

Universidade do Minho
Escola de Engenharia

DESENVOLVIMENTO DE SISTEMAS DE SOFTWARE

TRABALHO PRÁTICO

Grupo 7

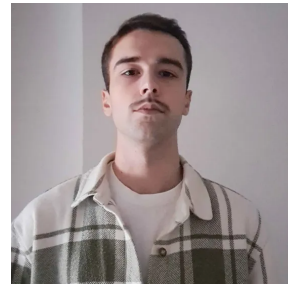
Link Github: <https://github.com/LEI-DSS/DSS2425-Grupo-07>



Rui Ribeiro a100816



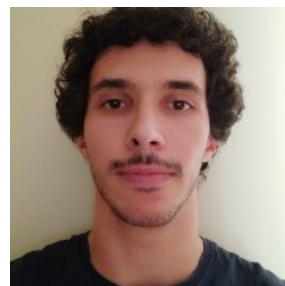
Gonçalo Gonçalves
a100833



Fábio Leite a100902



Carlos Costa a88551



João Vale a93219

Conteúdo

1	Introdução	4
2	Resultados obtidos	5
3	Modelo de Domínio	6
3.1	Entidades e suas definições	7
4	Modelo de Use Case	8
4.1	Atores	9
4.2	Use Cases	9
4.2.1	Iniciar sessão (Utilizador)	9
4.2.2	Terminar sessão (Utilizador)	10
4.2.3	Importar Lista de Alunos (Diretor de curso)	10
4.2.4	Gerar Horários (Diretor de curso)	11
4.2.5	Publicar Horários dos Alunos (Diretor de curso)	11
4.2.6	Consultar Horário Pessoal (Aluno)	12
5	Arquitetura geral	13
5.1	HorariosFacade	14
5.2	UtilizadoresFacade	14
6	Implementação	15
6.1	Modelação de classes final	15
6.2	Diagramas de sequência	16
6.2.1	Importar alunos	16
6.2.2	Gerar horário	16
6.2.3	Consultar horário	17
6.2.4	Validar horário	18
6.2.5	Publicar Horários	18
6.2.6	Listar Alunos de um dado Turno	19
6.3	Diagrama de componentes	20
7	Utilização do sistema	21
7.1	Acesso ao Sistema	21
7.2	Funcionalidades do Diretor de Curso	21
7.3	Funcionalidades do Aluno	24
8	Conclusão e análise crítica dos resultados	25

Lista de Tabelas

1	Tabela de entidades atualizada para o modelo de domínio de um sistema de distribuição de turnos práticos.	7
---	---	---

Lista de Figuras

1	Diagrama do Modelo de Domínio	6
2	Modelo de Use Case	8
3	Especificação do use case 'Iniciar sessão'	9
4	Especificação do use case 'Terminar sessão'	10
5	Especificação do use case 'Importar Lista de Alunos'	10
6	Especificação do use case 'Gerar Horários'	11
7	Especificação do use case 'Publicar Horários dos Alunos'	11
8	Especificação do use case 'Consultar Horário Pessoal'	12
9	Diagrama de Packages	13
10	Diagrama referente ao HorarioFacade	15

11	Diagrama referente ao UtilizadorFacade	15
12	Diagrama referente ao UtilizadorFacade	16
13	Diagrama referente ao GerarHorarios	16
14	Diagrama referente ao newAuxAlocador	17
15	Diagrama referente ao ConsultarHorário	17
16	Diagrama referente ao ValidarHorário	18
17	Diagrama referente a publicar horários	19
18	Diagrama referente a listar alunos de um dado turno	19
19	Diagrama de componentes	20
20	Login do Diretor de Curso	21
21	Publicação de Horários	21
22	Listagem de Alunos	22
23	Listagem de Alunos com Estatuto	22
24	Definição de Preferências	23
25	Geração Automática de Horários	23
26	Alocação Manual de Alunos	23
27	Login e Consulta de Horários	24

1 Introdução

No âmbito da Unidade Curricular de Desenvolvimento de Sistemas de Software, foi-nos proposta a criação de um sistema para auxiliar na gestão de turnos práticos. Este sistema visa centralizar a distribuição de alunos pelos diferentes turnos das Unidades Curriculares (UC) do curso, de forma a minimizar colisões de horários e otimizar a organização das aulas.

Após uma discussão entre todos os elementos do grupo, chegamos a esta proposta de resolução, tendo em conta o que nos fora pedido para esta primeira fase (**modelo de domínio** e **modelo e especificação dos Use Case**).

Na primeira fase, focamo-nos na modelação do problema, de modo a documentar as decisões tomadas e a descrever os componentes principais do sistema, facilitando a sua compreensão e implementação. O objetivo foi desenvolver dois modelos fundamentais: o Modelo de Domínio, que descreve as entidades envolvidas e os seus relacionamentos, oferecendo uma visão clara e estática do problema; e o Modelo de Casos de Uso, que apresenta as funcionalidades do sistema através dos fluxos de interação dos utilizadores com o sistema.

Nesta segunda fase entramos mais na parte da implementação, desenvolvendo diagramas que nos permitiram ter consciência das interações das diversas classes com o sistema de forma a obtermos o resultado pretendido. Graças ao foco na implementação, foram efetuadas algumas mudanças nos modelos anteriormente feito, de forma a esses modelos irem de acordo com o melhor funcionamento do sistema.

2 Resultados obtidos

Depois de uma fase de análise do problema e modelação da solução a implementar, conseguimos entregar uma aplicação funcional e bem estruturada. Em termos de funcionalidades, conseguimos atender a todos os requisitos funcionais propostos para esta segunda fase.

Foram desenvolvidas funcionalidades maioritariamente para o diretor de curso mas também para a entidade aluno, sendo também implementadas funcionalidades consideradas necessárias pelo grupo (opção de login e logout).

Seguido do login inicial, ficam expostas as operações com base no tipo de funcionário que realizou o login.

O diretor de curso é permitido a diversas funcionalidades, como importar uma lista de alunos, gerar horários para esses mesmos alunos e publicá-los, consultar tanto a lista de alunos de um turno como também os alunos que apresentam estatuto.

Por parte do aluno, após ser inserido no sistema após a importação do diretor de curso, o mesmo pode fazer registo, definindo a sua 'password', ganhando acesso a ingressar o sistema. O aluno apenas consegue consultar o horário que lhe foi atribuído, só o conseguindo visualizar após a publicação deste por parte do diretor de curso.

Fora as funcionalidades do sistema, foram implementados DAOs para acesso e manipulação dos dados, garantindo que a persistência fosse realizada de maneira eficiente. Foram também utilizadas Interfaces para implementar métodos das classes e Facades para aceder a funcionalidades mais complexas.

3 Modelo de Domínio

O modelo de domínio da figura 1, baseia-se nos **requisitos do problema** e tem como principal objetivo capturar as suas **entidades e os relacionamentos** entre elas estabelecidos. De maneira a facilitar a compreensão do mesmo, este faz-se acompanhar de uma tabela com a **definição de cada entidade**. Ao longo das fases foi sofrendo algumas alterações conforme a nossa visão acerca do sistema, chegando por fim a este modelo.

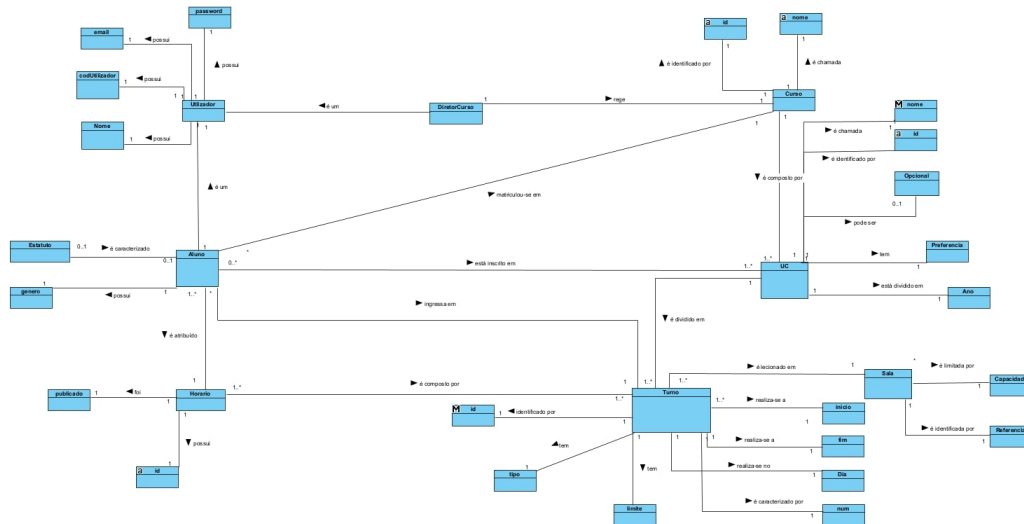


Figura 1: Diagrama do Modelo de Domínio

3.1 Entidades e suas definições

Entidade	Definição
Utilizador	Entidade genérica que representa qualquer utilizador do sistema. Identificado por um código único (<i>codUtilizador</i>), nome, e-mail e senha.
Aluno	Especialização de Utilizador que representa um estudante matriculado em um curso. Cada aluno é identificado por atributos herdados, podendo ou não possuir estatuto e pode estar inscrito em várias UCs.
Estatuto	Representa o estatuto de um aluno, podendo ser <i>trabalhador-estudante</i> ou <i>atleta</i> , que influencia a sua priorização nos turnos.
Gênero	Informação associada ao aluno, representando o gênero do mesmo.
DiretorCurso	Especialização de Utilizador responsável por coordenar e gerir o curso, incluindo horários e turnos. Está sempre associado a um único curso.
Curso	Entidade que agrupa as unidades curriculares (UCs) e alunos matriculados. Cada curso é gerido por um Diretor de Curso e identificado por um nome único.
UC (Unidade Curricular)	Componente curricular do curso, identificada por um nome e um código único. Pode ser obrigatória ou opcional, e é composta por um ou mais turnos.
Ano	Divisão do curso que agrupa as UCs e os alunos matriculados naquele ano específico. Cada UC pertence a um único ano.
Turno	Conjunto de aulas que compõem uma UC. Cada turno é identificado por um código único, está associado a um tipo (T, TP ou PL) e possui um limite de alunos.
Tipo de Turno	Classificação dos turnos em Teórica (T), Teórico-Prática (TP) ou Prática-Laboratorial (PL). Define as características e limites de capacidade do turno.
Sala	Local onde os turnos são realizados, identificada por uma referência única e limitada por uma capacidade.
Capacidade	Número máximo de alunos que uma sala pode acomodar para um turno ou aula, com base nas características do tipo de aula.
Horário	Estrutura que organiza a distribuição dos turnos ao longo dos dias e horas da semana.
Dia	Representa o dia da semana em que o turno ocorre. Faz parte do horário.
Hora de Início e Fim	Especifica o horário de início e fim de um turno no contexto do Horário.
Preferência	Representa as preferências de um aluno em relação aos turnos. Está associada a uma UC específica.
Opcional	Classificação de uma UC que indica que ela não é obrigatória para todos os alunos do curso.
Publicado	Estado do Horário que indica se ele já foi disponibilizado para os alunos.
Referência	Informação única que identifica uma sala onde os turnos ocorrem.
Limite	Restrição que define o número máximo de alunos permitido em um turno, dependendo do tipo de aula e da capacidade da sala.

Tabela 1: Tabela de entidades atualizada para o modelo de domínio de um sistema de distribuição de turnos práticos.

4 Modelo de Use Case

Na figura 2 apresentamos o **diagrama de Use Case** em que o nosso sistema assenta. Através dele conseguimos obter informação sobre os **requisitos funcionais** do mesmo, ou seja, que tipo de serviços oferece e a quem (atores do sistema).

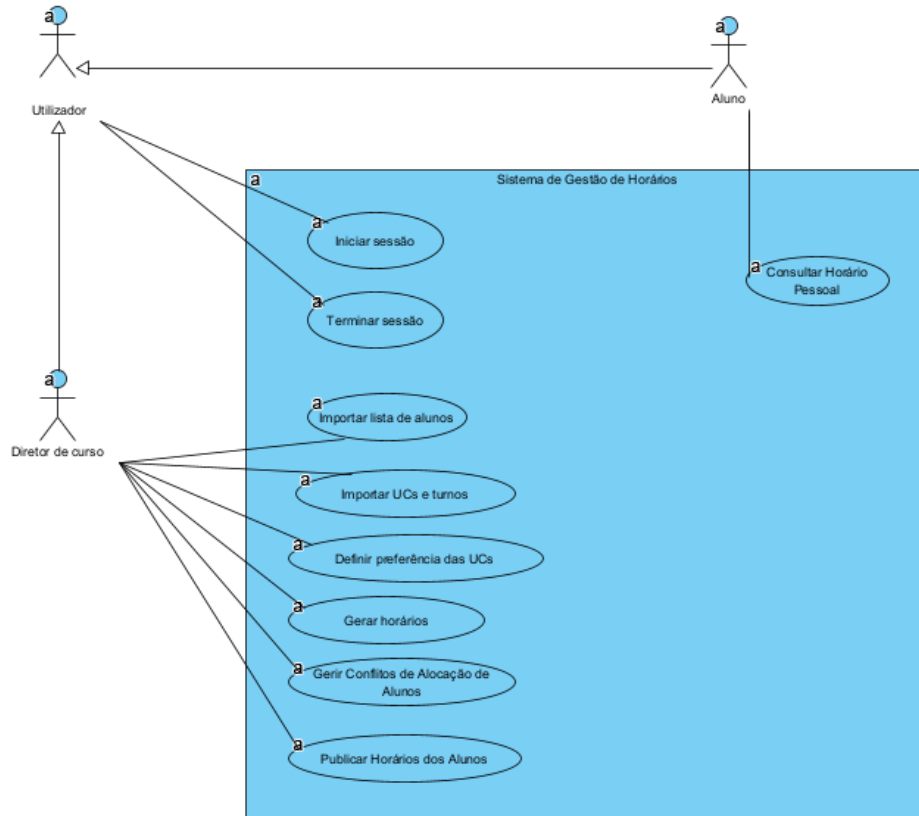


Figura 2: Modelo de Use Case

4.1 Atores

- **Utilizador**

O 'utilizador' refere-se tanto ao aluno quanto ao diretor de curso, sendo estes elementos que interagem com o sistema. Cada um possui funções específicas: o aluno consulta e exporta dados, enquanto o diretor gere o sistema.

- **Diretor**

Representa o ator do sistema cujas funcionalidades disponíveis são registar pedido de orçamento, registar entrega do equipamento, verificar disponibilidade do serviço expresso, adicionar o equipamento à lista das reparações, receber pagamento e atualizar listagens

- **Aluno**

O aluno utiliza o sistema para consultar o seu horário, que foi gerado pelo diretor de curso, e pode exportar essa informação. Ele tem funções limitadas e só acessa informações relativas aos seus turnos e aulas.

4.2 Use Cases

As especificações de Use Cases são muito importantes na descrição não só da unidade funcional de um sistema, mas também como forma de representação dos requisitos funcionais deste. Esta análise e descrição são vantajosas principalmente porque permitem um maior envolvimento no desenvolvimento do projeto em si, quer antes, quer em paralelo com os responsáveis pela sua implementação/programadores, entendendo-se melhor o contexto de cada requisito.

Segue-se o conjunto das especificações do Use Cases apresentados anteriormente, no Diagrama de Use Cases, seguidos pelo ator responsável pela sua execução.

4.2.1 Iniciar sessão (Utilizador)

Situação em que utilizador se autentica no sistema, iniciando uma sessão. Só é iniciada a sessão se as credenciais que fornece forem validadas pelo sistema. A autenticação é necessária para que o utilizador usufrua das suas permissões dentro do sistema.

USE CASE:	Iniciar sessão
DESCRIÇÃO:	O utilizador acede à aplicação de gestão de turnos e autentica-se com as suas credenciais de acesso ao sistema.
CENÁRIO:	Cenário 1 e 2
PRÉ-CONDIÇÃO:	O utilizador tem acesso à aplicação e não tem sessão iniciada no sistema.
PÓS-CONDIÇÃO:	O utilizador é autenticado na aplicação
FLUXO NORMAL	
	1. O ator indica que pretende iniciar sessão.
	2. O sistema solicita as credenciais do ator.
	3. O ator insere as respetivas credenciais.
	4. O sistema verifica as credenciais apresentadas.
	5. O sistema valida as credenciais apresentadas.
	5. O ator ganha acesso ao seu horário dentro da aplicação.
FLUXO DE EXCEÇÃO	(1) [As credenciais do ator não existem] (passo 4)
	4.1 O sistema informa que não existe nenhum utilizador com as credenciais apresentadas.
FLUXO DE EXCEÇÃO	(2) [As credenciais não correspondem] (passo 5)
	5.1 O sistema informa que as credenciais não correspondem.

Figura 3: Especificação do use case 'Iniciar sessão'

4.2.2 Terminar sessão (Utilizador)

Situação em que um utilizador pretende terminar a sessão da sua conta. Este processo faz com que o utilizador perca qualquer possível interação com o sistema, fora um novo início de sessão.

USE CASE:	Terminar sessão
DESCRIÇÃO:	O utilizador termina a sessão na aplicação de gestão de turnos
CENÁRIO:	Cenário 1 e 2
PRÉ-CONDIÇÃO:	O ator está autenticado no sistema.
PÓS-CONDIÇÃO:	O ator é desconectado do sistema.
FLUXO NORMAL:	1. O ator seleciona a opção "Terminar Sessão" no sistema. 2. O sistema confirma que o ator deseja encerrar a sessão. 3. O sistema encerra a sessão do ator.
FLUXO DE EXCEÇÃO	(1) [O ator não deseja terminar sessão] (passo 2) 2.1. O ator não confirma a ação de terminar sessão. 2.2. O sistema mantém o usuário autenticado e não encerra a sessão.

Figura 4: Especificação do use case 'Terminar sessão'

4.2.3 Importar Lista de Alunos (Diretor de curso)

Situação em que o diretor de curso importa uma lista de alunos inscritos nas unidades curriculares (UCs) para o sistema, facilitando a gestão e a alocação de alunos aos turnos.

USE CASE:	Importar Lista de Alunos
DESCRIÇÃO:	O ator importa a lista de alunos inscritos às UCs do curso, diretamente da blackboard, para a aplicação de gestão de turnos.
CENÁRIOS:	Cenário 1
PRÉ-CONDIÇÃO:	O ator está autenticado e a lista de alunos já foi obtida da blackboard
PÓS-CONDIÇÃO:	A lista de alunos foi integrada ao sistema e encontra-se disponível para qualquer operação.
FLUXO NORMAL:	1. O ator seleciona a opção de importar a lista de alunos. 2. O sistema processa a lista e insere dados no sistema.
FLUXO DE EXCEÇÃO	(1) [Lista de alunos fornecida encontra-se no formato errado] (passo 2) 2.1. O ator insere um ficheiro no formato errado. 2.2. O sistema exibe uma mensagem de erro a informar que o ficheiro é inválido.
FLUXO DE EXCEÇÃO	(2) [Lista de alunos está vazia] (passo 2) 2.1. O ator insere um ficheiro para guardar a lista de alunos. 2.2. O sistema exibe uma mensagem de erro a informar que a lista de alunos está vazia.

Figura 5: Especificação do use case 'Importar Lista de Alunos'

4.2.4 Gerar Horários (Diretor de curso)

Situação em que o diretor de curso pede ao sistema a geração de horários para os alunos, alocando-os em turnos de acordo com as preferências e restrições definidas previamente. O sistema tenta minimizar conflitos e sinaliza quando surgem problemas.

USE CASE:	Gerar Horários
DESCRIÇÃO:	O diretor solicita ao sistema que gere a alocação dos alunos aos turnos de acordo com as preferências configuradas.
CENÁRIO:	Cenário 1
PRÉ-CONDIÇÃO:	O utilizador deve estar autenticado A lista de alunos e UCs com horários já está carregada e as preferências foram definidas.
PÓS-CONDIÇÃO:	Todos os alunos foram alocados aos turnos de forma automática.
FLUXO NORMAL:	1. O ator solicita a geração automática de horários. 2. O sistema aloca os alunos aos turnos, respeitando as preferências das Ucs.
FLUXO DE EXCEÇÃO	(1) [Alocação de alunos não completa] (passo 2) 2.1. Alocação gerada, mas alguns alunos ficam sem turno. 2.2. O sistema exibe uma lista de alunos não alocados.
FLUXO DE EXCEÇÃO	(2) [Alocação de alunos com sobreposição] (passo 2) 2.1. O sistema não consegue gerar a alocação por sobreposição de horários. 2.2. O sistema apresenta a lista de alunos com sobreposição de turnos.

Figura 6: Especificação do use case 'Gerar Horários'

4.2.5 Publicar Horários dos Alunos (Diretor de curso)

Situação em que o diretor publica os horários finais dos alunos após resolver todos os conflitos de alocação. Uma vez publicados, os horários ficam disponíveis para os alunos na aplicação.

USE CASE:	Publicar Horários
DESCRIÇÃO:	O diretor de curso solicita ao sistema que publique os horários.
CENÁRIOS:	Cenário 1
PRÉ-CONDIÇÃO:	O ator deve estar autenticado e os horários já foram gerados.
PÓS-CONDIÇÃO:	Os horários são publicados com sucesso
FLUXO NORMAL:	1. O diretor seleciona a opção "Publicar Horários" no sistema 2. O sistema verifica a integridade e a validação dos horários 3. O sistema publica os horários
FLUXO ALTERNATIVO	(2) [Horários incompletos ou com erros (passo 2.)] 2.1. O sistema detecta que os horários estão incompletos ou contêm erros 2.2. O sistema exibe uma mensagem de erro ao diretor, especificando os problemas encontrados 2.3. Regressa a 3.

Figura 7: Especificação do use case 'Publicar Horários dos Alunos'

4.2.6 Consultar Horário Pessoal (Aluno)

Situação em que o aluno consulte o seu horário completo, com a lista de turnos e aulas atribuídas, que foi gerado e publicado pelo diretor de curso.

USE CASE:	Consultar Horário Pessoal
DESCRIÇÃO:	O ator consulta o seu horário na aplicação de gestão de turnos, que foi gerado e publicado pelo diretor de curso.
CENÁRIOS:	Cenário 2
PRÉ-CONDIÇÃO:	O ator deve estar autenticado e o horário do ator foi publicado.
PÓS-CONDIÇÃO:	O ator visualiza o seu horário completo e atualizado.
FLUXO NORMAL:	
	1. O ator navega para a secção de horários.
	2. O sistema exibe ao ator a opção de consultar o seu horário.
	3. O ator navega para a secção de consulta do horário.
	4. O sistema fornece o horário atribuído.
FLUXO DE EXCEÇÃO	(1) [Horário indisponível] (passo 3.)
	3.1. O horário do ator não está disponível.
	3.2. O sistema informa que o horário não está disponível.
FLUXO ALTERNATIVO	(2) [Horário sem turnos] (passo 3.)
	3.1. O horário do ator está vazio.
	3.2. Regressa para 4.

Figura 8: Especificação do use case 'Consultar Horário Pessoal '

5 Arquitetura geral

A arquitetura do sistema implementado foi dividido em três pacotes principais: UI, Business e Data, com clara separação de responsabilidades.

A camada Business contém as interfaces ISUtilizadores e ISHorarios, implementadas pelas classes UtilizadoresFacade e HorariosFacade, respectivamente. A classe UtilizadoresFacade gere os utilizadores, permitindo a importação, listagem, autenticação e registro de novos utilizadores. Já a classe HorariosFacade é responsável pela gestão de horários, incluindo validação, geração, publicação e importação de dados relacionados a cursos, turnos e unidades curriculares. A interface ISwapData no pacote Data conecta a lógica de negócio à manipulação de dados. O design modular garante flexibilidade, escalabilidade e manutenibilidade ao sistema.

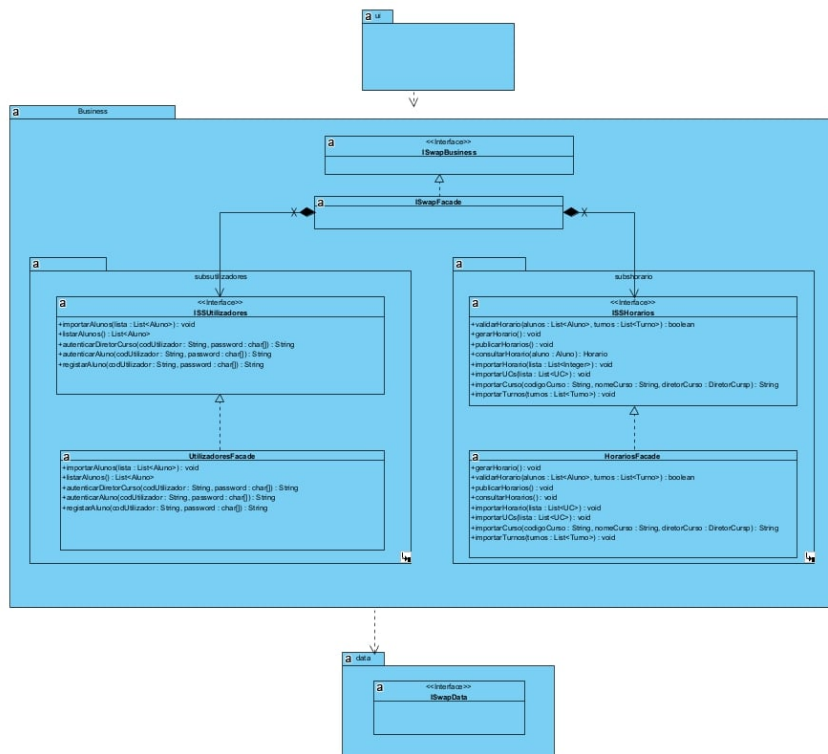
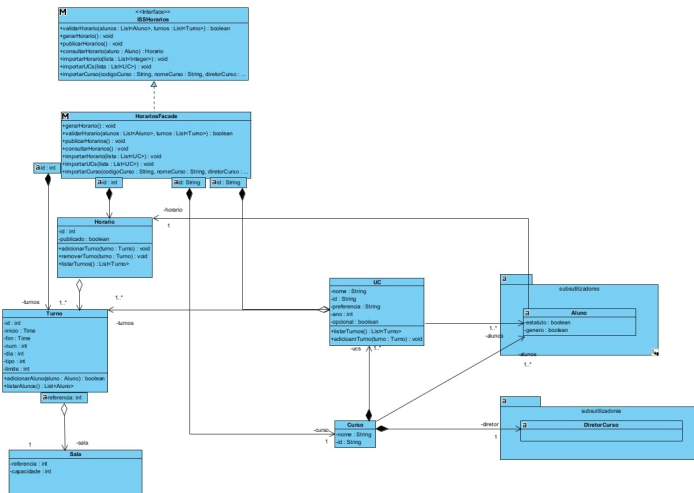


Figura 9: Diagrama de Packages

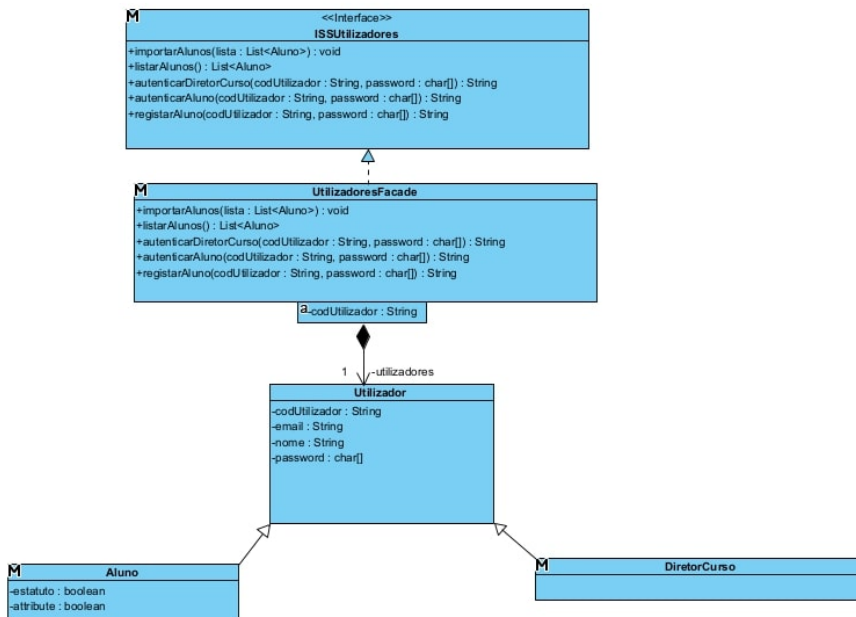
5.1 HorariosFacade

O diagrama de Horário detalha a gestão de horários, com a interface ISHorarios e sua implementação na classe HorariosFacade. Ele organiza elementos como Horário, composto por Turnos, que possuem alunos e estão vinculados a uma Sala. Unidades Curriculares (UCs) são associadas a cursos geridos por diretores.



5.2 UtilizadoresFacade

O diagrama de Utilizador apresenta a gestão de utilizadores por meio da interface ISUtilizadores, implementada pela classe UtilizadoresFacade. A classe base Utilizador é especializada em Aluno e DiretorCurso.



6.2 Diagramas de sequência

6.2.1 Importar alunos

O diagrama de sequência ilustra o processo de importação de alunos, iniciado pelo ator através do método ImportarAlunos na classe UtilizadoresFacade. Para cada aluno na lista fornecida, o sistema verifica se o codUtilizador já existe no repositório AlunoDAO. Caso não exista, o aluno é adicionado; caso contrário, é ignorado.

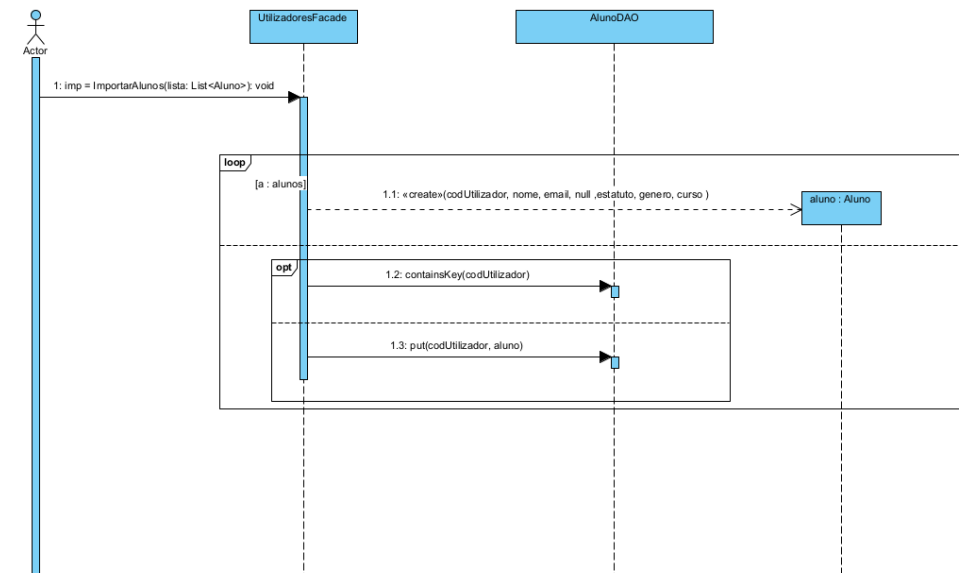


Figura 12: Diagrama referente ao UtilizadorFacade

6.2.2 Gerar horário

O método gerarHorario percorre a lista de alunos verificando as UCs a que se encontra inscrito, colocando o aluno num turno teórico e prático pertencente ao conjunto de turnos que compõe essa tal UC. Caso a UC apresente preferência, a UC é colocada numa lista e a alocação dos alunos é feita no final do método, sendo tratada uma por uma as UCs que se encontram nessa lista.

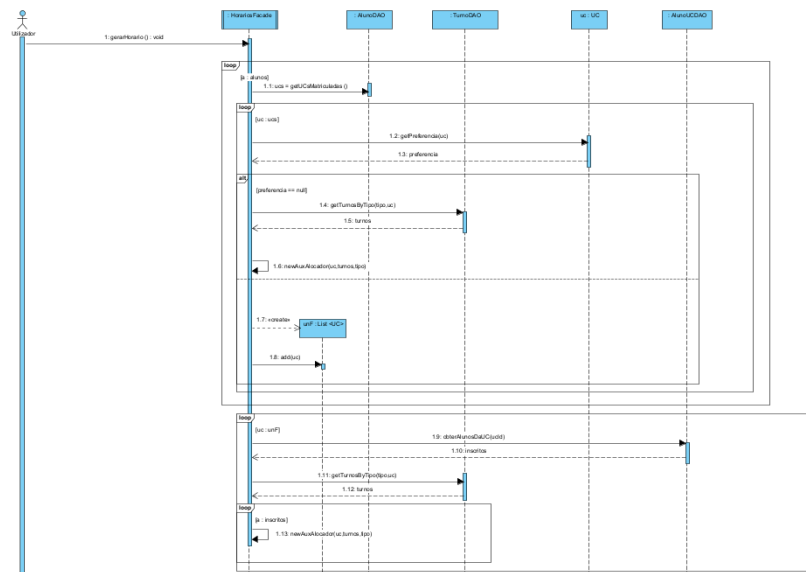


Figura 13: Diagrama referente ao GerarHorarios

A inserção dos alunos tanto nos respectivos turnos, bem como a inserção e modificação da informação sobre essas ligações na base de dados são possíveis através do método auxiliar 'newAuxAlocador'. Esse método trata também de fazer as verificações relativamente ao preenchimento dos turnos aonde um dado aluno pode ingressar.

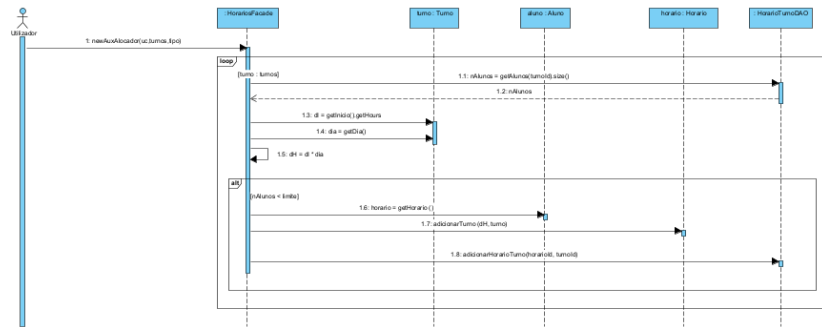


Figura 14: Diagrama referente ao newAuxAlocador

6.2.3 Consultar horário

O método consultarHorario permite consultar o horário de um aluno específico, retornando informações detalhadas sobre os turnos associados ao horário do aluno. Inicialmente, valida se o objeto do aluno fornecido é válido. Em seguida, verifica se o horário do aluno existe e está publicado. Caso não existam turnos associados ao horário, uma mensagem é exibida. Caso contrário, os turnos são ordenados por dia e horário de início, e os detalhes são exibidos antes de retornar o horário do aluno.

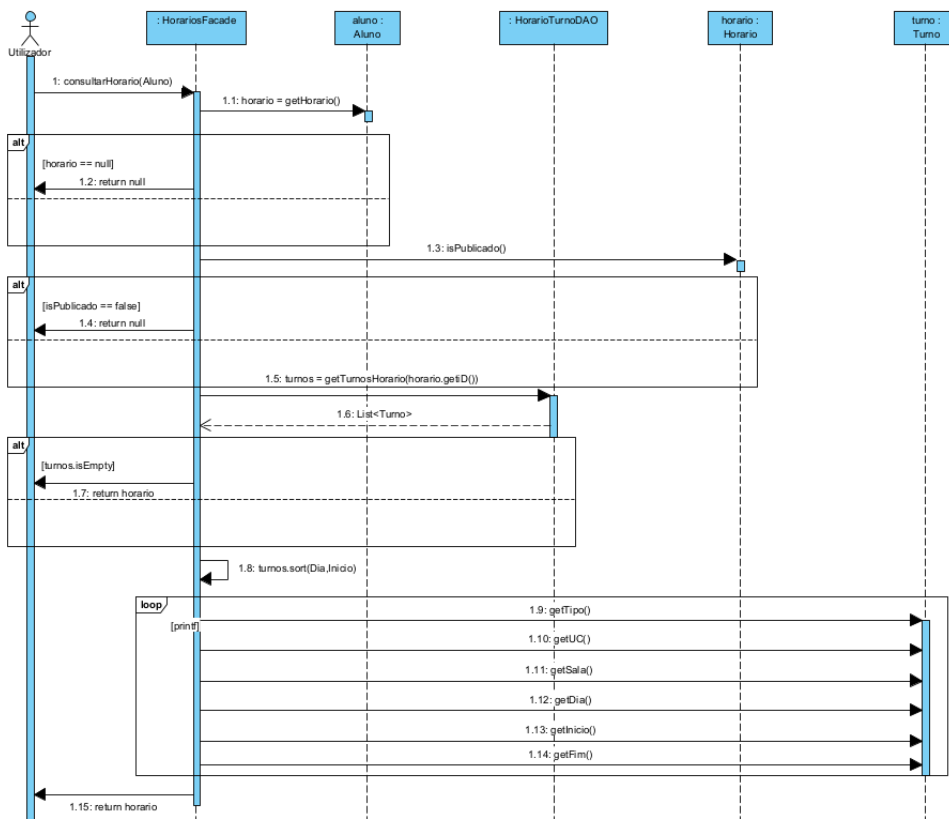


Figura 15: Diagrama referente ao ConsultarHorário

6.2.4 Validar horário

Este método verifica se foi corretamente atribuído um horário a um dado aluno, sendo que esse horário deverá incluir um turno de cada tipo para cada UC à qual o aluno se encontra inscrito.

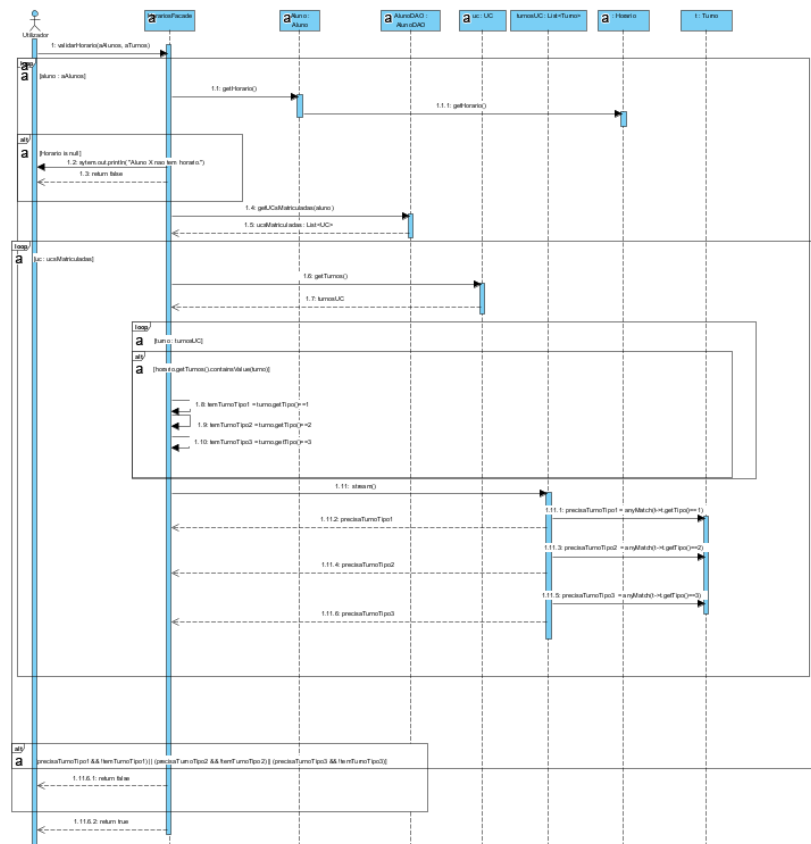


Figura 16: Diagrama referente ao ValidarHorário

6.2.5 Publicar Horários

O método publicarHorarios é responsável por marcar todos os horários gerados como publicados, desde que sejam válidos. Inicialmente, obtém todos os horários armazenados no DAO e verifica se existem horários disponíveis. Em seguida, valida os horários utilizando a lista de alunos e turnos. Após a validação, o método percorre cada horário, marcando-o como publicado e atualizando-o no DAO. Em caso de erros durante o processo, uma exceção é capturada e uma mensagem detalhada é exibida.

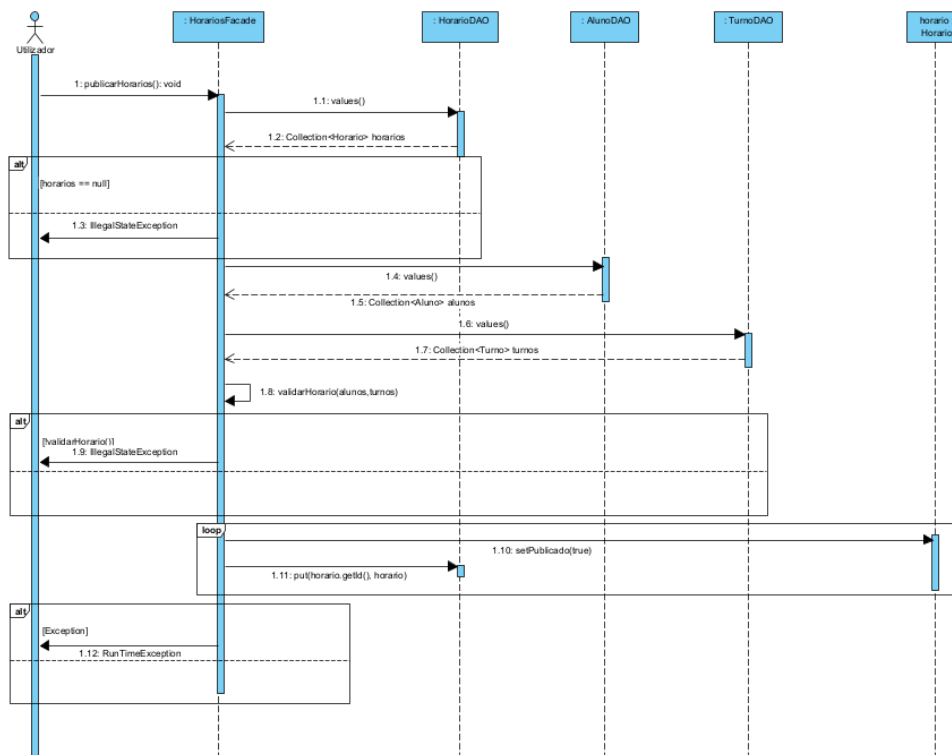


Figura 17: Diagrama referente a publicar horários

6.2.6 Listar Alunos de um dado Turno

Este método encontra-se disponível no menu do Diretor de Curso e permite-lhe visualizar a lista de Alunos de um dado turno através do Id desse turno.

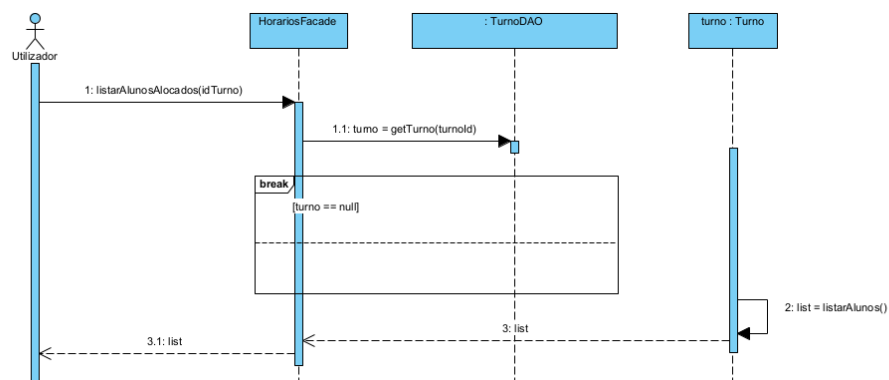


Figura 18: Diagrama referente a listar alunos de um dado turno

6.3 Diagrama de componentes

O diagrama de componentes permite-nos identificar os componentes que constituem o sistema, bem como as dependências existentes entre os mesmos, sendo tratado como componente cada subsistema que constitui o sistema geral. A cada componente é também atribuído uma interface. Encontra-se em baixo anexado o diagrama de componentes resultante da nossa implementação do sistema, apresentando apenas dois componentes.

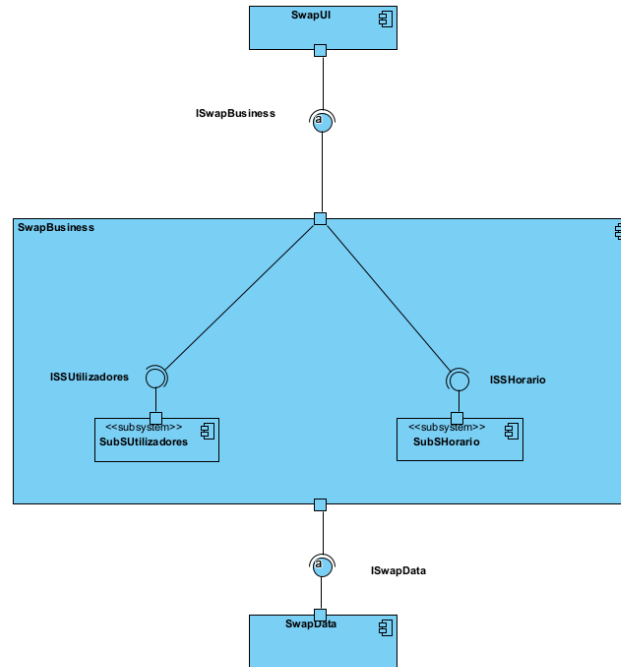


Figura 19: Diagrama de componentes

7 Utilização do sistema

O sistema desenvolvido permite a interação de diferentes perfis de utilizadores, o Diretor de Curso e os Alunos. A seguir, são apresentados os principais passos e funcionalidades, ilustrados com exemplos do sistema em funcionamento.

7.1 Acesso ao Sistema

Na imagem abaixo, é apresentado o ecrã de autenticação do Diretor de Curso. O diretor insere o seu código e password para aceder às funcionalidades administrativas do sistema.

```
C:\Users\POIPI\JUKS\openjuk-23.0.1\bin\java.exe
Script executado: ../SQL/create.sql
Script executado: ../SQL/populate.sql
Banco de dados inicializado com sucesso!
Página Inicial da Swap
0 - Sair do sistema
1 - Interface de Aluno
2 - Interface de Diretor de Curso
>>> 2
Área do Diretor de Curso
0 - Retroceder
1 - Entrar
>>> 1
Insere código de Diretor: d1001
Insere a password: admin
Bem-vindo Carlos Pereira !
0 - Retroceder
1 - Importar
2 - Listar alunos com estatuto
3 - Gerar Horários
4 - Gerar Horários Manualmente
5 - Publicar Horários
6 - Configurar Preferências
7 - Listar Alunos de um Turno
>>> |
```

Figura 20: Login do Diretor de Curso

7.2 Funcionalidades do Diretor de Curso

Após a validação dos horários gerados, o diretor pode publicá-los para que fiquem disponíveis aos alunos. Na imagem, observa-se a interface onde os horários são publicados com sucesso.

```
Bem-vindo Carlos Pereira !
0 - Retroceder
1 - Importar
2 - Listar alunos com estatuto
3 - Gerar Horários
4 - Gerar Horários Manualmente
5 - Publicar Horários
6 - Configurar Preferências
7 - Listar Alunos de um Turno
>>> 5
Todos os horários foram publicados com sucesso.
Bem-vindo Carlos Pereira !
```

Figura 21: Publicação de Horários

O sistema permite ao diretor listar todos os alunos inscritos, exibindo detalhes como o número de aluno e o nome. Abaixo está a interface com essa funcionalidade.

```

4 - Gerar Horários Manualmente
5 - Publicar Horários
6 - Configurar Preferências
7 - Listar Alunos de um Turno
>>> 7
Turnos:
Turno ID: 30, Turno: T1, UC: Base de Dados , N° Alunos: 19
Turno ID: 31, Turno: T2, UC: Base de Dados , N° Alunos: 0
Turno ID: 32, Turno: TP1, UC: Base de Dados , N° Alunos: 19
Turno ID: 33, Turno: TP2, UC: Base de Dados , N° Alunos: 0
Turno ID: 34, Turno: TP3, UC: Base de Dados , N° Alunos: 0
Turno ID: 35, Turno: TP4, UC: Base de Dados , N° Alunos: 0
Turno ID: 24, Turno: T1, UC: Cálculo de Programas , N° Alunos: 15
Turno ID: 25, Turno: T2, UC: Cálculo de Programas , N° Alunos: 4
Turno ID: 26, Turno: PL1, UC: Cálculo de Programas , N° Alunos: 19
Turno ID: 27, Turno: PL2, UC: Cálculo de Programas , N° Alunos: 0
Turno ID: 28, Turno: PL3, UC: Cálculo de Programas , N° Alunos: 0
Turno ID: 29, Turno: PL4, UC: Cálculo de Programas , N° Alunos: 0
Turno ID: 18, Turno: T1, UC: Desenvolvimento de Sistemas de Software , N° Alunos: 47
Turno ID: 19, Turno: T2, UC: Desenvolvimento de Sistemas de Software , N° Alunos: 0
Turno ID: 20, Turno: PL1, UC: Desenvolvimento de Sistemas de Software , N° Alunos: 25
Turno ID: 21, Turno: PL2, UC: Desenvolvimento de Sistemas de Software , N° Alunos: 22
Turno ID: 22, Turno: PL3, UC: Desenvolvimento de Sistemas de Software , N° Alunos: 0
Turno ID: 23, Turno: PL4, UC: Desenvolvimento de Sistemas de Software , N° Alunos: 0
Turno ID: 1, Turno: T2, UC: Programação Funcional , N° Alunos: 19
Turno ID: 2, Turno: TP1, UC: Programação Funcional , N° Alunos: 19
Turno ID: 3, Turno: TP2, UC: Programação Funcional , N° Alunos: 0
Turno ID: 4, Turno: TP3, UC: Programação Funcional , N° Alunos: 0
Turno ID: 5, Turno: TP4, UC: Programação Funcional , N° Alunos: 0
Turno ID: 6, Turno: T1, UC: Programação Imperativa , N° Alunos: 9
Turno ID: 7, Turno: T2, UC: Programação Imperativa , N° Alunos: 0
Turno ID: 8, Turno: PL1, UC: Programação Imperativa , N° Alunos: 9
Turno ID: 9, Turno: PL2, UC: Programação Imperativa , N° Alunos: 0
Turno ID: 10, Turno: PL3, UC: Programação Imperativa , N° Alunos: 0
Turno ID: 11, Turno: PL4, UC: Programação Imperativa , N° Alunos: 0
Turno ID: 12, Turno: T1, UC: Programação Orientada a Objetos , N° Alunos: 34
Turno ID: 13, Turno: T2, UC: Programação Orientada a Objetos , N° Alunos: 0
Turno ID: 14, Turno: TP1, UC: Programação Orientada a Objetos , N° Alunos: 25
Turno ID: 15, Turno: TP2, UC: Programação Orientada a Objetos , N° Alunos: 9
Turno ID: 16, Turno: TP3, UC: Programação Orientada a Objetos , N° Alunos: 0
Turno ID: 17, Turno: TP4, UC: Programação Orientada a Objetos , N° Alunos: 0

Insira código do Turno: 0
Lista de Alunos:
João Pires
Lucas Teixeira
Pedro Carvalho
Eduardo Correia
Lara Barros
Melissa Lima
Sofia Oliveira
Joana Barbosa
Sérgio Almeida

```

Figura 22: Listagem de Alunos

Além da listagem geral, o diretor pode visualizar alunos com estatuto específico, como bolsheiros ou atletas. Abaixo, observa-se a lista filtrada por esta categoria.

```

Bem-vindo Carlos Pereira !
0 - Retroceder
1 - Importar
2 - Listar alunos com estatuto
3 - Gerar Horários
4 - Gerar Horários Manualmente
5 - Publicar Horários
6 - Configurar Preferências
7 - Listar Alunos de um Turno
>>> 2
Aluno: Michael Brown, Número: 200022
Aluno: Jessica Taylor, Número: 200058
Aluno: Pedro Silva, Número: 200002
Aluno: John Smith, Número: 200018
Aluno: Richard Taylor, Número: 200025
Aluno: Nathan Collins, Número: 200076
Aluno: Matthew Johnson, Número: 200046
Aluno: Bruno Ferreira, Número: 200014
Aluno: Ethan Campbell, Número: 200063
Aluno: Sophia Rodriguez, Número: 200040
Aluno: Patrick Wilson, Número: 200028
Aluno: Paulo Ribeiro, Número: 200010
Aluno: James Miller, Número: 200020
Aluno: Diego Martinez, Número: 200071
Aluno: Chris Evans, Número: 200033
Aluno: Mariana Lopes, Número: 200005
Aluno: Edward Carter, Número: 200039

```

Figura 23: Listagem de Alunos com Estatuto

O diretor pode configurar preferências para as UCs, como critérios de ordenação de alunos (e.g., por gênero, média ou grupo). Na imagem abaixo, é mostrado o processo de configuração de uma preferência.

```
Bem-vindo Carlos Pereira !
0 - Retroceder
1 - Importar
2 - Listar alunos com estatuto
3 - Gerar Horários
4 - Gerar Horários Manualmente
5 - Publicar Horários
6 - Configurar Preferências
7 - Listar Alunos de um Turno
>>> 6
Insira código da UC: J304N4
Insira preferência: genero
Preferência configurada com sucesso para a UC J304N4.
```

Figura 24: Definição de Preferências

O sistema permite alocar automaticamente os alunos aos turnos disponíveis. Na imagem, é exibida a conclusão da alocação automática de 118 alunos.

```
Bem-vindo Carlos Pereira !
0 - Retroceder
1 - Importar
2 - Listar alunos com estatuto
3 - Gerar Horários
4 - Gerar Horários Manualmente
5 - Publicar Horários
6 - Configurar Preferências
7 - Listar Alunos de um Turno
>>> 1
Importação dos horarios concluída com sucesso.
Importação de alunos concluída com sucesso.
Importação de ucs concluída com sucesso.
Bem-vindo Carlos Pereira !
0 - Retroceder
1 - Importar
2 - Listar alunos com estatuto
3 - Gerar Horários
4 - Gerar Horários Manualmente
5 - Publicar Horários
6 - Configurar Preferências
7 - Listar Alunos de um Turno
>>> 3
Aluno 200072 já está alocado no horário: 8 do dia 6
Aluno 200078 já está alocado no horário: 8 do dia 6
Aluno 200054 já está alocado no horário: 8 do dia 6
ALUNOS ALOCADOS : 118
```

Figura 25: Geração Automática de Horários

Quando necessário, o diretor pode alocar manualmente alunos a turnos específicos. Abaixo está a interface para essa funcionalidade.

```
Bem-vindo Carlos Pereira !
0 - Retroceder
1 - Importar
2 - Listar alunos com estatuto
3 - Gerar Horários
4 - Gerar Horários Manualmente
5 - Publicar Horários
6 - Configurar Preferências
7 - Listar Alunos de um Turno
>>> 4
Insira código do Aluno: 200077
UCs em que o aluno 200077 está inscrito:
ID: J304N4, Nome: Base de Dados, Ano: 2, Opcional: Não, T: Não, TP: Não, PL: Não
ID: J305N2, Nome: Desenvolvimento de Sistemas de Software, Ano: 3, Opcional: Não, T: Não, TP: Não, PL: Não

Insira código da UC: J304N4
Turnos disponíveis para a UC J304N4:
ID: 30, Tipo: T, Dia: 1, Inicio: 08:00:00, Fim: 10:00:00, Limite: 30, Sala: 15
ID: 31, Tipo: T, Dia: 1, Inicio: 10:00:00, Fim: 12:00:00, Limite: 30, Sala: 15
ID: 32, Tipo: TP, Dia: 3, Inicio: 14:00:00, Fim: 16:00:00, Limite: 30, Sala: 14
ID: 33, Tipo: TP, Dia: 3, Inicio: 16:00:00, Fim: 18:00:00, Limite: 30, Sala: 14
ID: 34, Tipo: TP, Dia: 3, Inicio: 14:00:00, Fim: 16:00:00, Limite: 30, Sala: 14
ID: 35, Tipo: TP, Dia: 3, Inicio: 16:00:00, Fim: 18:00:00, Limite: 30, Sala: 14

Insira código do Turno: 30
```

Figura 26: Alocação Manual de Alunos

7.3 Funcionalidades do Aluno

Os alunos podem autenticar-se no sistema para aceder à área destinada a eles, onde é possível consultar os horários gerados. Abaixo está o exemplo de login de um aluno com o registo de password e a consulta do seu horário com detalhes dos turnos.

```
Página Inicial da Swap
0 - Sair do sistema
1 - Interface de Aluno
2 - Interface de Diretor de Curso
>>> 1
Área do Aluno
0 - Retroceder
1 - Entrar
2- Registar Password
>>> 2
Insere o ID do Aluno: 200079
Insere a password: 1234
Aluno registado com sucesso.
Área do Aluno
0 - Retroceder
1 - Entrar
2- Registar Password
>>> 1
Insere o ID do Aluno: 200079
Insere a password: 1234
Área do Aluno: João Andrade
0 - Retroceder
1 - Consultar Horário
>>> 1
Obtendo horário da base de dados...
Horário do aluno: João Andrade
-----
Turno ID: 30 | Tipo: T | UC: Base de Dados | Sala: 15 | Dia: Segunda-feira | Início: 08:00:00 | Fim: 10:00:00
Turno ID: 32 | Tipo: TP | UC: Base de Dados | Sala: 14 | Dia: Quarta-feira | Início: 14:00:00 | Fim: 16:00:00
Turno ID: 18 | Tipo: T | UC: Desenvolvimento de Sistemas de Software | Sala: 2 | Dia: Quinta-feira | Início: 08:00:00 | Fim: 10:00:00
Turno ID: 20 | Tipo: PL | UC: Desenvolvimento de Sistemas de Software | Sala: 8 | Dia: Quinta-feira | Início: 16:00:00 | Fim: 18:00:00
-----
```

Figura 27: Login e Consulta de Horários

8 Conclusão e análise crítica dos resultados

Durante esta fase final do desenvolvimento do sistema, alcançamos a implementação completa das funcionalidades previamente planeadas, integrando métodos essenciais para a gestão eficiente dos horários dos alunos. Através de uma abordagem focada na solução dos problemas identificados, conseguimos automatizar a alocação de alunos aos turnos com base em preferências e restrições, minimizando conflitos e assegurando uma distribuição equilibrada.

A funcionalidade de geração de horários revelou-se fundamental para garantir que os alunos fossem alocados corretamente aos turnos teóricos e práticos das unidades curriculares, respeitando as limitações impostas por preferências individuais ou requisitos de capacidade. Adicionalmente, a validação dos horários gerados, realizada pelo método de validação, assegurou a consistência e a adequação dos horários atribuídos, permitindo que apenas horários completos e corretos fossem disponibilizados.

A publicação dos horários representou outro marco importante, possibilitando que os horários fossem disponibilizados apenas após a sua validação. Este processo garantiu a fiabilidade da informação fornecida aos utilizadores. Paralelamente, a funcionalidade de consulta de horários ofereceu aos alunos uma forma intuitiva de acederem aos seus horários, com detalhes organizados e ordenados por dia e hora, melhorando significativamente a usabilidade do sistema.

Do ponto de vista técnico, o sistema foi desenvolvido com uma arquitetura modular, garantindo uma separação clara entre as responsabilidades das camadas de interface, lógica de negócio e dados. Essa abordagem não apenas facilitou a implementação, como também proporcionou maior flexibilidade e escalabilidade para futuras alterações e melhorias.

Apesar dos desafios enfrentados durante o desenvolvimento, incluindo ajustes nos modelos iniciais e nos diagramas de sequência, foi possível adaptar e evoluir as soluções de forma iterativa, alinhando o sistema às necessidades práticas identificadas. Este processo assegurou que os objetivos principais fossem atingidos, embora ainda existam oportunidades de otimização, nomeadamente no tratamento de exceções e na realização de testes automatizados para cenários mais complexos.

Concluindo, o sistema desenvolvido cumpre os objetivos traçados e oferece uma solução eficiente para a gestão de horários e turnos em contextos educacionais. Com uma base sólida e potencial para melhorias futuras, este sistema representa um avanço significativo na organização e no suporte às atividades académicas.

Desenvolvimento de Sistemas de Software

Licenciatura em Engenharia Informática

Departamento de Informática

Universidade do Minho

2024/2025

Enunciado do Trabalho

José Creissac Campos
jose.campos@di.uminho.pt

António Nestor Ribeiro
anr@di.uminho.pt

Rafael Costa
d13414@di.uminho.pt

(actualizado em 21/11/2024)

Conteúdo

1	Introdução	1
2	Objectivo do trabalho	1
3	Cursos, horários, unidades curriculares e alunos	1
4	Realização do trabalho	2
4.1	Entrega intermédia	2
4.2	Entrega final	2
5	Apresentação e discussão do trabalho	3
6	Avaliação	4
7	Grupos de Trabalho	4

1 Introdução

Este documento apresenta o enunciado do trabalho prático da Unidade Curricular (UC) de Desenvolvimento de Sistemas Software para o ano lectivo 2024/2025. **Leia-o com atenção**, já que descreve, não só o sistema a desenvolver, como o processo que deve seguir para a realização do trabalho. Quaisquer dúvidas devem ser esclarecidas junto dos docentes da UC.

2 Objectivo do trabalho

Em cursos de grande dimensão, como é o caso da Licenciatura em Engenharia Informática (LEI), os alunos têm de ser distribuídos por turnos para viabilizar a realização das aulas. A gestão dessa distribuição por turnos é um processo complexo, uma vez que é necessário compatibilizar os turnos das diferentes Unidades Curriculares (UC) de cada ano do curso (e, eventualmente, de diferentes anos do curso), de modo a que os alunos não tenham colisões no seu horário.

Historicamente, os docentes de cada UC geriam a definição dos turnos da sua UC. Esta solução mostrou não ser escalável, criando situações em que os alunos tinham dificuldades em obter horários sem colisões e atrasando o início das aulas.

Uma forma de facilitar o processo é centralizar a criação dos turnos nas direções de curso. No entanto, essa criação é complexa e demorada, consumindo recursos, pelo que só é viável se as listas de inscritos forem conhecidas com antecedência suficiente. Além disso, tem as suas próprias desvantagens, uma vez que não permite a cada UC definir a sua política para a criação dos turnos (por exemplo, requerendo que todos os elementos de cada grupo prático frequentem o mesmo turno), nem permite aos alunos controlo sobre o seu horário.

Com este projeto pretende-se desenvolver um sistema para auxiliar a criação dos turnos práticos de um curso, dadas as suas UC, o horário e as listas de inscritos. O sistema deverá suportar diferentes algoritmos de priorização de alunos na criação dos turnos.

3 Cursos, horários, unidades curriculares e alunos

Para a realização deste trabalho recorra aos conhecimentos que tem sobre cursos, horários, UC e alunos. Relembre que um curso é organizado em semestres, tendo em cada semestre um conjunto de UC (algumas poderão ser UC opcionais com várias alternativas), podendo os alunos estar inscritos a UC de semestres diferentes.

Ao pensar na sua solução tenha em atenção que diferentes tipos de aula podem ter limites de alunos por turma diferentes. No caso das aulas teóricas, o limite é a capacidade da sala a que a aula está alocada no horário. No caso de aulas teórico-práticas (TP) e práticas laboratoriais (PL) o limite é definido UC a UC e poderá ser inferior ao tamanho da sala.

Tenha ainda em atenção que os alunos com estatuto especial (trabalhadores-estudantes e atletas, por exemplo) devem ter prioridade na alocação a turnos. A forma de refletir essa prioridade nos turnos depende da estratégia adoptada para a sua criação. Estratégias comumente utilizadas no passado incluem reservar alguns lugares em cada turno para esses alunos, ou pedir de antemão as suas preferências (quando feito de forma manual, apenas viável em turmas pequenas).

Para esclarecimento de eventuais dúvidas contacte os docentes de DSS.

4 Realização do trabalho

A concepção e desenvolvimento da aplicação deverá seguir uma abordagem baseada em modelos (suportada por UML), de acordo com o processo de entregas faseadas descrito nas aulas teóricas. A aplicação deverá ser desenvolvida utilizando uma arquitectura multi-camada e tecnologias orientadas a objectos (preferencialmente, Java). Irá ser criado um repositório no GitHub¹ para cada grupo, onde deverá ser mantida a versão actualizada do trabalho.

Para facilitar o processo de concepção e desenvolvimento, o trabalho será realizado em duas fases.

4.1 Entrega intermédia

Análise de requisitos – a entregar até às 23h59 de 19 de Outubro.

Objectivos:

- Um Modelo de Domínio com as entidades relevantes
- Um Modelo de Use Case (diagramas mais especificações do Use Case) com as funcionalidades propostas para o sistema

O resultado desta fase será sujeito a avaliação qualitativa.

4.2 Entrega final

Modelação conceptual e implementação da solução – a entregar até às 23h59 de 7 de dezembro (atualizações possíveis até 4 de janeiro de 2025).

¹<https://github.com>

Objectivos:

- Uma arquitectura conceptual do sistema, capaz de suportar os requisitos identificados – em particular a solução adoptada deverá garantir, tanto quanto possível, que não são cometidos erros como gerar turnos com conflitos
- Os modelos comportamentais necessários para descrever o comportamento pretendido para o sistema
- Os modelos que considere necessários à descrição da implementação do sistema
- A implementação do sistema
- Documento técnico com todos os modelos desenvolvidos (em PDF).

Pretende-se que o documento técnico sirva de apoio à análise do trabalho, pelo que **deverá ter a seguinte estrutura:**

- **Capa com identificação** da Unidade Curricular, **do grupo (com fotos dos elementos)** e o URL do **repositório do trabalho**.
- Descrição dos resultados obtidos (máximo uma página).
- Diagramas relativos à **análise de requisitos** (Modelação de Domínio, Diagramas de *Casos de Uso* e correspondentes descrições dos casos de uso).
- Diagramas relativos à **modelação conceptual da solução** proposta (Diagramas de Classe e de Sequência).
- Diagramas com a descrição da **solução efectivamente implementada** (Diagramas de Classe, de Sequência, de Componentes e de *packages*).
- Manual de utilização do sistema desenvolvido.
- Em anexo, este enunciado.

Os diagramas mencionados acima podem ser complementados com outros que considerem relevante incluir.

5 Apresentação e discussão do trabalho

Para a apresentação do trabalho deverão preparar uma apresentação com a duração máxima de 15 minutos. Esta apresentação deverá descrever a solução e a

abordagem seguida para a atingir, desde a análise dos cenários até a implementação e demonstração da solução final. A apresentação deverá terminar com uma análise crítica dos resultados obtidos.

Após essa apresentação, seguir-se-á um período de análise e discussão do trabalho de até 30 minutos.

6 Avaliação

A apresentação e discussão final do trabalho será realizada na semana de 6 a 10 de janeiro de 2025, em horários a combinar. A **presença** na discussão do trabalho é **obrigatória**.

Os pesos relativos de cada componente do trabalhos serão os seguintes:

- Modelo de domínio e análise de requisitos: 25%
- Modelação conceptual: 25%
- Modelação final e implementação: 35%
- Apresentação e discussão: 15%

A nota de cada elemento do grupo será individual, tendo em consideração a nota do trabalho e a avaliação por pares. A equipa docente reserva-se a possibilidade de ajustar as notas, em função da sua avaliação de cada elemento durante a discussão do trabalho.

7 Grupos de Trabalho

Os grupos de trabalho deverão obrigatoriamente ser constituídos por de 3 a 5 elementos. A definição dos grupos de trabalho será realizada no Blackboard, **terminando a 2 de outubro**.