

Relatório de trabalho prático

Trabalho Prático LP2

Rui Costa, helder sá

17005, 17023

Trabalho realizado sob a orientação de:

Luís Ferreira

**Linguagens de Programação II**

**Licenciatura em Engenharia de Sistemas Informáticos**

Barcelos, Abril de 2020

Índice

1 Classes 1

1.1 Classes Implementadas 1

1.1.1 Objetivo das classes 1

1.1.2 Diagrama de classes 2

2 Collections 4

3 Métodos 7

4 <Conclusão ... > 11

Bibliografia 13

Anexos 14

Lista de Figuras

Figura 1: Diagrama de Classes 2

# Classes

## Classes implementadas

Para resolver este problema decidimos implementar 3 principais classes, a classe “Artistas”, a classe “Albuns” e a Classe “Musicas”.

### Objetivo das classes

A classe Artistas será composta por um array de Albuns, e por ultimo a classe Albuns será composta por um array de albuns, assim, as músicas tem obrigatoriamente de estar ligadas a um album que por sua vez tem de estar ligado a um artista, artista esse que tem de fazer parte da editora.

### Diagrama de Classes

Uma imagem com captura de ecrã

Descrição gerada automaticamente

**2 – Collections**

A estrutura de dados que escolhemos para a implementação deste problema foram as Listas, pertencentes à biblioteca “System.Collections.Generic” pois foi a que achamos mais simples e a que melhor se adequava à nossa proposta de resolução. Assim no programa existem 3 listas, uma de artistas, uma de álbuns e uma de musicas.

**3 – Métodos**Em todas as classes existem 4 principais métodos, um que serve para descobrir a posição de cada objeto na lista, um para saber se determinado objeto existe na lista, um para adicionar objetos à respetiva lista e outro para os remover da lista.  
Existe depois uns métodos extra, nomeadamente um que associa álbuns a artistas (na classe artista) e musicas aos álbuns (na classe álbuns).  
  
Imagem 1: Devolve a posição do objeto na lista (Também existe nas classes álbuns e musicas, chamando-se “ObterAlbumIndex” e “ObterMusicaIndex” respetivamente).  
Uma imagem com sentado, portátil, ecrã

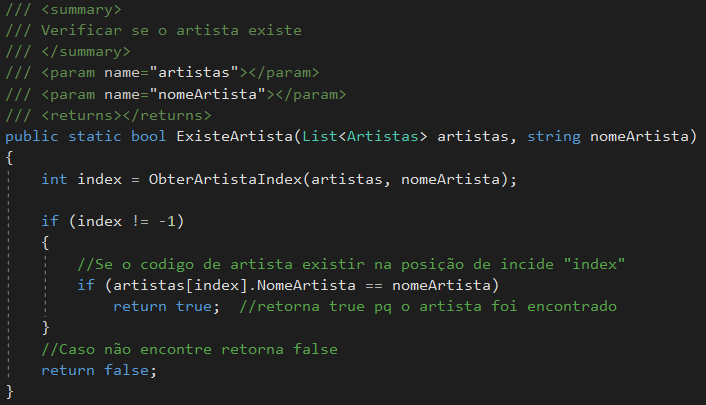
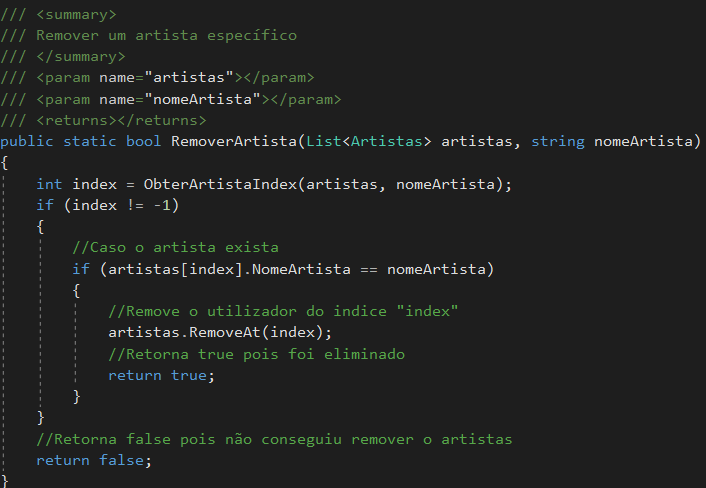
Descrição gerada automaticamente  
Imagem 2: Devolve um booleano, true se existir o artista, false se não existir (também existe nas classes álbuns e musicas, chamando-se “ExisteAlbum” e “ExisteMusica” repetivamente).  


Imagem 3: Adiciona um artista à lista caso ainda não exista nenhum com o nome lido (também existe nas classes álbuns e musicas, chamando-se “RegistarAlbum” e “RegistarMusica” respetivamente).  
Uma imagem com captura de ecrã

Descrição gerada automaticamente  
  
  
Imagem 4:Remove um artista da lista, caso exista um artista com o nome lido (também existe nas classes álbuns e musicas, chamando-se “RemoverAlbum” e “RemoverMusica”, respetivamente).  
  
Imagem 5: Permite atribuir um álbum anteriormente criado a um artista também anteriormente criado (na classe artistas) e uma musica anteriormente criada a um álbum também anteriormente criado (na classe álbuns chama-se “AtribuirMusica”).  
Uma imagem com captura de ecrã

Descrição gerada automaticamente

# <Conclusão ... >

<A formatação deve ser consistente ao longo de todo o documento. Uma das formas de manter a formatação consistente é recorrer a estilos para formatar os principais formatos utilizados.>

# Bibliografia

<Não esqueça de incluir a bibliografia neste ponto, a qual deverá seguir o formato APA.

O título da bibliografia e dos anexos é formatado de forma semelhante ao capítulos de texto, mas sem numeração. Deve ser inserida uma quebra de secção “página ímpar” antes do título, tal como se apresenta no modelo.>

# Anexos

<Incluir aqui os anexos necessários>