



UNIVERSIDADE DO MINHO

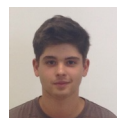
MESTRADO INTEGRADO DE ENGENHARIA INFORMÁTICA

## Desenvolvimento de Sistemas de Software

### Autores:

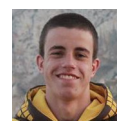
Frederico Pinto

A73639



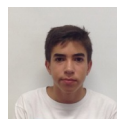
João Vieira

A76516



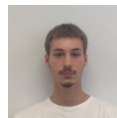
José Sousa

A74678



Ricardo Leal

A75411



18 de Dezembro de 2017

## 1 Introdução

Hoje em dia, vemos muitos cursos universitários com grande número de alunos, o que complica a gestão de horários. No âmbito da unidade curricular de Desenvolvimento de Sistemas de Software, foi nos proposto como trabalho prático desenvolver um sistema de gestão dos turnos práticos de um curso (Licenciatura ou Mestrado Integrado).

Com esta aplicação podemos armazenar os alunos e as UCs de um dado curso, e para cada aluno, gerar os respetivos turnos das suas UCs, tendo em conta que um aluno não deverá ter aulas sobrepostas, a não ser que seja inevitável.

Os alunos, poderão ainda, fazer trocas de turnos entre eles, sendo que os que tenham estatuto especial não necessitam de outro aluno para efetuar a troca.

## 2 Análise de Requisitos

Após analisar o enunciado, tivemos de identificar o que seria importante para o programa poder cumprir com todos os requisitos. Primeiro, a aplicação a ser desenvolvida deverá suportar o registo das Unidades Curriculares, dos alunos e a gestão da alocação dos alunos nos turnos das unidades curriculares. Posteriormente a alocação dos alunos deverá ser responsabilidade da Direcção de Curso, contudo deverá existir uma fase em que os alunos podem realizar trocas entre eles. Aos alunos com estatuto especial é dada a possibilidade de mudarem de turno, sem necessidade de efetuarem trocas com outros alunos inscritos. A capacidade de cada turno depende da sala em que o turno é lecionado e do tipo de turno (turnos práticos e práticos-laboratoriais têm limites máximos de alunos, definidos pelo docente responsável da UC). Após o início das aulas entra-se numa terceira fase, em que se faz a gestão dos turnos. Nesta fase, apenas o docente responsável da UC pode alterar a composição dos turnos. Caso um aluno falte a um quarto das aulas, perde o seu lugar no turno. Para além disso, o sistema deverá ser robusto procurando evitar que os alunos tenham aulas sobrepostas no horário; mas também flexível, por exemplo, sendo capaz de lidar com situações em que a sobreposição é inevitável, permitindo que trocas/mudanças efectuadas sejam anuladas ou que, pontualmente, um aluno possa assistir a um turno que não o seu. Deverá ser disponibilizada informação sobre a evolução dos turnos ao longo do semestre.

Para iniciar este projeto iremos, neste relatório, primeiramente apresentar o modelo de domínio proposto para a resolução deste problema.

Posteriormente, iremos apresentar o diagrama de use cases, e respetivas especificações, que representa as várias ações que os diversos utilizadores realizam no sistema.

Por fim, iremos terminar com uma conclusão.

### 3 Modelo de Domínio

O modelo de domínio ilustra as classes conceituais significativas no domínio do problema, sem representar as componentes do software.

Assim, este modelo contém todas as entidades envolvidas no sistema como também os relacionamentos existentes entre estas.

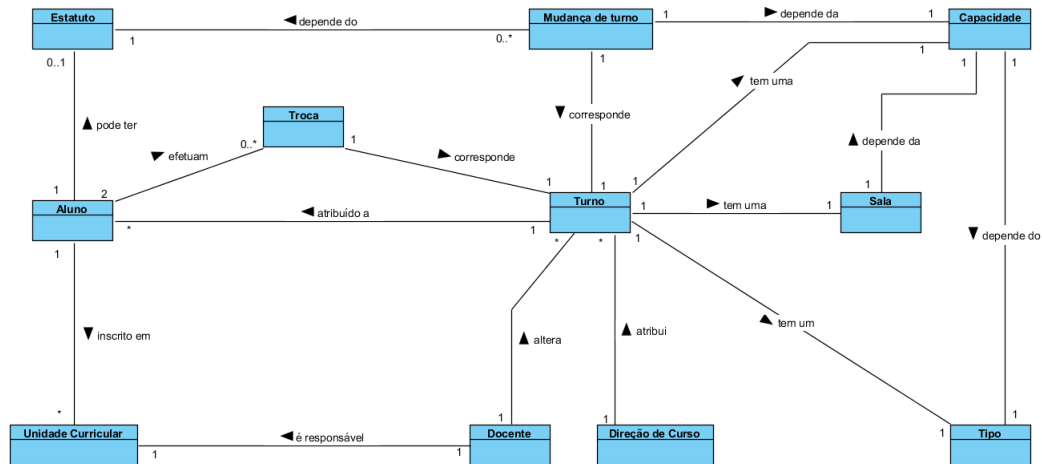


Figura 1: Modelo de domínio

Depois de analisar o modelo de domínio podemos concluir que este tem as seguintes entidades principais:

- Aluno
- Docente
- Direção de curso
- Turno

#### 3.1 Aluno

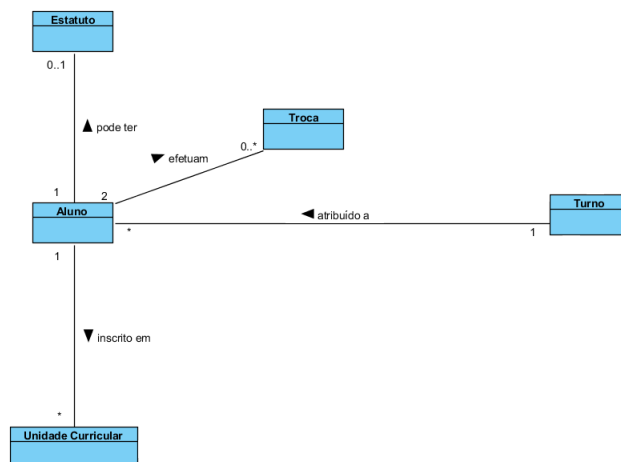


Figura 2: Porção do modelo de domínio correspondente ao Aluno

### 3.2 Docente

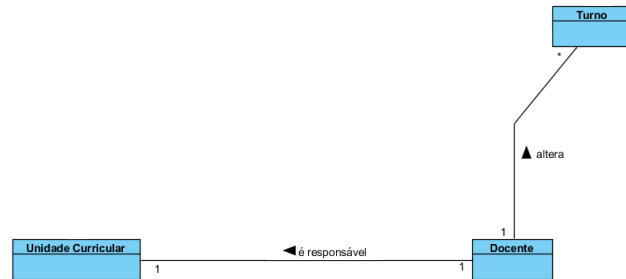


Figura 3: Porção do modelo de domínio correspondente ao Docente

### 3.3 Direção de curso

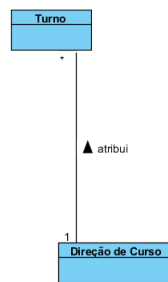


Figura 4: Porção do modelo de domínio correspondente à Direção de Curso

### 3.4 Turno

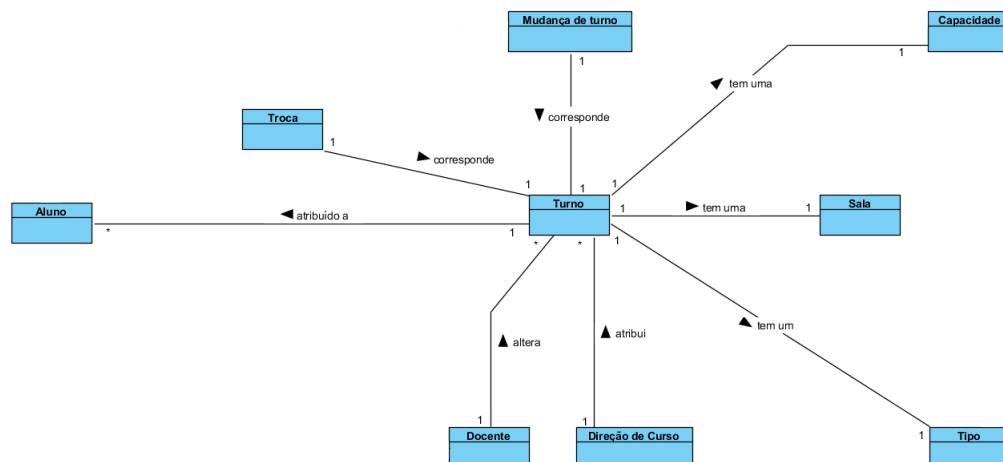


Figura 5: Porção do modelo de domínio correspondente ao Turno

## 4 Diagrama dos Use Cases

O diagrama de Use Cases concetualiza as principais funcionalidades entre o utilizador e o sistema em questão. Este diagrama, tendo por base o modelo de domínio, descreve as principais ações dos utilizadores do sistema que neste caso são, o aluno, o docente e a direção de curso.

Sendo assim, as funcionalidades deste modelo de use cases são:

- Registo
- Autenticação
- Insere aluno num turno
- Marcação de falta
- Retira aluno de turno
- Alocar turnos
- Mudança de turno
- Pedir troca
- Anular troca

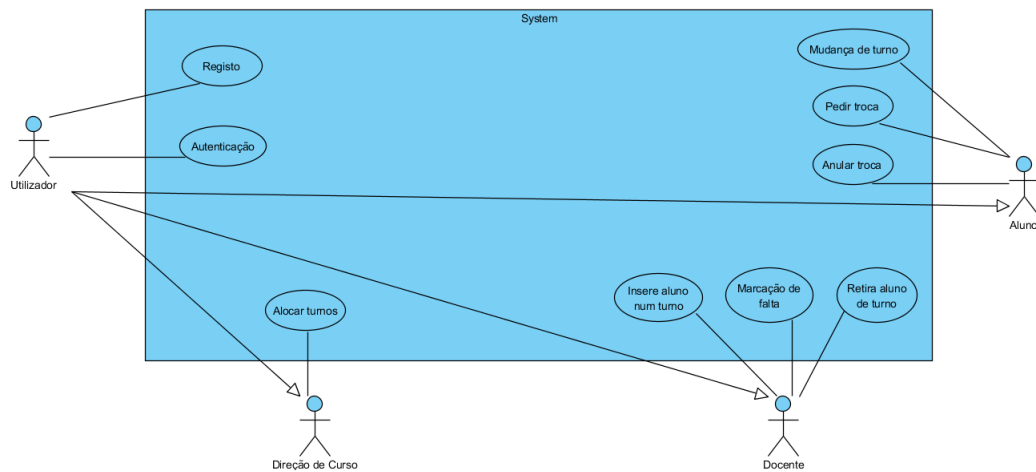


Figura 6: Diagrama de use cases

## 5 Especificações de Use Cases

### 5.1 Registo

<b>Use Case:</b>	Registo		
<b>Descrição:</b>	Registo na aplicação		
<b>Pré-condição:</b>	Utilizador não ter registo		
<b>Pós-condição:</b>	Utilizador fica registado na aplicação		
<b>Comportamento Normal</b>		Ator	Sistema
	1	Inserir email e password	
	2		Valida dados
	3		Mostra menu de tipo de utilizador
	4	Escolhe um tipo de utilizador	
	5		Valida dados
	6		Finalização do registo
<b>Comportamento Alternativo</b> [Dados são inválidos] (passo 2)		Ator	Sistema
	2.1		Indica invalidade dos dados
	2.2		Regressa a 1
<b>Exceção</b> [Não tem permissão] (passo 5)		Ator	Sistema
	5.1		Avisa sobre falta de permissão para escolha do utilizador

Figura 7: Especificação do use case Registo

### 5.2 Autenticação

<b>Use Case:</b>	Autenticação		
<b>Descrição:</b>	Autentica o utilizador na aplicação		
<b>Pré-condição:</b>	Utilizador não tem sessão iniciada		
<b>Pós-condição:</b>	Utilizador tem sessão iniciada		
<b>Comportamento Normal</b>		Ator	Sistema
	1	Inserir email e password	
	2		Valida dados
	3		Inicia sessão do utilizador
<b>Exceção</b> [Dados inválidos] (passo 2)		Ator	Sistema
	2.1		Indica invalidade dos dados

Figura 8: Especificação do use case Autenticação

### 5.3 Insere aluno num turno

<b>Use Case:</b>	Insere aluno num turno		
<b>Descrição:</b>	Docente insere aluno em novo turno		
<b>Pré-condição:</b>	Docente tem sessão iniciada		
<b>Pós-condição:</b>	Mudança de turno realizada		
<b>Comportamento Normal</b>		Ator	Sistema
	1	Insere email e novo turno do aluno	
	2		Valida dados
	3		Verifica se troca é possível
	4		Realiza a troca de turno
<b>Exceção [Email inválido] (passo 2)</b>		Ator	Sistema
	2.1		Indica invalidade do email
<b>Exceção [Troca impossível] (passo 3)</b>		Ator	Sistema
	3.1		Indica impossibilidade de troca

Figura 9: Especificação do use case Insere aluno num turno

### 5.4 Marcação de falta

<b>Use Case:</b>	Marcação de falta		
<b>Descrição:</b>	Adiciona falta a aluno		
<b>Pré-condição:</b>	Docente tem sessão iniciada		
<b>Pós-condição:</b>	Registo da falta		
<b>Comportamento Normal</b>		Ator	Sistema
	1	Escolhe unidade curricular	
	2		Pede email e turno do aluno
	3	Insere email e turno	
	4		Valida dados
	5		Regista falta
	6		Verifica percentagem de faltas do aluno
<b>Exceção [Dados inválidos] (passo 4)</b>		Ator	Sistema
	4.1		Indica invalidade dos dados
<b>Comportamento Alternativo [25% Faltas dadas] (passo 6)</b>		Ator	Sistema
	6.1		Indica que aluno tem 25% de faltas

Figura 10: Especificação do use case Marcação de falta

## 5.5 Retira aluno de turno

<b>Use Case:</b>	Retira aluno de turno		
<b>Descrição:</b>	Docente retira aluno de um turno		
<b>Pré-condição:</b>	Docente tem sessão iniciada		
<b>Pós-condição:</b>	Aluno retirado do turno		
<b>Comportamento Normal</b>		Ator	Sistema
	1	Escolhe unidade curricular	
	2		Pede email do aluno e turno a retirar
	3	Inserir email e turno	
	4		Valida dados
	5		Retira aluno do turno
<b>Exceção [Dados inválidos] (passo 4)</b>		Ator	Sistema
	4.1		Indica invalidade dos dados

Figura 11: Especificação do use case Retira aluno de turno

## 5.6 Alocar turnos

<b>Use Case:</b>	Alocar turnos		
<b>Descrição:</b>	Direção aloca aluno a turno		
<b>Pré-condição:</b>	Direção de curso tem sessão iniciada		
<b>Pós-condição:</b>	Aluno alocado em turno		
<b>Comportamento Normal</b>		Ator	Sistema
	1	Escolhe unidade curricular	
	2	Escolhe turno	
	3		Pede email do aluno
	4	Inserir email do aluno	
	5		Valida dados
	6		Regista aluno no turno
<b>Exceção [Dados inválidos] (passo 5)</b>		Ator	Sistema
	5.1		Indica invalidade dos dados

Figura 12: Especificação do use case Alocar turnos



## 5.7 Mudança de turno

<b>Use Case:</b>	Mudança de turno		
<b>Descrição:</b>	Aluno muda de turno		
<b>Pré-condições:</b>	Aluno tem sessão iniciada e tem estatuto		
<b>Pós-condições:</b>	Mudança de turno realizada		
<b>Comportamento Normal</b>		Ator	Sistema
	1	Escolhe unidade curricular	
	2		Apresenta turnos
	3	Escolhe turno pretendido	
	4		Verifica estatuto
	5		Verifica se é possível realizar mudança
	6		Realiza mudança de turno
<b>Exceção [Estatuto inválido] (passo 4)</b>		Ator	Sistema
	4.1		Indica estatuto inválido do aluno
<b>Exceção [Mudança de turno Indisponível] (passo 5)</b>		Ator	Sistema
	5.1		Indica impossibilidade de efetuar mudança

Figura 13: Especificação do use case Mudança de turno

## 5.8 Pedir troca

<b>Use Case:</b>	Pedir troca		
<b>Descrição:</b>	Aluno realiza troca de turno		
<b>Pré-condição:</b>	Aluno tem sessão iniciada		
<b>Pós-condição:</b>	Troca de turno realizada		
<b>Comportamento Normal</b>		Ator	Sistema
	1	Insere turno atual, turno pretendido e a unidade curricular respetiva	
	2		Verifica se troca é possível
	3		Realiza troca
<b>Comportamento Alternativo [Troca impossível] (passo 2)</b>		Ator	Sistema
	2.1		Indica impossibilidade de troca
	2.2		Regista pedido de troca

Figura 14: Especificação do use case Pedir troca

## 5.9 Anular troca

Use Case:	Anular troca		
Descrição:	Aluno remove pedido de troca		
Pré-condição:	Aluno tem sessão iniciada		
Pós-condição:	Pedido de troca removido		
Comportamento Normal		Ator	Sistema
	1	Insere dados do pedido de troca	
	2		Valida dados
	3		Remove pedido de troca
Exceção [Dados inválidos] (passo 2)		Ator	Sistema
	2.1		Indica invalidade de dados

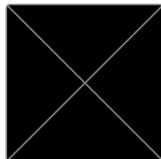
Figura 15: Especificação do use case Anular troca

## 6 Proposta de Interface

### 6.1 Menu Iniciar Sessão

#### SwapClass

Bem vindo!



Email:

Password:

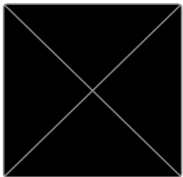
Entrar

Criar conta?

Registo

Figura 16: Menu 1

## 6.2 Menus Registrar



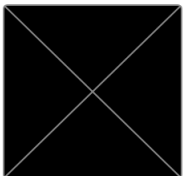
**Registo:**

Email:

Password:

**Seguinte**

Figura 17: Menu 2



**Registo:**

☐ Docente

☐ Aluno

☐ Direção de Curso

**Seguinte**

Figura 18: Menu 3

### 6.3 Menu Docente

**Menu Docente**

#### **Manutenção de Turnos**



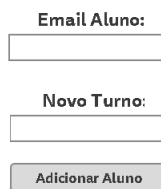
Three buttons are displayed vertically, each with a light gray background and a thin black border. The top button is labeled "Adicionar Aluno", the middle button is labeled "Remover Aluno", and the bottom button is labeled "Registrar faltas".

Figura 19: Menu 4

### 6.4 Menu em que o Docente consegue adicionar um Aluno a um Turno

**Menu Docente**

**Adicionar aluno:**



A form for adding a student. It consists of two text input fields and one button. The first input field is labeled "Email Aluno:" and the second is labeled "Novo Turno:". Below the second input field is a button labeled "Adicionar Aluno".

Figura 20: Menu 5

## 6.5 Menu em que o Docente consegue remover um Aluno de um Turno

**Menu Docente**

**Remover aluno:**

Email Aluno:

Turno:

Figura 21: Menu 6

## 6.6 Menu em que o Docente consegue marcar falta a um Aluno

**Menu Docente**

**Marcar falta:**

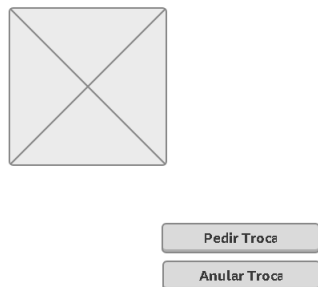
Email Aluno:

Turno:

Figura 22: Menu 7

## 6.7 Menu Aluno

### Menu Aluno



A square button with a light gray background and a black border, containing a black 'X' formed by two diagonal lines. Below the button are two smaller, rounded rectangular buttons with light gray backgrounds and black borders. The top button is labeled 'Pedir Troca' and the bottom button is labeled 'Anular Troca'.

Figura 23: Menu 8

## 6.8 Menu em que o Aluno consegue pedir uma troca de turno

### Menu Aluno

#### Pedir Troca:

Turno atual:

Turno desejado:

- ☐ PF
- ☐ Calculo
- ☐ Algebra
- ☐ EES
- ☐ TFM
- ☐ Lh
- ☐ ISD
- ☐ PI
- ☐ DSS
- ☐ SD
- ☐

Figura 24: Menu 9

## 6.9 Menu em que o Aluno consegue anular uma troca efetuada anteriormente

### Menu Aluno

#### Anular Troca:

Email:

Turno atual:

Turno desejado:

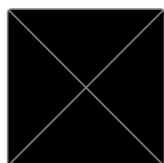
☐ PF  
☐ Calculo  
☐ Algebra  
☐ EES  
☐ TFM  
☐ Lh  
☐ ISD  
☐ PI  
☐ DSS  
☐ SD  
☐

Anular Troca

Figura 25: Menu 10

## 6.10 Menu Aluno com estatuto

### Menu Aluno com Estatuto



Pedir Troca

Anular Troca

Figura 26: Menu 11

## 6.11 Menu em que o Aluno com estatuto consegue pedir uma troca de turno

### Menu Aluno com Estatuto

#### Unidades Curriculares:

- ☐ PF
- ☐ Calculo
- ☐ Algebra
- ☐ EES
- ☐ TFM
- ☐ Un
- ☐ ISD
- ☐ PI
- ☐ DSS
- ☐ SD
- ☐

Seguinte

Figura 27: Menu 12

### Menu Aluno com Estatuto

#### Turnos:

- ☐ TP1
- ☐ TP2
- ☐ TP3
- ☐ TP4
- ☐ TP5
- ☐

Pedir Troca

Figura 28: Menu 13



## 6.12 Menu em que o Aluno com estatuto consegue anular uma troca efetuada anteriormente

### Menu Aluno com Estatuto

#### Anular Troca:

Email:

Turno atual:

Turno desejado:

☐ PF  
☐ Calculo  
☐ Algebra  
☐ EES  
☐ TFM  
☐ Lh  
☐ ISD  
☐ PI  
☐ DSS  
☐ SD  
☐

Anular Troca

Figura 29: Menu 14

## 6.13 Menu Direção de Curso para manutenção de turnos

### Menu Direção de Curso

#### Manutenção de Turnos

☐ PF  
☐ Calculo  
☐ Algebra  
☐ EES  
☐ TFM  
☐ Lh  
☐ ISD  
☐ PI  
☐ DSS  
☐ SD  
☐

Seguinte

Figura 30: Menu 15

## Menu Direção de Curso

### Manutenção de Turnos

Email Aluno:

- ☐ TP1
- ☐ TP2
- ☐ TP3
- ☐ TP4
- ☐ TP5

Inserir Aluno

Figura 31: Menu 16

## 7 Conclusão

Numa parte inicial deste projeto tivemos que fazer uma análise de requisitos onde verificamos todos os pormenores sobre o problema apresentado de maneira a perceber sobre o funcionamento do sistema. Após isso foi idealizado o modelo de dominio. Posteriormente abordamos os requisitos funcionais do sistema desenvolvendo o diagrama de Use Case onde foram identificados os atores do sistema e as suas ações, depois de todos os Use Case definidos, fizemos a sua especificação para perceber melhor a relação entre o utilizador e o sistema. Por fim, fizemos esboços da interface da nossa aplicação de modo a idealizarmos o nosso projeto.