

JUSTIFICACIÓN DE LA IMPLEMENTACIÓN DEL PATRÓN MVC

En este documento se justifican las decisiones tomadas durante la aplicación del patrón de diseño Modelo-Vista-Controlador (MVC) en el proyecto de la asignatura (RecSys).

Este patrón se puede identificar fácilmente en la estructura del proyecto, ya que se han empleado estos mismos nombres para los paquetes que contienen la lógica de programación de cada una de las partes de este patrón, tal y como se puede ver en la Figura 1.

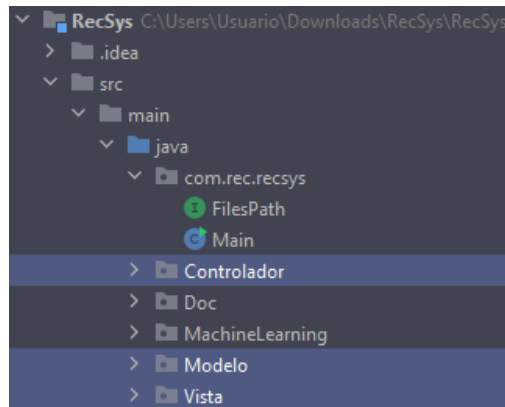


Figura 1. Estructura del proyecto RecSys

A continuación, se desglosan las decisiones más relevantes en la implementación de este proyecto:

- En primer lugar, en el paquete “com.rec.recsys” se encuentra la clase “Main”, que es la encargada de lanzar la aplicación. Esta clase incluye la creación de todos los componentes del patrón de diseño (modelo, vista y controlador). Dado que se han creado dos vistas: una vista inicial y una vista principal que contiene la lógica de la aplicación, en esta clase se define la “Vista” correspondiente al patrón MVC como “MainScene”, pero se crea y lanza la vista inicial (IniScene).

```
//Clase Main
public class Main extends Application {

    @Override
    public void start(Stage primaryStage) {

        ImplementacionControlador controlador = new ImplementacionControlador();
        ImplementacionModelo modelo = new ImplementacionModelo();
        MainScene vista = new MainScene();

        modelo.setVista(vista);
        controlador.setVista(vista);
        controlador.setModelo(modelo);
        vista.setModelo(modelo);
        vista.setControlador(controlador);

        primaryStage.setScene(IniScene.createIniScene(primaryStage, vista));
        primaryStage.show();
    }
}
```

Figura 2. Clase ejecutable principal

En este paquete también se puede encontrar la interfaz “FilePath” que contiene todas las rutas a los ficheros utilizados en el proyecto, de forma que se puedan modificar fácilmente.

- En segundo lugar, en los paquetes Modelo, Vista y Controlador se incluyen las clases correspondientes a dichos componentes del patrón de diseño MVC separados en interfaces (definición de métodos) e implementación.

De estos paquetes cabe señalar la decisión tomada en la clase “Implementación del Modelo”, donde se usa el sistema de recomendación creado en prácticas anteriores (RecSys). La decisión más relevante en esta clase se encuentra en el método “generateRecommendation” el cual genera una lista de recomendaciones utilizando los parámetros recibidos desde el controlador, y a su vez definidos por el usuario en la vista. En este caso se ha tomado la decisión, por cuestiones de eficiencia, de no entrenar el modelo otra vez si el usuario no modifica los parámetros de búsqueda. Es decir, si el usuario selecciona el mismo algoritmo (KNN o KMeans) y la misma distancia (Euclidean o Manhattan) para búsquedas consecutivas no se reentrenará el modelo. Esto se consigue guardando la última configuración solicitada (currentConfig) y comparándola con la configuración de la nueva búsqueda solicitada por el usuario (newConfig).

- Por último, en el paquete “MachineLearning” se pueden encontrar todas las clases necesarias para el funcionamiento de la aplicación.

Para finalizar, es importante tener en cuenta que el esfuerzo de implementar el patrón MVC en este proyecto es demasiado grande para la simplicidad del proyecto. Es decir, no sería necesario. Sin embargo, el haber aplicado este patrón le da gran flexibilidad y escalabilidad al proyecto. Es decir, si en un futuro se quisieran añadir nuevas funcionalidades más complejas, no sería necesario modificar la estructura del mismo.