



**Universidade do Minho**  
Escola de Engenharia

## **Desenvolvimento de Sistemas de Software**

### **Trabalho Prático – Fase 2**

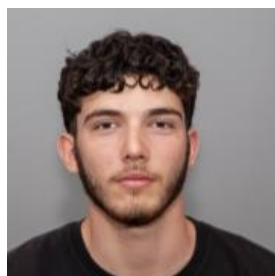
Grupo 17

Link GitHub: <https://github.com/LEI-DSS/DSS2425-Grupo-17>



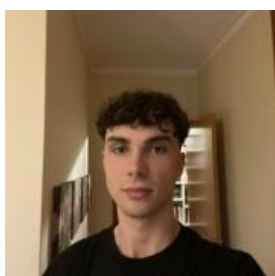
Júlia Costa

A103993



João Rodrigues

A104274



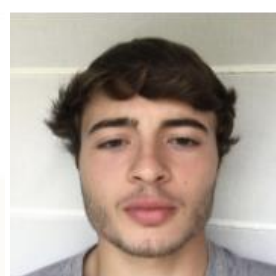
Tomás Barbosa

A104532



Gabriel Ribeiro

A104171



Luís Cunha

A104613

# Índice

Modelo de domínio .....	3
Diagramas de Use Cases .....	4
Use Case – Importar Listas .....	4
Use Case – Definir Preferências .....	5
Use Case – Gerar Horário .....	5
Use Case – Alocar Manualmente .....	5
Use Case – Consultar Horário .....	6
Use Case – Exportar Horário .....	6
Use Case – Fazer Login.....	7
Use Case – Fazer Logout.....	7

# Modelo de domínio

Durante o desenvolvimento de um sistema de software, o planeamento e análise do problema são fases que devem preceder a implementação, garantindo que este processo seja o mais eficiente e organizado possível, resultando num código melhor estruturado.

Na fase de design de software procedemos ao desenvolvimento de um modelo de domínio, representado na Fig.1, no qual representamos as entidades relativas ao problema e as relações entre si.

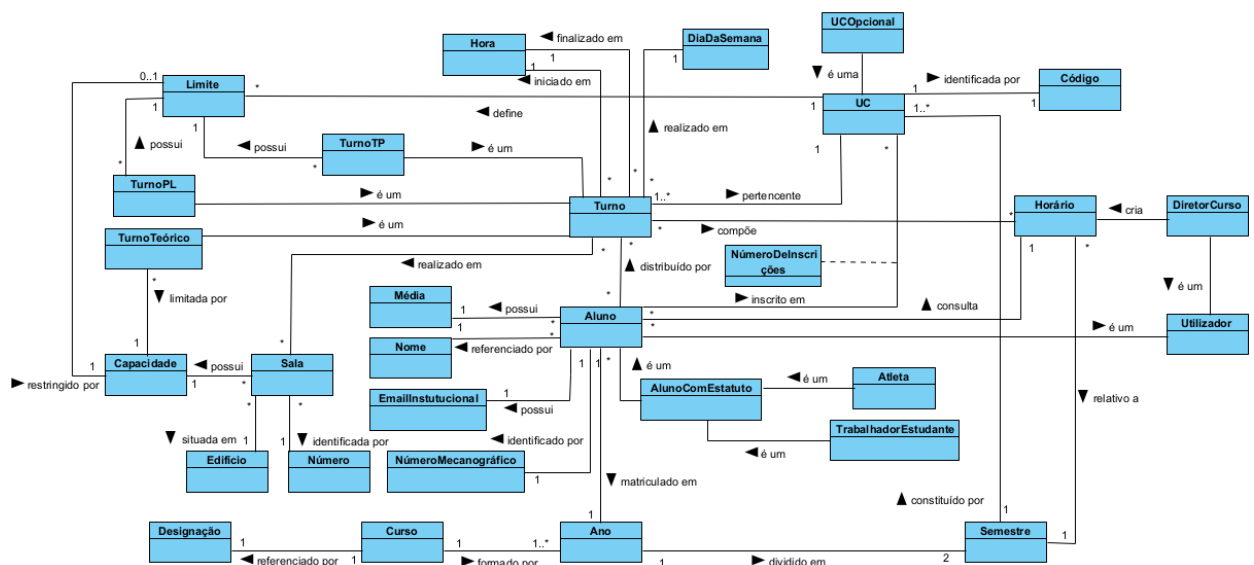


Figura 1 – Modelo de Domínio

# Diagramas de Use Cases

De maneira a representar as interações dos demais utilizadores passamos ao desenvolvimento de um diagrama de Use Cases, ilustrando as funcionalidades que o sistema lhes oferece e como os atores interagem com elas. Estes são os Use Cases base mais importantes para o funcionamento da aplicação, mas serão tidos em consideração outros à medida que se tornem necessários (Ex: Adicionar alunos à lista manualmente, etc.).

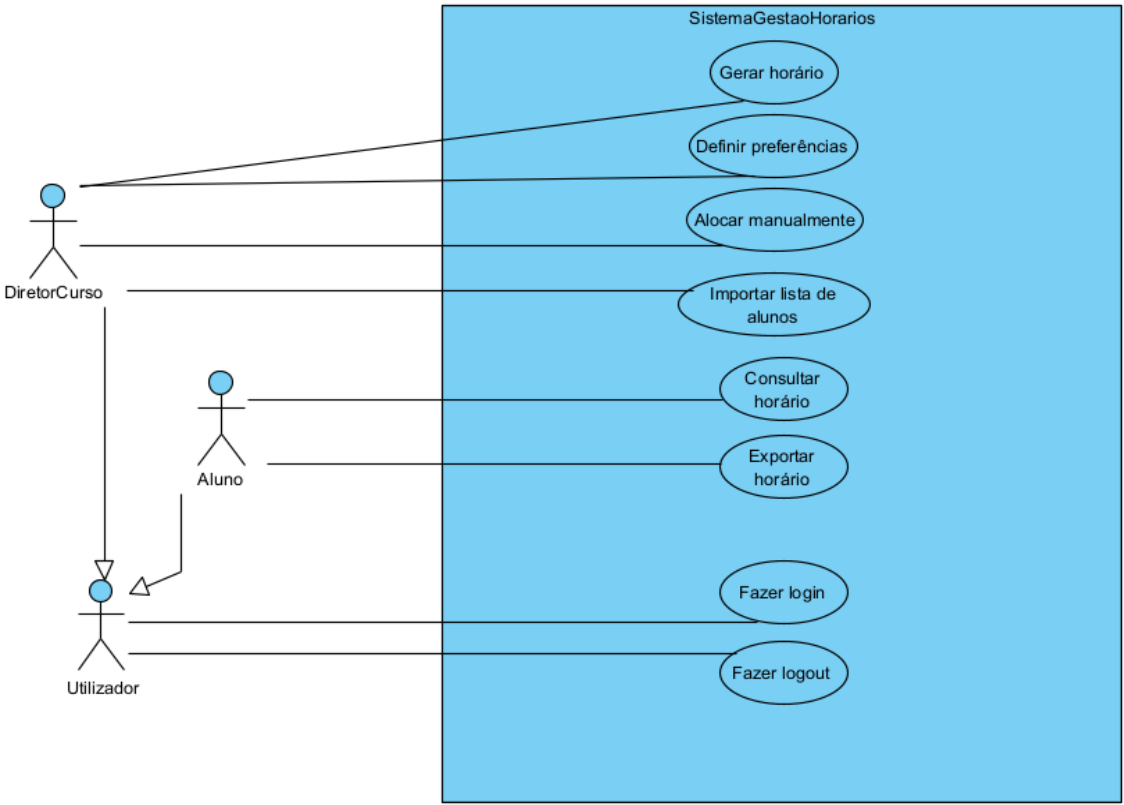


Figura 2 – Diagrama de Use Cases

## Use Case – Importar Listas

<b>USE CASE:</b>	Importar listas
<b>DESCRIÇÃO:</b>	
<b>CENÁRIOS:</b>	Cenário - O Diretor de Curso
<b>PRÉ-CONDIÇÃO:</b>	O ator está autenticado e as listas estão disponíveis
<b>PÓS-CONDIÇÃO:</b>	O sistema importa as listas com sucesso
<b>FLUXO NORMAL:</b>	
	1. O ator seleciona a opção de importar listas
	2. O ator carrega os documentos no sistema e confirma a seleção
	3. O sistema valida e importa as listas com sucesso
<b>FLUXO DE EXCEÇÃO</b>	(1) [As listas não são validadas pelo sistema] (passo 3)
	3.1 O sistema rejeita as listas
	3.2 O sistema termina o processo

Figura 3 – Use Case Importar Listas

## Use Case – Definir Preferências

<b>USE CASE:</b>	Definir preferências
<b>DESCRIÇÃO:</b>	
<b>CENÁRIOS:</b>	Cenário - O Diretor de Curso
<b>PRÉ-CONDIÇÃO:</b>	O ator está autenticado no sistema
<b>PÓS-CONDIÇÃO:</b>	O sistema guarda as preferências dos docentes das UCs
<b>FLUXO NORMAL:</b>	
	1. O ator seleciona a opção definir preferências
	2. O ator seleciona a UC que quer determinar as preferências
	3. O ator escolhe as preferências
	4. O sistema guarda as preferências da UC
<b>FLUXO DE EXCEÇÃO</b>	(1) [Valor de preferência inválido] (passo 3)
	3.1 O sistema informa que a preferência é inválida
	3.2 O sistema termina o processo

Figura 4 – Use Case Definir Preferências

## Use Case – Gerar Horário

<b>USE CASE:</b>	Gerar horário
<b>DESCRIÇÃO:</b>	
<b>CENÁRIOS:</b>	Cenário - O Diretor de Curso
<b>PRÉ-CONDIÇÃO:</b>	O ator está autenticado e listas necessárias estão importadas no sistema
<b>PÓS-CONDIÇÃO:</b>	O sistema gera horários dos alunos
<b>FLUXO NORMAL:</b>	
	1. O ator solicita a geração da alocação inicial dos alunos
	2. O sistema gera horário para cada aluno
	3. O sistema indica que os horários foram gerados
<b>FLUXO ALTERNATIVO</b>	(1) [Ocorreram conflitos na geração de horários] (passo 2)
	2.1 O sistema avisa que foram encontrados conflitos em horários e indica os alunos afetados
	2.2 Regressa a 3

Figura 5 – Use Case Gerar Horário

## Use Case – Alocar Manualmente

<b>USE CASE:</b>	Alocar manualmente
<b>DESCRIÇÃO:</b>	
<b>CENÁRIOS:</b>	Cenário - O Diretor de Curso
<b>PRÉ-CONDIÇÃO:</b>	O ator está autenticado, o sistema já gerou os horários dos alunos e há alunos sem horário
<b>PÓS-CONDIÇÃO:</b>	Os alunos restantes são alocados manualmente nos turnos disponíveis
<b>FLUXO NORMAL:</b>	
	1. O sistema apresenta lista de alunos não alocados
	2. O ator aloca manualmente cada aluno nos turnos conforme a sua prioridade
	3. O sistema indica que os alunos foram colocados nos turnos disponíveis.
<b>FLUXO ALTERNATIVO</b>	(1) [O aluno tem conflitos de horário] (passo 1)
	1.1 O sistema informa ator de conflitos no horário do aluno
	1.2 Regressa a 2.
<b>FLUXO ALTERNATIVO</b>	(2) [Não foi possível cumprir as preferências das UC] (passo 2)
	2.1 O sistema informa ator que não foi possível cumprir preferências da UC
	2.2 Regressa a 2.

Figura 6 – Use Case Alocar Manualmente

## Use Case – Consultar Horário

<b>USE CASE:</b>	Consultar horário
<b>DESCRIÇÃO:</b>	O ator visualiza o seu horário mostrado pelo sistema
<b>CENÁRIOS:</b>	Cenários - Os Alunos
<b>PRÉ-CONDIÇÃO:</b>	O ator está autenticado e tem horário atribuído
<b>PÓS-CONDIÇÃO:</b>	O sistema apresenta horário do ator
<b>FLUXO NORMAL:</b>	
	1. Ator seleciona opção de consultar horário
	2. O sistema apresenta horário do ator

Figura 7 – Use Case Consultar Horário

## Use Case – Exportar Horário

<b>USE CASE:</b>	Exportar horário
<b>DESCRIÇÃO:</b>	O ator exporta o seu horário para consulta fora da aplicação
<b>CENÁRIOS:</b>	Cenários - Os Alunos
<b>PRÉ-CONDIÇÃO:</b>	O ator está autenticado e tem horário atribuído
<b>PÓS-CONDIÇÃO:</b>	O sistema exporta horário do ator
<b>FLUXO NORMAL:</b>	
	1. O ator seleciona opção de exportar o horário
	2. O sistema transforma horário num formato para output
	3. O sistema descarrega o horário para fora da aplicação
<b>FLUXO DE EXCEÇÃO</b>	(1) [Há erro na transformação do horário] (passo 2)
	2.1 O sistema informa que não consegue transformar o horário no formato desejado
	2.2 O sistema encerra processo
<b>FLUXO DE EXCEÇÃO</b>	(2) [Há erro na exportação do horário] (passo 3)
	3.1 O sistema informa que não consegue exportar o horário
	3.2 O sistema encerra processo

Figura 8 – Use Case Exportar Horário

## Use Case – Fazer Login

<b>USE CASE:</b>	Fazer login
<b>DESCRIÇÃO:</b>	O ator autentica-se no sistema
<b>CENÁRIOS:</b>	Cenários - O Diretor de Curso, Os Alunos
<b>PRÉ-CONDIÇÃO:</b>	O ator está registado no sistema
<b>PÓS-CONDIÇÃO:</b>	O ator fica autenticado no sistema
<b>FLUXO NORMAL:</b>	
	1. O ator indica email e password
	2. O sistema valida as credenciais
	3. O ator fica autenticado e entra no sistema
<b>FLUXO DE EXCEÇÃO</b>	(1) [Credenciais fornecidas são inválidas] (passo 2)
	2.1 O sistema indica que as credenciais estão inválidas
	2.2 O sistema cancela o processo

Figura 9 – Use Case Fazer Login

## Use Case – Fazer Logout

<b>USE CASE:</b>	Fazer logout
<b>DESCRIÇÃO:</b>	O ator desautentica-se do sistema
<b>CENÁRIOS:</b>	Cenários - O Diretor de Curso, Os Alunos
<b>PRÉ-CONDIÇÃO:</b>	O ator está autenticado no sistema
<b>PÓS-CONDIÇÃO:</b>	O ator não está autenticado no sistema
<b>FLUXO NORMAL:</b>	
	1. O ator escolhe opção de logout
	2. O ator confirma que quer fazer logout
	3. O sistema remove autenticação da conta do utilizador
<b>FLUXO DE EXCEÇÃO</b>	(1) [O ator não confirma logout] (passo 2)
	O sistema cancela processo e o ator continua autenticado na aplicação
	2.1 aplicação

Figura 10 – Use Case Fazer Logout

## Fase 2

## Alterações no Modelo de Domínio

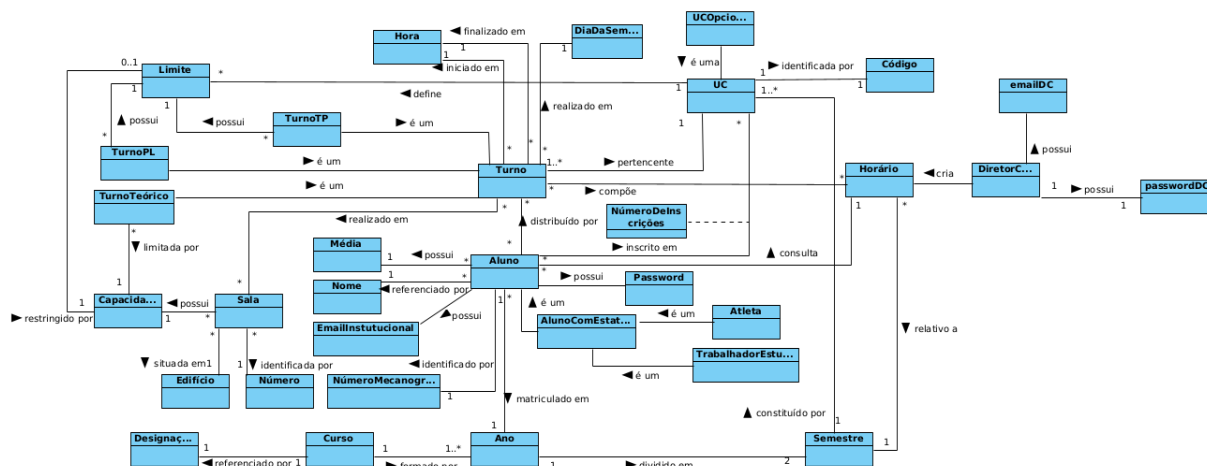


Figura 11 - Alterações no modelo de domínio relativamente à primeira fase

### Diagrama de componentes

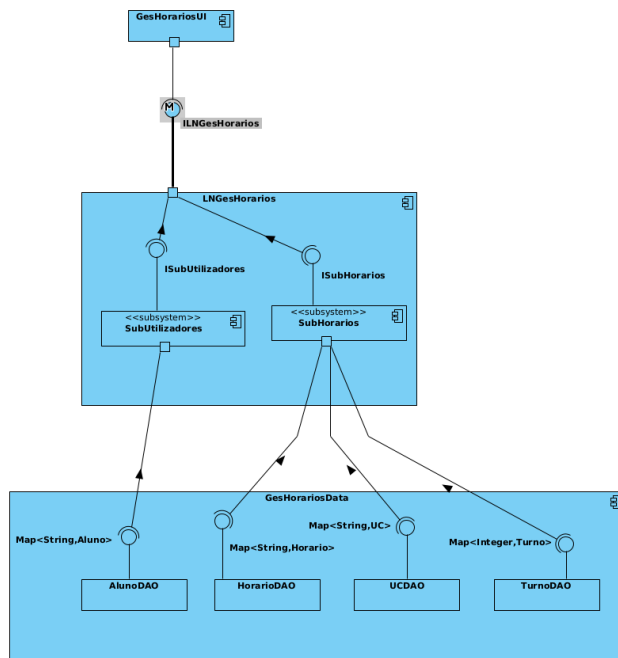


Figura 12 – Diagrama de componentes



## Arquitetura Geral

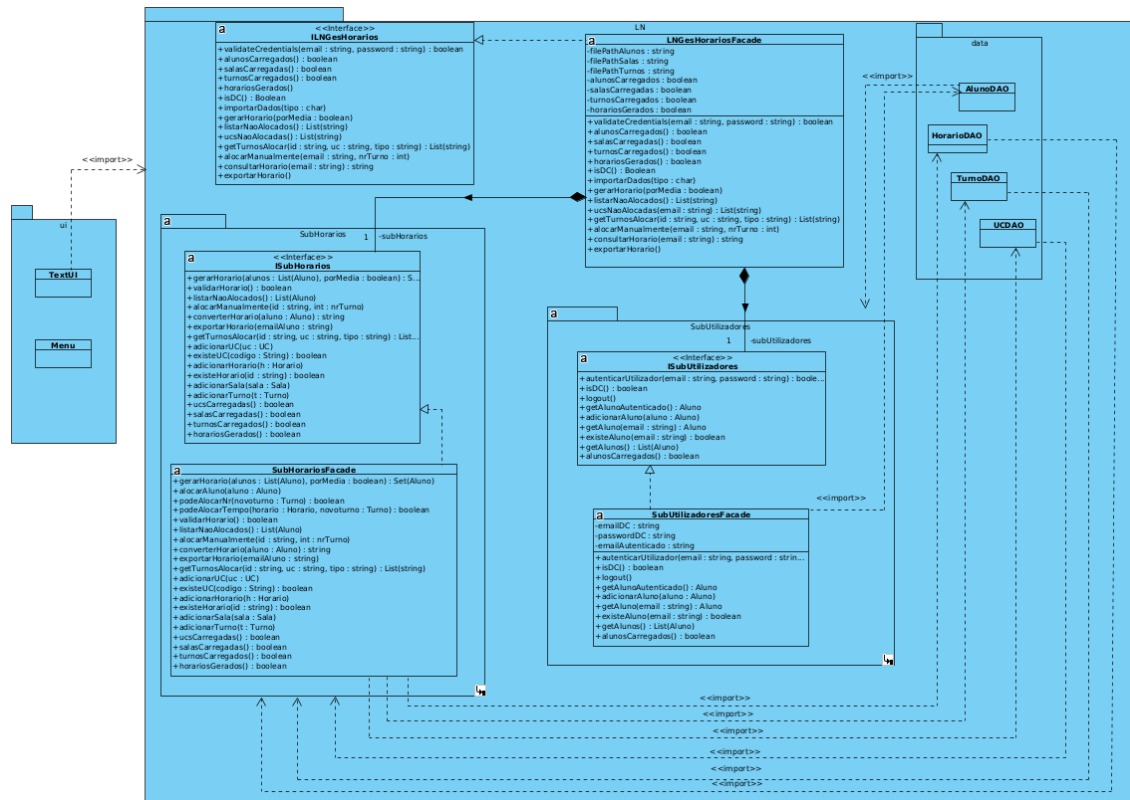


Figura 13 – Arquitetura geral

### Diagrama de classes SubHorarios sem DAO

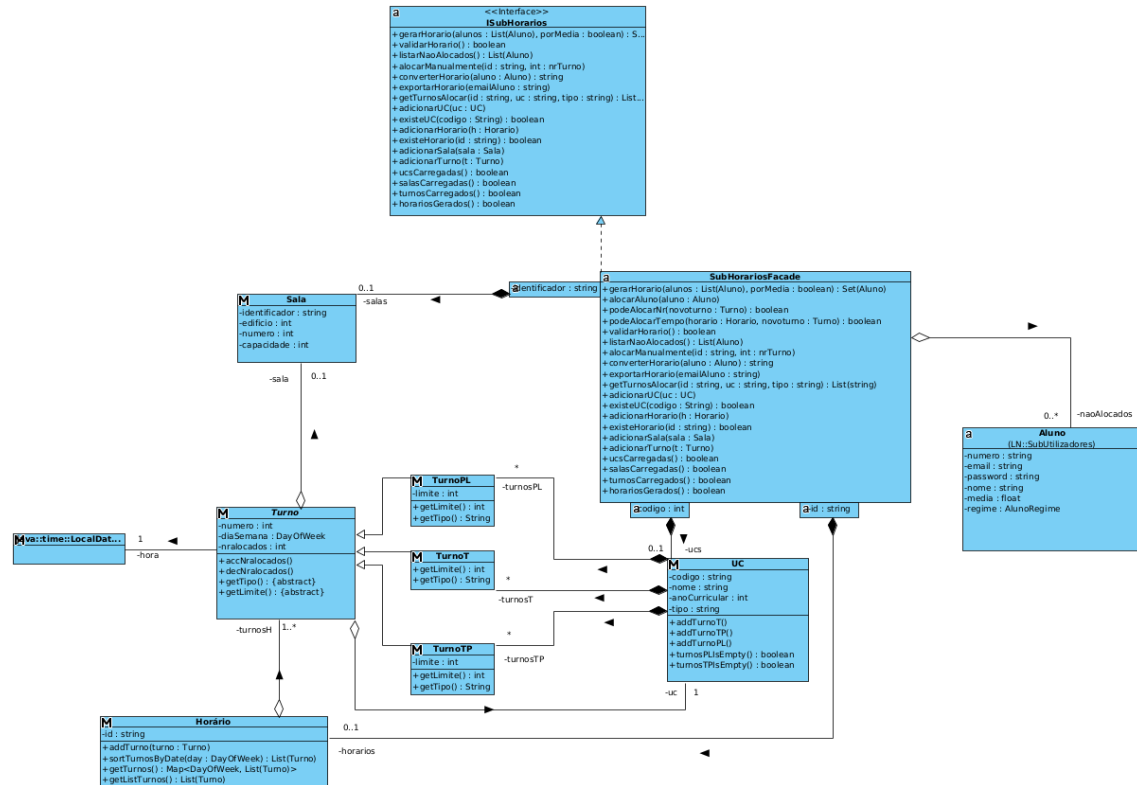


Figura 14 – Diagrama de classes SubHorarios sem DAO

### Diagrama de classes SubUtilizadores sem DAO

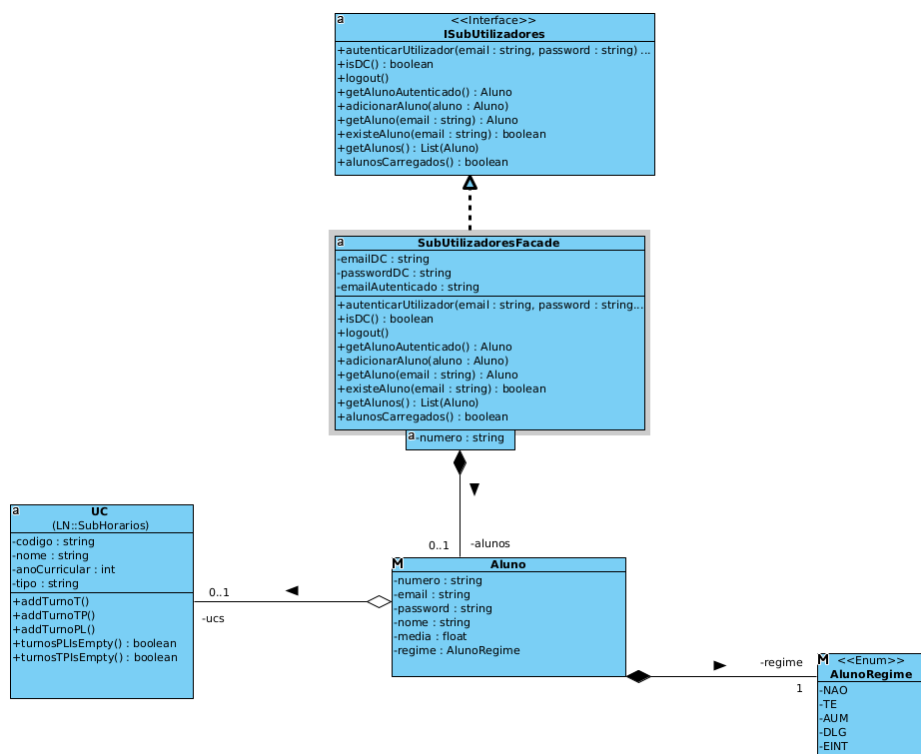


Figura 15 – Diagrama de classes SubUtilizadores sem DAO

### Diagrama de classes SubHorarios com DAO

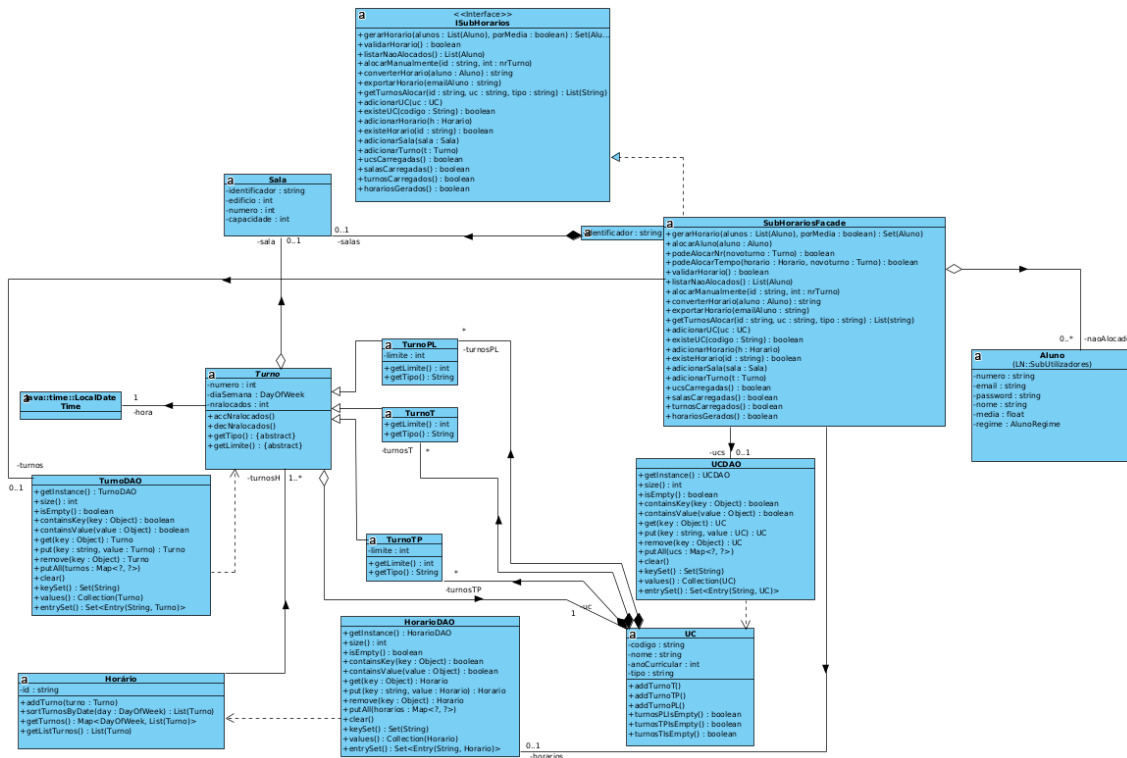


Figura 16 – Diagrama de classes SubHorarios com DAO

Diagrama de classes SubUtilizadores com DAO

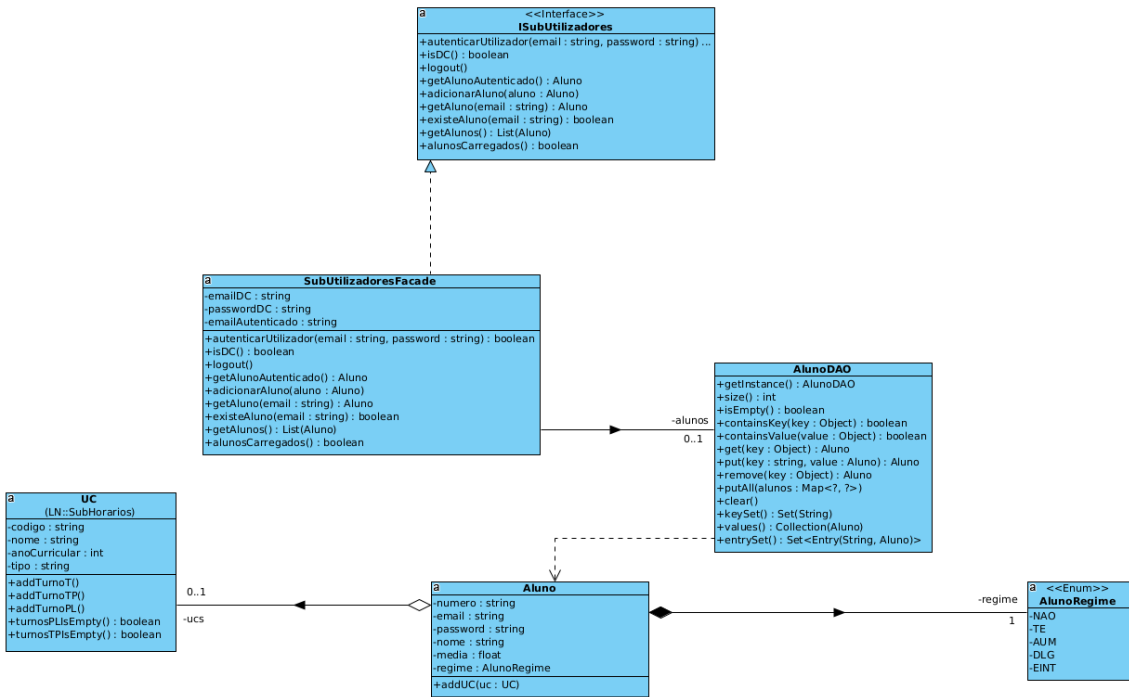


Figura 17 – Diagrama de classes SubUtilizadores com DAO

Diagrama de Sequência - Consultar Horário

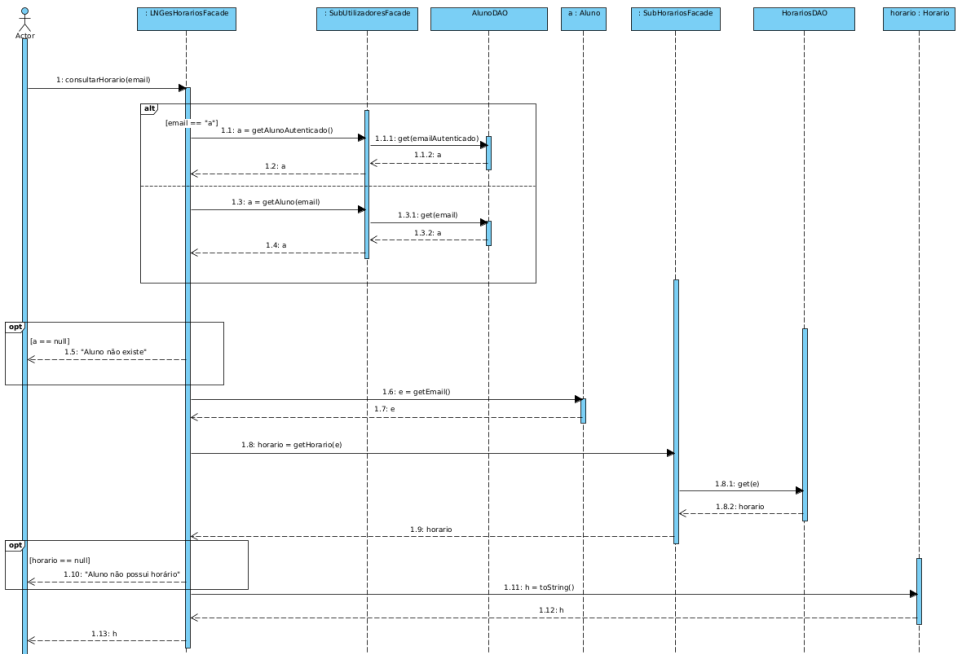


Figura 18 – Diagrama de sequência - Consultar Horário

Diagrama de Sequência - Importar Aluno

O método utilizado para importar os alunos contém a importação de todos os dados, mas no diagrama de sequência tratamos apenas do caso dos alunos

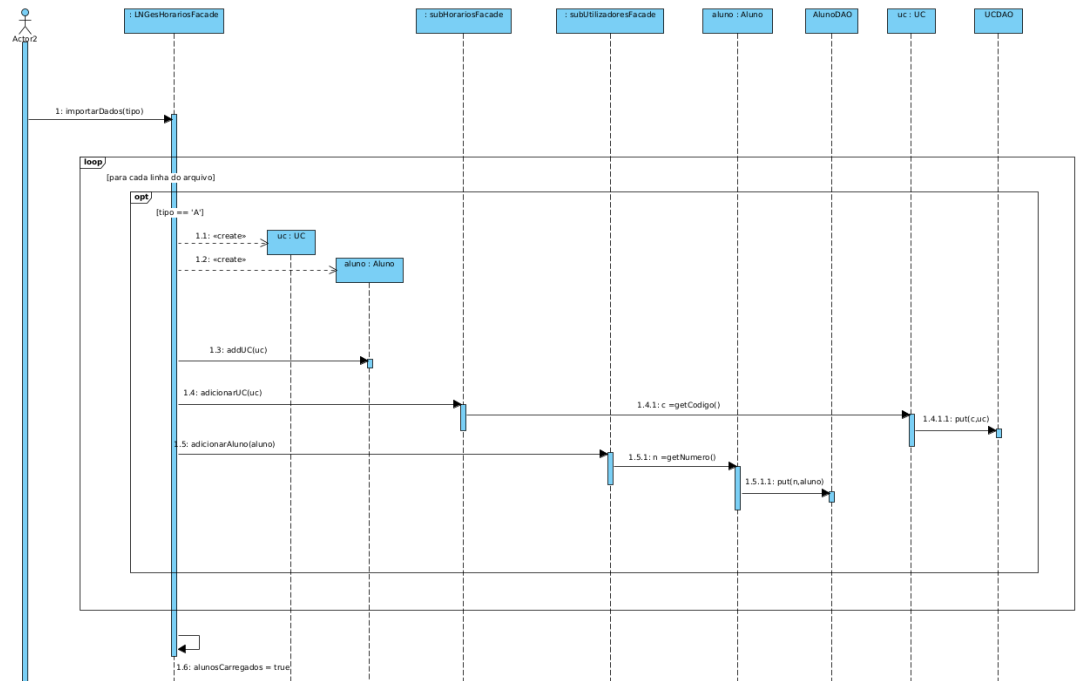


Figura 19 – Diagrama de sequência - Importar Aluno

Diagrama de Sequência para Gerar Horário

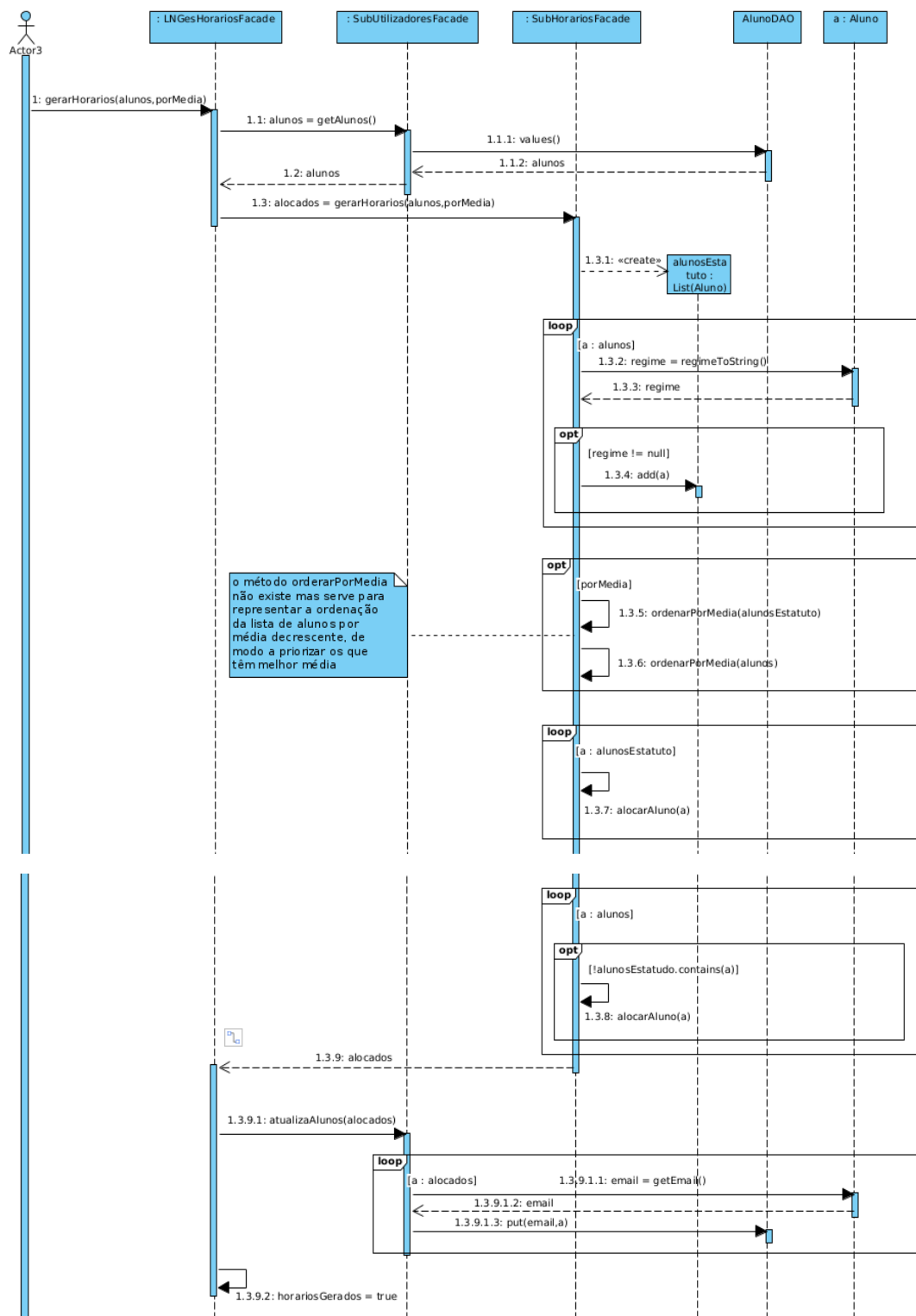


Figura 20 – Diagrama de sequência de Gerar Horários

## Resultados obtidos

Neste projeto desenvolvemos um sistema de gestão de horários com o propósito de permitir ao diretor de curso gerar e modificar horários e de permitir aos alunos consultarem e exportarem o seu horário.

Começamos por criar um modelo de domínio e vários use cases que ajudaram a dar-nos uma ideia do que seria a estrutura a ser implementada, através das entidades e relações dos problemas a serem enfrentados, junto com as funcionalidades e atores do sistema.

Após a entrega na primeira fase, reparamos que já não seria necessário ter um utilizador, portanto decidimos retirá-lo no modelo de domínio, fazendo com que o Diretor de curso passasse a ter um email e um palavra-passe própria, e os Alunos um email institucional e uma palavra-passe.

Após esta alteração decidimos então construir a arquitetura geral do sistema junto com os seus subSistemas representados de uma forma global no diagrama de componentes, e de seguida de uma forma mais detalhada nos diagramas de classes.

Os diagramas de sequência foram desenvolvidos com base nos diagramas de classes, garantindo uma relação entre a sua estrutura e os métodos utilizados na mesma. Ao longo da realização dos mesmos foram feitas umas pequenas alterações visto que ao realizar a implementação do sistema, iam aparecendo novos métodos para utilizar nos diagramas.

Após esta fase decidimos começar a implementar o sistema. Os diagramas desenvolvidos anteriormente foram essenciais, uma vez que tornaram o trabalho muito mais simples e direto. As decisões das estruturas e dos métodos já tinham sido tomadas previamente e foi apenas necessário desenvolvê-las na prática.

Ao longo do desenvolvimento do sistema fomos necessitando de métodos que se revelavam necessários para o bom funcionamento e também fomos fazendo alguns ajustes relativamente à nossa visão inicial de como seria a implementação.

Foram criados 4 DAOs para Alunos, UCs, Turnos e Horários. Escolhemos criar DAOs para os tipos de dados que seriam mais utilizados e modificados e que seriam essenciais ter na base de dados. Cada um deles implementa um `Map<key,value>`, tendo sido implementadas as funções relativas aos maps adaptadas a cada DAO.

Desenvolvemos métodos para importar dados relativos a alunos, ucs, salas e turnos através de ficheiros csv. Foi conseqüentemente necessário definir condições para limitar a ordem de importação dos dados, devido às dependências entra as tabelas da base de dados.

A geração de horários foi feita de seguida possibilitando a alocação dos alunos por ordem decrescente de média ou de forma aleatória, mas sempre dando prioridade aos alunos com estatuto especial. Para cada aluno é criado um horário, mesmo que fique incompleto ou vazio devido a limitações quer do número de vagas nos turnos, quer de sobreposições de turnos.

Para lidar com os alunos com horários incompletos foi criada a funcionalidade de alocação manual. Aqui há a opção de ver a lista dos alunos não alocados e para cada um deles ver o seu horário atual e os turnos que necessita, juntamente com a justificação de não ter sido alocado. Pode-se escolher o turno que se quer alocar num aluno, mesmo que isso implique não respeitar os limites e caso o aluno seja totalmente alocado, sai da lista dos não alocados.

No caso da interface dos alunos estão disponíveis as opções de consultar o horário, que imprime o horário no terminal, e de exportar o horário, que cria um ficheiro txt com o respetivo horário.

O programa acabou por ficar um pouco lento devido aos vários acessos à base de dados, mas as funcionalidades foram bem implementadas e funcionam como eram esperadas.



## Manual de utilização

Inicialmente irá aparecer o seguinte menu:

```
*** SISTEMA DE GESTÃO DE HORÁRIOS ***  
  
1 - Log in  
0 - Sair  
Opção:
```

Ao escolhermos a opção de log in podemos colocar o email e a password:

```
*** LOGIN ***  
Introduza o email institucional:  
a2000000@alunos.uminho.pt  
Introduza a password:  
2000002|
```

Caso entremos como Diretor de Curso aparece o seu menu:

*** MENU DC ***	*** MENU DC ***
1 - Importar alunos/ucs	1 - Importar alunos/ucs
2 - Importar salas	2 - Importar salas
3 - Importar Turnos	3 - ---
4 - Gerar horário	4 - ---
5 - Alocar manualmente	5 - ---
0 - Sair	0 - Sair
Opção:	Opção:

À esquerda é o menu com todas opções as disponíveis, enquanto que à esquerda é o menu quando ainda não se importou dados. As opções vão ficando disponíveis à

medida que as condições se verifiquem (ex: só dá gerar horários depois de se importar todos os dados).

Nas opções de importação de dados apenas aparece a confirmação de que a importação foi bem sucedida.

```
*** IMPORTAR DADOS ***  
  
Alunos e UCs importados  
  
Press any key to continue
```

Na geração de horários podemos escolher entre alocar por média ou alocar aleatoriamente e de seguida é mostrada a confirmação da geração.

```
*** GERAR HORÁRIOS ***  
  
1 - Alocar alunos por média  
2 - Alocar alunos aleatoriamente  
0 - Sair  
Opção: 2|
```

No caso da alocação manual é necessário indicar o email do aluno a alocar e depois uma das UCs para posteriormente escolher um dos turnos disponíveis.

```
*** ALOCAR MANUALMENTE ***
```

```
Introduza o email do aluno que quer alocar:
```

```
a200075@alunos.uminho.pt
```

```
Horário incompleto do aluno:
```

```
MONDAY:
```

```
J305N3 -> Turno TP15: 14:00 - Ed 2 2.12
```

```
WEDNESDAY:
```

```
J305N2 -> Turno T6: 14:00 - Ed 2 1.03
```

```
FRIDAY:
```

```
J305N3 -> Turno T12: 18:00 - Ed 1 0.22
```

```
UCs não alocadas:
```

```
J305N2 Programação Funcional - turno TP/PL
```

```
Introduza o código da UC que quer alocar:
```

```
J305N2
```

```
Turnos:
```

```
8 - THURSDAY 16:00 : limite de alunos do turno excedido
```

```
9 - FRIDAY 17:00 : limite de alunos do turno excedido e sobrepõe outro turno do horário
```

```
11 - TUESDAY 16:00 : limite de alunos do turno excedido
```

```
21 - FRIDAY 12:00 : limite de alunos do turno excedido
```

```
22 - TUESDAY 14:00 : limite de alunos do turno excedido
```

```
Introduza o número do turno que quer alocar:
```

```
11
```

No caso de se fazer login com um aluno aparece o menu com as suas opções.

```
*** MENU ALUNO ***
```

```
1 - Consultar horário
```

```
2 - Exportar horário
```

```
0 - Sair
```

```
Opção: |
```

Na opção de consultar horário é impresso no terminal o horário e na opção de exportar horário é criado o ficheiro txt com o horário, mostrando a confirmação da criação do mesmo.

Consultar horário:

```
*** MENU ALUNO ***

1 - Consultar horário
2 - Exportar horário
0 - Sair
Opção: 1
*** HORÁRIO ***

MONDAY:
J304N3 -> Turno T1: 09:00 - Ed 1 0.08
WEDNESDAY:
J304N3 -> Turno TP5: 08:00 - Ed 2 0.07
```

Exportar horário:

```
*** MENU ALUNO ***

1 - Consultar horário
2 - Exportar horário
0 - Sair
Opção: 2
*** EXPORTAR HORÁRIO ***

Horário exportado

Press any key to continue
```