

Universidade do Minho

Escola de Engenharia

Licenciatura em Engenharia Informática

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Programação Orientada aos Objetos

Ano Letivo de 2024/2025

POO - SpotifUM - Grupo 41

André Carvalho
a100818

Sandro Coelho
a105672

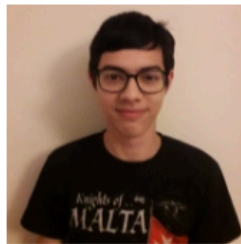
Duarte Barbosa
a104184

Maio, 2025

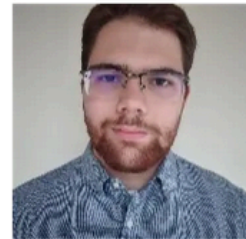
POO



Sandro Coelho
a105672



André Carvalho
a100818



Duarte Barbosa
a104184

Figura 1: Alunos Pertencentes ao Projeto

Índice

1. Introdução	1
2. Arquitetura de Classes	2
2.1. Entidades da aplicação	2
2.2. Exceções	2
2.3. Interação com o utilizador	3
3. Descrição da aplicação	4
4. Conclusão	10

1. Introdução

Através da Unidade Curricular de Programação Orientada aos Objetos, foi pedido para que o nosso grupo desenvolve-se uma aplicação de gestão de músicas e listas de reprodução. Esta aplicação permite assim a gestão desses conteúdos, playlists, álbuns e a gestão de utilizadores para o uso da própria aplicação, havendo a existência de utilizadores com permissões diferentes que outros utilizadores. Sendo ainda implementados uma área de estatísticas de vários dados da aplicação, a existência de músicas explícitas e playlist de acordo com as preferências e critérios do utilizador.

2. Arquitetura de Classes

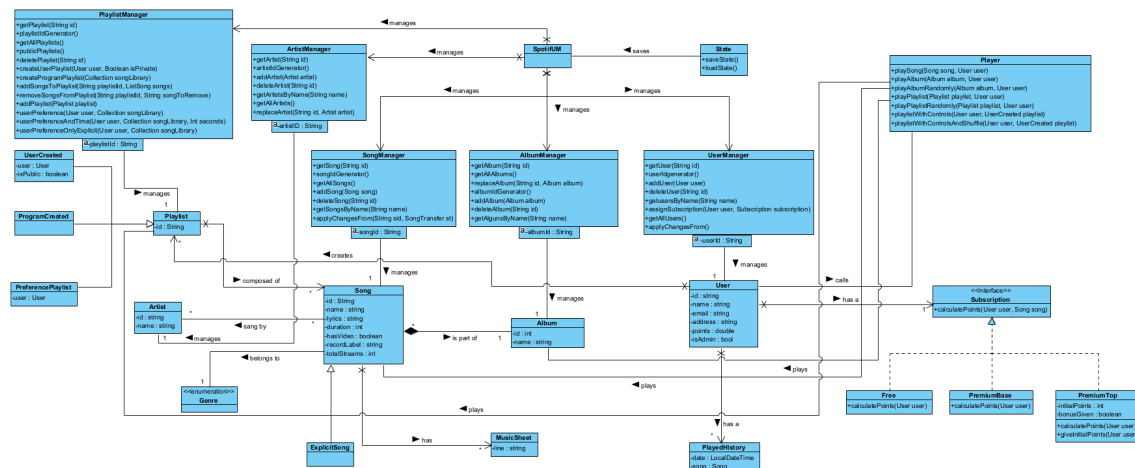


Figura 2: Diagrama de classes

2.1. Entidades da aplicação

As principais classes da nossa arquitetura são a música (Song), o álbum (Album), o utilizador (User), o artista (Artist) e a playlist (Playlist), sendo que cada uma destas é gerida por um *Manager* que contém um mapa que armazena o respetivo objeto e utiliza o seu Id como chave. Todos estes *Managers* são por sua vez geridos por um *Manager* geral que recebeu o nome da aplicação, SpotifyUM, que reencaminha os pedidos do Controller para os respetivos managers.

Existem três tipos de playlists que estendem a classe mãe, as *UserCreated* que são criadas pelos utilizadores Premium e podem, ou não, ser públicas. As *ProgramCreated* que são geradas de forma aleatória pelos administradores e são sempre públicas. Por fim, as *PreferencePlaylists* são geradas pelos utilizadores de forma aleatória e utilizam o seu histórico de músicas ouvidas para que os resultados sejam próximos às suas preferências.

Os géneros musicais são uma classe *enum* sendo assim mais fácil controlar, gerir e adicionar novos géneros caso necessário.

As subscrições são facilmente adicionadas uma vez que estas implementam uma interface *Subscription* que facilita também o *typechecking* que é necessário para verificar se o utilizador possui uma subscrição premium para certas ações.

Todas as funções de reprodução de conteúdos estão contidas no *Player*.

2.2. Exceções

Foram definidas diversas exceções para gerir erros e dar alguma cobertura ao utilizador. Utilizadores, artistas, músicas e álbuns partilham algumas exceções que lidam com duplicados (*DuplicateAlbumException* por exemplo), com entradas inválidas (*InvalidAlbumException*) e com *nulls* (*AlbumNotFoundException*).

Existe ainda uma exceção que é levantada sempre que um utilizador tenta aceder a algo que não tem permissão (`UnauthorizedAccessException`) e uma que é levantada no caso de não existirem músicas suficientes para a criação de uma playlist (`InsufficientSongsException`).

Olhando para trás teria sido melhor condensar algumas exceções numa só, por exemplo, tornar as exceções de duplicados, entradas inválidas e *nulls* numa só exceção chamada “`AlbumException`”, no caso dos álbuns, o que reduziria a duplicação de código e o tornaria mais fácil de manter.

2.3. Interação com o utilizador

A nossa aplicação utiliza o padrão MVC (*Model-Controller-View*) onde a pasta *Model* guarda todas as classes do projeto, a *View* que armazena todos os ficheiros referentes à UI e o *Controller* que serve como ponte entre o *Model* e a *View*. Este é invocado na main e é ele que trata de todas as operações, invocando funções disponíveis no *Model* e na *View*.

O controlador chama os métodos da *View* que imprimem a *UI* na consola e recolhem as escolhas do utilizador. O controlador recebe estas escolhas e invoca os métodos que se encontram no *Model* para responder aos pedidos do utilizador.

Para passar os dados dos utilizadores e músicas entre a *View* e o controlador foram utilizadas duas classes denominadas *Data Transfer Objects* que contêm os atributos que pretendemos passar para o controlador para que ele os possa processar posteriormente. Estas classes não contêm qualquer tipo de lógica e foram utilizadas para que a *View* não tivesse acesso à criação de músicas ou utilizadores. Estes DTOs são utilizados tanto na criação como na edição.

Todas as exceções são “apanhadas” no controlador e impressas na consola utilizando métodos presentes na *View*, sendo que, apenas esta é capaz de imprimir na consola ou coletar inputs. Há, no entanto, uma exceção, o método de *playSong* imprime a letra da musica diretamente na consola, uma vez que, no contexto do nosso programa, isso simula a reprodução de uma música.

Houve a necessidade de integrar diretamente a recolha de inputs e a impressão de mensagens no método de reprodução de playlist com controlos (seguinte/anterior). Esta exceção foi feita porque, como a reprodução de música é apenas simulada no terminal, os controlos de navegação precisam ocorrer em tempo real durante o “momento de reprodução”.

Apesar destas duas exceções, o restante do programa segue o padrão MVC.

3. Descrição da aplicação

Após correr a aplicação, o utilizador será deparado com uma tela de *login* onde terá de escolher o seu *userID* já definido pelo *admin*.

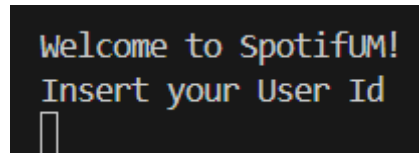


Figura 3: Tela de Login

De seguida, o utilizador depara-se com o seguinte menu:

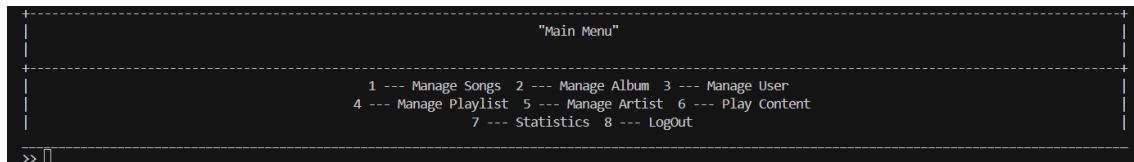


Figura 4: Menu Principal da Aplicação

Para que seja possível progredir na aplicação, será necessário que seja criado um álbum e artista, devido a esses dados serem necessários para criar uma música. Porém, esses dados apenas podem ser criados por um utilizador que tenha autoridade de *admin*. Caso o mesmo seja um *admin*, conseguirá ter acesso a todas as opções disponíveis.

Tal como todas as opções de criação, detalhes e remoção de músicas, álbuns, artistas, utilizadores e playlists utilizam uma lógica similar de criação para os mesmos, com a exceção de certas opções das playlists que iremos abordar.

Quando se seleciona o menu das músicas, são entregues as seguintes opções:

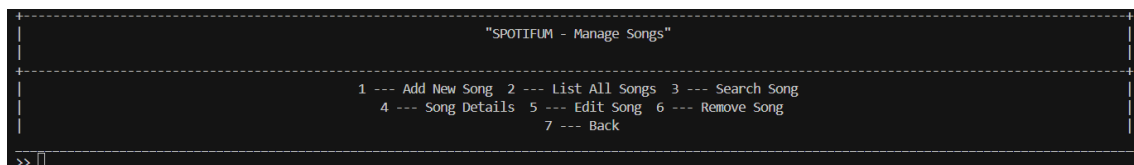


Figura 5: Menu das Músicas

Assim que se seleciona a opção de criação de uma música, é pedido para entregar o *ID* do álbum que esta música irá pertencer, caso este seja recusado ou mal entregue pelo utilizador, retoma ao menu inicial. Em seguida pede pelo nome da música, as letras da mesma e o seu género, seguida pela sua duração, e por informação se a mesma é explícita e se tem vídeo clipe, seguindo com a gravadora, e com os artistas, sendo que é possível ter mais que um artista por musica foi decidido que seria possível adicionar artistas à música ate que houvesse um *input* de "0" pelo utilizador, acabando com a quantidade de linhas da *music sheet* da musica e o valor das mesmas.

```

Please insert the album ID:
A5
=== Add New Song ===
Enter song name: FairyTale
Enter song lyrics: Im in love with the a fairytale
Enter genre (e.g., POP, ROCK, HIPHOP): FOLK
Enter duration (in seconds): 185
Is the song explicit? (1 for Yes, 0 for No): 0
Does the song have a video? (1 for Yes, 0 for No): 1
Enter record label: Universal Studios
Enter artist ID (0 to finish): Ar4
Enter artist ID (0 to finish): 0
How many lines does the music sheet have? 1
Enter line 1: abcabcabc

```

Figura 6: Criação de uma Música

Para realizar a listagem de músicas, apenas temos de realizar o *input* e recebemos todas as músicas existentes e alguma informação de forma reduzida das mesmas, tal como o *ID* e o nome da música, o *ID* do álbum e o seu nome, a quantidade de vezes que a mesma foi reproduzida e o *ID* e nome de cada artista que participa na mesma.

```

=== Songs List ===
ID: S3, Name: Andromeda, Album Id: A1, Album Name: Still Still Stellar, Stream Count: 5, Artist ID: Ar2, Artist Name: Suisei //
ID: S4, Name: Je t'aime, Album Id: A1, Album Name: Still Still Stellar, Stream Count: 6, Artist ID: Ar2, Artist Name: Suisei //
, Artist ID: Ar1, Artist Name: Ado //
ID: S5, Name: 7days, Album Id: A2, Album Name: Spectre, Stream Count: 8, Artist ID: Ar2, Artist Name: Suisei //
ID: S11, Name: FINALE, Album Id: A5, Album Name: UNDERTALE SoundTrack, Stream Count: 3, Artist ID: Ar4, Artist Name: Toby Fox //
ID: S6, Name: Crime and Punishment, Album Id: A4, Album Name: Ado's Utatemita, Stream Count: 8, Artist ID: Ar1, Artist Name: Ado //
ID: S10, Name: Fallen Down, Album Id: A5, Album Name: UNDERTALE SoundTrack, Stream Count: 1, Artist ID: Ar4, Artist Name: Toby Fox //
ID: S7, Name: Aishite Aishite Aishite, Album Id: A4, Album Name: Ado's Utatemita, Stream Count: 5, Artist ID: Ar1, Artist Name: Ado //
ID: S13, Name: Sweet Child o mine, Album Id: A8, Album Name: Appetite For Destruction, Stream Count: 2, Artist ID: Ar7, Artist Name: Guns N Roses //
ID: S12, Name: How you like that, Album Id: A7, Album Name: The Album, Stream Count: 1, Artist ID: Ar6, Artist Name: BLACKPINK //
, Artist ID: Ar1, Artist Name: Ado //
, Artist ID: Ar2, Artist Name: Suisei //
ID: S9, Name: Godish, Album Id: A4, Album Name: Ado's Utatemita, Stream Count: 5, Artist ID: Ar1, Artist Name: Ado //
ID: S1, Name: Stolen Dance, Album Id: A6, Album Name: Sadnecessary, Stream Count: 4, Artist ID: Ar5, Artist Name: Milky Chance //
ID: S2, Name: FairyTale, Album Id: A3, Album Name: FairyTales, Stream Count: 3, Artist ID: Ar3, Artist Name: Alexander Rybak //

```

Figura 7: Listagem de todas as Músicas

Temos de seguida as opções de procurar músicas de 2 maneiras, sendo que o *search* de uma música é realizada através do nome da música, enquanto o detalhes de uma música se realizam através do *ID* da mesma. Se houver mais que uma música com o mesmo nome, ambas serão disponibilizadas, assim como mais informações da mesma. Se um utilizador quiser mais informações sobre uma música, este deve usar a opção de detalhes de uma musica, aparecendo assim todas as informações possíveis.

```

=== Search Song ===
Insert the song name:
FairyTale
Songs found:
ID: S16, Name: FairyTale, Album Id: A5, Album Name: Fairytales, Duration: 185s, Stream Count: 0, Artist ID: Ar4, Artist Name: Alexander Rybak //

```

Figura 8: Search de uma música através do seu nome

```

=== Song Details ===
Insert the song ID:
S16
Song{id='S16', name='FairyTale', lyrics='Im in love with the a fairytale', genre=FOLK, duration=185, hasVideo=true, recordLabel='Universal Studios', totalStreams=0, album=A5 - Fairytales, music sheet=[abcabcabc] , artists=[Artist {id=Ar4, name=Alexander Rybak}]} | Explicit: No

```

Figura 9: Detalhes de uma música através do seu ID

Para editar as informações de uma música, tem de ser entregue o *ID* da música, seguindo pelo aspeto que se quer editar. Realizando o mesmo de forma correta altera a música em todas as maneiras comuns.


```

=== Edit Song ===
Insert the song ID:
S16
What do you want to edit?
1 --- Name
2 --- Lyrics
3 --- Genre
4 --- Duration
5 --- Explicit
6 --- Video
7 --- Record Label
8 --- Music Sheet
9 --- Back to Main Menu

```

Figura 10: Opções de edição de uma música

Por último, para realizar a remoção de uma música é pedido pelo *ID* da mesma, acabando com a remoção da mesma de toda a aplicação.

```

=== Remove Song ===
Insert the song ID:
S2
Song deleted successfully!

```

Figura 11: Remoção de uma música

Tal como foi dito anteriormente, esta lógica de tratamento de dados aplica-se para os álbuns, artistas, utilizadores e playlists, sendo que as playlists possuem informações extra para as mesmas.

Este menu das playlists difere através das seguintes opções, adição e remoção de músicas para a playlist, a geração de playlists de forma automática (geradas pelo *admin* para todos os utilizadores) e a geração de playlists de acordo com as preferências musicais do utilizador.

```

"SPOTIFUM - Manage Playlists"
-----
1 --- Add New Playlist 2 --- List All Public Playlists 3 --- Playlist Details
4 --- Remove Playlist 5 --- Add Song to Playlist 6 --- Remove Song from Playlist
7 --- Generate Auto Playlist 8 --- Preference Playlist Menu 9 --- Back
>>

```

Figura 12: Menu das Playlists

Para a adição e remoção de músicas, apenas é necessário entregar o *ID* da playlist e das músicas em questão. A criação de playlists automáticas apenas pode ser realizada pelo *admin*, sendo que estas estão disponibilizadas para todos os utilizadores. Por fim, para a última opção das playlists, encontra-se a possibilidade de criar playlists através das preferências musicais do utilizador, podendo ser baseadas pelo género de música mais ouvido, pelo género mais ouvido e um limite de tempo para a playlist e a criação de uma playlist de acordo com o género mais ouvido mas contendo apenas músicas explícitas.

```

"SPOTIFUM - Manage Playlists"
-----
1 --- Playlist based in Genre 2 --- Playlist based in Genre with max duration (in seconds) 3 --- Playlist based in Genre with only explicit songs
4 --- Back
>>

```

Figura 13: Menu das Playlists de Preferências Musicais

Um utilizador pode também reproduzir uma música, playlist ou álbum, sendo que quando reproduz um álbum ou playlist, pode escolher entre ouvir a mesma na ordem da playlist normal ou dar *shuffle* na reprodução da playlist, porém, caso o mesmo for de subscription *Free* o mesmo não conseguirá reproduzir as playlists em shuffle nem conseguirá avançar ou retroceder na playlist, enquanto que essas opções se encontram disponíveis para os utilizadores *premium*.

```
=== Play Song ===
Insert the song ID:
S16
Now Playing: FairyTale
Lyrics:
Im in love with the a fairytale
Success: Song Played With Success
```

Figura 14: Reprodução de uma música

```
=== Play Playlist ===
Insert the playlist ID:
P5
Select the playlist mode:
0 --- Normal
1 --- Random
2 --- Back to Menu
0
Now Playing: FINALE
Lyrics:
No Lyrics
[n] Next | [p] Previous | [q] Quit
n
Now Playing: Crime and Punishment
Lyrics:
Hou wo sasusasa no yamatedoori
[n] Next | [p] Previous | [q] Quit
n
```

Figura 15: Reprodução de uma playlist UserCreated por ordem

```

=== Play Playlist ===
Insert the playlist ID:
p5
Select the playlist mode:
0 --- Normal
1 --- Random
2 --- Back to Menu
1
Now Playing: Crime and Punishment
Lyrics:
Hou wo sasu asa no yamatedoori
[n] Next | [p] Previous | [q] Quit
n
Now Playing: Aishite Aishite Aishite
Lyrics:
Aishite x3
[n] Next | [p] Previous | [q] Quit
n

```

Figura 16: Reprodução de uma playlist UserCreated em shuffle

```

=== Play Playlist ===
Insert the playlist ID:
P7
Select the playlist mode:
0 --- Normal
1 --- Random
2 --- Back to Menu
1
Now Playing: FINALE
Lyrics:
No Lyrics
Now Playing: Stolen Dance
Lyrics:
I want you by my side
Now Playing: Sweet Child o mine
Lyrics:
Shes got a smile that it seems to me
Now Playing: Crime and Punishment
Lyrics:
Hou wo sasu asa no yamatedoori
Now Playing: 7days
Lyrics:
kizukanai mama totsuzen

```

Figura 17: Reprodução de uma playlist Auto Created

Por fim, encontram-se o menu das estatísticas da aplicação, em que estão disponíveis as seguintes estatísticas.



Figura 18: Menu das Estatísticas

Estas são as estatísticas específicas à aplicação, sendo que o utilizador apenas precisa de seleccionar qual estatística pretende visualizar e recebe o resultado da mesma.

4. Conclusão

Em suma, consideramos que a aplicação foi desenvolvida com sucesso tendo conseguido satisfazer todos os requisitos, desde os mais básicos como adicionar músicas até à geração de playlists de acordo com preferências. Foram utilizados diversos recursos da linguagem Java como por exemplo Enums, as interfaces “Comparable” e “Serializable”, entre outros.

O projeto apresentou vários desafios que exigiram investigação e adaptação por parte do grupo, o que resultou numa aprendizagem significativa a nível técnico.

Estamos, por isso, bastante satisfeitos com o trabalho realizado e confiantes de que conseguimos consolidar os conteúdos lecionados, respeitando os objetivos propostos e entregando uma aplicação funcional, coesa e bem estruturada.