

Projecto prático de Programação Orientada aos Objectos

LEI + LCC

edição 24/25

Conteúdo

1	Introdução	1
1.1	As <i>Músicas</i>	2
1.2	As <i>Playlists</i>	2
1.3	Os <i>Utilizadores</i>	2
2	O Funcionamento do programa	3
3	Requisitos	3
3.1	Requisitos base de gestão das entidades	3
3.2	Efectuar estatísticas sobre o estado do programa	3
3.3	Criar a noção de música <i>MusicaExplicita</i>	4
3.4	Gerar uma playlist de acordo com critérios	4
4	Relatório	4
5	Salvaguarda do estado da aplicação	5
6	Cronograma	5

1 Introdução

Considere que se pretende criar uma aplicação que faça a gestão das músicas e listas de reprodução de músicas que os utilizadores podem ouvir. Essa aplicação, o **SpotifUM**, deverá permitir ter como conteúdos principais as **músicas** e os **utilizadores**. Em relação às músicas estas poderão estar também organizadas, e poderem ser pesquisadas, por álbuns e listas de reprodução (*playlists*).

Em relação aos utilizadores estes deverão estar categorizados em utilizadores que são utilizadores ocasionais ou utilizadores premium. Utilizadores ocasionais terão acesso apenas a músicas que são reproduzidas de forma aleatória e utilizadores premium, dependendo do seu plano de subscrição poderão ter a possibilidade de criar playlists, guardar playlists e álbuns para posterior reprodução.

Relacionado com os utilizadores existe o conceito de plano de subscrição. A aplicação deverá permitir que no futuro novos planos de subscrição sejam criados e que sejam adicionados à aplicação de forma simples (obrigando apenas a compilar as novas classes). De momento os planos existentes deverão permitir a criação de uma biblioteca pessoal de álbuns e playlists e deverão permitir ter playlists em que o utilizador possa avançar ou retroceder nas músicas. As playlists mais básicas só permitem avançar para a música seguinte.

A aplicação **SpotifUM** para os seus utilizadores premium faz a análise dos seus hábitos de reprodução de músicas (e dos artistas que as tocam) e poderá criar para os utilizadores a quem tal se aplique, listas de músicas geradas automaticamente e que correspondam ao estilo de músicas que ele usualmente consome. Essas listas de reprodução serão designadas por *Listas de favoritos*.

O objectivo da aplicação deverá ser fazer a gestão destas entidades e permitir reproduzir as músicas a que um utilizador tem acesso. De acordo com o seu perfil, um utilizador poderá escolher reproduzir uma música, um álbum ou uma playlist. Por uma questão de simulação, e simplificação, a reprodução das músicas é feita de forma simplificada enviando para o écran o texto com a letra da música.

A aplicação deverá também colectar informação de indicadores de utilização, nomeadamente sabendo que utilizadores reproduziram determinadas músicas e quantas vezes é que o fizeram. Essa informação poderá ser depois utilizada para a geração de playlists personalizadas para cada utilizador.

1.1 As *Músicas*

Uma música é uma entidade que tem as seguintes características:

- nome
- intérprete
- nome da editora
- letra ou poema
- música (representada por uma colecção de linhas de caracteres musicais representados em texto)
- género musical (ex: pop, rock, clássica, etc.)
- duração em segundos

A música aparece sempre no contexto de um álbum e pode possuir algumas características que a identifiquem. Por exemplo, existem músicas que são classificadas como sendo do tipo **Explícita** e outras como sendo do tipo **Multimédia** porque possuem vídeo associado à música. As músicas que não possuem nenhuma destas características não possuem esta informação.

Sempre que uma música é reproduzida, quer seja individualmente quer seja no contexto de uma playlist, é actualizado um contador que determina o número de reproduções.

1.2 As *Playlists*

O programa deve permitir ter o conceito de **Playlist** como sendo uma colecção de músicas, obtida de forma aleatória ou então construída pelos utilizadores. Só os utilizadores premium é que podem construir playlists. As playlists aleatórias, as que podem ser reproduzidas pelos utilizadores não pagantes, apenas permitem ser reproduzidas e a ordem de reprodução é determinada pelo **SpotifUM**. As playlists construídas pelos utilizadores premium permitem que este avance na reprodução de músicas ou retroceda na mesma. Para estas playlists a ordem das músicas é definida pelo utilizador, sendo que se este o desejar poderá colocar as músicas a serem reproduzidas em modo aleatório.

Existe um outro tipo de playlist, as Listas de Favoritos, que é gerada em função dos gostos musicais que se podem inferir dos utilizadores (apenas dos premium). Um outro tipo de playlist é aquele que podem ser definidas com base num tempo máximo de um determinado género musical (ex: gerar uma playlist de música clássica com 60 minutos de duração).

1.3 Os *Utilizadores*

No que diz respeito aos utilizadores, estes deverão possuir informação acerca do seu nome, email e morada. Os utilizadores premium deverão ter informação de qual é o seu plano de subscrição. Neste momento, existe o plano de subscrição *Free*, que é o plano por omissão que apenas permite reproduzir músicas aleatórias que o **SpotifUM** determina, o plano *PremiumBase*, que permite aos utilizadores criarem uma biblioteca com álbuns existentes e criarem playlists, e o plano *PremiumTop* que permite aos utilizadores terem as funcionalidades

do *PremiumBase* mais a capacidade de ter acesso a playlists geradas pela aplicação com base nas preferências inferidas a partir da experiência do utilizador.

Os utilizadores com planos *PremiumBase* e *PremiumTop* poderão criar playlists e torná-las públicas o que permite a outros utilizadores poderem acederem às mesmas e incorporá-las na sua biblioteca.

Por cada música que for reproduzida cada utilizador receberá *pontos*. Os utilizadores dos planos Free recebem 5 pontos por música, os do plano *PremiumBase* recebem 10 pontos por cada música reproduzida e os do plano *PremiumTop* recebem 100 pontos pela adesão e por cada nova música reproduzida 2.5% dos pontos já acumulados.

2 O Funcionamento do programa

Pretende-se com este programa criar um ambiente que permita criar uma gestão de conteúdos musicais e dos utilizadores que nela estão inscritos.

1. o programa deve ter uma fase em que é possível criar utilizadores, músicas, álbuns e playlists. Note-se que esta informação deverá ser estruturalmente coerente. Uma hipótese é esta informação ser carregada de um ficheiro, dispensando quem utiliza o programa de fazer o setup da informação sempre que tal seja necessário.
2. deverá ser possível adicionar conteúdos ao sistema **SpotifUM**.
3. deverá ser possível criar utilizadores e associá-los a um plano de subscrição dos existentes.
4. deverá ser possível reproduzir conteúdos, tanto individuais (músicas) e playlists, bem como pedir ao sistema que gere conteúdos na forma de playlists.

3 Requisitos

O trabalho proposto tem vários níveis de requisitos, desde os mais básicos até aos mais complexos. Para que os vários níveis de requisitos identificados sejam considerados cumpridos o programa deve respeitar os princípios da programação por objectos, ser robusto no tratamento de erros, bem como gravar, e também carregar, o estado do programa para um ficheiro de objectos. Deverá possuir uma camada de interacção com o utilizador, que se aconselha ser feita por menus em modo texto (note que **não** é solicitada a existência de camada gráfica) e que permita aceder às funcionalidades do programa. Durante a sessão de apresentação será também avaliada a forma como o programa está construído, nomeadamente as classes existentes e como é que estão organizadas numa arquitectura que permita responder ao que é pedido, e a utilização das estruturas de dados e a forma como são utilizadas e manipuladas.

3.1 Requisitos base de gestão das entidades

Como requisitos base pretende-se ter a capacidade de criar utilizadores, músicas, planos de subscrição e playlists e registar a reprodução de um conteúdo por parte de um utilizador, bem assim como efectuar a salvaguarda e a leitura do estado a partir de um ficheiro.

Nota máxima: 12 valores

Tenham em atenção que, mesmo que se permitam criar utilizadores e músicas, se não for possível registar a reprodução de uma música por parte de um utilizador a nota máxima será de 10 valores.

3.2 Efectuar estatísticas sobre o estado do programa

Além do anterior, o programa deverá ser capaz de actualizar a informação do sistema e ainda permitir saber:

1. qual é a música mais reproduzida

2. qual é o intérprete mais escutado
3. qual o utilizador que mais músicas ouviu num período ou desde sempre
4. qual é o utilizador com mais pontos
5. qual o tipo de música mais reproduzida
6. quantas playlists públicas é que existem
7. qual o utilizador que tem mais playlists

Nota máxima: 16 valores

3.3 Criar a noção de música *MusicaExplicita*

Além dos requisitos anteriores, o programa deverá permitir a existência de músicas que sejam *MusicaExplicita* por forma a que seja possível classificar músicas deste tipo novas músicas deste tipo.

Nota máxima: 18 valores

3.4 Gerar uma playlist de acordo com critérios

Tendo em conta as características dos utilizadores o programa poderá gerar playlists para um determinado utilizador permitindo que se obtenha:

- uma playlist com as preferências musicais do utilizador
- uma playlist com as preferências musicais do utilizador e restrita a um determinado tempo máximo
- uma playlist com as preferências musicais do utilizador mas apenas com músicas do tipo *MusicaExplicita*

Nota máxima: 20 valores

Obviamente que a nota a atribuir, em cada um dos escalões, irá reflectir a estruturação da solução que deverá respeitar as normas da programação orientada aos objectos, cf aulas teóricas, nomeadamente o encapsulamento, a abstração de implementação e a capacidade de a aplicação evoluir de forma controlada.

Notem que **não serão considerados como merecedores de nota positiva trabalhos que não permitam a criação e visualização da informação relativa às entidades do programa.**

4 Relatório

O relatório deve descrever o trabalho realizado para desenvolver a aplicação solicitada. No mínimo, devem ser abordados os seguintes pontos:

- Capa com identificação da Unidade Curricular e do grupo (nome, número e fotografia).
- Descrição da arquitectura de classes utilizada (classes, atributos, etc.) e das decisões que foram tomadas na sua definição. Deverá ser entregue um **Diagrama de Classes** com a arquitectura de classes que suporta o programa desenvolvido.
- Descrição da aplicação desenvolvida (ilustração das funcionalidades).

Todos estes itens são obrigatórios!

A cotação do relatório é de 2 valores (e está incluída nos patamares anteriormente identificados).

5 Salvaguarda do estado da aplicação

O programa deve permitir que em qualquer momento se possa guardar em ficheiro a informação existente em memória sobre a informação relevante das entidades. A gravação deve ser feita de forma a permitir que o estado que foi gravado seja recuperado novamente. Na altura da entrega do projecto deve ser também entregue um estado (guardado em ficheiro) que possa ser carregado durante a apresentação.

6 Cronograma

1. Entrega final de código, ficheiro com os dados a carregar durante a apresentação e relatório de projecto (feita por via electrónica no git)

Data Entrega: *17 de maio de 2025*

2. Apresentação do projecto

Semana de: *2 a 5 de junho de 2025 - data acordada com a direcção de curso*