

**Relatório**

**Linguagens de Programação II**

**Alunos:  
Rui Costa – 17005   
Helder Sá – 17023**

**Professor: Luís Ferreira**

**Licenciatura em Engenharia de Sistemas Informáticos**

Barcelos, outubro, 2018

Resumo

Este trabalho consistia na implementação em C# de um projeto de tema de certa forma livre, assim, a nossa proposta baseou-se no trabalho prático de recurso de LP2. Assim, decidimos implementar um software de gestão para uma editora, editora esta que trabalha com artistas, álbuns e com músicas. Assim nesta proposta é possível adicionar, remover e editar álbuns, artistas e músicas. Associar álbuns a artistas e músicas a álbuns. Listar todos os artistas, listar todos os álbuns (ou listar por estilo ou por artista), listar todas as músicas (ou listar por álbum). Disponibiliza também uma Dashboard onde são mostrados o álbum mais e menos vendido, a quantidade total de artistas associados à editora e a quantidade total de músicas.  
Será de salientar também que este projeto respeita a arquitetura NTier, assim como usa o tipo de ficheiros Json para guardar todos os dados relativos aos intervenientes.

**Palavras-Chave:  
C#  
NTier  
Collections  
POO**

Índice de Figuras

Figura 1 - Diagrama de Classes 7

Figura 2 - ObterAlbumIndex…………………………………………………………………………………………………………..8

Figura 3 - ExisteArtista…………………………………………………………………………………………………………………..9

Figura 4 - RegistaArtista…………………………………………………………………………………………………………………9

Figura 5 - RemoverArtista…………………………………………………………………………………………………………….10

Figura 6 - AtribuirAlbum………………………………………………………………………………………………………………10

Índice

[1. Introdução 7](#_Toc22296270)

[1.1. Contextualização 7](#_Toc22296271)

[1.2. Motivação e objetivos 7](#_Toc22296272)

[1.3. Estrutura do Documento 7](#_Toc22296273)

[2. Estado da arte 8](#_Toc22296274)

[3. Implementação 9](#_Toc22296275)

[3.1. Descrição do problema 9](#_Toc22296276)

[3.2. Solução 9](#_Toc22296277)

[4. Análise e Testes 10](#_Toc22296278)

[5. Conclusão 11](#_Toc22296279)

[5.1. Lições aprendidas 11](#_Toc22296280)

[5.2. Apreciação final 11](#_Toc22296281)

# Introdução Este trabalho foi realizado no âmbito da Unidade Curricular de Linguagens de Programação II, lecionada pelo docente Luís Ferreira, e focado na linguagem de programação orientada a objetos C#.

## Contextualização

## O tema deste trabalho era livre, e a nossa escolha foi reimplementar o trabalho de recurso de Linguagens de Programação I e desenvolver um software de gestão de uma editora/produtora musical.

## Motivação e objetivos

## O nosso principal objetivo com a realização deste trabalho é sem dúvida perceber melhor como funcionam as linguagens de programação orientadas a objetos, e neste caso, mais concretamente o C#.

## Estrutura do Documento O projeto está estruturado em 4 camadas, de entre elas a Business Objets (BO), a Data Layer (DL), a Business Rules (BL) e a camada de interação que neste projeto tem o nome de Principal. Em cada uma das camadas, e para mais fácil organização, dividimos os métodos por classes, cada camada tem uma classe relacionada com os artistas, uma classe relacionada com os álbuns e uma classe relacionada com as musicas. Havendo um extra na Business Layer que foi onde decidimos implementar os métodos relativos aos Json Files.

# Diagrama de Classes

# Uma imagem com captura de ecrã Descrição gerada automaticamente

# Implementação

## Descrição do problema

O problema pedia que conseguíssemos desenvolver um software capaz de ajudar a gerir uma editora/produtora musical. Assim teria de ser possível adicionar, remover e editar artistas, musicas e álbuns. Teria também de ser possível associar álbuns aos artistas e associar musicas a álbuns. Dado isto era também pedido para listar artistas, álbuns e musicas, com as particularidades de que os álbuns teriam de ser possíveis listar por artista e por estilo, e as musicas teriam de ser possíveis listar por álbum. Era também pedida uma dashboard com informação relativa a qual o álbum mais e menos vendido, e qual o total de artistas e de musicas associados à editora em questão.

## Solução

Inicialmente começamos a desenvolver o projeto sem usar nenhum tipo de arquitetura em especial, mas desde o inicio que as listas foram as nossas collections de eleição. Assim os primeiros métodos implementados foram  
  
Figura 2 – ObterAlbumIndex – Este método tem como objetivo devolver a posição do objeto na lista de álbuns, foi também implementado usando o mesmo algoritmo, para a lista de artistas e de musicas.  
Uma imagem com sentado, portátil, ecrã

Descrição gerada automaticamente

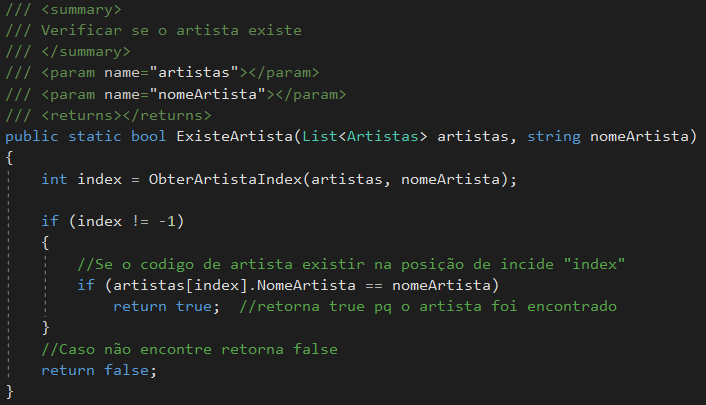
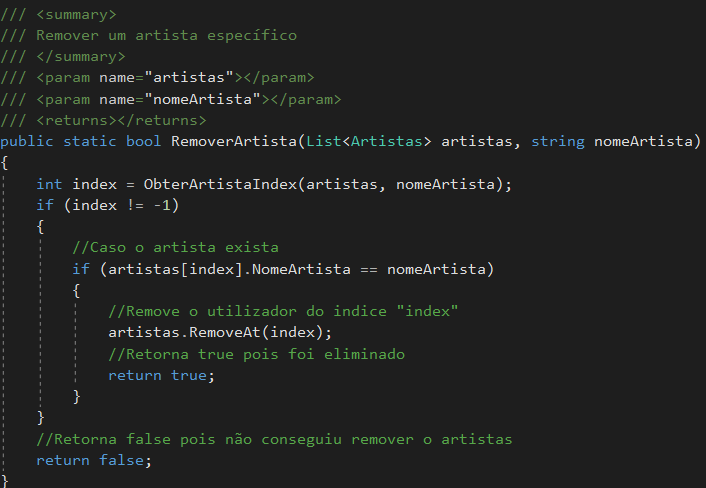
Figura 3- ExisteArtista – Este método devolve um booleano, true caso o artista em questão exista e false caso não exista, foi também implementado, usando o mesmo algoritmos, para verificar a existência de álbuns e de musicas.  


Figura 4 - RegistarArtista – Este método permite adicionar um artista à lista de artistas, foi também implementado, usando o mesmo algoritmo para adicionar álbuns à lista de álbuns e musicas à lista de musicas.  
Uma imagem com captura de ecrã

Descrição gerada automaticamente

Figura 5 – RemoverArtista – Este método tem como objetivo eliminar um artista da lista de artistas, tendo sido também implementado, usando o mesmo algoritmo base para remover álbuns e musicas, com a particularidade de que ao remover álbuns não basta remover da lista de álbuns, mas também tem de remover do artista a que ele está associado, caso esteja, e as musicas devem ser removidas também dos álbuns a que estão associadas.  
  
  
Figura 6 – AtribuirAlbum – Este método permite associar um álbum previamente criado a um artista também já existente, passando este álbum a fazer parte de uma lista de álbuns que o artista eventualmente terá. Foi também implementado, usando o mesmo algoritmo para ser possível atribuir musicas a álbuns.  
Uma imagem com captura de ecrã

Descrição gerada automaticamente

* + 1. **Restantes Métodos**Praticamente todos os outros métodos deste programa foram criados fazendo referencia a estes principais, logo, para não nos alongarmos muito neste relatório decidimos apenas apresentar esses pois são sem duvida a base de todo o projeto.

# Análise e Testes Fomos efetuando testes à medida que fomos implementando novas funcionalidades, para ir percebendo se estas estavam a funcionar corretamente ou não, os testes foram simples, pois consistiam na execução do programa e tentar efetuar determinada tarefa, para conseguirmos perceber se esta estava funcional ou se precisava de ser retificada, e caso precisasse de ser retificada, o porquê.

## 

# Conclusão

## Lições aprendidas Com a conclusão deste trabalho ficamos a perceber muito melhor o conceito das POO’s assim como a utilidade da arquitetura NTier durante o desenvolvimento do projeto, principalmente por uma questão de organização.

## Apreciação final Concluido o projeto, e tendo efetuado vários testes, acreditamos que o programa tenha capacidade de efetuar todas as funções pedidas no enunciado, podendo eventualmente apresentar um bug ou outro, o que poderia ser resolvido caso tivéssemos mais tempo para a conclusão. Assim, temos uma apreciação final muito positiva e estamos estamos muito satisfeitos com a proposta apresentada.

Bibliografia

https://docs.microsoft.com/en-us/dotnet/api/system.web.script.serialization.javascriptserializer?view=netframework-4.8