

Escuela Colombiana de Ingenieria Julio Garavito
Ciclos de vida del desarrollo de software
Solución lab 5 MVC
2024

- ¿Por qué MVC obtiene ese nombre? (puede apoyarse de <https://www.javatpoint.com/spring-mvc-tutorial>)

Recibe su nombre pese a sus siglas:

M (Model): Hace referencia a la capa logica de datos del negocio de la app, manejando el acceso y manipulacion de datos para procesar la informacion
Ejm: base de datos, archivos , servicios web.

V (vista): Responsable de la presentación visual de la información , encargada de mostrar la interfaz y presentar los datos al usuario final de una manera comprensible y atractiva.

C (controller): intermediario entre el Modelo y la Vista, y maneja las solicitudes del usuario. El Controlador interpreta las acciones del usuario y coordina las interacciones entre el Modelo y la Vista

- ¿Cuáles son las ventajas de usar MVC?

- MVC separa el código en tres partes: Modelo, Vista y Controlador.
- Facilita la reutilización y mantenimiento del código.
- Permite realizar pruebas de manera más sencilla y efectiva.
- Ayuda a escalar la aplicación a medida que crece en complejidad.
- Mejora la estructura, la organización y la mantenibilidad de las aplicaciones
- Promueve una estructura organizada y modular del proyecto.
- Facilita el proceso de desarrollo y promueve prácticas de programación sólidas y eficientes.

- ¿Qué diferencia tiene la estructura de directorios de este proyecto comparado con las de proyectos pasados (con solo maven y java EE)?

Las diferencias son el resultado de la adición de nuevas funcionalidades, la reorganización del código existente, la eliminación de código obsoleto o la refactorización del código para mejorar su calidad y mantenibilidad.

- La inclusión de un framework web específico, como lo es Spring MVC
- Directorio src/main/resources/static donde se colocan los recursos estáticos de la aplicación web
- Ausencia de directorios específicos de Java EE
- Configuración adicional de Spring como application.properties

- ¿Qué anotaciones usaste y cuál es la diferencia entre ellas?

Anotaciones usadas en el Springframe de java

@controller se utiliza para marcar una clase como un controlador en Spring MVC. Los controladores manejan las solicitudes HTTP y devuelven las respuestas adecuadas.

@GetMapping("/greeting") : mapea solicitudes HTTP GET a métodos específicos dentro de un controlador.

@SpringBootApplication : Esta anotación se utiliza para marcar la clase principal de una aplicación Spring Boot. Esta anotación combina tres anotaciones: @Configuration, @EnableAutoConfiguration, y @ComponentScan. Indica que la clase es una clase de configuración, habilita la configuración automática de Spring Boot basada en el contenido del classpath y escanea los paquetes para componentes de Spring.

@ResponseBody indicar que el valor devuelto por un método de controlador debe ser vinculado directamente al cuerpo de la respuesta HTTP, en lugar de ser vinculado a una vista. Es útil cuando se quiere devolver datos en formato JSON o XML directamente desde un controlador.

@RequestMapping("/User/{id}") Esta anotación se utiliza para mapear solicitudes HTTP a métodos de controlador específicos y permite el uso de variables de ruta, como {id} en este caso, que serán pasadas como parámetros al método del controlador.