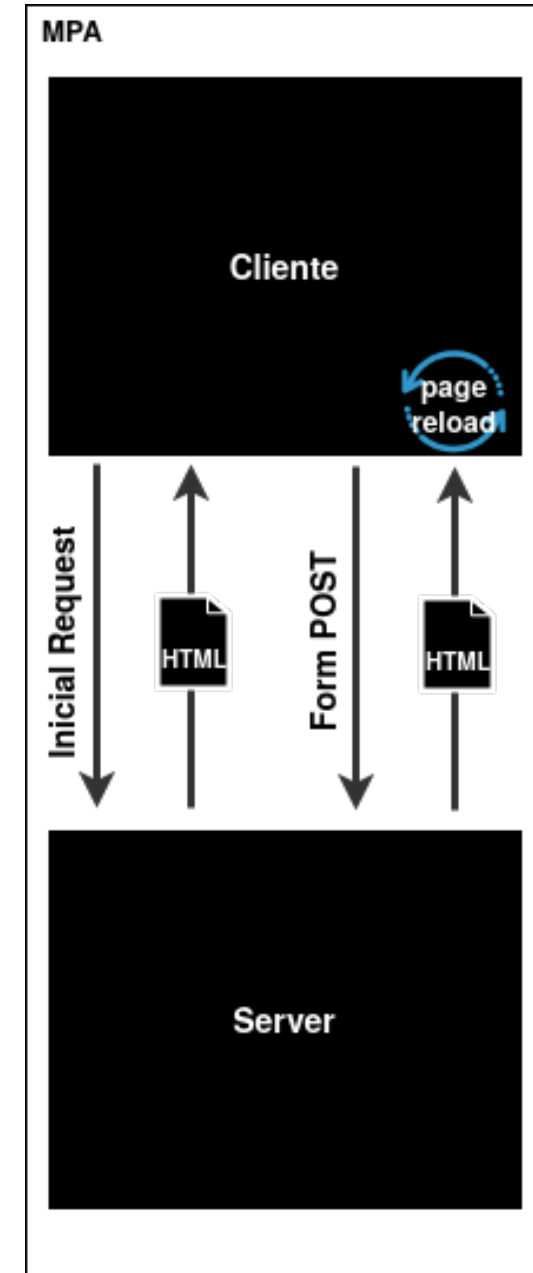


Aplicación multipágina

son las aplicaciones web tradicionales que vuelven a cargar toda la página y muestran la nueva cuando un usuario interactúa con la aplicación web.

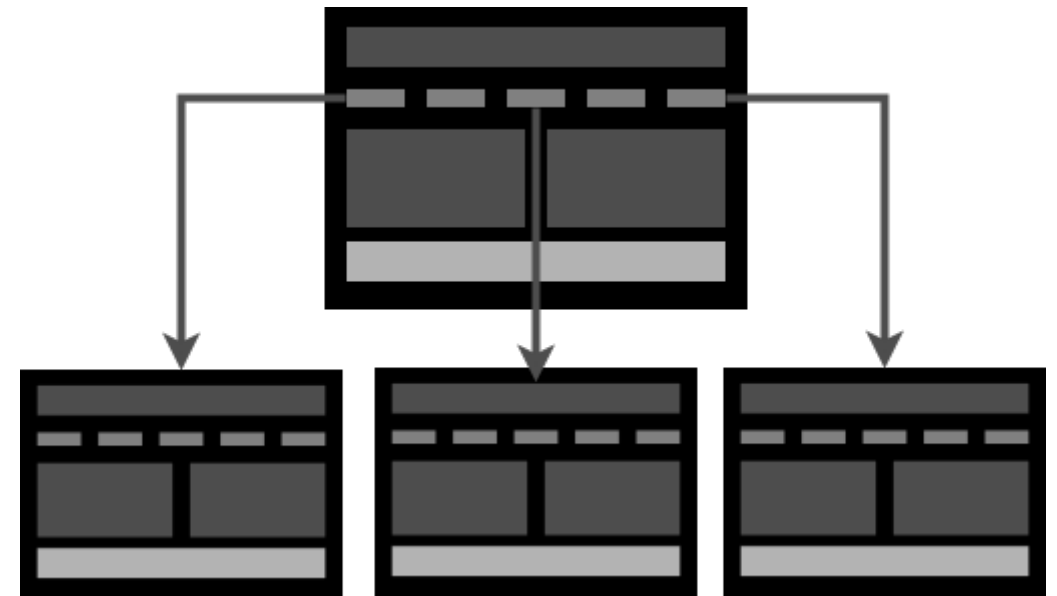
Cada vez que se intercambian datos de un lado a otro, se solicita una nueva página del servidor para que se muestre en el navegador web. Este proceso lleva tiempo para generar las páginas en el servidor, enviarlo a un cliente y mostrarlo en el navegador que puede afectar la experiencia del usuario.

Sin embargo, AJAX había hecho posible renderizar solo un componente particular de la aplicación, pero esto hace que el proceso de desarrollo sea aún más difícil y complejo. Las aplicaciones de varias páginas tienen una arquitectura más clásica. Cada página envía una solicitud al servidor y actualiza completamente todos los datos. Incluso si estos datos son pequeños. Por lo tanto, el rendimiento se gasta en mostrar los mismos elementos.



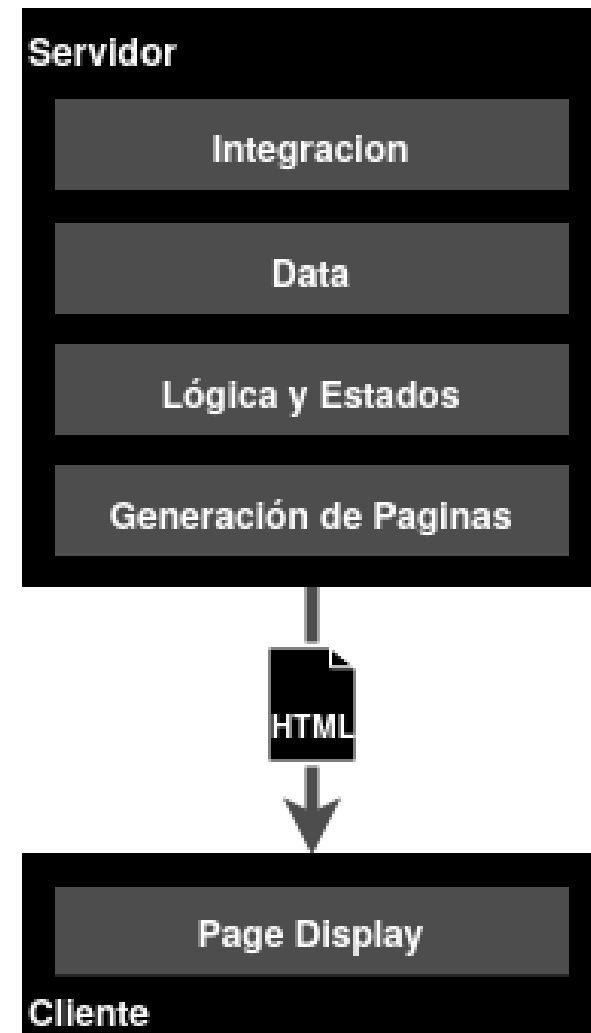
Ventajas

- **SIMPLE OPTIMIZACIÓN SEO:** La arquitectura de **MPA** le permite optimizar fácilmente cada página para los motores de búsqueda. El desarrollador puede agregar Meta etiquetas para cualquier página.
- **FÁCIL DESARROLLO:** Por lo general, el desarrollo de una aplicación de varias páginas requiere una pila más pequeña de tecnologías.
- **VARIAS SOLUCIONES:** Usando **MPA** puede encontrar una **boxed solution** adecuada. Por ejemplo, use Magento, OpenCart para desarrollar una aplicación web de comercio electrónico o Dolphin, Elgg para el desarrollo de redes sociales



Desventajas

- **TOMA MUCHO MÁS TIEMPO DESARROLLAR APLICACIONES MÓVILES:** En la mayoría de los casos, necesitará codificar el **back-end** desde cero.
- **ES DIFÍCIL SEPARAR EL Back-end Y EL Front-end:** El trabajo de los desarrolladores front-end y back-end se vuelve más complicado. Ya que es mas complicado separarlos.



Frameworks para el desarrollo

- **LENGUAJES PARA DESARROLLO WEB**

existe una gran cantidad de lenguajes de programación enfocados al desarrollo web. En sus orígenes, estos lenguajes tenían un formato estático. Sin embargo, con el paso de los años y la evolución a la hora de crear páginas web han evolucionado en lenguajes dinámicos.

Nuevos avances que se enfocan en permiten al usuario interaccionar más con la página y mejorar la experiencia en la navegación.

Existen muchos lenguajes de lenguaje de programación web con diferentes enfoque de programación para este tipo de desarrollos. Los as importantes son



Frameworks para el desarrollo

- **QUE ES UN FRAMEWORK WEB**

Los frameworks web son un conjunto de herramientas, estilos y librerías dispuestas a través de una estructura o esqueleto base, para el desarrollo de aplicaciones web más escalables y sencillas de mantener.

Gracias a estos frameworks web, podemos ahorrar grandes cantidades de tiempo y costes en desarrollo y mantenimiento.

- **RAZONES PARA UTILIZAR**

- 1) Evitar escribir código repetitivo
- 2) Utilizar buenas prácticas
- 3) Permitir hacer Desarrollos avanzadas de manera facil.
- 4) Desarrollar más rápido

- **QUE FRAMEWORKS USAR**

Debemos tener claro cuáles son las características y necesidades de nuestro proyecto para elegir que tecnología utilizar

factores

- 1) Documentación y comunidad.
- 2) Reutilización del código
- 3) Arquitectura y metodología.
- 4) Plantillas web
- 5) Seguridad web
- 6) Posicionamiento en motores de búsqueda
- 7) Ofertas de empleo

Frameworks para el desarrollo

- **JAVA FRAMEWORKS WEB**

El lenguaje java es uno de los lenguajes de mas usados en proyectos web, y posee un gran cantidad de frameworks para facilitar el desarrollo.

Spring MVC:

framework de desarrollo de aplicaciones ligero para Java EE que en realidad se puede utilizar para todo tipo de aplicaciones. Dispone de varios sub-frameworks como: Spring Core, que dota de funcionalidad a todo lo demás; SpringMVC, para la parte que UI que se devuelve al navegador; Spring ORM, para el acceso a datos; Spring Security, para autenticación y autorización de las aplicaciones; etc...

Hibernate

Es un **ORM** que nos permite crear la capa de acceso a datos de nuestra aplicación utilizando objetos en lugar de consultas, disminuyendo la fricción entre el mundo de las bases de datos relacionales y la Programación Orientada a Objetos.

Struts

Mantenido por la **Fundación Apache**, Está basado en el patrón **MVC**, dispone de una arquitectura basada en plugins que permiten extenderlo fácilmente con cualquier funcionalidad que requieras y posee muchos plugins de serie (procesar JSON, crear servicios REST...).

Frameworks para el desarrollo

- **JAVA FRAMEWORKS WEB**

JavaServer Faces (JSF):

Es en realidad una especificación "estándar" para la creación de interfaces de usuario para aplicaciones Web, aunque se puede usar también para crear aplicaciones nativas. La creó Oracle y su primera versión data del año 2004, utiliza el patrón MVC, y está basado en componentes.

Grails:

Es un framework de desarrollo Web que utiliza el **lenguaje Groovy** para la plataforma Java. Groovy es un lenguaje orientado a objetos que compila a Bytecode y que está enfocado a fondo en la productividad del desarrollador. Por ese motivo, Grails lo que te da es una productividad altísima y te permite estar produciendo código para tu aplicación de manera casi inmediata

Dropwizard

Es un framework Java pensado para hacer las cosas de forma muy sencilla, sin demasiadas sofisticaciones, para que los desarrolladores puedan implementar todo de forma rápida. Se puede crear aplicaciones web RESTful con un alto rendimiento, estables y confiables.

Google Web Toolkit (GWT)

Es un framework para desarrollar aplicaciones web de alto rendimiento en lenguaje Java, y destaca porque es capaz de convertir código Java en código JavaScript, y lo hace en paquetes amigables para el navegador, aportando compatibilidad entre navegadores.

Frameworks para el desarrollo

- **C# y .NET FRAMEWORKS**

C# es un lenguaje elegante, con seguridad de tipos y orientado a objetos que permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas que se ejecutan en el ecosistema de .NET. El ecosistema compuesto por todas las implementaciones de .NET, incluidas, entre otras, .NET Core y .NET Framework.

ASP.NET MVC Framework

- Es un framework de aplicaciones web que implementa el patrón modelo-vista-controlador (MVC). Basado en ASP.NET, permite a los desarrolladores de software construir una aplicación web.

ASP.NET Razor

En una sintaxis basada en C# que permite usarse como motor de programación en las vistas o plantillas de nuestros controladores. Es una de las novedades de ASP.NET MVC 3.

Frameworks para el desarrollo

- **PHP**

Es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, PHP se utiliza para generar páginas web dinámicas.

Laravel

acceso a funciones como la autenticación de usuarios, la administración de sesiones y el almacenamiento en caché. En general, Laravel reúne toda la funcionalidad que necesitarás para construir una aplicación PHP moderna. Laravel también se integra a la perfección con otras bibliotecas y plataformas de terceros, como Amazon Web Services (AWS), que te permite crear aplicaciones altamente escalables.

CodeIgniter

CodeIgniter es un framework PHP que usa una arquitectura de Model View Controller (MVC). En términos sencillos, eso significa que CodeIgniter utiliza diferentes componentes para manejar tareas de desarrollo específicas.

Symfony

- Bastante flexible. Cuenta con un paquete y un sistema de componentes que te permite elegir las funciones de PHP que necesitas, o simplemente usar todo el framework. Para garantizar que tus aplicaciones se ejecuten correctamente, el framework también incluye la funcionalidad de prueba incorporada

Zend

framework basado en MVC, orientado a objetos, que permite cargar solo los componentes que quieres como bibliotecas individuales.

Frameworks para el desarrollo

Phalcon

- Su código fuente está escrito en C, por lo que es básicamente una extensión C de PHP. Por estar escrito en C es uno de los frameworks más rápidos, también es muy ligero en cuanto a recursos, y utiliza una arquitectura MVC.

CakePHP

primer framework MVC de PHP en salir al mercado. Aprovecha un amplio conjunto de componentes. Usa las convenciones de CakePHP para programar proyectos más rápido.

Frameworks para el desarrollo

- **PYTHON**

Es un lenguaje de programación de código abierto, orientado a objetos, muy simple y fácil de entender. Tiene una sintaxis sencilla que cuenta con una vasta biblioteca de herramientas.

Django

Es un framework de aplicaciones web gratuito y de código abierto escrito en Python. Manejar la autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para tu sitio web, formularios, una forma de subir archivos, etc. Django redefine este modelo como MVT: Modelo-Vista-Template.

Flask

proporciona herramientas y funciones útiles que hacen que crear aplicaciones web en Python. Ofrece a los desarrolladores flexibilidad y un marco más accesible, es extensible y no fuerza una estructura de directorio concreta .

- **M significa "Model" (Modelo)**, la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
- **T significa "Template" (Plantilla)**, la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web o otro tipo de documento.
- **V significa "View" (Vista)**, la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelos y las plantillas.

MVC Modelo Vista Controlador

- **Patrón de Arquitectura**

es una solución general y reutilizable a un problema común en la arquitectura de software dentro de un contexto dado. Los patrones arquitectónicos son similares al patrón de diseño de software pero tienen un alcance más amplio.

Patrones Simples

Layers, Pipes-y-Filtros, Pizarrón, Repositorio

Sistemas Distribuidos

Broker (Microkernel, Pipes-y-Filtros), CAGS, Cliente-Servidor

Sistemas Interactivos

Modelo-View-Controlador, Presentación-
Abstracción-Control

Patrones Adaptables

MicroKernel, Reflexion

Patrón de diseño

Los patrones de diseño son unas técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Patrones creacionales

Object Pool, Abstract Factory, Builder, Factory Method.

Patrones estructurales

Adapter o Wrapper , Bridge, Composite, Decorator , Facade, Flyweight, Proxy, Module.

Patrones de comportamiento

Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor.

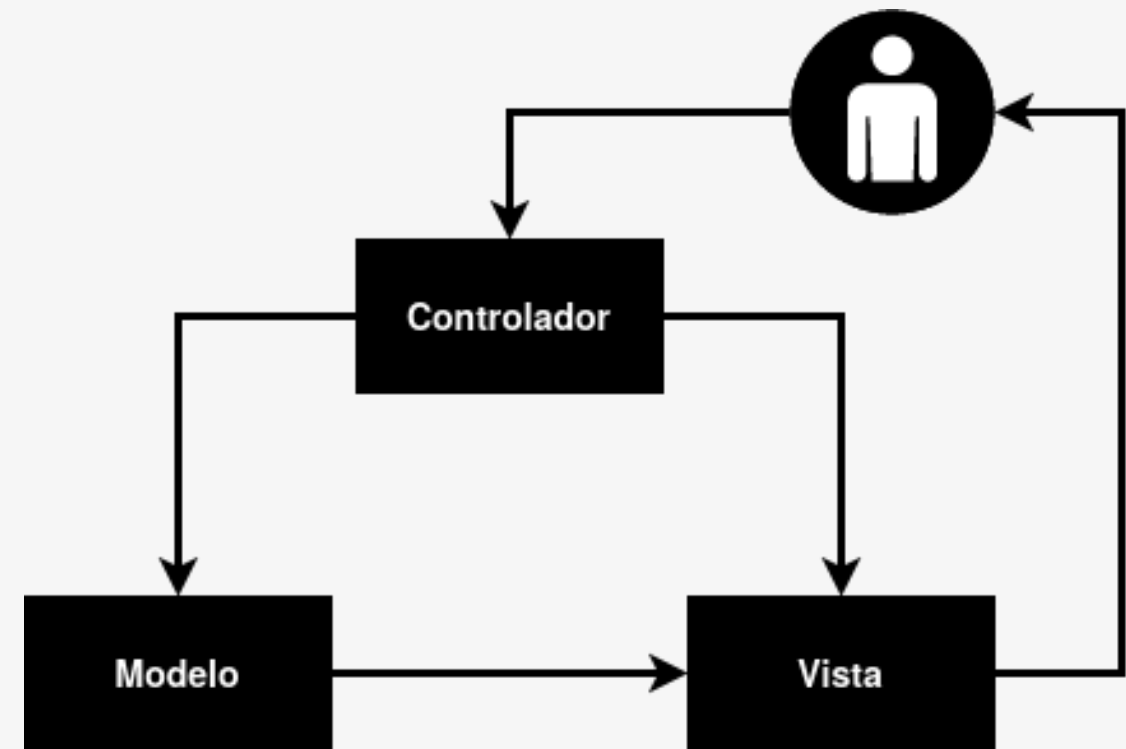
MVC Modelo Vista Controlador

QUE ES MVC

Es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos.

Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman **Modelos, Vistas y Controladores**, o lo que es lo mismo, **Model, Views & Controllers**,



MVC Modelo Vista Controlador

Ventajas

- 1) Separación clara de dónde tiene que ir cada tipo de lógica, facilitando el mantenimiento y la escalabilidad de nuestra aplicación.
- 2) Sencillez para crear distintas representaciones de los mismos datos.
- 3) Facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar desarrollo guiado por pruebas (Test Driven Development o TDD).
- 4) Reutilización de los componentes.
- 5) No existe ciclo de vida de las páginas. Con menos peso, menos complejidad.
- 6) Motor de Routing asociando una URL concreta con su correspondiente controlador, permitiendo URL semánticas. Las URL semánticas se indexan mejor en los buscadores.
- 7) Recomendable para el diseño de aplicaciones web compatibles con grandes equipos de desarrolladores y diseñadores web que necesitan gran control sobre el comportamiento de la aplicación.

MVC Modelo Vista Controlador

Definiciones

- **Modelo:** este componente se encarga de manipular, gestionar y actualizar los datos. Si se utiliza una base de datos aquí es donde se realizan las consultas, búsquedas, filtros y actualizaciones.
- **Vista:** este componente se encarga de mostrarle al usuario final las pantallas, ventanas, páginas y formularios como resultado de una solicitud enviada a través del controlador. Desde la perspectiva del programador este componente es el que se encarga del frontend; la programación de la interfaz de usuario si se trata de un aplicación de escritorio, o bien, la visualización de las páginas web (CSS, HTML, HTML5 y Javascript).
- **Controlador:** este el componente principal de la aplicación, donde se especifican los métodos y funcionalidades que una aplicación (o módulo de una aplicación) tienen que realizar. Se encarga de gestionar las instrucciones que se reciben, atenderlas y procesarlas. A través del controlador se realizan las consultas al modelo (una búsqueda por ejemplo), y una vez se hayan obtenido dichos datos, se envía a la vista las instrucciones necesarias para poder mostrarlos de una forma legible para el usuario

MVC Modelo Vista Controlador

