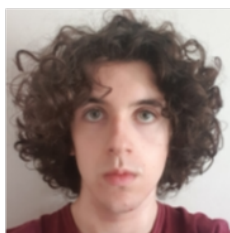


Universidade do Minho
Licenciatura em Ciências da Computação

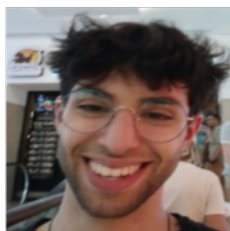
Programação Orientada aos Objetos Trabalho Prático

Relatório



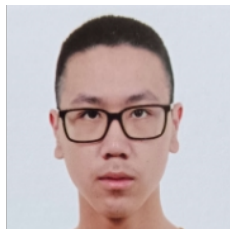
Tiago Silva Costa

(A108657)



João Francisco Sousa Oliveira

(A108647)



Pedro Dong Mo

(A108398)

Conteúdo

1	Introdução	3
2	Classes	4
2.1	Hierarquias	4
2.1.1	Music	4
2.1.2	Playlist	4
2.1.3	SubClasses de Playlist	4
2.1.4	Plan	5
2.2	Interfaces	5
2.2.1	MusicCollection	5
2.3	Simples	5
2.3.1	Album	5
2.3.2	Streamed	5
2.3.3	Genre	5
2.3.4	User	6
2.4	Composição e Agregação	6
2.5	Factory	6
2.6	Exceptions	6
3	Estrutura	7
3.1	Menu Principal (<i>showMainMenu()</i>)	7
3.2	Menu de Utilizador (<i>showUserMenu()</i>)	7
3.3	Menu de Administrador (<i>showAdminMenu()</i>)	7
3.4	Biblioteca (<i>showLibraryMenu()</i>)	8
3.5	Menus de Álbum e Playlist (<i>showAlbumMenu()</i> , <i>showPlaylistMenu()</i>)	8
3.6	Player de Álbuns e Playlists (<i>showSequentialPlayMenu()</i>)	8
3.7	Player de Músicas (<i>showMusicMenu()</i>)	8
3.8	Menu de Playlists Aleatórias (<i>showStatisticsMenu()</i>)	8
3.9	Menus de Estatísticas (<i>showChooseRandomPlaylistMenu()</i>)	8
	Diagrama de Classes	10
4	Ilustração das Funcionalidades	11

Capítulo 1

Introdução

No âmbito da unidade curricular de Programação Orientada aos Objetos, foi-nos solicitado a criação uma aplicação que faça a gestão das músicas, playlists e álbuns que contém músicas que os utilizadores podem ouvir. Também nos foi pedido que os utilizadores tivessem planos de subscrição que davam restrição e permissão de certas funcionalidades dentro da aplicação.

Capítulo 2

Classes

2.1 Hierarquias

2.1.1 Music

Para a classe das músicas, utilizamos uma hierarquia, sendo a super-classe *Music* uma classe abstrata, tendo como sub classes *NormalMusic*, *ExplicitMusic* e *MusicMedia*, para poder fazer uma coleção de músicas mais facilmente e abstraído. Dentro desta classe temos, de diferente um HashSet para os géneros e os intérpretes porque podem ser vários.

2.1.2 Playlist

Para as playlist utilizamos uma superclasse com *Playlist* e sub classes com *Favorites*, *RandomPlaylist* e *GenPlaylist*, isto com o intuito de reutilização de código e abstração na parte de fazer os métodos, fazendo com que a biblioteca por exemplo usufrua desta hierarquia de classes. Esta classe contém, um User (O seu criador), um ArrayList de *Music*, porque ela tem ordem e poder acessar uma certa musica através de um índice, uma String do seu nome e Um Booleano (isPublic) para o User possa trocar para publico ou privado quando quiser.

2.1.3 SubClasses de Playlist

Estas classes diferem na construção e para que o plano possa escolher quais destas é que tem permissão.

RandomPlaylist

Dentro deste construtor, recebe o spotifyUM e o numero de musicas, que pega em várias músicas do spotifyUM e da shuffle e adiciona à playlist.

Favorites

Recebe um *User* e através do histórico dele, faz uma playlist de até 30 músicas com os géneros mais ouvidos.

GenPlaylist

Foram implementados três construtores distintos conforme o enunciado, que permite a criação de playlists com base num conjunto de géneros, numa restrição de tempo total ou num tipo específico de música.

2.1.4 Plan

Acerca dos planos de subscrição, a hierarquia é composta por a super-classe abstrata *Plan* e as subclasses *PlanFree*, *PlanBasePremium* e *PlanTopPremium*, que são responsáveis por dar as permissões, vantagens e restrições aos utilizadores, através de métodos simples, para facilitar a criação de novos planos.

2.2 Interfaces

2.2.1 MusicCollection

Esta interface é utilizada pela hierarquia *Playlist* e da classe *album*, a biblioteca e o play destas classes aproveitam se disto, para abstração e reutilização de código.

2.3 Simples

2.3.1 Album

A classe dos álbuns, tem o mesmo conceito das playlists, diferem apenas nos dados, os artistas em vez de um user e são sempre publicos.

2.3.2 Streamed

Esta classe apenas serve para guardar como histórico do que um *User* ouviu, com um *Music* e um *LocalDateTime*.

2.3.3 Genre

Os géneros de uma musica são apenas um Enum.

2.3.4 User

E a classe do utilizador, tem os seus dados, um ArrayList de MusicCollection, e de Streamed (Agregação), por causa de ordem.

2.4 Composição e Agregação

O User utiliza a composição para a biblioteca, porque só existe um álbum ou playlist desse Artista ou User e Agregação para o Streamed e Plano, porque é único do User. As MusicCollection, tanto como as playlists e os álbuns, utilizam composição para as músicas e o criador(User), pelo mesmo motivo, apenas são únicos, e por exemplo para a música, serem todos sincronizados com o numero de streams.

2.5 Factory

Para a criação de um User, Playlists, Albuns, etc, de uma maneira mais dinâmica, é uma interação através do scanner e escolhas do cliente.

2.6 Exceptions

No uso desta aplicação, pode haver exceções, como NoMusicException, por exemplo, quando um User quer criar uma playlist, mas não existem musicas e NoUserException, por exemplo, quando for ver as estatísticas, sem existir um User.

Capítulo 3

Estrutura

A estrutura da aplicação assenta numa classe principal (*main*) que tem como responsabilidade carregar para a aplicação os dados guardados da última utilização. De seguida, é criado um plano e um utilizador administrador (*admin*), que terá acesso a uma área destinada à criação de músicas, playlists (normais e aleatórias) e álbuns. Posteriormente, é criado um objeto *app*, que possui um método responsável por executar a lógica dos menus da aplicação, ou seja, permite ao utilizador escolher as ações que pretende realizar.

3.1 Menu Principal (*showMainMenu()*)

Este menu dá acesso a várias opções: realizar login com o *handle* de um utilizador, criar um novo utilizador, entrar no modo de administração para criação de objetos da aplicação, e ainda aceder ao menu de estatísticas da aplicação.

3.2 Menu de Utilizador (*showUserMenu()*)

Após o login, o utilizador tem acesso ao menu pessoal, que, dependendo do seu plano de subscrição, disponibiliza várias funcionalidades: acesso à biblioteca pessoal, visualização de todas as músicas, álbuns e playlists públicas da aplicação, pesquisa e reprodução de músicas, álbuns, playlists públicas e aleatórias. Também é possível consultar o histórico de músicas ou os pontos acumulados, bem como o tipo de plano de subscrição associado à conta.

3.3 Menu de Administrador (*showAdminMenu()*)

Este menu está reservado ao utilizador administrador e permite a criação de músicas, álbuns, playlists (normais e aleatórias), além de permitir visualizar a quantidade de objetos já existentes na aplicação.

3.4 Biblioteca (*showLibraryMenu()*)

No menu da biblioteca, também sujeito às permissões do plano de subscrição, é possível visualizar todos os objetos que pertencem à biblioteca do utilizador (músicas, álbuns, etc.), assim como reproduzi-los. Adicionalmente, é possível criar playlists normais (com opção de torná-las públicas ou privadas), playlists de favoritos e playlists geradas automaticamente com base em critérios como estilo musical, duração ou tipo de música.

3.5 Menus de Álbum e Playlist (*showAlbumMenu()*, *showPlaylistMenu()*)

As funcionalidades destes menus são semelhantes. Permitem reproduzir as músicas de um álbum ou playlist, visualizar os seus conteúdos e adicioná-los à biblioteca pessoal do utilizador.

3.6 Player de Álbuns e Playlists (*showSequential-PlayMenu()*)

Este menu, consoante as permissões do utilizador, permite a reprodução de músicas de forma sequencial ou aleatória, bem como avançar ou retroceder entre as faixas.

3.7 Player de Músicas (*showMusicMenu()*)

Integrado no menu do utilizador, este menu permite reproduzir uma música individualmente, consultar a sua informação detalhada e adicioná-la a uma playlist existente.

3.8 Menu de Playlists Aleatórias (*showStatistics-Menu()*)

Este menu apresenta todas as playlists aleatórias criadas pelo administrador. Após a seleção de uma playlist, o utilizador é redirecionado para o menu *showRandomPlaylistMenu()*, onde pode consultar o seu conteúdo e reproduzir as músicas numa só direção.

3.9 Menus de Estatísticas (*showChooseRandom-PlaylistMenu()*)

Este menu apresenta informações estatísticas sobre a aplicação, como:

- Música, artista e género mais reproduzidos;
- Utilizador que ouviu mais músicas (no total ou num determinado intervalo de tempo);
- Utilizador com mais pontos;
- Quantidade de playlists públicas existentes na aplicação;
- Utilizador com mais playlists na sua biblioteca.

Aqui é apresentado o diagrama de classes da aplicação, que representa a estrutura e as relações entre os diferentes componentes do sistema. Caso as classes não estejam legíveis por motivos visuais, disponibilizamos o diagrama online, bastando pressionar *Open with Draw.io* no Google Drive para aceder à aplicação.



Figura 3.1: Diagrama de classes da aplicação.

Capítulo 4

Ilustração das Funcionalidades

Neste capítulo são apresentadas as funcionalidades da aplicação, sendo utilizado para isso *screenshots* do terminal, simulando o uso por parte de um utilizador.

As imagens seguintes mostram os menus disponíveis, quer para utilizadores comuns, quer para administradores, bem como outras interações relevantes na utilização da aplicação.

```

joao@Figure:~/uni/GrupoTP-49$ make compile
Compiling source files...
javac -d bin src/Album.java src/AlbumFactory.java src/AppController.java src/ExplicitMusic.java src/Favorites.java src/GenPlaylist.java src/GenPlaylistFactory.java src/Genre.java src/
Main.java src/MediaMusic.java src/MediaHandler.java src/Music.java src/MusicCollection.java src/MusicFactory.java src/MusicException.java src/NoMusicException.java src/NormalMusic.j
ava src/Plan.java src/PlanMusic.java src/PlanMusicPremium.java src/PlanFree.java src/PlanPremium.java src/Playlist.java src/PlaylistFactory.java src/RandomPlaylist.java src/RandomPl
aylistFactory.java src/SpotifyUM.java src/Streamed.java src/User.java src/UserFactory.java

```

Figura 4.1: Compilar aplicação

```

joao@Figure:~/uni/GrupoTP-49$ make script
Compiling source files...
javac -d bin src/Album.java src/AlbumFactory.java src/AppController.java src/ExplicitMusic.java src/Favorites.java src/GenPlaylist.java src/GenPlaylistFactory.java src/Genre.java src/
Main.java src/MediaMusic.java src/MediaHandler.java src/Music.java src/MusicCollection.java src/MusicFactory.java src/MusicException.java src/NoMusicException.java src/NormalMusic.j
ava src/Plan.java src/PlanMusic.java src/PlanMusicPremium.java src/PlanFree.java src/PlanPremium.java src/Playlist.java src/PlaylistFactory.java src/RandomPlaylist.java src/RandomPl
aylistFactory.java src/SpotifyUM.java src/Streamed.java src/User.java src/UserFactory.java
Preparing the input file...

```

Figura 4.2: Compilar Script

```

joao@Figure:~/uni/GrupoTP-49$ make run
java -cp bin Main

Welcome! Choose an option:
1. Login as user
2. Register new user
3. Admin mode
4. Statistics' menu
0. Exit
Option: 

```

Figura 4.3: Arranque da aplicação e Menu Principal

```

Admin mode activated.

Admin options:
1. Create music
2. Create album
3. Create Playlist
4. Create random playlist
5. Show musics
6. Show albums
7. Show playlists
8. Show random playlists
0. Exit admin
Option: 

```

Figura 4.4: Menu de administrador

```

Option: 4

Choose an option:
1. Most played song
2. Most listened to interpreter
3. Which user has listened to the most songs over a period of time or ever since?
4. User with the most points
5. Most Played Genre
6. how many public playlists are on SpotifyUM
7. User with the most playlists
0. Exit
Option: 

```

Figura 4.5: Menu de Estatísticas

```

Select user:
user1 - user withPremiumBase (user1@gmail.com)
user2 - user withFREE (user2@gmail.com)
pepega - pepe PremiumTop. (pepega@hotmail.com)
admin - Admin ()
Enter your handle to login: █

```

Figura 4.6: Login

```

Welcome pepe PremiumTop.!

Choose an option:
1. Library
2. Play Random Playlists
3. Show all Music Names
4. Search a Music
5. Show all Albums
6. Play an Album
7. Show all public Playlists
8. Play a public playlist
9. Show user's history
10. My Points
11. Show Subscription Plan
0. Exit
Option: █

```

Figura 4.7: Menu de Utilizador

```

Library options:
1. Show your Library
2. Play
3. Create a Playlist
4. Create your Favourite Playlist
5. Generate a Playlist according to criteria
6. Make Playlist Public/Private
0. Exit
Option: █

```

Figura 4.8: Biblioteca

```

Name for the Playlist:
PlaylistCriteria:
Select Generated Playlist type:
1. Playlist with the user's musical preference
2. Playlist with the user's musical preference and restricted to a certain maximum time
3. Playlist with the user's musical preference but only with songs of the Explicit Music type
Option: █

```

Figura 4.9: Opções para playlist geradas

```

Enter music to search (or 'exit'): MusicWithAllGenres

MusicWithAllGenres
Music options:
1. Play
2. Show Music info
3. Add to a playlist
0. Exit
Option: █

```

Figura 4.10: Menu de Musica

```

Option: 10
Your total Points: 102

```

Figura 4.11: Ver pontos de Utilizador

```

Choosing Random Playlists:
1. Show Random Playlists
2. Choose
0. Exit.
Option: █

```

Figura 4.12: Menu para Playlists aleatórias

```

Showing available Random Playlists:
[RandomPlaylist2, RandomPlaylist1]
Enter a playlist name to play (or 'exit'): RandomPlaylist2

RandomPlaylist2
Play Random Playlists options:
1. Show Musics
2. Play the Playlist
0. Exit.
Option: █

```

Figura 4.13: Menu da Playlist Aleatória

```
Option: 2
Playing RandomPlaylist2

Digital Ghosts
Play Music options:
1. Play
2. Show Music Info
0. Exit
Option: █
```

Figura 4.14: Tocar Playlist Aleatória

```
Option: 8
Showing available playlists:
[PlaylistCriada2, PlaylistCriada1]
Enter Playlist to play (or 'exit'): PlaylistCriada2
Playing: PlaylistCriada2
Playlist options:
1. Show musics
2. Play
3. Add to Library
0. Exit.
Option: █
```

Figura 4.15: Menu da Playlist

```
Now playing: Digital Ghosts
Play options:
1. Play this song
2. Next Song
3. Previous Song
4. Show all Songs
5. Random Song
0. Exit
Option: █
```

Figura 4.16: Tocar Playlist

```
Option: 9
Showing available Albums:
[A Right at the Opera, Echoes in the Dark, Rebirth and Ruin, AlbumCriado1, AlbumWithAllGenres, Electric Phantom]
Enter Album to play (or 'exit'): Rebirth and Ruin
Playing: Rebirth and Ruin
Album options:
1. Show musics
2. Play
3. Add to Library
0. Exit.
Option: █
```

Figura 4.17: Menu do Album

```
Now playing: From Ashes We Sing
Play options:
1. Play this song
2. Next Song
3. Previous Song
4. Show all Songs
0. Exit
Option: █
```

Figura 4.18: Tocar Album

```
Showing user's history from the oldest to the most recent:
Streamed[musicName="Name: MusicWithAllGenres
Interpreter: [Artist1]
Record Label: Producers1
Genres: [HIPHOP, ROCK, COUNTRY, RNB, CLASSICAL, JAZZ, ELECTRONIC, POP, FUNK, SOUL]
Lyrics:
"
This is a text to put
on the lyrics of the
music with all Music
with all genres
"
Music:
"
A B C D E F G
H I J K L M N
O P Q R S T U
"
Duration: 15
Stream: 1
', time=2025-05-17T10:20:45.745215708)
Your total Points: 102
```

Figura 4.19: Histórico Utilizador

```
Option: 11
Subscription Plan: PlanTopPremium@3d71d552
```

Figura 4.20: Ver plano do Utilizador

Capítulo 5

Conclusão

Este projeto permitiu aplicar os principais conceitos da Programação Orientada aos Objetos em Java, com foco na reutilização de código, modularidade e organização da aplicação em diferentes camadas. Através da implementação de hierarquias de classes e do uso de interfaces, foi possível criar uma base sólida e escalável para uma aplicação multimédia interativa.