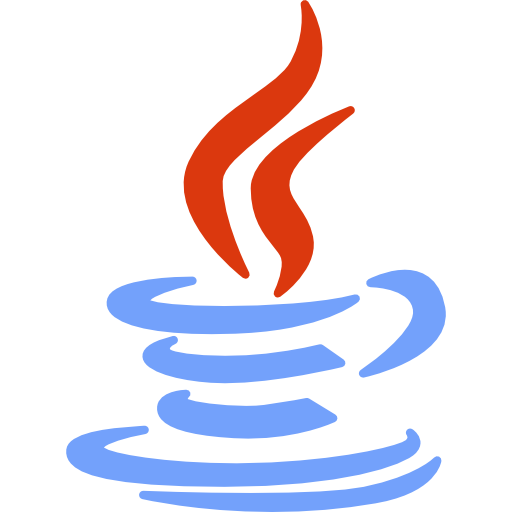
Uma imagem com texto

Descrição gerada automaticamente

Programação Avançada

2021/2022



Meta 2

**Trabalho realizado:**

Rúben Santos – 2019116244

Índice

[Introdução 3](#_Toc102059794)

[Opções e Descrições Tomadas 4](#_Toc102059795)

[Diagrama da Máquina de Estados 6](#_Toc102059796)

[Descrição das Classes 6](#_Toc102059797)

[Aluno 7](#_Toc102059798)

[Docente 7](#_Toc102059799)

[Proposta 7](#_Toc102059800)

[Candidatura 7](#_Toc102059801)

[ManagementData 7](#_Toc102059802)

[AtribuicaoManualAluno, AtribuicaoManualDocente, RemocaoManualAluno, RemocaoManualDocente 7](#_Toc102059803)

[CommandManager 8](#_Toc102059804)

[CommandAdapter 8](#_Toc102059805)

[ICommand 8](#_Toc102059806)

[ConfiguracaoState, CandidaturaState, PropostaState, OrientadoresState, ConsultaState 8](#_Toc102059807)

[ConfiguracaoFechadaState, CandidaturaFechadaState, PropostaFechadaState, OrientadoresFechadaState, ConsultaFechadaState 8](#_Toc102059808)

[ManagementStateAdapter 8](#_Toc102059809)

[ManagementContext 9](#_Toc102059810)

[Relacionamento entre Classes 10](#_Toc102059811)

[Funcionalidades da Aplicação 11](#_Toc102059813)

# Introdução

Este projeto consiste no desenvolvimento de uma aplicação para apoiar ao processo de gestão de projetos e estágios do Departamento de Engenharia Informática e de Sistemas do ISEC.

Para esta fase foram desenvolvidas todas as funcionalidades pedidas no que toca à interface gráfica.

# Opções e Descrições Tomadas

Tendo em conta que esta aplicação dispõe de cinco fases, foram criados cinco estados inicialmente, utilizando o modelo FSM foi criada uma ligação entre os vários estados e também foram impostas as devidas restrições para a transição entre eles.

Como existem restrições para quando um dos estados se encontra fechado, foram criados mais cinco estados relacionados com os anteriores para poder haver uma distinção entre o estado no seu funcionamento normal e no estado fechado.

A package data é onde estão todas as classes armazenadas, e onde toda a informação vai ser gerida relativa a:

* Alunos, Docentes, Propostas e Candidaturas.

A interface do utilizador é a única parte da aplicação em que será possível o utilizador ter alguma interação e é a partir daí que ele controla para onde quer ir ou que ações pretende executar.

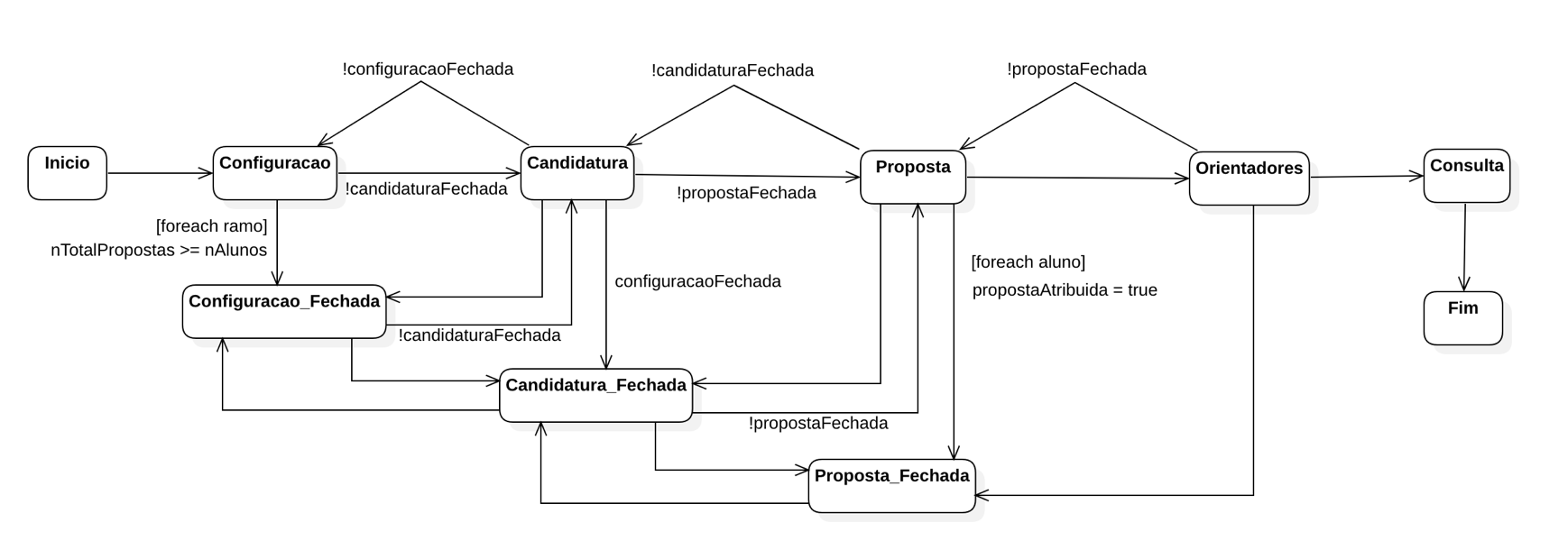
Para a leitura de comandos relativamente à atribuição manual de propostas para alunos ou docente, foram criadas as respetivas classes de atribuição e remoção, assim como os métodos de undo e redo.

A maneira como estas funcionalidades foram implementadas:

* ManagementData: toma o comportamento de receiver (alvo das ações/comandos)
* ManagementContext: toma o comportamento de facade (cria um objeto Command e dependendo do método chamado desencadeia as ações para executar o comando).

Na GUI optei por utilizar uma classe Singleton presente em todas as classes que pretendam apresentar algum output. Com esta classe é possível ter um único objeto com a intenção de guardar nele a ultima mensagem gerada, sendo assim possível de ser acedida através dos objetos da GUI e apresentada ao utilizador.

# Diagrama da Máquina de Estados



# Descrição das Classes

## **Aluno**

Classe que guarda a informação relativa aos alunos.

## **Docente**

Classe que guarda a informação relativa aos docentes.

## **Proposta**

Classe que guarda a informação relativa às propostas. Serve como classe base para as derivadas, que são:

* Estagio
* Projeto
* Autoproposto

Cada classe derivada representa um tipo de proposta distinto tendo alguns parâmetros exclusivos.

## **Candidatura**

Classe que guarda a informação relativa às candidaturas. Tem por objetivo saber a que propostas o aluno se candidatou.

## **ManagementData**

Classe que gere toda a informação e decorre todas as ações vindas da interface do utilizador.

## **AtribuicaoManualAluno, AtribuicaoManualDocente, RemocaoManualAluno, RemocaoManualDocente**

Classes utilizadas através de comandos, em que o objetivo é possibilitar as funcionalidades de undo e redo.

## **CommandManager**

Classe usada para gerir o histórico de comandos invocados, com o objetivo de devolver o comando respetivo quando for chamado o método undo ou redo.

## **CommandAdapter**

Classe abstrata que implementa a interface de comandos, tem como objetivo obter o *receiver* para a gestão das ações dos comandos.

## **ICommand**

Classe do tipo interface que tem como objetivo a gestão dos métodos da máquina de estados.

## **ConfiguracaoState, CandidaturaState, PropostaState, OrientadoresState, ConsultaState**

Classes que definem os cinco estados da aplicação. Tem como objetivo gerir as ações possíveis dos diversos estados.

## **ConfiguracaoFechadaState, CandidaturaFechadaState, PropostaFechadaState, OrientadoresFechadaState, ConsultaFechadaState**

Classes que definem os cinco estados fechados da aplicação. Tem como objetivo limitar as ações possíveis dos diversos estados quando se encontram fechados.

## **ManagementStateAdapter**

Classe abstrata que implementa a interface de estados e a interface de comandos. Tem como objetivo a gestão dos métodos da interface.

## **ManagementContext**

Classe que contém objetos Command, ManagementData e IManagement, com o objetivo de, dependendo do método que foi chamado, redirecionar para o método específico do objeto.

**IManagementInterface**

Classe do tipo interface que tem como objetivo a gestão dos métodos da máquina de estados.

**ManagementState**

Classe do tipo enum (enumerável) que tem como objetivo guardar os vários estados para ajudar na transição dos mesmos.

**ModelManager**

Classe que gere todas as chamadas a funções vindas da interface gráfica interagindo com o context e a data. Contém as propriedades do PropertyChangeListener, que nos permite atualizar o estado dos objetos na interface gráfica.

**RootPane**

Contém as definições da nossa janela inicial.

**AppMenu**

Classe responsável por desencadear ações relativas à barra de menu no topo da janela. Barra que tem como funções o **save**, **load** e **exit**.

**ListPane**

Classe que gera uma lista com informação relativa a cada janela. Esta informação é atualizada consoante a janela em que o utilizador se encontre.

**Classes da GUI**

Também foram criadas classes na GUI para cada fase.

# Relacionamento entre Classes

# Funcionalidades da Aplicação

|  |  |  |  |
| --- | --- | --- | --- |
|  | Implementado | Não implementado | Especificar |
| Configuração | | | |
| Importar e Exportar CSV | Totalmente |  | Alunos, Docentes e Propostas |
| Alternar entre os modos de gestão | Totalmente |  | Gestão de Alunos, Docentes e Propostas |
| Inserção, consulta, edição e eliminação dos dados referentes a alunos | Totalmente |  | Gestão de dados de alunos |
| Inserção, consulta, edição e eliminação dos dados referentes a docentes | Totalmente |  | Gestão de dados de docentes |
| Inserção, consulta, edição e eliminação dos dados referentes a propostas | Totalmente |  | Gestão de dados de propostas |
| Fechar a fase | Totalmente |  | Fechar a fase atual |
| Avançar para a fase seguinte | Totalmente |  | Avançar para a fase seguinte |
| Candidatura | | | |
| Importar e Exportar CSV | Totalmente |  | Candidaturas |
| Inserção, consulta, edição e eliminação de candidaturas, | Totalmente |  | correspondem à  indicação das propostas pretendidas. |
| Obtenção de listas de alunos | Totalmente |  | Com autoproposta, candidatura já registada, sem candidatura registada |
| Obtenção de listas de propostas de projeto/estágio | Totalmente |  | Autopropostas de alunos, propostas de docentes, propostas com candidaturas, propostas sem candidatura. |
| Fechar a fase | Totalmente |  | Fechar a fase atual |
| Avançar para a fase seguinte | Totalmente |  | Avançar para a fase seguinte |
| Regresso à fase anterior | Totalmente |  | Regresso à fase anterior tendo em conta as restrições caso esteja fechada |
| Proposta | | | |
| Exportar CSV | Totalmente |  | Opções de candidatura, proposta atribuída e ordem desta nas suas opções de candidatura |
| Atribuição automática | Totalmente |  | Das autopropostas ou propostas de docentes com aluno associado, e das propostas disponíveis aos alunos ainda sem atribuições  definidas |
| Gestão da atribuição/remoção manual | Totalmente |  | Com operações de undo e redo |
| Obtenção de listas de alunos | Totalmente |  | Têm autoproposta associada, têm candidatura já registada, têm proposta atribuída, Não têm qualquer proposta atribuída |
| Obtenção de listas de propostas de projeto estágio | Totalmente |  | Autopropostas de alunos, propostas de docentes, propostas disponíveis (não atribuídas a alunos), propostas atribuídas |
| Fechar a fase | Totalmente |  | Fechar a fase atual |
| Avançar para a fase seguinte | Totalmente |  | Avançar para a fase seguinte |
| Regresso à fase anterior | Totalmente |  | Regresso à fase anterior tendo em conta as restrições caso esteja fechada |
| Orientadores | | | |
| Exportar CSV | Totalmente |  | Opções de candidatura, proposta atribuída, ordem desta nas suas opções de candidatura e orientador atribuído. |
| Associação automática dos docentes | Totalmente |  | Trata-se de confirmar a associação proposta-docente |
| Atribuição, consulta, alteração e eliminação de um orientador | Totalmente |  | De um orientador do ISEC (docente) aos alunos com propostas atribuídas |
| Gestão da atribuição/remoção manual | Totalmente |  | Com operações de undo e redo |
| Obtenção de dados | Totalmente |  | Listagem com filtros |
| Fechar a fase | Totalmente |  | Fechar a fase atual |
| Avançar para a fase seguinte | Totalmente |  | Avançar para a fase seguinte |
| Regresso à fase anterior | Totalmente |  | Regresso à fase anterior tendo em conta as restrições caso esteja fechada |
| Consulta | | | |
| Exportar CSV | Totalmente |  | Opções de candidatura, proposta atribuída, ordem desta nas suas opções de candidatura e orientador atribuído. |
| Obtenção de dados diversos sobre todo o processo | Totalmente |  | Listagem com filtros |
| GUI | | | |
| Todas as funcionalidades anteriores | Totalmente |  | As funcionalidades descritas anteriormente estão todas a funcionar em modo gráfico. |
| Save | Totalmente |  | Guarda o estado completo do programa. |
| Load | Totalmente |  | O programa carrega o estado anteriormente guardado. |
| Graficos de Barras | Totalmente |  | Na ultima fase (Consulta), mostra os gráficos de barras como pedido. |
| Gráfico PieChart | Totalmente |  | Na ultima fase (Consulta), mostra o gráfico PieChart com a percentagem e o valor absoluto. |
| JavaDoc | Totalmente |  | JavaDoc compilado. |