

Licenciatura em Engenharia Informática

# Índice

[Índice 2](#_Toc105809834)

[Introdução 3](#_Toc105809835)

[Implementação 4](#_Toc105809836)

[Funcionalidades Implementadas 5](#_Toc105809837)

# Introdução

O trabalho prático de Programação Avançada consiste na criação de uma aplicação, em Java, que sirva de apoio ao processo de gestão de estágios e projetos do Departamento de Engenharia Informática e Sistemas do ISEC.

A aplicação está dividida em várias fases em que o utilizador vai inserindo informação e o programa manipula essa mesma informação de modo a facilitar todo o processo de atribuição de estágios e projetos.

Na primeira meta esta aplicação conta com uma interface em modo de consola mas pretende-se implementar uma interface gráfica mais à frente com recurso a JavaFX.

Uma imagem com texto

Descrição gerada automaticamente

# Implementação

A aplicação que estamos a desenvolver baseia-se numa máquina de estados finita (FSM – Finite State Machine) e, portanto, começámos por implementar as classes que iam representar cada estado. Precisámos também de criar uma enumeração com os vários estados possíveis, um adaptador para as várias classes-estado, uma interface com os métodos que podem fazer com que a FSM mude de estado e um contexto para a máquina de estados que permitisse a interface interagir com a mesma.

De seguida, começámos a trabalhar na interface em modo de texto mas precisávamos de classes que guardassem os dados e que representassem as várias entidades existentes, tais como os alunos e os docentes, então foi mesmo isso que fizemos.

A classe de dados contém coleções (*ArrayList*) que guardam os objetos das várias classes que compõem e representam os dados e criámos métodos que pudessem obter e alterar esses valores.

Para a primeira fase da máquina de estados implementámos os métodos para ler um ficheiro CSV com os dados dos alunos, dos docentes e das propostas e criámos objetos de cada tipo à medida que íamos lendo cada linha do ficheiro. Na segunda fase fizemos o mesmo só que para as candidaturas.

A terceira fase foi a mais trabalhosa pois tivemos de implementar um algoritmo que atribuísse automaticamente as propostas aos vários alunos. A quarta fase foi essencialmente a mesma coisa só que em vez de atribuir propostas tivemos de atribuir orientadores de forma automática e também manual.

Quanto à quinta fase, esta foi bastante mais simples visto que apenas foi necessário implementar pesquisas simples relativas a todo o processo.

Para que fosse possível guardar o estado da aplicação quando esta fosse fechada fizemos com que todas as classes do modelo implementassem a interface *Serializable* e utilizámos métodos da classe *ObjectInputStream* e *ObjectOutputStream* para ler e escrever o contexto da máquina de estados.

# Funcionalidades Implementadas

Todas as funcionalidades relativas à primeira meta foram implementadas exceto a consulta de alunos com autoproposta associada e consulta de propostas de alunos autopropostos.

De resto, as 5 fases da máquina de estados encontram-se implementadas, bem como guardar e retomar o estado da aplicação.

Uma imagem com texto

Descrição gerada automaticamente