

Facultad: Ingeniería en Sistemas

Universidad: Da Vinci

Curso: Estructura De Datos

Tema: Playlist

Nombre: Ezry Adilia Maribeth Leal Franco

Carnet: 202200202

“PLAYLIST”

INTRODUCCIÓN

Este trabajo describe la creación de una aplicación de reproducción de música en java, playlist que permitirá al usuario agregar o eliminar canciones, permitirá la reproducción de la música. Utilizando clases abstractas y concretas.

DESCRIPCION DEL PROBLEMAS Y SOLUCION PROPUESTA

1. PROBLEMA:

Consiste en diseñar un TDA playlist que permita la administración desde la aplicación para agregar, eliminar y reproducir canciones en lista, ordenar las canciones por duración o por artista y de forma aleatoria.

2. SOLUCION

La propuesta involucra el crear una serie de clases en java, que modelan canciones y playlist. Enlazando la estructura de datos para la lista, lo cual permite la creación, eliminación y edición de elementos de modo mas eficiente. Siendo la clase abstracta Playlist el que define las operaciones básicas, mientras que la clase concreta PlaylistConcreta implementa operaciones. Se implementa una interfaz con el usuario en la consola el cual permitirá una interacción con la playlist mediante comandos. Siendo la base para poder correr la consola el archivo Main.

CODIGO FUENTE

1. *CANCION.JAVA*

Clase abstracta para representar una canción

```
J Cancion.java > ...
1  import java.util.*;
2
3  abstract class Cancion {
4      private String titulo;
5      private String artista;
6      private int duracion;
7
8      public Cancion(String titulo, String artista, int duracion) {
9          this.titulo = titulo;
10         this.artista = artista;
11         this.duracion = duracion;
12     }
13
14     public String getTitulo() {
15         return titulo;
16     }
17
18     public String getArtista() {
19         return artista;
20     }
21
22     public int getDuracion() {
23         return duracion;
24     }
25 }
```

2. *CANCIONCONCRETA.JAVA*

Clase concreta para representar una canción específica

```
J CancionConcreta.java > ...
1  class CancionConcreta extends Cancion {
2      public CancionConcreta(String titulo, String artista, int duracion) {
3          super(titulo, artista, duracion);
4      }
5  }
```

3. *PLAYLIST.JAVA*

Clase abstracta para representar una playlist

```
J Playlist.java > ...
1  import java.util.*;
2
3  public abstract class Playlist {
4      protected List<Cancion> canciones;
5
6      public Playlist() {
7          this.canciones = new ArrayList<>();
8      }
9
10     public void agregarCancion(Cancion cancion) {
11         this.canciones.add(cancion);
12     }
13
14     public void eliminarCancion(Cancion cancion) {
15         this.canciones.remove(cancion);
16     }
17
18     public Cancion reproducirSiguiente() {
19         if (!this.canciones.isEmpty()) {
20             return this.canciones.remove(index:0);
21         } else {
22             return null;
23         }
24     }
25
26     public void ordenarPorDuracion() {
27         this.canciones.sort(Comparator.comparingInt(Cancion::getDuracion));
28     }
29
30     public void ordenarPorArtista() {
31         this.canciones.sort(Comparator.comparing(Cancion::getArtista));
32     }
```

```
33
34     public void reproducirAleatorio() {
35         Collections.shuffle(this.canciones);
36     }
37
38     public List<Cancion> getCanciones() {
39         return this.canciones;
40     }
41 }
```

4. *PLAYLISTCONCRETA.JAVA*

Clase que representa una playlist especifica

```
J PlaylistConcreta.java > ...  
1  public class PlaylistConcreta extends Playlist {  
2      public PlaylistConcreta() {  
3          super();  
4      }  
5  }
```

5. *MAIN.JAVA*

Archivo para correr la consola

```

1  import java.util.Scanner;
2  import java.util.List;
3  import java.util.ArrayList;
4
5  public class Main {
6      Run | Debug
7      public static void main(String[] args) {
8          Scanner scanner = new Scanner(System.in);
9          List<Playlist> playlists = new ArrayList<>();
10         PlaylistConcreta currentPlaylist = null;
11
12         while (true) {
13             System.out.println(x:"Seleccione una opción:");
14             System.out.println(x:"1. Crear una nueva playlist");
15             System.out.println(x:"2. Seleccionar una playlist existente");
16             System.out.println(x:"3. Agregar una canción a la playlist");
17             System.out.println(x:"4. Eliminar una canción de la playlist");
18             System.out.println(x:"5. Reproducción de la canción");
19             System.out.println(x:"6. Ordenar canciones por duración");
20             System.out.println(x:"7. Ordenar canciones por artista");
21             System.out.println(x:"8. Reproducir canciones en orden aleatorio");
22             System.out.println(x:"9. Salir");
23             int opcion = scanner.nextInt();
24             scanner.nextLine(); // Consumir la nueva línea
25
26             switch (opcion) {
27                 case 1:
28                     currentPlaylist = new PlaylistConcreta();
29                     playlists.add(currentPlaylist);
30                     System.out.println(x:"Nueva playlist creada y seleccionada.");
31                     break;
32                 case 2:
33                     System.out.println(x:"Seleccione el número de la playlist:");
34                     for (int i = 0; i < playlists.size(); i++) {
35                         System.out.println((i + 1) + ". Playlist " + (i + 1));
36                     }
37                     int playlistIndex = scanner.nextInt() - 1;
38                     scanner.nextLine(); // Consumir la nueva línea
39                     if (playlistIndex >= 0 && playlistIndex < playlists.size()) {
40                         currentPlaylist = (PlaylistConcreta) playlists.get(playlistIndex);
41                         System.out.println("Playlist " + (playlistIndex + 1) + " seleccionada.");
42                     } else {
43                         System.out.println(x:"Índice inválido.");
44                     }
45                     break;
46                 case 3:
47                     if (currentPlaylist != null) {
48                         System.out.print(s:"Ingrese el título de la canción: ");
49                         String titulo = scanner.nextLine();
50                         System.out.print(s:"Ingrese el artista de la canción: ");
51                         String artista = scanner.nextLine();
52                     }
53             }
54         }
55     }
56 }

```

```

51         System.out.print(s:"Ingrese la duración de la canción (en minutos): ");
52         int duracion = scanner.nextInt();
53         scanner.nextLine(); // Consumir la nueva línea
54         currentPlaylist.agregarCancion(new CancionConcreta(titulo, artista, duracion));
55         System.out.println(x:"Canción agregada a la playlist.");
56     } else {
57         System.out.println(x:"No hay ninguna playlist seleccionada.");
58     }
59     break;
60     case 4:
61         if (currentPlaylist != null) {
62             System.out.print(s:"Ingrese el título de la canción a eliminar: ");
63             String titulo = scanner.nextLine();
64             Cancion aEliminar = null;
65             for (Cancion cancion : currentPlaylist.getCanciones()) {
66                 if (cancion.getTitulo().equals(titulo)) {
67                     aEliminar = cancion;
68                     break;
69                 }
70             }
71             if (aEliminar != null) {
72                 currentPlaylist.eliminarCancion(aEliminar);
73                 System.out.println(x:"Canción eliminada de la playlist.");
74             } else {
75                 System.out.println(x:"Canción no encontrada.");
76             }
77         } else {
78             System.out.println(x:"No hay ninguna playlist seleccionada.");
79         }
80         break;
81     case 5:
82         if (currentPlaylist != null) {
83             System.out.println(x:"Reproducción de la canción:");
84             for (Cancion cancion : currentPlaylist.getCanciones()) {
85                 System.out.println(cancion.getTitulo() + " - " + cancion.getArtista() + " (" + cancion.getDuracion() + " minutos)");
86             }
87         } else {
88             System.out.println(x:"No hay ninguna playlist seleccionada.");
89         }
90         break;
91     case 6:
92         if (currentPlaylist != null) {
93             currentPlaylist.ordenarPorDuracion();
94             System.out.println(x:"Canciones ordenadas por duración.");
95         } else {
96             System.out.println(x:"No hay ninguna playlist seleccionada.");
97         }
98         break;
99     case 7:
100         if (currentPlaylist != null) {
101             currentPlaylist.ordenarPorArtista();
102             System.out.println(x:"Canciones ordenadas por artista.");
103         } else {
104             System.out.println(x:"No hay ninguna playlist seleccionada.");
105         }
106         break;
107     case 8:

```

```

108         if (currentPlaylist != null) {
109             currentPlaylist.reproducirAleatorio();
110             System.out.println(x:"Reproducción aleatoria activada.");
111         } else {
112             System.out.println(x:"No hay ninguna playlist seleccionada.");
113         }
114         break;
115     case 9:
116         System.out.println(x:"Saliendo...");
117         scanner.close();
118         return;
119     default:
120         System.out.println(x:"Opción inválida. Inténtelo de nuevo.");
121     }
122 }
123 }
124 }

```

EJEMPLO DEL USO

Seleccione una opción:

1. Crear una nueva playlist
2. Seleccionar una playlist existente
3. Agregar una canción a la playlist
4. Eliminar una canción de la playlist
5. Reproducción de la canción
6. Ordenar canciones por duración
7. Ordenar canciones por artista
8. Reproducir canciones en orden aleatorio
9. Salir

1

Nueva playlist creada y seleccionada.

Seleccione una opción:

1. Crear una nueva playlist
2. Seleccionar una playlist existente
3. Agregar una canción a la playlist
4. Eliminar una canción de la playlist
5. Reproducción de la canción
6. Ordenar canciones por duración
7. Ordenar canciones por artista
8. Reproducir canciones en orden aleatorio
9. Salir

3

Ingrese el título de la canción: No Se Va

Ingrese el artista de la canción: Morat

Ingrese la duración de la canción (en minutos): 4

Canción agregada a la playlist.

Seleccione una opción:

1. Crear una nueva playlist
2. Seleccionar una playlist existente
3. Agregar una canción a la playlist
4. Eliminar una canción de la playlist
5. Reproducción de la canción
6. Ordenar canciones por duración
7. Ordenar canciones por artista
8. Reproducir canciones en orden aleatorio
9. Salir

2

Seleccione el número de la playlist:

1. Playlist 1
2. Playlist 2

1

Playlist 1 seleccionada.

Seleccione una opción:

1. Crear una nueva playlist
2. Seleccionar una playlist existente
3. Agregar una canción a la playlist
4. Eliminar una canción de la playlist
5. Reproducción de la canción
6. Ordenar canciones por duración
7. Ordenar canciones por artista
8. Reproducir canciones en orden aleatorio
9. Salir

5

Reproducción de la canción:

No Se Va - Morat (4 minutos)

TSQ - Humbe (3 minutos)

CONCLUSIÓN

En esta aplicación se aprende como se utilizan las clases abstractas y concretas para modelar entidades del mundo real, como lo es las canciones y playlist. Se implementaron las interfaces de usuario en la consola, lo cual permitió al usuario interactuar con la aplicación de forma sencilla.

LINK

<https://github.com/EzryLeal/Playlist.git>