Samuel Eduardo Amaya Ruìz

14/11/2023

Código: 202220313

Taller 5 – Patrones

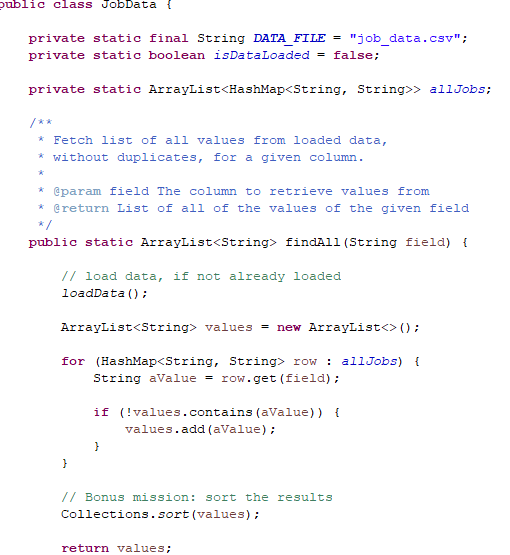
1. Información general del proyecto y link del repositorio

<https://github.com/LaunchCodeEducation/techjobs-mvc>

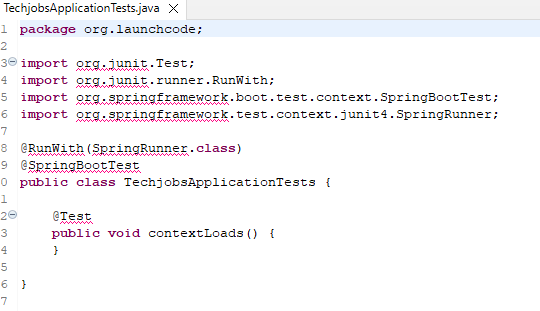
El proyecto adjunto se seleccionó con el propósito de que se pueda analizar y ver la manera en la que se implementa el patrón en dicho proyecto. Lo que le permite al usuario hacer este proyecto es realizar búsquedas sobre documentos de Excel (csv). En el proyecto se implementan y se hace uso de varias clases. En la clase JobData es donde se implementa toda la lógica del proyecto que en este caso hace referencia al modelo del proyecto. Además de esto hay otras clases que hacen referencia a los controladores del proyecto y finalmente en el proyecto se importa un springframework el cual se hace cargo de todo lo que tiene que ver con la vista del proyecto. Esa sería la manera en la que se encuentra estructurado el diseño del proyecto. Finalmente, uno de los grandes retos que enfrenta el proyecto o que lo hace difícil es que hay partes en las que las conexiones están incompletas y hace falta implementación de código en el proyecto.

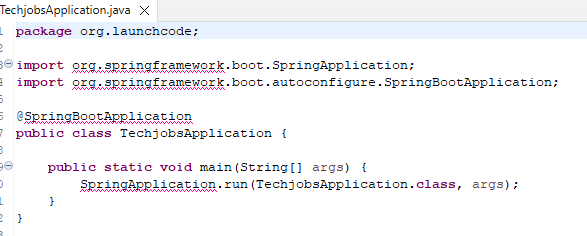
1. A continuación, se mostrará información y estructura de algunos fragmentos del proyecto en donde aparece el patrón y la manera en la que se implementa dentro del proyecto.

Primero que todo en la clase JobData es donde se implementa toda la lógica y por lo tanto tiene la función del modelo dentro del patrón MVC .



En este fragmento se puede ver la inicialización de la lógica JobData del proyecto y por lo tanto el que tendría el papel del modelo dentro del patrón MVC. En esta parte de la implementación del código lo que se esta haciendo es analizar todo tipo de valor de la información que se haya cargado para ser revisada haciendo uso de loadData(). Además de esto también se hace la carga del archivo en caso de que no haya sido cargado.





Dentro del proyecto también se hace uso del framework de springframework.boot, con el uso de este framework lo que se logra es tomar información rápidamente de una base de datos. En este fragmento del proyecto lo que se esta haciendo es que se esta llamando la función de SpringApplication.run en donde se comienza a correr la aplicación dentro del framework mencionado anteriormente. Como se puede evidenciar en esta parte del proyecto lo que se esta implementando es la vista del patrón MVC en donde se hace uso de este framework el cual implementa lo necesario de manera autónoma haciendo la implementación del modelo y de los controladores.

Texto

Descripción generada automáticamente

Aquí se puede ver que dentro del proyecto se crea un paquete llamado launchcode.controllers en donde se encuentran guardados cada uno de los controladores que se manejan dentro del proyecto. Cada uno de estos archivos se conectan a la página web por medio de la vista. Dentro de estos archivos se hace uso de RequestMapping, el cual sirve para administrar las solicitudes y enviarlos al modelo para que dentro de la lógica de este se analicen y se retornen a la vista para que se le muestre al usuario como tal. El controlador ListController permite manejar las solicitudes que filtra según las necesidades que haya indicado el usuario. Por último, dentro del SearchController se procesan los datos que se manejan dentro del proyecto que el usuario ingresa para ser analizados. Estos tres archivos es en donde se ve reflejado el uso del controlador dentro del patrón MVC.

1. Información sobre el patrón MVC

El patrón de diseño Modelo-Vista-Controlador separa una aplicación en tres componentes principales que son el Modelo, la Vista y el Controlador. Cada uno de estos componentes tienen una responsabilidad única y específica en la aplicación y de la misma manera están diseñados para que funcionen de manera independiente entre sí. El objetivo principal de este patrón es proporcionar una estructura que facilite tanto el desarrollo, como la modificación y el mantenimiento a largo plazo del código.

Lo que se hace dentro del modelo es representar los datos y la lógica de la aplicación. También responde a las consultas sobre su esta y permite realizar modificaciones en caso de ser necesarias. Finalmente, dentro del modelo se notifica a la vista cualquier cambio en los datos.

La vista es la parte del patrón responsable de mostrar la interfaz y presentar la información al usuario. Además, recibe los datos que se trabajan en el modelo y la muestra de manera adecuada. Finalmente, otra de las funciones que cumple es que no realiza modificaciones directas en los datos, sino que solo presenta la información al usuario.

Finalmente, el controlador actúa como un intermediario entre el modelo y la vista, ya que recibe las interacciones del usuario a través de la vista y actualiza el modelo teniendo en cuenta los datos ingresados por parte del usuario. También, actualiza todo lo que sea necesario de la vista cuando el modelo se modifica.

Algunos de los usos más comunes que se le dan a este patrón son primero que todo el desarrollo web, ya que permite separar la lógica del servidor y la presentación en el navegador. Otro de los usos que se le da es para el desarrollo de aplicaciones de escritorio y de aplicaciones móviles.

1. ¿Por qué tiene sentido haber utilizado el patrón en este proyecto? ¿Qué ventajas tiene?

Para este proyecto haber hecho uso del patrón MVC permite que la implementación del código sea mucho más sencilla y además permite que sea mucho más fácil de entender para un tercero. Otra de las ventajas que tiene hacer uso de MVC es que facilita la reutilización del código ya que existe una separación de las responsabilidades. Además, facilita el mantenimiento ya que se pueden realizar cambios en una parte del sistema sin afectar otras partes. Una de las ventajas más importantes que tiene este patrón es que reduce el acoplamiento.

1. ¿Qué desventajas tiene haber utilizado el patrón en ese punto del proyecto?

Una de las desventajas que trae el patrón MVC es que puede ser complejo de implementar, especialmente para proyectos que son pequeños o simples, de la misma manera puede agregar una sobrecarga innecesaria. Además, otra de las desventajas que presenta es el exceso de acoplamiento entre la vista y el controlador y puede presentar ciertos problemas en aplicaciones en tiempo real.

1. ¿De qué otras formas se le ocurre que se podrían haber solucionado, en este caso particular, los problemas que resuelve el patrón?

Otra manera en la que se pudo haber solucionado el problema que resuelve el patrón es que simplemente toda la implementación del código se pudo hacer en un único archivo o lo que es lo mismo en una única clase.