**Trabalho Prático DSS**

**FASE I**

Braga, 6 de novembro de 2017



Francisco Oliveira Raul Vilas Boas Vitor Peixoto

A78416 A79617 A79175

**Mestrado Integrado em Engenharia Informática**

**Universidade do Minho**

Enunciado e objetivos

Neste trabalho prático foi-nos pedido a criação de um sistema de gestão dos turnos práticos de um curso. Essa aplicação deverá suportar o registo de Unidades Curriculares (UCs), bem como a sua gestão de alocação e trocas de turnos entre alunos e UCs. Sendo que numa fase inicial a alocação é gerada automaticamente pela Direção do Curso, numa fase posterior os alunos poderão realizar trocas entre eles, com direitos especiais (não necessitarem de outro aluno para trocar) para alunos com estatuto especial. Numa fase posterior a esta, apenas o docente responsável da UC pode alterar a composição dos turnos de acordo com determinadas regras.

Este sistema deve evitar sobreposições de turnos (excetuando quando esta é mesmo inevitável), podendo haver anulações de trocas ou um aluno assistir um turno que não o seu pontualmente.

Nesta fase, no entanto, é-nos apenas pedido: um Modelo de Domínio com as entidades que serão mais relevantes para o sistema; um Modelo de Use Case com as funcionalidades propostas para o sistema e a especificação dos Use Case.

Esta fase do trabalho tem como objetivo planear a estrutura e o comportamento que a aplicação vai ter para podermos ter uma perspetiva geral de como vai funcionar. Assim poderemos ter uma melhor perspetiva do sistema e facilmente detetar erros ou adicionar novas funcionalidades.

Assim nesta fase iremos estabelecer como objetivo a criação desses requisitos para a Fase I para a nossa proposta do sistema de gestão de horários.

Descrição do Trabalho

Modelo de Domínio

O modelo de domínio foi criado com o objetivo de descrever o tipo de estrutura que o sistema iria ter contruindo as suas classes, os seus atributos a as relações entre elas.

Com esse objetivo em mente criamos 9 classes: aluno, horário, turno, UC, professor, troca, sala, tipo de aula e estatuto.

A classe aluno possui os seguintes atributos: id, nome, email e estatuto que representam as variáveis que a classe vai ter. Para o estatuto foi criado uma nova classe para distinguir o estudante do trabalhador estudante.

Sendo o aluno a base do modelo, ele vai pertencer à classe da UC e à do horário. Também vai puder realizar trocas de turnos. A classe da UC tem como atributos o id dela e o seu nome e cada UC tem um professore regente.

A classe Turno tem um professor que a leciona, uma sala que tem uma capacidade, um tipo de aula que pode ser prática ou prática-laboratorial. Os seus atributos são: o id, o dia, a hora e o limite de turno.

A classe professor tem como atributos o id, o nome e o email. Cada professor leciona um turno e por cada UC há um regente da UC.

Modelo de Use Case

O modelo de Use Case é uma maneira de representar a interação do utilizador com o sistema de modo a descrever o comportamento que a aplicação terá, permitindo assim uma melhor perspetiva do sistema.

Para tal, assim como consta no enunciado, dividimos o modelo em 3 fases. A primeira para o registo dos alunos, das UC´s e para a sua alocação nos turnos. A segunda para a troca dos turnos entre os alunos. E por fim uma terceira, respetiva ao regente da UC que lhe permite alterar alunos de turno ou removê-los, caso excedam o limite de faltas.

Para além disto também há use cases que representam outras ações que serão possíveis de efetuar na aplicação como o login, logout, mudar de fase, adição do turno, entre outras.

Especificação dos Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | **Descrição** | **Respetivo à/ao** | **Fase** |
| Login | Permite ao utilizador efetuar o login se fornecer as credenciais corretas | Direção do curso, direção da UC, trabalhador-estudante e aluno |  |
| Logout | Permite ao utilizador sair da sua conta | Direção do curso, direção da UC, trabalhador-estudante e aluno |  |
| Mudar de fase | Altera a fase atual | Direção do curso |  |
| Registar UC | Fornece as informações da UC para a criar | Direção do curso | 1 |
| Registar aluno | Fornece as informações do aluno para o registar | Direção do curso | 1 |
| Alocar aluno | Aloca o aluno num turno inicial | Direção do curso | 1 |
| Trocar Turno (TE) | Permite ao trabalhador estudante trocar de turno caso o turno não se encontre cheio | Trabalhador Estudante | 2 |
| Trocar Turno (Aluno) | Permite ao aluno realizar uma troca de turno com outro aluno, para isto tem de enviar um pedido de troca a outro aluno e este tem de o aceitar | Aluno | 2 |
| Gerir trocas | Deixa o aluno aceitar ou recusar pedidos de troca de turnos | Trabalhador Estudante, Aluno | 2 |
| Trocar de Turno  (Diretor) | Concede a possibilidade de um aluno trocar de turno caso o turno pretendido não se encontre cheio | Diretor da UC | 3 |
| Remover aluno do turno | Remove o aluno do turno | Diretor da UC | 3 |
| Adicionar turno | Cria um novo turno, fornecendo a UC e as suas informações | Direção do Curso, Diretor da UC |  |
| Editar Turno | Permite alterar as informações de um turno | Direção do Curso, Diretor da UC |  |
| Remover Turno | Permite remover um turno | Direção do Curso, Diretor da UC |  |
| Alterar Turno | Super Use Case de: adicionar turno, Editar Turno, Remover Turno | Direção do Curso, Diretor da UC |  |
| Verificar Turno | Permite aos alunos verificarem em que turno estão e as suas informações | Trabalhador Estudante, Aluno |  |

Proposta de Interface

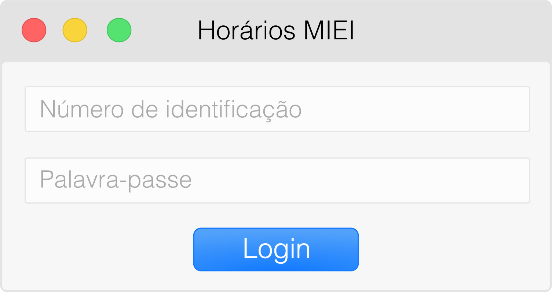
 Login

Fig. 1 – Janela de Login no sistema

Dependendo do número de identificação introduzido, o sistema irá abrir uma janela para cada tipo de utilizador: Direção de Curso; Estudante e Trabalhador Estudante.

Direção de Curso

A janela da Direção de Curso irá ter 4 separadores para cada função permitida a esse utilizador: Registar UC; Registar aluno; Alocar aluno e Gestão dos turnos.

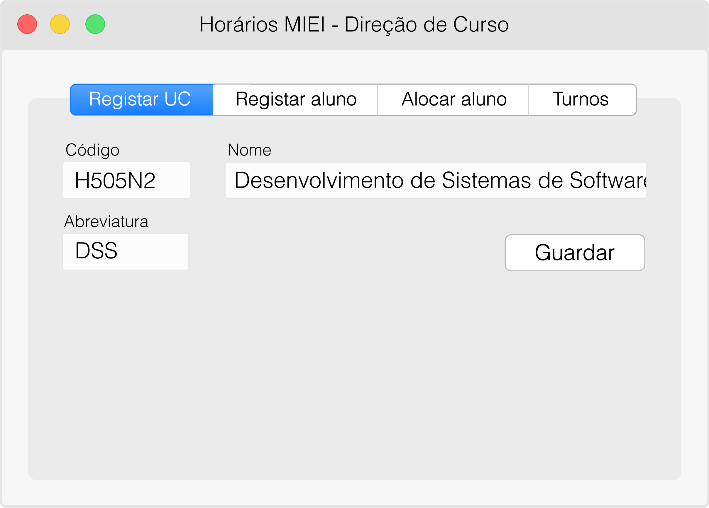
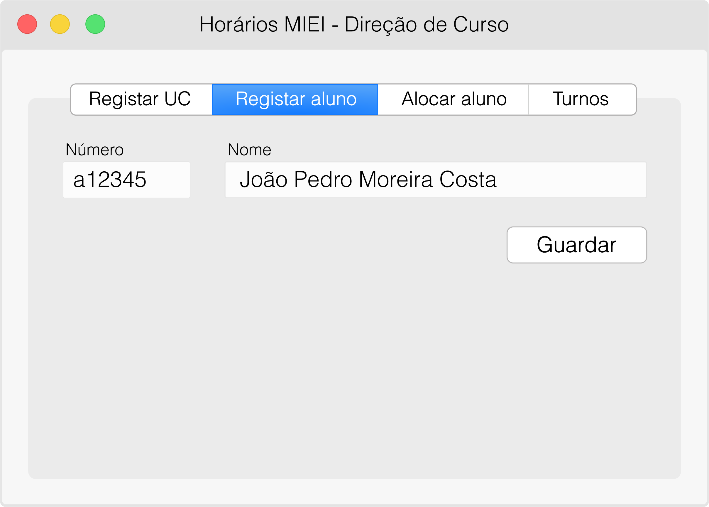




Fig. 2 – Janela da Direção de Curso

Estudante

A janela do Estudante irá ter 2 separadores para cada função permitida a esse utilizador: Pedir alteração de turno e visualizar pedidos de alteração de turno.

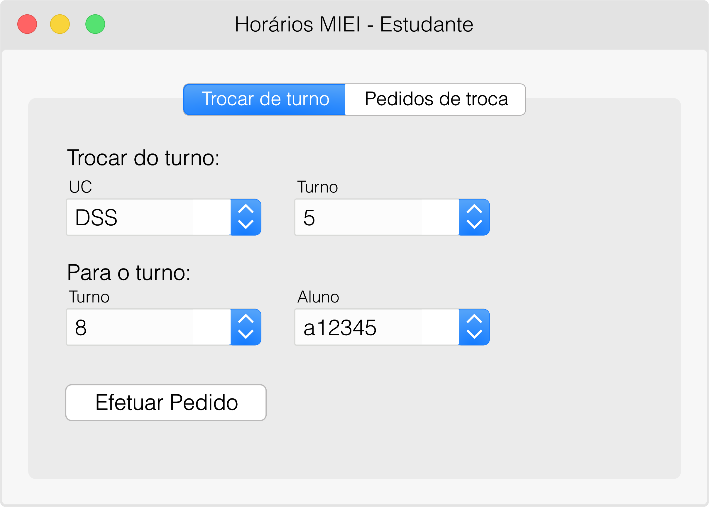
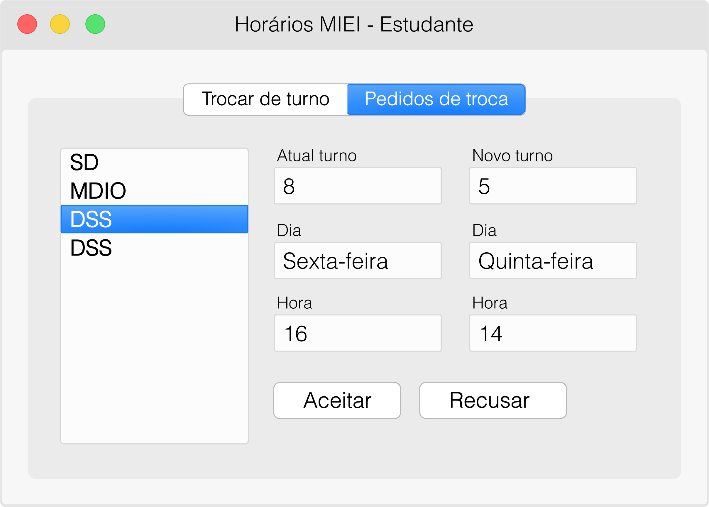


Fig. 3 – Janela do Estudante

Trabalhador estudante

A janela do Trabalhador estudante irá ter 2 separadores para cada função permitida a esse utilizador: Efetuar uma alteração de turno e visualizar pedidos de alteração de turno.

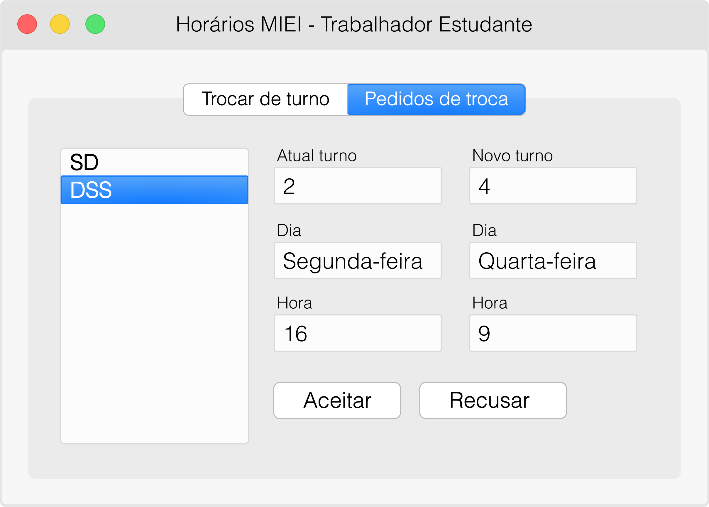
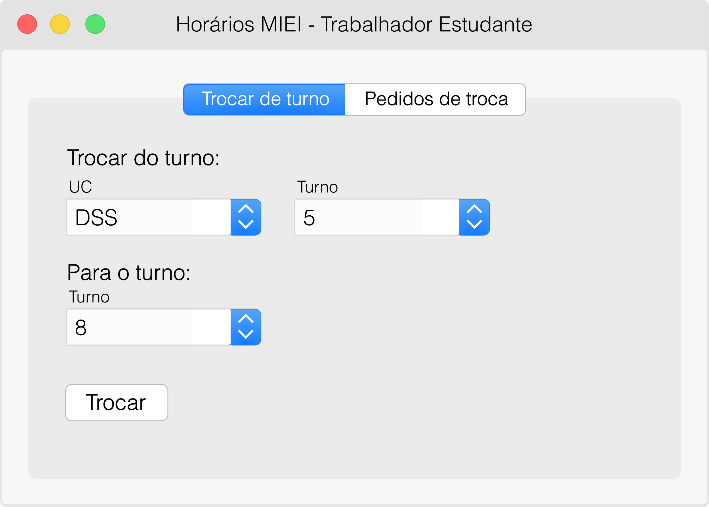


Fig. 4 – Janela do Trabalhador estudante

Outras

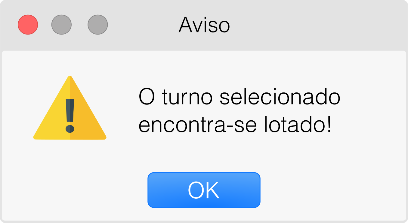


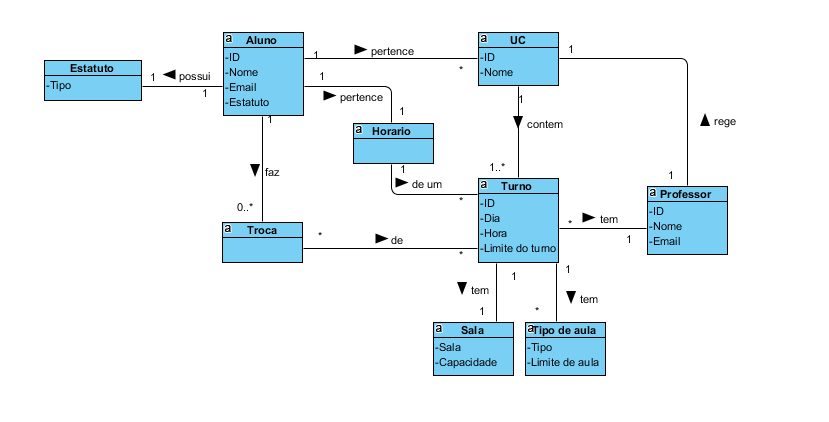
Fig. 5 – Janela de aviso

Conclusão

Apesar de estarmos numa fase inicial do projeto, com a criação dos modelos (de domínio e de use case) é-nos permitido ter uma melhor visão de como a aplicação vai ser elaborada em termos de classes (com a ajuda do modelo de domínio) e das funções necessárias (com a ajuda do use cases). Assim estes modelos vão servir como guia na elaboração da aplicação.

Apêndice

Modelo de Domínio



Modelo de Use Case

