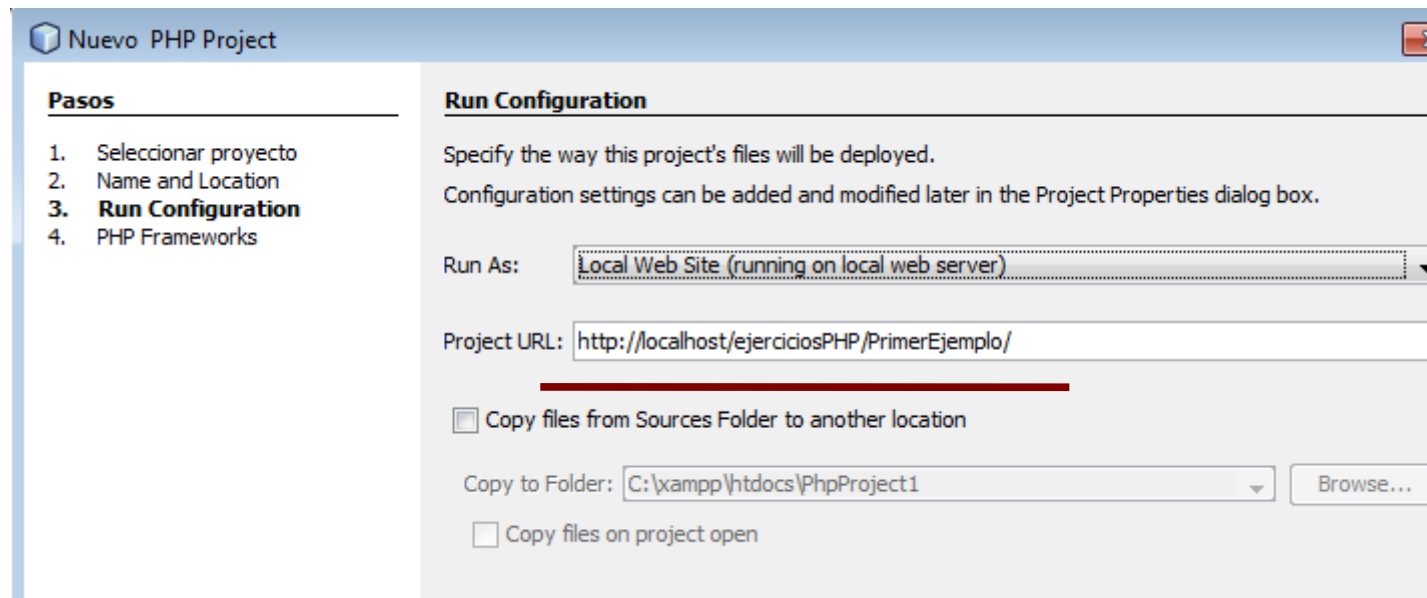
Elegimos PHP



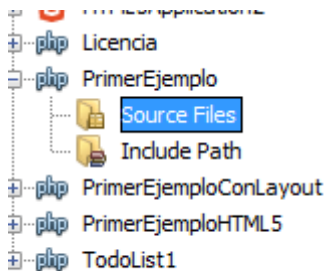
En el directorio c/xampp/htdocs, creamos una carpeta y la seleccionamos

Le ponemos nombre al proyecto, puede ser el mismo o diferente que el nombre de la carpeta

# Crear un proyecto nuevo en NetBeans



# Estructura de un proyecto NetBeans



- La carpeta Source files va a contener todos los ficheros de los que constará nuestra aplicación (hojas de estilo, varios ficheros .php, ficheros .html si los hay, ...)
- En ella, botón dcho y añadimos un nuevo PHP File (<?php ...?>) de nombre index.php

```
<?php  
/*
```

```
* To change this license header, choose License Headers in Project Properties.  
* To change this template file, choose Tools | Templates  
* and open the template in the editor.  
*/
```


```
?>
```

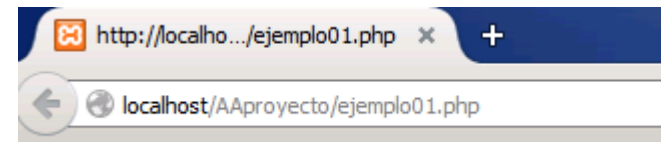
```
echo "<p>Mi primera prueba en PHP</p>";
```

Modificamos  
index.php y  
añadimos el código  
subrayado



# Ejecutar un proyecto NetBeans

- *En el menú Run elegir Run Project (F6) o *
- *o colocarse en el archivo y Botón derecho /Run*
- *o clickar en el script y Botón derecho /Run File*



## Ejercicios PHP 1ª Evaluación

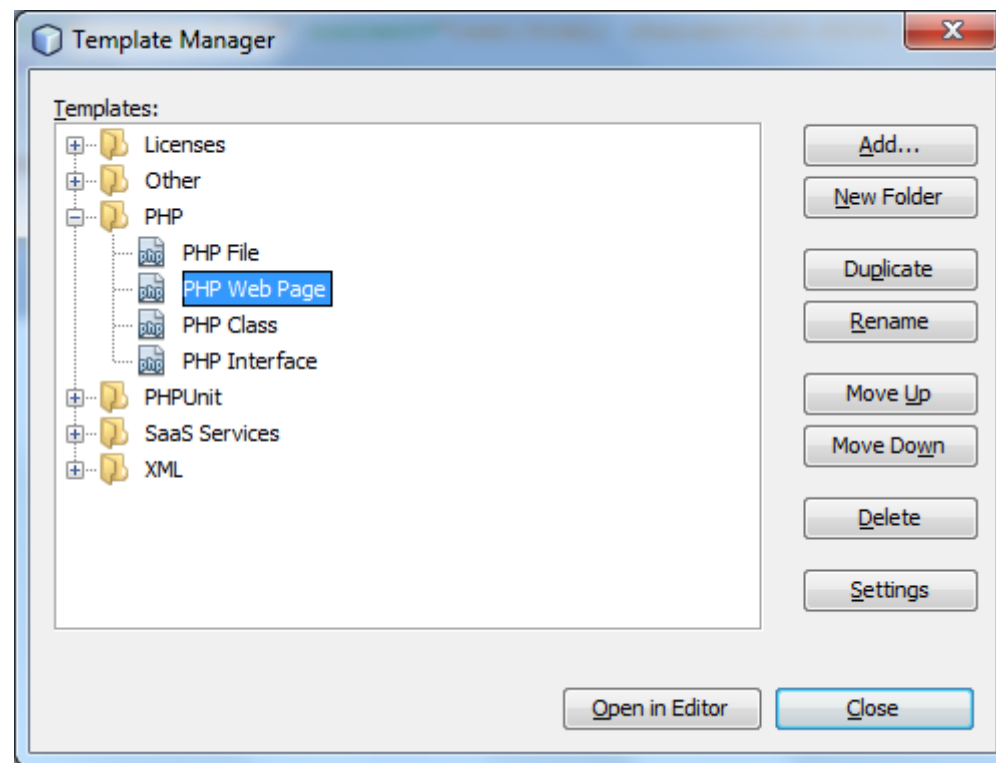
Primera prueba en PHP

# Generando HTML válido y renombrando archivos

- Recordad los plugins: HTML Validator, HTML 5 Validator si queremos utilizarlos.
- Más adelante crearemos nuestro propio *index.php*. Inicialmente llamaremos a los scripts con otros nombres, *ejemplo01.php*, *bienvenida.php*, *calcular.php*, ....
- Para renombrar un archivo (Botón dcho -> *Rename*) en el archivo php y le ponemos *ejemplo01.php*. Así se va a llamar este script PHP.

# Cambiando plantilla *PHP Web Page* en NetBeans

- Podemos cambiar la plantilla de *PHP Web Page*
- *Tools / Templates / PHP / PHP Web Page / Open in Editor*



# Plantilla *PHP Web Page* en NetBeans

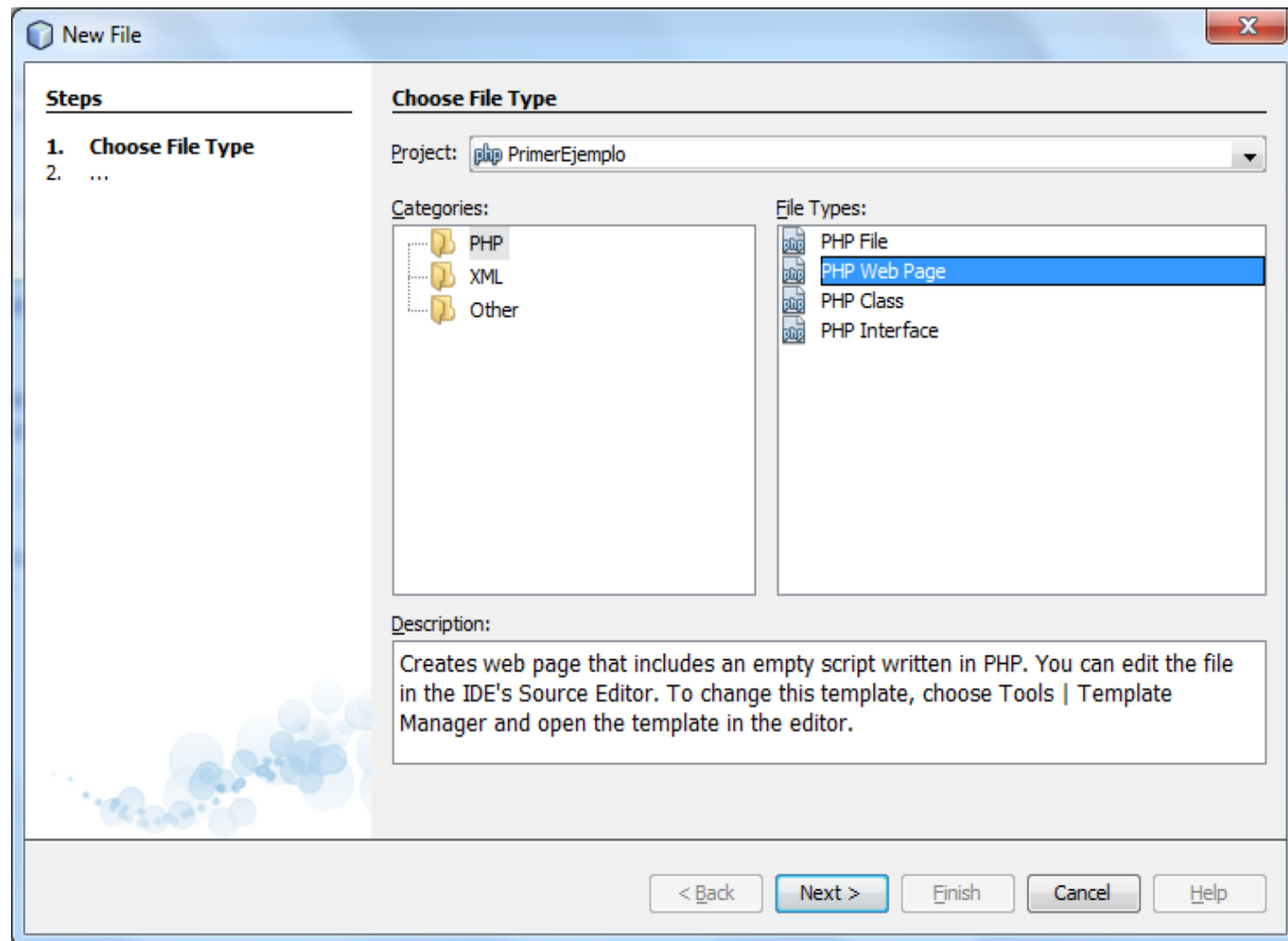
```

    ${doctype}
] <#assign licenseFirst = "<!--">
  <#assign licensePrefix = "">
- <#assign licenseLast = "-->">
  <#include "${project.licensePath}">
<html>
  <head>
    <meta charset="${project.encoding}">
    <title></title>
  </head>
  <body>
    <h1>Ejercicios PHP 1ª Evaluación</h1>
    <?php
      // put your code here
    ?>
  </body>
</html>
```

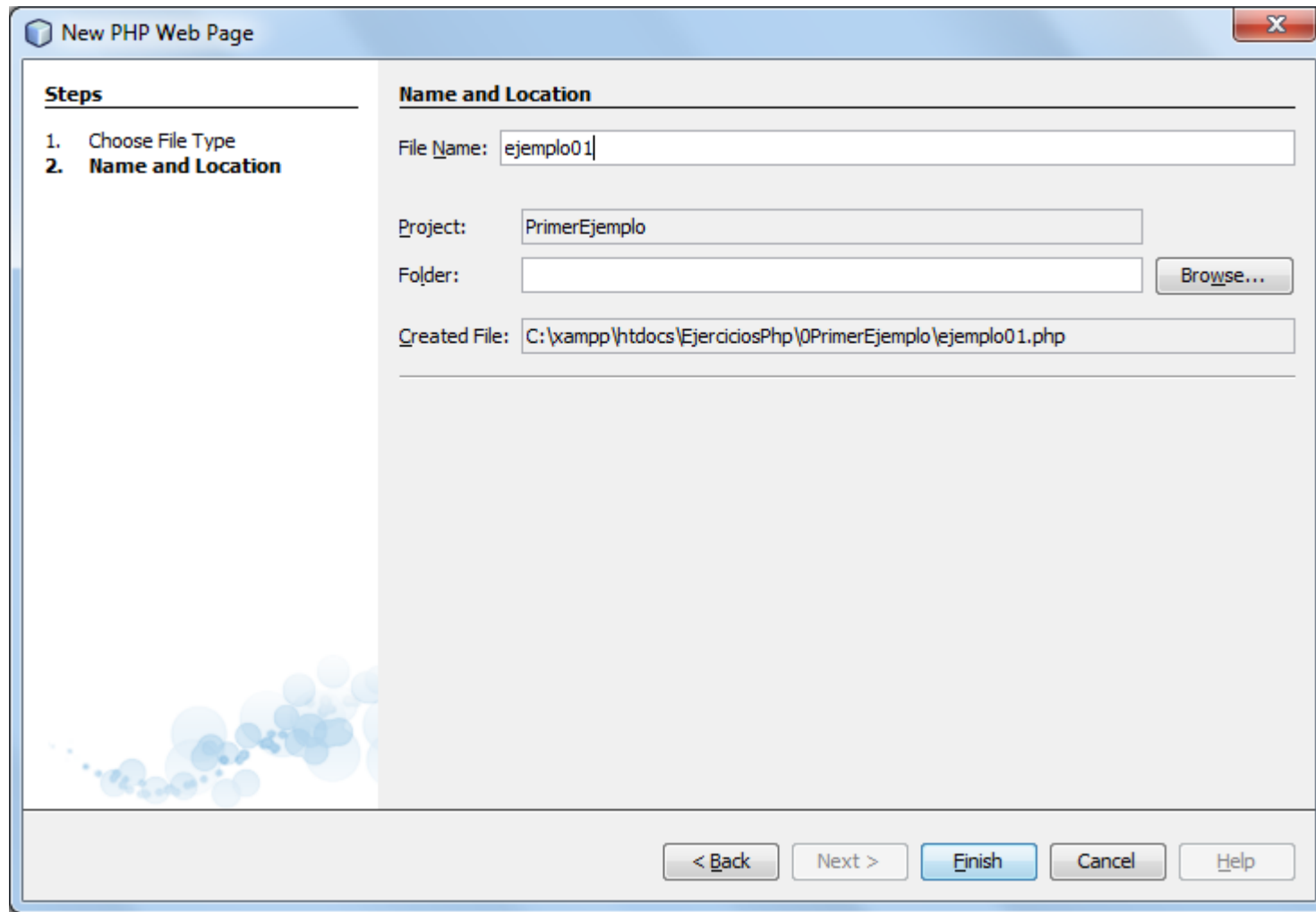
Podríamos incluir código y salvar la plantilla, así nuestros archivos generados *PHP Web Page* tendrían dicho código

# Añadir *PHP Web Page* en NetBeans

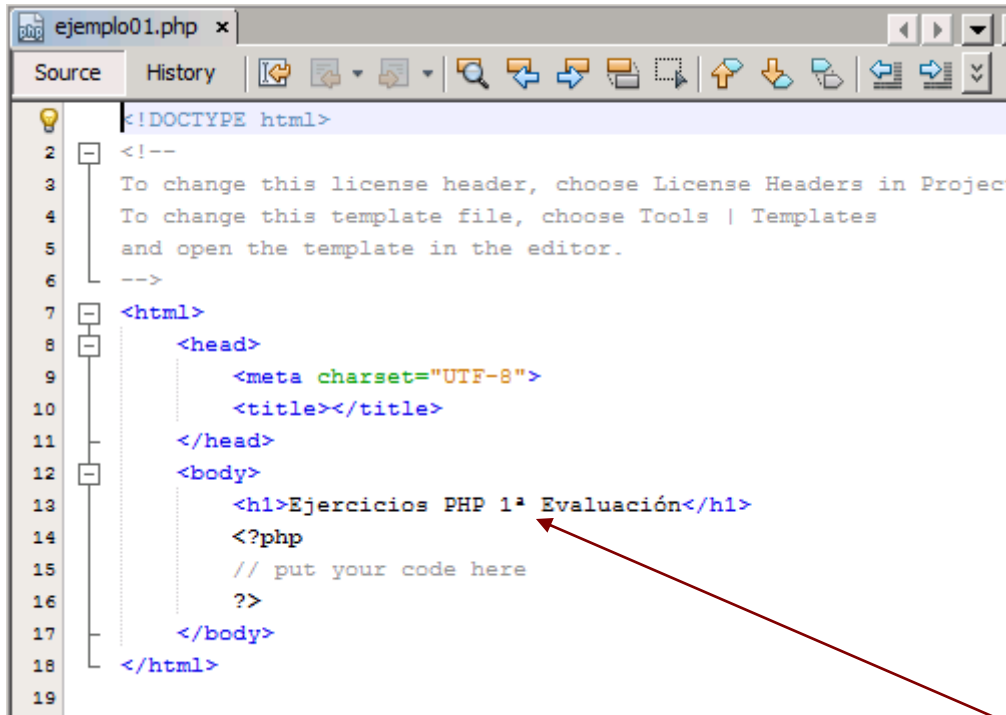
- *File / New File*



# Añadir *PHP Web Page* en NetBeans



# ejemplo01.php



```
ejemplo01.php x
Source History
<!DOCTYPE html>
2 <!--
3 To change this license header, choose License Headers in Project
4 To change this template file, choose Tools | Templates
5 and open the template in the editor.
6 -->
7 <html>
8 <head>
9 <meta charset="UTF-8">
10 <title></title>
11 </head>
12 <body>
13 <h1>Ejercicios PHP 1ª Evaluación</h1>
14 <?php
15 // put your code here
16 ?>
17 </body>
18 </html>
19
```

localhost/Aproyecto/ejemplo01.php

## Ejercicios PHP 1ª Evaluación

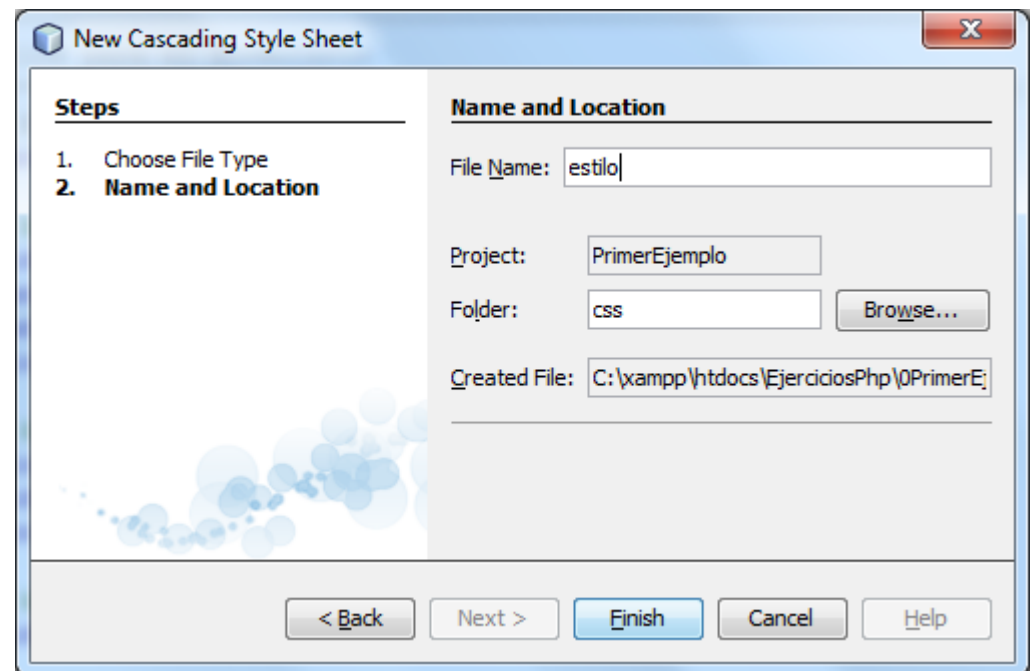
Primera prueba en PHP

Ya ha  
cambiado la  
plantilla

- Seleccionar *ejemplo01.php*  
*botón derecho / Run*

# Podemos añadir una hoja de estilo a la aplicación

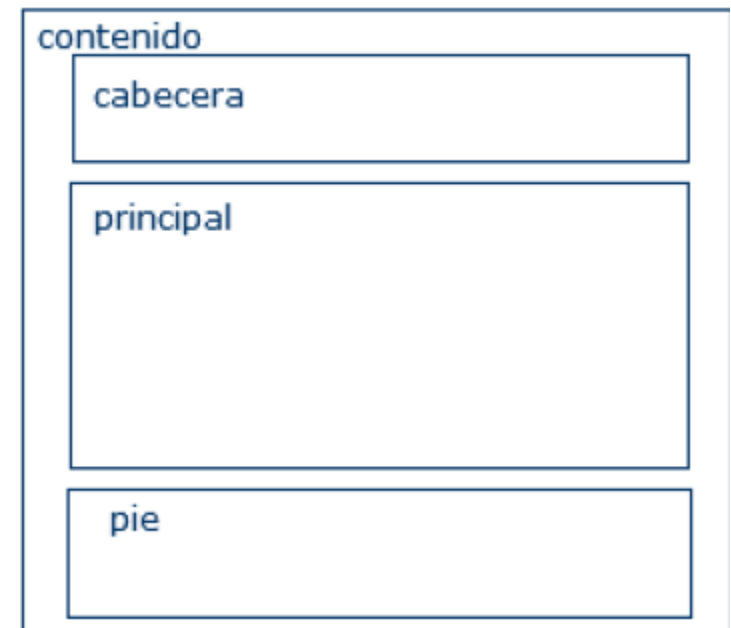
- *Nodo proyecto / botón derecho / New / Folder*
  - damos el nombre *css* a la carpeta
- *Seleccionamos carpeta *css* / botón derecho / New / Cascading style sheet*
  - damos el nombre *estilo*
- Incluimos estilos





# Añadimos una hoja de estilo a la aplicación

- Modificamos *ejemplo01.php* y obtenemos (PrimerEjemploConLayout) al incluir la hoja de estilo.
- Damos una estructura lógica a la página (con elementos *div* o con otras etiquetas) para que se apliquen correctamente los estilos (os valdrá para el resto del curso) .
- A medida que avance el curso y vayáis trabajando con HTML 5 y CSS3 sería conveniente realizar el cambio.

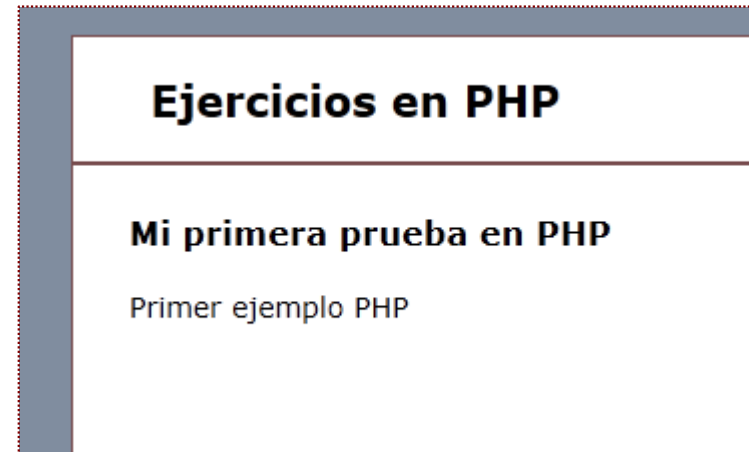


# Añadimos una hoja de estilo a la aplicación tal y como sabemos

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Es un ejemplo</title>
    <link rel="stylesheet" type="text/css" href="css/estilo.css" />
    <title>Primer ejemplo</title>
  </head>  <body>
    <div id="contenido">
      <div id="cabecera">
        <h1>Ejercicios en PHP</h1>
      </div> <!-- fin de cabecera -->
      <div id="principal">
        <h3>Mi primera prueba en PHP</h3>
        <?php
        echo "<p>Primer ejemplo PHP</p>";
        ?>
      </div>  <!-- fin de principal-->
```

# Añadimos una hoja de estilo a la aplicación

```
<div id="pie">
  <p class="copyright">
    &copy; <?php echo date("Y"); ?> Ejercicios iniciales de PHP
  </p>
</div>
</div><!-- fin de contenido -->
</body>
</html>
```



# Complementos para Firefox

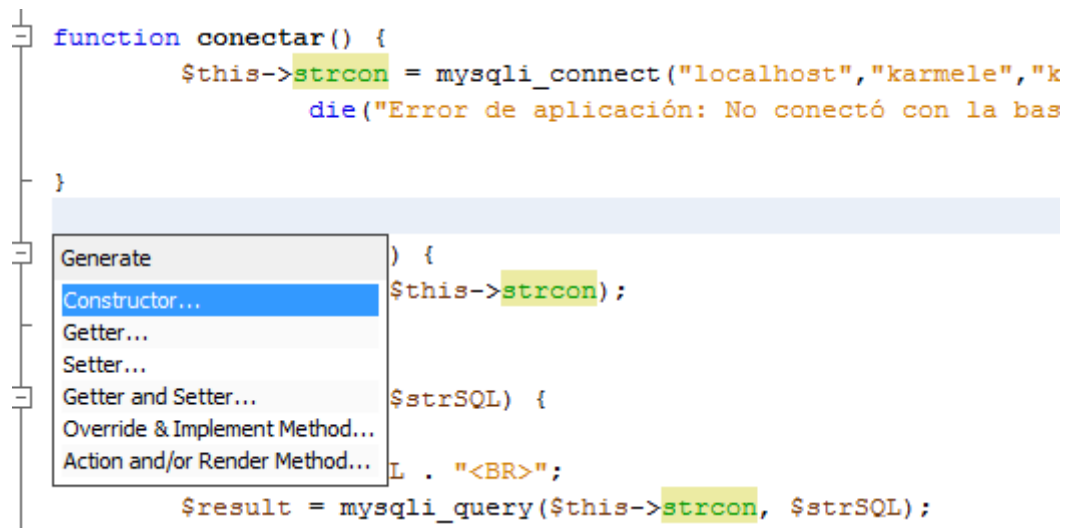
- Los navegadores pueden incluir complementos varios (Add-ons) de ayuda para desarrolladores web
- **FireBug.** Nos permite editar, eliminar errores, ver los códigos HTML, CSS y Javascript en cualquier página.
- **FirePHP** [http://www.dailymotion.com/video/xa4chg\\_complemento-firephp-para-debug-php\\_tech](http://www.dailymotion.com/video/xa4chg_complemento-firephp-para-debug-php_tech)
- **Web Developer** - extensión Firefox que provee de una serie de utilidades de edición y depuración para desarrolladores web.



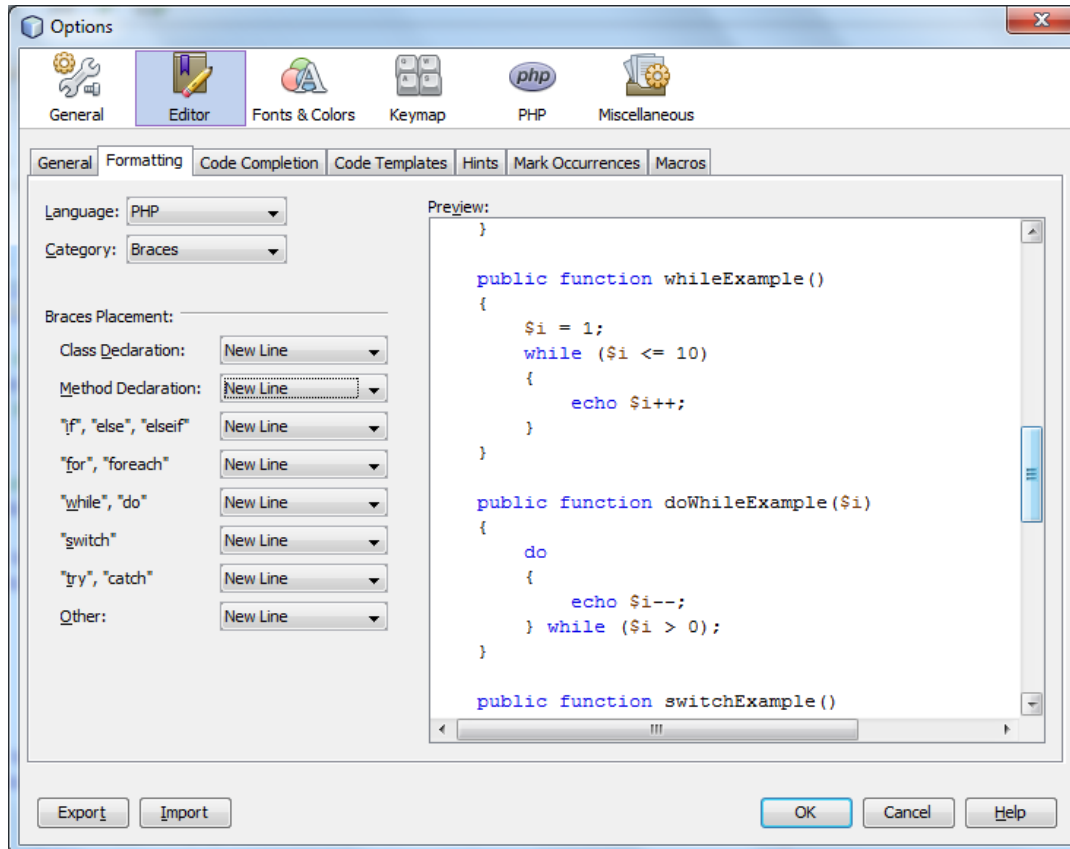
<http://www.cosassencillas.com/articulos/11-complementos-de-firefox-esenciales-para-el-desarrollo-web>

# Atajos de teclado para NetBeans

- algunos atajos de teclado útiles
  - *ctrl-space* (*code completion* – completar código)
  - *Source Format* o seleccionar un trozo de código y *botón derecho/Format*
  - *Alt+Insert* (*POO*):



# Configurando algunas opciones del editor en NetBeans



- EL editor de código en NetBeans es configurable
- *Tools/Options/Editor/Formatting*
  - Podemos elegir las opciones tal cómo se muestra en la *figura A*, así la generación automática de {} se hace en una nueva línea.

Figura A