

Práctica 4: Interfaces gráficas de usuario

Introducción

A lo largo de estas sesiones de prácticas has desarrollado una aplicación que se ejecuta en una consola. En este último bloque de prácticas vas a crear una interfaz gráfica de usuario para interaccionar con tu aplicación.

Vas a utilizar la biblioteca **JavaFX** para desarrollar la interfaz gráfica.

Además de que te familiarices con el desarrollo con **JavaFX**, lo más importante de este último bloque es que implementes la nueva funcionalidad siguiendo el (meta)patrón de diseño Modelo-Vista-Controlador (MVC), tal y como lo hemos visto en clase de teoría.

Sesiones

Esta práctica la vas a desarrollar en tres sesiones.

Objetivos

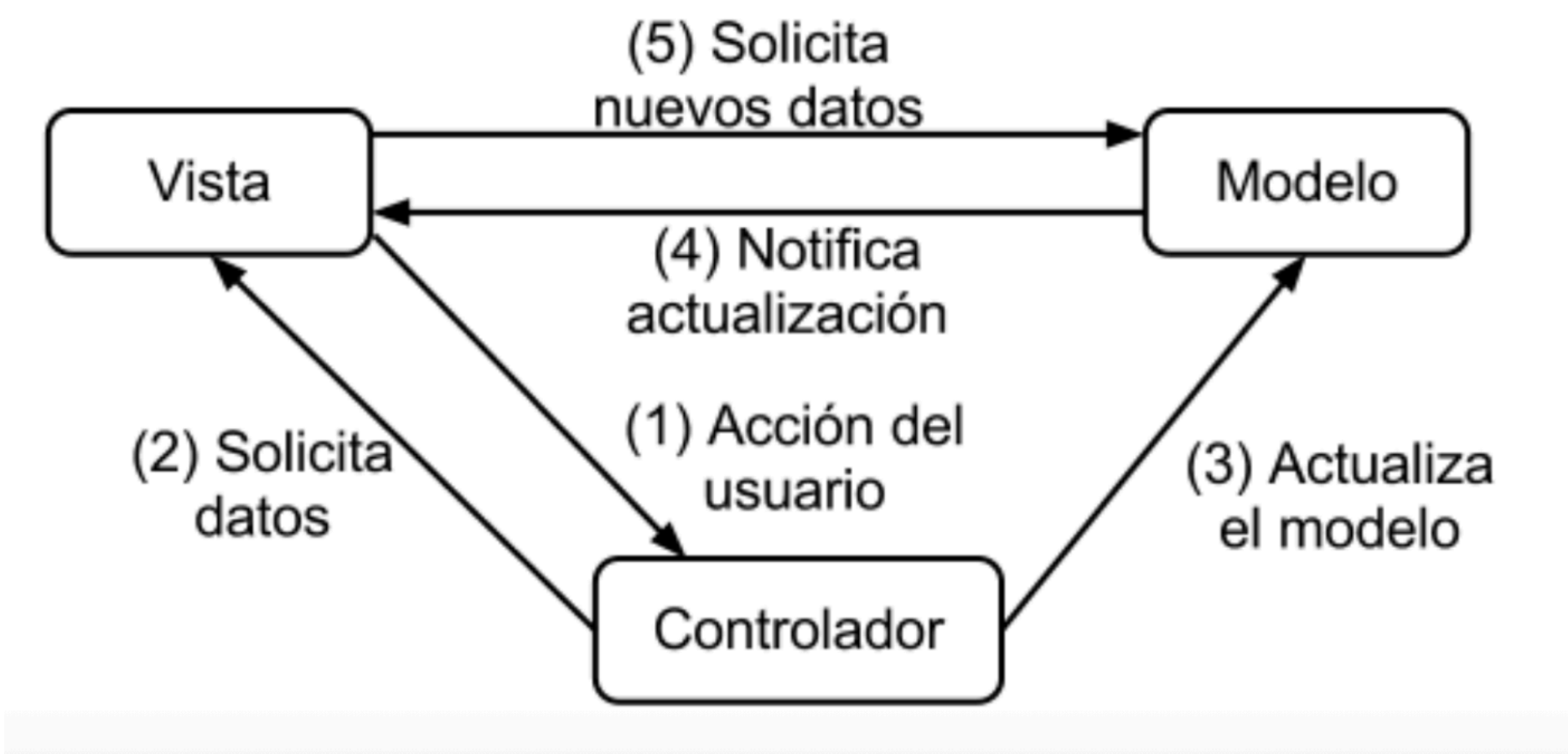
Los objetivos de esta práctica son:

1. Dotar de una interfaz gráfica a tu aplicación, basada en **JavaFX**.
2. Utilizar el patrón de diseño MVC en la implementación.

Metodología

Modelo-Vista-Controlador (MVC) es un (meta)patrón de diseño, que separa los datos (Modelo), la lógica (Controlador) y el interfaz de usuario (Vista). El Modelo corresponde a los datos con los que trabaja la aplicación. Esto puede representar a los datos que se transfieren entre Visa y Controlador, o cualquier otro dato relacionado con la lógica de la aplicación. La Vista se utiliza para toda la lógica de la interfaz de usuario. Incluye todos los componentes de la interfaz de usuario, tales como cuadros de texto, checkboxes, menús, etc., y la captura eventos del usuario en la interfaz, p.e. hacer clic en un botón, seleccionar un elemento, etc. Finalmente, el Controlador actúa como intermediario entre el Modelo y la Vista, para procesar toda la lógica de la aplicación. Manipula los datos utilizando el Modelo, e interactúa con la Vista para obtener de la interfaz la información necesaria .

Observa la figura siguiente para comprender el flujo de control entre los diferentes componentes del patrón MVC. Fíjate que no hay una flecha que vaya desde el Modelo al Controlador: efectivamente, el Modelo solo mantiene los datos de la aplicación, e informa a la Vista cuando hay cambios en los datos; no inicia la lógica de la aplicación (lo cual es tarea del controlador).



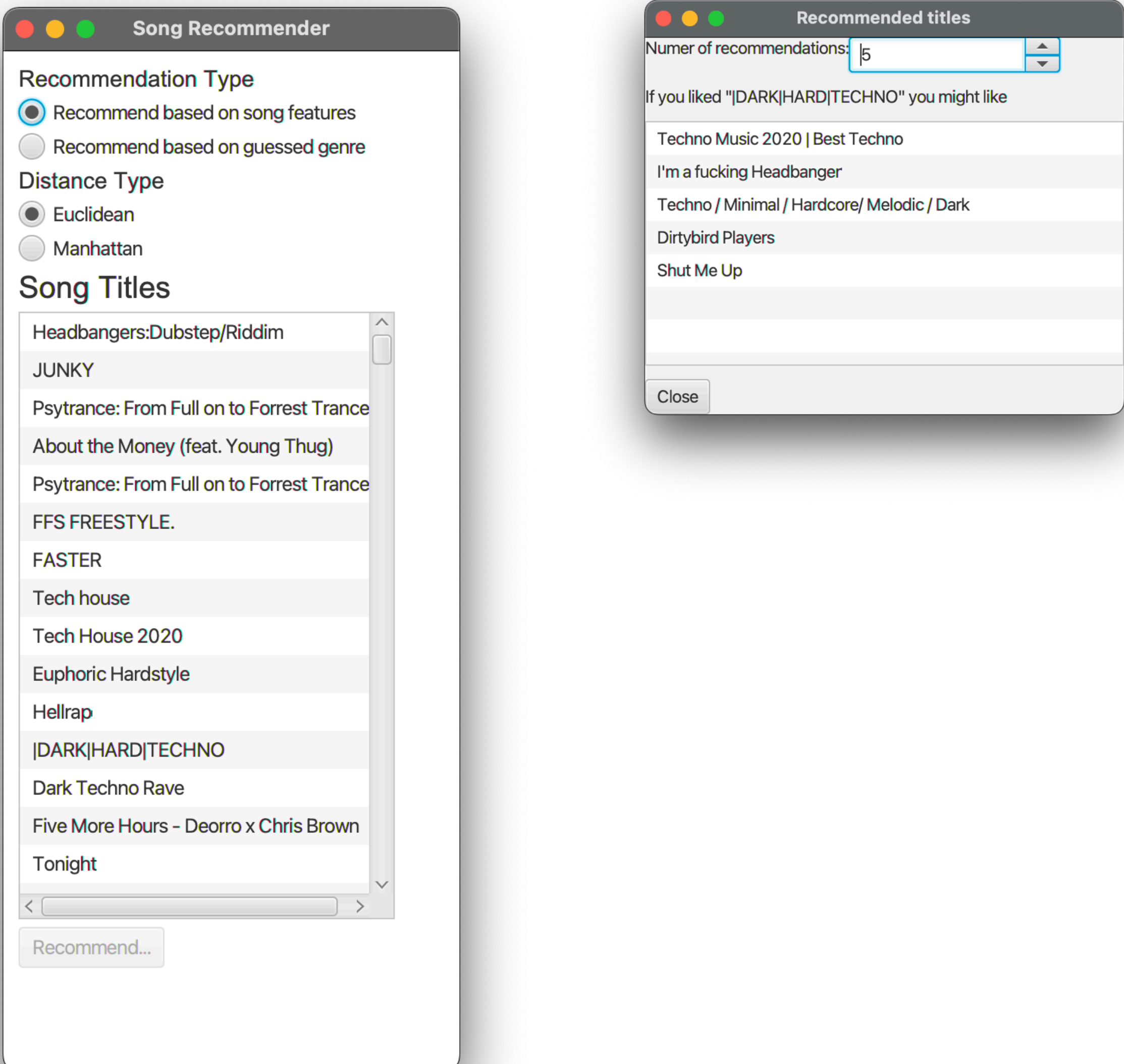
Es fácil «despistarse» implementando una aplicación con interfaz gráfica de usuario y el patrón MVC. Una guía que te puede ayudar, aunque por sí sola no es ninguna garantía, es que crees tres nuevos paquetes en tu proyecto, uno por cada uno de los tres roles que define el patrón MVC. De este modo, cada vez que crees una nueva clase, tendrás que decidir dentro de qué paquete debes incluir la nueva clase. Como hemos visto en las sesiones de teoría, en MVC cada clase debe tomar un único rol de los tres que hay disponibles.

Como aplicación, implementaremos una interfaz gráfica para el sistema de recomendación que hemos venido desarrollando en las prácticas anteriores. Utilizaremos los mismos datos sobre canciones que hemos usado en las prácticas P2 y P3.

La interfaz gráfica de usuario a desarrollar debe permitir, al menos, las siguientes funcionalidades:

- Seleccionar el tipo de recomendación (kMeans o KNN) y la distancia (euclidiana o Manhattan)
- Seleccionar, de una lista de todas las canciones, la canción en la que se basará la recomendación
- Visualizar las recomendaciones a partir de una canción de la lista, y permitir cambiar dinámicamente el número de recomendaciones
- Implementar, en la medida de lo posible, acciones para evitar errores del usuario (por ejemplo, deshabilitar el botón "Recommend" cuando ninguna canción ha sido seleccionada).

Como una imagen vale más que mil palabras, aquí tienes un ejemplo de una interfaz de usuario (básica) que implementa las funcionalidades solicitadas.



Esto es un simple ejemplo para demostrar la funcionalidad. No se pretende que tu proyecto sea exactamente igual que el ejemplo mostrado, al contrario, la intención es que crees una interfaz de usuario fácil de usar, pero de acuerdo a tus preferencias. Junto a la interfaz de usuario, evidentemente, también hay que prestar atención al rendimiento, para que la aplicación funcione sin problemas y sin grandes tiempos de espera.

En cuanto a la implementación, debes incorporar el patrón MVC a tu implementación de la práctica P3. Has de adaptar las clases (e interfaces) existentes considerando el patrón MVC (por supuesto, usando paquetes como siempre). A continuación, puedes reutilizar todas las clases/funcionalidades que ya implementaste. En particular, usarás "RecSys" para realizar todas las funcionalidades relacionadas con las recomendaciones, y tus nuevas clases del patron MVC reemplazarán a SongRecSys (que era una interfaz de usuario de estilo de línea de comandos).

Pruebas

Realizar pruebas para validar el funcionamiento correcto de una aplicación con interfaz gráfica de usuario es, por si sola, una tarea costosa, con unas técnicas específicas que no tienen cabida en los objetivos del curso. Por lo tanto, realiza pruebas manuales para comprobar que tu aplicación funciona correctamente.