



# Desarrollo de Aplicaciones Web

**DESARROLLO WEB**  
**ENTORNO SERVIDOR**  
**DWES**

## ¿Qué es una aplicación web?

Es una aplicación web ha sido creada para ser ejecutada por un navegador.

Una aplicación web es una aplicación a la que accedemos mediante protocolo HTTP utilizando un navegador web.

Es un software que permite al usuario realizar una tarea o servicio.

Las aplicaciones web se crean *usando* HTML y aunque los navegadores no pueden compilar lenguajes clásicos como Java, sí contienen intérpretes muy potentes para los lenguajes de programación web (JavaScript).

## Tipos de páginas Web

### Páginas estáticas

Diremos que una página es estática cuando sus contenidos no pueden ser modificados –ni desde el *servidor* que la aloja ni tampoco desde el *cliente* (navegador)– mediante ninguna intervención del usuario ni tampoco a través de ningún programa.

## Páginas dinámicas.

Llamaremos dinámicas a las páginas cuyos contenidos sí pueden ser modificados –de forma automática o mediante la intervención de un usuario– bien sea desde el *cliente* y/o desde el *servidor*.

Para que esas modificaciones puedan producirse es necesario que *algo o alguien* especifique: *qué, cómo, cuándo, dónde* y *de qué forma* deben realizarse, y que exista otro *algo o alguien* capaz de acceder, interpretar y ejecutar tales instrucciones en el momento preciso.

Las *especificaciones* y las *instrucciones* requieren: un lenguaje para definir las; un soporte para almacenarlas y un intérprete capaz de *ejecutarlas*.

# INTRODUCCIÓN A LA PROGRAMACIÓN WEB

## PÁGINA WEB ESTÁTICA

Se presenta sin movimiento y sin funcionalidades más allá de los enlaces

Se le puede agregar **Hojas de Estilo (CSS)**, que amplían los estilos de los elementos (colores, fuentes, links, etc)

```
<A HREF="pagina2.htm" style="color:red"> Página 2</A>
```

Su código está escrito en el lenguaje HTML (Hyper Text Mark up Language)

Indica dónde colocar cada elemento en la página, y con qué características

A través de etiquetas

```
<A HREF="pagina2.htm"> Página 2 </A>
```

# INTRODUCCIÓN A LA PROGRAMACIÓN WEB

## PÁGINA WEB

Archivo compuesto por diversos elementos:

TEXTO

ENLACES

MULTIMEDIA

Dispuestos en pantalla según ciertas características

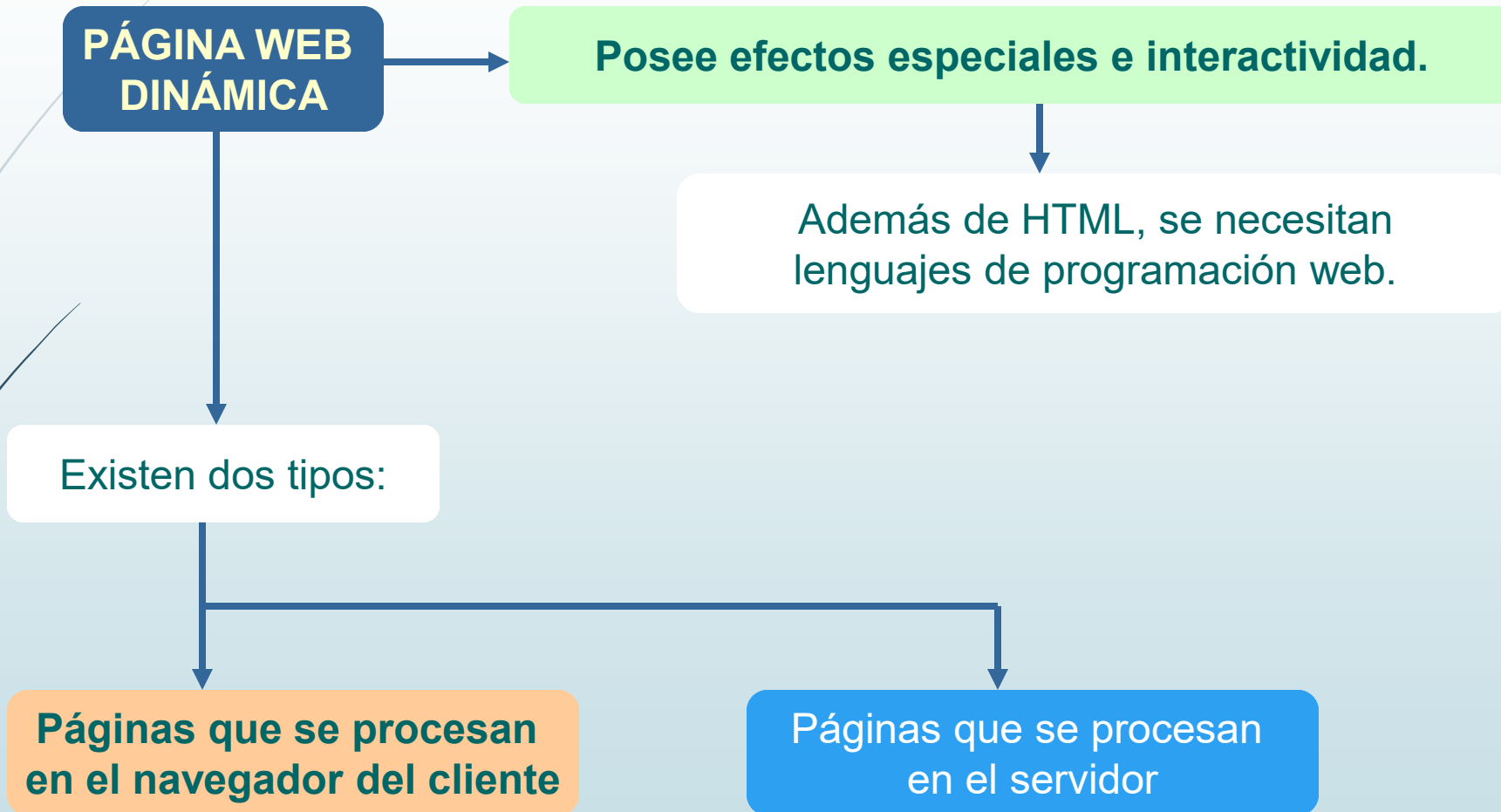
Interactividad y efectos ( menús, movimiento de objetos, fecha y hora, imágenes, mensajes, etc.)

Validaciones, acceso a bases de datos (creación, consultas,...), manejo de claves, cookies...

ESTÁTICAS

DINÁMICAS

# INTRODUCCIÓN A LA PROGRAMACIÓN WEB



## Scripts (guiones)

Se llama *script* a un conjunto de *instrucciones* escritas en un *lenguaje* determinado que van incrustadas dentro de una *página WEB* de modo que su *intérprete* pueda acceder a ellas en el momento en el que se requiera su *ejecución*.

Cuando se incrustan *scripts* en una *página WEB* empiezan a *convivir* en un mismo documento informaciones destinadas a distintos *intérpretes*.

- Por una parte, el *código HTML* que ha de ser *interpretado* por el *navegador*.
- Por la otra, los *scripts* que han de ser *ejecutados* -dependiendo del lenguaje en el que hayan sido escritos- por su *intérprete* correspondiente.



# INTRODUCCIÓN A LA PROGRAMACIÓN WEB

## Scripts (guiones)

- La manera de diferenciar los contenidos es delimitar los scripts marcando su comienzo con una etiqueta de apertura `<script>` y señalando el final con una etiqueta de cierre `</script>`.
- Lo que no está contenido entre esas etiquetas se considerará código HTML.
- La posibilidad de insertar en un mismo documento scripts desarrollados en distintos lenguajes obliga a especificar cuál se ha utilizado en cada caso, para que en el momento en el que vayan a ser ejecutados se invoque el intérprete adecuado.
- Para el caso concreto de PHP, existe una sintaxis que es la que se usa de forma estándar

```
<?php
```

```
.....
```

```
.....
```

```
instrucciones
```

```
?>
```

## Scripts (guiones)

Los lenguajes de *script* pueden clasificarse en dos tipos:

- Del lado del cliente
- Del lado del servidor

## Lenguajes del lado del cliente:

Diremos que un lenguaje es del lado del cliente cuando el intérprete que ha de ejecutar sus scripts es accesible desde éste –el cliente– sin que sea necesario hacer ninguna petición al servidor.

Cuando los *scripts* contenidos en un documento son de este tipo, el *servidor lo entrega al cliente* si efectuar ningún tipo de modificación.

## Lenguajes del lado del servidor

- Un lenguaje es *del lado del servidor* cuando la ejecución de sus *scripts* se efectúa, por *instancia* de este —el *servidor*—, antes de *dar respuesta a la petición*, de manera que el *cliente* no recibe el documento original sino el resultante de esa interpretación previa.
- Cuando se usan estos tipos de lenguaje el *cliente* recibe un documento en el que cada *script* contenido en el original habrá sido *sustituido* por los resultados de su ejecución.
- En este caso, los usuarios no tendrán la posibilidad de visualizar el código fuente, mientras que cuando se trata de *lenguajes del lado del cliente* siempre es posible visualizar los *scripts*.
- La utilización de este tipo de *scripts* requiere que el *intérprete* del lenguaje sea *accesible* desde el propio *servidor*, es decir, que esté del lado del servidor.

# INTRODUCCIÓN A LA PROGRAMACIÓN WEB

**PÁGINA WEB  
DINÁMICA**

**Páginas que se procesan en el navegador del cliente**

Control de ventanas y formularios, cálculos,  
fecha y hora, efectos de imagen y de texto...

Se logran con código insertado en el HTML,  
llamado script (Guion)

Los scripts se pueden programar en:

**JavaScript**

# INTRODUCCIÓN A LA PROGRAMACIÓN WEB

**PÁGINA WEB  
DINÁMICA**

**Páginas que se procesan en el navegador del cliente**

**JavaScript**

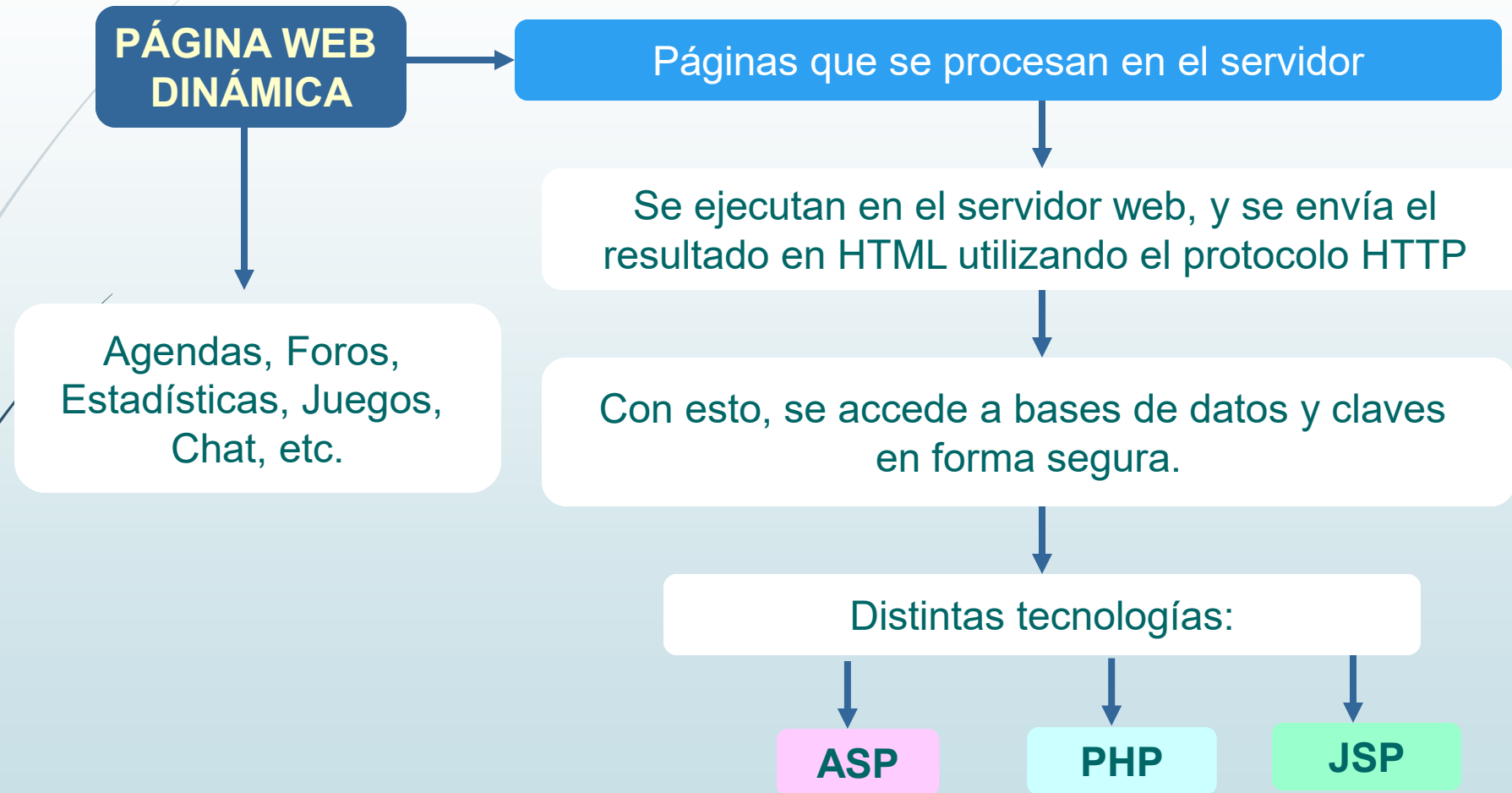
Permite la programación de pequeños scripts y de programas orientados a objetos, con funciones, estructuras de datos complejas, etc.

NO ES JAVA

Su sintaxis deriva del lenguaje C

Funciona en cualquier navegador,  
Es considerado un estándar.

# INTRODUCCIÓN A LA PROGRAMACIÓN WEB



# INTRODUCCIÓN A LA PROGRAMACIÓN WEB

**PÁGINA WEB  
DINÁMICA**

Páginas que se procesan en el servidor

**PHP**

**Hypertext PreProcesor**

Gratuito, multiplataforma,  
rápido y seguro.

Está montado sobre servidores  
Linux o Unix o Windows

Posee un lenguaje  
propio derivado del C

Servidor web: Apache



## Ejecución de código en el servidor y en el cliente.

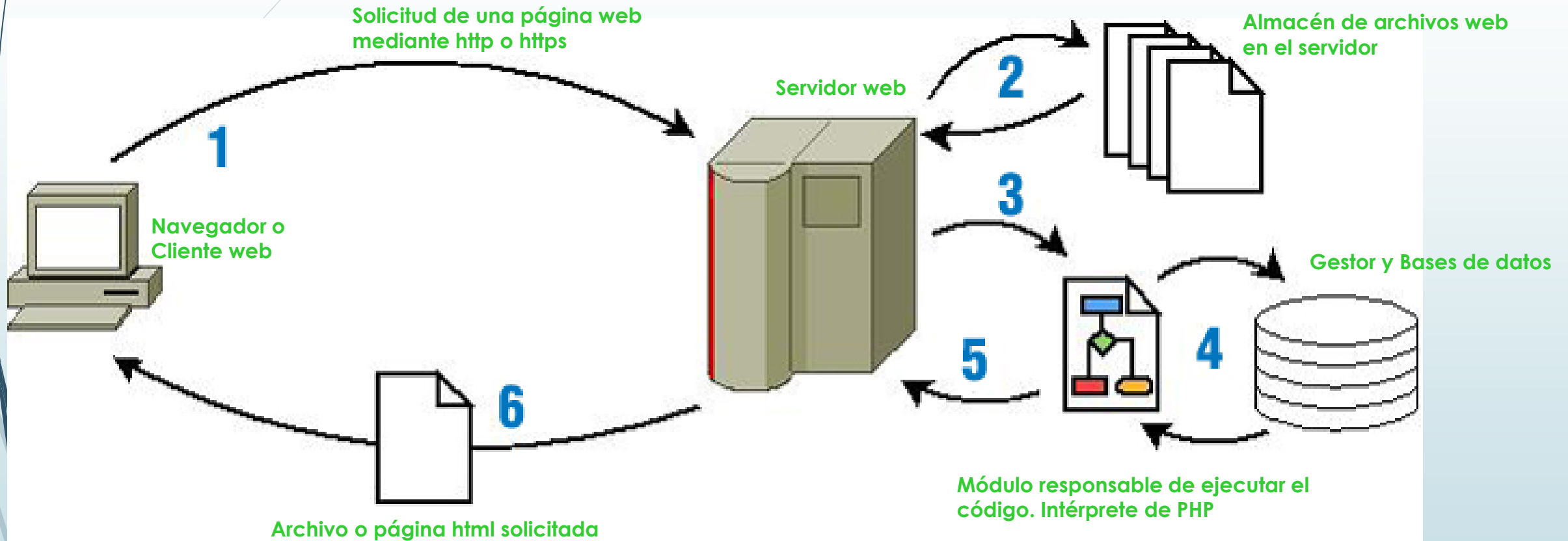
El navegador solicita a un servidor web una página, es posible que antes de enviártela haya tenido que ejecutar, por sí mismo o por delegación, algún programa para obtenerla. Ese programa es el que genera, en parte o en su totalidad, la página web que llega a tu equipo. En estos casos, el código se ejecuta en el entorno del servidor web.

Además, cuando una página web llega a tu navegador, es también posible que incluya algún programa o fragmentos de código que se deban ejecutar. Ese código, normalmente en lenguaje JavaScript, se ejecutará en tu navegador y, además de poder modificar el contenido de la página, también puede llevar a cabo acciones como la animación de textos u objetos de la página o la comprobación de los datos que introduces en un formulario. El código se ejecuta en el entorno del cliente web.

Estas dos tecnologías se complementan una con otra.

# INTRODUCCIÓN A LA PROGRAMACIÓN WEB

El esquema de funcionamiento de una página web dinámica en el entorno del servidor



# INTRODUCCIÓN A LA PROGRAMACIÓN WEB

El esquema de funcionamiento de una página web dinámica en el entorno del servidor es el siguiente:

Pasos:

1. El cliente web (navegador) de tu ordenador solicita a un servidor web una página web.
2. El servidor busca esa página y la recupera.
3. En el caso de que se trate de una página web dinámica, es decir, que su contenido deba ejecutarse para obtener el HTML que se devolverá, el servidor web contacta con el módulo responsable de ejecutar el código y se lo envía.
4. Como parte del proceso de ejecución, puede ser necesario obtener información de algún repositorio (*Cualquier almacén de información digital, normalmente una base de datos*), como por ejemplo consultar registros almacenados en una base de datos.
5. El resultado de la ejecución será una página en formato HTML, similar a cualquier otra página web no dinámica.
6. El servidor web envía el resultado obtenido al navegador, que la procesa y muestra en pantalla.

## Componentes principales para programar aplicaciones que se ejecuten en un servidor web:

1. **Un servidor web** no es más que un programa que atiende peticiones Http y responde sirviendo archivos html. El servidor web recibe las peticiones de los clientes web (navegadores) y les envía la página que solicitan (una vez generada puesto que hablamos de páginas web dinámicas). El servidor web debe conocer el procedimiento a seguir para generar la página web: qué módulo se encargará de la ejecución del código y cómo se debe comunicar con él.
2. **El módulo encargado de ejecutar el código o programa y generar la página web resultante.** Este módulo debe integrarse de alguna forma con el servidor web, y dependerá del lenguaje y tecnología que utilicemos para programar la aplicación web.
3. **Una aplicación de base de datos**, que normalmente también será un servidor. Este módulo no es estrictamente necesario pero en la práctica se utiliza en todas las aplicaciones web que utilizan grandes cantidades de datos para almacenarlos.
4. **El lenguaje de programación** que utilizarás para desarrollar las aplicaciones.

¿Cómo resuelve sus dudas el servidor?

Dado que en unos casos el servidor debe entregar el documento original – páginas estáticas o páginas dinámicas en las que se usan lenguajes del lado del cliente– mientras que en otros casos –páginas dinámicas usando lenguajes del lado del servidor– tiene que devolver el resultado de la ejecución de los scripts, es razonable que te preguntes:

¿cómo sabe el servidor qué debe hacer en cada caso?

- Se indica al poner la **extensión al documento**.
- Si en la petición se alude a un documento con extensión .htm o .html el servidor entenderá que esa página no requiere la intervención previa de ningún intérprete de su lado y entregará la página tal cual.
- Si en esa petición se aludiera a una extensión distinta –.php, por ejemplo– el servidor entendería que antes de servir la página debe leerla y requerir al intérprete de PHP que ejecute los scripts desarrollados en ese lenguaje (en caso de que los contuviera) y devolvería al cliente el documento que resultara de las eventuales ejecuciones de tales scripts.

**En este curso haremos uso de XAMPP: un paquete de código abierto y libre distribución cuya instalación nos provee de las siguientes herramientas:**

- 1. Servidor web http: Apache**
- 2. Sistema gestor de base de datos MySQL y MariaDB**
- 3. Intérprete de PHP**
- 4. Intérprete de Perl**

**XAMPP: acrónimo donde X indica que sirve para cualquier sistema operativo, A por Apache, M (Mysql o MariaDB), P por PHP y P por Perl**

- **El intérprete de PHP se instala como un módulo del servidor Apache, y se encarga de interpretar nuestro código PHP.**
- **MySQL nos permite crear bases de datos para su consulta por parte de los programas en PHP.**
- **El intérprete de Perl queda fuera del contexto de este curso**

**El nombre XAMPP es en realidad un acrónimo:**

- **X (para cualquiera de los diferentes sistemas operativos)**
- **Apache**
- **MariaDB/MySQL**
- **PHP**
- **Perl**

**A partir de la versión 5.6.15, XAMPP cambió la base de datos MySQL por MariaDB.**



## Servidor y cliente en una misma máquina

- Dado que servidor y cliente son únicamente aplicaciones es perfectamente posible que ambas convivan dentro de la misma máquina.
- Utilizaremos una IP (**habitualmente la 127.0.0.1**) reservada para estos casos –preestablecida en la configuración del servidor– y a través de ella se canalizarían las *peticiones a nuestro propio servidor.*
- La dirección 127.0.0.1 tiene asignado el nombre **localhost** en el archivo hosts.

C:\Windows\System32\drivers\etc\hosts

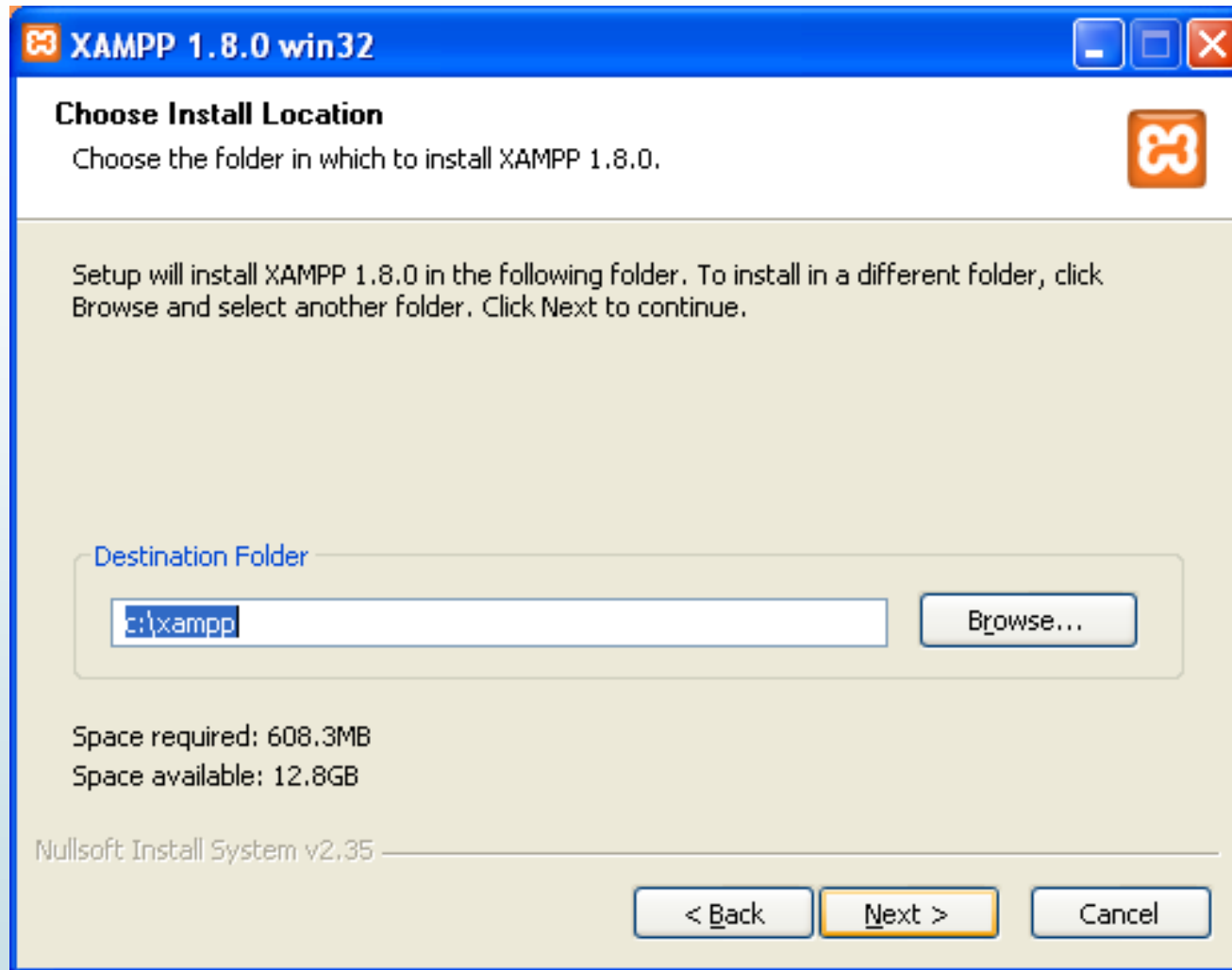
- Para descargar XAMPP accedemos a su web de descargas para Windows:

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

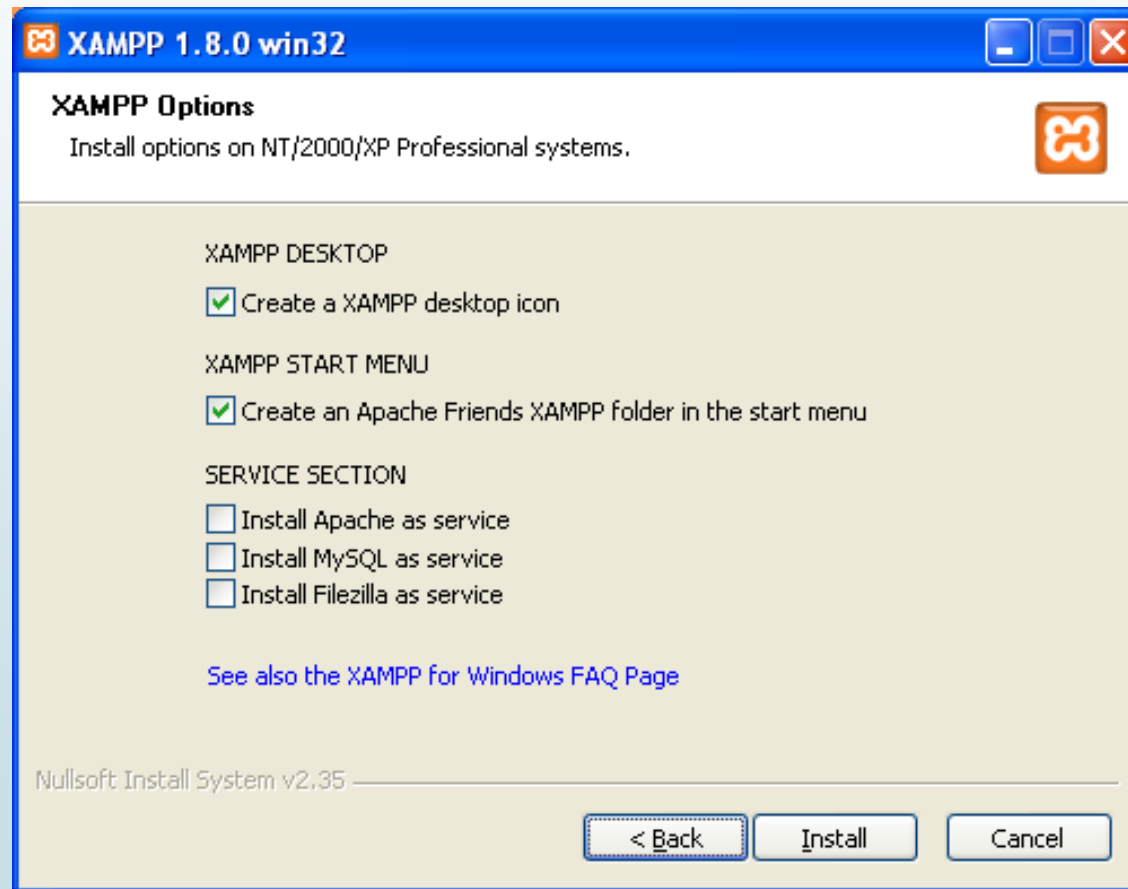
- Existe una versión portable:

<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/8.1.17/xampp-portable-windows-x64-8.1.17-0-VS16.zip/download>

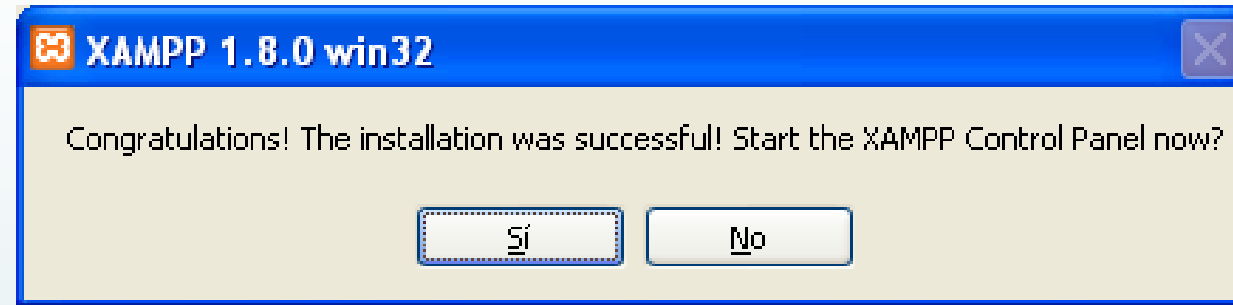
- Descargamos última versión disponible y siempre con un intérprete de php superior o igual a la versión 8.
- Ejecutamos el fichero. Tras seleccionar el lenguaje (English) y pulsar Next en la pantalla de bienvenida podremos ver la siguiente pantalla de instalación:



Pulsamos *Next* y aparecerá la ventana siguiente. Pulsamos el botón etiquetado con **Install**.



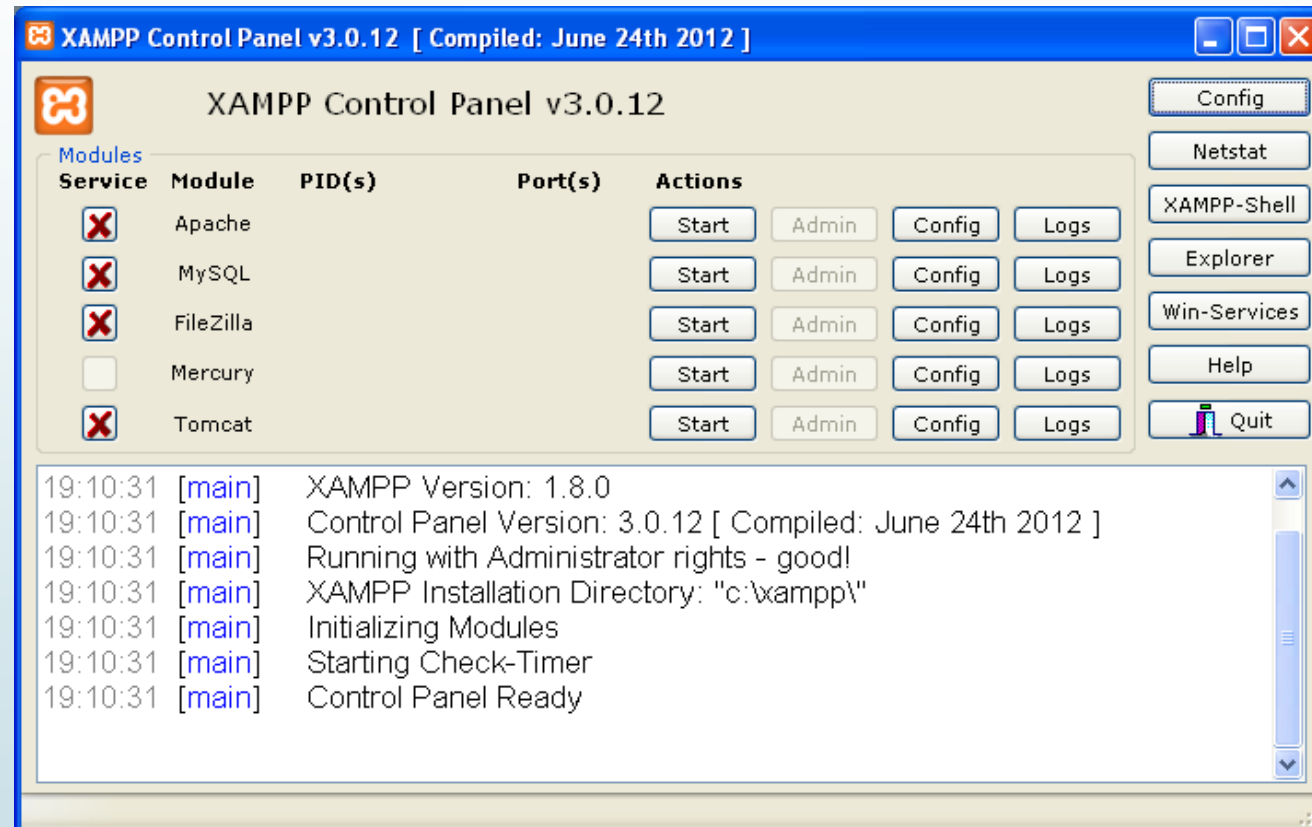
Esperamos a que nos salga la siguiente pantalla:



A continuación, seleccionamos:



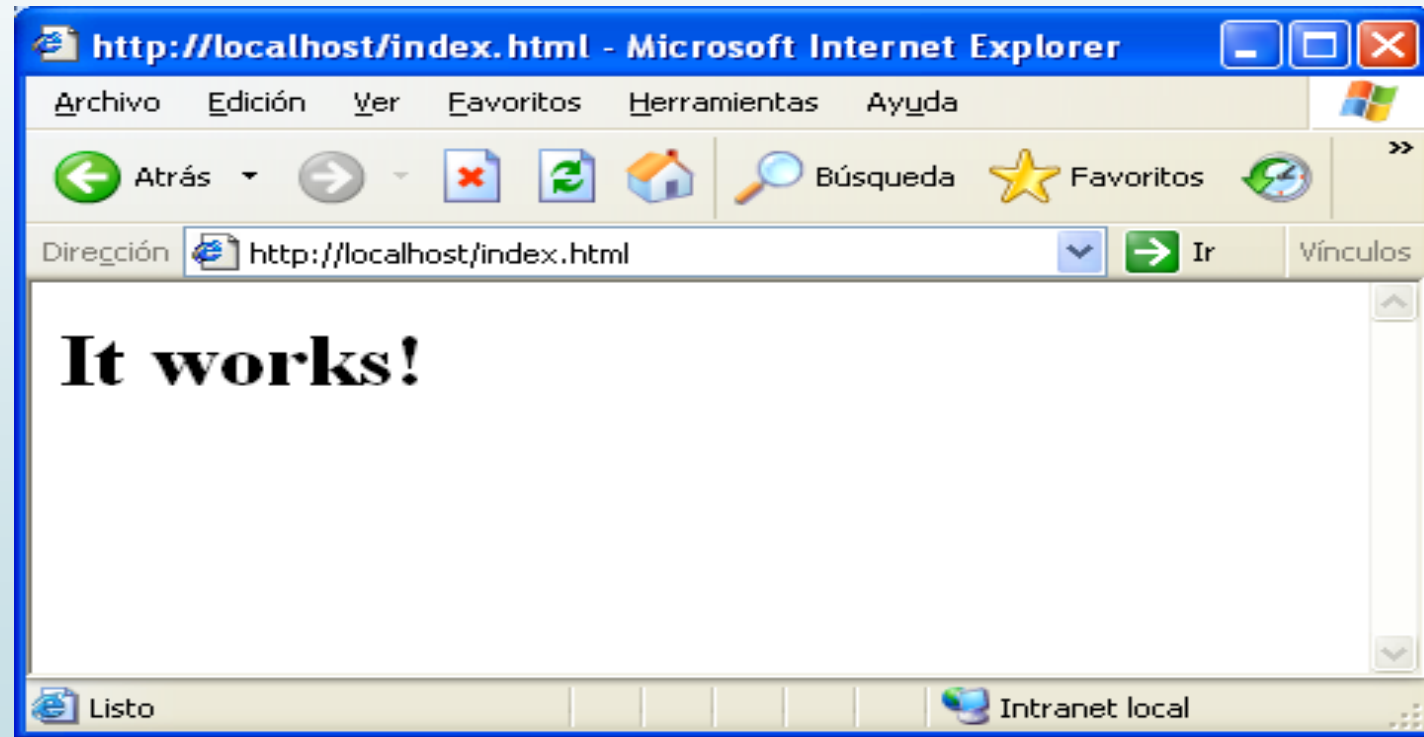
## Se nos abrirá el Panel de Control de Xampp



Pulsamos el botón Start para arrancar el Apache. Para probar que el servidor está funcionando abre tu navegador web favorito y accede a la siguiente dirección:

**<http://localhost/index.html>**

Debe salir:



## Script de prueba:

```
<?php  
    echo "<h1>¡HoLa Mundo!</h1>";  
?>
```

Ahora lo guardaremos en **C:\xampp\htdocs**

Este es el directorio raíz del servidor

El nombre del fichero debe ser **prueba1.php**



- Ya tenemos el fichero colocado en el directorio raíz de nuestro servidor web.
- Para acceder a él debemos abrir nuestro navegador web y hacer una petición de este fichero, que se encuentra en nuestra propia máquina.
- Esto lo hacemos mediante la IP 127.0.0.1, o mejor, mediante el sobrenombre localhost:  
  
<http://localhost/prueba1.php>
- Lo que ve en el navegador es el resultado de ejecución del fichero PHP.

## Ejercicio de Prueba

Su nombre:

Su edad:

Enviar consulta

## Ejercicio de Prueba

### formulario.html

```
<form action="respuesta.php" method="get">
  <p>Su nombre: <input type="text" name="nombre" /></p>
  <p>Su edad: <input type="text" name="edad" /></p>
  <p><input type="submit" value="Enviar consulta" /></p>
</form>
```

### respuesta.php

```
<?php

    var_dump($_GET);

?>
```