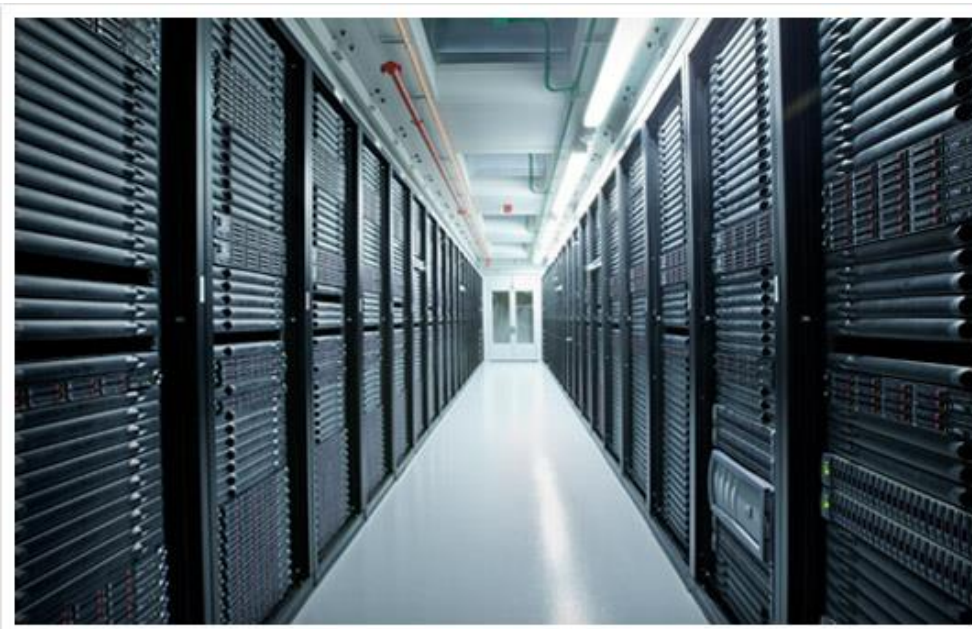


# Relatório do projeto de Programação Orientada aos Objetos



UNIVERSIDADE D  
COIMBRA

Centro de Informática e Sistemas da Universidade de Coimbra (CISUC)



Trabalho realizado por:

- Alexandre Cortez Santos 2018277730 - TP8
- Guilherme Eusébio Reis Gaspar 2018278327 - TP7

## Projeto - Classe normal

É uma classe normal, constituída por 11 atributos:

```
10     private String nome,acronimo;
11     private int dEstimada;
12     private Data dataInicial,dataFinal;
13     private int custo;
14     private boolean estado;
15     private ArrayList<Docente> docentes;
16     private ArrayList<Bolsseiros> bolsseiros;
17     private ArrayList<Tarefa> tarefas;
18     private Docente IP;
```

A data final só é definida quando o projeto está concluído, cujo o estado é verificado pelo atributo “estado, que é atualizado sempre que uma tarefa for concluída.

## Data – Classe normal

Classe com os atributos, dia, mês e ano. E que tem como métodos gets e sets de cada variável.

## Pessoa – Superclasse

Esta classe é abstrata. Contém 3 atributos, o nome, o mail e a disponibilidade de tipo “double” que tem como objetivo verificar se o individuo está sobrecarregado. E tem um método abstrato custo, que vai servir para devolver o custo mediante cada subclasse.

- **Bolsseiros – Subclasse abstrata:** constituído pelos atributos da superclasse Pessoa mais os atributos data de inicio e data final que são variáveis do tipo de classe Data.
  - **Formandos – Subclasse abstrata:** o atributo, para além daqueles que são herdados pela classe bolsseiros, é um ArrayList de Docentes que estão associados ao mesmo projeto que os bolsseiros
    - **Licenciados – Subclasse:** Esta subclasse só difere no método custo, no qual devolve 800
    - **Mestres – Subclasse:** Esta subclasse só difere no método custo, no qual devolve 1000
    - **Doutorados – Subclasse:** Semelhante á subclasse Formandos, só que não é abstrata, não contem o atributo ArrayList de Docentes e não tem os métodos get e set para esse atributo. E no método custo devolve 1200
    - **Docentes – Subclasse:** também constituído pelos atributos da superclasse Pessoa em conjunto com os atributos numero mecanográfico e área.

## Tarefa – Superclasse abstrata

Esta superclasse contem 6 atributos: nome, percentagem de conclusão, data de inicio, data final, duração estimada e a pessoa responsável. E contem 2 métodos abstratos e 1 método essencial.

## Métodos referidos:

- **txEsf()** – é um metodo abstrato que devolve o valor da taxa de esforço consoante a tarefa
- **putResp(Pessoa p)** – é o set mas com outro nome
- **setResp(Pessoa p)** – é um metodo abstrato que devolve um boolean consonte, a pessoa ficou responsável ou não pela tarefa que depende da taxa de esforço sobrecarregar ou não a pessoa
  - **Design – Subclasse:** Esta subclasse so difere no método txEsf, no qual devolve 0.5
  - **Documentacao – Subclasse:** Esta subclasse so difere no método txEsf, no qual devolve 1
  - **Desenvolvimento – Subclasse:** Esta subclasse so difere no método txEsf, no qual devolve 0.25

## Cisuc – Classe

Classe responsável pela interface. Possui os métodos responsáveis pela ação dos botões e de todos os outros componenetes da Graphic User Interface.

# Funcionamento do programa

Ao iniciar o programa, é iniciado uma frame, onde nos é apresentado um painel com as seguintes opções:



Em “Listar Pessoas”, vamos poder obter a informação de todas as pessoas do ficheiro “Pessoas.txt”, usando uma comboBox, de forma a saber que tipo de pessoas estamos a falar (Docentes, Licenciados, Mestres ou Doutorados);

Em “Criar Projeto”, podemos inserir os dados do projeto que pretendemos criar. Ao criar, o projeto é adicionado ao sistema.

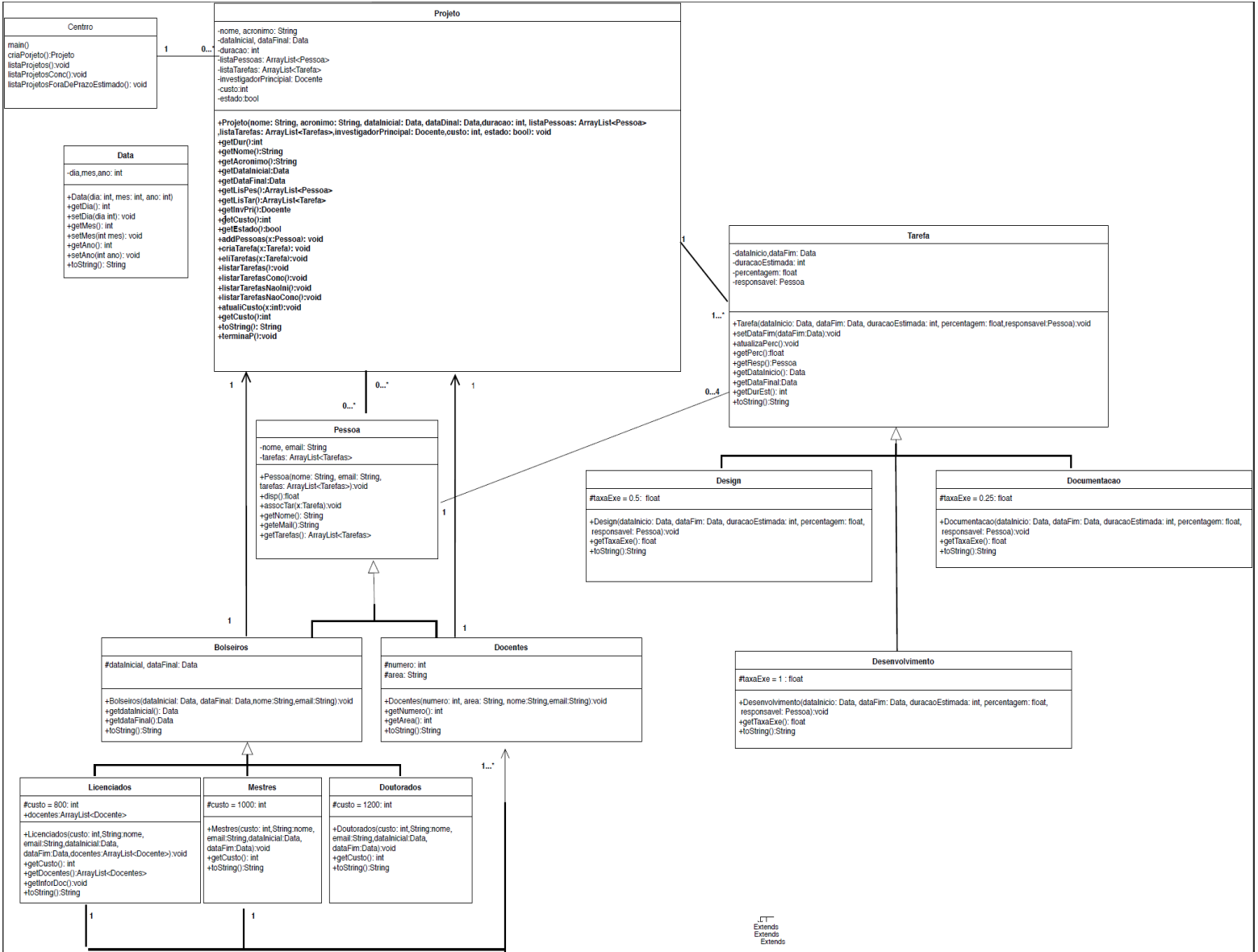
Em “Listar Projetos”, é nos dadas as opções “Listar Todos os projetos”, “Listar projetos concluídos” e “Listar projetos não concluídos”.

Por fim, “Associar pessoa a um projeto”, que tal como o nome indica, permite-nos associar uma pessoa a um projeto.

Todos os painéis possuem um botão voltar, que assim permite-nos navegar pela aplicação sem ter a necessidade de fechar a aplicação se quisermos voltar a um painel anterior.

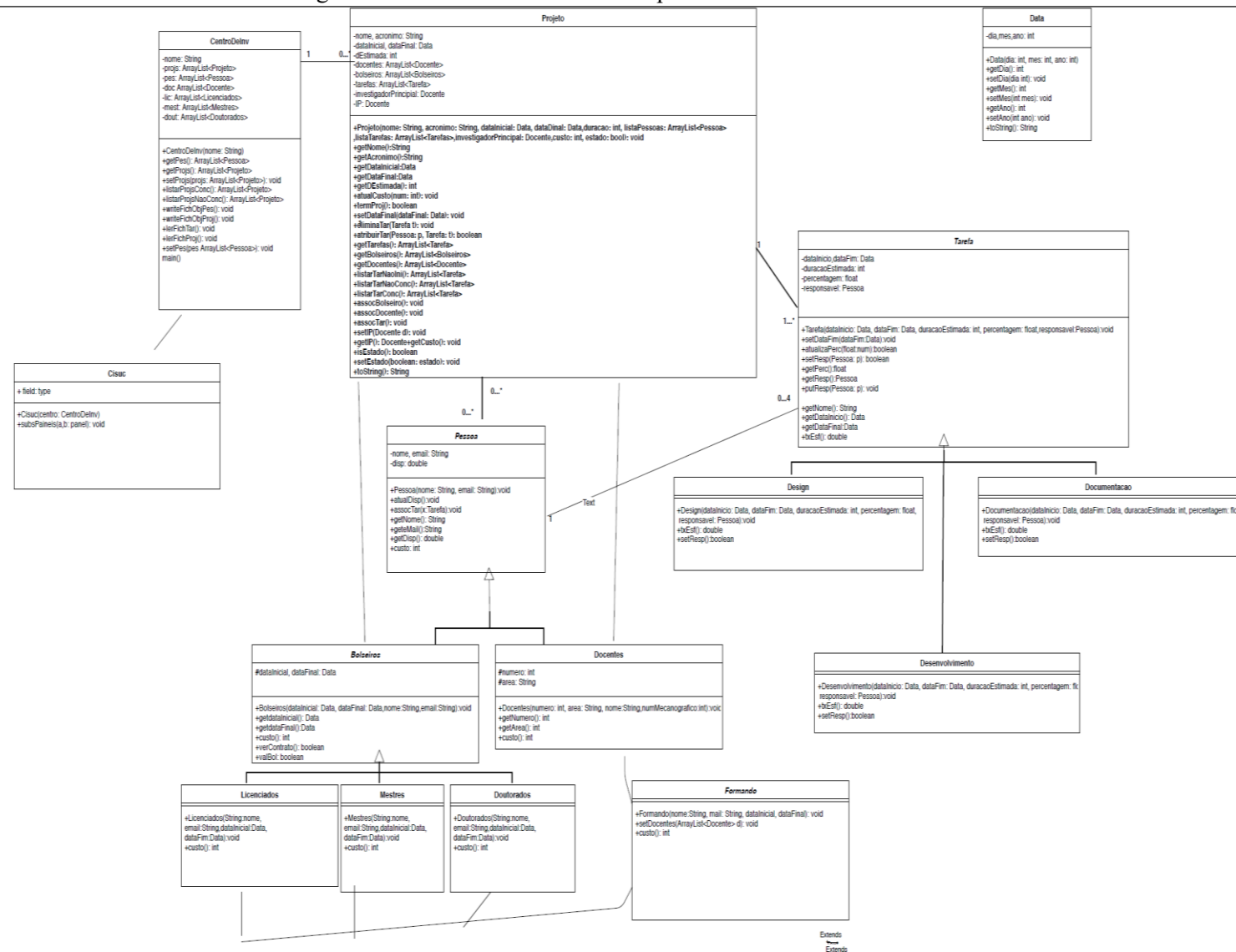
# Diagramas

Inicial:



# Final:

Nota: o que está em *itálico>* é porque é abstrato e a classe Data está associada a todas as classes que a chamam. Problemas com a nitidez do diagrama... será enviado em formato pdf também.



## Ficheiros

Utilizamos três ficheiros de texto, “Pessoas.txt”, “Projetos.txt”, ”Tarefas.txt”. O ficheiro “Pessoas.txt” está constituído por 100 pessoas, sendo que os seus dados são apresentados da seguinte forma: Tipo, Nome, Mail, NumeroMecanografico, se estivermos a falar de um docente e Tipo, Nome, Mail, DataInicioBolsa, DataFimBolsa se estivermos a falar de um bolseiro. Estes dados são carregados para um ficheiro de objetos “pessoas.obj”.

O ficheiro “Projetos.txt” está constituído por 10 projetos, sendo que os seus dados são apresentados da seguinte forma: NomeDoProjeto, Acronimo, Custo, DataInicio, DuraçãoEstimada, DataFinal, Estado, NumeroMecanograficoDo DocenteResponsavel, NumeroMecanograficoDosDocentesAssociados, MailDosBolseirosAssociados e estes dados são carregados para um ficheiro de objetos “projetos.obj”.

O ficheiro “Tarefas.txt” está constituído da seguinte forma: AcronimoProjeto, TipoTarefa#NomeTarefa#DuraçãoEstimadaExecução#DataInicio#DataFim#Responsavel, TipoTarefa#...

## Conclusão

Com este projeto, construímos uma pequena aplicação para gerir um CISUC, através da linguagem Java, no âmbito da cadeira de Programação Orientada aos Objetos. Ganhámos uma maior aprendizagem no que diz respeito à herança, ao polimorfismo e ao encapsulamento/proteção de dados e às bases deste tipo de linguagem. O nosso relatório final está também com algumas mudanças em relação ao inicial.