

Unidad 1- Introducción al desarrollo páginas web

ÍNDICE

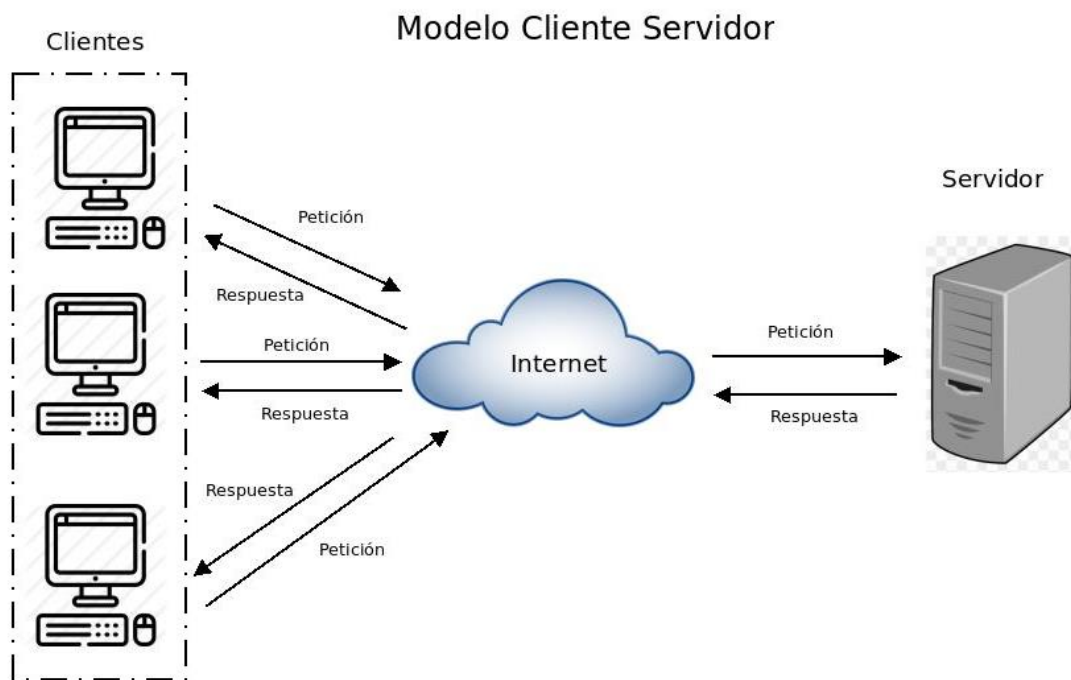
1. Arquitecturas web.....	1
1.1 Arquitectura a 3 capas	2
1.1.1 Tier.....	2
1.1.2 Layer	4
1.2. Arquitectura SPA (Single Page Application)	5
1.3 MVC.....	6
2. Generación dinámica de páginas web	7
2.1 Integración con los lenguajes de marca	8
3. Puesta en marcha.....	9
3.1 XAMPP.....	9
3.2 Visual Studio Code.....	9

1. Arquitecturas web

Las arquitecturas web definen la forma en que las páginas de un sitio web están estructuradas y enlazadas entre sí.

Las aplicaciones web se basan en un modelo cliente-servidor.

Es un modelo en el que hay 2 tipos de elementos: cliente y servidor.



En la imagen podemos ver como uno o varios clientes acceden a un servidor.

Las nuevas arquitecturas sustituyen el servidor por un balanceador de carga de manera que N servidores, dan respuesta a M clientes.

La comunicación entre clientes y servidores web se realiza mediante protocolo HTTP (HTTPS), que tiene reservado los puertos 80 y 553 respectivamente.

1.1 Arquitectura a 3 capas

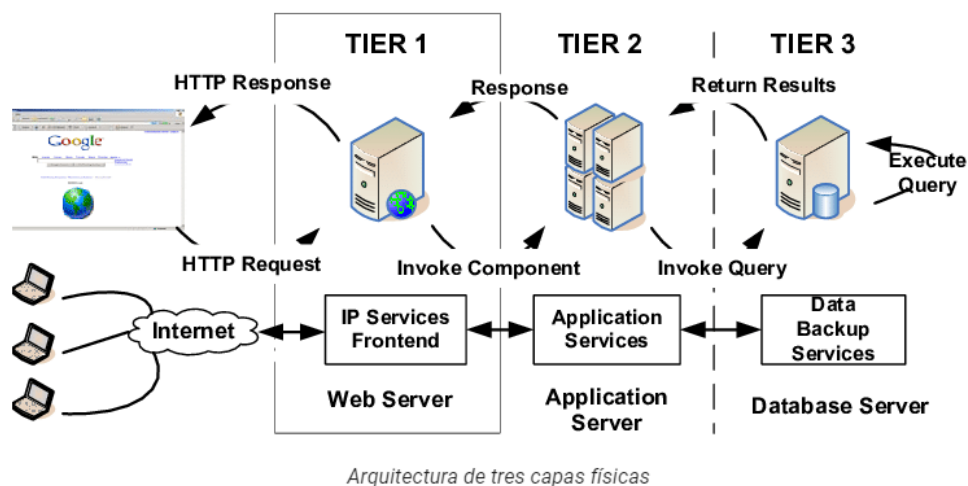
La arquitectura a 3 capas es una ampliación del modelo cliente-servidor. Hay que distinguir entre capas **físicas** (*tier*) y capas **lógicas** (*layer*).

1.1.1 Tier

Es la capa física de una arquitectura. Supone un nuevo elemento hardware separado físicamente. Las capas físicas más alejadas del cliente están más protegidas, tanto por firewalls como por VPN.

Ejemplo de arquitectura en tres capas físicas (3 tier):

- Servidor Web
- Servidor de Aplicaciones
- Servidor de base de datos



Nota: No confundir las capas con la cantidad de servidores. Actualmente se trabaja con arquitecturas con múltiples servidores en una misma capa física mediante un *cluster*, para ofrecer tolerancia a errores y escalabilidad horizontal.

Veamos cada uno:

Servidor web

Un servidor web es un software que recibe peticiones HTTP (GET, POST, DELETE, ...) del lado del servidor. Devuelve el recurso solicitado (HTML, CSS, JS, JSON, imágenes, etc...)

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

- Software libre y multiplataforma
- Sistema de módulos dinámicos (PHP, Python, Perl)
- Utiliza el archivo .htaccess para su configuración

En la actualidad, *Apache* está perdiendo mercado respecto a Nginx (<https://www.nginx.com>). Se trata de un producto más moderno (2004) y que en determinados escenarios tiene mejor rendimiento que Apache.

- Comparativa servidores
web: https://w3techs.com/technologies/history_overview/web_server/ms/q

Servidor de Aplicaciones

Un servidor de aplicaciones es un software que ofrece servicios adicionales a los de un servidor web, como puede ser:

- *Clustering*
- Balanceo de carga
- Tolerancia a fallos

Un ejemplo de servidor de aplicaciones es *Tomcat* (<http://tomcat.apache.org/>) que es un servidor de aplicaciones open source y multiplataforma de referencia para una arquitectura Java.

Servidor de base de datos

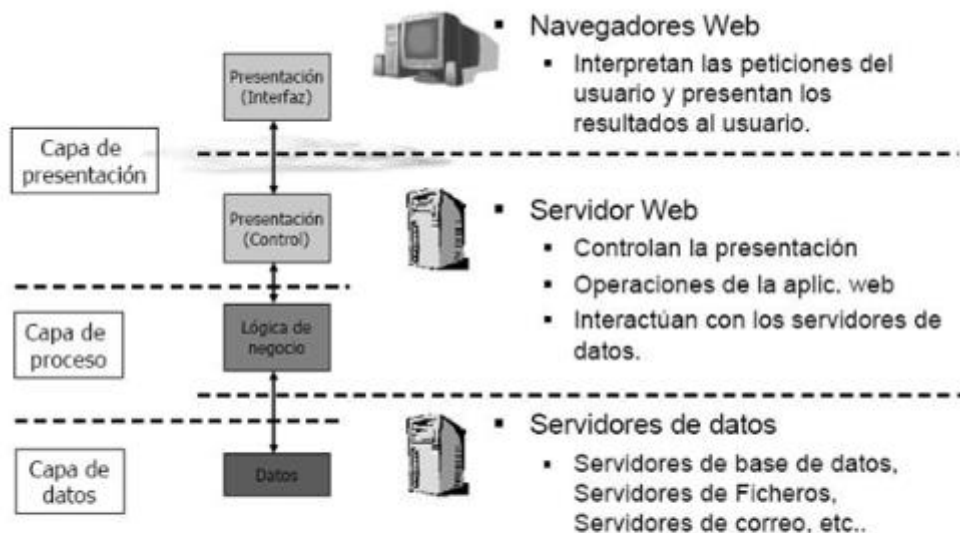
Un servidor de base de datos (*database server* o RDBMS), es un tipo de software de servidor que permiten la organización de la información mediante el uso de tablas, índices y registros.

Un ejemplo de servidor de base de datos es *MySQL server*

1.1.2 Layer

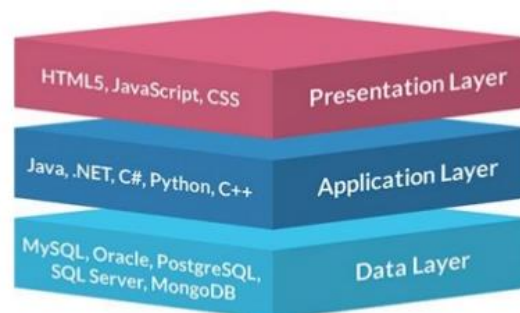
En cambio, las capas lógicas (*layers*) organizan el código respecto a su funcionalidad:

- Capa de presentación. Para la interfaz del usuario. Muestra información y permite interactuar con el sistema al usuario.
- Capa de negocio. Para la lógica propia de la aplicación. Se comunica con las otras dos capas.
- Capa de datos. Para gestionar la base de datos.



Arquitectura de tres capas físicas en tres lógicas

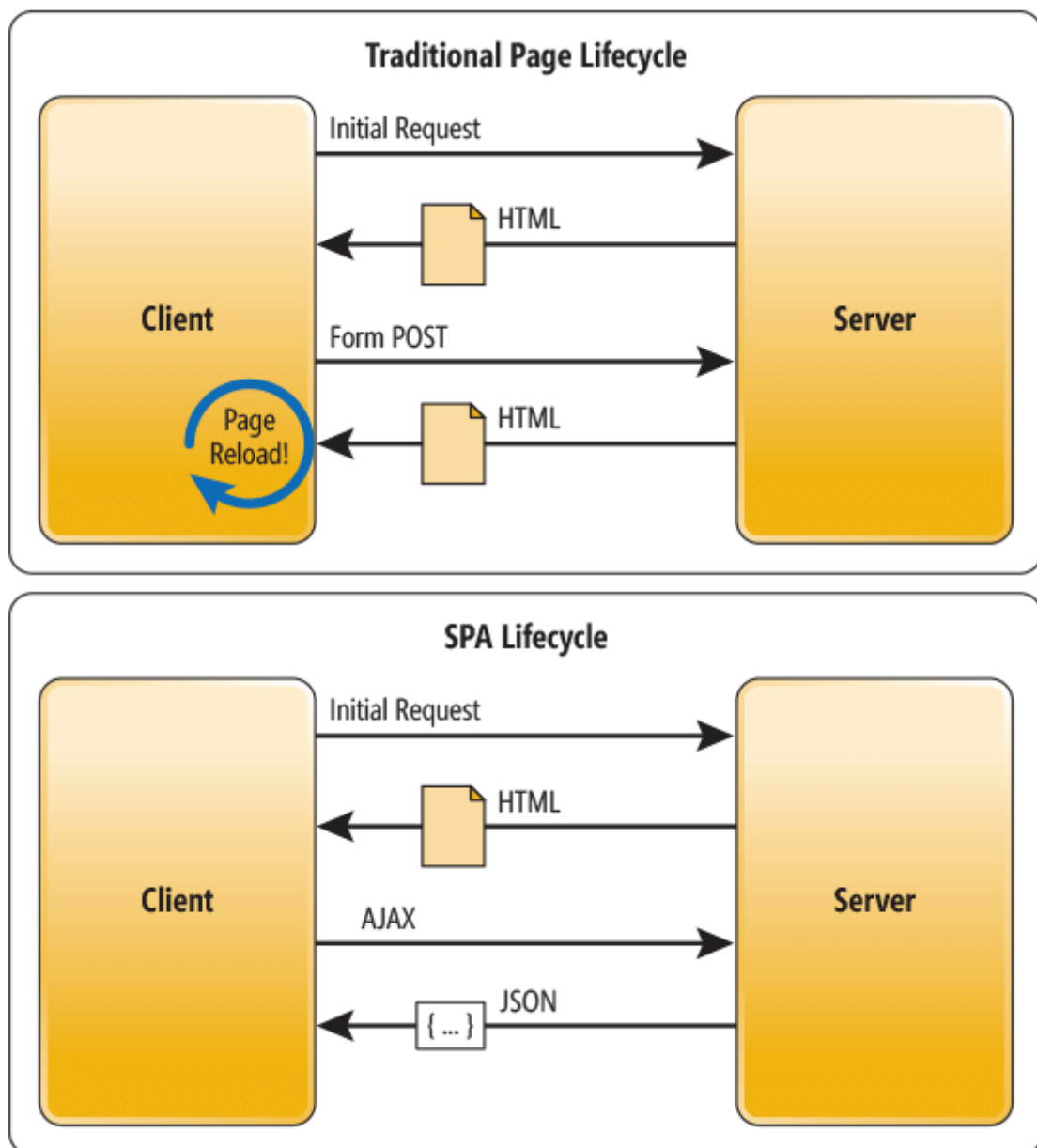
En la siguiente imagen podemos ver como cada una de las capas se puede implementar con diferentes lenguajes de programación y/o herramientas.



1.2. Arquitectura SPA (Single Page Application)

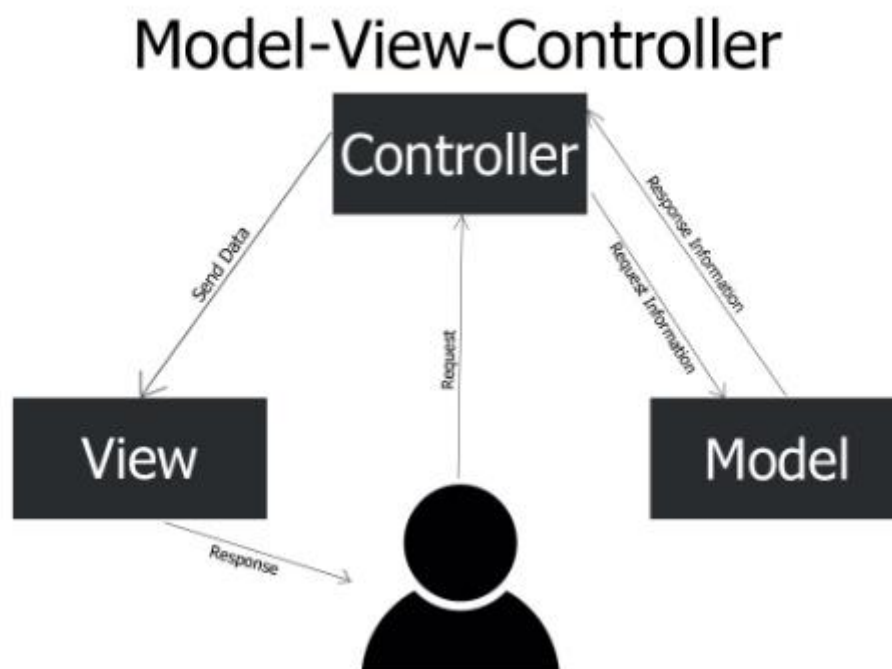
A día de hoy, gran parte del desarrollo web está transicionando de una arquitectura web cliente-servidor clásica donde el cliente realiza una llamada al *backend*, por una **arquitectura SPA** donde el cliente gana mucho mayor peso y sigue una programación reactiva que accede a servicios remotos REST que realizan las operaciones (comunicándose mediante JSON).

Veámoslo:



1.3 MVC

Model-View-Controller o Modelo-Vista-Controlador es un modelo de arquitectura que separa los datos y la lógica de negocio respecto a la interfaz de usuario y el componente encargado de gestionar los eventos y las comunicaciones. Al separar los componentes, permite reutilizar el código y mejorar su organización y mantenimiento.



Sus elementos son:

- **Modelo:** representa la información y gestiona todos los accesos a ésta, tanto consultas como actualizaciones provenientes, normalmente, de una base de datos. Se accede vía el controlador.
- **Controlador:** Responde a las acciones del usuario, y realiza peticiones al modelo para solicitar información. Tras recibir la respuesta del modelo, le envía los datos a la vista.
- **Vista:** Presenta al usuario de forma visual el modelo y los datos preparados por el controlador. El usuario interactúa con la vista y realiza nuevas peticiones al controlador.

Lo estudiaremos en más detalle al profundizar en el uso de los *frameworks* PHP.

2. Generación dinámica de páginas web

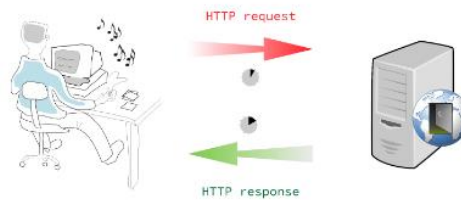
El lenguaje básico para la web es el HTML + CSS. Una página que esté escrita usando solo este lenguaje será **estática**, es decir, mostrará siempre el mismo contenido.

Una **página web dinámica** es aquella que no ofrece el mismo contenido a todos los usuarios, sino que parte del contenido se genera dinámicamente.

Por ejemplo, muchas páginas muestran anuncios diferentes según dónde esté el usuario.

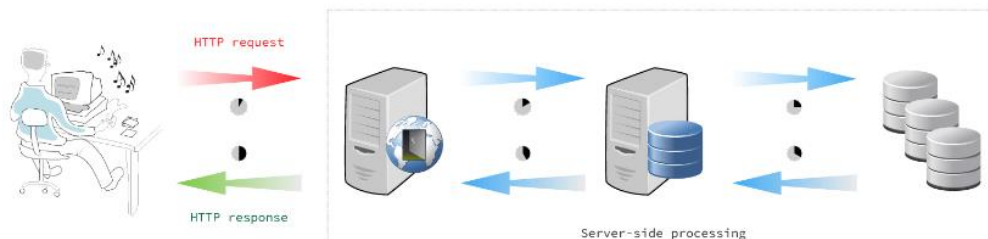
Static Website

Scheme A



Dynamic Website

Scheme B



Para generar una página dinámica, donde el contenido cambia, a día de hoy tenemos dos alternativas:

- Utilizar un lenguaje de servidor que genere el contenido, ya sea mediante el acceso a una BD o servicios externos.
- Utilizar servicios REST de terceros invocados desde JS.

Las tecnologías empleadas (y los perfiles de desarrollo asociados) para la generación de páginas dinámicas son:

Perfil	Herramienta	Tecnología / Lenguajes de programación
Front-end / cliente	Navegador Web	HTML + CSS + JavaScript
Back-end / servidor	Servidor Web + BBDD	PHP, Python, Ruby, Java / JSP, .Net / .asp

Nota: Cuando en las ofertas de trabajo hacen referencia a un Full-stack developer, están buscando un perfil que domina tanto el front-end como el back-end

2.1 Integración con los lenguajes de marca

Las páginas dinámicas se componen de una parte estática HTML y una parte dinámica en algún lenguaje de programación. Por eso se habla de lenguaje de plantillas.

Alguno de los lenguajes que se utilizan son:

- PHP
- JavaEE: Servlets / JSP
- Python
- ASP.NET → Visual Basic .NET / C#
- Ruby

Por ejemplo:

Cuando se solicita una página, el servidor busca en ella bloques de código. Si los encuentra, los ejecuta y los sustituye por su salida.

En PHP los bloques se delimitan por `<?php" y" ?>`

En la siguiente imagen hay un bloque PHP en las líneas 7-9:

```
1  <!DOCTYPE html>
2  <html>
3  |   <head>
4  | |   <title>Numero aleatorio</title>
5  | |   </head>
6  | |   <body>
7  | |   <?php
8  | | |   echo rand(0, 100);
9  | |   ?>
10 |   </body>
11 </html>
```

Al solicitar la página, el servidor ejecuta el bloque y lo sustituye por su salida, en este caso un número aleatorio entre 0 y 100. Si sale un 71, enviará al cliente:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Número aleatorio</title>
  </head>
  <body>
    71 </body>
</html>
```

3. Puesta en marcha

Para poder trabajar con un entorno de desarrollo local, hemos de preparar nuestro entorno de desarrollo con las herramientas comentadas. A lo largo del curso vamos a utilizar la versión 8.1 de PHP.

3.1 XAMPP

XAMPP (<https://www.apachefriends.org/es/index.html>) es una distribución compuesta con el software necesario para desarrollar en entorno servidor. Se compone de las siguientes herramientas en base a sus siglas:

- X para el sistema operativo (de ahí que se conozca también como LAMP o WAMP).
- A para Apache.
- M para MySQL / MariaDB. También incluye phpMyAdmin para la administración de la base de datos desde un interfaz web.
- P para PHP.
- la última P para Perl.

3.2 Visual Studio Code

Vamos a emplear *Visual Studio Code* (<https://code.visualstudio.com>) como entorno de desarrollo (IDE). Aunque existen otras alternativas.