

**Universidade do Minho**

**Escola de Engenharia**

**Desenvolvimento de Sistemas de Software**

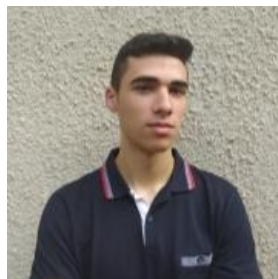
**Trabalho Prático – Fase 1**

Grupo 4

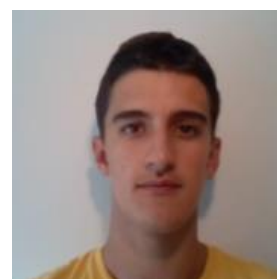
Link GitHub: <https://github.com/LEI-DSS/DSS2425-Grupo-04>



Gonçalo Cunha  
A104003



Gonçalo Cruz  
A104346



Nuno Ribeiro  
A104177

# Índice

Introdução.....	3
Modelo de Domínio .....	4
Modelo de Use Case.....	6
Conclusão .....	15

# Índice de Figuras

Figura 1 - Modelo de Domínio .....	4
Figura 2 - Modelo de Use Case.....	6

# Introdução

Este relatório foi desenvolvido no âmbito da UC de Desenvolvimento de Sistemas de Software. A gestão de horários e turnos num contexto académico representa um problema que requer um cuidado especial, dado o grande volume de dados e as diversas restrições envolvidas, como a capacidade máxima de cada turno e a distribuição equitativa dos alunos.

O objetivo deste projeto é a criação de uma plataforma que automatize a alocação de horários aos alunos, respeitando as preferências impostas pelos regentes de cada UC e otimizando a distribuição dos estudantes.

A plataforma deverá ser capaz de processar uma lista de alunos e um conjunto de horários previamente definidos, atribuindo os turnos de maneira eficiente, evitando sobrecargas e garantindo uma distribuição justa e funcional.

Este relatório apresenta o Modelo de Domínio (que fornece uma visão estrutural dos elementos principais) e o Modelo de Use Case (que descreve os cenários de interação entre os utilizadores e o sistema) desenvolvidos para representar as funcionalidades e as interações que a plataforma deverá contemplar.

## Modelo de Domínio

O modelo de domínio apresentado na figura abaixo serve como uma representação abstrata e concisa do problema, focando-se nas principais entidades e nas suas relações, sem propor soluções técnicas. No caso deste projeto, o modelo desenvolvido (figura 1) serve para organizar os elementos envolvidos na distribuição de horários aos alunos, tendo em consideração os turnos e todas as restrições.

O objetivo é identificar as entidades que descrevem o problema, como alunos, horários e turnos, e os seus relacionamentos, permitindo uma visão clara das dependências e restrições que o sistema deve considerar. Este modelo oferece uma base sólida para compreender a natureza do problema e para o desenvolvimento posterior de soluções.

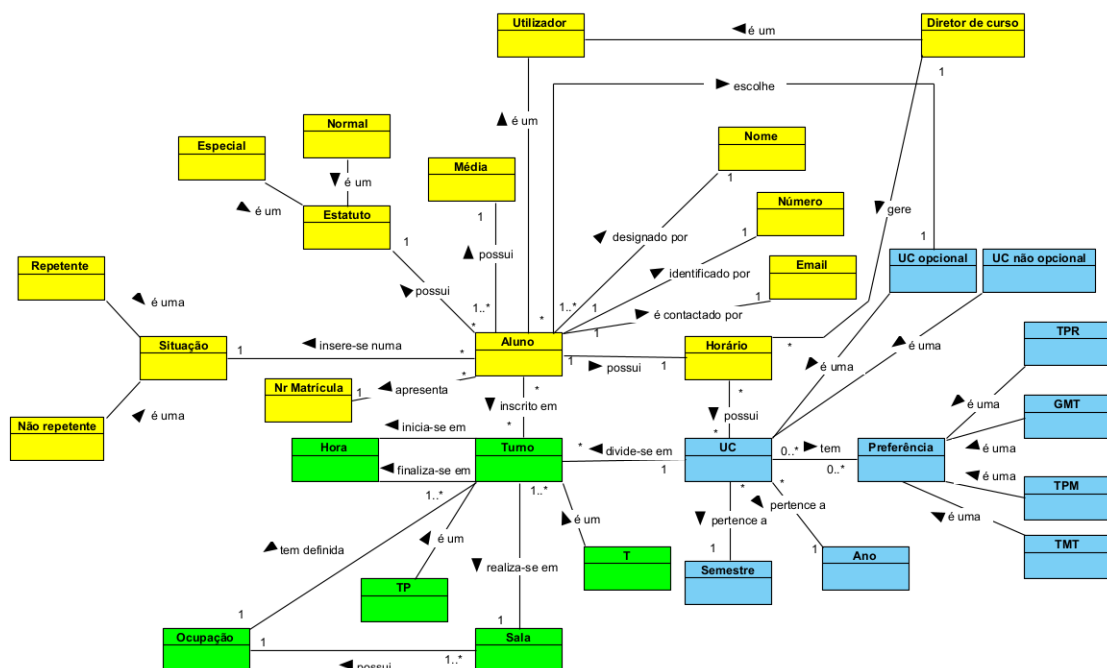


Figura 1 - Modelo de Domínio

Tivemos o cuidado de separar as entidades principais através das três cores exibidas na imagem. Entidades essas que são: aluno, turno e UC.

**Aluno:** é um utilizador, que possui um nome, número, email, média e estatuto (pode ser normal ou então especial, referente a trabalhadores-estudantes e atletas). Caracteriza-se também pela situação em que se encontra (se é aluno repetente ou não) e pelo número da sua matrícula (1ª matrícula significa que é aluno de 1º ano, e assim adiante). Cada aluno tem também um horário individual (que é gerido pelo diretor de curso).

**Turno:** os turnos de cada UC estão divididos em turno Teórico (T) ou turno Prático (TP). Todos eles têm uma hora de início e uma hora de fim e uma ocupação máxima, que pode ser a ocupação máxima da sala onde este decorre, ou então inferior à mesma.

**UC:** cada Unidade Curricular pertence a um semestre e ano específico. Esta pode ainda ser uma UC opcional (em que o aluno tem direito a escolher qual deseja frequentar) e uma UC não opcional. Em cada uma delas, o seu regente pode ou não indicar uma(s) preferência(s), entre as quais TPR (turno para repetentes - alunos repetentes são colocados em turnos distintos dos alunos de primeira inscrição), GMT (grupo do mesmo turno - os elementos de cada grupo têm de ficar no mesmo turno TP), TPM (turno com proximidade de média - alunos são distribuídos pelos turnos de modo a ficarem agrupados por proximidade da média de curso), TMT (tamanho máximo de turno – o regente define tamanhos máximos para os turnos TP, diferentes do valor por omissão usado no curso).

# Modelo de Use Case

O modelo de Use Case é uma ferramenta essencial para descrever as interações entre os utilizadores e o sistema e que especifica os diferentes cenários de uso e as funcionalidades que o sistema deve suportar. No contexto deste projeto, o modelo de use case desenvolvido (figura 2) tem como principal objetivo ilustrar as principais ações que podem ser realizadas pelos utilizadores da plataforma de distribuição de horários.



Figura 2 - Modelo de Use Case

## **Use Case: Trocar turno**

**Descrição:** Um aluno com estatuto altera o seu turno previamente atribuído para um que desejar

**Cenários:** O Francisco, que é um atleta, apercebeu-se que tem uma aula de TMD na sexta-feira à tarde (dia em que tem treino de ténis), pelo que decidiu alterar esse turno para a terça-feira de manhã

**Pré-condição:** Existe(m) turno(s) com vagas, e utilizador está autenticado e não é administrador

**Pós-condição:** O aluno troca para o turno pretendido

### **Fluxo normal:**

1. O sistema pergunta qual turno o aluno pretende trocar
2. O aluno escolhe um dos seus turnos para trocar
3. O sistema valida o turno inserido (verifica que o aluno está inscrito nesse turno)
4. O sistema pede que o aluno insira o turno para o qual pretende ir
5. O aluno escolhe um turno específico em que se pretende inscrever
6. O sistema verifica se há vagas nesse turno
7. O sistema averigua se essa troca é possível ser realizada, respeitando restrições do regente
8. O sistema efetua a troca
9. O aluno passa a estar inscrito no novo turno, o que faz com que o turno inicial tenha mais uma vaga
10. O sistema atualiza o horário pessoal do aluno
11. O horário do aluno não tem sobreposições de turnos

### **Fluxo alternativo (1): [há sobreposição de turnos] (passo 11)**

- 11.1. O horário do aluno tem sobreposições de turnos
- 11.2. O sistema emite uma notificação para o aluno a indicar que existe sobreposição

### **Fluxo de exceção (2): [o aluno não está inscrito nesse turno] (passo 3)**

- 3.1. O sistema verifica que o aluno não está no turno indicado
- 3.2. O sistema emite uma notificação para o aluno a referir que a troca não foi efetuada
- 3.3. O horário do aluno não é alterado

**Fluxo de exceção (3):** [não há vagas] (passo 6)

- 6.1. O sistema verifica que a troca não é possível porque não há vagas nesse turno
- 6.2. O sistema emite uma notificação para o aluno a referir que a troca não foi efetuada
- 6.3. O horário do aluno não é alterado

**Fluxo de exceção (4):** [não respeita restrições] (passo 7)

- 7.1. O sistema verifica que a troca não é possível porque não respeita as restrições da UC
- 7.2. O sistema emite uma notificação para o aluno a referir que a troca não foi efetuada
- 7.3. O horário do aluno não é alterado

**Use Case:** Escolher UC opcional

**Descrição:** Um aluno escolhe uma das UCs opcionais que estão disponíveis

**Cenários:** O Francisco, que é um aluno de primeiro ano, escolhe a UC “Matemática das Coisas” para sua UC Opção Uminho

**Pré-condição:** Existe(m) UC(s) opcional(is) com vagas e utilizador está autenticado e não é administrador

**Pós-condição:** O aluno fica inscrito na UC escolhida

**Fluxo normal:**

- 1. O sistema apresenta a lista das UCs opcionais e pergunta em qual o aluno se deseja inscrever
- 2. O aluno escolhe a UC opcional que pretende
- 3. O sistema valida a UC
- 4. A inscrição do aluno na UC é efetivada

**Fluxo de exceção (2):** [não existem vagas] (passo 3)

- 3.1. O sistema emite uma mensagem a indicar que não existem vagas na UC que o aluno escolheu
- 3.2. O aluno fica sem UC escolhida



**Use Case:** Ver horário

**Descrição:** O aluno consulta o horário

**Cenários:** O Francisco consulta o horário

**Pré-condição:** Existe um horário para cada ano do curso, e utilizador está autenticado e não é administrador

**Pós-condição:** O aluno conseguiu ter acesso ao horário

**Fluxo normal:**

1. O sistema pergunta se o utilizador deseja ver o horário pessoal ou o geral
2. O aluno seleciona o seu horário pessoal
3. O sistema verifica se o horário pessoal do aluno está disponível
4. O aluno observa o seu horário

**Fluxo alternativo (1):** [consulta de horário geral] (passo 2)

- 2.1. O sistema solicita o ano do horário a consultar
- 2.2. O aluno seleciona o ano
- 2.3. O sistema apresenta o horário geral ao aluno
- 2.4. O aluno observa o horário do ano indicado

**Fluxo de exceção (2):** [aluno não tem horário pessoal] (passo 3)

- 3.1. O sistema emite uma notificação a indicar que o aluno ainda não tem um horário
- 3.2. O aluno não consegue consultar o horário

#### **Use Case:** Gerir conflitos

**Descrição:** O diretor resolve incongruências manualmente para os casos que as possuírem, casos esses que são informados pelo sistema

**Cenários:** O diretor de curso António, após a geração automática dos horários, verificou que ocorreram conflitos. Alguns desses conflitos envolveram sobreposição de turnos, outros estavam relacionados com alunos que não estavam inscritos em nenhum turno, para uma ou mais cadeiras

**Pré-condição:** Utilizador está autenticado como administrador e os horários estão criados

**Pós-condição:** Conflitos para alunos de primeira inscrição ficam garantidamente resolvidos

#### **Fluxo normal:**

1. O sistema verifica que ocorreram problemas na geração dos horários
2. O sistema apresenta uma notificação a informar sobre esses problemas
3. O diretor de curso resolve as inconsistências manualmente ajustando os limites de alunos dos turnos
4. Os problemas dos horários dos alunos de primeira inscrição ficam garantidamente resolvidos

#### **Use Case:** Criar horários

**Descrição:** O diretor pede ao sistema para gerar automaticamente horários para os alunos

**Cenários:** O António, diretor de curso, solicita ao sistema que este gere os horários

**Pré-condição:** Utilizador está autenticado como administrador e UCs, salas, alunos e turnos estão criados

**Pós-condição:** Os horários são criados

#### **Fluxo normal:**

1. O diretor de curso seleciona a opção de gerar os horários individuais de cada aluno
2. O sistema gera os horários

## **Use Case: Criar UCs**

**Descrição:** O diretor cria uma UC, associando-lhe as respetivas preferências do regente

**Cenários:** O diretor de curso, António, faz corresponder à UC “EPTN” a preferência GMT (imposta pelo regente da mesma)

**Pré-condição:** Utilizador está autenticado e é administrador

**Pós-condição:** A UC é criada, incluindo as preferências definidas pelo regente da mesma

### **Fluxo normal:**

1. O sistema solicita o nome da UC a criar
2. O diretor de curso insere o nome
3. O sistema valida a UC (que ainda não existe nenhuma UC com esse nome)
4. O sistema apresenta as preferências de UC disponíveis e solicita quais é que vão ser adicionadas
5. O diretor escolhe as preferências, conforme previamente fornecidas pelo regente da UC
6. O sistema valida as preferências inseridas
7. O sistema cria a UC

### **Fluxo de exceção (1): [a UC já existe] (passo 3)**

- 3.1. O sistema apresenta uma mensagem que informa que a UC já existe
- 3.2. A UC não é criada

### **Fluxo de exceção (2): [as preferências não são válidas] (passo 6)**

- 6.1. O sistema informa que as preferências não foram corretamente inseridas
- 6.2. A UC não é criada

**Use Case:** Criar salas

**Descrição:** O diretor de curso cria uma sala, definindo a ocupação máxima da mesma

**Cenários:** O diretor de curso, António, define a nova sala “2-1.12” com uma ocupação de 32 lugares

**Pré-condição:** Utilizador está autenticado e é administrador

**Pós-condição:** A sala é criada, com a ocupação associada

**Fluxo normal:**

1. O sistema solicita a inserção do número da sala, no formato <N.º edifício>-<N.º da sala>
2. O diretor de curso adiciona o número da sala
3. O sistema valida a sala (que não existe e que o formato está correto)
4. O sistema pede a ocupação dessa sala
5. O diretor insere a ocupação
6. O sistema valida a ocupação inserida
7. A sala é criada

**Fluxo de exceção (1):** [a sala não é válida] (passo 3)

- 3.1. O sistema emite uma notificação indicando que a sala inserida não é válida
- 3.2. A sala não é criada

**Fluxo de exceção (2):** [a ocupação não é válida] (passo 6)

- 6.1. O sistema apresenta uma notificação que informa que a ocupação não é correta
- 6.2. A sala não é criada

**Use Case:** Criar alunos

**Descrição:** O diretor de curso cria um aluno, inserindo os dados do mesmo

**Cenários:** O António (o diretor de curso) cria o aluno Martim, inserindo o seu nome, número, email, média, número de matrícula, estatuto e situação

**Pré-condição:** Utilizador está autenticado e é administrador

**Pós-condição:** O aluno fica criado no sistema

**Fluxo normal:**

1. O sistema solicita os dados do aluno
2. O diretor de curso fornece os dados ao sistema
3. O sistema valida os mesmos
4. O aluno é criado

**Fluxo de exceção (1):** [dados inválidos] (passo 3)

- 3.1. O sistema notifica o utilizador de que os dados não são válidos
- 3.2. O aluno não é criado

## **Use Case:** Definir turnos

**Descrição:** O diretor de curso cria um turno, indicando qual a UC ao qual este se refere, e definindo o dia da semana e a hora no qual se vai realizar e ainda a sala.

**Cenários:** O António, diretor de curso, cria o turno TP2 para a cadeira de CC, que se vai realizar das 11:00 às 13:00, terça-feira na sala “2-1.12”.

**Pré-condição:** As UCs e as salas estão criadas

**Pós-condição:** O turno é criado e é adicionado ao horário geral do ano da UC

### **Fluxo normal:**

1. O sistema solicita a UC do turno a ser criado
2. O diretor insere o nome da UC
3. O sistema solicita a inserção das horas inicial e final de realização desse turno
4. O diretor insere a hora inicial e final
5. O sistema valida as horas
6. O sistema solicita que a sala seja inserida
7. O diretor de curso atribui uma sala para esse turno
8. O sistema valida essa sala, verificando que não está ocupada no horário definido
9. O turno é criado

### **Fluxo de exceção (1):** [as horas são inválidas] (passo 5)

- 5.1. O sistema emite uma notificação que informa que as horas não são válidas
- 5.2. O turno não é criado

### **Fluxo de exceção (2):** [a sala não é válida] (passo 8)

- 8.1. O sistema emite uma mensagem que informa que a sala inserida está ocupada no horário indicado
- 8.2. O turno não é criado

## Conclusão

Neste trabalho, foram desenvolvidos e apresentados o modelo de domínio e o modelo de use case, ambos essenciais para a compreensão e análise do problema proposto. O modelo de domínio permitiu identificar e estruturar as principais entidades e os seus relacionamentos, oferecendo uma visão clara dos elementos que compõem o sistema e dos seus relacionamentos. Já o modelo de use case focou-se em representar as interações entre os utilizadores e o sistema, destacando as funcionalidades essenciais que a aplicação deve oferecer.

Embora estes modelos não resolvam o problema em si, constituem uma base sólida para o desenvolvimento futuro da aplicação, proporcionando uma compreensão clara do problema e das operações esperadas. A partir destes, será possível avançar para a implementação de soluções técnicas que satisfaçam os requisitos.