



Universidade do Minho

Mestrado Integrado em Engenharia Informática

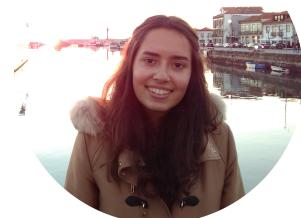
Desenvolvimento de Sistemas de Software

Sistema de Gestão de Turnos Práticos

Grupo 7



Hugo Oliveira (a78565)



Maria Valente (a79093)



Paula Pereira (a77672)



Rui Calheno (a78085)

30 de Dezembro de 2017

Conteúdo

1	Introdução	1
2	Descrição do Problema	1
3	Explicação da abordagem	2
4	Modelo de Domínio	4
4.1	Entidades	5
4.2	Entidades relevantes na aplicação	6
5	Diagrama de Use Case	8
5.1	Atores	9
5.1.1	Aluno	9
5.1.2	Trabalhador Estudante	9
5.1.3	Docente	9
5.1.4	Docente Responsável	9
5.1.5	Direção de curso	9
5.2	Use Cases	10
5.2.1	Iniciar Sessão	10
5.2.2	Registar aluno	11
5.2.3	Registar UC	12
5.2.4	Atribuir turnos	13
5.2.5	Mudar fase	13
5.2.6	Definir capacidade de um turno	14
5.2.7	Registar Faltas	14
5.2.8	Remover Aluno de um Turno	15
5.2.9	Adicionar Aluno a um Turno	16
5.2.10	Ver turnos	17
5.2.11	Aceder ao histórico de uma UC	17
5.2.12	Enviar pedido de troca	18
5.2.13	Aceitar pedido	19
5.2.14	Mudar de turno	20
5.2.15	Terminar Sessão	21
6	Diagramas de Sequência de Sistemas	22
6.1	Iniciar Sessão	22
6.2	Registar Aluno	22
6.3	Registar UC	23
6.4	Atribuir turnos	23
6.5	Mudar fase	24
6.6	Definir capacidade de um turno	24
6.7	Registar Faltas	25

6.8	Remover Aluno de um Turno	25
6.9	Adicionar Aluno a um Turno	26
6.10	Ver turnos	27
6.11	Aceder ao histórico de uma UC	27
6.12	Enviar pedido de troca	28
6.13	Aceitar pedido	29
6.14	Mudar de turno	30
6.15	Terminar Sessão	30
7	Diagramas de Sequência de Subsistemas	31
7.1	Iniciar Sessão	32
7.2	Registar Aluno	33
7.3	Registar UC	34
7.4	Atribuir turnos	35
7.5	Mudar fase	36
7.6	Definir capacidade de um turno	37
7.7	Registar Faltas	38
7.8	Remover Aluno de um Turno	39
7.9	Adicionar Aluno a um Turno	40
7.10	Ver turnos	41
7.11	Aceder ao histórico da UC	42
7.12	Enviar pedido de troca	43
7.13	Aceitar pedido	44
7.14	Mudar de turno	45
7.15	Terminar Sessão	46
8	Diagrama de Classes	47
8.1	Diagrama de Classes	48
8.2	Diagram de ORM	49
9	Diagramas de Packages	50
10	Diagramas de Sequência de implementação	51
10.1	Iniciar Sessão	51
10.2	Registar Aluno	52
10.3	Registar UC	53
10.4	Mudar fase	54
10.5	Definir capacidade de um turno	55
10.6	Registar Faltas	56
10.7	Remover Aluno de um Turno	57
10.8	Adicionar Aluno a um Turno	58
10.9	Ver turnos	59
10.10	Aceder ao histórico da UC	60
10.11	Enviar pedido de troca	61

10.12 Aceitar pedido	62
10.13 Mudar de turno	63
10.14 Terminar Sessão	64
11 Diagramas de Estado	65
11.1 Mapa de Navegação	65
11.1.1 Menu do Aluno	66
11.1.2 Menu do Docente	67
11.1.3 Menu da Direção	68
11.1.4 Menu de Pedidos Pendentes	69
11.1.5 Registar Faltas	69
11.1.6 Estados de um Pedido	70
12 Diagrama de Instalação	71
13 Idealização da interface gráfica na primeira fase do projeto	72
13.1 Aluno	72
13.2 Docente	75
13.3 Direção de Curso	77
14 Interface final	78
14.1 Iniciar Sessão	78
14.2 Menu Principal	78
14.2.1 Alunos	79
14.2.2 Docentes	80
14.2.3 Direção de curso	80
14.3 Menus dos alunos	81
14.3.1 Menu Trocar Turno	81
14.3.2 Menu Pedidos Pendentes	81
14.3.3 Menu Mudar Turno	82
14.3.4 Notificações do aluno	82
14.4 Menus dos docentes	83
14.4.1 Menu Registar Faltas	83
14.4.2 Menu Ver Histórico	83
14.4.3 Menu Editar Capacidade	84
14.4.4 Menu Adicionar Aluno	84
14.4.5 Menu Remover Aluno	85
14.4.6 Menu Pedidos Pendentes	85
14.4.7 Notificações do docente	86
14.5 Menus da direção de curso	86
14.5.1 Menu Ver Alunos	86
14.5.2 Menu Ver UCs	87
14.5.3 Menu Registar Aluno	87
14.5.4 Menu Registar UC	88

14.5.5 Menu Mudar Fase	88
15 Base de Dados do projeto	89
16 Conclusão	90
17 Anexos	92
17.1 Script de Criação MySQL	92

Figuras

1	Modelo de Domínio.	4
2	Modelo de Use case.	8
3	Especificação do Use Case "Iniciar sessão".	10
4	Especificação do Use Case "Registar Aluno".	11
5	Especificação do Use Case "Registar UC".	12
6	Especificação do Use Case "Atribuir turnos".	13
7	Especificação do Use Case "Mudar fase".	13
8	Especificação do Use Case "Definir capacidade de um turno".	14
9	Especificação do Use Case "Registar Faltas".	14
10	Especificação do Use Case "Remover Aluno de um Turno".	15
11	Especificação do Use Case "Adicionar Aluno a um Turno".	16
12	Especificação do Use Case "Ver turnos".	17
13	Especificação do Use Case "Aceder ao histórico de uma UC".	17
14	Especificação do Use Case "Enviar pedido de troca".	18
15	Especificação do Use Case "Aceitar pedido de troca".	19
16	Especificação do Use Case "Mudar de turno".	20
17	Especificação do Use Case "Terminar sessão".	21
18	Diagrama de sequencia do Use Case "Iniciar sessão".	22
19	Diagrama de sequencia do Use Case "Registar Aluno".	22
20	Diagrama de sequencia do Use Case "Registar UC".	23
21	Diagrama de sequencia do Use Case "Atribuir turnos".	23
22	Diagrama de sequencia do Use Case "Mudar fase".	24
23	Diagrama de sequencia do Use Case "Definir capacidade de um turno".	24
24	Diagrama de sequencia do Use Case "Registar Faltas".	25
25	Diagrama de sequencia do Use Case "Remover Aluno de um Turno".	25
26	Diagrama de sequencia do Use Case "Adicionar Aluno a um Turno".	26
27	Diagrama de sequencia do Use Case "Ver turnos".	27
28	Diagrama de sequencia do Use Case "Aceder ao histórico de uma UC".	27
29	Diagrama de sequencia do Use Case "Enviar pedido de troca".	28
30	Diagrama de sequencia do Use Case "Aceitar pedido".	29
31	Diagrama de sequencia do Use Case "Mudar de turno".	30
32	Diagrama de sequencia do Use Case "Terminar sessão".	30
33	Modelo de subsistemas utilizado.	31
34	Diagrama de sequencia de subsistemas do Use Case "Iniciar sessão".	32
35	Diagrama de sequencia de subsistemas do Use Case "Registar Aluno".	33
36	Diagrama de sequencia de subsistemas do Use Case "Registar UC".	34
37	Diagrama de sequencia de subsistemas do Use Case "Atribuir turnos".	35
38	Diagrama de sequencia de subsistemas do Use Case "Mudar fase".	36
39	Diagrama de sequencia de subsistemas do Use Case "Definir capacidade de um turno".	37
40	Diagrama de sequencia de subsistemas do Use Case "Registar faltas".	38

41	Diagrama de sequencia de subsistemas do Use Case "Remover Aluno de um Turno".	39
42	Diagrama de sequencia de subsistemas do Use Case "Adicionar Aluno a um Turno".	40
43	Diagrama de sequencia de subsistemas do Use Case "Ver turnos".	41
44	Diagrama de sequencia de subsistemas do Use Case "Aceder ao histórico da UC".	42
45	Diagrama de sequencia de subsistemas do Use Case "Enviar pedido de troca".	43
46	Diagrama de sequencia de subsistemas do Use Case "Aceitar pedido".	44
47	Diagrama de sequencia de subsistemas do Use Case "Mudar de turno".	45
48	Diagrama de sequencia de subsistemas do Use Case "Terminar sessão".	46
49	Diagrama de classes.	48
50	Diagrama de ORM.	49
51	Diagrama de Packages.	50
52	Diagrama de sequencia de implementação do Use Case "Iniciar sessão".	51
53	Diagrama de sequencia de implementação do Use Case "Registar Aluno".	52
54	Diagrama de sequencia de implementação do Use Case "Registar UC".	53
55	Diagrama de sequencia de implementação do Use Case "Mudar fase".	54
56	Diagrama de sequencia de implementação do Use Case "Definir capacidade de um turno".	55
57	Diagrama de sequencia de implementação do Use Case "Registar faltas".	56
58	Diagrama de sequencia de implementação do Use Case "Remover Aluno de um Turno".	57
59	Diagrama de sequencia de implementação do Use Case "Adicionar Aluno a um Turno".	58
60	Diagrama de sequencia de implementação do Use Case "Ver turnos".	59
61	Diagrama de sequencia de implementação do Use Case "Aceder ao histórico da UC".	60
62	Diagrama de sequencia de implementação do Use Case "Enviar pedido de troca".	61
63	Diagrama de sequencia de implementação do Use Case "Aceitar pedido".	62
64	Diagrama de sequencia de implementação do Use Case "Mudar de turno".	63
65	Diagrama de sequencia de implementação do Use Case "Terminar sessão".	64
66	Diagrama de estados - Mapa de Navegação.	65
67	Mapa de Navegação - Menu do Aluno.	66
68	Mapa de Navegação - Menu do Docente.	67
69	Mapa de Navegação - Menu da Direção.	68
70	Menu de Pedidos Pendentes.	69
71	Menu de Registar Faltas.	69
72	Possíveis estados de um pedido.	70
73	Diagrama de Instalação.	71
74	Menu inicial de um aluno.	72

75	Menu de troca de um aluno.	73
76	Turnos de um aluno.	73
77	Pedidos de troca pendentes de um aluno.	74
78	Menu inicial de um docente.	75
79	Edição de turnos por parte de um docente.	75
80	Marcação de faltas por parte de um docente.	76
81	Aceitar Pedidos de troca.	76
82	Consultar o histórico de um turno.	77
83	Menu inicial da direção de curso.	77
84	Menu inicial para iniciar sessão.	78
85	Menu principal do aluno sem estatuto.	79
86	Menu principal do aluno trabalhador estudante.	79
87	Menu principal do docente.	80
88	Menu principal da direção de curso.	80
89	Menu "Trocar Turno".	81
90	Menu "Pedidos Pendentes".	81
91	Menu "Mudar Turno".	82
92	Notificações do aluno.	82
93	Menu "Registar Faltas".	83
94	Menu "Ver Histórico".	83
95	Menu "Editar Capacidade".	84
96	Menu "Adicionar Aluno".	84
97	Menu "Remover Aluno".	85
98	Menu "Pedidos Pendentes".	85
99	Notificações do docente.	86
100	Menu "Ver Alunos".	86
101	Menu "Ver UCs".	87
102	Menu "Registar Aluno".	87
103	Menu "Registar UC".	88
104	Menu "Mudar Fase".	88
105	Modelo lógico da base de dados.	89

Tabelas

1	Entidades e suas definições.	5
1	Entidades e suas definições (Continuação).	6

Resumo

Este documento diz respeito ao projeto desenvolvido na unidade curricular de Desenvolvimento de Sistemas de Software, do curso de Engenharia Informática da Universidade do Minho.

O projeto consiste no desenvolvimento de um sistema de gestão de turnos práticos de um curso.

O presente relatório consiste numa apresentação das decisões tomadas ao longo do desenvolvimento do projeto.

1 Introdução

No inicio de cada ano escolar, as direções de cursos deparam-se com problemas que dizem respeito à atribuição de horários aos alunos, gestão de salas, distribuição de cadeiras pelos professores, entre outros.

Cabe às direções de curso assegurar que todos os alunos tenham um horário que lhes permita assistir às aulas de todas as unidades curriculares a que estão inscritos, sem que hajam sobreposições.

Este trabalho prático tem como objetivo construir um sistema que permite não só alocar os alunos nos respetivos turnos, mas também assegurar que estes possam efetuar trocas no seu horário e evitar sobreposições de aulas.

2 Descrição do Problema

Pretende-se com este trabalho que seja desenvolvido um sistema de gestão dos turnos práticos de um curso (mestrado integrado ou licenciatura).

A aplicação a desenvolver deverá suportar o registo das Unidades Curriculares (UCs) e dos alunos e a gestão da alocação dos alunos aos turnos das UCs.

A alocação dos alunos deverá ser, numa primeira fase, responsabilidade da Direção do Curso. Posteriormente deverá existir uma fase em que os alunos podem realizar trocas entre eles. Aos alunos com estatuto especial (por exemplo, trabalhador-estudante) é dada a possibilidade de mudarem de turno, sem necessidade de trocarem com outro aluno, desde que exista capacidade no turno que pretendem. A capacidade 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 25% das aulas, perde o seu lugar no turno.

O sistema deverá ser robusto, por exemplo, procurando evitar que os alunos não tenham aulas sobrepostas no horário; mas também flexível, por exemplo, sendo capaz de lidar com situações em que a sobreposição é inevitável (podem existir aulas teóricas de anos diferentes em conflito), permitindo que trocas/mudanças efetuadas sejam anuladas ou que, pontualmente, um aluno possa assistir a um turno que não o seu.

Para permitir uma avaliação do desempenho da gestão de turnos, deverá ser disponibilizada informação sobre a evolução dos turnos ao longo do semestre.

Para além dos aspetos descritos, o sistema deve ainda cobrir outras situações que surjam da análise do problema.

3 Explicação da abordagem

Tendo em conta os requisitos estabelecidos, apresentamos agora a nossa abordagem do problema:

A aplicação deve ser utilizada:

1. pelos alunos, para verificar quais os turnos que lhe foram atribuídos, solicitar e gerir trocas de turnos com outros alunos. Caso se trate de um aluno com estatuto de trabalhador estudante, este pode ainda efetuar uma mudança de turno.
2. pelos docentes, para verificar quais os turnos que lhe foram atribuídos, registar as faltas dos alunos nas aulas desse turno e ainda visualizar um histórico da UC. Caso se trate do docente responsável da UC, este deve ser ainda capaz de gerir pedidos de troca e mudanças de turnos, alterar a capacidade dos turnos e adicionar e remover alunos de turnos.
3. pela direção de curso, para registar os alunos, as UC's, alterar a fase, visualizar todos os alunos e UCs do curso e ainda proceder à atribuição de turnos práticos.

Assim sendo, a aplicação funciona da seguinte forma: Inicialmente só têm acesso à aplicação a direção de curso e os professores do mesmo. Cabe à direção de curso registar os alunos, inserindo os seus dados pessoais e académicos e atribuir-lhes uma password. Assim, qualquer utilizador registado, seja direção de curso, docente ou aluno, inicia sessão na aplicação com as suas credenciais.

Para além de ter a função de registar os alunos, na primeira fase, a direção tem também a importante função de registar as UCs a atribuir turnos práticos, pelo que, no inicio do ano, deve inserir toda a informação necessária para tal registo assim como fazer a atribuição dos turnos aos alunos já registados.

Atribuídos os turnos, inicia-se a segunda fase. Durante este período, os alunos consultam os turnos que lhes foram atribuídos e caso não fiquem satisfeitos têm a possibilidade de trocar com outros alunos. Assim, depois de encontrar um colega disponível para realizar essa troca, envia-lhe, na aplicação, um pedido de troca que ao ser aceite leva a que a constituição dos turnos seja alterada. No entanto, se o segundo aluno decidir que não tenciona concretizar a troca, recusa-a e tudo se mantém igual.

Para além de trocas com outros estudantes, nesta fase, e caso se trate de um aluno com estatuto de trabalhador-estudante, este pode mudar para qualquer outro turno, desde que este ainda tenha vagas disponíveis.

Quando a direção de curso determina o inicio da terceira fase, as mudanças e trocas passam a ter que ser aprovadas pelo docente responsável da UC.

Sempre que algum evento altera a constituição dos turnos de uma UC, os alunos envolvidos nessa modificação e o docente responsável pela mesma recebem uma notificação na sua página inicial. Para além disso, o histórico da UC, que pode ser consultado pelo docente responsável, é atualizado.

No que diz respeito à gestão das UC's, o docente responsável tem ainda a função de, caso seja necessário, alterar a capacidade dos vários turnos, desde que essa

capacidade não seja superior ao número de lugares disponíveis na sala onde o turno é lecionado.

Uma das medidas adotadas para assegurar que os alunos que faltam às aulas não estejam a ocupar um lugar num turno que outro aluno poderia preencher, é o controlo de faltas. Tendo isto, cabe ao docente que leciona cada turno registar as faltas dos alunos em cada aula, sendo que quando o limite de faltas é atingido, o seu lugar no turno fica disponível para outro aluno, pelo que o próprio sistema remove esse aluno do turno inicial. Quando isto acontece, o aluno tem de falar pessoalmente com o docente para que lhe seja atribuído um novo turno, sendo que posteriormente, o docente adiciona-o a um turno com vagas disponíveis.

Por fim, quando o utilizador já não tiver necessidade de utilizar a aplicação, pode simplesmente terminar sessão.

4 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 um glossário de termos.

Tendo sido uma das tarefas a que dedicamos mais tempo, as únicas alterações que sofreu foram o acréscimo da entidade "Histórico da UC" e da alteração da entidade "Presença" para "Falta".

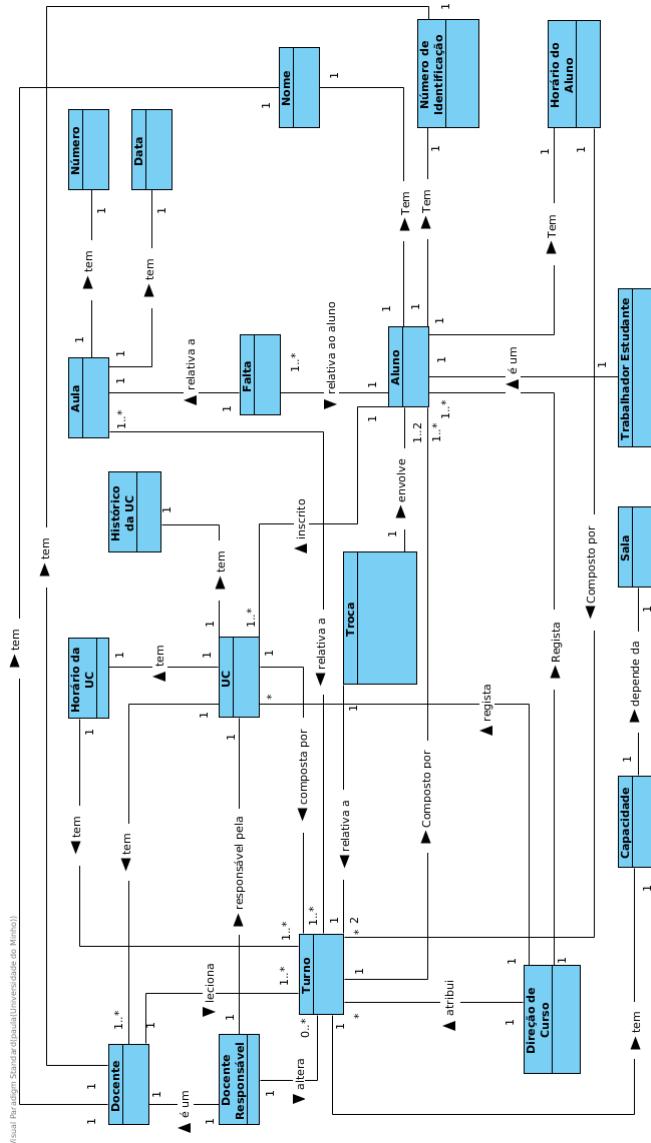


Figura 1: Modelo de Domínio.

4.1 Entidades

Para melhor compreensão do problema e do modelo de domínio apresentado, segue-se uma listagem das entidades, assim como a sua definição.

Entidade	Definição
Docente	Docente que leciona uma qualquer aula de uma determinada UC.
Docente Responsável	Docente responsável por uma determinada UC.
Aluno	Aluno que, frequentando o curso, está inscrito em vários turnos e assiste às aulas relativas aos mesmos.
Trabalhador estudante	Aluno do curso com estatuto especial de trabalhador estudante.
Direção de curso	Conjunto de professores responsável por tomar decisões que visem o bom funcionamento do curso.
UC	Unidade curricular do curso.
Horário da UC	Conjunto de todas os turnos relativas a uma determinada UC, isto incluí aulas teóricas, aulas teórico-práticas e aulas labororiais.
Horário do Aluno	Conjunto de todos os turnos de um aluno.
Aula	Aula relativa a um turno.
Número	Número de uma determinada aula.
Data	Data em que uma aula é lecionada.
Turno	Período de tempo do horário de uma UC frequentado por um conjunto de alunos nele inscrito.
Capacidade	Número máximo de alunos que podem frequentar um determinado turno.

Tabela 1: Entidades e suas definições.

Entidade	Definição
Sala	Sala onde é lecionada uma aula.
Falta	Representa a ausência de um aluno numa aula.
Nome	Nome de um aluno ou de um docente.
Número de Identificação	Número que identifica um aluno ou um docente.
Histórico da UC	Conjunto de todas as alterações relacionadas com a constituição dos turnos.
Troca	Qualquer alteração na constituição dos turnos, pelo que pode corresponder a uma troca entre dois alunos, ou a uma mudança de turno de um aluno com estatuto.

Tabela 1: Entidades e suas definições (Continuação).

4.2 Entidades relevantes na aplicação

Utilizador

Diz respeito a qualquer pessoa que faz uso da aplicação. Possui um nome, um número de identificação e uma password que lhe é atribuída, uma lista de turnos e uma lista de trocas. Fazem parte dos utilizadores:

1. os alunos, que guardam ainda informação sobre o seu estatuto (estudante normal ou trabalhador). Neste caso, a lista de turnos diz respeito aos turnos em que este está inscrito.
2. os docentes, que guardam informação acerca de serem responsáveis ou não da UC. Neste caso, a lista de turnos diz respeito aos turnos pelos quais é responsável.

UC

Diz respeito a qualquer unidade curricular lecionada no curso. Possui um nome, a lista dos seus turnos práticos, o seu histórico e ainda o docente responsável pela Unidade Curricular.

Turno

Diz respeito a um turno prático de uma unidade curricular. Possui um nome, a UC a que diz respeito, o número de alunos atual e a capacidade máxima, o docente que o leciona, os alunos que nele estão inscritos, o número de aulas previstas e as faltas dos alunos.

Troca

Diz respeito a qualquer pedido de troca/mudança de turno, ou seja, troca com algum outro aluno ou mudança de um aluno com estatuto de trabalhador estudante para um turno com vagas. Possui os alunos e turnos envolvidos.

5 Diagrama 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).

A principal diferença do modelo de Use Case agora apresentado para o inicial diz respeito à evidente diferenciação entre os vários tipos de alunos e docentes e as funcionalidades associadas a cada um deles. Para além disso, foram adicionados/modificados/removidos alguns use cases de acordo com as necessidades que surgiram durante o desenvolvimento do projeto.



Figura 2: Modelo de Use case.

5.1 Atores

5.1.1 Aluno

Representa o aluno sem estatuto cujas funcionalidades disponíveis são enviar e aceitar pedidos de troca e ver os turnos em que está inscrito.

5.1.2 Trabalhador Estudante

Representa o aluno com estato de trabalhador estudante que tem disponíveis todas as funcionalidades de um estudante e ainda a possibilidade de mudar de turno.

5.1.3 Docente

Representa o docente de uma determinada UC a quem é permitido ver os turnos que leciona, registar faltas dos alunos e ainda aceder ao histórico da UC.

5.1.4 Docente Responsável

Representa o docente responsável pela UC, pelo que para além de ter todas as funcionalidades de um docente, pode ainda gerir os pedidos de troca e mudança de turnos, alterar a capacidade máxima de um turno e adicionar ou remover um aluno de um determinado turno.

5.1.5 Direção de curso

Representa a direção de curso que tem disponíveis as funcionalidades de registrar UC's, registar alunos e ainda de atribuir os turnos das várias UC's aos alunos.

5.2 Use Cases

Nesta secção apresentamos as especificações dos vários use cases que compõe o modelo de Use Case da figura 2.

5.2.1 Iniciar Sessão

Use Case	Iniciar sessão																						
Descrição	Um utilizador registado no sistema inicia sessão na aplicação.																						
Pré-condição	O utilizador não está autenticado no sistema.																						
Pós-condição	O utilizador tem sessão iniciada no sistema.																						
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Indica que pretende iniciar sessão.</td><td></td></tr> <tr> <td>2</td><td></td><td>Solicita número de identificação e password.</td></tr> <tr> <td>3</td><td>Insere número de identificação e password.</td><td></td></tr> <tr> <td>4</td><td></td><td>Verifica credenciais</td></tr> <tr> <td>5</td><td></td><td>Valida credenciais.</td></tr> <tr> <td>6</td><td></td><td>Confirma acesso.</td></tr> </tbody> </table>		Autor	Sistema	1	Indica que pretende iniciar sessão.		2		Solicita número de identificação e password.	3	Insere número de identificação e password.		4		Verifica credenciais	5		Valida credenciais.	6		Confirma acesso.	
	Autor	Sistema																					
1	Indica que pretende iniciar sessão.																						
2		Solicita número de identificação e password.																					
3	Insere número de identificação e password.																						
4		Verifica credenciais																					
5		Valida credenciais.																					
6		Confirma acesso.																					
Exceção 1 [Credenciais inválidas] (Passo 4)	4.1	Informa que as credenciais não são válidas.																					

Figura 3: Especificação do Use Case "Iniciar sessão".

Este use case reflete a situação em que um qualquer utilizador inicia sessão na aplicação. Para tal, indica o seu número de identificação e a password que lhe foi atribuída, sendo que a sessão apenas é iniciada se as credenciais forem validadas pelo sistema.

5.2.2 Registar aluno

Use Case	Registrar aluno.																	
Descrição	Direção de curso regista novo aluno.																	
Pré-condição	Direção de curso autenticada e aluno não registado no sistema.																	
Pós-condição	Aluno registado com sucesso no sistema.																	
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Insere informações relativas ao novo aluno (Nome, número de identificação, password, estatuto).</td><td></td></tr> <tr> <td>2</td><td></td><td>Verifica informações.</td></tr> <tr> <td>3</td><td></td><td>Regista novo aluno.</td></tr> <tr> <td>4</td><td></td><td>Confirma registo do aluno.</td></tr> </tbody> </table>				Autor	Sistema	1	Insere informações relativas ao novo aluno (Nome, número de identificação, password, estatuto).		2		Verifica informações.	3		Regista novo aluno.	4		Confirma registo do aluno.
	Autor	Sistema																
1	Insere informações relativas ao novo aluno (Nome, número de identificação, password, estatuto).																	
2		Verifica informações.																
3		Regista novo aluno.																
4		Confirma registo do aluno.																
Exceção 1 [Informações não são válidas.] (Passo 2)	2.1		Informa sobre impossibilidade de registrar o novo aluno.															

Figura 4: Especificação do Use Case "Registrar Aluno".

Este use case reflete a situação em que a direção de curso adiciona as informações sobre um novo aluno, ficando estes registado e apto a usar a aplicação.

5.2.3 Registar UC

Use Case	Registrar UC.																
Descrição	Direção de curso regista uma nova UC.																
Pré-condição	Direção de curso autenticada e UC não registada no sistema.																
Pós-condição	UC registada com sucesso no sistema.																
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Insere informações relativas à nova UC (nome, regime de funcionamento, docente responsável).</td><td></td></tr> <tr> <td>2</td><td></td><td>Verifica informações.</td></tr> <tr> <td>3</td><td></td><td>Regista UC no sistema.</td></tr> <tr> <td>4</td><td></td><td>Confirma registo da UC.</td></tr> </tbody> </table>		Autor	Sistema	1	Insere informações relativas à nova UC (nome, regime de funcionamento, docente responsável).		2		Verifica informações.	3		Regista UC no sistema.	4		Confirma registo da UC.	
	Autor	Sistema															
1	Insere informações relativas à nova UC (nome, regime de funcionamento, docente responsável).																
2		Verifica informações.															
3		Regista UC no sistema.															
4		Confirma registo da UC.															
Exceção 1 [Informações inválidas] (Passo 2)	2.1	Informa sobre impossibilidade de registrar a nova UC.															

Figura 5: Especificação do Use Case ”Registrar UC” .

Este use case reflete a situação em que a direção de curso adiciona uma nova UC ao sistema, indicando assim todas as informações sobre a UC.

5.2.4 Atribuir turnos

Use Case	Atribuir turnos.																						
Descrição	A direção de curso atribui turnos aos alunos selecionados.																						
Pré-condição	A direção de curso está autenticada e os alunos estão registados no sistema.																						
Pós-condição	Os alunos selecionados ficam com os turnos atribuídos.																						
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Solicita lista de alunos.</td><td></td></tr> <tr> <td>2</td><td></td><td>Mostra lista de alunos.</td></tr> <tr> <td>3</td><td>Seleciona alunos a atribuir turnos.</td><td></td></tr> <tr> <td>4</td><td></td><td>Atribui turnos aos alunos selecionados.</td></tr> <tr> <td>5</td><td></td><td>Atualiza a constituição dos turnos.</td></tr> <tr> <td>6</td><td></td><td>Confirma atribuição de turnos.</td></tr> </tbody> </table>		Autor	Sistema	1	Solicita lista de alunos.		2		Mostra lista de alunos.	3	Seleciona alunos a atribuir turnos.		4		Atribui turnos aos alunos selecionados.	5		Atualiza a constituição dos turnos.	6		Confirma atribuição de turnos.	
	Autor	Sistema																					
1	Solicita lista de alunos.																						
2		Mostra lista de alunos.																					
3	Seleciona alunos a atribuir turnos.																						
4		Atribui turnos aos alunos selecionados.																					
5		Atualiza a constituição dos turnos.																					
6		Confirma atribuição de turnos.																					

Figura 6: Especificação do Use Case ”Atribuir turnos”.

Este use case reflete a situação em que a direção de curso atribui turnos a um conjunto de alunos indicados pela própria.

5.2.5 Mudar fase

Use Case	Mudar fase.																
Descrição	A direção de curso altera a fase em que o sistema se encontra.																
Pré-condição	A direção de curso está autenticada no sistema.																
Pós-condição	Fase alterada com sucesso.																
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Indica que pretende alterar a fase.</td><td></td></tr> <tr> <td>2</td><td>Indica a nova fase.</td><td></td></tr> <tr> <td>3</td><td></td><td>Regista alteração no sistema.</td></tr> <tr> <td>4</td><td></td><td>Informa que a fase foi alterada com sucesso.</td></tr> </tbody> </table>		Autor	Sistema	1	Indica que pretende alterar a fase.		2	Indica a nova fase.		3		Regista alteração no sistema.	4		Informa que a fase foi alterada com sucesso.	
	Autor	Sistema															
1	Indica que pretende alterar a fase.																
2	Indica a nova fase.																
3		Regista alteração no sistema.															
4		Informa que a fase foi alterada com sucesso.															

Figura 7: Especificação do Use Case ”Mudar fase”.

Este use case reflete a situação em que a direção de curso altera a fase em que o sistema se encontra, o que condiciona o forma como as trocas se processam.

5.2.6 Definir capacidade de um turno

Use Case	Definir capacidade de um turno.																
Descrição	Docente responsável define uma capacidade para um determinado turno.																
Pré-condição	O docente responsável está autenticado no sistema.																
Pós-condição	A capacidade do turno foi alterada com sucesso.																
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Indica nova capacidade.</td> <td></td> </tr> <tr> <td>2</td> <td></td> <td>Valida capacidade.</td> </tr> <tr> <td>3</td> <td></td> <td>Atualiza capacidade do turno.</td> </tr> <tr> <td>4</td> <td></td> <td>Confirma alteração.</td> </tr> </tbody> </table>			Autor	Sistema	1	Indica nova capacidade.		2		Valida capacidade.	3		Atualiza capacidade do turno.	4		Confirma alteração.
	Autor	Sistema															
1	Indica nova capacidade.																
2		Valida capacidade.															
3		Atualiza capacidade do turno.															
4		Confirma alteração.															
Exceção 1 [Nova capacidade excede limite da sala] (Passo 2)	2.1	Informa que a nova capacidade não é válida.															
Exceção 2 [Número de alunos superior a nova capacidade] (Passo 2)	2.1	Informa que o número de alunos atual é superior à nova capacidade.															

Figura 8: Especificação do Use Case "Definir capacidade de um turno".

Este use case reflete a situação em que o docente responsável pela UC define uma nova capacidade para um determinado turno.

5.2.7 Registar Faltas

Use Case	Registrar faltas.																
Descrição	Docente regista faltas dos alunos numa aula.																
Pré-condição	O docente está autenticado no sistema.																
Pós-condição	Faltas registadas com sucesso.																
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Seleciona os alunos que faltaram.</td> <td></td> </tr> <tr> <td>2</td> <td>Submete as faltas.</td> <td></td> </tr> <tr> <td>3</td> <td></td> <td>Regista faltas.</td> </tr> <tr> <td>4</td> <td></td> <td>Informa que o registo foi efetuado com sucesso.</td> </tr> </tbody> </table>			Autor	Sistema	1	Seleciona os alunos que faltaram.		2	Submete as faltas.		3		Regista faltas.	4		Informa que o registo foi efetuado com sucesso.
	Autor	Sistema															
1	Seleciona os alunos que faltaram.																
2	Submete as faltas.																
3		Regista faltas.															
4		Informa que o registo foi efetuado com sucesso.															

Figura 9: Especificação do Use Case "Registrar Faltas".

Este use case reflete a situação em que o docente de um determinado turno regista as faltas dos alunos de uma determinada aula.

5.2.8 Remover Aluno de um Turno

Use Case	Remover aluno de um turno.																						
Descrição	Docente responsável remove o aluno de um turno.																						
Pré-condição	Docente responsável está autenticado no sistema.																						
Pós-condição	O aluno não está inscrito no turno.																						
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Seleciona UC.</td><td></td></tr> <tr> <td>2</td><td>Seleciona turno.</td><td></td></tr> <tr> <td>3</td><td></td><td>Apresenta lista de alunos do turno.</td></tr> <tr> <td>4</td><td>Indica aluno que quer remover.</td><td></td></tr> <tr> <td>5</td><td></td><td>Remove aluno do turno.</td></tr> <tr> <td>6</td><td></td><td>Informa que aluno foi removido.</td></tr> </tbody> </table>		Autor	Sistema	1	Seleciona UC.		2	Seleciona turno.		3		Apresenta lista de alunos do turno.	4	Indica aluno que quer remover.		5		Remove aluno do turno.	6		Informa que aluno foi removido.	
	Autor	Sistema																					
1	Seleciona UC.																						
2	Seleciona turno.																						
3		Apresenta lista de alunos do turno.																					
4	Indica aluno que quer remover.																						
5		Remove aluno do turno.																					
6		Informa que aluno foi removido.																					

Figura 10: Especificação do Use Case ”Remover Aluno de um Turno”.

Este use case reflete a situação e que o docente responsável por uma UC remove um aluno de um turno em que este estava inscrito.

5.2.9 Adicionar Aluno a um Turno

Use Case	Adicionar aluno a um turno.																															
Descrição	O docente responsável de uma UC adiciona um aluno a um turno.																															
Pré-condição	O docente responsável está autenticado no sistema.																															
Pós-condição	O aluno é adicionado ao turno.																															
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Indica UC.</td><td></td></tr> <tr> <td>2</td><td>Indica turno.</td><td></td></tr> <tr> <td>3</td><td>Indica número de identificação do aluno a adicionar.</td><td></td></tr> <tr> <td>4</td><td></td><td>Verifica se o aluno existe.</td></tr> <tr> <td>5</td><td></td><td>Apresenta informação sobre o aluno indicado.</td></tr> <tr> <td>6</td><td></td><td>Solicita que confirme que se trata do aluno correto.</td></tr> <tr> <td>7</td><td>Confirma aluno.</td><td></td></tr> <tr> <td>8</td><td></td><td>Adiciona aluno ao turno indicado.</td></tr> <tr> <td>9</td><td></td><td>Informa que o aluno foi adicionado com sucesso ao turno.</td></tr> </tbody> </table>		Autor	Sistema	1	Indica UC.		2	Indica turno.		3	Indica número de identificação do aluno a adicionar.		4		Verifica se o aluno existe.	5		Apresenta informação sobre o aluno indicado.	6		Solicita que confirme que se trata do aluno correto.	7	Confirma aluno.		8		Adiciona aluno ao turno indicado.	9		Informa que o aluno foi adicionado com sucesso ao turno.	
	Autor	Sistema																														
1	Indica UC.																															
2	Indica turno.																															
3	Indica número de identificação do aluno a adicionar.																															
4		Verifica se o aluno existe.																														
5		Apresenta informação sobre o aluno indicado.																														
6		Solicita que confirme que se trata do aluno correto.																														
7	Confirma aluno.																															
8		Adiciona aluno ao turno indicado.																														
9		Informa que o aluno foi adicionado com sucesso ao turno.																														
Exceção 1 [Aluno não existe] (Passo 4)	4.1	Informa que o número indicado não corresponde a nenhum aluno.																														
Exceção 2 [Não confirma que se trata do aluno certo] (Passo 7)	7.1	Informa sobre a impossibilidade de adicionar aluno ao turno.																														
Exceção 3 [Aluno já possui turno] (Passo 8)	8.1	Informa que o aluno já possui um turno atribuído.																														
Exceção 4 [Não existem vagas disponíveis] (Passo 8)	8.1	Informa que não existem vagas disponíveis no turno indicado.																														

Figura 11: Especificação do Use Case ”Adicionar Aluno a um Turno”.

Este use case reflete a situação em que o docente responsável por uma UC adiciona um aluno a um turno com vagas disponíveis.

5.2.10 Ver turnos

Use Case	Ver Turnos.		
Descrição	Utilizador vê os seus turnos.		
Pré-condição	Utilizador está autenticado no sistema.		
Pós-condição	Os turnos do utilizador são apresentados.		
Comportamento Normal		Autor	Sistema
	1	Solicita ver os turnos os seus turnos.	
	2		Apresenta os turnos do utilizador.

Figura 12: Especificação do Use Case "Ver turnos".

Este use case reflete a situação em que um utilizador vê os seus turnos. No caso de um aluno, este conjunto reflete os turnos em que o aluno está inscrito. Se se tratar de um docente, são apresentados os turnos que este leciona ou todos os turnos de uma UC, se se tratar do responsável da UC.

5.2.11 Aceder ao histórico de uma UC

Use Case	Aceder ao histórico da UC.		
Descrição	Docente acede ao histórico de uma das suas UC.		
Pré-condição	Docente está autenticado.		
Pós-condição	O histórico é apresentado ao docente.		
Comportamento Normal		Autor	Sistema
	1	Seleciona UC.	
	2	Solicita ver histórico da UC.	
	3		Apresenta informação sobre o histórico da UC.
Exceção 1 [Informação não está disponível] (Passo 3)	3.1		Informa que o histórico da UC não está disponível.

Figura 13: Especificação do Use Case "Aceder ao histórico de uma UC".

Este use case reflete a situação em que um docente acede ao histórico de uma das UCs que leciona.

5.2.12 Enviar pedido de troca

Use Case	Enviar pedido de troca.																			
Descrição	Aluno envia um pedido de troca de turno a outro aluno.																			
Pré-condição	O aluno está autenticado no sistema.																			
Pós-condição	O pedido de troca foi enviado com sucesso.																			
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Seleciona UC.</td><td></td></tr> <tr> <td>2</td><td>Seleciona turno.</td><td></td></tr> <tr> <td>3</td><td>Seleciona aluno com quem pretende trocar.</td><td></td></tr> <tr> <td>4</td><td></td><td>Envia pedido de troca para o segundo aluno.</td></tr> <tr> <td>5</td><td></td><td>Informa que o pedido foi enviado.</td></tr> </tbody> </table>		Autor	Sistema	1	Seleciona UC.		2	Seleciona turno.		3	Seleciona aluno com quem pretende trocar.		4		Envia pedido de troca para o segundo aluno.	5		Informa que o pedido foi enviado.	
	Autor	Sistema																		
1	Seleciona UC.																			
2	Seleciona turno.																			
3	Seleciona aluno com quem pretende trocar.																			
4		Envia pedido de troca para o segundo aluno.																		
5		Informa que o pedido foi enviado.																		

Figura 14: Especificação do Use Case "Enviar pedido de troca".

Este use case reflete a situação em que um aluno efetua um pedido de troca de turno. Para tal, o aluno seleciona a UC, o turno para o qual pretende trocar e o aluno com quem ele pretende trocar.

5.2.13 Aceitar pedido

Use Case	Aceitar pedido.																												
Descrição	Utilizador aceita pedido.																												
Pré-condição	O utilizador está autenticado no sistema.																												
Pós-condição	Na fase 2, a troca ou mudança é efetuada no sistema, ou, na fase 3, o utilizador é aluno e o pedido é enviado para o docente responsável da UC.																												
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Solicita ver pedidos pendentes.</td><td></td></tr> <tr> <td>2</td><td></td><td>Apresenta pedidos pendentes.</td></tr> <tr> <td>3</td><td>Seleciona pedido.</td><td></td></tr> <tr> <td>4</td><td></td><td>Apresenta pedido.</td></tr> <tr> <td>5</td><td>Aceita pedido.</td><td></td></tr> <tr> <td>6</td><td></td><td>Regista pedido no Sistema.</td></tr> <tr> <td>7</td><td></td><td>Notifica os alunos envolvidos na troca que esta foi aceite.</td></tr> <tr> <td>8</td><td></td><td>Informa que a troca foi efetuada.</td></tr> </tbody> </table>		Autor	Sistema	1	Solicita ver pedidos pendentes.		2		Apresenta pedidos pendentes.	3	Seleciona pedido.		4		Apresenta pedido.	5	Aceita pedido.		6		Regista pedido no Sistema.	7		Notifica os alunos envolvidos na troca que esta foi aceite.	8		Informa que a troca foi efetuada.	
	Autor	Sistema																											
1	Solicita ver pedidos pendentes.																												
2		Apresenta pedidos pendentes.																											
3	Seleciona pedido.																												
4		Apresenta pedido.																											
5	Aceita pedido.																												
6		Regista pedido no Sistema.																											
7		Notifica os alunos envolvidos na troca que esta foi aceite.																											
8		Informa que a troca foi efetuada.																											
Exceção 1 [Não aceita pedido troca e utilizador é Aluno] (Passo 5)	<table border="1"> <tr> <td>5.1</td><td></td><td>Notifica o outro aluno que troca não foi aceite.</td></tr> <tr> <td>5.2</td><td></td><td>Informa que o outro aluno foi notificado.</td></tr> </table>	5.1		Notifica o outro aluno que troca não foi aceite.	5.2		Informa que o outro aluno foi notificado.																						
5.1		Notifica o outro aluno que troca não foi aceite.																											
5.2		Informa que o outro aluno foi notificado.																											
Exceção 2 [Não aceita pedido de troca e utilizador é docente responsável da UC] (Passo 5)	<table border="1"> <tr> <td>5.1</td><td></td><td>Notifica os alunos envolvidos na troca que esta não foi aceite.</td></tr> <tr> <td>5.2</td><td></td><td>Informa que os alunos foram notificados.</td></tr> </table>	5.1		Notifica os alunos envolvidos na troca que esta não foi aceite.	5.2		Informa que os alunos foram notificados.																						
5.1		Notifica os alunos envolvidos na troca que esta não foi aceite.																											
5.2		Informa que os alunos foram notificados.																											
Exceção 2 [Não aceita pedido de mudança e utilizador é docente responsável da UC] (Passo 5)	<table border="1"> <tr> <td>5.1</td><td></td><td>Notifica o aluno que a mudança não foi aceite.</td></tr> <tr> <td>5.2</td><td></td><td>Informa que o aluno foi notificado.</td></tr> </table>	5.1		Notifica o aluno que a mudança não foi aceite.	5.2		Informa que o aluno foi notificado.																						
5.1		Notifica o aluno que a mudança não foi aceite.																											
5.2		Informa que o aluno foi notificado.																											
Alternativa 1 [Estamos na Fase 3 e utilizador é Aluno] (Passo 6)	<table border="1"> <tr> <td>5.1</td><td></td><td>Envia o pedido de troca para o docente responsável da UC.</td></tr> <tr> <td>5.2</td><td></td><td>Informa que o pedido foi enviado ao docente.</td></tr> </table>	5.1		Envia o pedido de troca para o docente responsável da UC.	5.2		Informa que o pedido foi enviado ao docente.																						
5.1		Envia o pedido de troca para o docente responsável da UC.																											
5.2		Informa que o pedido foi enviado ao docente.																											
Alternativa 2 [Aceita pedido de troca e utilizador é aluno] (Passo 7)	<table border="1"> <tr> <td>7.1</td><td></td><td>Notifica o outro aluno que a troca foi aceite.</td></tr> </table>	7.1		Notifica o outro aluno que a troca foi aceite.																									
7.1		Notifica o outro aluno que a troca foi aceite.																											
Alternativa 3 [Aceita pedido de mudança e utilizador é docente responsável da UC] (Passo 7)	<table border="1"> <tr> <td>7.1</td><td></td><td>Notifica o aluno que a mudança foi aceite.</td></tr> </table>	7.1		Notifica o aluno que a mudança foi aceite.																									
7.1		Notifica o aluno que a mudança foi aceite.																											

Figura 15: Especificação do Use Case "Aceitar pedido de troca".

Este use case reflete a situação em que um utilizador aceita um pedido qualquer que possa alterar a constituição dos turnos. Assim, um pedido pode corresponder a uma troca entre dois alunos, e pode ser aceite por um aluno ou por um docente, ou a uma mudança de turno de um aluno com estatuto, que, caso se trate da fase 2, pode ser aceite por um docente.

5.2.14 Mudar de turno

Use Case	Mudar de turno.																			
Descrição	Aluno com estatuto trabalhador-estudante muda de turno.																			
Pré-condição	O aluno está autenticado no sistema e é trabalhador estudante.																			
Pós-condição	Na fase 2, a mudança é efetuada no sistema, ou, na fase 3, o pedido de mudança é enviado para o Docente responsável.																			
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Seleciona UC.</td><td></td></tr> <tr> <td>2</td><td>Seleciona Turno.</td><td></td></tr> <tr> <td>3</td><td></td><td>Verifica se existem vagas disponíveis no turno.</td></tr> <tr> <td>4</td><td></td><td>Regista mudança no Sistema.</td></tr> <tr> <td>5</td><td></td><td>Informa o aluno que a mudança foi efetuada.</td></tr> </tbody> </table>		Autor	Sistema	1	Seleciona UC.		2	Seleciona Turno.		3		Verifica se existem vagas disponíveis no turno.	4		Regista mudança no Sistema.	5		Informa o aluno que a mudança foi efetuada.	
	Autor	Sistema																		
1	Seleciona UC.																			
2	Seleciona Turno.																			
3		Verifica se existem vagas disponíveis no turno.																		
4		Regista mudança no Sistema.																		
5		Informa o aluno que a mudança foi efetuada.																		
Alternativa 1 [Estamos na fase 3] (Passo 4)	<table border="1"> <tr> <td>4.1</td> <td></td> <td>Envia pedido de mudança para o Docente responsável da UC.</td> </tr> <tr> <td>4.2</td> <td></td> <td>Informa que o pedido foi enviado ao Docente Responsável.</td> </tr> </table>	4.1		Envia pedido de mudança para o Docente responsável da UC.	4.2		Informa que o pedido foi enviado ao Docente Responsável.													
4.1		Envia pedido de mudança para o Docente responsável da UC.																		
4.2		Informa que o pedido foi enviado ao Docente Responsável.																		
Exceção 1 [Não existem vagas disponíveis] (Passo 3)	<table border="1"> <tr> <td>3.1</td> <td></td> <td>Informa que não existe disponibilidade no turno em questão.</td> </tr> </table>	3.1		Informa que não existe disponibilidade no turno em questão.																
3.1		Informa que não existe disponibilidade no turno em questão.																		

Figura 16: Especificação do Use Case "Mudar de turno".

Este use case reflete a situação em que um aluno com estatuto de trabalhador estudante troca para um turno com vagas disponíveis, sem que isso implique uma troca com outro aluno.

5.2.15 Terminar Sessão

Use Case	Terminar sessão.										
Descrição	Utilizador termina sessão na aplicação.										
Pré-condição	O utilizador está autenticado no sistema.										
Pós-condição	O utilizador deixa de estar autenticado no sistema.										
Comportamento Normal	<table border="1"> <thead> <tr> <th></th> <th>Autor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Informa que pretende terminar sessão na aplicação.</td><td></td></tr> <tr> <td>2</td><td></td><td>Termina sessão do utilizador.</td></tr> </tbody> </table>		Autor	Sistema	1	Informa que pretende terminar sessão na aplicação.		2		Termina sessão do utilizador.	
	Autor	Sistema									
1	Informa que pretende terminar sessão na aplicação.										
2		Termina sessão do utilizador.									

Figura 17: Especificação do Use Case "Terminar sessão".

Este use case reflete a situação em que um qualquer utilizador termina a sua sessão na aplicação, deixando de ter acesso às funcionalidades que esta oferece.

6 Diagramas de Sequência de Sistemas

Nesta secção iremos apresentar todos os diagramas de sequencia de sistemas associados aos *use cases* acima apresentados, diagramas esses que refletem o modo de funcionamento da aplicação.

6.1 Iniciar Sessão

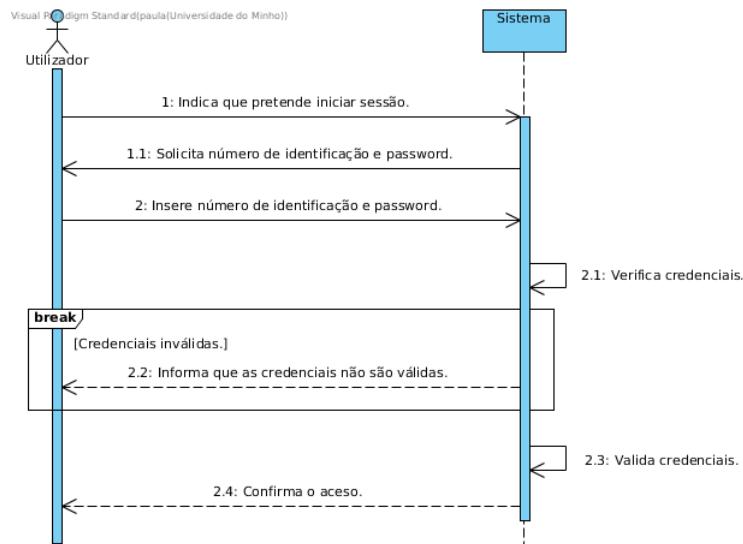


Figura 18: Diagrama de sequencia do Use Case ”Iniciar sessão”.

6.2 Registar Aluno

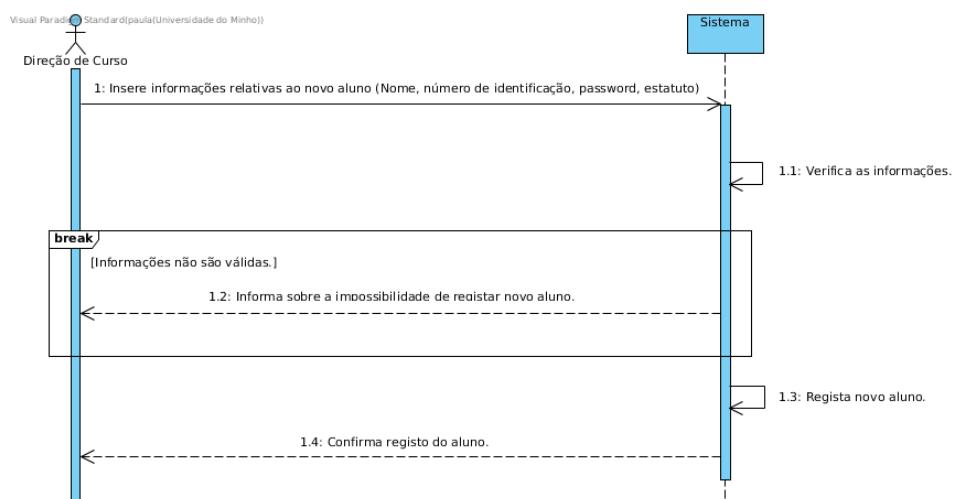


Figura 19: Diagrama de sequencia do Use Case ”Registrar Aluno”.

6.3 Registar UC

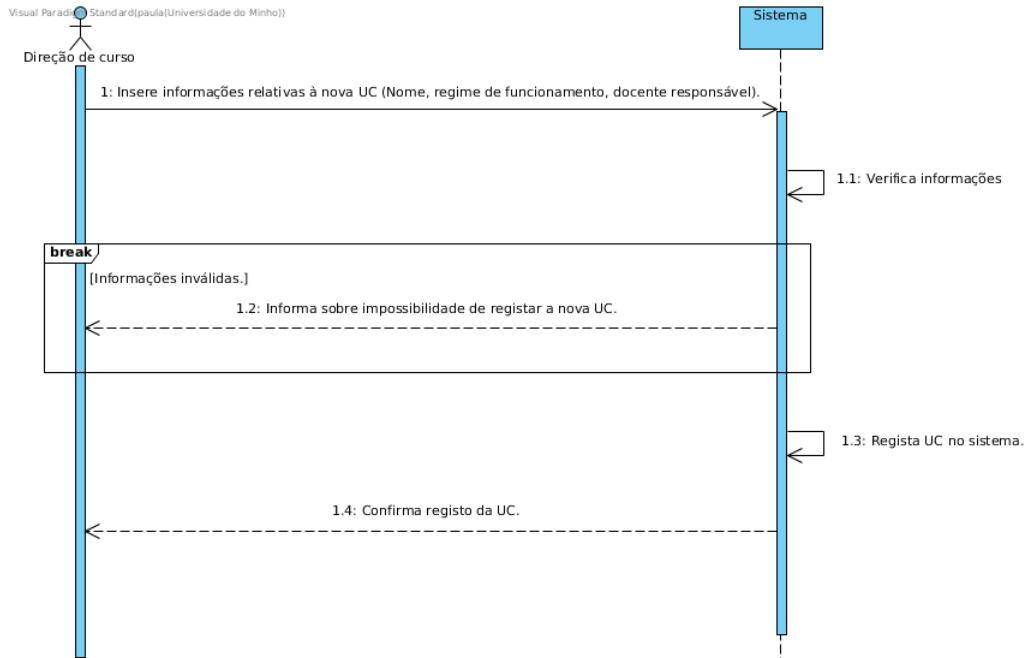


Figura 20: Diagrama de sequencia do Use Case ”Registrar UC”.

6.4 Atribuir turnos

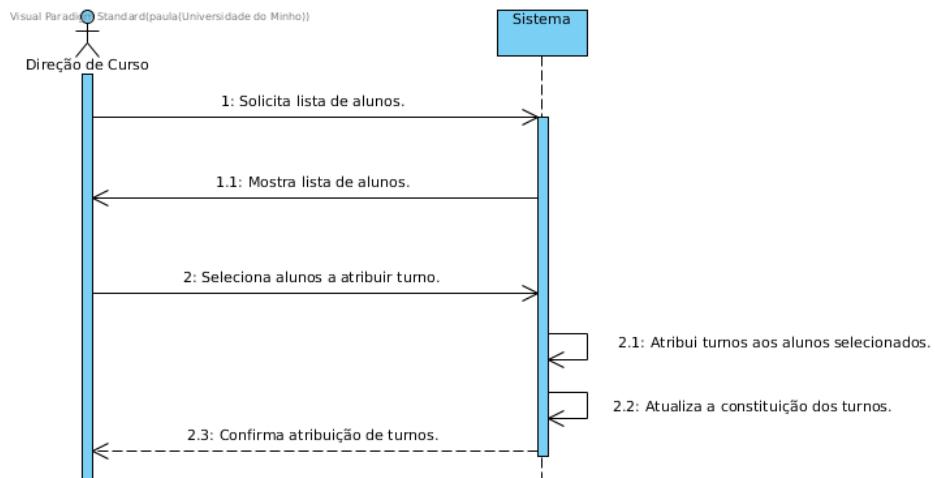


Figura 21: Diagrama de sequencia do Use Case ”Atribuir turnos”.

6.5 Mudar fase

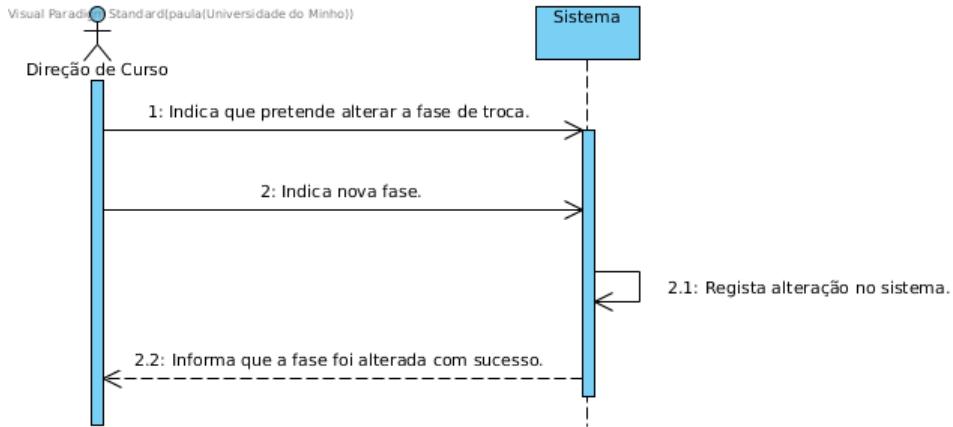


Figura 22: Diagrama de sequencia do Use Case ”Mudar fase”.

6.6 Definir capacidade de um turno

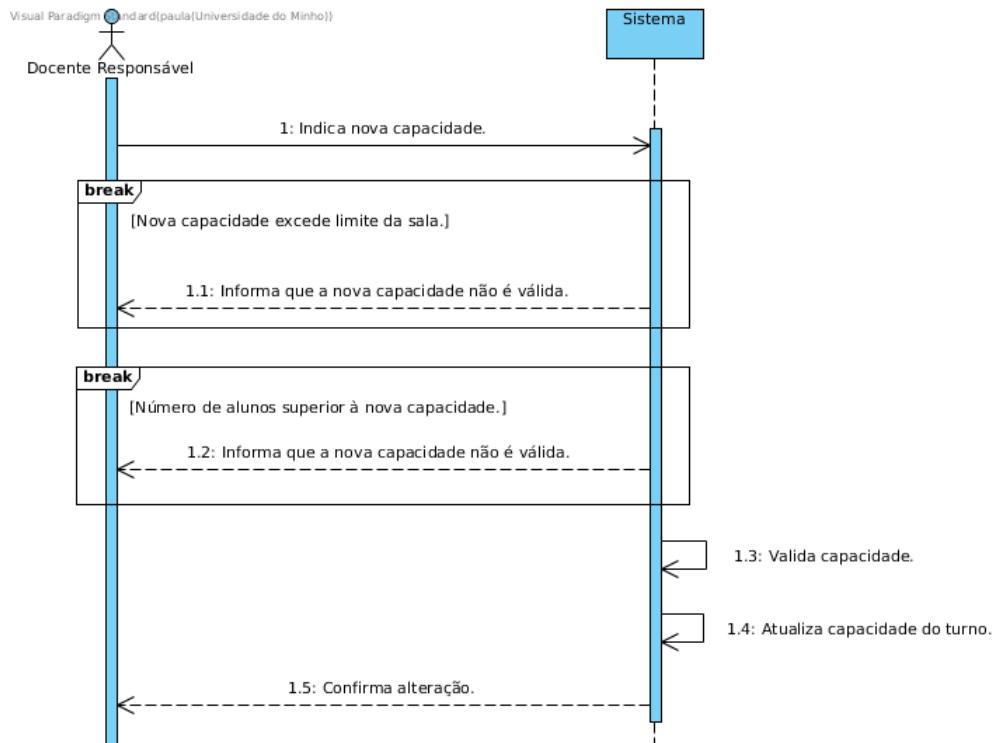


Figura 23: Diagrama de sequencia do Use Case ”Definir capacidade de um turno”.

6.7 Registar Faltas

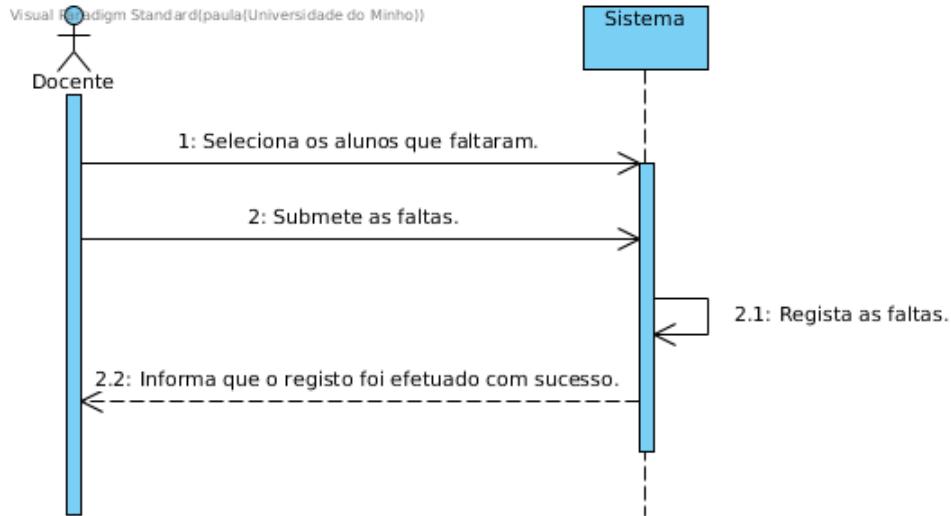


Figura 24: Diagrama de sequencia do Use Case ”Registrar Faltas”.

6.8 Remover Aluno de um Turno

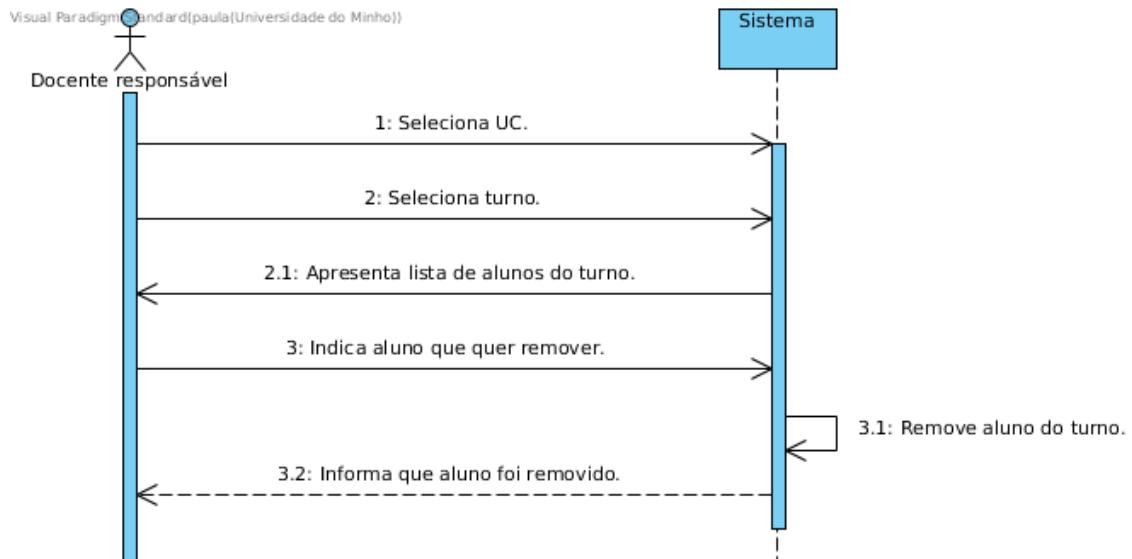


Figura 25: Diagrama de sequencia do Use Case ”Remover Aluno de um Turno”.

6.9 Adicionar Aluno a um Turno

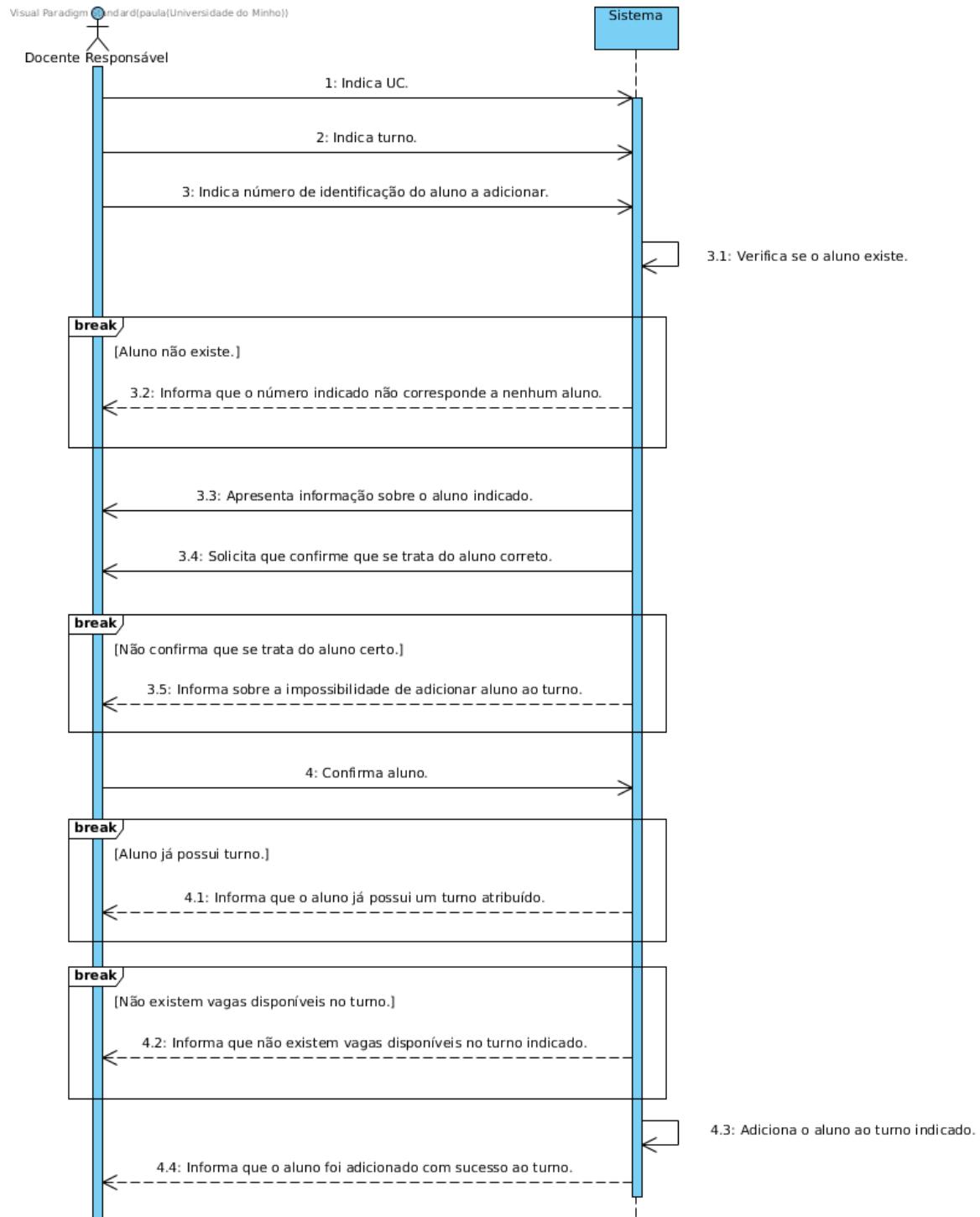


Figura 26: Diagrama de sequencia do Use Case ”Adicionar Aluno a um Turno”.

6.10 Ver turnos

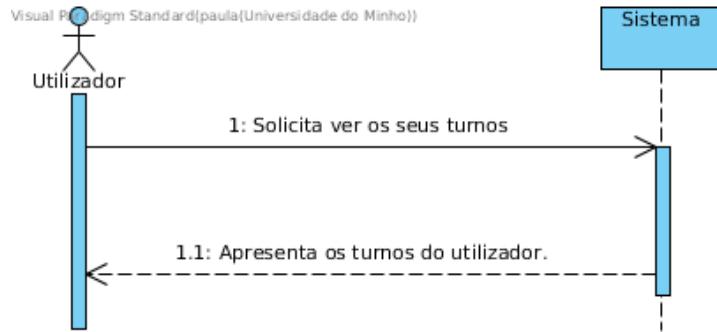


Figura 27: Diagrama de sequencia do Use Case "Ver turnos".

6.11 Aceder ao histórico de uma UC

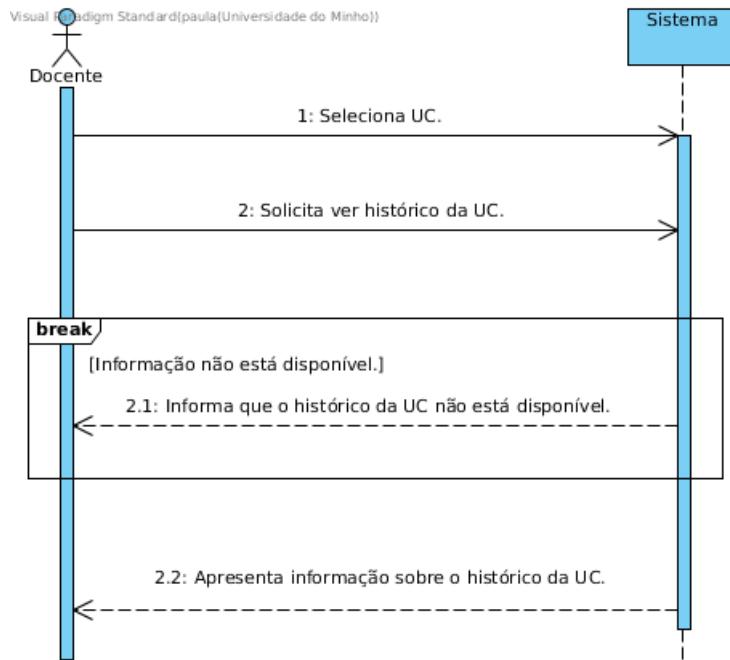


Figura 28: Diagrama de sequencia do Use Case "Aceder ao histórico de uma UC".

6.12 Enviar pedido de troca

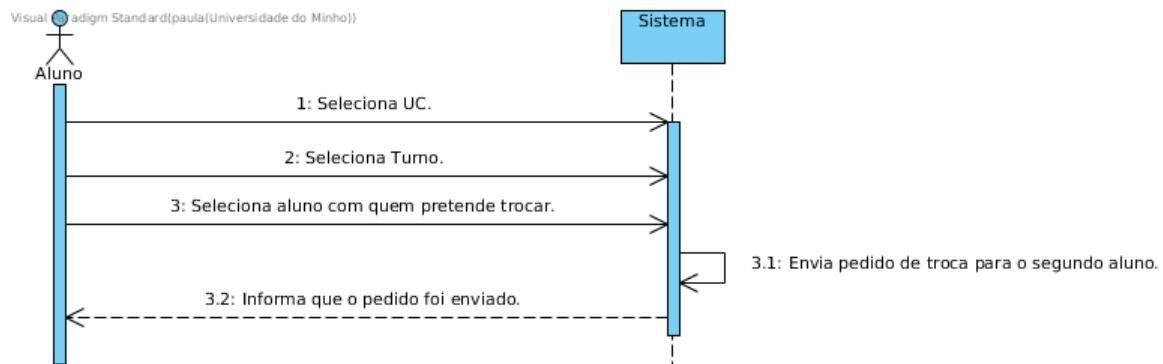


Figura 29: Diagrama de sequencia do Use Case ”Enviar pedido de troca”.

6.13 Aceitar pedido

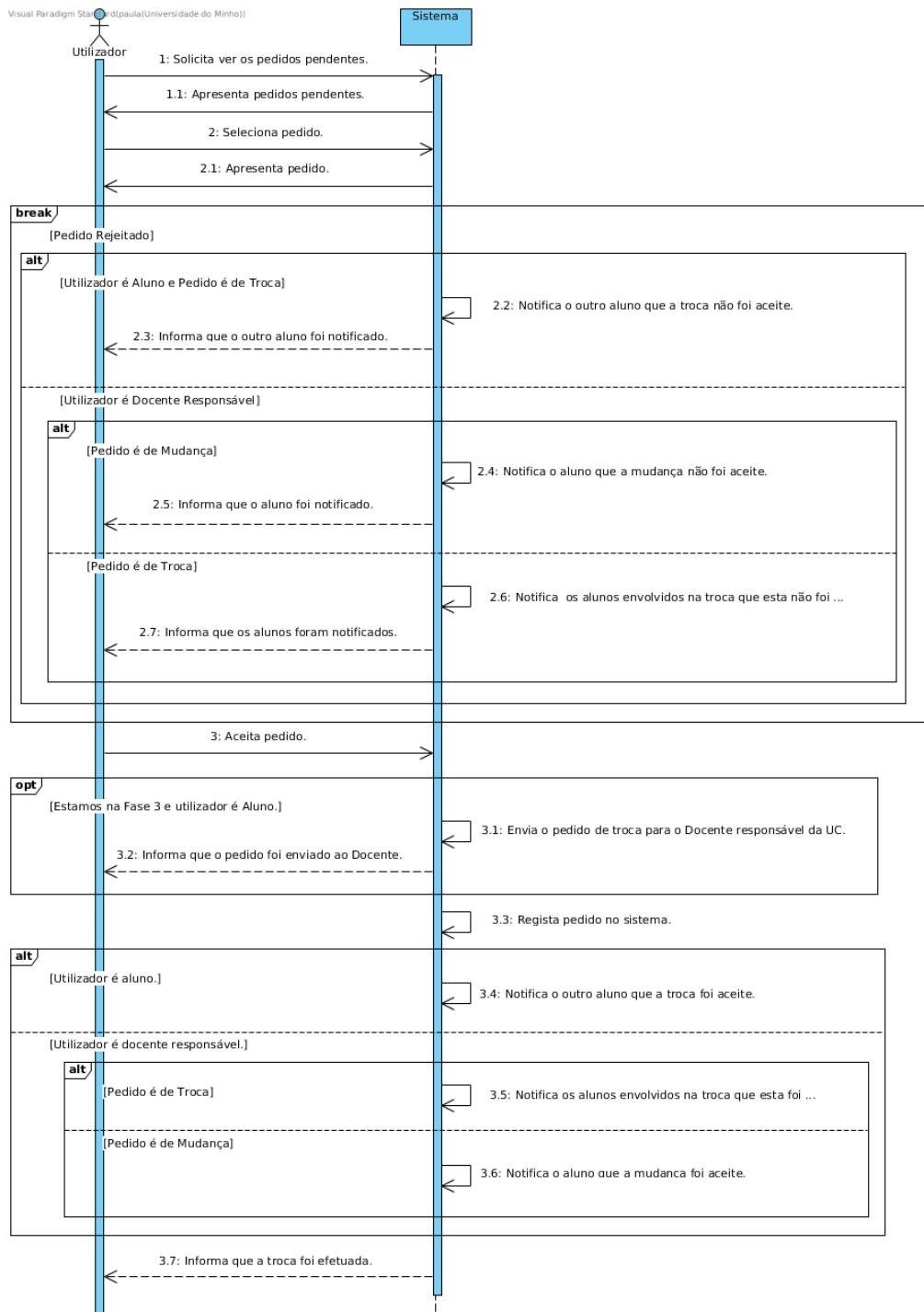


Figura 30: Diagrama de sequencia do Use Case ”Aceitar pedido”.

6.14 Mudar de turno

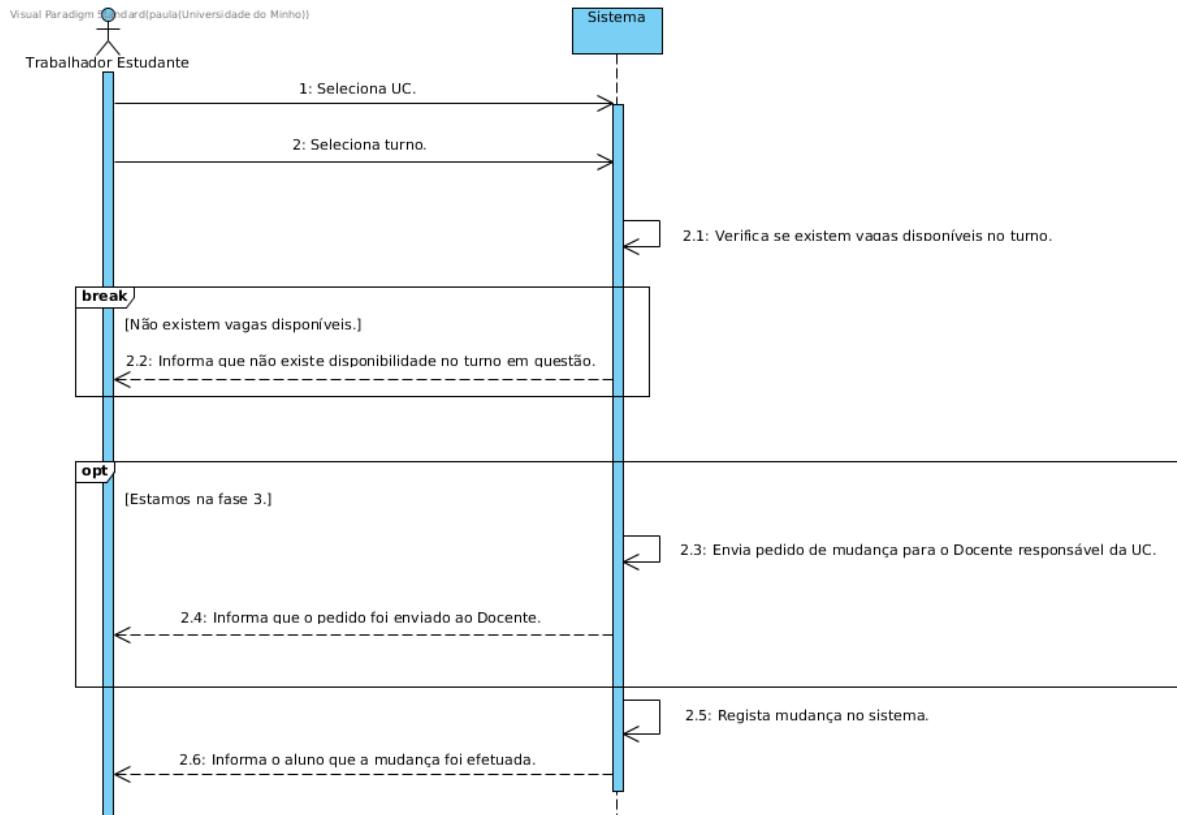


Figura 31: Diagrama de sequencia do Use Case "Mudar de turno".

6.15 Terminar Sessão

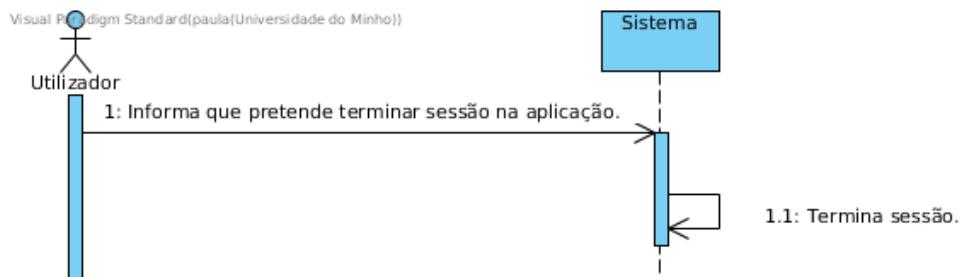


Figura 32: Diagrama de sequencia do Use Case "Terminar sessão".

7 Diagramas de Sequência de Subsistemas

Nesta secção, para cada Use Case definiu-se o respetivo diagrama de subsistemas, sendo que esses subsistemas foram identificados através do modelo de domínio agrupando entidades cujo contexto é semelhante.

Estes diagramas são essenciais no desenvolvimento do projeto para adicionar mais comportamento a cada use case, aumentando assim o seu grau de especificação, o que se reflete numa maior percepção das interações do sistema.

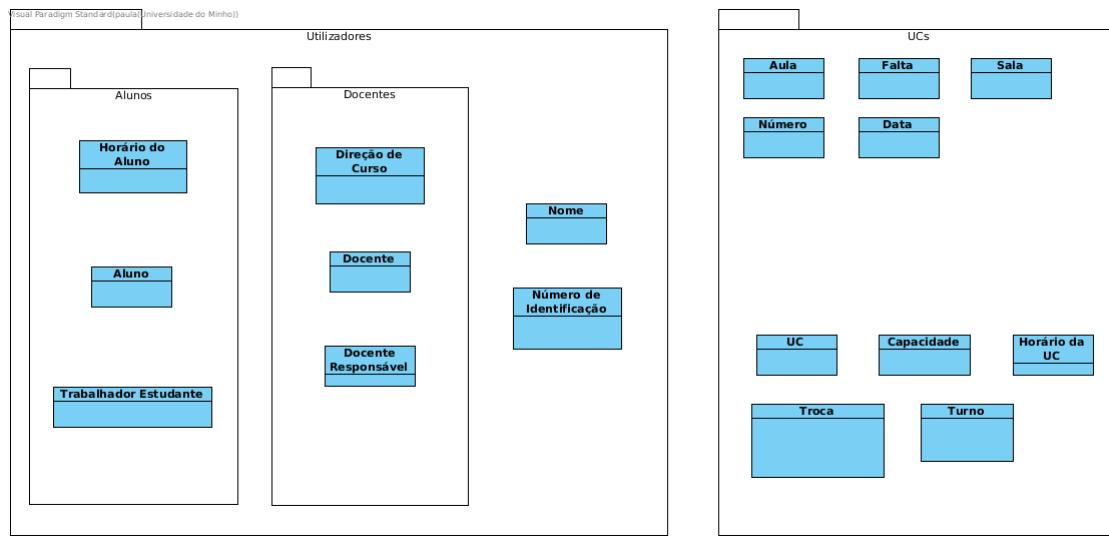


Figura 33: Modelo de subsistemas utilizado.

7.1 Iniciar Sessão

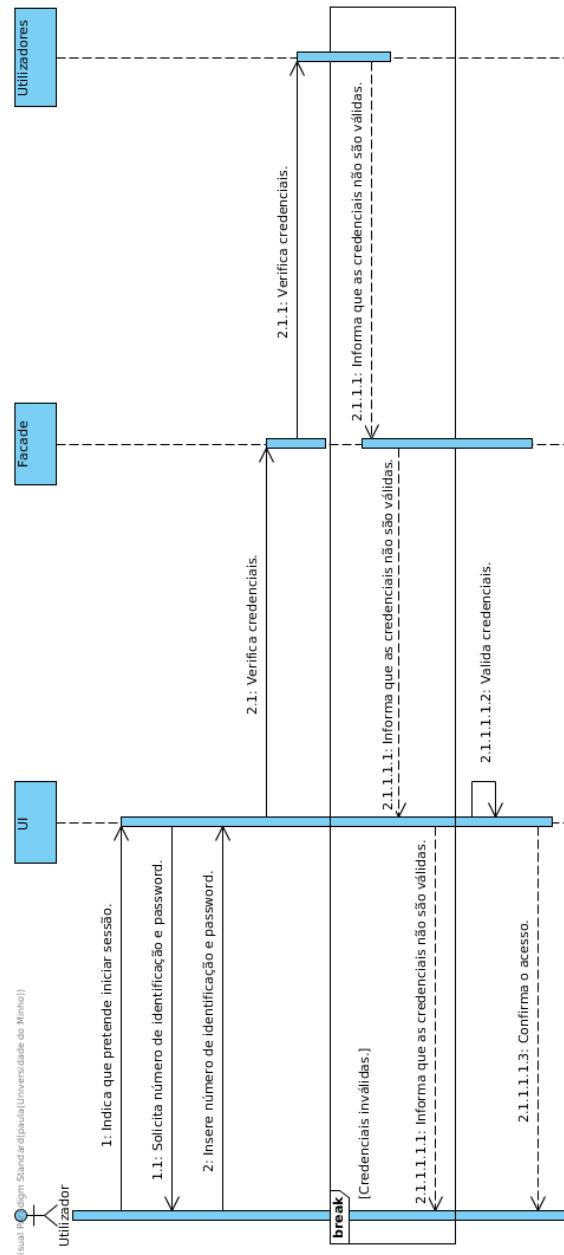


Figura 34: Diagrama de sequencia de subsistemas do Use Case "Iniciar sessão".

7.2 Registar Aluno

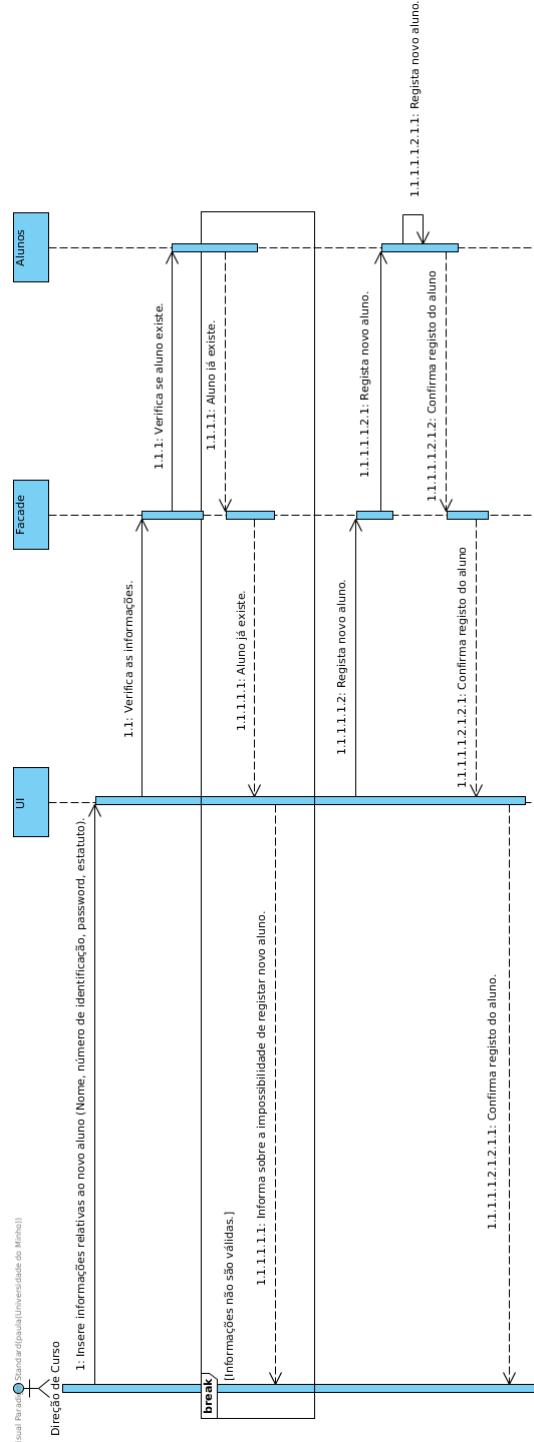


Figura 35: Diagrama de sequencia de subsistemas do Use Case "Registrar Aluno".

7.3 Registar UC

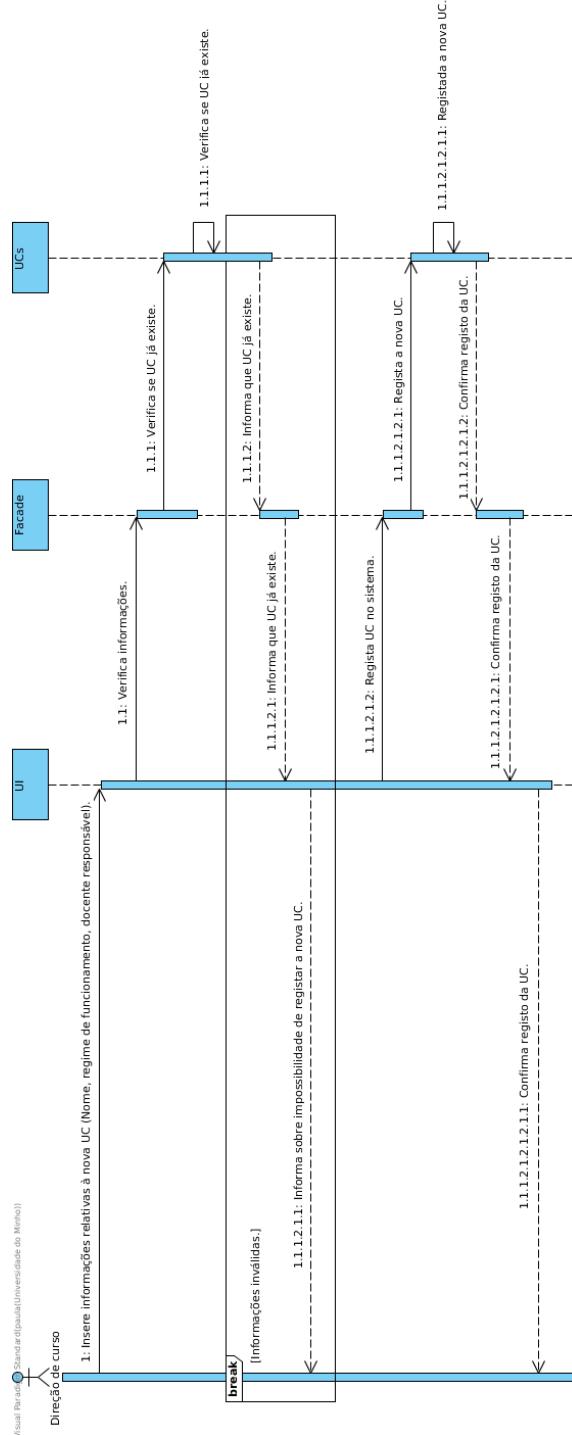


Figura 36: Diagrama de sequencia de subsistemas do Use Case "Registrar UC".

7.4 Atribuir turnos

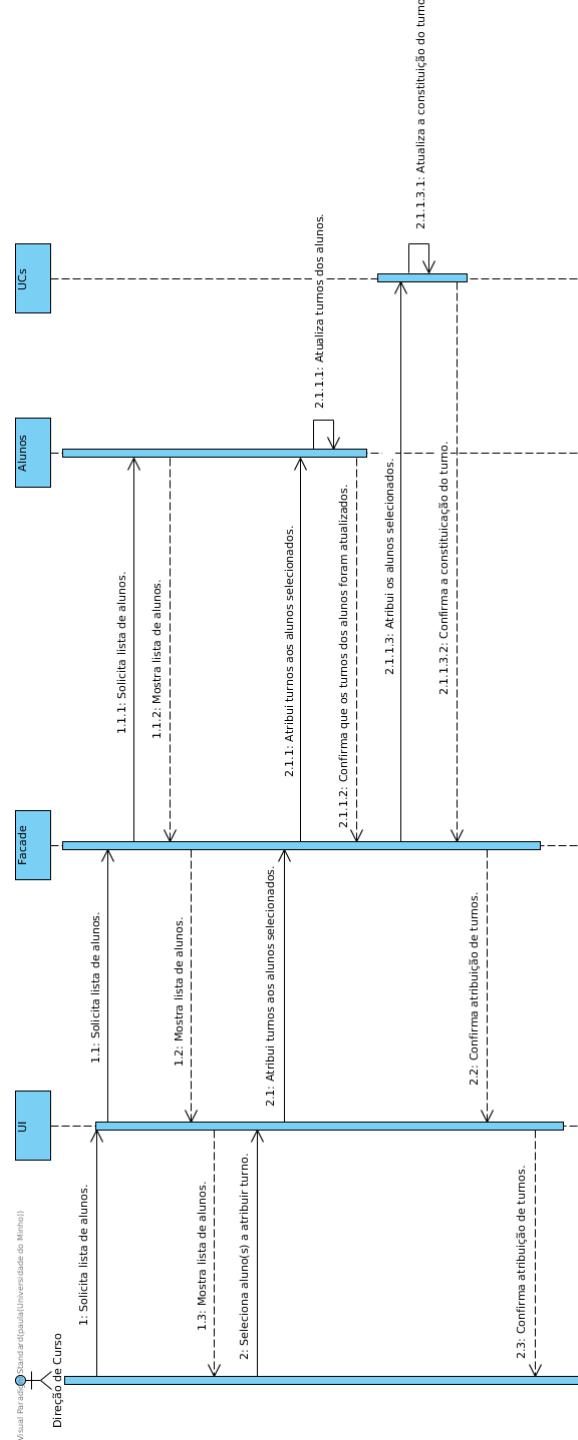


Figura 37: Diagrama de sequencia de subsistemas do Use Case "Atribuir turnos".

7.5 Mudar fase

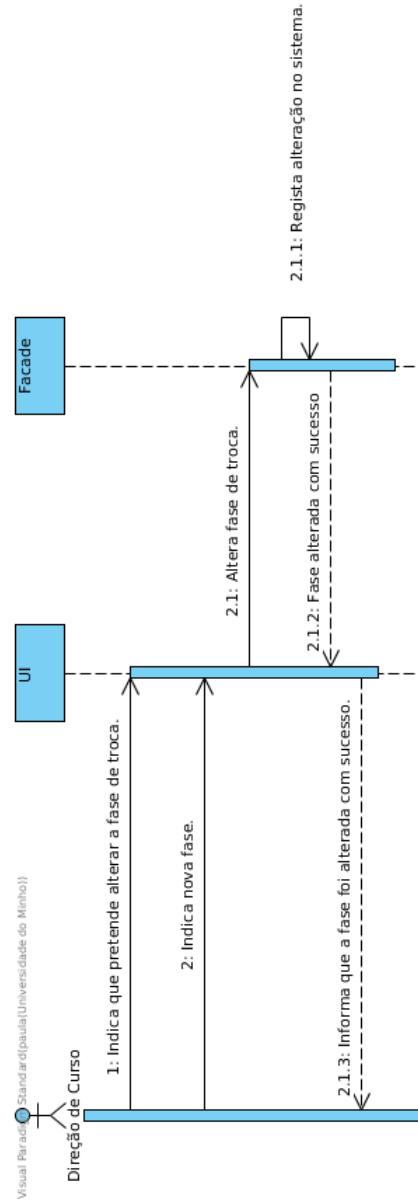


Figura 38: Diagrama de sequencia de subsistemas do Use Case "Mudar fase".

7.6 Definir capacidade de um turno

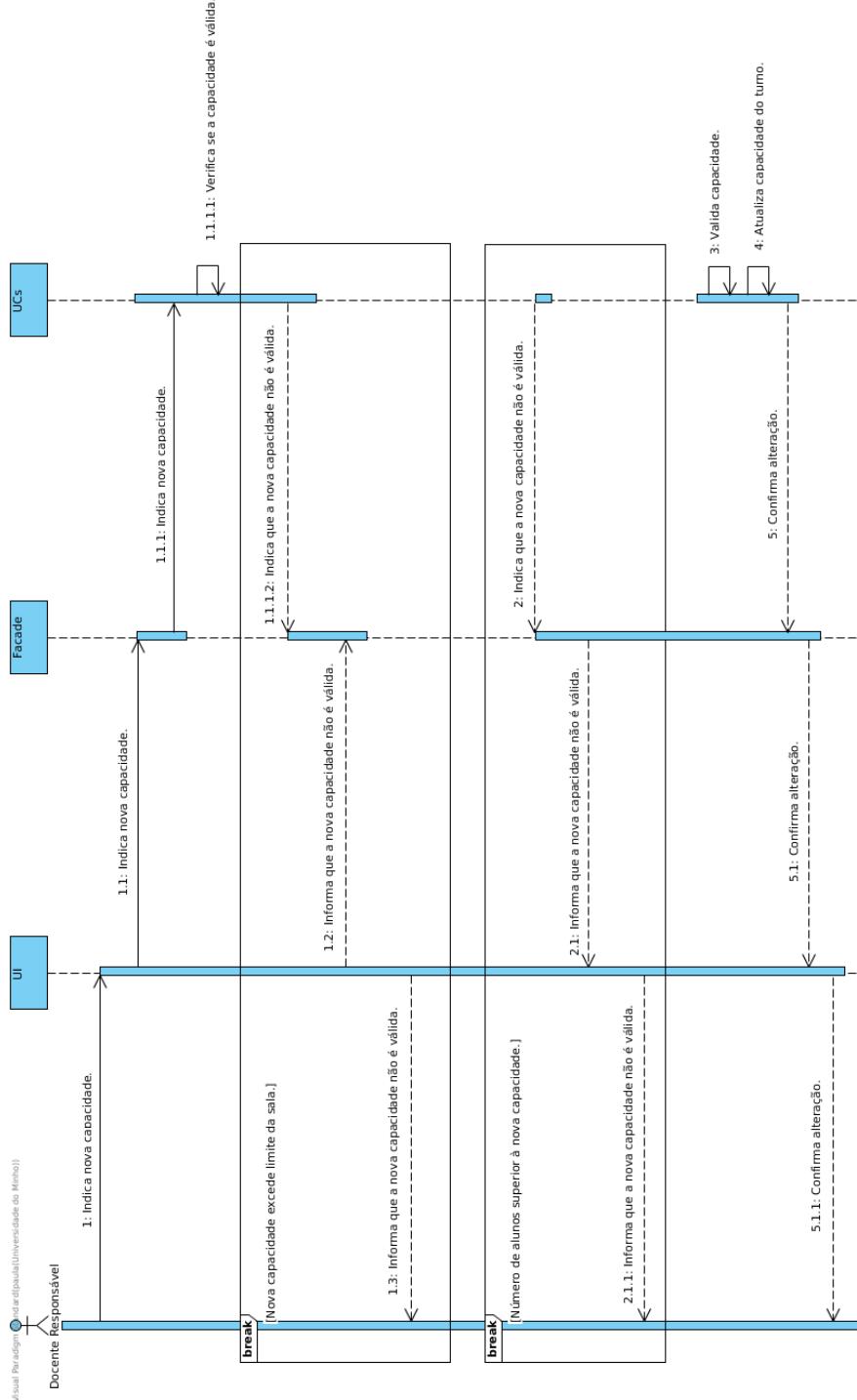


Figura 39: Diagrama de sequencia de subsistemas do Use Case "Definir capacidade de um turno".

7.7 Registrar Faltas

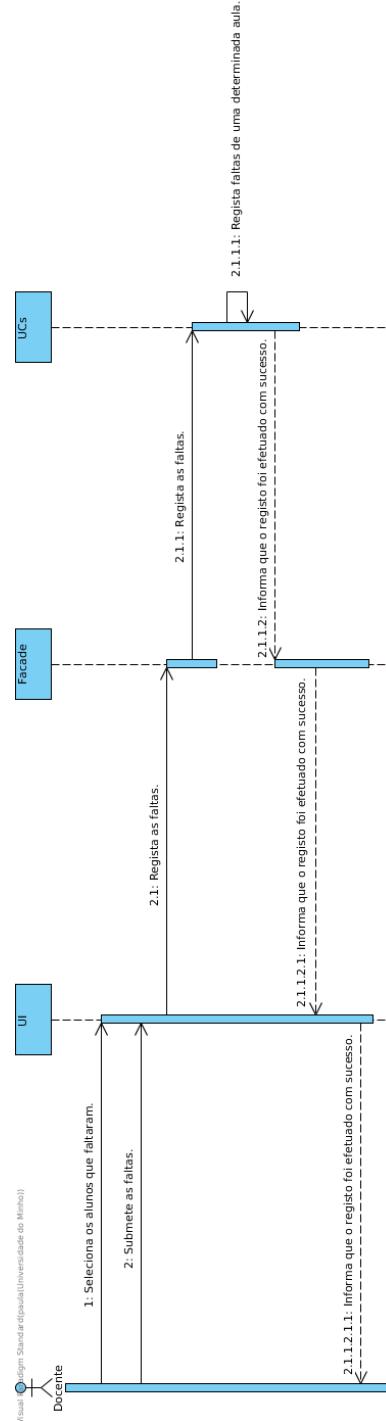


Figura 40: Diagrama de sequencia de subsistemas do Use Case "Registrar faltas".

7.8 Remover Aluno de um Turno

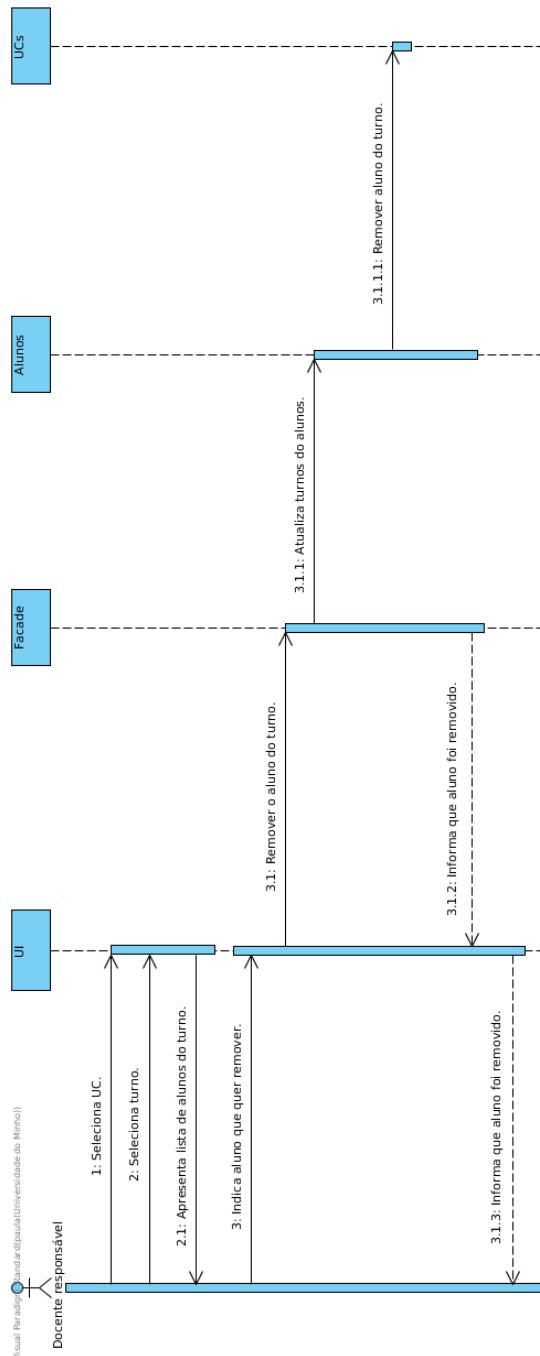


Figura 41: Diagrama de sequencia de subsistemas do Use Case "Remover Aluno de um Turno".

7.9 Adicionar Aluno a um Turno

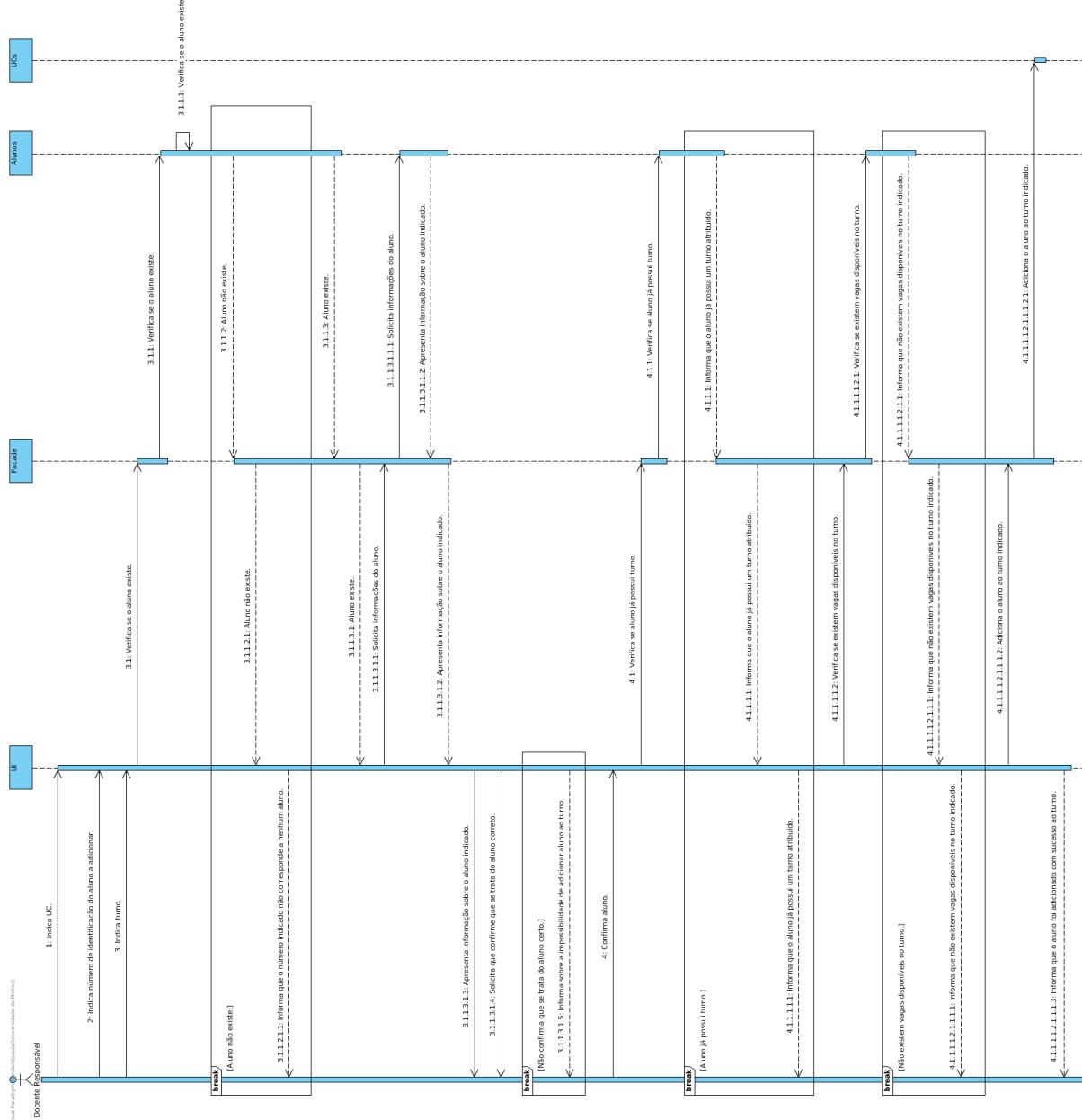


Figura 42: Diagrama de sequencia de subsistemas do Use Case "Adicionar Aluno a um Turno".

7.10 Ver turnos

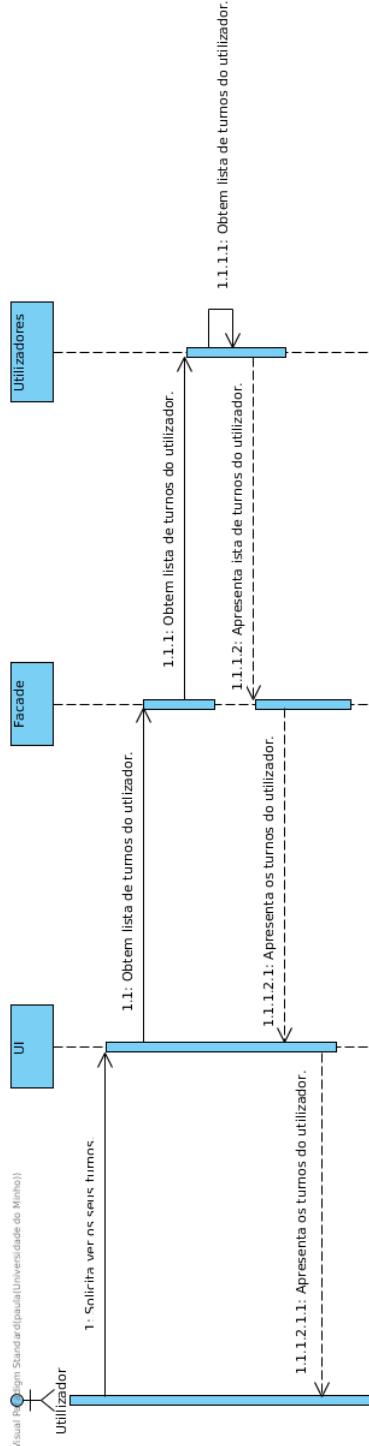


Figura 43: Diagrama de sequencia de subsistemas do Use Case "Ver turnos".

7.11 Aceder ao histórico da UC

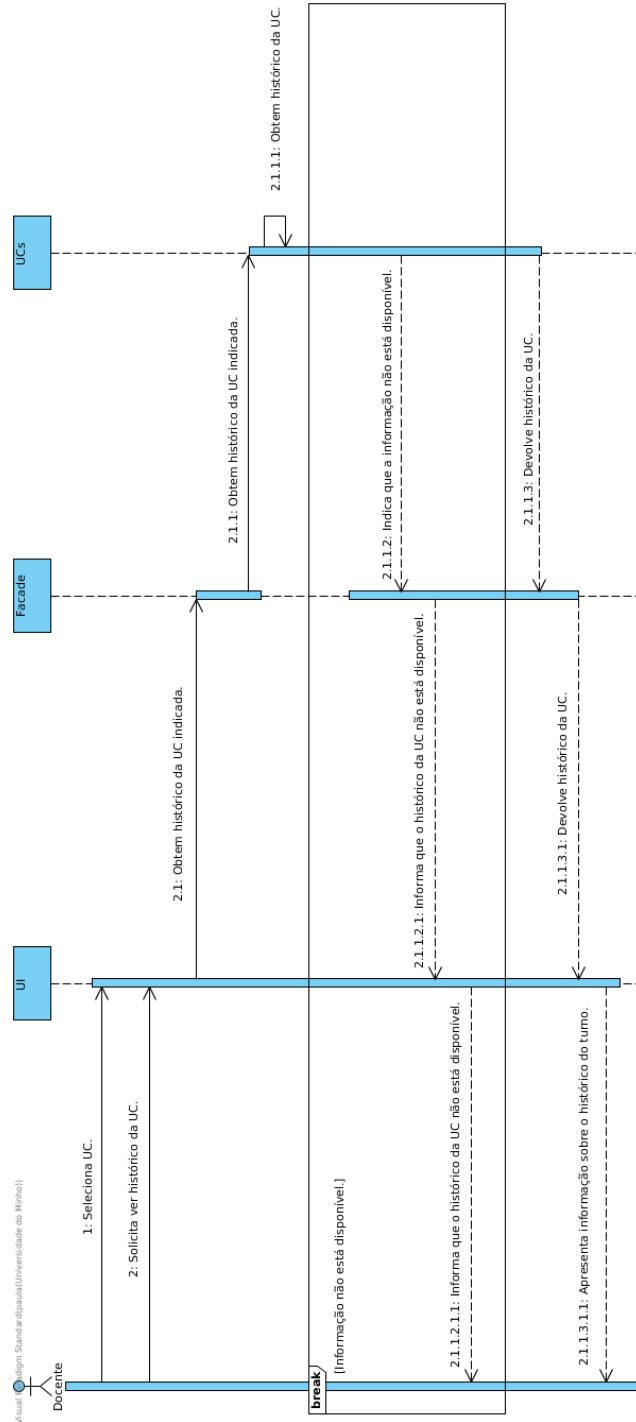


Figura 44: Diagrama de sequencia de subsistemas do Use Case ”Aceder ao histórico da UC”.

7.12 Enviar pedido de troca

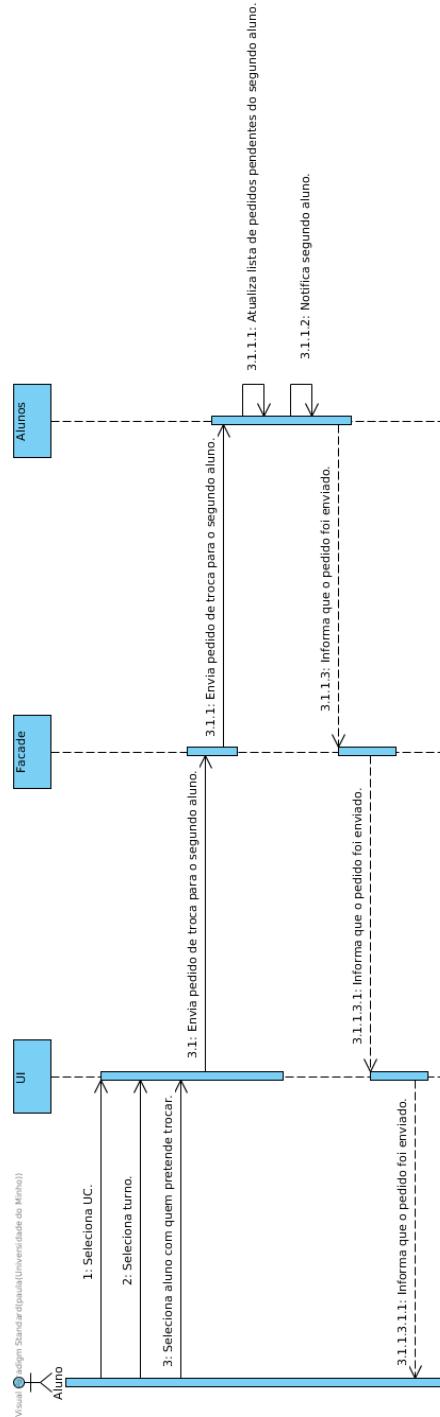


Figura 45: Diagrama de sequencia de subsistemas do Use Case ”Enviar pedido de troca” .

7.13 Aceitar pedido

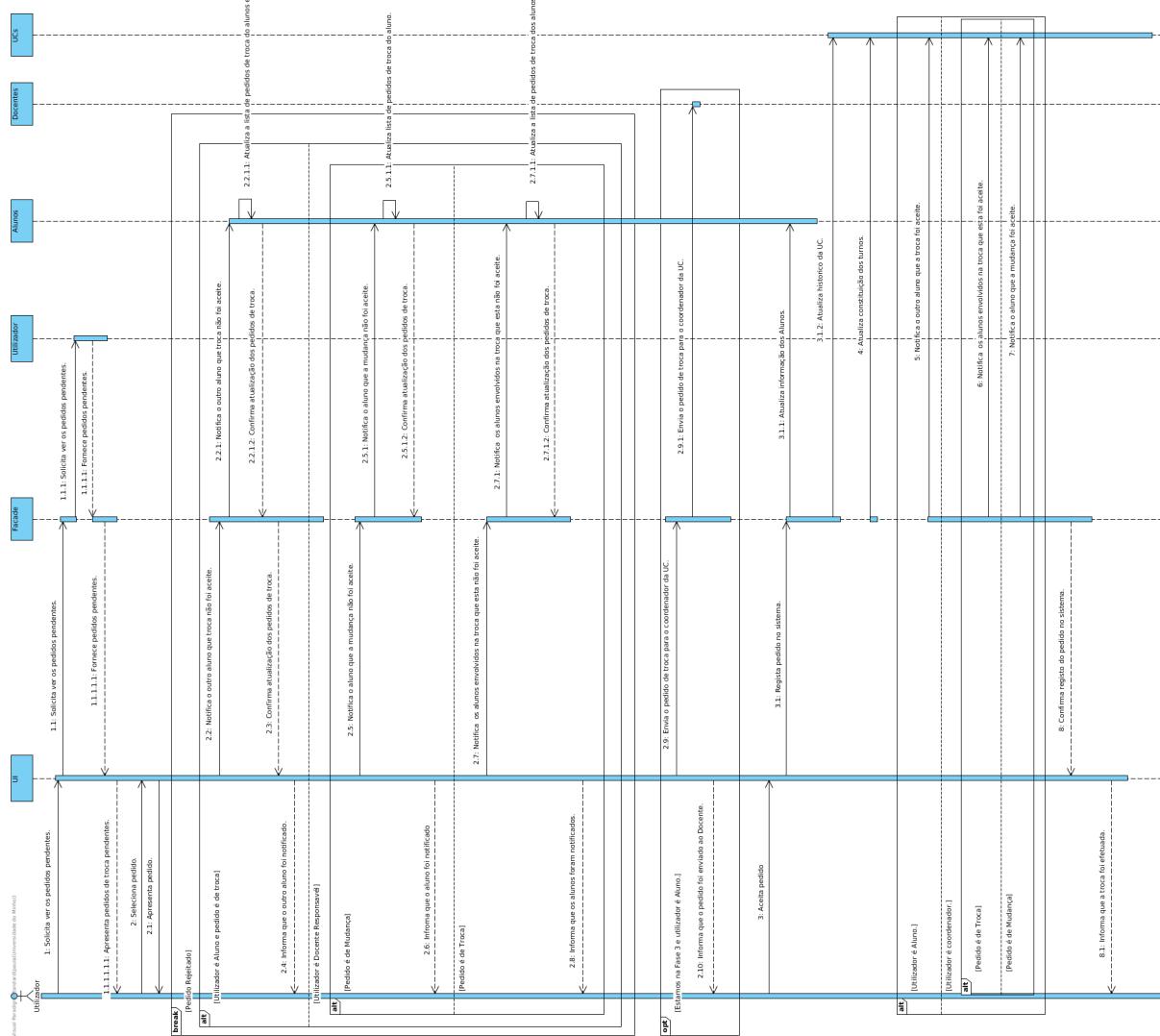


Figura 46: Diagrama de sequencia de subsistemas do Use Case "Aceitar pedido".

7.14 Mudar de turno

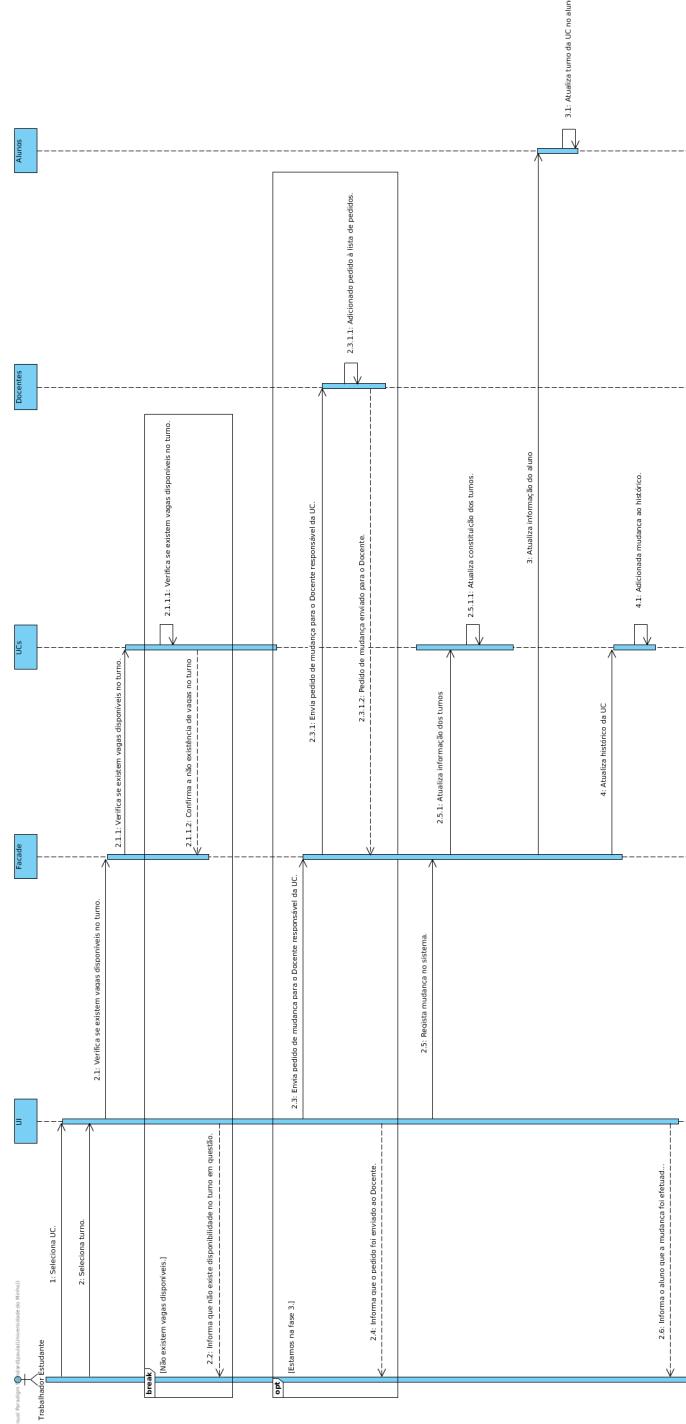


Figura 47: Diagrama de sequencia de subsistemas do Use Case "Mudar de turno".

7.15 Terminar Sessão



Figura 48: Diagrama de sequencia de subsistemas do Use Case "Terminar sessão".

8 Diagrama de Classes

Partindo da análise do modelo de domínio e dos use cases anteriormente especificados é-nos possível identificar as principais entidades que participam no funcionamento da nossa aplicação. Estas são agora potenciais classes que serão implementadas na mesma.

Pensando já na codificação da aplicação, consideramos necessária a conceção dos diagramas de classe de modo a ter uma visão geral acerca das classes a implementar e dos seus respectivos atributos e métodos.

Inicialmente foi desenvolvido o diagrama apresentado na figura 49 onde os dados se encontram guardados em Collections e Maps. De seguida, implementou-se persistência, sendo que os dados passaram agora a estar armazenados numa Base de Dados. O diagrama referente a esta nova implementação encontra-se na figura 50.

8.1 Diagrama de Classes

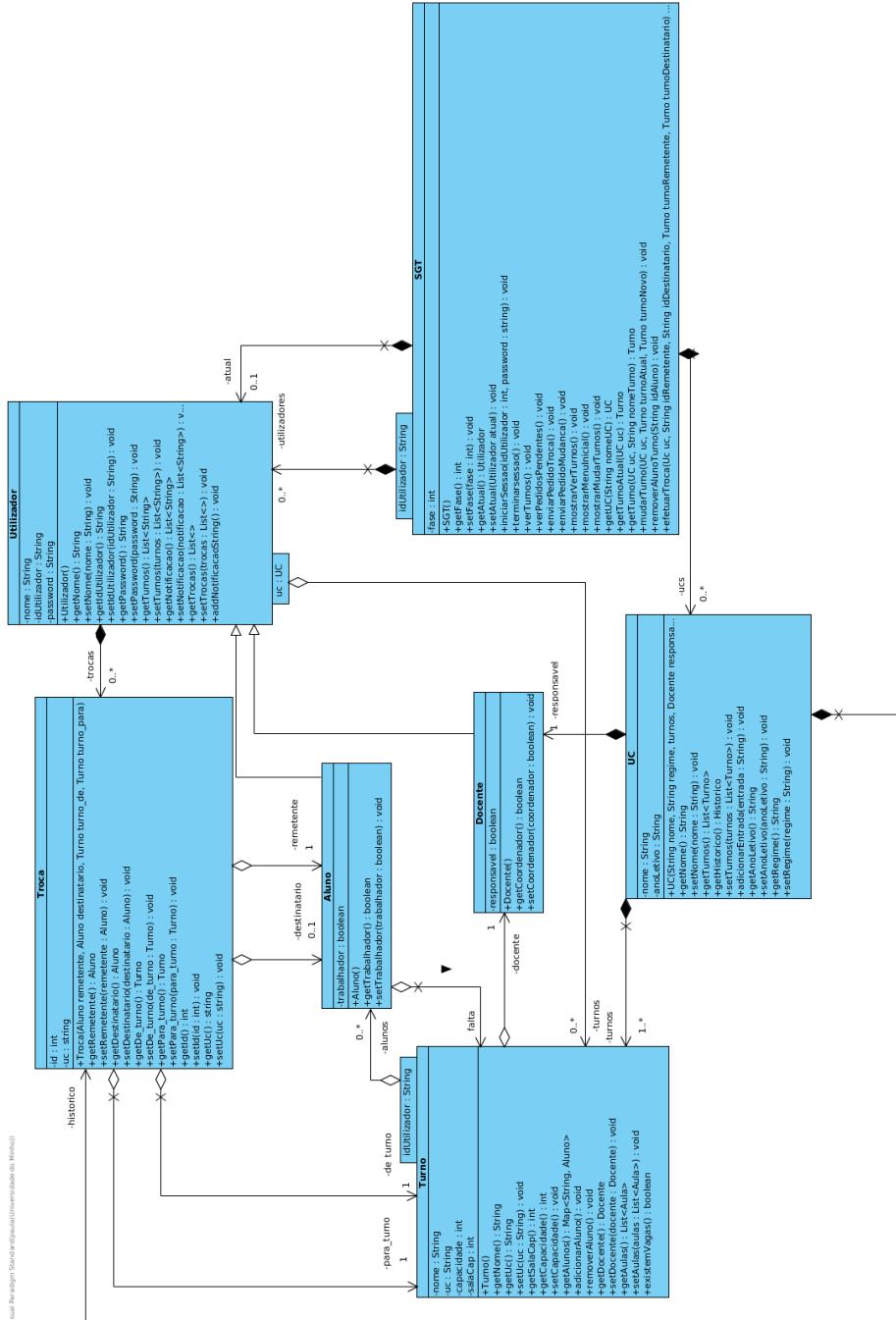


Figura 49: Diagrama de classes.

8.2 Diagram de ORM

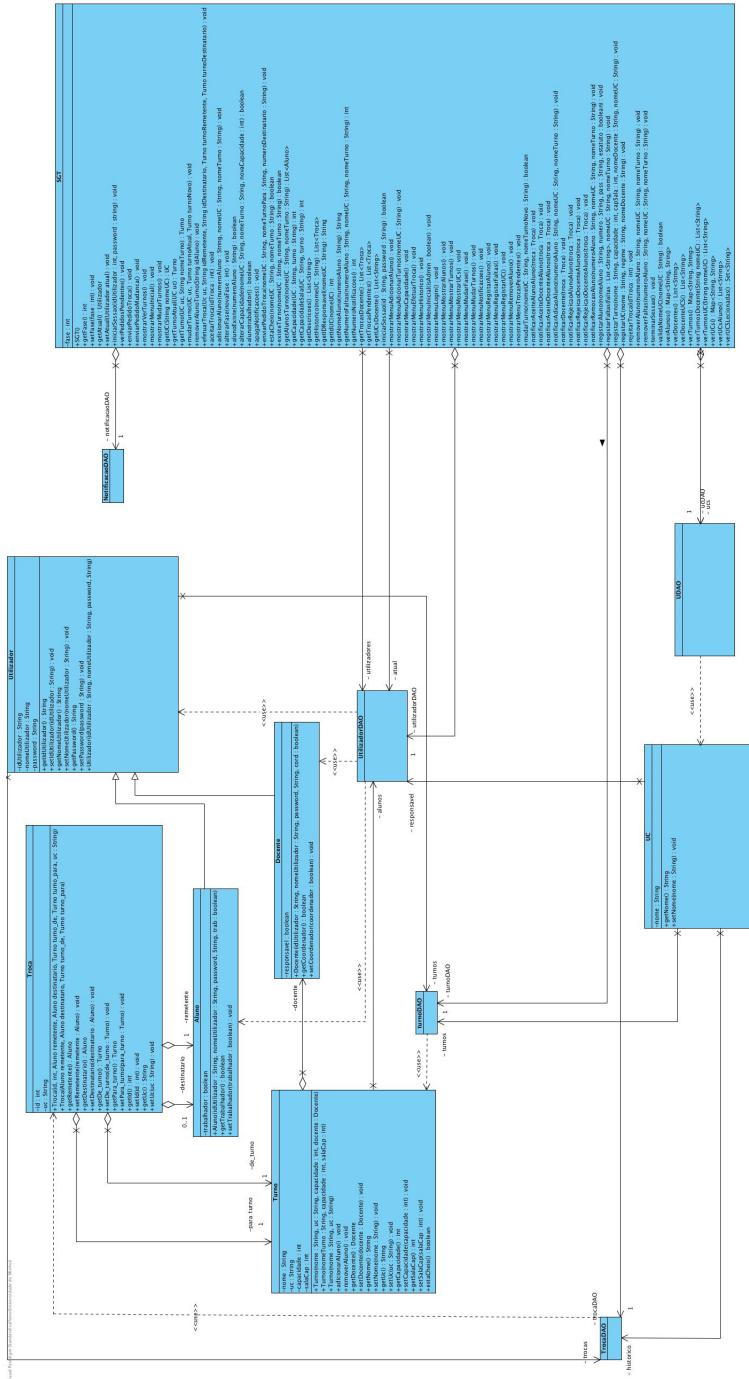


Figura 50: Diagrama de ORM.

9 Diagramas de Packages

Após definidas as classes do sistema, procedemos a criação do diagrama de *packages* através do qual conseguimos mostrar o agrupamento das classes do sistema.

Assim, com a identificação das dependências entre os vários packages conseguimos descrever informação importante para a evolução do sistema.

De salientar, que este agrupamento assentou no modelo das três camadas, sendo por isso criados três packages:

- **Presentation** - Agrupa as classes responsáveis pela apresentação do conteúdo da aplicação ao utilizador e a consequente comunicação entre este e o sistema.
- **Business** - Agrupa as classes responsáveis pelo tratamento dos dados presentes na aplicação.
- **Data** - Agrupa as classes DAO, cuja função é estabelecer a conexão entre a *Business Layer* e a Base de Dados

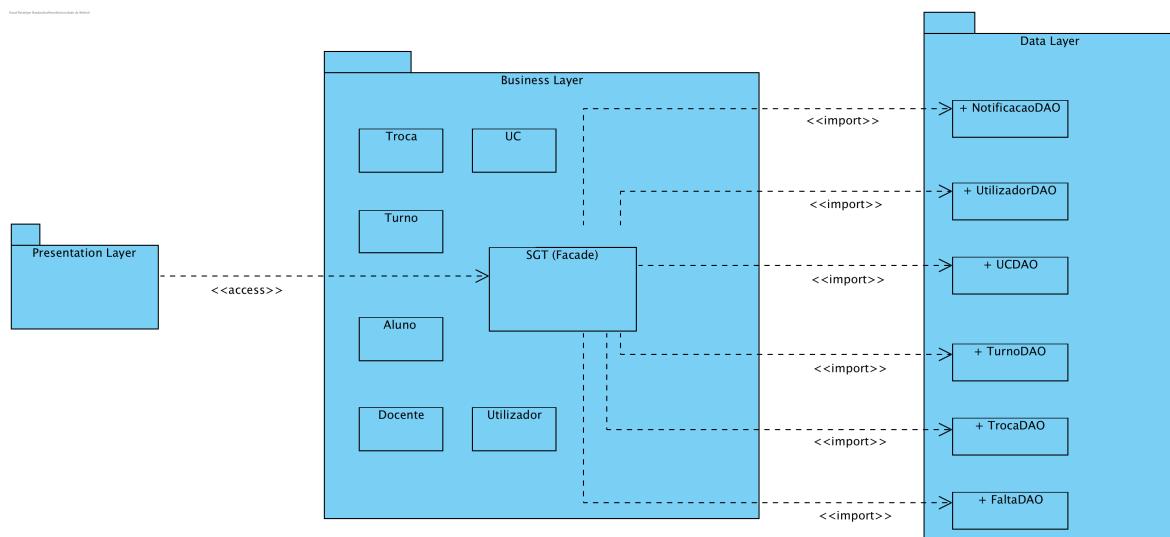


Figura 51: Diagrama de Packages.

10 Diagramas de Sequência de implementação

Nesta secção, desdobramos os use cases em diagramas de sequência de implementação, mostrando detalhadamente o código exigido para realizar as operações.

10.1 Iniciar Sessão

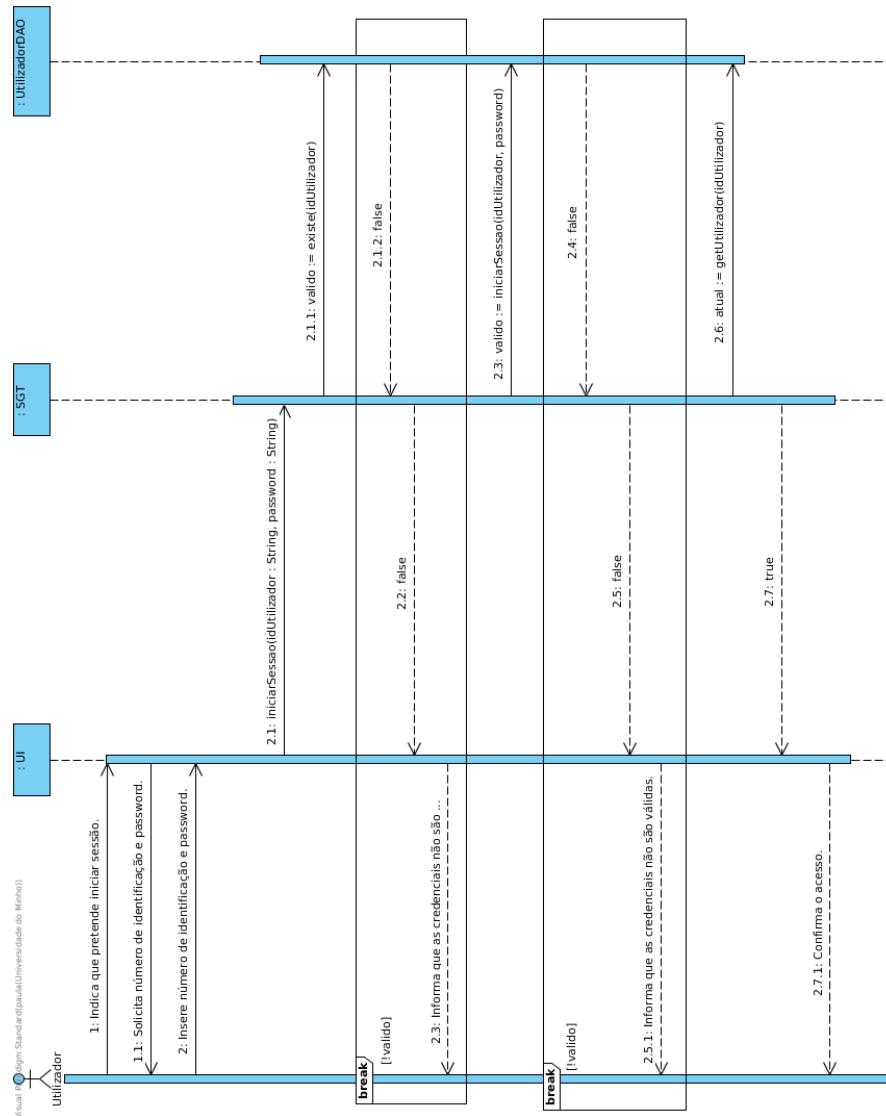


Figura 52: Diagrama de sequencia de implementação do Use Case ”Iniciar sessão”.

10.2 Registar Aluno

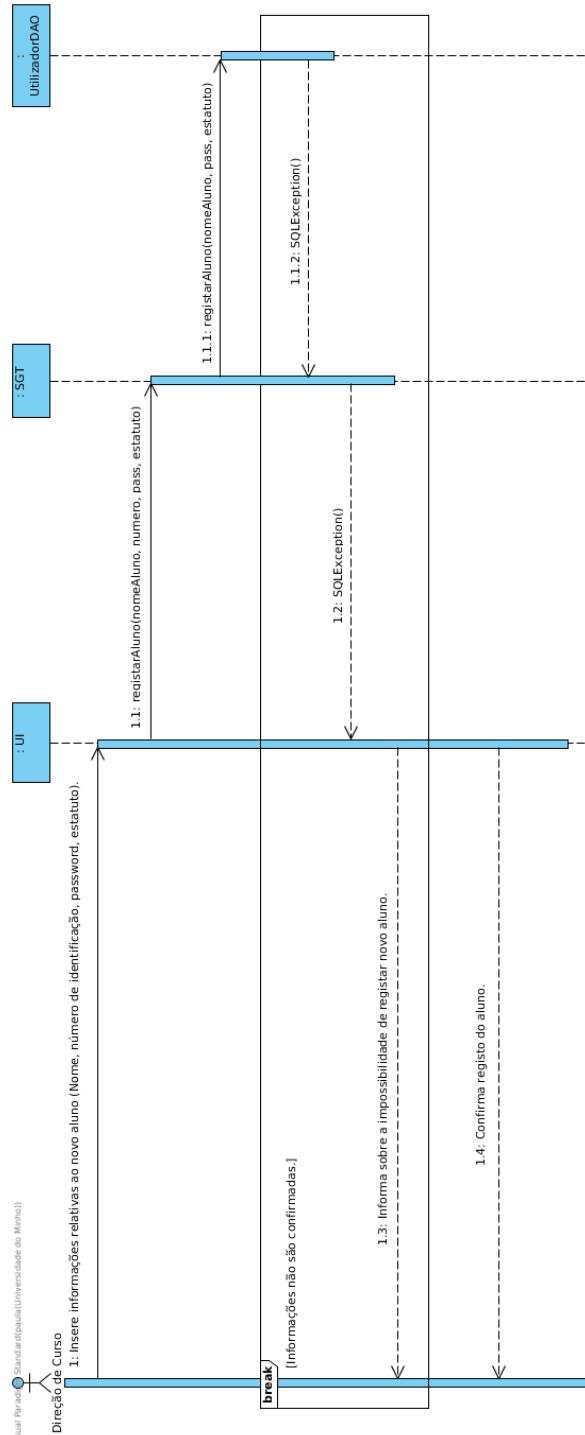


Figura 53: Diagrama de sequencia de implementação do Use Case ”Registrar Aluno”.

10.3 Registar UC

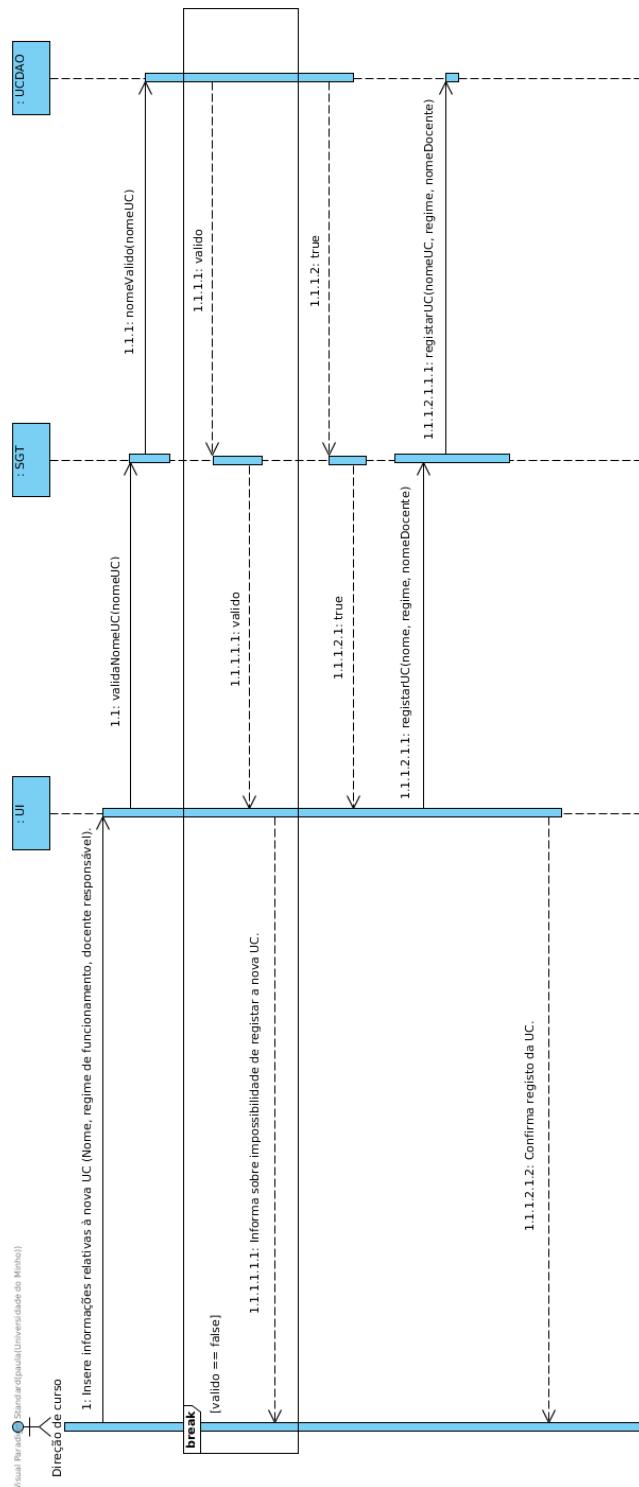


Figura 54: Diagrama de sequencia de implementação do Use Case ”Registrar UC”.

10.4 Mudar fase

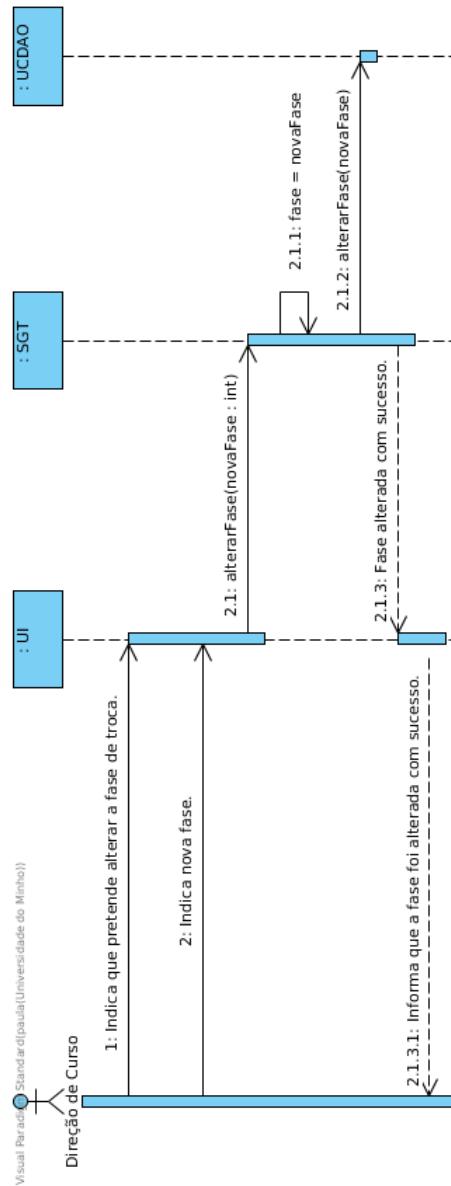


Figura 55: Diagrama de sequencia de implementação do Use Case ”Mudar fase”.

10.5 Definir capacidade de um turno

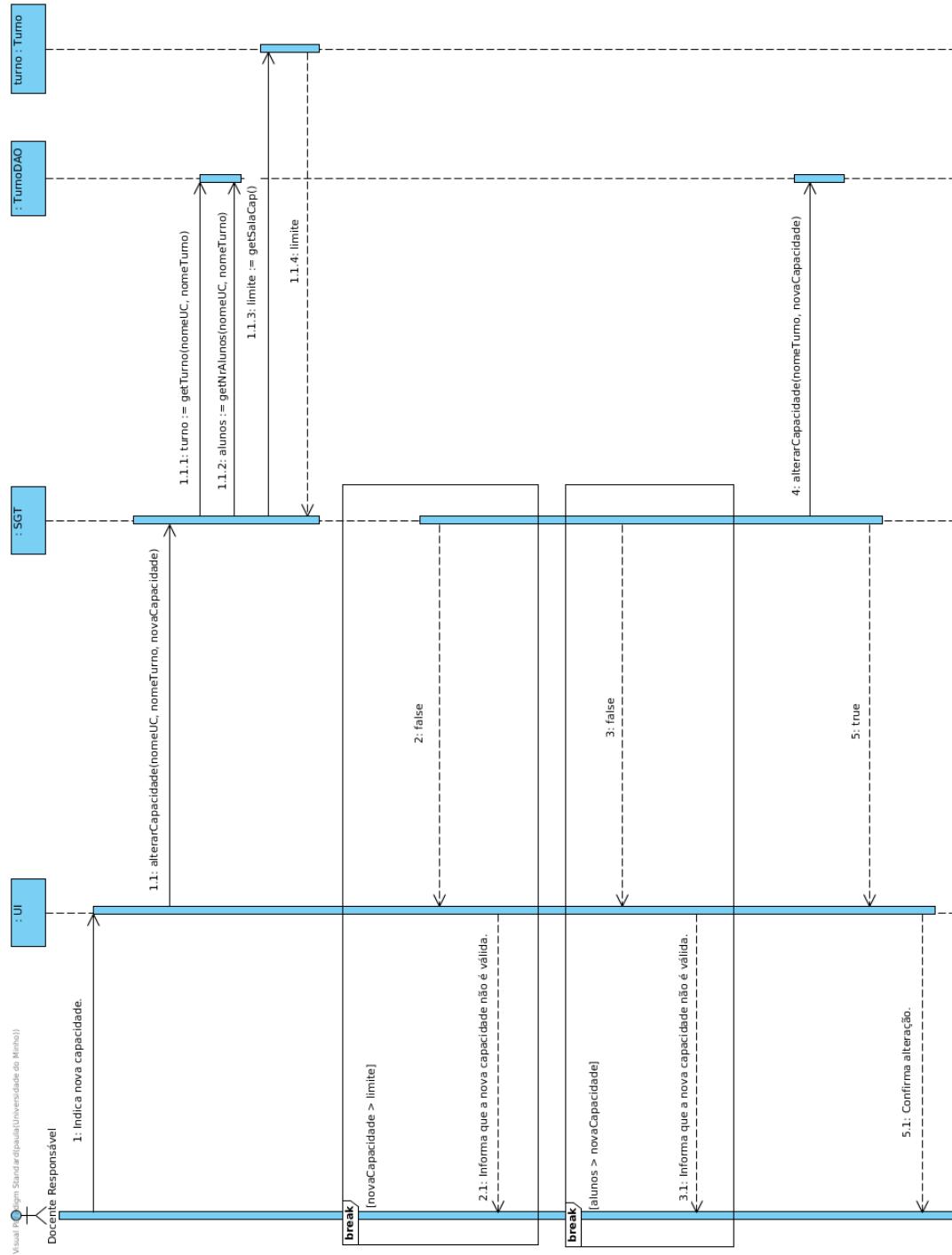


Figura 56: Diagrama de sequencia de implementação do Use Case ”Definir capacidade de um turno”

10.6 Registar Faltas

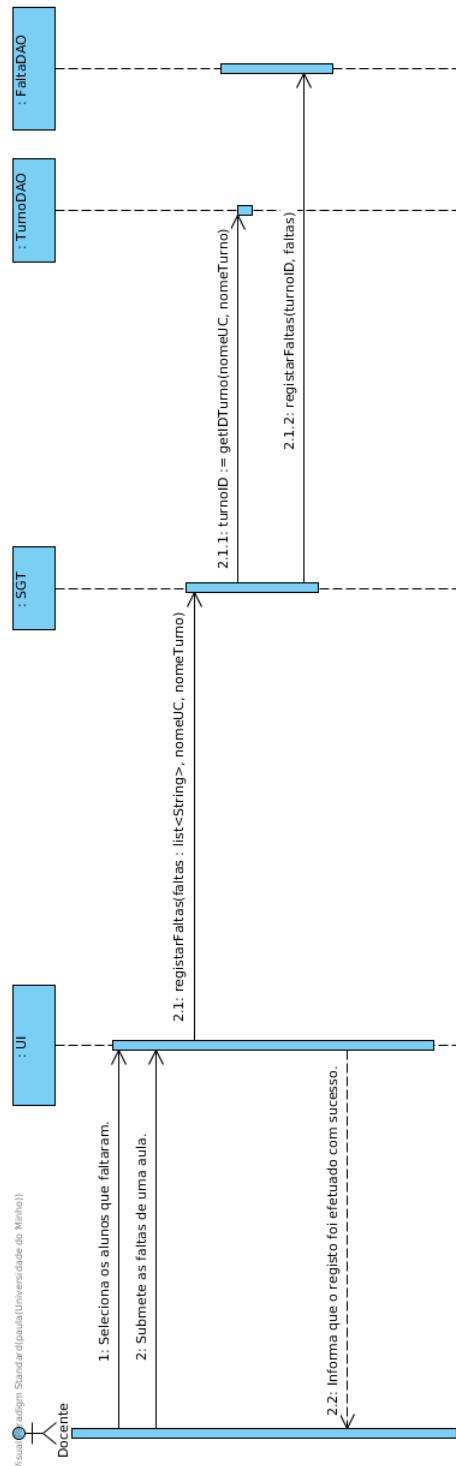


Figura 57: Diagrama de sequencia de implementação do Use Case ”Registrar faltas”.

10.7 Remover Aluno de um Turno

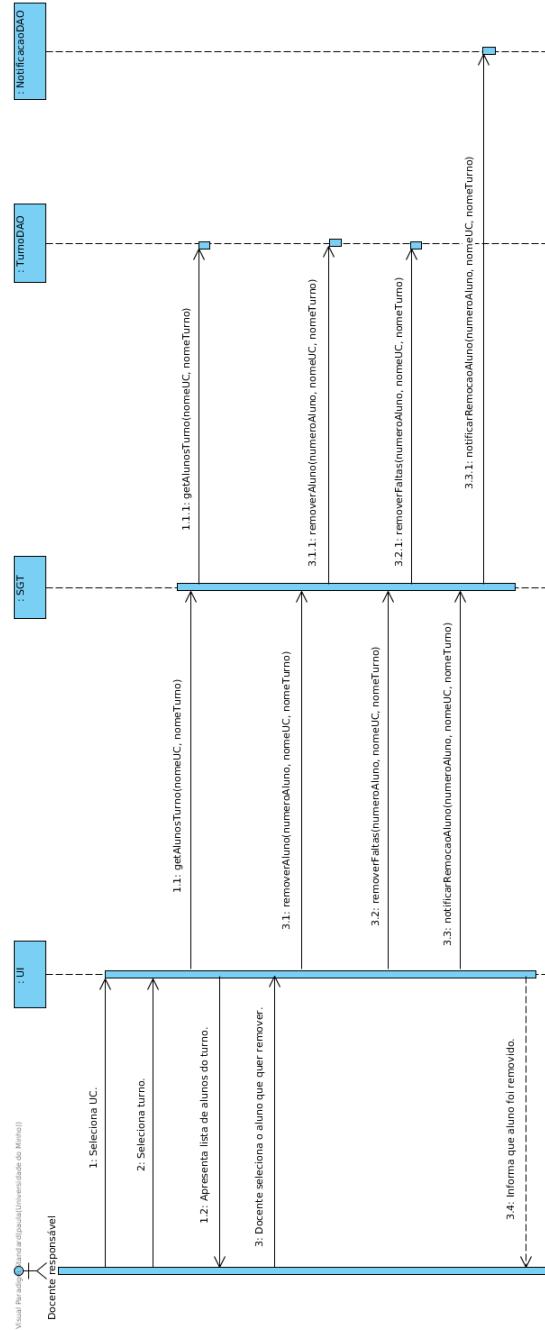


Figura 58: Diagrama de sequencia de implementação do Use Case "Remover Aluno de um Turno".

10.8 Adicionar Aluno a um Turno

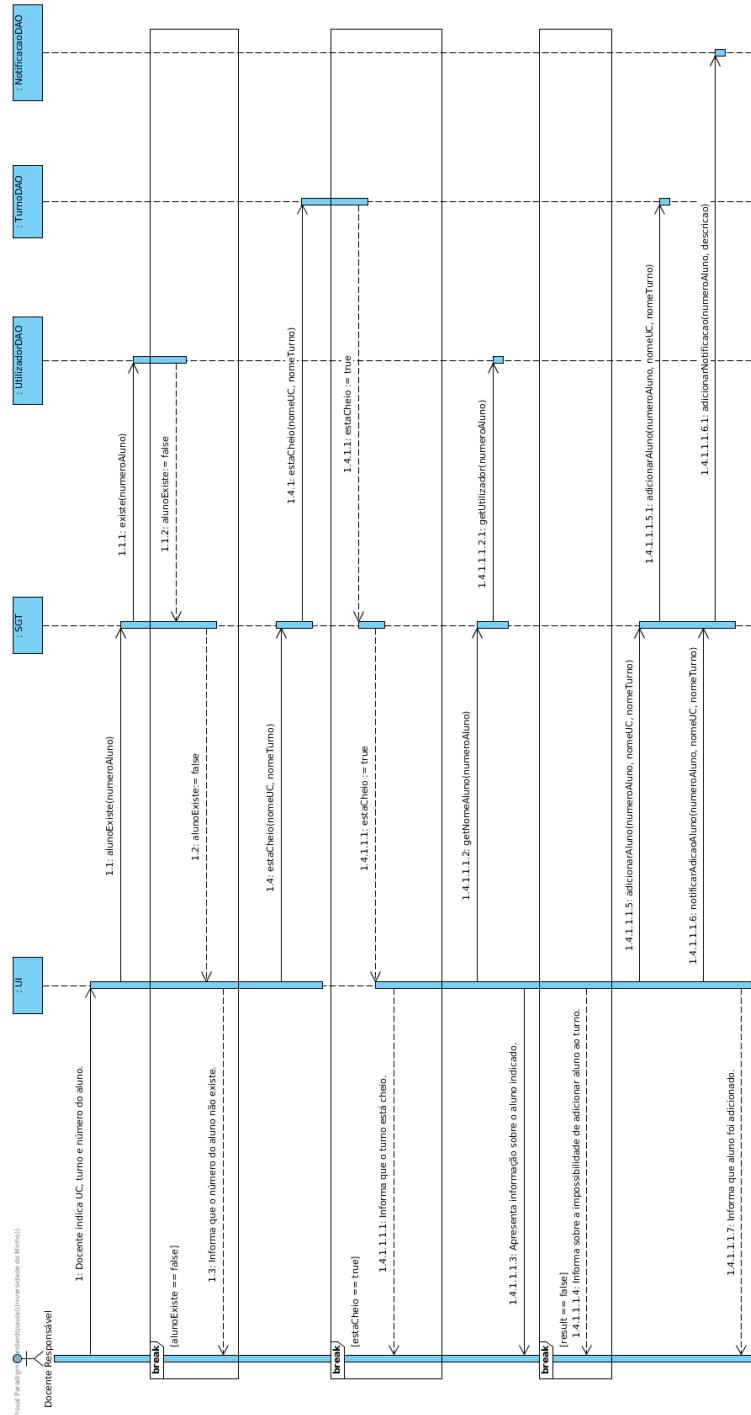


Figura 59: Diagrama de sequencia de implementação do Use Case ”Adicionar Aluno a um Turno”.

10.9 Ver turnos

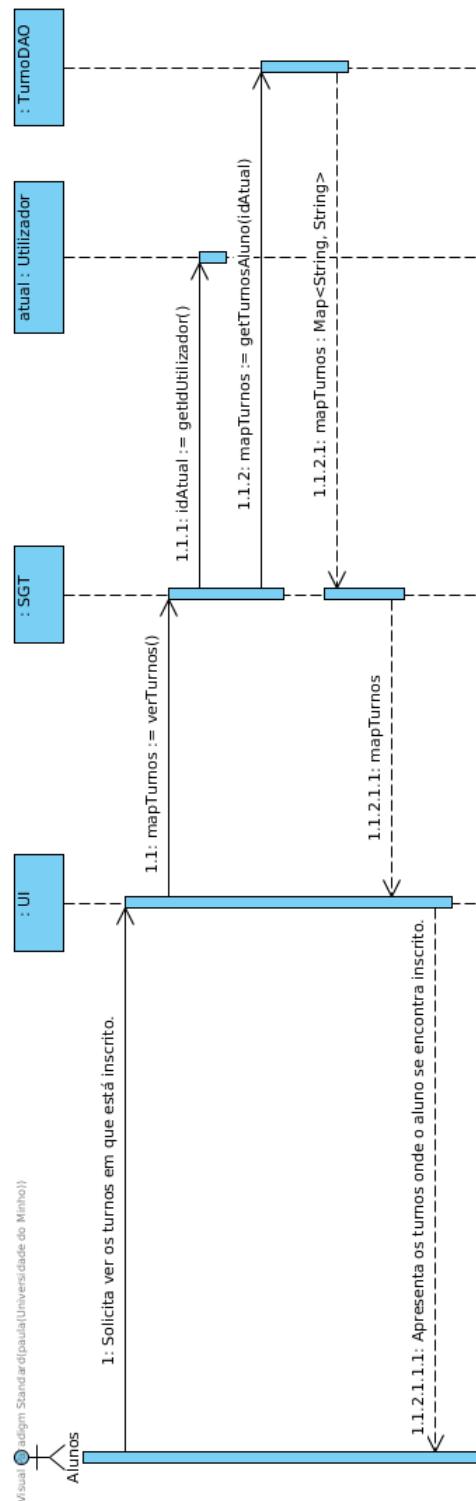


Figura 60: Diagrama de sequencia de implementação do Use Case "Ver turnos".

10.10 Aceder ao histórico da UC

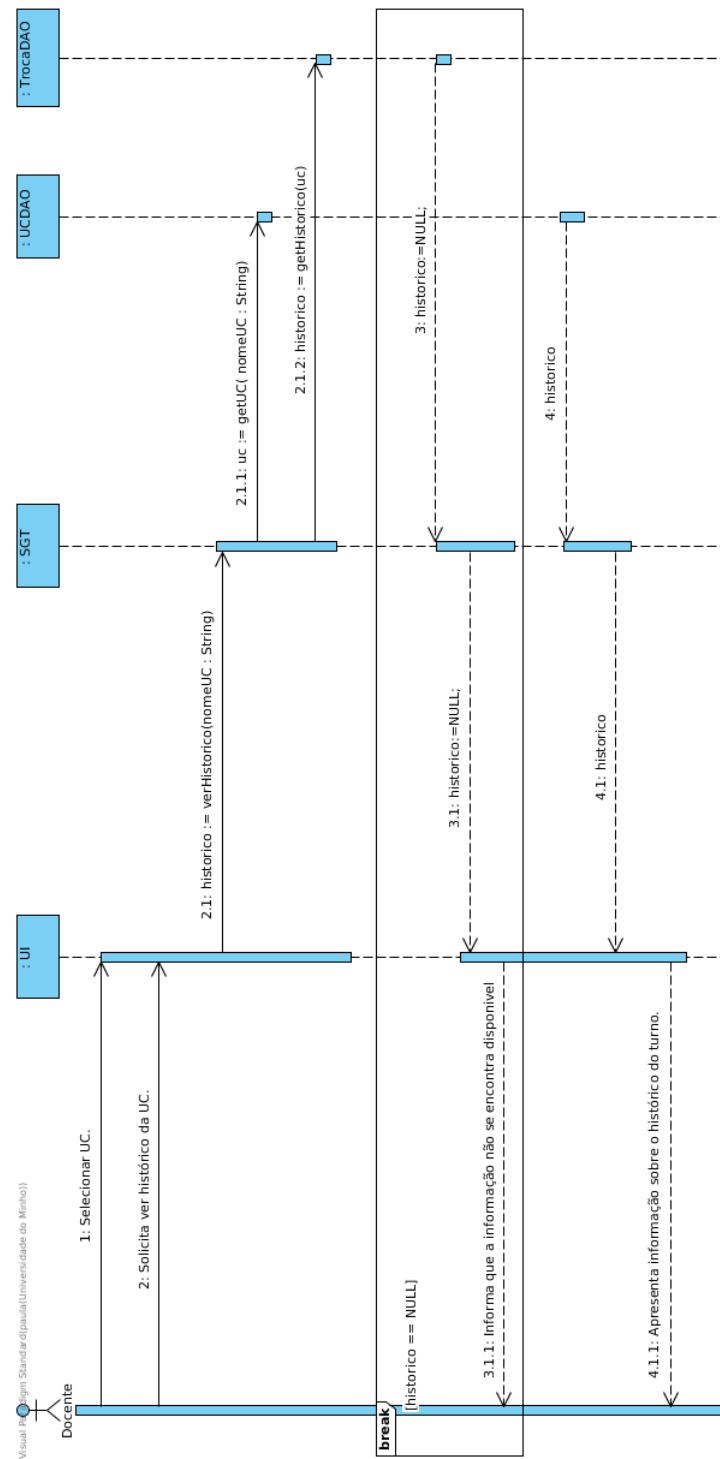


Figura 61: Diagrama de sequencia de implementação do Use Case ”Aceder ao histórico da UC”.

10.11 Enviar pedido de troca

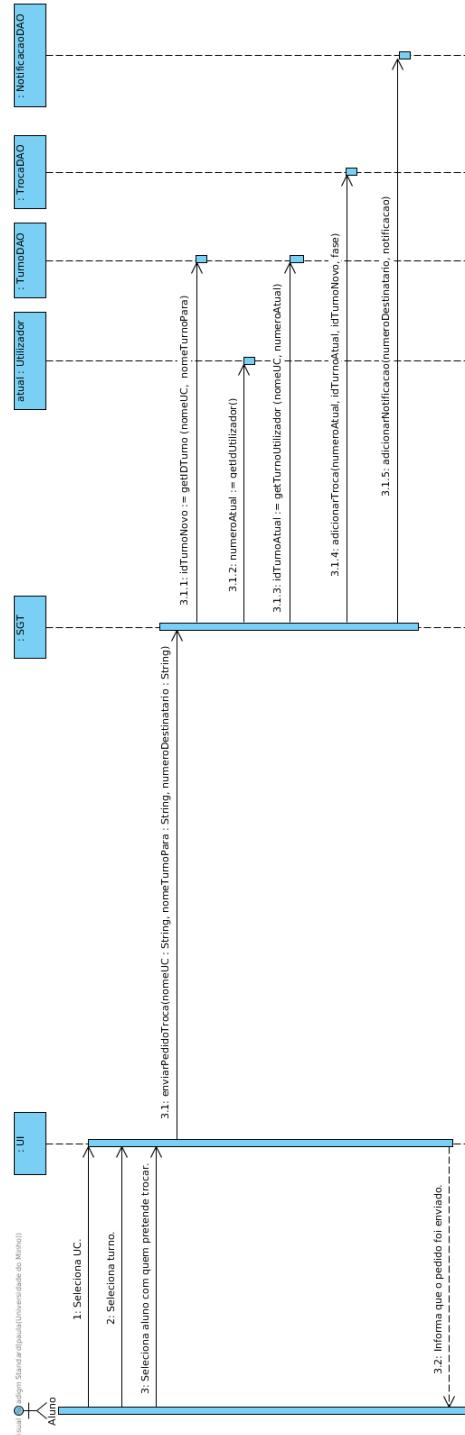


Figura 62: Diagrama de sequencia de implementação do Use Case ”Enviar pedido de troca”.

10.12 Aceitar pedido

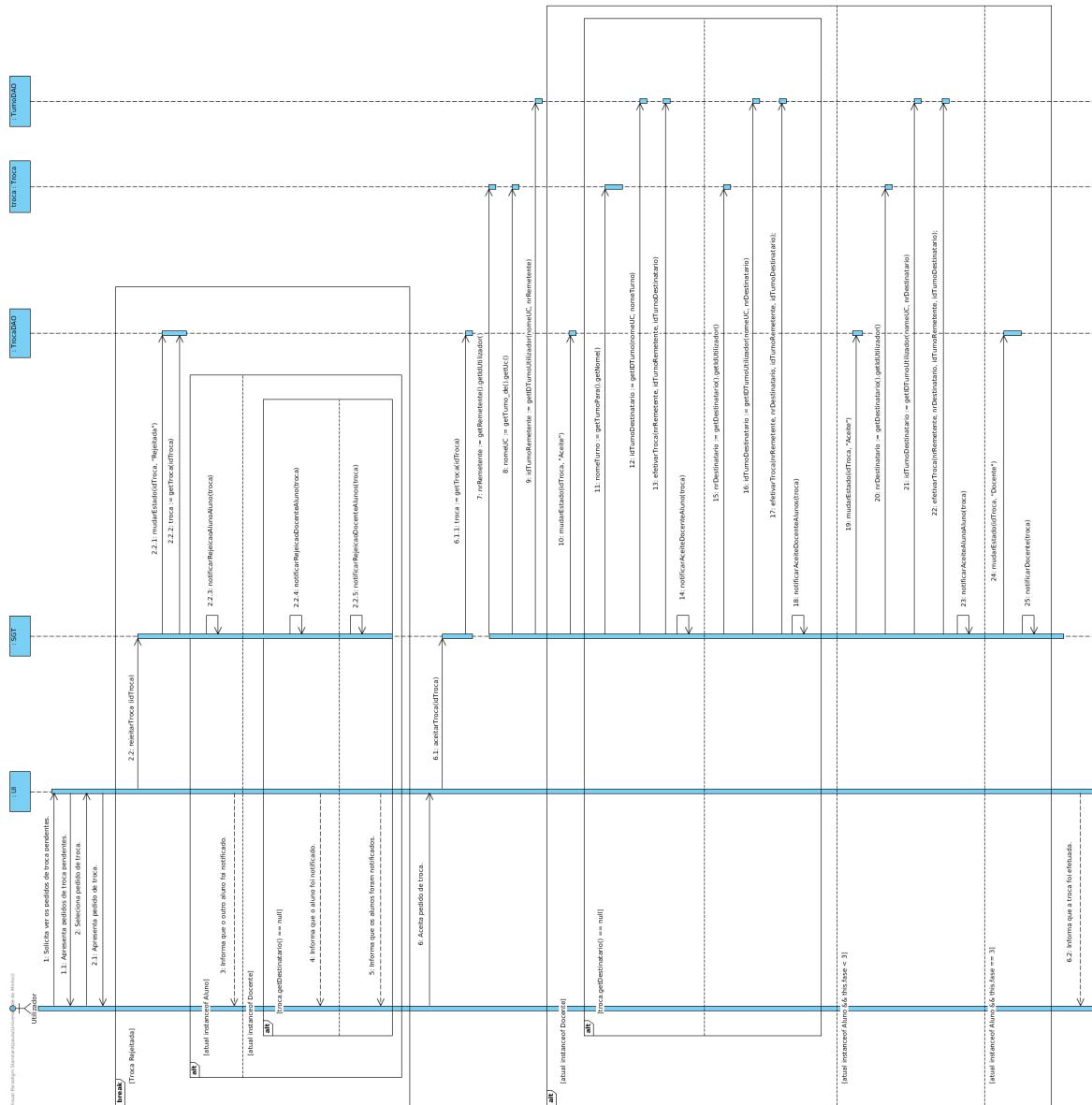


Figura 63: Diagrama de sequencia de implementação do Use Case ”Aceitar pedido”

10.13 Mudar de turno

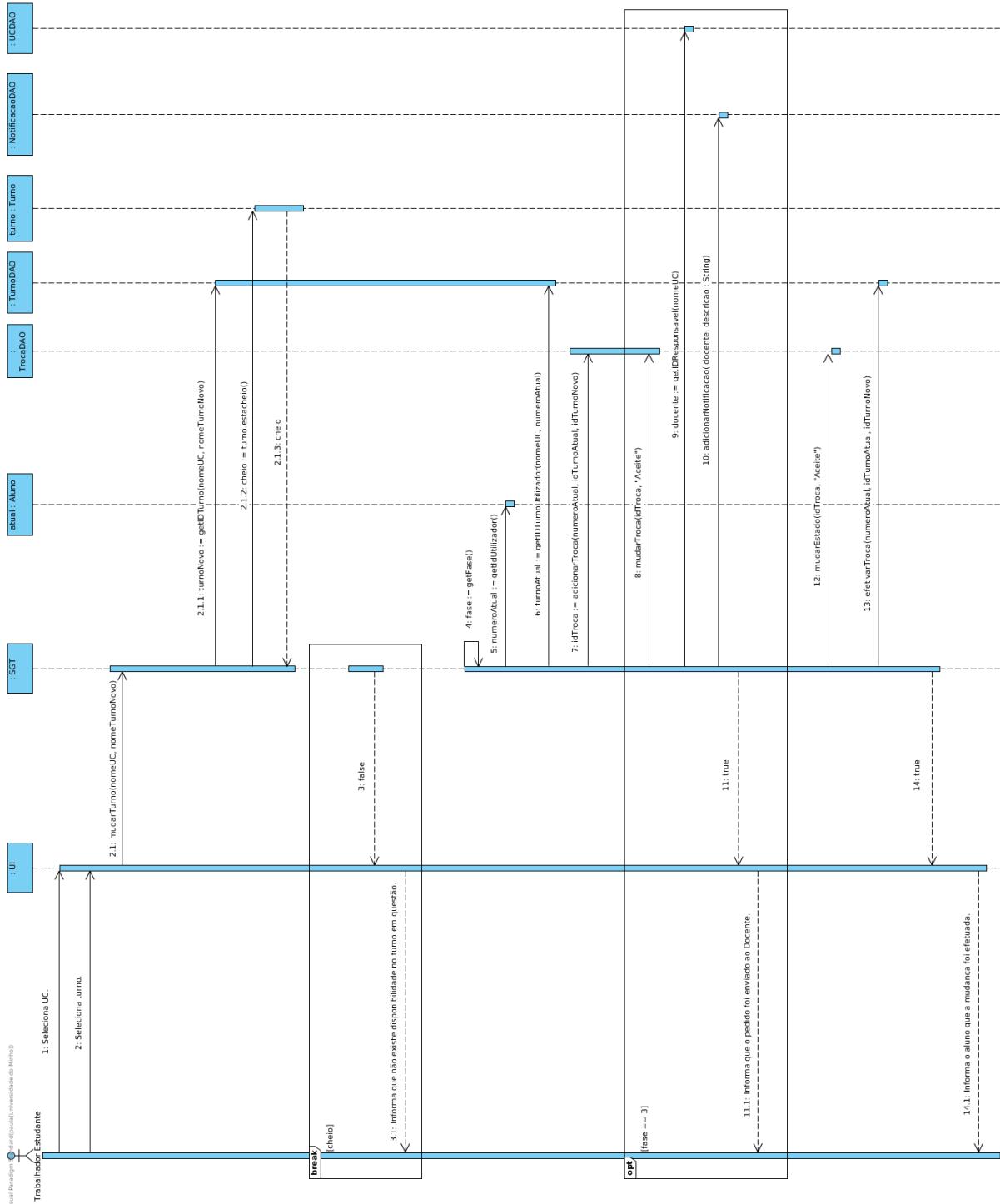


Figura 64: Diagrama de sequencia de implementação do Use Case "Mudar de turno".

10.14 Terminar Sessão

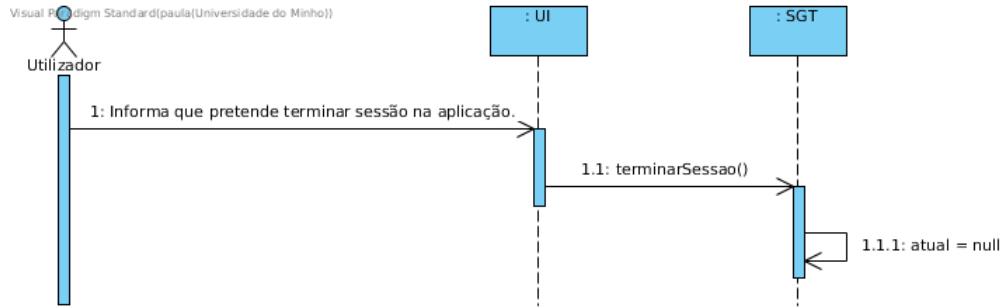


Figura 65: Diagrama de sequencia de implementação do Use Case ”Terminar sessão”.

11 Diagramas de Estado

Nesta secção iremos apresentar alguns diagramas estado, cujo principal objetivo é modelar o comportamento dos objetos/sistemas. Assim sendo, e tendo em conta que nestes diagramas os sistemas são vistos como um todo, estes tornam-se bastante úteis para perceber de que forma funciona a interface com o utilizador.

11.1 Mapa de Navegação

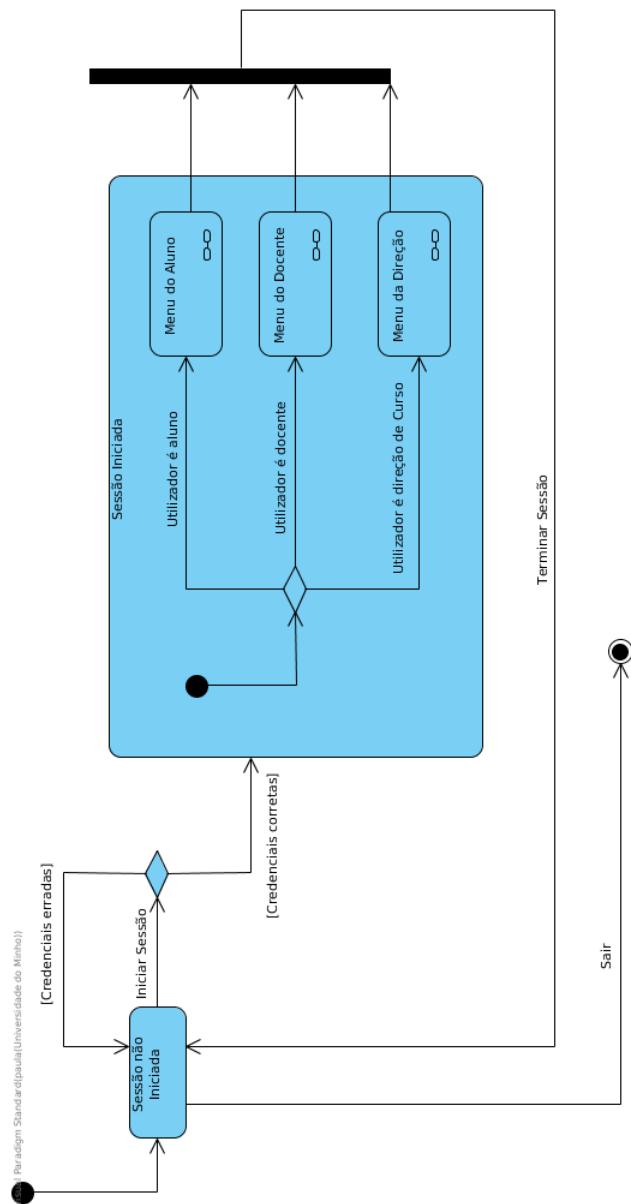


Figura 66: Diagrama de estados - Mapa de Navegação.

11.1.1 Menu do Aluno

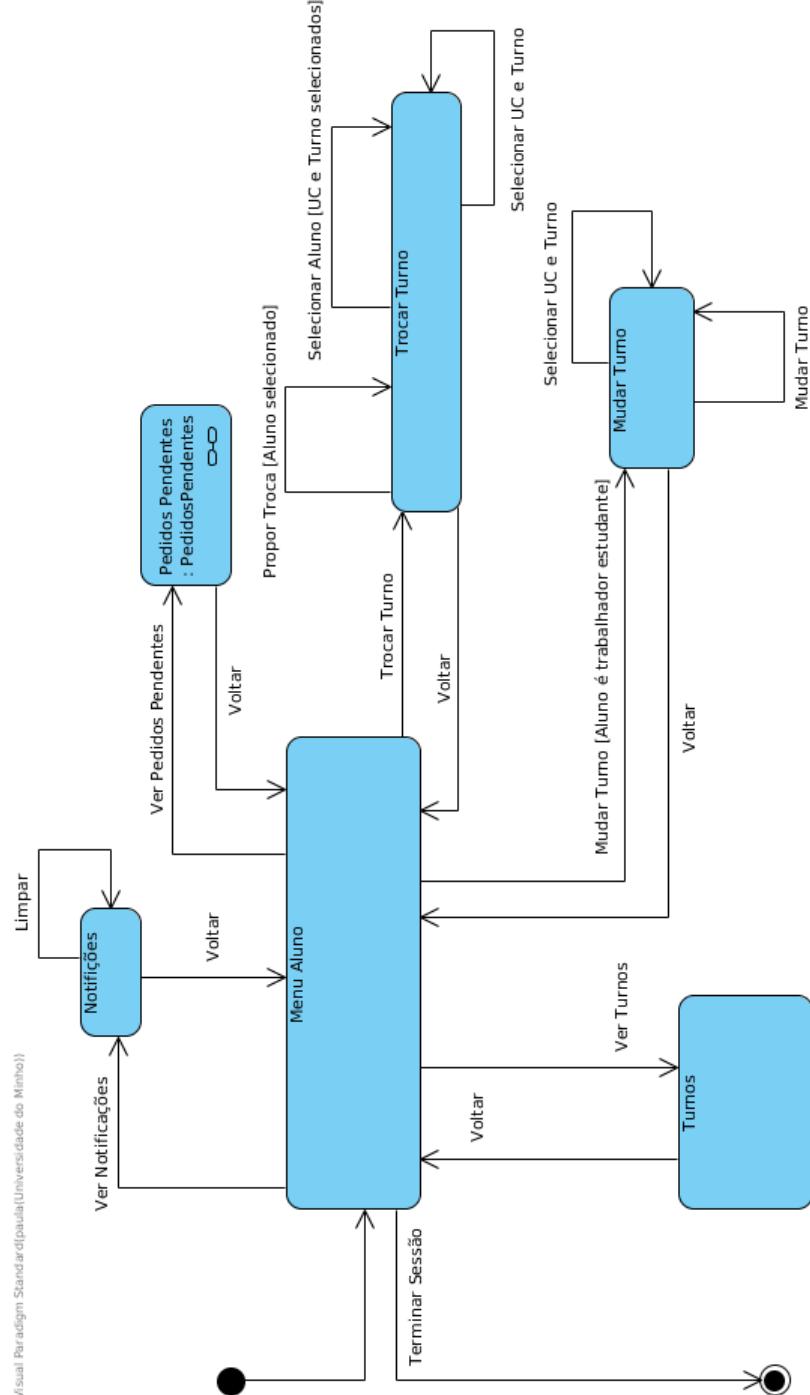


Figura 67: Mapa de Navegação - Menu do Aluno.

11.1.2 Menu do Docente

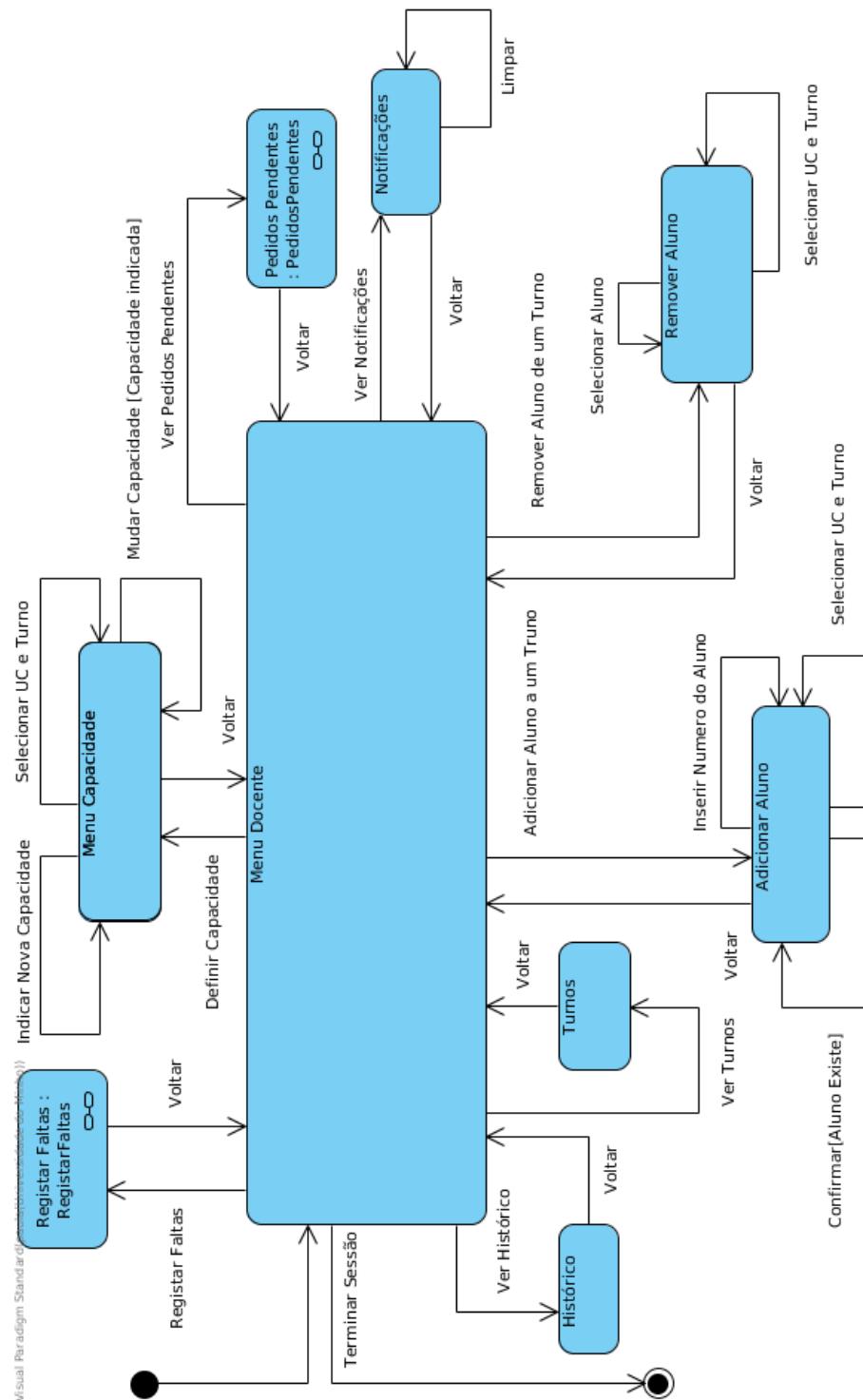


Figura 68: Mapa de Navegação - Menu do Docente.

11.1.3 Menu da Direção

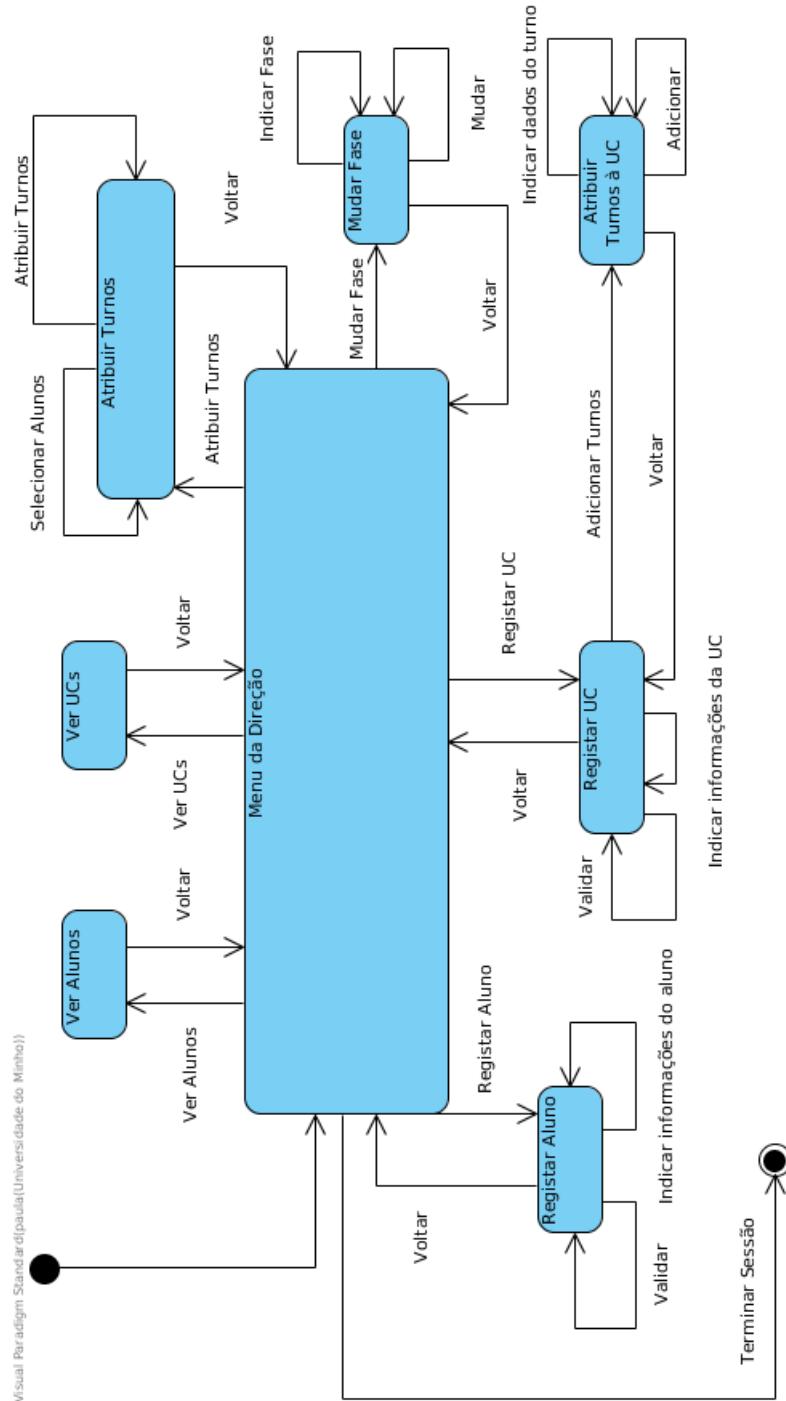


Figura 69: Mapa de Navegação - Menu da Direção.

11.1.4 Menu de Pedidos Pendentes

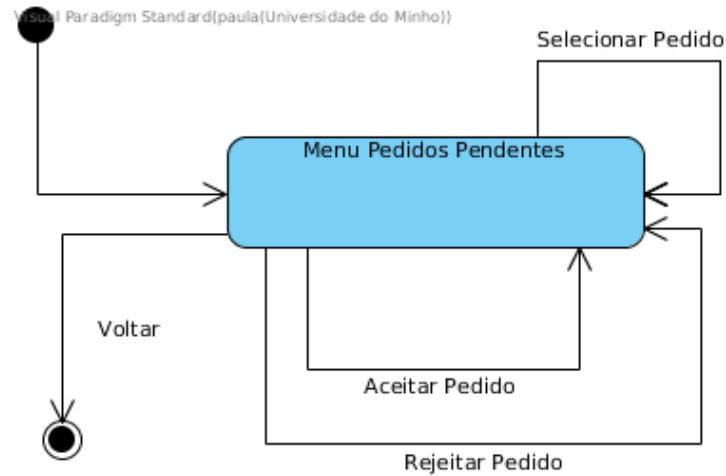


Figura 70: Menu de Pedidos Pendentes.

11.1.5 Registar Faltas

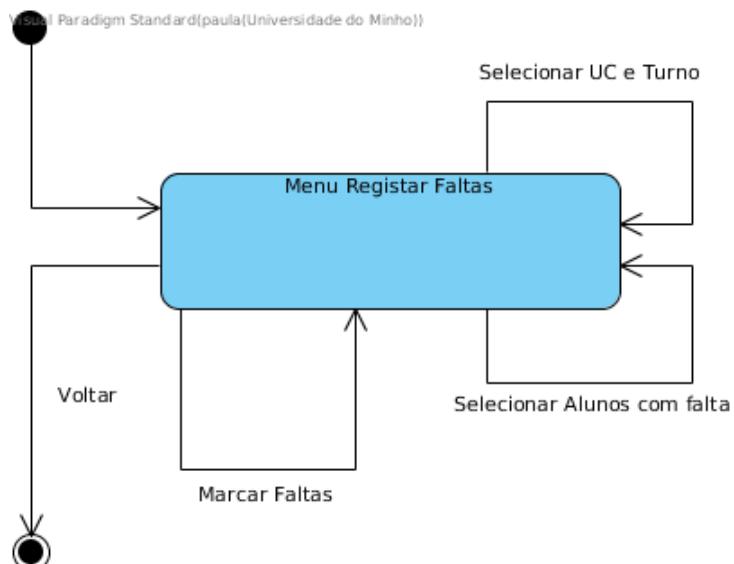


Figura 71: Menu de Registar Faltas.

11.1.6 Estados de um Pedido

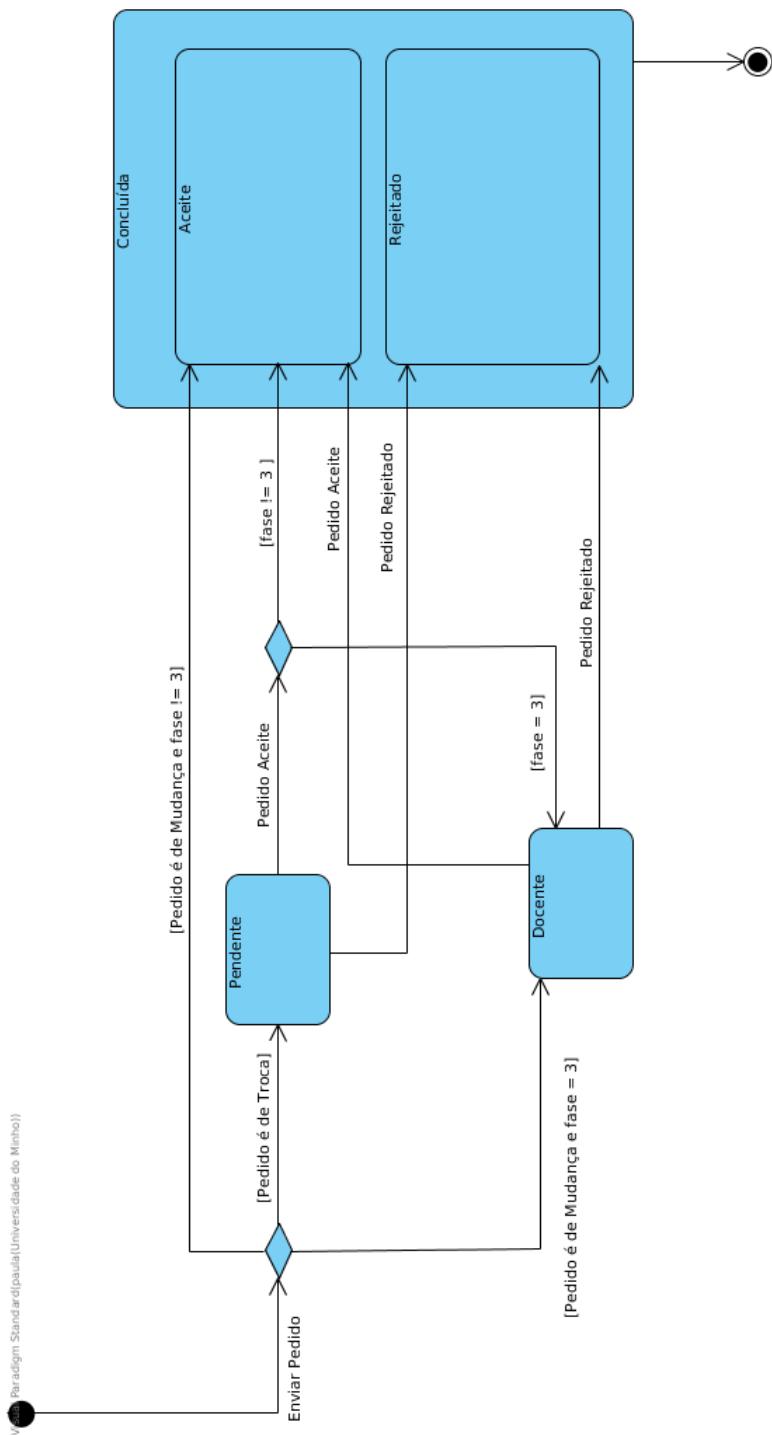


Figura 72: Possíveis estados de um pedido.

12 Diagrama de Instalação

O diagrama de instalação da figura 73 tem como objetivo ilustrar os componentes de hardware sobre a qual o nosso sistema é executado.

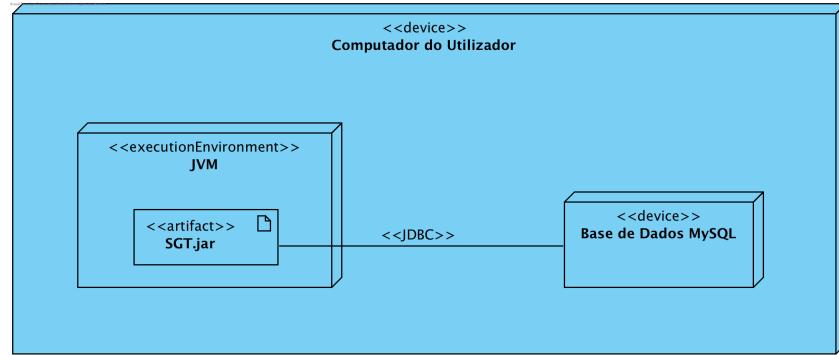


Figura 73: Diagrama de Instalação.

Este diagrama permite concluir que a nossa aplicação é executada no computador do utilizador e que para o seu funcionamento é executado código que procede ao envio de instruções SQL para a uma base de dados relacional em MySQL que contém toda a informação necessária à gestão dos turnos.

13 Idealização da interface gráfica na primeira fase do projeto

Nesta secção relembramos a proposta de interface gráfica apresentada na primeira fase do projeto, tendo em conta apenas o modelo de domínio e os use cases.

13.1 Aluno

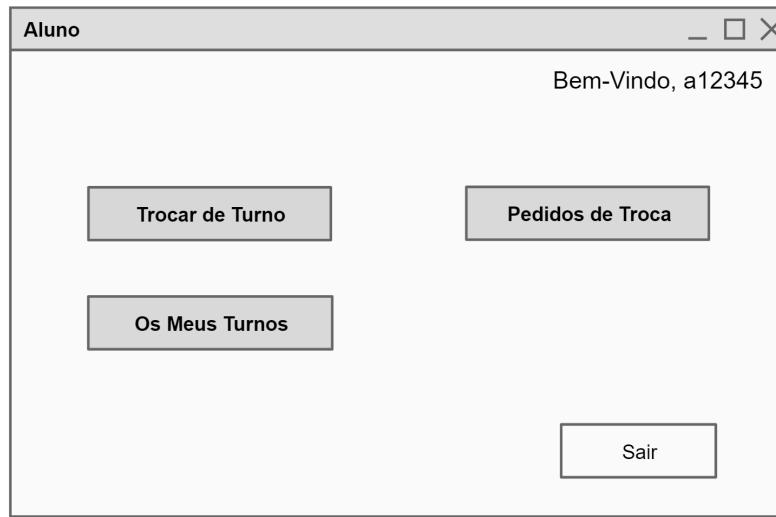


Figura 74: Menu inicial de um aluno.

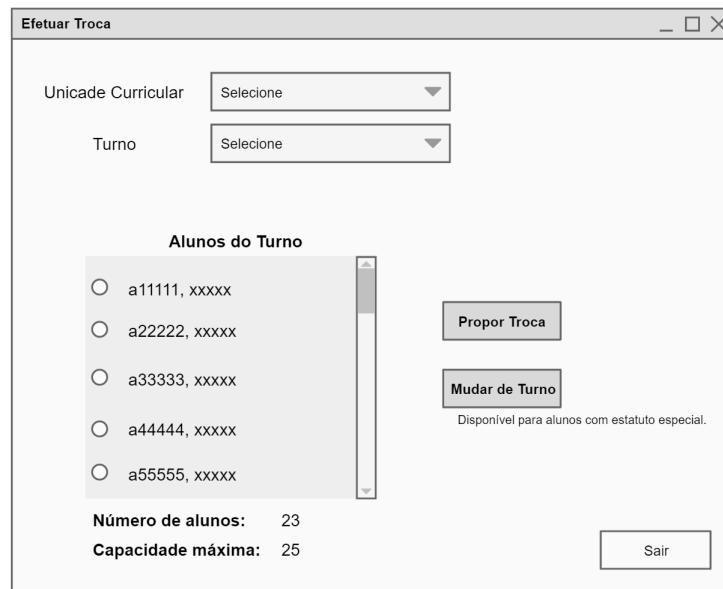


Figura 75: Menu de troca de um aluno.

Turnos	
Unidade Curricular 1	TP_x
Unidade Curricular 2	TP_x
Unidade Curricular 3	TP_x
Unidade Curricular 4	TP_x
Unidade Curricular 5	TP_x
Unidade Curricular 6	TP_x

Figura 76: Turnos de um aluno.

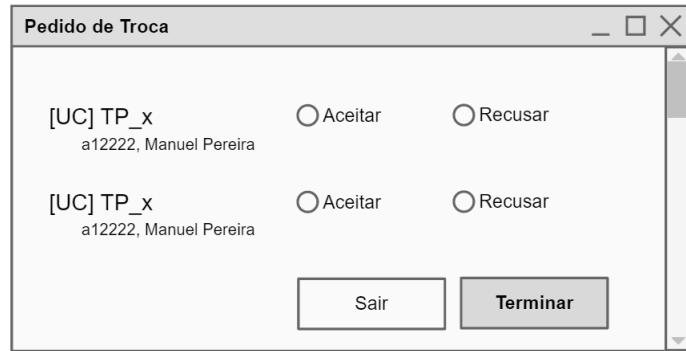


Figura 77: Pedidos de troca pendentes de um aluno.

13.2 Docente

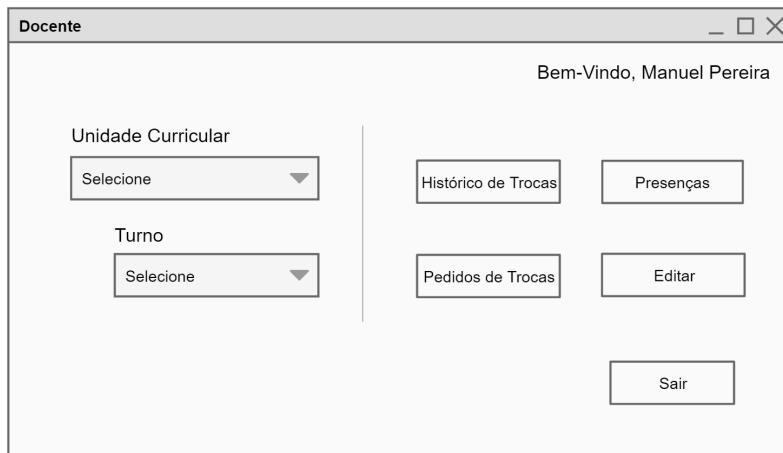


Figura 78: Menu inicial de um docente.

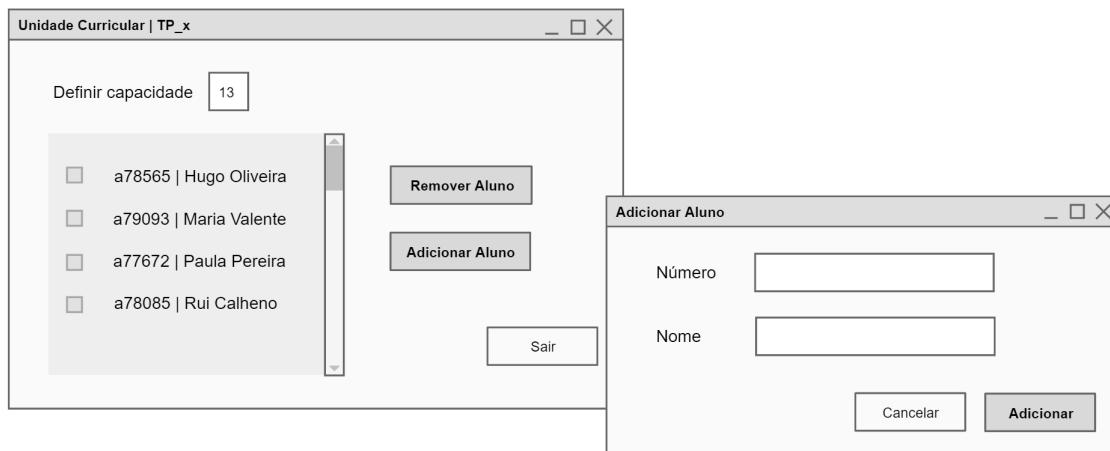


Figura 79: Edição de turnos por parte de um docente.

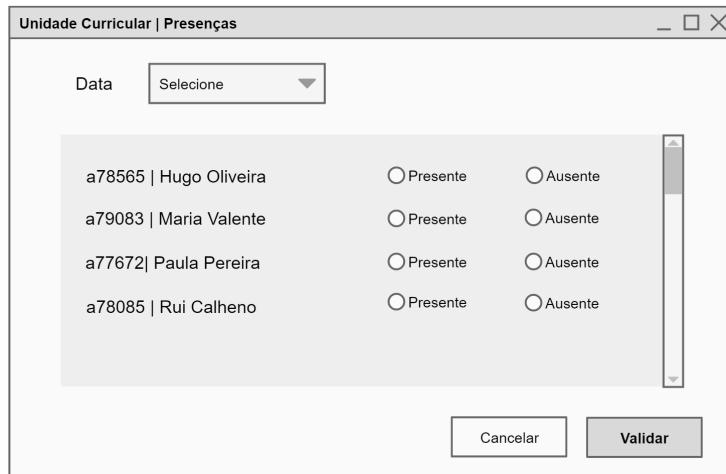


Figura 80: Marcação de faltas por parte de um docente.

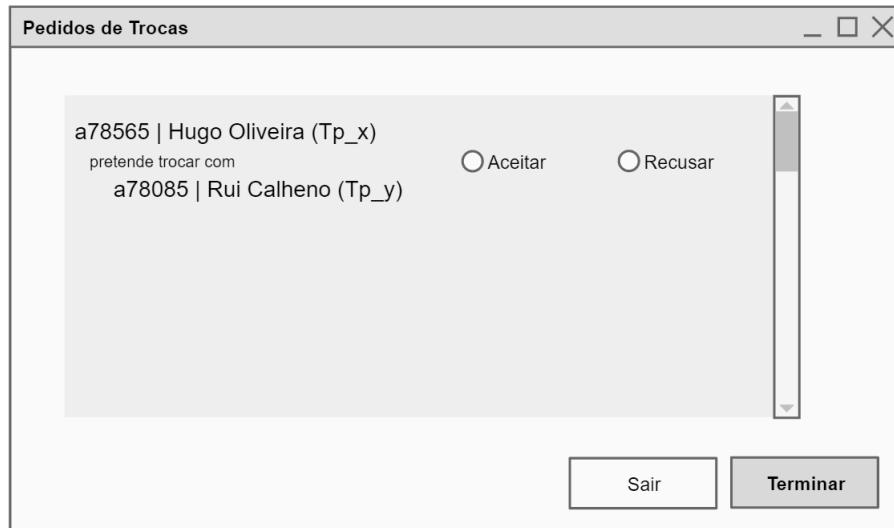


Figura 81: Aceitar Pedidos de troca.

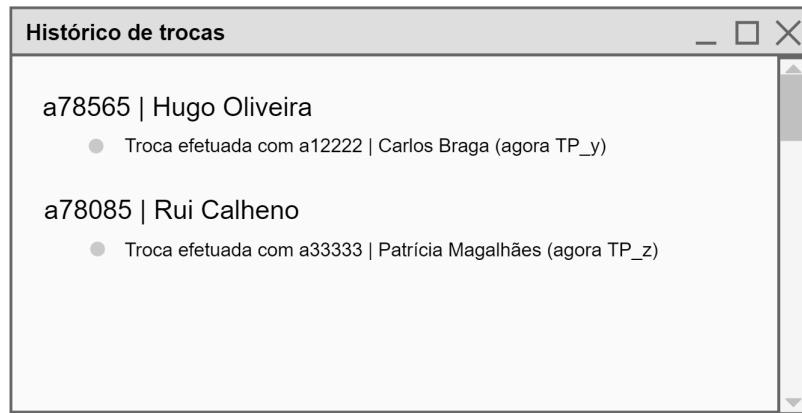


Figura 82: Consultar o histórico de um turno.

13.3 Direção de Curso

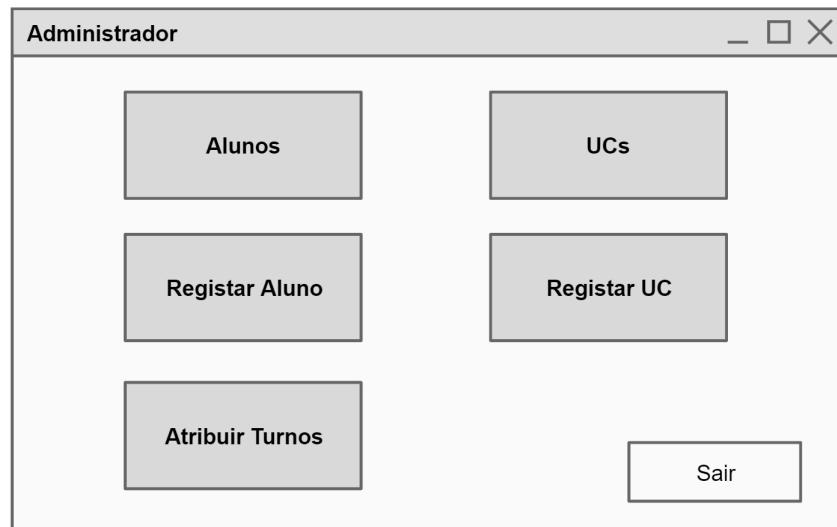


Figura 83: Menu inicial da direção de curso.

14 Interface final

Nesta secção, apresentamos a interface implementada tendo em conta toda a modulação anteriormente apresentada.

O principal objetivo foi alcançar uma aplicação cujo uso fosse bastante intuitivo, ou seja, *user-friendly*. Assim sendo, a implementação que apresentamos, de forma geral, corresponde a um conjunto de botões que vão abrindo novas páginas de acordo com as funcionalidades.

14.1 Iniciar Sessão

Este é o primeiro menu com que qualquer utilizador contacta. Consiste numa janela que permite ao utilizador iniciar sessão na aplicação com as suas credenciais.



Figura 84: Menu inicial para iniciar sessão.

14.2 Menu Principal

Tendo em conta que existem 3 tipos de utilizadores da aplicação, o menu principal depende do tipo de utilizador que inicia sessão. Assim sendo, existe um menu principal direcionado a alunos, a docentes e à direcção de curso que disponibilizam diferentes funcionalidades.

14.2.1 Alunos



Figura 85: Menu principal do aluno sem estatuto.



Figura 86: Menu principal do aluno trabalhador estudante.

Tal como se pode ver pelas figuras 85 e 86, as únicas diferenças entre os menus de estudantes sem estatuto e dos estudantes com estatuto de trabalhador estudante é que o botão "Mudar Turno" só é clicável na segunda versão, uma vez que essa funcionalidade só se encontra disponível para estudantes com estatuto.

14.2.2 Docentes



Figura 87: Menu principal do docente.

Apesar dos docentes terem o mesmo menu, sendo responsável ou não, a funcionalidade de cada botão depende do tipo de docente. Se o docente for responsável por alguma UC, os botões "Editar Capacidade", "Pedidos Pendentes", "Remover Aluno" e "Adicionar Aluno" serão funcionais, caso contrário os botões serão clicáveis mas não terão funcionalidades.

14.2.3 Direção de curso



Figura 88: Menu principal da direção de curso.

14.3 Menus dos alunos

14.3.1 Menu Trocar Turno

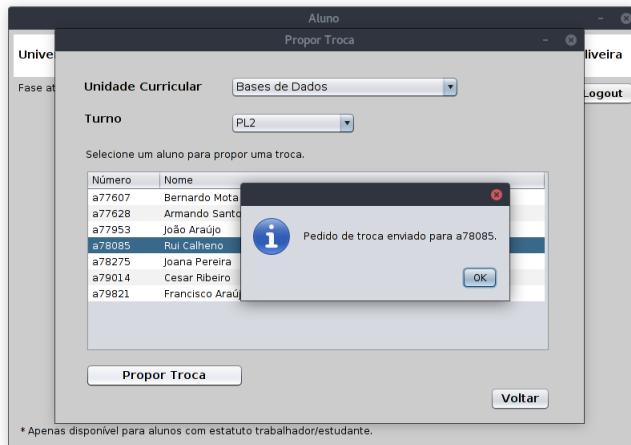


Figura 89: Menu "Trocar Turno".

É a partir deste menu que o aluno seleciona a unidade curricular e o turno para o qual pretende trocar, assim como, o aluno para realizar a troca.

14.3.2 Menu Pedidos Pendentes

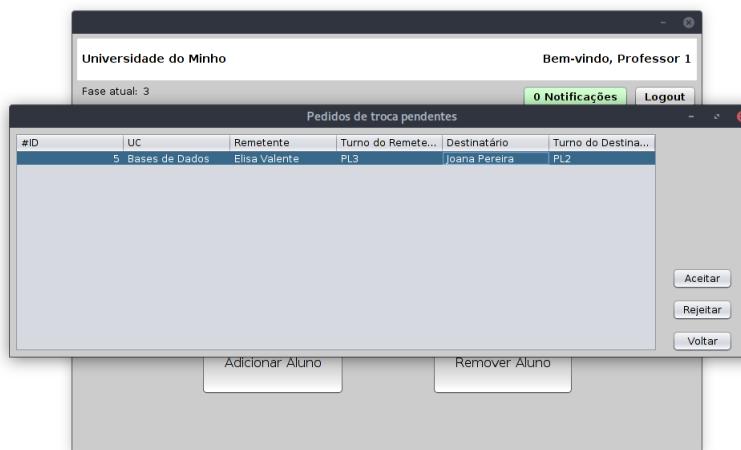


Figura 90: Menu "Pedidos Pendentes".

A partir deste menu, os alunos podem visualizar os pedidos pendentes de troca e proceder à aceitação ou rejeição do pedido.

14.3.3 Menu Mudar Turno

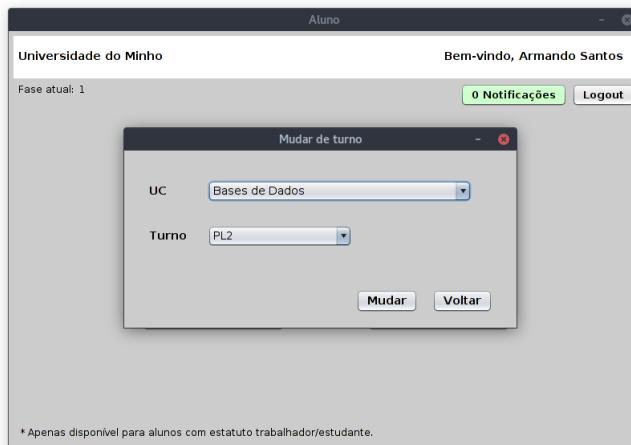


Figura 91: Menu ”Mudar Turno”.

A este menu apenas têm acesso os alunos com estatuto trabalhador-estudante. É a partir deste menu que escolhem a unidade curricular e o turno para o qual pretendem mudar.

14.3.4 Notificações do aluno



Figura 92: Notificações do aluno.

As notificações enviadas para o aluno são apresentadas no canto superior direito do menu principal do aluno. Este poderá ser notificado quando recebe um pedido de troca, quando uma troca é concluída ou ainda, quando o aluno é removido ou adicionado a um turno.

14.4 Menus dos docentes

14.4.1 Menu Registar Faltas

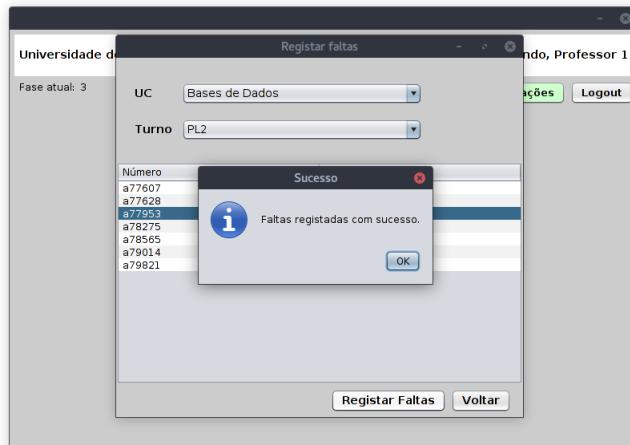


Figura 93: Menu ”Registrar Faltas”.

Este menu é utilizado para registar as faltas relativas a um turno de uma determinada UC.

14.4.2 Menu Ver Histórico

#ID	Remetente	Turno Remetente	Destinatário	Turno Destinatário
1	Hugo Oliveira	PL5	Rui Calheiros	PL2
4	Paula Pereira	PL3	Elisa Valente	PL6
5	Elisa Valente	PL3	Joana Pereira	PL2
6	Armando Santos	PL2		PL5
7	Armando Santos	PL5		PL2
8	Armando Santos	PL2		PL6

 A 'Voltar' (Back) button is at the bottom right of the history table.

Figura 94: Menu ”Ver Histórico”.

Neste menu, os docentes podem ver todas as alterações na constituição dos vários turnos de uma UC.

14.4.3 Menu Editar Capacidade



Figura 95: Menu ”Editar Capacidade”.

Para a alteração da capacidade, o docente responsável tem que selecionar a unidade curricular assim como o turno e inserir o valor da nova capacidade.

14.4.4 Menu Adicionar Aluno

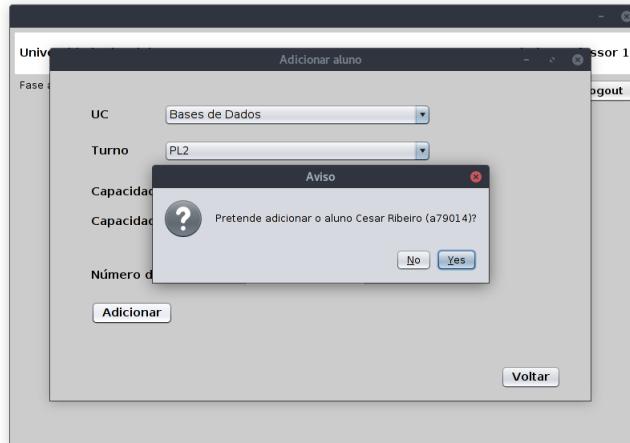


Figura 96: Menu ”Adicionar Aluno”.

Quando o docente responsável por um turno pretende adicionar um aluno ao mesmo, este deverá selecionar a UC e o respetivo turno, assim como o número do aluno a inserir.

14.4.5 Menu Remover Aluno

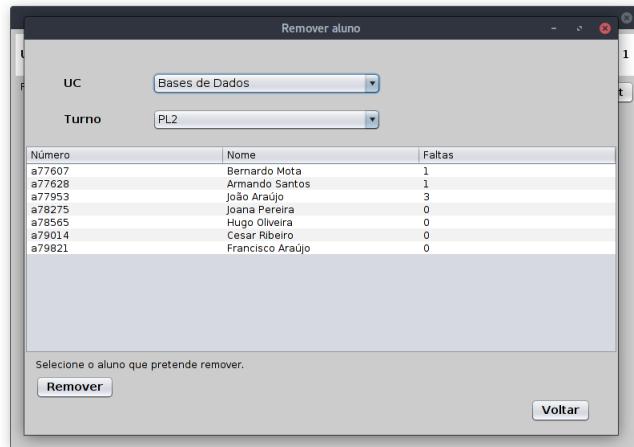


Figura 97: Menu ”Remover Aluno”.

Para a remoção de alunos de um turno, o docente responsável seleciona a UC, o turno e o aluno que pretende remover, confirmando a remoção clicando no botão remover.

14.4.6 Menu Pedidos Pendentes



Figura 98: Menu ”Pedidos Pendentes”.

A partir deste menu os docentes responsáveis podem visualizar os pedidos pendentes de troca ou mudança, e proceder à aceitação ou rejeição do pedido.

14.4.7 Notificações do docente

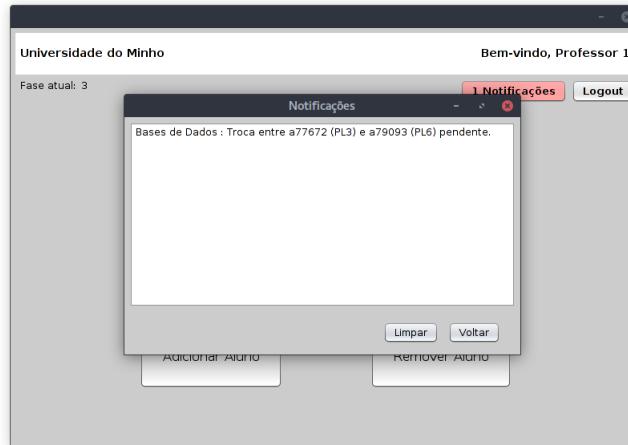


Figura 99: Notificações do docente.

As notificações enviadas para o docente são apresentadas no canto superior direito do menu principal do docente. Este poderá ser notificado quando recebe um pedido de troca entre alunos ou um pedido de mudança de um aluno.

14.5 Menus da direção de curso

14.5.1 Menu Ver Alunos

Número	Nome
a79093	Elisa Valente
a78275	Joana Pereira
a75077	Alberto Faria
a78095	Ricardo Melo
a77953	João Araújo
a79821	Francisco Araújo
a77763	Luís Neto
a78565	Hugo Oliveira
a78961	Tiago Monteiro
a77672	Paula Pereira
a77431	Claudia Correia
a79014	Cesar Ribeiro
a77607	Bernardo Mota
a78508	Fábio Araújo
a77628	Armando Santos
a78848	Raphael Oliveira

Figura 100: Menu "Ver Alunos".

Com este menu a direção de curso tem acesso à lista de todos os alunos do curso.

14.5.2 Menu Ver UCs

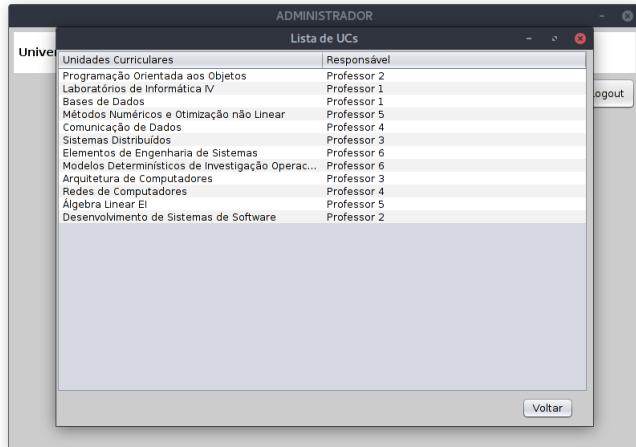


Figura 101: Menu ”Ver UCs”.

Através do botão ”Ver Ucs” a direção de curso tem acesso à lista de todas as unidades curriculares lecionadas no curso.

14.5.3 Menu Registrar Aluno

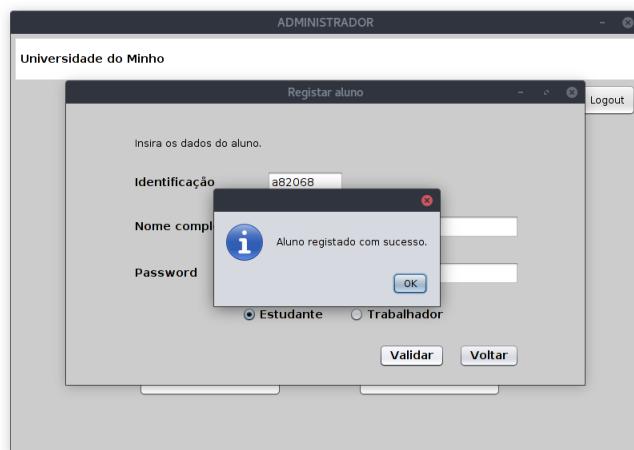


Figura 102: Menu ”Registrar Aluno”.

Para o registo de um aluno, a direção de curso deverá inserir a identificação, o nome, a password e o estatuto do mesmo (estudante ou trabalhador).

14.5.4 Menu Registar UC



Figura 103: Menu ”Registrar UC”.

Para o registo de uma unidade curricular, a direção de curso acede ao menu representado em cima e insere o nome da nova UC, define o regime (semestral ou anual) e o docente responsável. Assim que a aplicação valide os dados inseridos, a direção de curso deverá adicionar os turnos.

14.5.5 Menu Mudar Fase

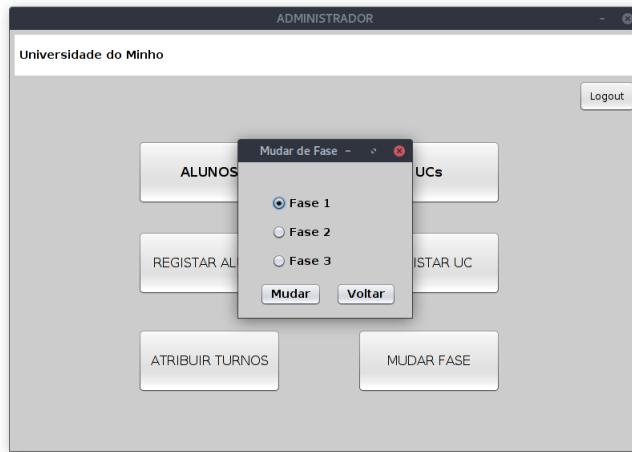


Figura 104: Menu ”Mudar Fase”.

É através deste menu que a direção de curso consegue alterar a fase, tendo apenas de selecionar a fase que pretende iniciar.

15 Base de Dados do projeto

Para o suporte da persistência dos dados na aplicação, criou-se uma base de dados que atendesse os requisitos do sistema.

A base de dados desenvolvida trata-se de uma base de dados relacional desenvolvida em *MySQL*, cujo modelo lógico é o apresentado na figura 105.

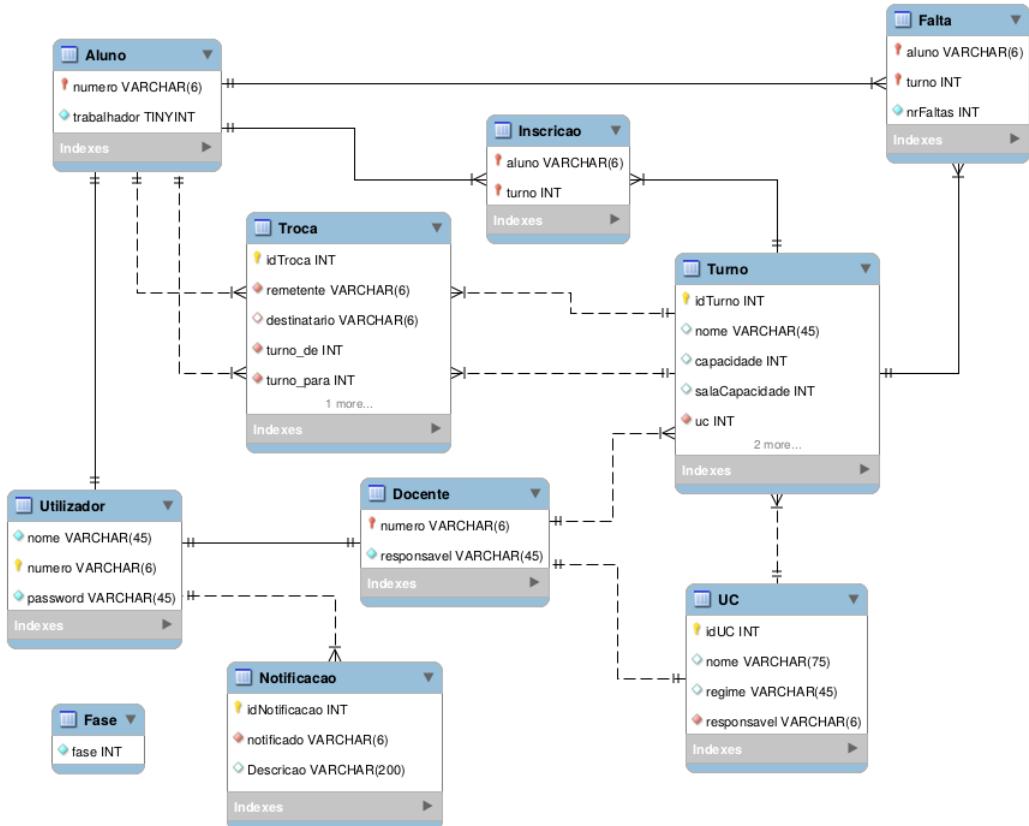


Figura 105: Modelo lógico da base de dados.

16 Conclusão

Para a realização deste trabalho, em primeiro lugar, realizou-se um estudo dos requisitos. Depois de estes estarem perfeitamente compreendidos, procedeu-se à criação do modelo de domínio.

Este permitiu perceber o contexto do problema e extrair, não só os atores do sistema, mas também as funcionalidades que teriam de ser disponibilizadas, obtendo-se assim os diagramas de use case e as respetivas especificações. Para que estas fossem de fácil compreensão, usamos a notação tabular que se revelou útil na fase seguinte: a conceção dos diagramas de sequência.

Posteriormente, com o intuito de mostrar mais detalhadamente o funcionamento do sistema, identificaram-se os diversos subsistemas presentes no modelo de domínio e, procedeu-se à atribuição de responsabilidades através dos diagramas de sequência de subsistemas.

Posto isto, partiu-se para a elaboração de diagramas de classes que nos permitiram definir as classes, bem como os respetivos atributos e métodos que foram utilizados para construir a aplicação.

De seguida, para obter uma visão mais organizada das classes definidas construiu-se um diagrama de *packages* onde estas foram agrupadas tendo em conta o seu contexto na aplicação.

A realização destes diagramas permitiu ter uma ideia clara de como o sistema funciona perante cada situação, e que tarefas são realizadas para atingir determinados objetivos. Assim sendo, realizaram-se os os diagramas de sequência de implementação, que mostram quais os métodos que são executados e quais as classes envolvidas.

Para modelar o comportamento dos objetos e, perceber de que forma iria funcionar a interface com o utilizador, procedeu-se à construção dos diagramas de estado.

Finalmente, foi concebido o diagrama de instalação tendo em conta os componentes de hardware sobre os quais a aplicação funciona.

Terminado o processo de desenvolvimento da aplicação e fazendo uma avaliação crítica do projeto, consideramos que este possui diversos pontos fortes, visto que se baseia num conjunto de diagramas e procedimentos regrados e robustos, o que resultou numa implementação organizada e de simples entendimento. A interface gráfica também se demonstra bastante intuitiva e fácil de usar, com especial foco para o sistema de trocas e notificações, que permite avisar os alunos quando as trocas, adições e remoções ocorrem.

No entanto, apesar do trabalho se encontrar realizado na sua totalidade e de estarmos satisfeitos com o resultado final, sentimos que a implementação da persistência no projeto (diagrama ORM e classes DAO) poderia ter sido executada de uma melhor forma. Contudo, isto não nos foi possível devido ao facto de já nos encontrarmos numa fase de desenvolvimento avançada, sendo que, a realização dessas melhorias ia resultar numa mudança significativa da aplicação até então codificada.

Face a isto, percebemos que o planeamento e modelação de um projeto são tão, ou mais importantes que a escrita do código. Pelo que, o tempo despendido na idealização do software revela-se como sendo uma mais valia para a sua futura

implementação.

Consideramos, assim, que o resultado final obtido corresponde às expectativas iniciais do projeto, alcançando os objetivos subjacentes à realização deste trabalho.

17 Anexos

17.1 Script de Criação MySQL

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-----
-- Schema Sgt
-----

-----
-- Schema Sgt
-----

CREATE SCHEMA IF NOT EXISTS `Sgt` DEFAULT CHARACTER SET utf8 ;
USE `Sgt` ;

-----
-- Table `Sgt`.`Utilizador`
-----

CREATE TABLE IF NOT EXISTS `Sgt`.`Utilizador` (
  `nome` VARCHAR(45) NOT NULL,
  `numero` VARCHAR(6) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`numero`)
)
ENGINE = InnoDB;

-----
-- Table `Sgt`.`Aluno`
-----

CREATE TABLE IF NOT EXISTS `Sgt`.`Aluno` (
  `numero` VARCHAR(6) NOT NULL,
  `trabalhador` TINYINT NOT NULL,
  PRIMARY KEY (`numero`),
  CONSTRAINT `fk_Aluno_Utilizador`
    FOREIGN KEY (`numero`)
    REFERENCES `Sgt`.`Utilizador` (`numero`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
ENGINE = InnoDB;
```

```
-- Table `Sgt`.`Docente`
-----
CREATE TABLE IF NOT EXISTS `Sgt`.`Docente` (
    `numero` VARCHAR(6) NOT NULL,
    `responsavel` VARCHAR(45) NOT NULL,
    PRIMARY KEY (`numero`),
    CONSTRAINT `fk_Docente_Utilizador`
        FOREIGN KEY (`numero`)
        REFERENCES `Sgt`.`Utilizador` (`numero`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `Sgt`.`UC`
-----
CREATE TABLE IF NOT EXISTS `Sgt`.`UC` (
    `idUC` INT NOT NULL AUTO_INCREMENT,
    `nome` VARCHAR(75) NULL,
    `regime` VARCHAR(45) NULL,
    `responsavel` VARCHAR(6) NOT NULL,
    PRIMARY KEY (`idUC`),
    INDEX `fk_UC_Docente1_idx` (`responsavel` ASC),
    CONSTRAINT `fk_UC_Docente1`
        FOREIGN KEY (`responsavel`)
        REFERENCES `Sgt`.`Docente` (`numero`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `Sgt`.`Turno`
-----
CREATE TABLE IF NOT EXISTS `Sgt`.`Turno` (
    `idTurno` INT NOT NULL AUTO_INCREMENT,
    `nome` VARCHAR(45) NULL,
    `capacidade` INT NULL,
    `salaCapacidade` INT NULL,
    `uc` INT NOT NULL,
    `docente` VARCHAR(6) NOT NULL,
    `aulasPrevistas` INT NOT NULL,
    PRIMARY KEY (`idTurno`),
    INDEX `fk_Turno_UC1_idx` (`uc` ASC),
    INDEX `fk_Turno_Docente1_idx` (`docente` ASC),
```

```

CONSTRAINT `fk_Turno_UC1`
    FOREIGN KEY (`uc`)
    REFERENCES `Sgt`.`UC` (`idUC`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Turno_Docente1`
    FOREIGN KEY (`docente`)
    REFERENCES `Sgt`.`Docente` (`numero`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- Table `Sgt`.`Aula`

CREATE TABLE IF NOT EXISTS `Sgt`.`Aula` (
    `idAula` INT NOT NULL AUTO_INCREMENT,
    `numero` INT NULL,
    `data` DATETIME NULL,
    `turno` INT NOT NULL,
    PRIMARY KEY (`idAula`),
    INDEX `fk_Aula_Turno1_idx` (`turno` ASC),
    CONSTRAINT `fk_Aula_Turno1`
        FOREIGN KEY (`turno`)
        REFERENCES `Sgt`.`Turno` (`idTurno`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- Table `Sgt`.`Inscricao`

CREATE TABLE IF NOT EXISTS `Sgt`.`Inscricao` (
    `aluno` VARCHAR(6) NOT NULL,
    `turno` INT NOT NULL,
    PRIMARY KEY (`aluno`, `turno`),
    INDEX `fk_Aluno_has_Turno_Turno1_idx` (`turno` ASC),
    INDEX `fk_Inscricao_Aluno1_idx` (`aluno` ASC),
    CONSTRAINT `fk_Aluno_has_Turno_Turno1`
        FOREIGN KEY (`turno`)
        REFERENCES `Sgt`.`Turno` (`idTurno`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Inscricao_Aluno1`
```

```
FOREIGN KEY (`aluno`)
  REFERENCES `Sgt`.`Aluno` (`numero`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `Sgt`.`Troca`
-- -----
CREATE TABLE IF NOT EXISTS `Sgt`.`Troca` (
  `idTroca` INT NOT NULL AUTO_INCREMENT,
  `remetente` VARCHAR(6) NOT NULL,
  `destinatario` VARCHAR(6) NULL,
  `turno_de` INT NOT NULL,
  `turno_para` INT NOT NULL,
  `estado` VARCHAR(45) NULL,
  PRIMARY KEY (`idTroca`),
  INDEX `fk_Troca_Turno1_idx` (`turno_de` ASC),
  INDEX `fk_Troca_Turno2_idx` (`turno_para` ASC),
  INDEX `fk_TrocaAluno1_idx` (`remetente` ASC),
  INDEX `fk_Troca_Aluno2_idx` (`destinatario` ASC),
  CONSTRAINT `fk_Troca_Turno1`
    FOREIGN KEY (`turno_de`)
    REFERENCES `Sgt`.`Turno` (`idTurno`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Troca_Turno2`
    FOREIGN KEY (`turno_para`)
    REFERENCES `Sgt`.`Turno` (`idTurno`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Troca_Aluno1`
    FOREIGN KEY (`remetente`)
    REFERENCES `Sgt`.`Aluno` (`numero`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Troca_Aluno2`
    FOREIGN KEY (`destinatario`)
    REFERENCES `Sgt`.`Aluno` (`numero`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `Sgt`.`Falta`
-----
CREATE TABLE IF NOT EXISTS `Sgt`.`Falta` (
  `aluno` VARCHAR(6) NOT NULL,
  `turno` INT NOT NULL,
  `nrFaltas` INT NOT NULL,
  INDEX `fk_Falta_Turno1_idx` (`turno` ASC),
  PRIMARY KEY (`aluno`, `turno`),
  CONSTRAINT `fk_Aluno_has_Aula_Aluno1`
    FOREIGN KEY (`aluno`)
    REFERENCES `Sgt`.`Aluno` (`numero`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Falta_Turno1`
    FOREIGN KEY (`turno`)
    REFERENCES `Sgt`.`Turno` (`idTurno`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `Sgt`.`Notificacao`
-----
CREATE TABLE IF NOT EXISTS `Sgt`.`Notificacao` (
  `idNotificacao` INT NOT NULL AUTO_INCREMENT,
  `notificado` VARCHAR(6) NOT NULL,
  `Descricao` VARCHAR(200) NULL,
  PRIMARY KEY (`idNotificacao`),
  INDEX `fk_Notificacao_Utilizador_idx` (`notificado` ASC),
  CONSTRAINT `fk_Notificacao_Utilizador`
    FOREIGN KEY (`notificado`)
    REFERENCES `Sgt`.`Utilizador` (`numero`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `Sgt`.`Fase`
-----
CREATE TABLE IF NOT EXISTS `Sgt`.`Fase` (
  `fase` INT NOT NULL)
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```