

Proyecto Integrador

Integrantes:

- Battistella Emiliano Ciro Jesús
- Cajales Benjamín
- Fernandez Ismael
- Flores Cesar
- Peñaranda Juan Pablo

Introducción:

Nuestro proyecto es un sistema de reproducción de audio al que decidimos llamar CUYOfy, sirve para reproducir varias canciones de distintos artistas famosos.

Proyecto:

<https://drive.google.com/file/d/1frNOcJIR15QNHbUq5Sst5s7kh2T6wZJd/view?usp=sharing>

Aclaración: El proyecto está pensado para funcionar solo en Windows, porque utiliza ubicaciones de archivos. El proyecto accede al nombre de usuario de la persona que ejecuta el proyecto, lo que sirve para evitar modificar el código dependiendo de la computadora. Además, es necesario descargar la siguiente carpeta que incluye el archivo en formato WAV de las canciones y una carpeta para que se cree el archivo <.db> al inicializar el programa. La carpeta debe colocarse en C:\Users\"nombre_de_usuario\" \Music.

https://drive.google.com/drive/folders/11zdJ8TJ72JRNvA_uKIMRIW79TKkCmAfv?usp=drive_link

Explicación de clases (no paneles):

-ProyectoIntegrador1:

ProyectoIntegrador1 es la clase main del sistema. Inicia el panel de registro, maneja la inicialización de PanelPlaylist, crea arrayList de las canciones según el artista, sube las canciones, y playlist a la base de datos y guarda todas las canciones en una arrayList global.

-Reproductor:

Reproductor es una clase abstracta que sirve para reproducir, detener, pausar y reanudar canciones. Además, se encarga de controlar el volumen y de dar información sobre el estado actual de la canción que se está reproduciendo, su nombre, su duración, su posición actual y su estado de pausa. Esta clase es responsable de que se pueda controlar la reproducción de las canciones de la Playlist, utilizando a la clase Canción para su funcionamiento.

-Canción:

Canción se agrega a la clase Lista. Sus atributos son "nombre", "artista", "duración" y "path". El path se define automáticamente en la creación de una nueva canción y es clave para se pueda reproducir. Esta clase tiene su propia tabla en la base de datos.

-Lista:

Lista es una clase abstracta que como su nombre lo indica es una lista de canciones. Tiene métodos para agregar y quitar canciones, un método para mostrarse en la interfaz gráfica que utiliza HTML y un método que sirve para clasificar a las canciones según su género.

-MeGusta:

MeGusta hereda de Lista y compone a la clase Cuenta. Añade un boolean IsLiked que informa si el usuario le ha dado “me gusta” a esa canción. Los “me gusta” tienen su propia tabla en la base de datos.

-PlayList:

PlayList hereda de Lista, en su constructor se incluyen los atributos de “nombre”, “duración”, “descripción” y una lista de canciones. La clase tiene un @Override sobre el método para añadir canciones que le suma el tiempo de la canción a variable la “duración” dentro PlayList. Además de un método para calcular su duración. Tiene una interfaz gráfica en donde se puede editar, reproducir y gestionar playlist, y las playlist del usuario se suben a la base de datos.

-Álbum:

Álbum hereda de PlayList y se agrega a la clase Artista. Agrega el atributo “fecha” de tipo date. Al iniciar el programa se definen los álbumes de cada artista de forma automática.

-Cuenta:

Cuenta tiene un atributo “nombre” que se sube a la base de datos junto a una contraseña.

-Artista:

Artista hereda de Cuenta, tiene los atributos “cantidad de oyentes”, “género” y un arrayList de álbumes.

-DataBase:

DataBase es una clase que maneja toda la lógica de la base de datos. Incluye métodos para crear las tablas de “reproducciones”, “me gusta”, “canciones”, “usuarios” y “playlists”, así como un método para eliminarlas. También contiene métodos para agregar registros, como guardarUsuarioEnBaseDeDatos; eliminar registros, como eliminarMeGusta; y modificar registros, como actualizarNombrePlaylist y actualizarDescripcionPlaylist. Además, métodos que sirven para obtener información de la base de datos, como obtenerPlaylist para consultar playlists y sus canciones o “devolverUsuario”.

-Interface:

Interface es una clase de tipo Java Interface


-ImplInterface:

ImplInterface implementa el código de Interface, utilizado para que se pueda reproducir una canción utilizando solo su nombre en lugar de el objeto canción.

Desarrollo:

Al iniciar el proyecto se crean las tablas(si es que no existen), se establece la conexión con la base de datos, se crean los objetos de las clases “Canción”, “Álbum” y “Artista”, cargan las canciones dentro de la base de datos y dentro de un arrayList de canciones global y se inicializa el panel de inicio de sesión.

Dentro del panel de inicio de sesión el usuario puede ingresar sus datos que serán verificados por la base de datos, si son incorrectos aparecerá un mensaje de error en la interfaz y si son correctos será trasladado hacia el panel de InterfazMain. En el caso de no tener una cuenta puede crear una nueva haciendo clic en el botón de registrarse.



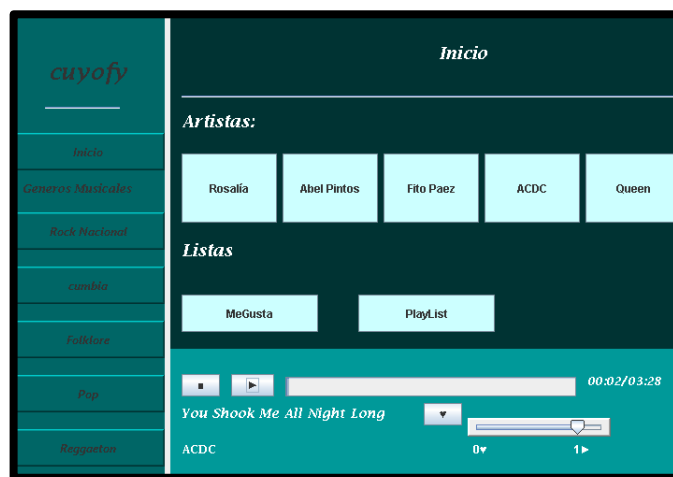
Dentro del panel de registro el usuario puede colocar un nombre y una contraseña, si el nombre es nulo aparece un mensaje de error en la interfaz, ocurre lo mismo si la longitud de la contraseña es menor a 5. Una vez el nombre y la contraseña son válidos se crea una nueva clase Cuenta y el usuario se guarda dentro de la tabla de usuarios (la primera columna es el nombre y la segunda la contraseña) el usuario se mueve nuevamente dentro del panel de inicio de sesión. Como añadido este panel al igual que el anterior tienen una “X” en la esquina para cerrar el programa sin necesidad de cerrarlo como una ventana tradicional.



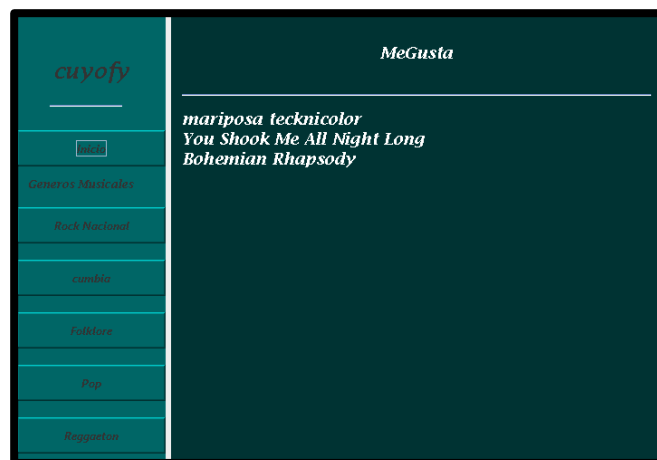
El panel de InterfazMain tiene un método que recibe un JPanel y lo muestra por pantalla. Además, tiene una barra lateral de botones que llaman a este método. El panel de inicio es el primer panel que es utilizado por este método.

Dentro del panel de inicio hay botones que seleccionan una canción aleatoria de un artista y la reproducen utilizando los métodos de la clase Reproductor, cuando se presiona uno de estos botones la reproducción es almacenada en una tabla de la base de datos, el texto de la parte inferior cambia para mostrar el nombre de la canción y el nombre del artista correspondiente e inician la barra de progreso, la etiqueta de tiempo en formato minutos/segundos y un contador de cuantos “me gusta” y cuantas reproducciones tiene la canción que se está reproduciendo

En la parte inferior de la pantalla también se incluyen botones para controlar el flujo de la reproducción, un botón para detener completamente la música y uno para pausarla. Además, una barra deslizable para controlar el volumen a gusto y un botón de “me gusta” que almacena la canción en la tabla de “me gusta” de la base de datos, si la canción ya está entre los “me gusta” del usuario el símbolo del corazón de la interfaz se reemplaza con una “X” que al presionarla quita la canción de la lista de “me gusta” del usuario y la borra de la base de datos.



Tanto la tabla de reproducciones como la tabla de “me gusta” guardan el nombre de la canción y el usuario que les dio “me gusta”. En el panel de “me gusta” aparecen las canciones que le gustan al usuario, que se guardan en la base de datos.



En el panel de InterfazMain, también se proporciona un botón “Playlist”, en donde te redirecciona a un Jpanel “PanelPlayList” que permite a los usuarios visualizar, editar, reproducir y gestionar playlists. Tiene varios métodos, el principal es cargarPlaylistsDefault(), que carga los nombres de todas las playlists desde la base de datos y las muestra en una lista cuando se inicia el panel. Además, muestra los detalles de una playlist específica, incluyendo su nombre, descripción y canciones cuando el usuario la selecciona. A esto último se le incluyen configurarNombrePlaylist() y configurarDescripcionPlaylist(), que permiten al usuario editar el nombre y la descripción de la playlist, guardando los cambios en la base de datos.

En este panel también se agrega la configuración de los botones de “reproducir” y “detener” que sirven para iniciar la reproducción de la playlist seleccionada y detenerla.

Para finalizar, configurarBusquedaCanciones() y configurarAgregarCancionBuscada(): Activa la búsqueda de canciones y los resultados de búsqueda a la playlist seleccionada. Este método utiliza un botón para mostrar los resultados de búsqueda en la interfaz. Adicionalmente, se encuentra configurarEliminarCancion(), que permite eliminar una canción de la playlist seleccionada.



En la barra lateral del panel de la interfaz principal están los géneros musicales, al hacerles clic aparecen las canciones según el género que se definen automáticamente con el método mostrar que se encuentra en la clase Lista.



Aprendizaje:

Para el desarrollo de este proyecto aplicamos lo aprendido sobre la programación orientada a objetos, relaciones entre clases, el uso de random, de arrayList y de excepciones. También investigamos sobre el uso de la clase Java Interface, sobre SQLite para el manejo de datos y sobre Java Swing para desarrollar la interfaz gráfica del sistema.

Conclusión:

El desarrollo del proyecto nos ayudó a aplicar los conocimientos vistos durante el cursado y los temas que investigamos por nuestra cuenta. A pesar de la limitación de compatibilidad exclusiva con Windows estamos satisfechos con el resultado del proyecto y creemos que pudimos demostrar nuestro conocimiento de Java y capacidad para relacionar y estructurar clases de manera eficiente y funcional.