|  |  |
| --- | --- |
|  | Trabalho Prático  Aplicação de apoio à gestão  de Projetos e Estágios Meta1  Programação Avançada  Ano Letivo 2021/2022 |

# Implementação

Para a elaboração deste trabalho prático seguiu-se o processo e estrutura de implementação descritas nas aulas, mais nomeadamente pelos exercícios: “Máquina de Gomas” e “Jogo Bolas Brancas e Bolas Pretas”.

Uma imagem com texto

Descrição gerada automaticamente

Assim, como pode ser visto na imagem à esquerda, código foi dividido de acordo com as suas funcionalidades, respeitando também a estrutura dos packages que o enunciado exige.

Toda a parte de dados e máquina de estados permanece separada da user interface em packages distintos.

O package terminado por “model” detém as classes responsáveis por toda a logica da aplicação. “Data” para dados , “fsm” para a maquina de estados e ui para a user interface.

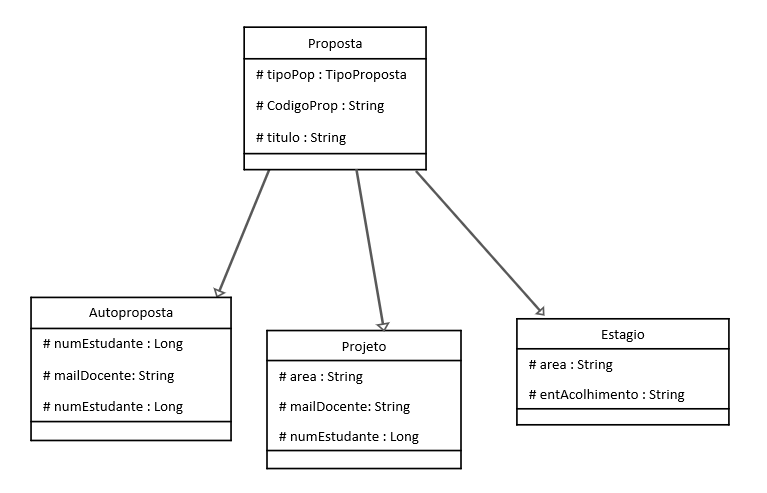
Para a ligação entre os dois pontos, logica e user interface, foi utilizada a Class Context.

Para o input da interface de texto foi utilizada e adaptada a Class PAInput, fornecida durante as aulas práticas de Programação Avançada.

# Classes

# Uma imagem com texto Descrição gerada automaticamente

Em relação às classes de dados, a única relação é a existente entre as classes Proposta, Autoproposta, Projeto e Estágio.



Como visto no diagrama, a classe Proposta é a Classe base das restantes sendo que ela própria não pode ser utilizada unicamente.

Uma imagem com mesa

Descrição gerada automaticamente

A classe ApoioPoeData é a engloba todas as classes do package. Um objeto desta Classe é o responsável por guardar a informação durante a execução da aplicação.

Uma imagem com texto

Descrição gerada automaticamente

No que diz respeito ao package que termina em “fsm”, todas as classes em “states”, que são as que representam cada estado da aplicação, derivam todas da classe base ApoioPoeStateAdapter.

ApoioPoeStateAdapter implementa a interface ApoioPoeIState.

ApoioPoeState é a classe que detém as enumerações/identificações dos estados.

# Tabela de Funcionalidades Implementadas

|  |  |  |
| --- | --- | --- |
| Meta | Funcionalidades | Cumprido |
| 1 | Fase 1 - Configuração de alunos/docentes/Propostas por importação de informação por meio de ficheiros CVS | Sim \* |
| Fase 1 - Exportação de informação sobre alunos/docentes/Propostas por meio de ficheiros CVS | Sim \* |
| Fase 1 - Obtenção de listas de alunos/docentes/Propostas | Sim \* |
| Fase 2 - Configuração de candidaturas por importação de ficheiros CVS | Sim \*\* |
| Fase 2 - Exportação de informação sobre candidaturas por meio de ficheiros CVS | Sim \*\* |
| Fase 2 - Obtenção de listas de candidaturas | Não |
| Fase 3 - Atribuição automática de propostas | Não |
| Fase 3 - Exportação de informação sobre propostas por meio de ficheiros CVS | Sim |
| Fase 4 - Atribuição automática de docentes | Não |
| Fase 4 - Exportação de informação sobre docentes por meio de ficheiros CVS | Sim |
| Fase 5 - Consulta de Listas | Não |
| Obtenção de listas devidamente filtradas | Não |
| Acionamento da conclusão de fase e consequente bloqueamento | Sim |
| Gravação/Carregamento do estado da aplicação usando formato binário | Sim |

Notas:

\* Durante a Fase 1 da Aplicação, recorrendo aos estados ModoAluno, ModoDocente e ModoPropostas é possível importar e exportar dados em cada uma das modalidades de forma independente.

\*\* Implementado na Fase 2, no estado OpcoesCandidaturas.