



C • FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

APLICAÇÃO PARA GESTÃO DE CENTROS DE INVESTIGAÇÃO

Relatório do Projeto

Relatório realizado no âmbito da unidade curricular Programação Orientada aos Objetos,
inserida no programa curricular do curso de Engenharia Informática da Faculdade de
Ciências e Tecnologia da Universidade de Coimbra

Bruno Ricardo Leitão Faria, 2018295474
Dylan Gonçalves Perdigão, 2018233092

Coimbra
2019

ÍNDICE

INTRODUÇÃO	3
CÓDIGO PRINCIPAL	4
Objeto Centro de Investigação	4
Objeto Pessoa	4
Objeto Projeto	4
Objeto Tarefa	5
INTERFACE GRÁFICA	5
ESTRUTURA DO FICHEIRO DE CONFIGURAÇÃO	6
EXECUÇÃO DO PROGRAMA	7
Menu Principal	7
Menu de Criação de Projetos	7
Menu de Gestão do Projeto	8
Menu de Criação de Tarefa	9
CONCLUSÃO	10
ANEXOS	11
Diagrama de Classes - Meta 1	11
Diagrama de Classes - Meta 2	12

1. INTRODUÇÃO

Este trabalho prático tem como objetivos desenvolver uma aplicação recorrendo à linguagem de programação JAVA para auxiliar o Centro de Informática e Sistemas da Universidade de Coimbra (CISUC) para gerir vários projetos de investigação. A aplicação usa uma interface gráfica que permite facilitar a utilização pelo utilizador.

O programa está dividido em dois packages (*GUI* e *SourceCode*) separando desta forma o código da interface gráfica do código principal.

O código principal está dividido em vários ficheiros representando cada um objetos em JAVA:

- *Main*, classe principal do programa.
- *InvestigationCenter*, classe relativa aos centros de investigação.
- *Person*, classe relativa às pessoas.
- *Teacher*, classe relativa às pessoas que são docentes.
- *Grantee*, classe relativa às pessoas bolseiras.
- *AdvisedStudent*, classe relativa aos bolseiros que têm um orientador.
- *Bachelor*, classe relativa aos bolseiros licenciados orientados.
- *Master*, classe relativa aos bolseiros mestres orientados.
- *PhD*, classe relativa aos bolseiros doutorados.
- *Project*, classe relativa aos projetos.
- *Task*, classe relativa às tarefas.
- *Development*, classe relativa às tarefas de desenvolvimento.
- *Documentation*, classe relativa às tarefas de documentação.
- *Design*, classe relativa às tarefas de design.

De maneira análoga temos os objetos da interface gráfica:

- *InvestigationCenterGUI*, classe relativa à *GUI* para gerir os projetos em geral.
- *ProjectCreatorGUI*, classe relativa à *GUI* para criar um projeto.
- *ProjectManagementGUI*, classe relativa à *GUI* para gerir um projeto específico.
- *TaskCreatorGUI*, classe relativa à *GUI* para criar uma tarefa.

2. CÓDIGO PRINCIPAL

O ficheiro `main.java` inicia o nosso programa pela verificação da existência de um ficheiro objeto (*.obj*) para o poder ler e atribuir o seu conteúdo aos objetos do programa. Caso ele não exista, vai ler um ficheiro texto (*init.txt*) que contém os dados necessário para iniciar a aplicação.

a. Objeto Centro de Investigação

Esse objeto é caracterizado pelo o seu nome e pelas listas (*ArrayList*) de pessoas e projetos que ele contém.

Ele tem uns métodos simples que permitem devolver os seus projetos que estão concluídos ou não.

b. Objeto Pessoa

O objeto Pessoa (*Person*) é a generalização de *Teacher* e *Grantee* ou seja, dos Docentes e Bolseiros. Os próprios bolseiros são a generalização dos Estudantes Orientados (*AdvisedStudent*) e dos Doutores (*PhD*). Os Licenciados (*Bachelor*) e os Mestres (*Master*) são Estudantes Orientados.

De forma geral uma Pessoa tem um nome, um e-mail, uma lista de Tarefas (*Task*). Consoante a instância de Pessoa, podemos ter um único Projeto (*Project*) ou uma lista no caso dos Docentes. Por outro lado os Bolseiros têm datas início e de fim de bolsa .Os Estudantes Orientados têm uma lista de Docentes e finalmente os Licenciados, Mestres e Doutorados têm um custo por mês da bolsa que difere consoante a instância.

Uma Pessoa tem métodos para atribuir e remover tarefas, e verificar se ela está sobrecarregada ao adicionar-lhe uma tarefa.

c. Objeto Projeto

Outra estrutura de dados utilizada é o objeto Projeto que tem um nome, uma data inicial, uma duração, uma data de fim que lhe será atribuída quando o projeto for arquivado recorrendo a atualização do seu atributo de estado (*status*). O projeto tem uma lista de tarefas bem como uma lista de pessoas. Os seus métodos principais permitem devolver várias informações úteis para depois serem tratadas pela interface gráfica, nomeadamente os professores do projeto, as tarefas (completadas, não concluídas, não concluídas no tempo estimado, não começadas ou todas), adicionar membros, criar tarefas, remover tarefas, calcular o custo e arquivar o projeto.

d. Objeto Tarefa

A última estrutura de dados utilizada é o objeto Tarefa, uma generalização de Documentação (*Documentation*), Desenvolvimento (*Development*) e Design (*Design*) onde as subclasses têm uma taxa de esforço diferente. À semelhança dos Projetos, uma tarefa tem um nome, uma data de início, uma duração (em dias), uma data de fim e o estado de conclusão da mesma (uma percentagem).

3. INTERFACE GRÁFICA

As diferentes classes de interface gráfica ocupam-se de inicializar e posicionar todos os botões, labels, listas etc... consoante o tipo de menu. Cada classe contém outra privada que permite fazer um *override* dos eventos da janela (por exemplo para poder guardar os dados quando a janela fecha) e do rato (para atualizar listas). Esta classe permite também gerir os eventos relativos aos botões e efetuar as ações descritas pelos próprios.

4. ESTRUTURA DO FICHEIRO DE CONFIGURAÇÃO

A leitura do ficheiro texto é feita por *tokens* que permitem identificar o tipo de informação que é adicionada ao centro de investigação (fig.1). Respeitando as tabulações, os tokens são os seguintes:

- “*INVESTIGATION CENTER\t*” decora o nome do centro de investigação.
- “*PROJECT\t*” lê as informações do projeto.
- “*\tTASK\t*” efetua a leitura das tarefas.
- “*\tTEACHER\t*” são lidos os dados relativos ao docente.
- “*\tBACHELOR\t*” faz a leitura das informações do licenciado
- “*\tMASTER\t*” lê os dados do mestre
- “*\tPhD\t*” efetua a leitura das informações do doutor.

Excepto no caso do centro de investigação, são lidas as informações uma por uma que estão separadas pelo caracter “/” e atribuídas à estrutura de dados correspondente.

```
=====
INVESTIGATION CENTER:  CISUC
=====
PROJECT:  ROBOT/ROB/01-10-2019/1
TASK:     Programing/DEV/01-10-2019/20
TASK:     Connect Cables/DEV/01-10-2019/10
TASK:     Connect LEDS/DEV/11-10-2019/10
TASK:     Connect Battery/DEV/21-10-2019/1
TASK:     Log/DOC/22-10-2019/2
TASK:     Text/DOC/22-10-2019/2
TASK:     Make Shape/DES/24-10-2019/3
TASK:     Paint Robot/DES/26-10-2019/5
TASK:     Javadoc/DES/22-10-2019/2
TASK:     Debug/DEV/24-10-2019/3

PROJECT:  ROBOT2/ROB2/01-12-2019/1
TASK:     Programing2/DEV/01-12-2019/20
TASK:     Connect Cables2/DEV/01-12-2019/10
TASK:     Connect LEDS2/DEV/11-12-2019/10
TASK:     Connect Battery2/DEV/21-12-2019/1
TASK:     Log2/DOC/22-12-2019/2
TASK:     Text2/DOC/22-12-2019/2
TASK:     Make Shape2/DES/24-12-2019/3
TASK:     Paint Robot2/DES/26-12-2019/5
TASK:     Javadoc2/DES/22-12-2019/2
TASK:     Debug2/DEV/24-12-2019/3

PEOPLE:
TEACHER:  Abílio dos Santos/abilio@dei.uc.pt/10000000/Computer/Science
TEACHER:  Joaquim Alberto/quimquim@dei.uc.pt/10000001/Jardinagem
TEACHER:  Zara Alberta Uva/zau@dei.uc.pt/10000010/Eletrô
BACHELOR: Dylan Perdigão/dpg@student.dei.uc.pt/01-10-2018/10-07-2020
BACHELOR: Bruno Faria/brunofaria@student.dei.uc.pt/01-10-2018/10-07-2020
MASTER:  Antonio Mendes/mendes@student.dei.uc.pt/01-10-2018/10-07-2020
MASTER:  Fernando Magalhães/fern.mag@student.dei.uc.pt/01-10-2018/10-07-2020
PhD:     Jorge Sousa/sousa@student.dei.uc.pt/01-10-2018/10-07-2020
PhD:     Eduardo Costa/costa@student.dei.uc.pt/01-10-2018/10-07-2020
PhD:     Mário Centeno/mario@student.dei.uc.pt/01-10-2018/10-07-2020
PhD:     Marcelo Costa/marcelo@student.dei.uc.pt/01-10-2018/10-07-2020
```

Fig.1 Ficheiro de texto com as configurações

5. EXECUÇÃO DO PROGRAMA

a. Menu Principal

Ao executar a aplicação o programa vai ler o ficheiro texto caso não exista ficheiro objeto. Além disso, a interface gráfica aparece da forma da fig.2, com três listas, uma contendo as pessoas do centro de investigação, outra com os projetos e a última com os membros do projeto selecionado.

Do lado direito temos botões para adicionar e remover um projeto selecionado na lista e adicionar e remover uma pessoa a um projeto. Caso o utilizador tenha efetuado mal as manipulações, aparecerá uma janela de erro.

Por baixo da lista de pessoas situa-se um botão que abre uma janela de diálogo mostrando as informações relativas à pessoa selecionada. Por último, entramos no gestor do projeto carregando no botão “*Enter in Project*”.

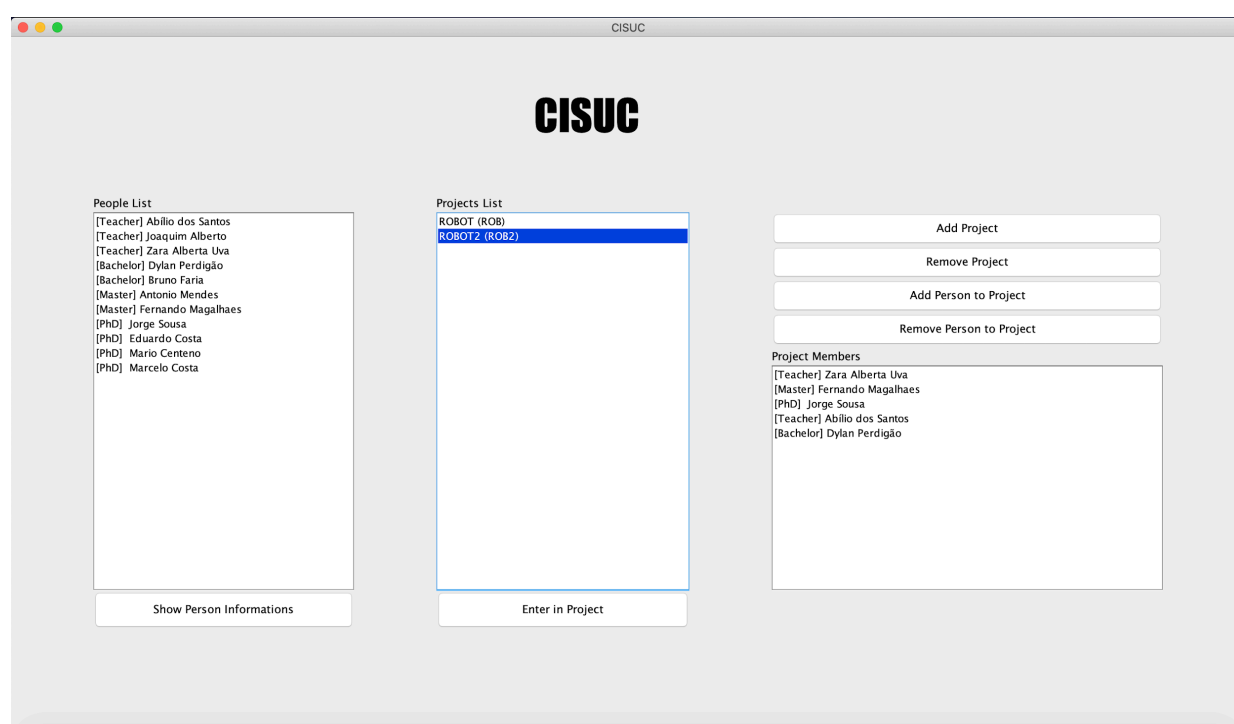


Fig.2 Menu Principal

b. Menu de Criação de Projetos

Ao criar um projeto (fig. 3), atribuímos um nome, um acrônimo, uma data de início e a duração prevista em meses. Temos a possibilidade de abortar a criação ao voltar para o menu principal com o botão *Cancel* ou então confirmar com o botão *Add Project*. Caso haja algum *JTextField* com parâmetros inválidos, uma janela de erro surge.

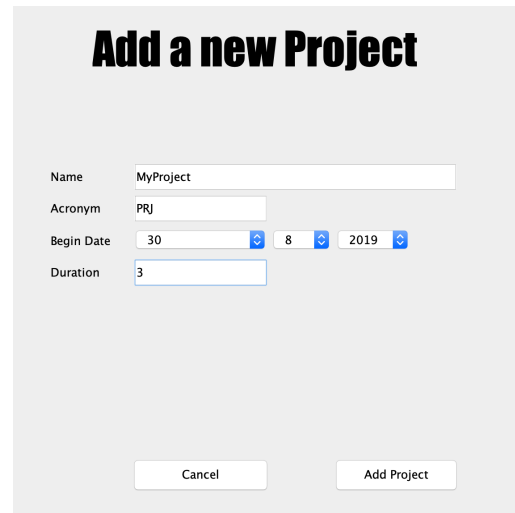


Fig.3 Criar projeto

c. Menu de Gestão do Projeto

O gestor do projeto (fig.4) é composto por diversas listas. Começando pela esquerda, temos as tarefas não começadas (com percentagem de progresso igual a zero e com a data de início inferior à atual), as tarefas não terminadas no tempo previsto e as tarefas completadas (com percentagem de progresso igual a 100%).

À direita começando por cima, podemos atribuir um investigador principal ao projeto. Em seguida temos duas listas onde as informações dos membros e das tarefas pode ser consultadas com os botões *Show Member Information* e *Show Task Information* respetivamente. Além disso temos, um botão para atribuir a um Estudante Orientado um ou mais Docentes que vão ser os seus orientadores via uma janela que permite escolher os docentes, um de cada vez.

Para atribuir uma tarefa a uma pessoa, seleciona-se, nas respetivas listas, a pessoa e a tarefa antes de carregar no botão.

É possível criar ou remover uma tarefa com os botões abaixo da lista de tarefas. Temos ainda a possibilidade de modificar a percentagem de conclusão de uma tarefa carregando no botão *Update Completion*.

Por último, os dados do projeto estão escritos na parte de baixo da janela, com o botão para poder arquivá-la (e desse modo o projeto ficar disponível só para consulta). Caso o projeto esteja arquivado, os botões para efetuar modificações desaparecem, evitando deste modo a manipulação dos dados.

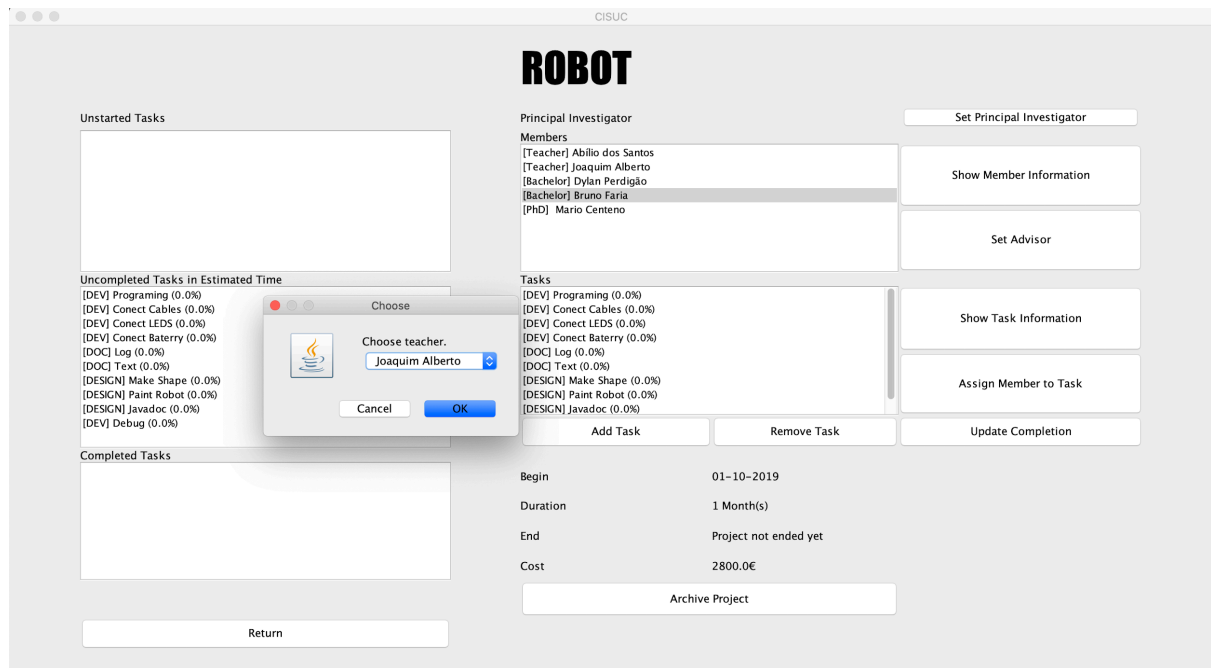


Fig.4 Gestor do projeto

d. Menu de Criação de Tarefa

À semelhança da interface gráfica de criação de projetos temos a interface das tarefas (fig.5) que é praticamente igual, mudando só um dos *inputs*, o tipo de Tarefa.

As funcionalidades dos botões também permitem recuar para a interface anterior (neste caso o gestor de projetos) e adicionar a tarefa.

Fig.5 Adicionar tarefas

6. CONCLUSÃO

De forma geral a nossa abordagem ao problema não foi muito alterada em relação à primeira meta do trabalho, como podemos observar nos diagramas de classes (ver secção dos anexos) tivemos de criar uma classe de Estudante Orientado para generalizar o facto dos Licenciados e Mestres terem orientadores.

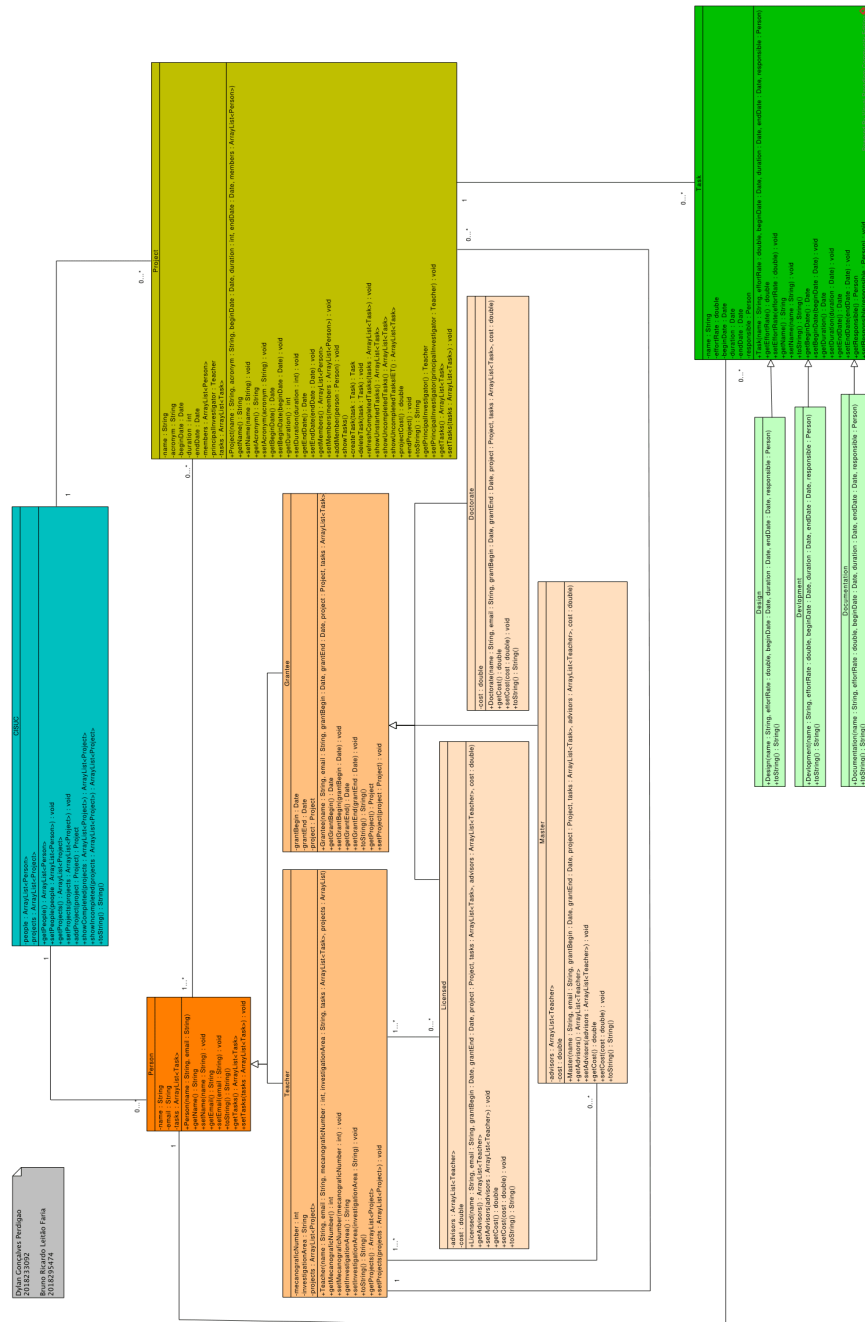
Certos métodos foram adicionados ao programa para ajudar à execução do mesmo, e foram adicionadas as classes da interface gráfica pedidas nesta meta.

De entre as várias linguagens existentes que nos possibilitavam a realização deste trabalho, este foi feito recorrendo à linguagem JAVA. Sendo esta uma linguagem de programação orientada objetos, permitiu-nos ter uma melhor perceção de como os dados são tratados e armazenados.

Posto isto, foi um projeto que nos proporcionou um conhecimento mais profundo no mundo da programação, mais especificamente da linguagem JAVA, e do qual retiramos competências que certamente nos vão proporcionar um melhor futuro como Engenheiros Informáticos.

7. ANEXOS

a. Diagrama de Classes - Meta 1



b. Diagrama de Classes - Meta 2

