

Programação III

Semestre de Inverno de 2021-2022





(Data de entrega 13 de fevereiro 2022)

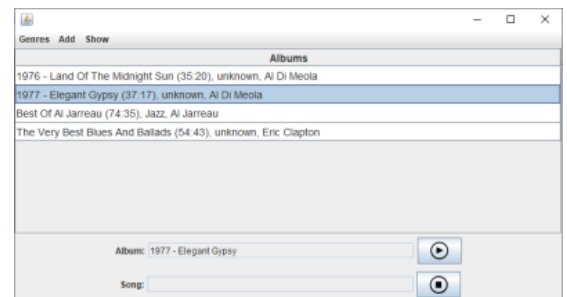
3º Trabalho prático

Em anexo a este enunciado são entregues as fontes de uma implementação parcial da aplicação *musics*, que terá de ser completada.

A aplicação suporta o carregamento de músicas presentes em ficheiros mp3, mantendo uma base de dados com informação das músicas carregadas, de modo a permitir a sua reprodução posterior. As músicas podem ser faixas isoladas, conjuntos de músicas presentes numa dada pasta (álbum) ou *playlists*. As *playlists* são seleções de músicas construídas a partir dos ficheiros previamente carregados. Em cada música é mantida informação adicional presente nos ficheiros mp3, nomeadamente o género musical, o artista e a sua duração em segundos. As faixas isoladas por vezes têm o nome do álbum onde estão contidas. Os álbuns e as *playlists* contêm o respetivo conjunto de faixas.

A aplicação fornece as seguintes funcionalidades, parcialmente implementadas:

- Carregamento de ficheiros mp3 existentes numa pasta ou hierarquia de pastas.
- Listagem da informação associada a músicas isoladas e álbuns previamente carregados.
- Listagem dos géneros musica e dos artistas associados aos ficheiros carregados.
- Reprodução de faixas isoladas, álbuns e playlists.
- Filtragem das músicas a visualizar/reproduzir por artista ou géneros musicais.
- Construção de playlists geradas aleatoriamente, a partir das músicas presentes na base de dados. Na construção duma *playlist* é possível indicar os géneros musicais, o número máximo de músicas e a duração limite.
- No visualizador é possível, dada uma lista de itens de música (faixas, álbuns ou *playlists*), selecionar uma determinada entrada. A seleção é realizada fazendo *click* no botão esquerdo do rato sobre o item a selecionar (como mostra a figura).
- Caso a lista visualizada seja de álbuns ou de *playlists* é possível fazer *zoom*, isto é, passar a visualizar a lista de faixas do álbum ou da *playlist* selecionada, fazendo um *click* com o botão direito no item selecionado (álbum ou *playlist*).
- Quando é feito um *click* no botão  o item musical selecionado no visualizador começa a tocar e a face do botão passa a ser . Caso agora se faça um *click* no botão é feita uma pausa na música e a face volta a ser . A música continua a reprodução quando for novamente premido o botão.
- Quando é feito um *click* no botão  a reprodução termina e as caixas de texto correspondentes ao álbum e à faixa ficam vazias.



A seguir indica-se de forma mais detalhada o que falta implementar:




1. Completar a hierarquia de **MusicItem**

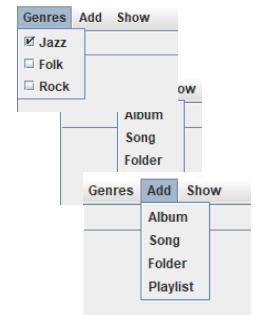
- Classe **Song**, recebe por parâmetro no construtor o título, o *pathname* completo da faixa, os artistas, os géneros, a duração e o nome do álbum. O método **playOn** invoca o método **play** sobre o **player** passando por parâmetro o *pathname*.
- Classe **Album** é uma música iterável de faixas musicais. Recebe por parâmetro no construtor o título, o *pathname* completo, as faixas, os artistas e os géneros que constam no álbum. A duração é o somatório das

durações. O método **playOn** invoca o método **play** sobre **player** passando por parâmetro uma lista de *pathnames* das faixas do álbum.

- Classe **Playlist** é uma sequência iterável de faixas musicais. No construtor só recebe por parâmetro o título. As faixas vão sendo adicionadas.
- Classe **MusicDB** implementa a base de dados e tem métodos para a adição de faixas e álbuns, para a construção de playlists e para a obtenção de coleções selecionadas de músicas (por exemplo: todos os álbuns, faixas musicais de determinado género, ou artista, etc.). Devem ser criadas várias estruturas de dados para responder de forma eficiente aos vários tipos de seleção.

2. Completar a classe **Mp3PlayerFrame** tendo em conta.

- Botão *start/pause* - inicialmente o botão tem na face , após ser premido fica com . Quando é premido novamente o botão fica novamente com  na face.
- Acrescentar ao *menu bar* os menus:
 - ◆ Item “*Genres*” – Selecionar géneros. Os géneros selecionados permitem selecionar determinados tipos de música. Usar uma instância de **MenuItemCheckList**.
 - ◆ Item “*Add*” – Para adicionar faixas, álbuns, diretórios e playlists. Para a construção das músicas usar os *getters* da classe **MusicItemFinder**.
 - ◆ Item “*Show*” – Para visualizar determinadas músicas. Para listar usar o método **setItems** do **ItemsViewer**.



Para a conceção deste trabalho são fornecidas as classes:

- **MusicItem** - classe base de qualquer item de música.
- **Mp3Player** - contém os métodos que permitem reproduzir uma música ou uma sequência de músicas:

```
play(String path)
play(Iterable<String> paths)
resume()
stop()
pause()
```

Permite registar *listeners* para os eventos: faixa começou a reprodução; terminou a reprodução das faixas; erro de reprodução.

```
setStartMusicListener(StartMusicEventListener listener)
setCompletedListListener(CompletedListEventListener listener)
setErrorListener(ErrorEventListener listener)
```

- **MusicItemFinder** que disponibiliza métodos para obter: uma faixa dado o nome do ficheiro; um álbum dada a diretoria; todas as músicas (faixas e álbuns) que estejam contidas numa diretoria e suas subdiretorias.

```
Song getSong(File filename)
Album getAlbum(File dirname)
List<MusicItem> getMusics(File rootFolder)
```

- **ItemsViewer** é um visualizador de itens que dispara um evento de seleção quando um item é selecionado ou um evento de *zoom* quando sobre o evento selecionado for premido o botão direito do rato, disponibiliza os seguintes métodos:

```
setItems( String title, Iterable<?> items,
         SelectedItemListener selectedListener,
         ZoomItemListener zoomListener)
public <E> E getSelectedItem()
```

- **MenuItemCheckList**, permite construir um menu cujos itens são **CheckBox**, disponibiliza os métodos :

```
MenuItemCheckList addItem(String text)
String[] getSelected()
```

Entrega do trabalho

O relatório deve conter:

- A descrição de como interagem os componentes principais da aplicação (**Mp3PlayerFrame**, **MusicDB**, **Mp3Player** e **MusicItemFinder**).
- A descrição de como funcionam os *listeners* (de **ItemsViewer** e **Mp3Player**). Na descrição deve constar: quando são adicionados e a quem são adicionados; quando são chamados e quem os chama.
- Para cada um dos elementos adicionados ou modificados deve ser explicado:
 - O local onde foram colocados na hierarquia de classes fornecida.
 - Os membros que foram acrescentados.
 - Os métodos mais relevantes.
- Uma descrição das estruturas de dados implementadas para armazenar as músicas. Na descrição deve constar:
 - Os dados.
 - Os métodos mais relevantes.
- Diagramas estáticos de classes UML (incluindo as que já estão implementadas).
- Um capítulo onde se evidencie quais dos temas estudados ao longo do semestre foram aplicados e quais as vantagens resultantes da sua aplicação na implementação da base de dados. Em particular, os seguintes temas: herança e polimorfismo; estruturas de dados; interfaces funcionais; *streams*; e programação *event driven*.
- Na conceção das soluções valoriza-se:
 - A qualidade do código produzido (facilidade de alteração; não repetição de código; modularização; documentação);
 - As estruturas de dados e os algoritmos utilizados;
 - A facilidade de utilização da aplicação;
 - Criatividade.

Bom trabalho

Manuela Sousa,
Jorge Martins