Instituto Superior de Engenharia de Coimbra

DEIS/LEI

Programação Avançada

Trabalho Prático: Sistema de Gestão de Projetos e Estágios do DEIS

2.ª Meta

Bruno Amado Sousa → n.º 2020132971 - LEI

Jorge Gabriel Querido dos Santos → n.º 2020133143 - LEI

Lousã, 10 de junho de 2022

Índice

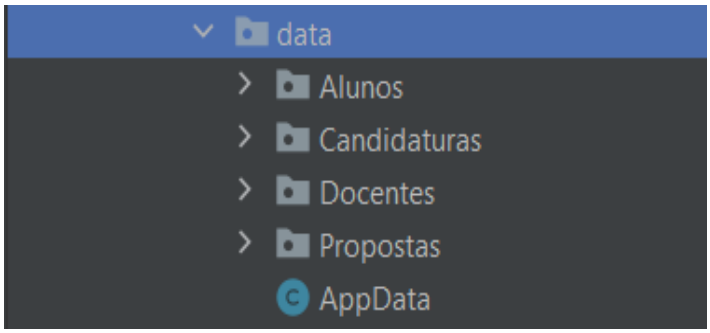
1. Introdução

Este relatório, requerido pelo docente da disciplina, tem por objetivo exemplificar e explicar as decisões tomadas pelo grupo.

Será apresentada uma descrição sintética das opções e decisões implementadas, o diagrama da máquina de estados, uma descrição sucinta das classes e dos relacionamentos respetivos entre elas.

2. Descrição das opções e decisões implementadas

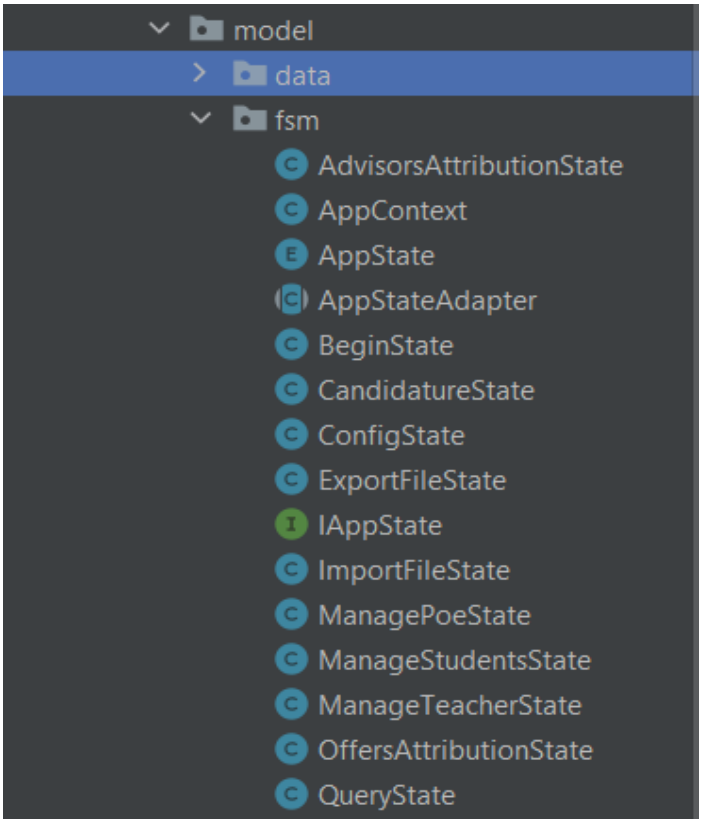
Como podemos ver na imagem que se segue, os dados estão organizados em várias packages que compõem a aplicação.



Decidimos usar Set's e ArrayList's (para o mecanismo de Base de Dados), para guardar múltiplas instâncias dos diferentes objetos (Alunos, Candidaturas, Docentes e Propostas).

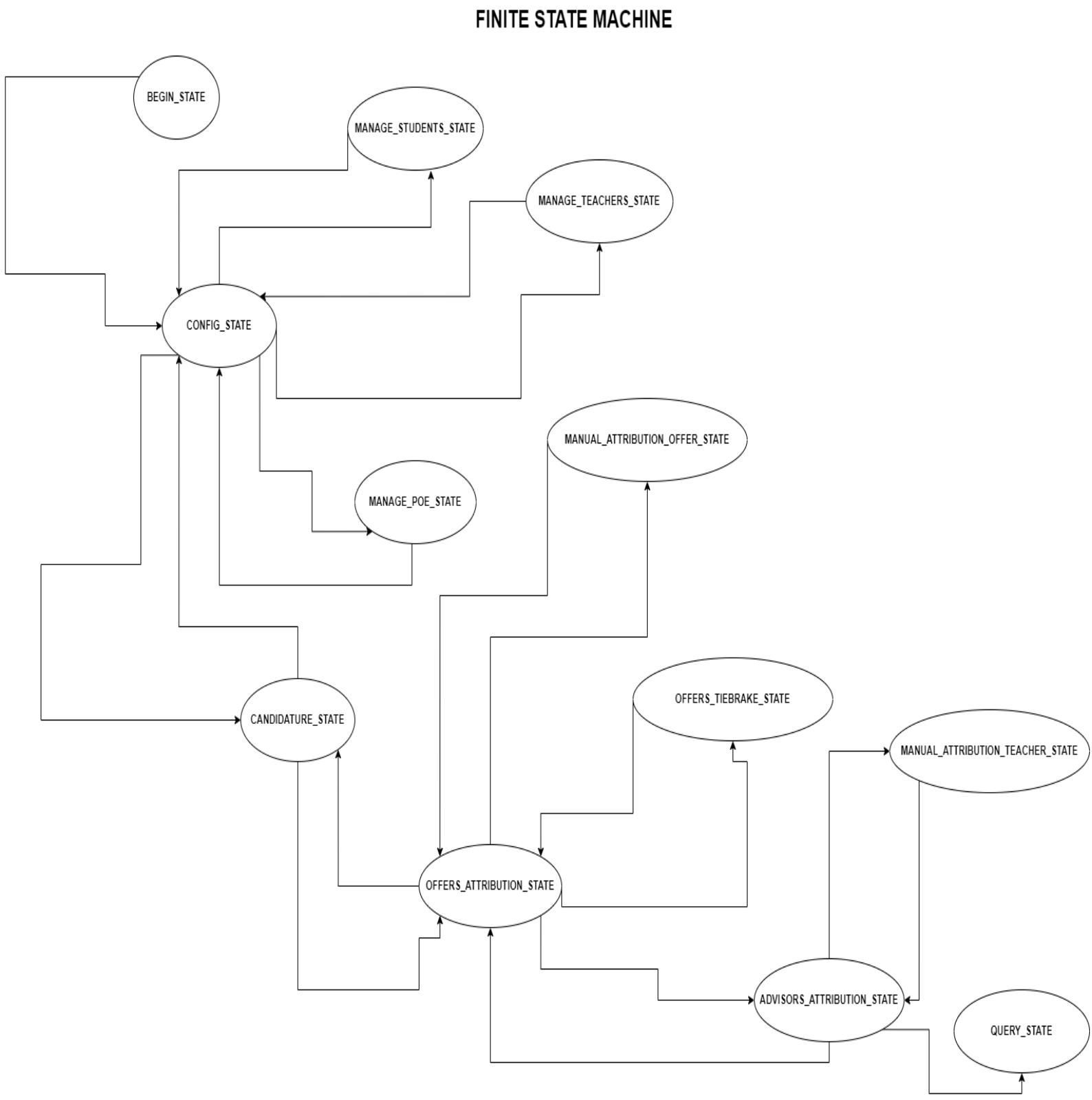
Decidimos, igualmente, criar uma classe - neste caso a AppData - que nos permite interligar os dados com a máquina de estados.

Seguindo a lógica, temos a máquina de estados que é composta pelos estados que se apresenta na imagem da direita.



3. Diagrama da máquina de estados

O diagrama da máquina de estados que idealizámos é o seguinte:



4. Descrição das classes utilizadas

Relativamente às classes que constam na package “data”, podemos dizer que:

- Aluno → é a classe que contêm toda a informação que define o aluno;
- Docente → é a classe que contêm toda a informação que define o docente;
- Proposta → é a classe que contêm toda a informação que constitui uma proposta;
- Candidatura → é a classe que contêm toda a informação que constitui uma candidatura;
- BaseDadosAlunos → é a classe que armazena as informações de cada aluno;
- BaseDadosDocente → é a classe que armazena as informações de cada docente;
- BaseDadosPropostas → é a classe que armazena as informações de cada proposta;
- BaseDadosCandidatura → é a classe que armazena as informações de cada candidatura;
- AppData → é a classe que junta a informação e envia para a state machine.
- FileManagement -> é a classe responsável por export e import dos ficheiros.

Relativamente às classes que constam na package “fsm”, podemos dizer que:

- AdvisorsAttributtionState → é a classe que define o estado de atribuição de orientadores (Fase 4);
- AppContext → é o que liga a lógica da máquina de estados à interface com o utilizador;
- AppState → é a classe que contém a fábrica de estados;
- BeginState → é a classe que define o estado inicial;
- CandidatureState → é a classe que define o estado relativo às candidaturas (Fase 2);
- ConfigState → é a classe que define o estado de configuração dos elementos que constituem a aplicação (Fase 1);
- IAppState → é a classe que contém a interface da máquina de estados;
- ManagePoeState → é a classe responsável pela gestão das propostas (estágios/projetos);
- ManageStudentsState → é a classe responsável pela gestão dos alunos;
- ManageTeacherState → é a classe responsável pela gestão dos docentes;
- OffersAttributionState → é a classe que define o estado de atribuição das propostas (Fase 3);
- QueryState → é a classe permite a consulta dos vários dados do programa (Fase 5).
- ManualAttributionOfferState → é a classe responsável pela atribuição manual das ofertas.
- ManualAttributionTeacherState → é a classe responsável pela atribuição manual dos docentes.
- AppManager → Serve de proxy do AppContext permitindo assim os comandos load, save, undo e redo.
- OffersTieBreakerState → Estado que decide manualmente o desempate entre os empatados na escolha automática de propostas.

Relativamente às classes que representam o modelo gráfico:

Na package states temos classes que definem a constituição do BorderState para cada estado:

- RootPane → Classe que usufrui do StackPane para modificar o tipo de BorderPane presente na janela da App. Basicamente é onde as várias classes se juntam.
- AppMenu → Classe que define um menu género toolbar e que permite abrir novo ficheiro, load, save , undo e redo (Nos estados possíveis).

5. Descrição dos relacionamentos entre classes

(TEXTO) A interface do utilizador (AppGUI) relaciona-se com o contexto da máquina de estados (AppContext) através do proxy AppManager que, por sua vez, guarda uma referência para a "AppData" e a "IAppState". Permitindo, assim, fazer a "ligação" entre a máquina de estados, "fsm", e o utilizador.)

(GRÁFICO) A interface do utilizador (MainJFX) tem os seus constituintes definidos num BorderPane definido no RootPane, cuja sua alteração é feita através de um StackPane e do tipo de estado que a aplicação se encontra. Este modo gráfico segue a sua ligação com a máquina de estados assim como a versão texto.

6. Funcionalidades

- Gestão de Alunos – Feito.
- Gestão de Docentes – Feito.
- Gestão de Projetos ou Estágios – Feito.
- Importar Ficheiros – Feito.
- Exportar Ficheiros – Sim , mas não sabemos para onde está a ir o ficheiro.
- Fase 1 (Configuração) – Feito.
- Fase 2 (Candidaturas) – Feito.
- Gestão de Candidaturas – Feito.
- Obter Listas Que são especificadas no enunciado em relação à fase das candidaturas- Feito.
- Regressar às fases anterior sempre que possível – feito.
- Fase 3 (Atribuição de propostas) – Feito.
- Atribuição automática – Feito.
- Atribuição manual – Feito.
- Redo e Undo – Feito.
- Load e Save – Feito, mas o load não carrega o estado apenas a informação.
- Obter listas referentes à fase 3 – Feito.
- Fase 4 (Atribuição de orientadores) – Feito.
- Atribuição manual e automática referente à fase 4 – Feito.
- Tiebreaker da Fase 3 na seleção automática – Não realizado devido à falta de tempo, pois só nos lembrámos desta feature ao escrever estas funcionalidades.
- Fase 5 – Feito.
- Todas as listas a obter na fase 4 e 5 foram feitas.

7. Conclusão

Com esta atividade, podemos concluir que o sistema-padrão da máquina de estados e da fábrica de objetos ajudam na organização do projeto, sendo assim, essencial a aprendizagem deste tipo de padrões/conceitos.

Ainda , apesar de ser útil, por vezes o javaFX mostra – se complicado de utilizar devido a alguns bugs que apresenta, um exemplo é o handle de mouse events no StackPane, que no nosso trabalho no início não funcionava e depois no final decidiu funcionar de repente.

No entanto, é uma biblioteca poderosa o suficiente para criar projetos interessantes, mas acredito que para 2022 é outdated visto que necessitamos de configurar a VM.

Para finalizar, apesar de ser um trabalho acessível em relação à dificuldade, não deixava de ser um projeto muito grande com muitas funcionalidades, deixando pouco tempo para outros trabalhos e até mesmo para aperfeiçoar este trabalho, no entanto, foi um projeto engraçado que permitiu assimilar os conteúdos dados nas aulas.