

2º Ano – Licenciatura em Engenharia Informática

Programação Orientada aos Objetos

Gestor de Publicações do CISUC

Relatório



Miguel António Gabriel de Almeida Faria | 2019216809

Índice

Introdução	3
Descrição de Classes, Métodos mais relevantes e Funcionamento do Programa	4
Manual de Utilizador	5
Diagrama de Classes	6
Conclusão	7
Referências	8

Introdução

Este trabalho foi realizado no âmbito da disciplina de POO (Programação Orientada aos Objetos).

Pretende-se que se desenvolva uma aplicação “Gestor de publicações do CISUC” que permita gerir os grupos de investigação, investigadores e publicações. Além disso, o principal objetivo é poder listar as publicações do CISUC de acordo com vários critérios de pesquisa de modo a dar resposta a todos os pedidos da equipa externa de avaliação.

Ao realizar este projeto, procurei seguir as convenções da programação em *Java*, assim como as recomendações dadas pela professora, nomeadamente a forma de escrita das variáveis, métodos e classes, o uso de ficheiros diferentes para classes diferentes, entre outros.

Descrição de Classes, Métodos mais relevantes e Funcionamento do Programa

O Projeto possui 14 classes (a indexação representa a hierarquia entre as classes):

- GestorDePublicacoesDoCISUC - Classe que contém o método 'main' onde é criada a aplicação.
- LeituraEscritaFicheiros - Classe que contém os métodos para a leitura e escrita de ficheiros.
- InterfaceDeConsola - Classe que contém os métodos para a interface de consola.
- Investigador - Classe abstrata de objetos de Investigadores.
 - Estudante - Classe que permite criar objetos de Estudantes.
 - MembroEfetivo - Classe que permite criar objetos de Membros Efetivos.
- GrupoDeInvestigacao - Classe que permite criar objetos de Grupos de Investigação.
- Publicacao - Classe abstrata de objetos de Publicações.
 - ArtigoDeConferencia - Classe que permite criar objetos de Artigos de Conferência.
 - ArtigoDeRevista - Classe que permite criar objetos de Artigos de Revista.
 - Livro - Classe que permite criar objetos de Livros.
 - CapitulosDeLivro - Classe que permite criar objetos de Capítulos de Livros.
 - LivroDeArtigosConferencia - Classe que contém os métodos para a leitura e escrita de ficheiros.
- Data - Classe que permite criar objetos de Datas.

Alguns dos métodos mais relevantes são:

- Métodos 'ficheiroInvestigadores', 'ficheirosGrupoDeInvestigacao' e 'ficheirosPublicacoes' (presentes na classe 'LeituraEscritaFicheiros') - permitem ler os ficheiros de objetos, ou caso não existam, ler os ficheiros de texto, criar os *ArrayLists* de objetos e escrevê-los em ficheiros de objetos.
- Métodos 'indicadoresGeraisCISUC', 'publicacoesGrupoUlt5AnosOrganizadas', 'membrosGrupoInvestigacaoAgrupados', 'publicacoesInvestigadorAgrupadas' e 'dadosGruposInvestigacao' (presentes na classe 'InterfaceDeConsola') - permitem produzir as listagens e indicadores necessários propostos no enunciado.

O programa começa por criar *ArrayLists* onde vão ficar os objetos 'Investigador', 'GrupoDeInvestigacao' e 'Publicacao'. De seguida são lidos os ficheiros de objetos e colocado o seu conteúdo nos *ArrayLists* respetivos; caso esses ficheiros não existam, são lidos os ficheiros de texto, criados os *ArrayLists* e estes são escritos num ficheiro de objetos, para a próxima vez que o programa for executado. Posteriormente, é apresentado um menu na consola, com as diferentes opções que permitem realizar as listagens propostas.

Manual de Utilizador

O projeto possui 14 Ficheiros “.java”, correspondentes às classes usadas no projeto, e 3 ficheiros “.txt”, contendo informações acerca dos investigadores (“Investigadore.txt”), acerca dos grupos de investigação (“GrupoDeInvestigacao”) e acerca das publicações (“Publicacoes.txt”). Os ficheiros “.java” existentes na pasta “ficheirosJava” devem ser colocados na pasta “src” do projeto e os ficheiros “.txt” existentes na pasta “ficheirosTexto” na pasta do projeto.

Relativamente ao *Javadoc*, deve ser aberto o ficheiro “index.html” presente na pasta “javadoc” de modo a consultar o sumário das classes.

Para executar o ficheiro “Projeto.jar” existente na pasta “executavel”, é necessário colocar esse ficheiro e os ficheiros de texto numa mesma pasta, abrir um terminal, colocar a diretoria dessa pasta e colocar o comando “java -jar Projeto.jar”. Em alternativa, pode-se executar o ficheiro “Projeto.bat”.

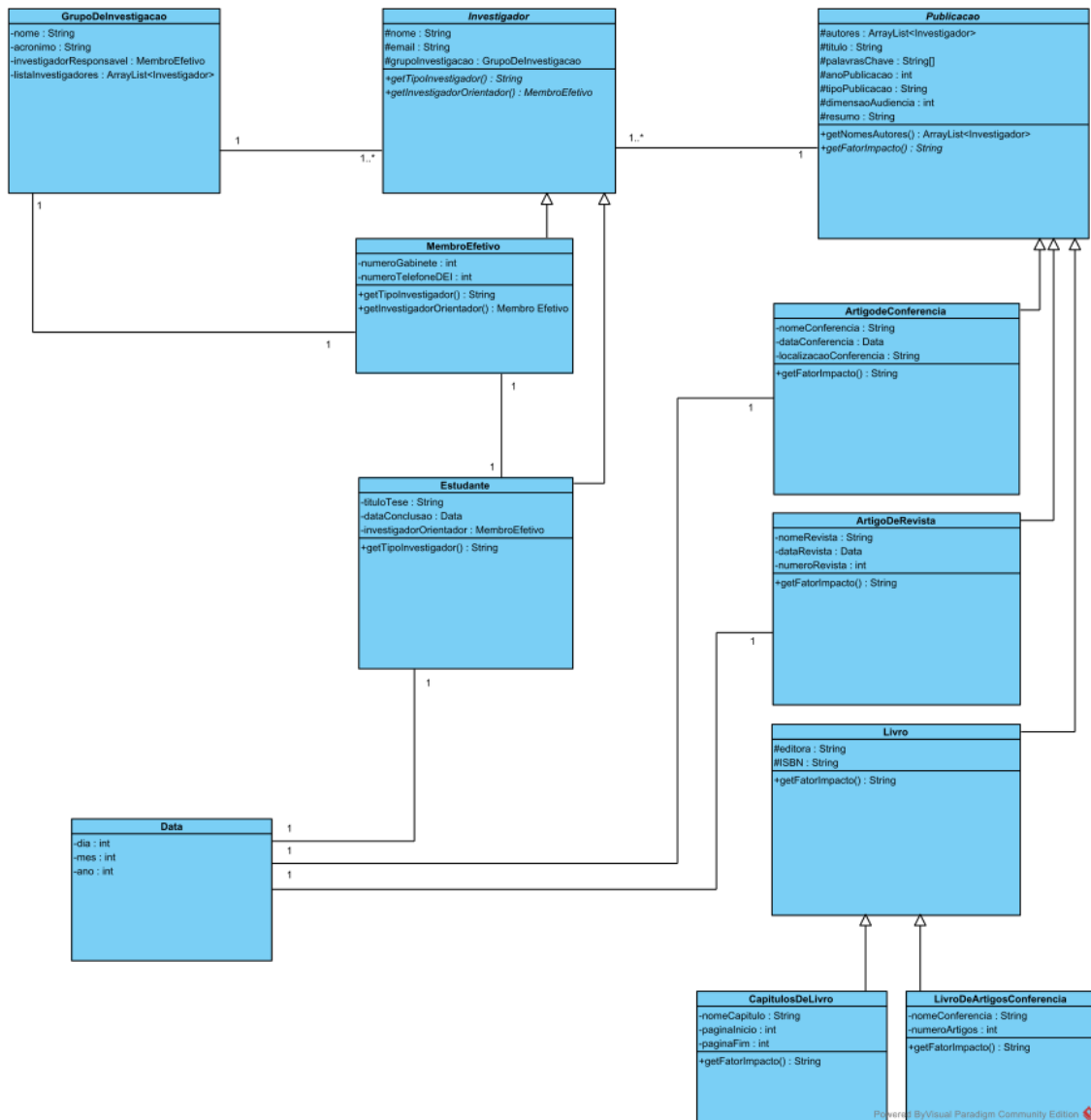
Relativamente ao funcionamento do programa, ao executá-lo, caso os ficheiros de objetos já existam, é apresentado o menu; caso contrário, são apresentadas mensagens para controlar a escrita dos ficheiros de objetos e só depois o menu.

No menu, estão as diferentes opções que o utilizador pode escolher:

- caso escolha a opção “1”, é apresentado um segundo menu em que o utilizador deve escolher novamente a opção que pretende.
- caso escolha a opção “2” e “3”, pede-se ao utilizador que introduza o nome ou acrónimo do grupo que pretende.
- caso escolha a opção “4”, pede-se ao utilizador que introduza o email do investigador que pretende.
- caso escolha a opção “5”, é apresentado logo a listagem.

Quando o utilizador pretender sair do programa, basta escolher a opção “6” no menu inicial.

Diagrama de Classes



Observações:

- As classes “GestorDePublicacoesDoCISUC”, “LeituraEscritaFicheiros” e “InterfaceDeConsola” não estão representadas no UML, uma vez que apenas possuem métodos para a resolução dos problemas separados de modo a uma mais fácil compreensão.
- Os construtores, “getters” e “setters” dos atributos e os “toStrings” das classes estão omitidos, para mais fácil visualização do diagrama.

Conclusão

Com a realização deste trabalho aprofundei os meus conhecimentos relativamente à programação orientada aos objetos e à linguagem Java.

Penso que esta aplicação, apesar de ser uma versão bastante simplificada de um gestor de publicações do CISUC, poderia ser aplicado numa situação real.

Referências

PowerPoints fornecidos pela professora

<https://www.oracle.com/technical-resources/articles/java/javadoctool.html>

<https://www.jetbrains.com/help/idea/working-with-code-documentation.html#generate-javadoc>

<https://blog.jetbrains.com/idea/2010/08/quickly-create-jar-artifact/>