

# **TEMA 1: Arquitecturas y tecnologías web**

Módulo

Despliegue de aplicaciones web

para los ciclos

Desarrollo de Aplicaciones Web



Despliegue FP-GS; Tema1:ArquitecturasYTecnologiasWeb

© Gerardo Martín Esquivel, Septiembre de 2023

Algunos derechos reservados.

Este trabajo se distribuye bajo la Licencia "Reconocimiento-No comercial-Compartir igual 3.0 Unported" de Creative Commons disponible en <http://creativecommons.org/licenses/by-nc-sa/3.0/>

<b>1.1 Aplicaciones web.....</b>	<b>3</b>
<b>1.2 Arquitecturas web.....</b>	<b>3</b>
1.2.1 Servidor web de contenido estático:.....	3
1.2.2 Servidor web de contenido dinámico.....	4
1.2.3 Servidor web de contenido dinámico con acceso a base de datos.....	4
1.2.4 Contenedores de servlet/JSP.....	4

## 1.1 Aplicaciones web

Las **aplicaciones web** son **aplicaciones distribuidas a las que accede el usuario a través de un navegador**.

El concepto de aplicación distribuida hace referencia a que se ejecuta sobre varias máquinas simultáneamente. Unas partes de la aplicación se ejecutan en una máquina y otras partes en otra. Generalmente se trata de un cliente y un servidor.

Por ejemplo, cuando estás consultando los movimientos de tu cuenta bancaria, estás usando una aplicación web. Una parte de esa aplicación web se ejecuta en un **servidor** que se encuentra en las instalaciones del banco. Otra parte de esa aplicación se ejecuta en tu ordenador de casa, que hace de **cliente**. Por ese motivo se considera que es distribuida. Además, la forma en la que tú interactúas con esa aplicación es a través de un navegador web (**Firefox, Chrome, Opera, Safari, Edge**, etc). Por tanto, se trata de una aplicación web.

El **despliegue de aplicaciones web** es el proceso de transferir y configurar recursos y aplicaciones a un servidor para su posterior ejecución. Esto se puede hacer de forma manual o automática.

## 1.2 Arquitecturas web

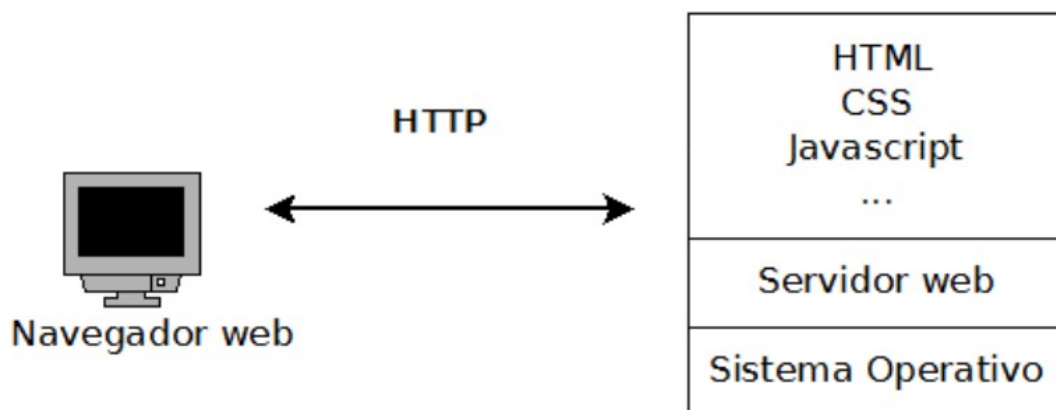
La arquitectura sobre la que montamos nuestra aplicación web puede ser simple o puede ser tan compleja como queramos. En las siguientes imágenes podemos ver distintas configuraciones desde la más simple a la más compleja:

### 1.2.1 Servidor web de contenido estático:

Es el modelo más sencillo para publicar una página web. En un equipo servidor instalamos un **servidor web** que es un programa que permanece a la escucha de peticiones mediante el protocolo **HTTP** (HyperText Transfer Protocol) y responde con el envío de ficheros de texto que contienen código **HTML, CSS, Javascript**, etc.

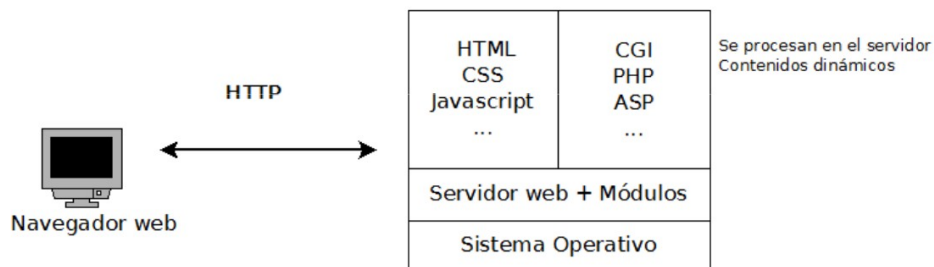
En el equipo cliente tenemos instalado un navegador web que lanza las peticiones **HTTP** e interpreta los ficheros que recibe como respuesta a esas peticiones.

Ejemplos de software de servidor web: **Apache** y **Microsoft IIS**.



### 1.2.2 Servidor web de contenido dinámico

Si a la estructura anterior le añadimos los módulos necesarios para interpretar lenguajes de script de servidor, como **PHP** o **ASP**, podrá servir contenido dinámico, es decir, páginas cuyo contenido se genera a partir de las peticiones del cliente.



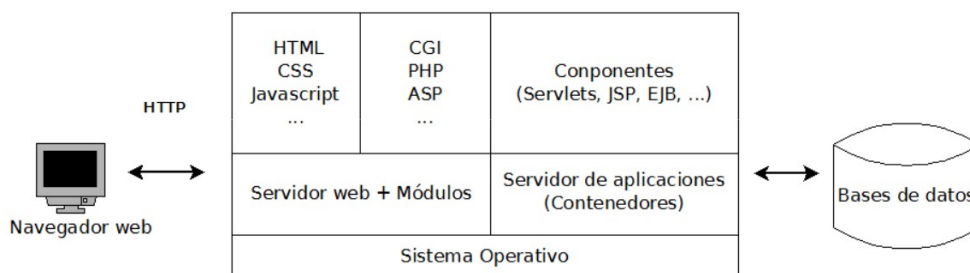
### 1.2.3 Servidor web de contenido dinámico con acceso a base de datos

A la estructura anterior podemos añadir una base de datos que consultaremos desde el código de los scripts de servidor.



### 1.2.4 Contenedores de servlet/JSP

Los **servlet** son objetos **Java** con una definición muy concreta. Esos **servlets** añaden funcionalidad a un servidor web. Un contenedor de **servlet** es un programa capaz de redirigir las peticiones **HTTP** hacia los **servlet**. Ejemplos de contenedores son el **Apache Tomcat** y el **JBoss**.



**Nota:** Lo que en estas imágenes aparece como servidor puede montarse en un único equipo o puede hacer uso de varios ordenadores. Con esto aumenta la complejidad pero puede mejorar el rendimiento y la seguridad. Por ejemplo, si el gestor de la base de datos lo instalamos en un equipo separado no tendrá que competir por los mismos recursos que el servidor web y además podría salir del **DMZ**.