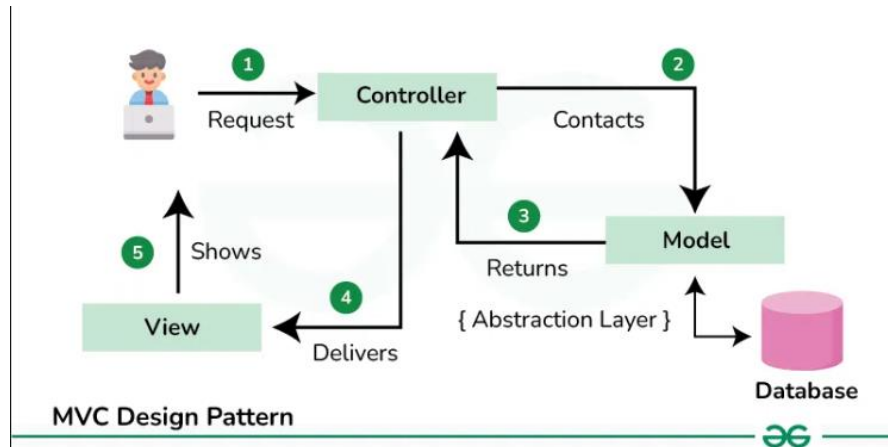


## ¿QUÉ ES EL MODELO VISTA CONTROLADOR?

El modelo-vista-controlador (MVC) es un patrón de diseño arquitectónico que separa una aplicación en tres componentes principales: el modelo, la vista y el controlador. En el contexto de Spring Framework, MVC se utiliza para desarrollar aplicaciones web escalables y mantenibles.

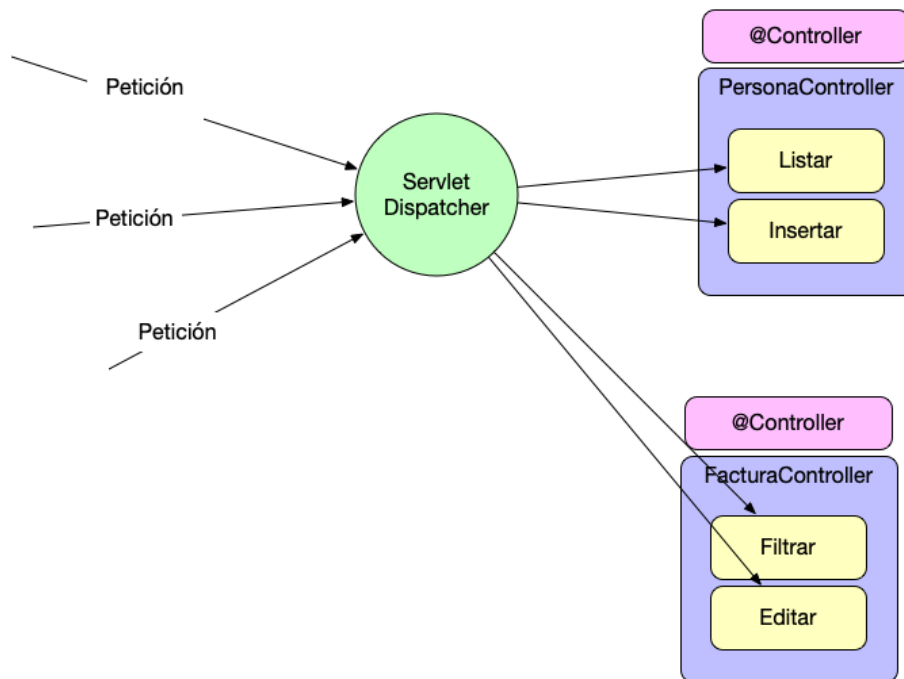


**Modelo:** El modelo representa los datos y la lógica de negocio de la aplicación. En una aplicación Spring MVC, el modelo puede ser una entidad de dominio, un objeto POJO (Plain Old Java Object) o una clase de servicio que interactúa con la capa de persistencia de datos. El modelo encapsula el estado de la aplicación y define las operaciones que se pueden realizar sobre esos datos.

**Vista:** La vista es la interfaz de usuario que presenta los datos al usuario final. En Spring MVC, la vista suele ser una plantilla (template) que se renderiza para producir la interfaz de usuario final. Las plantillas pueden ser HTML con fragmentos de código Java o XML, o pueden utilizar un motor de plantillas como Thymeleaf o FreeMarker. La vista se encarga de mostrar los datos del modelo al usuario y de capturar la entrada del usuario para enviarla al controlador.

**Controlador:** El controlador actúa como intermediario entre el modelo y la vista. En Spring MVC, los controladores son clases que manejan las solicitudes HTTP, procesan los datos del modelo y deciden qué vista mostrar al usuario. Los controladores pueden utilizar anotaciones de Spring, como **@Controller** y **@RequestMapping**, para mapear las solicitudes HTTP a métodos específicos y devolver el modelo y la vista apropiados. Los controladores también pueden realizar validaciones de entrada, ejecutar la lógica de negocio y manejar excepciones.

En Spring Framework, la implementación de MVC se basa en el patrón de diseño Front Controller, donde `DispatcherServlet` actúa como el controlador frontal que maneja todas las solicitudes HTTP entrantes y las enruta al controlador apropiado. Spring MVC proporciona una serie de características adicionales, como la inyección de dependencias, la gestión de transacciones, la seguridad integrada y la internacionalización, que facilitan el desarrollo de aplicaciones web robustas y escalables.



## VENTAJAS DE USAR MVC EN SPRING

1. **Separación de preocupaciones:** MVC separa la lógica de presentación de la lógica de negocio y los datos, lo que facilita la comprensión y el mantenimiento del código.
2. **Reutilización de código:** Al separar los componentes de la aplicación en modelos, vistas y controladores, es más fácil reutilizar y modificar cada componente de forma independiente.
3. **Escalabilidad:** MVC facilita la escalabilidad de la aplicación al permitir agregar nuevos controladores y vistas según sea necesario sin afectar el resto de la aplicación.
4. **Facilidad de prueba:** La separación clara de los componentes en MVC facilita la escritura de pruebas unitarias y de integración para cada componente de la aplicación.