



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
**COIMBRA**

Gonçalo Vasconcelos Correia, no 2019216331

## **Princípios de Programação Procedimental**

Janeiro 2020

## ÍNDICE

1. Introdução
2. Descrição das classes / métodos mais relevantes
3. Manual de utilizador
4. Diagrama de classes
5. Conclusão

## INTRODUÇÃO

No âmbito da disciplina de POO foi proposta a realização de um projeto. Este tem como dados de entradas três ficheiros: um com que contém os grupos de investigação da CISUC, outro que contém os investigadores e por último um que contém as publicações.

O objetivo deste projeto abrangia várias tópicos. Nomeadamente, listar dados em relação ao número de investigadores (totais/de cada tipo) que constituem os grupos, listar as publicações dos últimos cinco anos agrupadas por fator impacto e data de publicação, listar os investigadores ,agrupados por tipo, de cada grupo de investigação e listar as publicações de um investigador agrupadas por fator impacto e data de publicação. Para além disso, foi também requisitado um gráfico UML .

Neste projeto também é usada a interação com o utilizador através de uma interface de consola , por exemplo, quando lhe é solicitado o grupo de investigação(acrónimo) ou investigador que este pretende selecionar para ver as publicações correspondentes.

## DESCRIÇÃO DAS CLASSES / MÉTODOS MAIS RELEVANTES

Para a realização deste projeto usei doze classes. Uma classe para o menu(**MainCISUC**, onde o utilizador pode escolher a opção que pretende), outra que contém os métodos que gerem o programa em geral (**Gere**). Uma classe **GruposInv** composta pelos seus atributos pré-definidos no enunciado e por um *ArrayList* que contém as publicações desse grupo. Uma classe **Investigador** composta pelos seus atributos pré-definidos no enunciado, por um *ArrayList* das publicações do investigador em causa e o atributo que vai conter o tipo de investigador em causa. Uma classe **Estudante** e uma classe **MembroEfetivo** que são filhas da classe **Investigador** que contém os atributos pré definidos no enunciado. Uma classe **Publicacao** que contém os atributos pré-definidos do enunciado, um atributo que vai conter o tipo de publicação em causa, um método *comparaTo* que vai ordenar as publicações de acordo com o fator impacto(dimensão da audiência) e um método que vai calcular o fator impacto de cada publicação de acordo com o seu tipo(uso do polimorfismo). Uma classe **AritgoConferencia**, uma classe **ArtigoRevista**, uma classe **Livro** que são filhas da classe **Publicacao** contém os atributos pré-definidos do enunciado e a função que calcula o fator impacto correspondente. Por último temos as classes **Capitulo** e **LivroArtigo** que são filhas da classe **Livro** contém os atributos pré-definidos do enunciado e a função que calcula o fator impacto correspondente.

Na classe **Gere** temos como métodos mais relevantes temos o método *readFiles()* que é responsável pela leitura dos ficheiros texto/objetos , o método *publicacaoGrupo()* que é usado após a leitura dos ficheiros de texto apenas. Este é responsável por armazenar as publicações no *ArrayList* de publicações dos investigadores, armazenar os investigadores no *ArrayList* de investigadores das publicações e armazenar as publicações no *ArrayList* de publicações dos grupos de investigação. Paralelamente temos o método *getInv(String nomeAutor)* que dada a *string* com o nome do investigador retorna o investigador que lhe corresponde, o método *grupoInv(String grupo)* que dada a *string* com o nome do grupo de investigação retorna o grupo de investigação que lhe corresponde e o método *getGInv(Investigador inv)* que dado o investigador retorna o grupo de investigação em que te está inserido.

Por último o método que contém a informação para o ponto 1) do enunciado é *indicadoresGeraisCisuc()*, o método que contém a informação para o ponto 2) do enunciado é *listaPubGrupo()*, o método que contém a informação para o ponto 3) do enunciado é *listarMembrosGrupo()*, o método que contém a informação para o ponto 4) do enunciado é *pubInv()* e o método que contém a informação para o ponto 5) do enunciado é *listarTodosGruposInv()*.

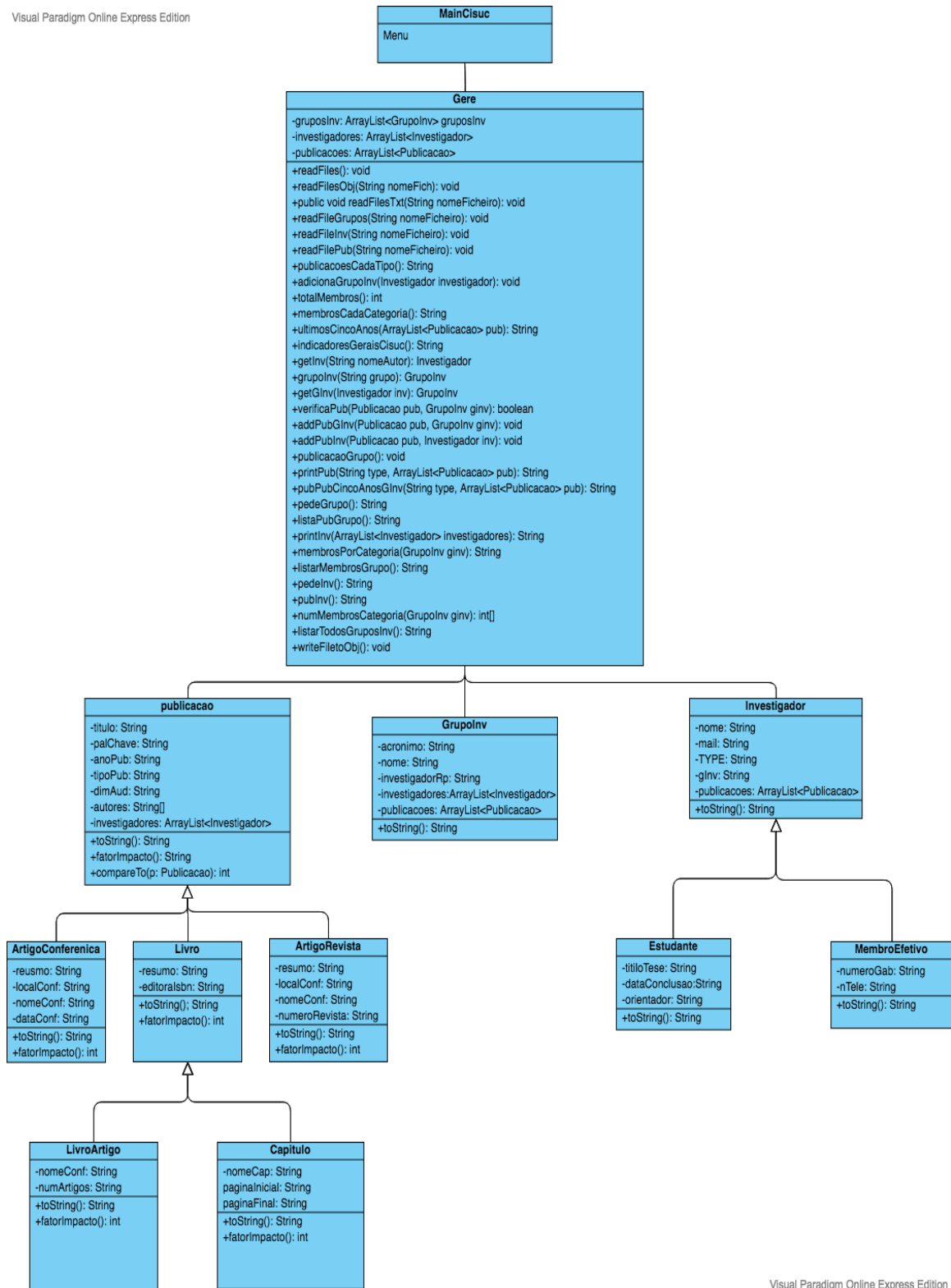
## **MANUAL DE UTILIZADOR**

De modo a correr o programa é bastante simples. É necessário assegurar que os ficheiros de entrada estão na mesma diretoria do projeto. De seguida basta correr o programa onde será apresentado um menu.

O menu terá seis opções. Para escolher a opção que pretende apenas é preciso digitar o número correspondente à opção que quer.

# DIAGRAMA DE CLASSES

Visual Paradigm Online Express Edition



Visual Paradigm Online Express Edition

## CONCLUSÃO

Concluindo, penso que há algumas melhorias que poderiam ser feitas. Nomeadamente, talvez haja um excesso de funções auxiliares das quais algumas poderiam ser suprimidas e incorporadas noutras. Por outro lado é possível que haja repetição de código em relação à componente dos “últimos cinco anos”. Ao longo do projeto fui mudando por vezes o raciocínio. Por exemplo, inicialmente tinha um contador para cada tipo de publicação e cada tipo de investigador. No entanto eu incrementava o contador quando lia a partir dos ficheiros de texto, não funcionando no caso de ler a partir de ficheiros de objetos.

Em suma, considero que o projeto em grande parte cumpre os requisitos sabendo que podem haver algumas falhas em relação a algum reaproveitamento de código e de algum raciocínio que possa ter complicado algo que poderia ser mais feito de uma forma mais simples.