

# IES Velázquez



## Desarrollo de Aplicaciones Web Desarrollo Web en Entorno Servidor Ciclo Formativo de Grado Superior

# ÍNDICE:Tema1.Arquitecturas y tecnologías de programación web

---

## 1. MECANISMOS DE EJECUCIÓN DE CÓDIGO.

### 1.1. El Servidor Web

#### 1.1.1. Integración del lenguaje de marcas con los servidores web

### 1.2. Servidor de aplicaciones

# MECANISMOS DE EJECUCIÓN DE CÓDIGO.

---

Se entiende por **lenguaje de programación** correspondiente a un **entorno servidor** a **aquel cuyo código**, bien sea como objeto precompilado o bien como código interpretado, **es ejecutado por un software específico en el componente que actúa como servidor**.

Así como *las aplicaciones de escritorio se ejecutan directamente sobre el propio Sistema Operativo*, *las páginas y aplicaciones web necesitan de una herramienta adicional que permita desplegarlas para su puesta en marcha*. Hablamos de **servidores web y servidores de aplicaciones**, **respectivamente** (en clientes ya hemos visto que teníamos los navegadores).

# MECANISMOS DE EJECUCIÓN DE CÓDIGO.

---

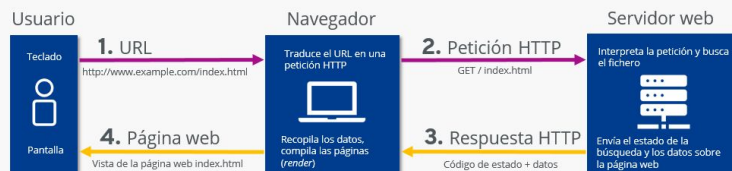
## Alternativas a la hora de ejecutar código en el servidor.

- **Scripts del lado del servidor:** Una de ellas es utilizar lenguajes de scripting (SSI, LiveWire, ASP, PHP, Perl, Python, etc.), cuya característica principal es que **el código es interpretado y que se intercala con una plantilla de código HTML** con la estructura básica de la página que se envía al cliente. Se **utiliza un módulo en el servidor web** (generalmente un intérprete), en lugar de un programa externo. Este módulo (*scripting engine*) **se encarga de ejecutar el código y pasarlo a HTML**.
- **Enlaces a programas y componentes ejecutables** (CGI, JSP, EJB, etc.), de tal forma que **el código ejecutado por el servidor está almacenado en unidades precompiladas y ejecutadas de forma independiente**, que generan las páginas enviadas al cliente.
- **Uso de estrategias “híbridas”** basadas en técnicas de respaldo (denominadas *code-behind*), como ASP.Net de Microsoft, en el que **algunos elementos de código se intercalan con la lógica** de presentación mientras que se **mantiene la funcionalidad de dichos elementos en ficheros o librerías independientes en el servidor**.
- **Servlets:** Los servlets de Java son **objetos que reciben una petición y devuelven una respuesta según la solicitud**. Para utilizar servlets, es **necesario contar con un contenedor web** que interactúe con ellos.

# El Servidor Web

Un servidor web es una aplicación que recibe una petición HTTP (normalmente a través de un navegador web) y devuelve la página web solicitada (escrita en lenguaje HTML y pudiendo contener código Javascript inscrustado) para que ésta sea interpretada y visualizada por el navegador de quién realizó la solicitud (el usuario).

## El proceso de comunicación según HTTP



# El Servidor Web

---

Los servidores web más utilizados en el mercado que podemos instalar son:

Los privativos:

Microsoft Internet Information Services (IIS): Un servidor web de Microsoft para sistemas operativos Windows.  
WebLogic Server, de Oracle.

De código abierto (open source):

El Servidor HTTP Apache.

El producto más implantado es Apache Web Server (<https://httpd.apache.org/>), creado en 1995.

- Es un software libre y multiplataforma

- Es un sistema de módulos dinámicos (se le pueden añadir módulos para incorporar diferentes lenguajes como PHP, Python o Perl).

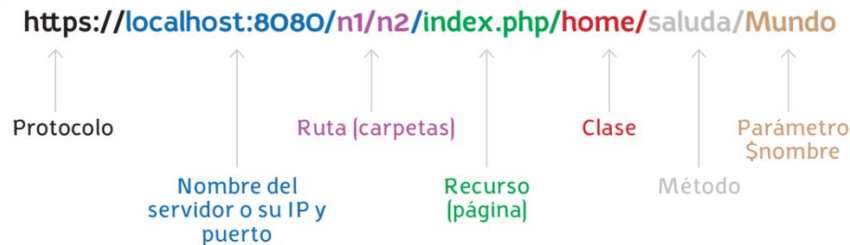
- Utiliza el archivo .htaccess para su configuración.

En la **actualidad**, Apache está perdiendo mercado respecto a **Nginx** (<https://www.nginx.com>). Es un servidor web **ligero y de alto rendimiento** utilizado comúnmente como proxy inverso o para servir contenido estático.

# Integración del lenguaje de marcas con los servidores web.

La **funcionalidad básica de un servidor web** es la de **proveer de contenido a un cliente** que ha realizado una petición a través de un navegador y para ello carga un archivo y lo sirve a través de la Red al navegador del solicitante. Es decir, que **el servidor** se convierte en un instrumento que **proporciona un lugar para guardar y administrar los recursos HTML**, que pueden ser accesibles por los usuarios de la Red a través de navegadores.

Para que el servidor pueda entender la petición del **cliente**, éste **tiene que realizar una petición “formal”**. Es decir, la petición **tiene que constar de unos elementos concretos y especificados en un orden determinado**. Las direcciones de las peticiones suelen ser de tipo URL (Localizador Uniforme de Recurso), que contiene una dirección, la referencia a un cierto protocolo (HTTP, FTP, etc.) y la descripción de un recurso concreto en forma de ruta al objeto que queremos descargar. Opcionalmente podemos incluir un campo adicional (puerto) indicando el número de puerto por el que el servidor está escuchando las peticiones del cliente.



Elementos de una URL completa que podemos encontrarnos cuando utilizamos un framework.

# Integración del lenguaje de marcas con los servidores web.

---

Con respecto a las **peticiones de los clientes** es importante destacar que existen **distintos modos o métodos para intercambiar información entre cliente y servidor** y que deberemos **tenerlo en cuenta en el desarrollo de aplicaciones web** en el entorno del servidor.

**Método GET:** es un método de invocación en el que **el cliente le solicita al servidor web que le devuelva la información identificada en la propia URL**. Se modifica *la URL para añadir tantos pares de variable o valor como hagan falta*. *Detrás de la URL añadiremos esa información como texto claro, detrás de un símbolo de interrogación «?».*

*Por ejemplo, si pedimos al buscador de Google que nos muestre información de “Comida”, generaría la URL <https://www.google.es/search?q=Comida>, donde q es la variable y, detrás del igual, su valor; en este caso, «Comida».*

Lo más común es que las peticiones se refieran a un documento HTML o a una imagen, aunque también se puede referir a un programa de base de datos. En tal caso, el servidor ejecuta ese programa y le devuelve al cliente el resultado generado tras esa petición.



# Integración del lenguaje de marcas con los servidores web.

**Método POST:** mientras que el método GET se utiliza para recuperar información, el método POST se usa habitualmente **para enviar información a un servidor web.**

*No se necesita modificar la URL.* **Aporta discreción**, al no guardar información en el cliente, **y flexibilidad**, al permitir enviar textos cortos, pero también podemos enviar imágenes o videos.

*Estos casos suelen darse al enviar el contenido de un formulario de autenticación, así como entradas de datos o especificar parámetros para algún tipo de componente ejecutado en el servidor.*

Como ya se ha comentado anteriormente, **la potencia de los servidores web aparece en el momento en que queremos desarrollar aplicaciones web interactivas y con una cierta complejidad.** En ese caso, podemos utilizar varias tecnologías implementadas en módulos específicos en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML. Dichas tecnologías suelen instalarse como extensiones en el software del servidor e incluyen soporte para scripts CGI, proporcionan seguridad SSL, permiten la ejecución de scripts, interactúan con la máquina virtual de Java, etc.

# Servidor de aplicaciones.

---

Los **servidores de aplicaciones** proporcionan un **entorno de ejecución para las aplicaciones**, con el **objetivo de liberar al programador de algunas tareas relacionadas con la infraestructura de la aplicación, como la seguridad o el equilibrio de carga**. Por lo general, **también incluyen un servidor web**.

Un servidor de aplicaciones es una aplicación que contiene una serie de servicios los cuales están accesibles a través de una API expuesta a través de Internet. **Normalmente los servidores de aplicaciones proporcionan más servicios que los servidores web**.

Por ejemplo, en el caso de los servidores de aplicaciones para Java o Python, éstos proporcionan un acceso transparente a la Base de Datos para que el desarrollador se centre exclusivamente en implementar la capa de negocio. Además, pueden proporcionar también servicios como fail-over o balanceo de carga, clustering o tolerancia a fallos.

Entre los servidores de aplicaciones más conocidos se encuentran los servidores de aplicaciones para Java EE (Enterprise Edition), como JBoss o WebSphere. Estos servidores **cuentan con contenedores para ejecutar los componentes JEE**, como contenedores de servlets y Java Enterprise Beans (EJB) o persistencia. Sin embargo, **no todos los servidores brindan soporte para todos los componentes**.

Otro ejemplo es Apache Tomcat (<http://tomcat.apache.org/>). Tomcat es el servidor de aplicaciones **open source y multiplataforma de referencia para una arquitectura Java**. Contiene un contenedor Web Java que interpreta Servlets y JSP. Actualmente es capaz de implementar varias especificaciones de Java EE como Servlets y JSP (Java Server Pages) y además proporciona un servidor web puro para que se use en combinación con el entorno Java

# Repasamos conceptos

---

Realiza el formulario que se plantea eligiendo la respuesta correcta.

[Formulario Mecanismos de Ejecución de Código](#)