Spring MVC

Spring MVC

- *framework* MVC para crear aplicaciones usando el patrón Modelo-Vista-Controlador.
- Permite desarrollar aplicaciones para la web con arquitectura Modelo-Vista-Controlador.

Qué es el patrón MVC

- **un patrón arquitectural**, un modelo o guía que expresa cómo organizar y estructurar los componentes de un sistema software, sus responsabilidades y las relaciones existentes entre cada uno de ellos.
- define la forma en que se organizan los componentes de presentación en sistemas distribuidos.

MVC

- La arquitectura MVC propone, independientemente de las tecnologías o entornos en los que se base el sistema a desarrollar, la **separación de los componentes de una aplicación en tres grupos** (o capas) principales: el modelo, la vista, y el controlador, y describe cómo se relacionarán entre ellos para mantener una estructura organizada, limpia y con un acoplamiento mínimo entre las distintas capas.

Modelo

- **Representación de los datos del dominio**, es decir, aquellas entidades que nos servirán para almacenar información del sistema que estamos desarrollando.

- Si nuestra aplicación forma parte de un sistema distribuido, es decir, consume servicios prestados por otros sistemas, en el Modelo encontraremos las **clases de transferencia de datos** (DTO, *Data Transfer Objects*) que nos permitirán intercambiar información con ellos.

Modelo

- Asimismo, encontraremos **la lógica de negocio de la aplicación**, es decir, la implementación de las reglas, acciones y restricciones que nos permiten gestionar las entidades del dominio. Será por tanto el responsable de que el sistema se encuentre siempre en un estado consistente e íntegro.

- Por último, el Modelo será también el encargado de gestionar el **almacenamiento y recuperación de datos y entidades** del dominio, o lo que es lo mismo, incluirá mecanismos de persistencia o será capaz de interactuar con ellos.

Modelo

Resumiendo:
el Modelo contiene principalmente las entidades que representan el dominio, la lógica de negocio, y los mecanismos de persistencia de nuestro sistema.

Vista

- Los componentes de la Vista son los responsables de generar **la interfaz de nuestra aplicación**, o sea, de componer las pantallas, páginas, o cualquier tipo de resultado utilizable por el usuario o cliente del sistema.
- Suele decirse que la Vista es una representación del estado del Modelo en un momento concreto y en el contexto de una acción determinada.

En la Vista encontraremos los componentes responsables de generar la interfaz con el exterior, por regla general, aunque no exclusivamente, el UI de nuestra aplicación.

Controlador

- Actúa como **intermediarios entre el usuario y el sistema.**
- Captura las acciones del usuario sobre la Vista.
- También se encarga de la transformación de datos para hacer que los componentes de la Vista y el Modelo se entiendan. Traducen la información enviada desde la interfaz, por ejemplo, los valores de campos de un formulario recibidos mediante el protocolo HTTP(S), a objetos que puedan ser comprendidos por el Modelo.
- Y tomará la información procedente del Modelo y la adaptará a formatos o estructuras de datos que la Vista sea capaz de manejar.

Controlador

- **El controlador es el coordinador general del sistema**, que regula la navegación y el flujo de información con el usuario, ejerciendo también como intermediario entre la capa de Vista y el Modelo.

En el Controlador se encuentran los componentes capaces de procesar las interacciones del usuario, consultar o actualizar el Modelo, y seleccionar las Vistas apropiadas en cada momento.

Relación entre Modelo, Vista y Controlador