

# Tema 1. Selección de arquitecturas y herramientas de programación

---

DESARROLLO EN ENTORNO SERVIDOR 2º DAW

MARÍA CRIADO DOMÍNGUEZ

# Índice

---

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

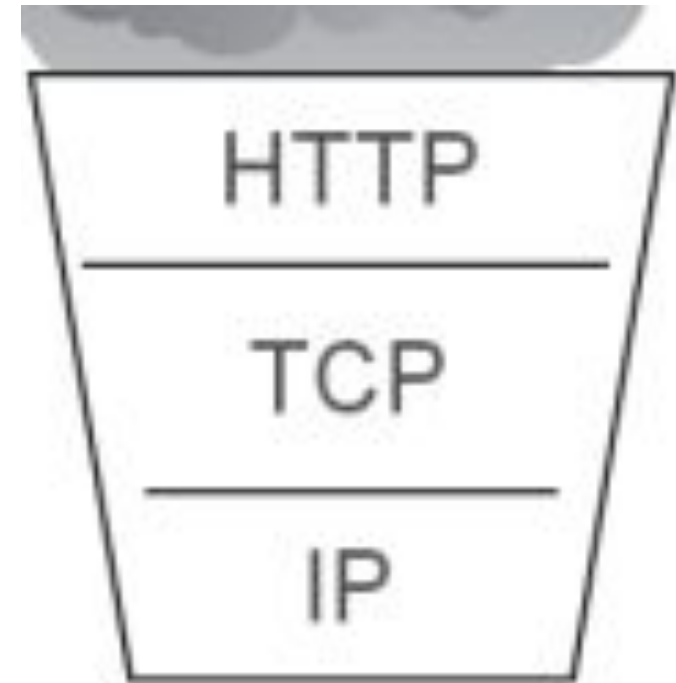
# Modelos de programación en entornos cliente/servidor

---

- La **World Wide Web** (o la Web, como se conoce comúnmente) representa un universo de información accesible globalmente a través de internet. Está formada por un conjunto de recursos interconectados que conforman el conocimiento humano actual.
- El funcionamiento de la Web es posible debido a la coexistencia de una serie de componentes software y hardware. Estos elementos abarcan desde los componentes físicos de internet (hubs, repetidores, puentes, pasarelas, encaminadores, etc.) y los protocolos de comunicaciones (TCP, IP, HTTP, FTP, SMTP, etc.) hasta la utilización del sistema de nombres de dominio (DNS) para la búsqueda y recuperación de recursos o la utilización de software específico para proveer y consumir dichos recursos.

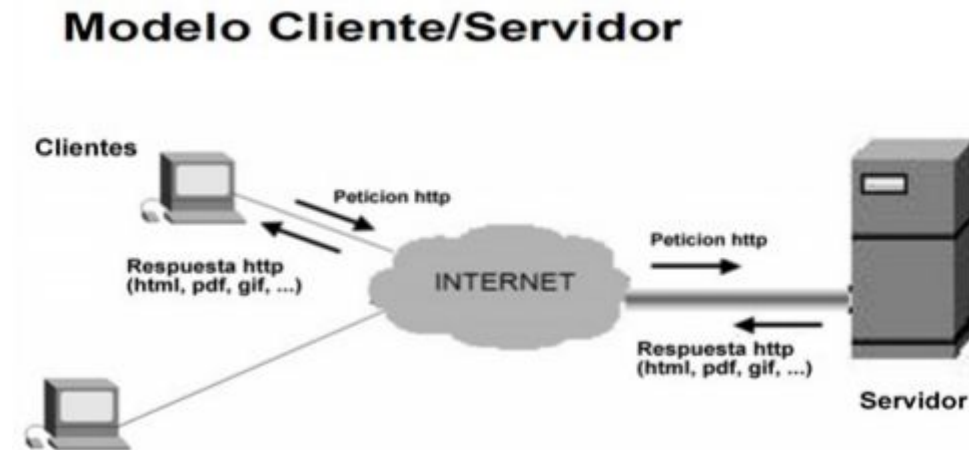
# Modelos de programación en entornos cliente/servidor

- El desarrollo en entornos web debe tener en cuenta la distribución de los elementos y la función que tiene cada uno de ellos. la configuración arquitectónica más habitual se basa en el modelo denominado Cliente/ Servidor
- Basado en la idea de servicio, en el que el cliente es un componente consumidor de servicios y el servidor es un proceso proveedor de servicios.
- Además, esta relación está robustamente cimentada en el intercambio de mensajes como el único elemento de acoplamiento entre ambos.



# Modelos de programación en entornos cliente/servidor

- El agente que solicita la información se denomina **cliente**, mientras que el componente software que responde a esa solicitud es el que se conoce como **servidor**. En un proceso habitual el cliente es el que inicia el intercambio de información, solicitando datos al servidor, que responde enviando uno o más flujos de datos al cliente.
- Además de la transferencia de datos real, este intercambio puede requerir información adicional, como la autenticación del usuario o la identificación del archivo de datos que vayamos a transferir



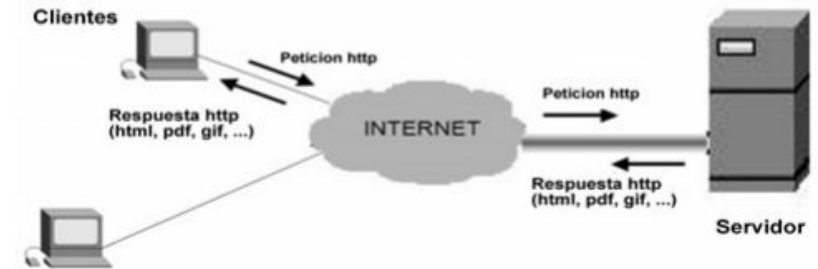
# Modelos de programación en entornos cliente/servidor

- Los pasos son los siguientes:
  1. Tu ordenador solicita a un servidor web una página con extensión .htm, .html .xhtml. .php .asp .jsp
  2. El servidor busca esa página en un almacén de páginas (cada una suele ser un fichero).
  3. Si el servidor encuentra esa página, la recupera.
  4. Y por último se la envía al navegador para que éste pueda mostrar su contenido.

Este es un ejemplo típico de una comunicación cliente-servidor.

El cliente es el que hace la petición e inicia la comunicación, y el servidor es el que recibe la petición y la atiende. En nuestro caso, el navegador es el cliente web.

**Modelo Cliente/Servidor**



# Generación dinámica de páginas web

---

## Páginas web estáticas

Estas páginas se encuentran almacenadas en su forma definitiva, tal y como se crearon, y su contenido no varía. Son útiles para mostrar una información concreta, y mostrarán esa misma información cada vez que se carguen. La única forma en que pueden cambiar es si un programador la modifica y actualiza su contenido.

## Páginas web dinámicas.

Estas páginas, como su nombre indica, se caracterizan porque su contenido cambia en función de diversas variables, como puede ser el navegador que estás usando, el usuario con el que te has identificado, o las acciones que has efectuado con anterioridad.

# Generación dinámica de páginas web





# Generación dinámica de páginas web

---

## Páginas web dinámicas.

### Cliente

Aquellas que incluyen código que ejecuta el navegador. En estas páginas el código ejecutable, normalmente en lenguaje JavaScript, se incluye dentro del HTML (o XHTML) y se descarga junto con la página.

Cuando el navegador muestra la página en pantalla, ejecuta el código que la acompaña. Este código puede incorporar múltiples funcionalidades que pueden ir desde mostrar animaciones hasta cambiar totalmente la apariencia y el contenido de la página.

# Generación dinámica de páginas web

---

## Páginas web dinámicas.

### Servidor

Hay muchas páginas en Internet que no tienen extensión .htm, .html o .xhtml. Muchas de estas páginas tienen extensiones como .php, .asp, .jsp, .cgi o .aspx.

En éstas, el contenido que se descarga al navegador es similar al de una página web estática: HTML (o XHTML). Lo que cambia es la forma en que se obtiene ese contenido.

Al contrario de lo que vimos hasta ahora, esas páginas no están almacenadas en el servidor; más concretamente, el contenido que se almacena no es el mismo que después se envía al navegador. El HTML de estas páginas se forma como resultado de la ejecución de un programa, y esa ejecución tiene lugar en el servidor web (aunque no necesariamente por ese mismo servidor).

# Generación dinámica de páginas web

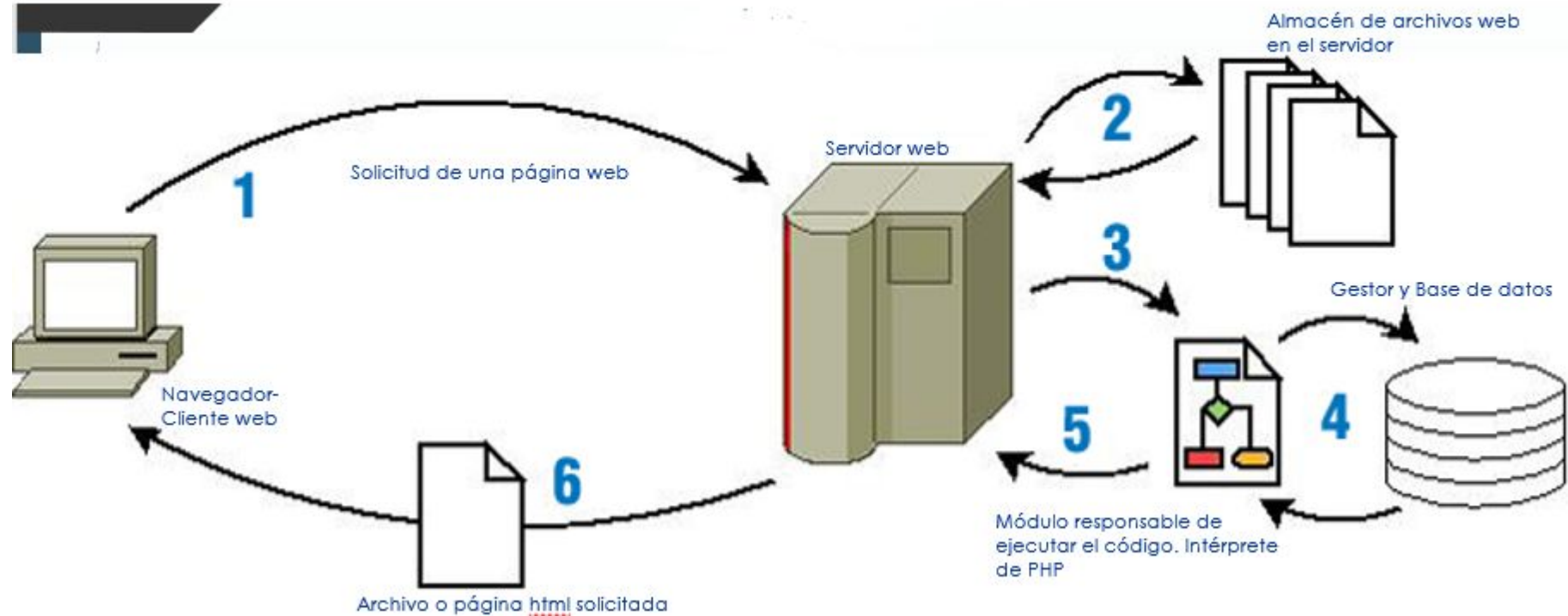
---

**Páginas web dinámicas.** El esquema de funcionamiento de una página web dinámica es el siguiente:

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, 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.

# Generación dinámica de páginas web

**Páginas web dinámicas.** El esquema de funcionamiento de una página web dinámica.



# Generación dinámica de páginas web

---

## Páginas web estáticas vs dinámicas.

Las primeras páginas web que se crearon en Internet fueron páginas estáticas. A esta web compuesta por páginas estáticas se le considera la primera generación.

La segunda generación de la web surgió gracias a las páginas web dinámicas.

Tomando como base las web dinámicas, han ido surgiendo otras tecnologías que han hecho evolucionar Internet hasta llegar a lo que ahora conocemos.

Las páginas web estáticas sólo necesitan un navegador para interpretarlas. Son archivos que pueden almacenarse en un soporte de almacenamiento como puede ser un disco óptico o una memoria USB y abrirse desde él directamente con un navegador web.

# Generación dinámica de páginas web

---

## Páginas web estáticas vs dinámicas.

Para que Google muestre un sitio web en sus resultados de búsqueda, previamente tiene que indexar su contenido. Es decir, un programa recorre las páginas del sitio consultando su contenido y clasificándolo.

Si las páginas se generan de forma dinámica, puede ser que su contenido, en parte o por completo, no sea visible para el buscador y por tanto no quedará indexado. Esto nunca sucedería en un sitio que utilizase páginas web estáticas.

# Generación dinámica de páginas web

---

## Aplicaciones web

Las aplicaciones web emplean páginas web dinámicas para crear aplicaciones que se ejecuten en un servidor web y se muestren en un navegador. Puedes encontrar aplicaciones web para realizar múltiples tareas.

Unas de las primeras en aparecer fueron los clientes de correo, que te permiten consultar los mensajes de correo recibidos y enviar los tuyos propios utilizando un navegador.

Hoy en día existen aplicaciones web para multitud de tareas como procesadores de texto, gestión de tareas, o edición y almacenamiento de imágenes. Estas aplicaciones tienen ciertas ventajas e inconvenientes si las comparas con las aplicaciones tradicionales que se ejecutan sobre el sistema operativo de la propia máquina.

# Generación dinámica de páginas web

---

## **Aplicaciones web** Ventajas de las aplicaciones web:

- No es necesario instalarlas en aquellos equipos en que se vayan a utilizar. Se instalan y se ejecutan solamente en un equipo, en el servidor, y esto es suficiente para que se puedan utilizar de forma simultánea desde muchos equipos.
- Como solo se encuentran instaladas en un equipo, es muy sencillo gestionarlas (hacer copias de seguridad de sus datos, corregir errores, actualizarlas).
- Se pueden utilizar en todos aquellos sistemas que dispongan de un navegador web, independientemente de sus características (no es necesario un equipo potente) o de su sistema operativo.
- Se pueden utilizar desde cualquier lugar en el que dispongamos de conexión con el servidor. En muchos casos esto hace posible que se pueda acceder a las aplicaciones desde sistemas no convencionales, como por ejemplo móviles



# Generación dinámica de páginas web

---

**Aplicaciones web** Inconvenientes de las aplicaciones web:

- El interface de usuario de las aplicaciones web es la página que se muestra en el navegador. Esto restringe las características del interface a aquellas de una página web.
- Dependemos de una conexión con el servidor para poder utilizarlas. Si nos falla la conexión, no podremos acceder a la aplicación web.
- La información que se muestra en el navegador debe transmitirse desde el servidor. Esto hace que cierto tipo de aplicaciones no sean adecuadas para su implementación como aplicación web (por ejemplo, las aplicaciones que manejan contenido multimedia, como las de edición de vídeo)

# Generación dinámica de páginas web

## Ejecución de código

---

- Cuando tu 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.

# Generación dinámica de páginas web

## Ejecución de código

---

- Estas dos tecnologías se complementan una con otra.
- Esta división es así porque el código que se ejecuta en el cliente web (en el navegador) no tiene, o mejor dicho tradicionalmente no tenía, acceso a los datos que se almacenan en el servidor.
- Desde hace unos años existe una técnica de desarrollo web conocida como AJAX, que nos posibilita realizar programas en los que el código JavaScript que se ejecuta en el navegador pueda comunicarse con un servidor de Internet para obtener información con la que, por ejemplo, modificar la página web actual

# Tecnologías para programación web del lado del servidor.

---

Los **componentes** principales con los que debes contar para ejecutar aplicaciones web en un servidor son los siguientes:

- Un **servidor web** para recibir las peticiones de los clientes web (normalmente navegadores) y enviarles 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.
- 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.
- 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.
- El **lenguaje de programación** que utilizarás para desarrollar las aplicaciones.

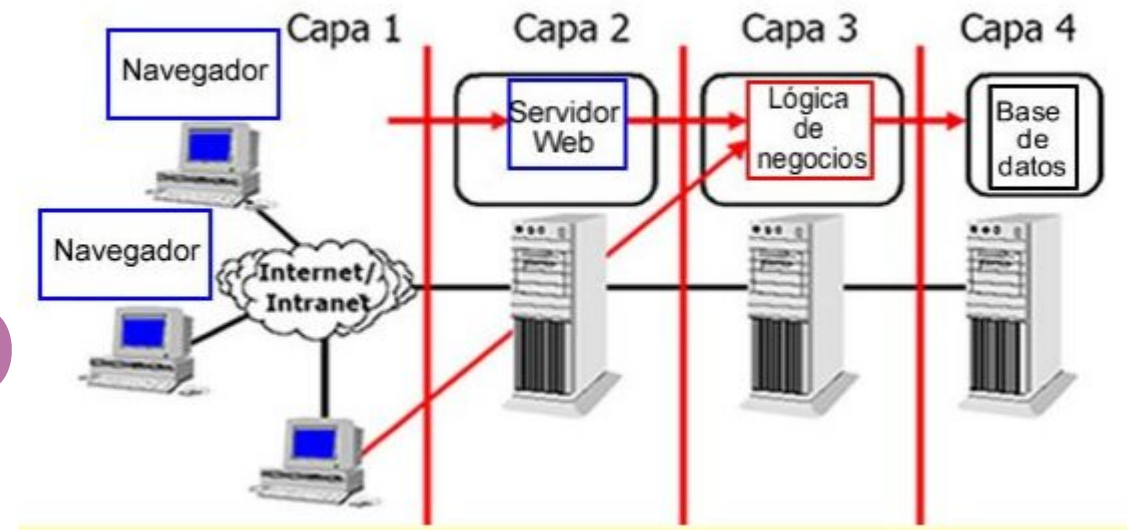
# Tecnologías para programación web del lado del servidor.

---

- También es importante decidir cómo vas a organizar el código de la aplicación. Muchas de las arquitecturas que se usan en la programación de aplicaciones web te ayudan a estructurar el código de las aplicaciones en capas o niveles.
- El motivo de dividir en capas el diseño de una aplicación es que se puedan separar las funciones lógicas de la misma, de tal forma que sea posible ejecutar cada una en un servidor distinto

# Tecnologías para programación web del lado del servidor.

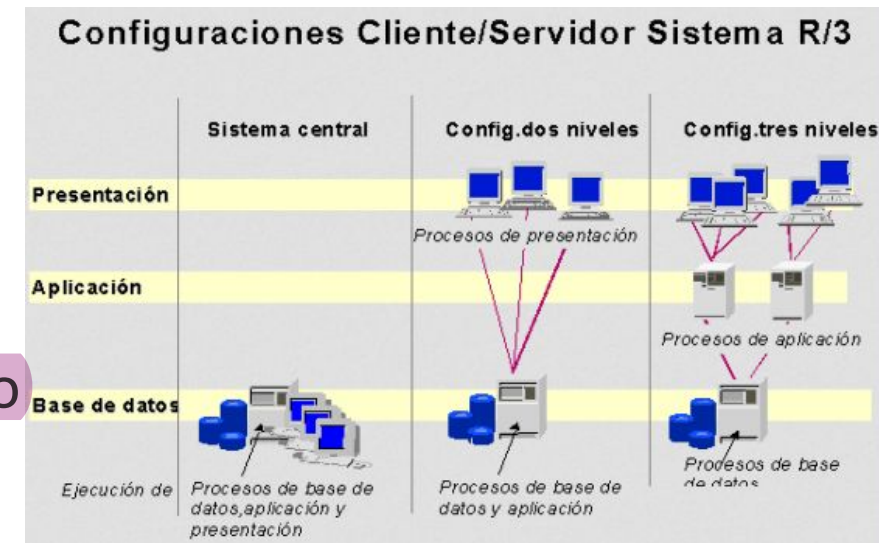
- En una aplicación puedes distinguir, de forma general:
- Funciones de presentación (se encarga de dar formato a los datos para presentárselo al usuario final)
- Lógica (utiliza los datos para ejecutar un proceso y obtener un resultado),
- Persistencia (que mantiene los datos almacenados de forma organizada) y acceso (que obtiene e introduce datos en el espacio de almacenamiento).



# Tecnologías para programación web del lado del servidor.

Cada capa puede ocuparse de una o varias de las funciones anteriores. Por ejemplo, en las aplicaciones de 3 capas nos podemos encontrar con:

- Una **capa cliente**, que es donde programarás todo lo relacionado con el interface de usuario, esto es, la parte visible de la aplicación con la que interactuará el usuario.
- Una **capa intermedia** donde deberás programar la funcionalidad de tu aplicación.
- Una **capa de acceso a datos**, que se tendrá que encargar de almacenar la información de la aplicación en una base de datos y recuperarla cuando sea necesario.



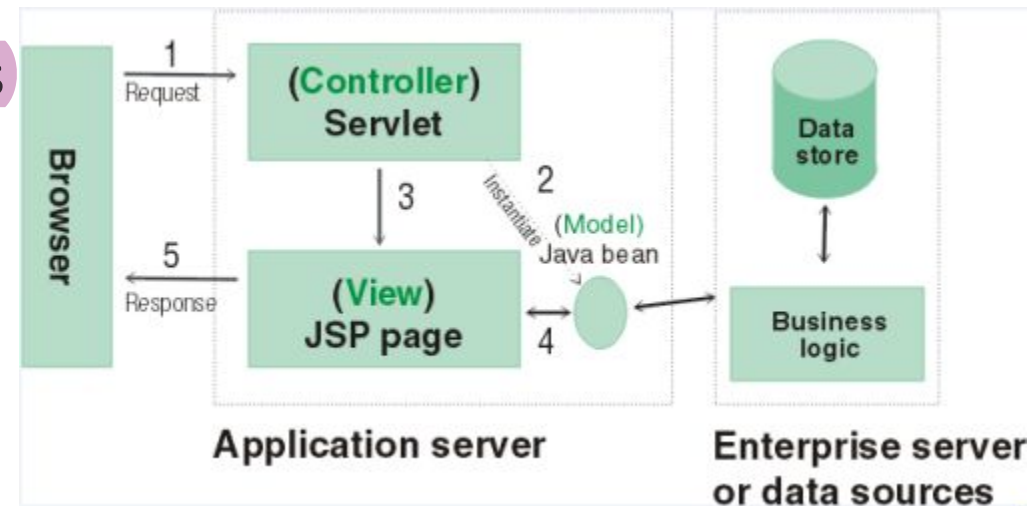


# Tecnologías para programación en el servidor.

## Arquitectura y Plataformas

### Java EE

- Es una plataforma orientada a la programación de aplicaciones en lenguaje Java. Puede funcionar con distintos gestores de bases de datos, e incluye varias librerías y especificaciones para el desarrollo de aplicaciones de forma modular. Está apoyada por grandes empresas como Sun y Oracle, que mantienen Java, o IBM.
- Es una buena solución para el desarrollo de aplicaciones de tamaño mediano o grande. Una de sus principales ventajas es la multitud de librerías existentes en ese lenguaje y la gran base de programadores que lo conocen.
- Dentro de esta arquitectura existen distintas tecnologías como las páginas JSP y los servlets, ambos orientados a la generación dinámica de páginas web, o los EJB, componentes que normalmente aportan la lógica de la aplicación web.





# Tecnologías para programación en el servidor.

## Arquitectura y Plataformas

---

### AMP

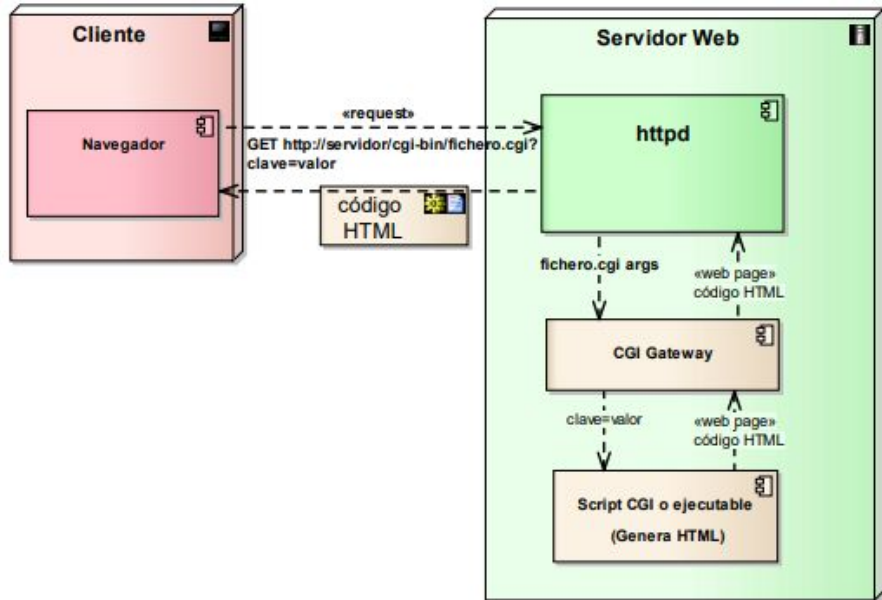
- Son las siglas de Apache, MySQL y PHP/Perl/Python. Las dos primeras siglas hacen referencia al servidor web (Apache) y al servidor de base de datos (MySQL). La última se corresponde con el lenguaje de programación utilizado, que puede ser PHP, Perl o Python,
- Dependiendo del sistema operativo que se utilice para el servidor, se utilizan las siglas LAMP (para Linux), WAMP (para Windows) o MAMP (para Mac).
- Todos los componentes de esta arquitectura son de código libre (open source). Es una plataforma de programación que permite desarrollar aplicaciones de tamaño pequeño o mediano con un aprendizaje sencillo. Su gran ventaja es la gran comunidad que la soporta y la multitud de aplicaciones de código libre disponibles

# Tecnologías para programación en el servidor.

## Arquitectura y Plataformas

### CGI

- Es la combinación de dos componentes: Perl, un potente lenguaje de código libre creado originalmente para la administración de servidores, y CGI, un estándar para permitir al servidor web ejecutar programas genéricos, escritos en cualquier lenguaje (también se pueden utilizar por ejemplo C o Python), que devuelven páginas web (HTML) como resultado de su ejecución. Es la más primitiva de las arquitecturas que comparamos aquí.



# Tecnologías para programación en el servidor.

## Arquitectura y Plataformas

---



### ASP.

- .Net es la arquitectura comercial propuesta por Microsoft para el desarrollo de aplicaciones. Es la parte de la plataforma .Net destinada a la generación de páginas web dinámicas. Proviene de la evolución de la anterior tecnología de Microsoft, ASP.
- El lenguaje de programación puede ser Visual Basic.Net o C#. La arquitectura utiliza el servidor web de Microsoft, IIS, y puede obtener información de varios gestores de bases de datos entre los que se incluye, como no, Microsoft SQL Server.
- Una de las mayores ventajas de la arquitectura .Net es que incluye todo lo necesario para el desarrollo y el despliegue de aplicaciones. Por ejemplo, tiene su propio entorno de desarrollo, Visual Studio, aunque hay otras opciones disponibles. La mayor desventaja es que se trata de una plataforma comercial de código propietario.

# Tecnologías para programación en el servidor.

## Arquitectura y Plataformas

---

### Selección de una arquitectura de programación web

Como has visto, hay muchas decisiones que debes tomar antes aún de comenzar el desarrollo de una aplicación web:

- La arquitectura
- El lenguaje de programación
- El entorno de desarrollo
- El gestor de bases de datos
- El servidor web
- La estructura de la aplicación

# Tecnologías para programación en el servidor.

## Arquitectura y Plataformas

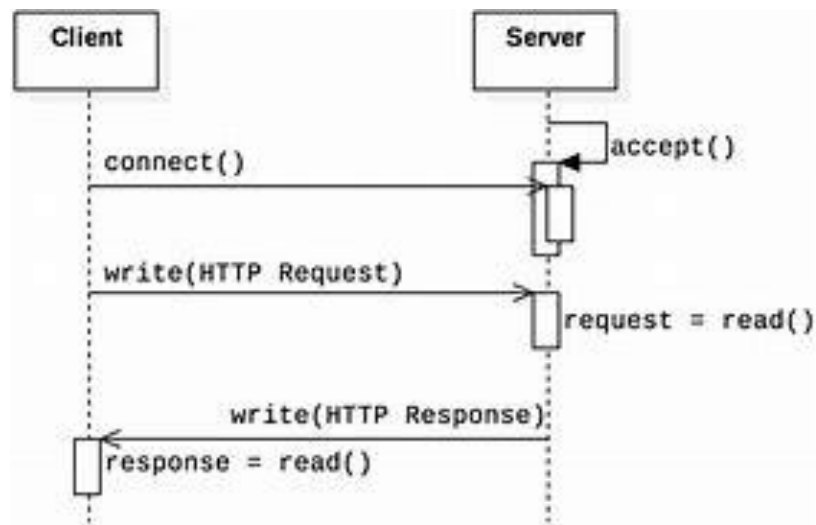
---

Para tomar una decisión correcta, deberás considerar entre otros los siguientes puntos:

- ¿Qué tamaño tiene el proyecto?
- ¿Qué lenguajes de programación conozco? ¿Vale la pena el esfuerzo de aprender uno nuevo?
- ¿Voy a usar herramientas de código abierto o herramientas propietarias? ¿Cuál es el coste de utilizar soluciones comerciales?
- ¿Voy a programar la aplicación yo solo o formaré parte de un grupo de programadores?
- ¿Cuento con algún servidor web o gestor de base de datos disponible o puedo decidir libremente utilizar el que crea necesario?
- ¿Qué tipo de licencia voy a aplicar a la aplicación que desarrolle?

# Tecnologías para programación en el servidor.

## Integración con el servidor web



- La comunicación entre un cliente web o navegador y un servidor web se lleva a cabo gracias al protocolo **HTTP**.
- En el caso de las aplicaciones web, HTTP es el vínculo de unión entre el usuario y la aplicación en sí.
- Cualquier introducción de información que realice el usuario se transmite mediante una petición HTTP, y el resultado que obtiene le llega por medio de una respuesta HTTP
- En el lado del servidor, estas peticiones son procesadas por el servidor web (también llamado servidor HTTP). Es por tanto el servidor web el encargado de decidir cómo procesar las peticiones que recibe.

# Tecnologías para programación en el servidor.

## Integración con el servidor web

---

- La tecnología más antigua es **CGI**. CGI es un protocolo estándar que existe en muchas plataformas. Lo implementan la gran mayoría de servidores web. Define qué debe hacer el servidor web para delegar en un programa externo la generación de una página web.
- Esos programas externos se conocen como **guiones CGI**, independientemente del lenguaje en el que estén programados (aunque se suelen programar en lenguajes de guiones como Perl).
- El **principal problema** de CGI es que cada vez que se ejecuta un guión CGI, el sistema operativo debe crear un nuevo proceso. Esto implica un mayor consumo de recursos y menor velocidad de ejecución.

# Tecnologías para programación en el servidor.

## Integración con el servidor web

---

- La arquitectura **Java EE** es más compleja. Para poder ejecutar aplicaciones Java EE en un servidor básicamente tenemos dos opciones: servidores de aplicaciones, que implementan todas las tecnologías disponibles en Java EE, y contenedores de servlets, que soportan solo parte de la especificación.
- Dependiendo de la magnitud de nuestra aplicación y de las tecnologías que utilice, tendremos que instalar una solución u otra.
- En la mayoría de ocasiones no es necesario utilizar un servidor de aplicaciones completo, sino que nos será suficiente un contenedor de servlets. En esta área, destaca **Tomcat**, la implementación por referencia de un contenedor de servlets, que además es de código abierto



# Tecnologías para programación en el servidor.

## Integración con el servidor web

---

- La arquitectura ASP.Net utiliza el servidor IIS de Microsoft, que ya integra soporte en forma de módulos para manejar peticiones de páginas dinámicas ASP y ASP.Net.
- La utilidad de administración del servidor web incluye funciones de administración de las aplicaciones web instaladas en el mismo.

# Lenguajes

---

Una de las diferencias más notables entre un lenguaje de programación web y otro es la manera en que se ejecutan en el servidor web. Debes distinguir tres grandes grupos:

- Lenguajes de guiones (scripting).
- Lenguajes compilados a código nativo
- Lenguajes compilados a código intermedio.

# Lenguajes

---

## Lenguajes de guiones (scripting)

- Son aquellos en los que los programas se ejecutan directamente a partir de su código fuente (Conjunto de instrucciones que componen un programa, y que no son ejecutables directamente, sino que deben traducirse utilizando un compilador, intérprete o similar antes de que pueda ser ejecutado por la máquina) original.
- Se almacenan normalmente en un fichero de texto plano. Cuando el servidor web necesita ejecutar código programado en un lenguaje de guiones, le pasa la petición a un intérprete, que procesa las líneas del programa y genera como resultado una página web.
- De los lenguajes que estudiaste anteriormente, pertenecen a este grupo Perl, Python, PHP y ASP.

# Lenguajes

---

## Ventajas e inconvenientes

Los lenguajes de guiones tienen la ventaja de que no es necesario traducir el código fuente original para ser ejecutados, lo que aumenta su portabilidad. Si se necesita realizar alguna modificación a un programa, se puede hacer en el momento.

Por el contrario el proceso de interpretación ofrece un peor rendimiento que las otras alternativas.

# Lenguajes

---

## Lenguajes compilados a código nativo

- Son aquellos en los que el código fuente se traduce a código binario, dependiente del procesador, antes de ser ejecutado.
- El servidor web almacena los programas en su modo binario, que ejecuta directamente cuando se les invoca.
- El método principal para ejecutar programas binarios desde un servidor web es CGI.
- Utilizando CGI podemos hacer que el servidor web ejecute código programado en cualquier lenguaje de propósito general como puede ser C.

# Lenguajes

---

## Ventajas e inconvenientes

Los lenguajes compilados a código nativo son los de mayor velocidad de ejecución, pero tienen problemas en lo relativo a su integración con el servidor web. Son programas de propósito general que no están pensados para ejecutarse en el entorno de un servidor web.

Por ejemplo, no se reutilizan los procesos para atender a varias peticiones: por cada petición que se haga al servidor web, se debe ejecutar un nuevo proceso. Además los programas no son portables entre distintas plataformas

# Lenguajes

---

## Lenguajes compilados a código intermedio

- Son lenguajes en los que el código fuente original se traduce a un código intermedio, independiente del procesador, antes de ser ejecutado
- Es la forma en la que se ejecutan por ejemplo las aplicaciones programadas en Java, y lo que hace que puedan ejecutarse en varias plataformas distintas.
- En la programación web, operan de esta forma los lenguajes de las arquitecturas Java EE (servlets y páginas JSP)

# Lenguajes

---

## Ventajas e inconvenientes

Los lenguajes compilados a código intermedio ofrecen un equilibrio entre las dos opciones anteriores. Su rendimiento es muy bueno y pueden portarse entre distintas plataformas en las que exista una implementación de la arquitectura (como un contenedor de servlets o un servidor de aplicaciones Java EE)



# Código embebido en html

---

Cuando la web comenzó a evolucionar desde las páginas web estáticas a las dinámicas, una de las primeras tecnologías que se utilizaron fue la ejecución de código utilizando CGI. Los guiones CGI son programas estándar, que se ejecutan por el sistema operativo, pero que generan como salida el código HTML de una página web. Por tanto, los guiones CGI deben contener, mezcladas dentro de su código, sentencias encargadas de generar la página web.

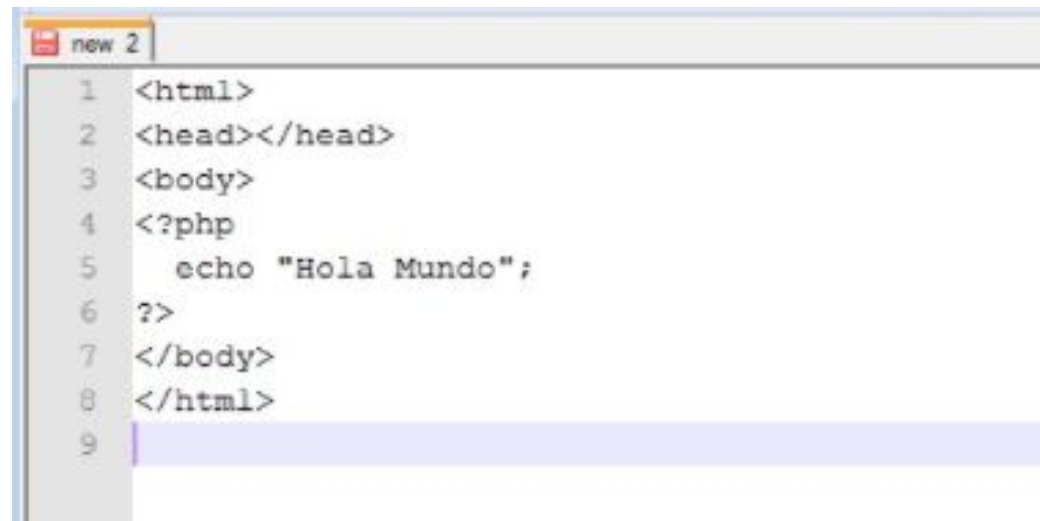
```
echo "Content-Type: text/html"
echo
echo

echo "<html><head></head>"
echo "<body>"
echo "Parameters are:<br />"
user=`echo $QUERY_STRING | cut -d"&" -f 1 | cut -d"=" -f 2`
pass=`echo $QUERY_STRING | cut -d"&" -f 2 | cut -d"=" -f 2`

echo $user $pass
echo "</body></html>"
```

# Código embebido en html

Un enfoque distinto consiste en integrar el código del programa en medio de las etiquetas HTML de la página web. De esta forma, el contenido que no varía de la página se puede introducir directamente en HTML, y el lenguaje de programación se utilizará para todo aquello que pueda variar de forma dinámica.



```
new 2
1 <html>
2 <head></head>
3 <body>
4 <?php
5     echo "Hola Mundo";
6 ?>
7 </body>
8 </html>
9
```

# Herramientas de programación

---

- Existen entornos integrados de desarrollo (IDE) que agrupan en un único programa muchas de estas herramientas.
- Algunos de estos entornos de desarrollo son específicos de una plataforma o de un lenguaje, como sucede por ejemplo con Visual Studio, el IDE de Microsoft para desarrollar aplicaciones en lenguaje C# o Visual Basic para la plataforma .Net.
- Otros como Eclipse o NetBeans te permiten personalizar el entorno para trabajar con diferentes lenguajes y plataformas, como Java EE o PHP.
- No es imprescindible utilizar un IDE para programar. En muchas ocasiones puedes echar mano de un simple editor de texto para editar el código que necesites.

# Preparación del entorno

---

Visual Studio Code

<https://code.visualstudio.com/>

Ubuntu server

<https://ubuntu.com/download/server>

Php

<https://www.php.net/downloads>

Apache

<https://httpd.apache.org/>