



INSTITUTO POLITÉCNICO DE BEJA  
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

LICENCIATURA EM ENGENHARIA INFORMÁTICA  
PROGRAMAÇÃO CENTRADA NA REDE

PROJECTO DA DISCIPLINA - FASE DE DESENHO

**SISTEMA DE GESTÃO  
DE MOMENTOS DE AVALIAÇÃO**

ANO LECTIVO: 2012/2013 / 2º SEMESTRE / 3º ANO

*Aluno:* Pedro MOREIRA N.10015

*Docente:* Luís BRUNO

25 de Setembro de 2013



# Conteúdo

<b>Índice Geral</b>	<b>3</b>
<b>Lista de Figuras</b>	<b>3</b>
<b>1 Introdução</b>	<b>5</b>
1.1 Objectivos . . . . .	5
1.2 Problema . . . . .	5
1.3 Solução . . . . .	5
<b>2 Descrição Geral</b>	<b>7</b>
2.1 Perspectiva . . . . .	7
2.1.1 Interfaces . . . . .	7
2.1.2 Funções . . . . .	7
2.1.3 Características do Utilizador . . . . .	8
<b>3 Diagrama de Casos de Uso</b>	<b>9</b>
3.1 EFECTUAR LOGIN . . . . .	9
3.1.1 Descrição . . . . .	9
3.1.2 Actores . . . . .	9
3.1.3 Cenário Principal . . . . .	9
3.1.4 Extensões ou Variações . . . . .	10
3.2 LISTAR AVALIAÇÕES . . . . .	11
3.2.1 Descrição . . . . .	11
3.2.2 Pré-condições . . . . .	11
3.2.3 Actores . . . . .	11
3.2.4 Cenário Principal . . . . .	11
3.2.5 Extensões ou Variações . . . . .	11
3.3 VALIDAR AVALIAÇÕES . . . . .	12
3.3.1 Descrição . . . . .	12
3.3.2 Pré-condições . . . . .	12
3.3.3 Actores . . . . .	12

3.3.4	Cenário Principal . . . . .	12
3.3.5	Extensões ou Variações . . . . .	12
3.4	MARCAR AVALIAÇÃO . . . . .	12
3.4.1	Descrição . . . . .	12
3.4.2	Pré-condições . . . . .	13
3.4.3	Actores . . . . .	13
3.4.4	Cenário Principal . . . . .	13
3.4.5	Extensões ou Variações . . . . .	13
3.5	INSCREVER EM AVALIAÇÃO . . . . .	14
3.5.1	Descrição . . . . .	14
3.5.2	Pré-condições . . . . .	14
3.5.3	Actores . . . . .	14
3.5.4	Cenário Principal . . . . .	14
3.5.5	Extensões ou Variações . . . . .	14
<b>4</b>	<b>Modelação de Interfaces</b>	<b>15</b>
4.1	Efectuar Login . . . . .	15
4.2	Inscriver em Momento de Avaliação . . . . .	16
4.3	Marcar e Cancelar Momento de Avaliação . . . . .	17
4.4	Validar e Cancelar Validação de Momentos de Avaliação . . . . .	18
4.5	Listar Momentos de Avaliação . . . . .	19
<b>5</b>	<b>Modelação da Base de Dados</b>	<b>20</b>
<b>6</b>	<b>Modelação UML</b>	<b>23</b>
6.1	Diagrama de Classes . . . . .	23
6.2	Diagramas de Sequência . . . . .	23
6.2.1	Efectuar Login . . . . .	23
6.2.2	Listar Avaliações . . . . .	27
6.2.3	Inscrição em Avaliação . . . . .	29
6.2.4	Marcar Avaliação . . . . .	31
6.2.5	Validar Avaliações . . . . .	33
<b>7</b>	<b>Implementação</b>	<b>35</b>
7.1	Introdução . . . . .	35
7.2	Decisões Globais de Implementação . . . . .	35
7.2.1	Tecnologias Utilizadas . . . . .	35
7.2.2	Armazenamento dos Dados . . . . .	35
7.2.3	Hierarquia de Ficheiros . . . . .	35
7.3	Decisões de Implementação Específicas . . . . .	36

7.3.1	Ligação à Base de Dados . . . . .	36
7.3.2	Funcionamento do controlador principal . . . . .	36
7.3.3	Login . . . . .	36
<b>8</b>	<b>Conclusão</b>	<b>37</b>

# Lista de Figuras

3.1	Diagrama: Casos de Uso . . . . .	10
4.1	Interface: Login . . . . .	15
4.2	Interface: Efectuar ou Cancelar Inscrição em Momento de Avaliação . . . . .	16
4.3	Interface: Marcar Novo Momento de Avaliação . . . . .	17
4.4	Interface: Validar ou Cancelar Validação de Momento de Avaliação . . . . .	18
5.1	Base de Dados: Modelo Físico . . . . .	22
6.1	UML: Diagrama de Classes . . . . .	24
6.2	Diagrama de Sequência: Login . . . . .	25
6.3	Diagrama de Sequência: Logout . . . . .	26
6.4	Diagrama de Sequência: Listar Avaliações Entre Datas . . . . .	27
6.5	Diagrama de Sequência: Listar Detalhes de Avaliação . . . . .	28
6.6	Diagrama de Sequência: Efectuar Inscrição em Momento de Avaliação . . . . .	29
6.7	Diagrama de Sequência: Cancelar Inscrição de Momento de Avaliação . . . . .	30
6.8	Diagrama de Sequência: Novo Momento de Avaliação . . . . .	31
6.9	Diagrama de Sequência: Cancelar Momento de Avaliação . . . . .	32
6.10	Diagrama de Sequência: Validar Avaliação . . . . .	33
6.11	Diagrama de Sequência: Cancelar Validação de Momento de Avaliação . . . . .	34

# Capítulo 1

## Introdução

### 1.1 Objectivos

Este relatório insere-se na avaliação da disciplina "Programação Centrada na Rede", leccionada no 2º semestre do 3º ano da licenciatura em Engenharia Informática.

O trabalho surge na sequência do projecto interdisciplinar de Engenharia de Software, Bases de Dados I e II, Hipermédia e Acessibilidade e, por último, Programação Centrada na Rede.

O tema é o desenvolvimento de uma aplicação que permita gerir os momentos de avaliação das diferentes disciplinas dos cursos da Estig. Nele pretende-se apresentar um resumo da análise, fase de desenho e implementação da aplicação para seis casos de uso, dois para cada um dos actores envolvidos na utilização da aplicação.

### 1.2 Problema

Actualmente, cada docente marca as suas diferentes avaliações de forma individual, podendo ou não enviar essa informação ao Coordenador de Curso. Este, para ter noção da distribuição de carga dos momentos de avaliação ao longo de um semestre tem que, manualmente, registar essa informação, que poderá ser alterada posteriormente, sem que ele receba qualquer notificação. Por outro lado não existe um repositório que centralize todos os momentos de avaliação realizados nas disciplinas ao longo do semestre e que seja transparente para os seus diferentes beneficiários.

### 1.3 Solução

Neste projecto pretende-se a criação de uma aplicação na linguagem PHP, com suporte de MySQL, que centralize o registo de todos os momentos de avaliação mantendo os respectivos

intervenientes actualizados acerca das alterações às mesmas.

Por forma a evitar uma carga excessiva aos alunos, as avaliações terão obrigatoriamente que ser analisadas pelo coordenador do curso em causa.

O presente projecto contemplará a implementação de cinco casos de uso, três para cada um dos actores e dois que serão comuns a todos eles.



## Capítulo 2

# Descrição Geral

### 2.1 Perspectiva

Este trabalho irá resultar numa aplicação que possibilite aos intervenientes poderem efectuar operações sobre os momentos de avaliação da ESTIG num repositório comum que mantenha os respectivos intervenientes informados das eventuais alterações aos registos associados ao seu perfil.

#### 2.1.1 Interfaces

##### Sistema

O sistema deverá funcionar como repositório para todos os momentos de avaliação dos cursos da Estig.

##### Utilizador

A interface de utilizador deverá ser compatível com os principais browsers e respeitar as principais regras de usabilidade e acessibilidade.

##### Software

O projecto será implementado na linguagem PHP com ligação a base de dados mySQL. Serão utilizados os APIs jQuery e Smarty.

#### 2.1.2 Funções

O sistema implementado irá suportar as seguintes funções:

- Efectuar Login
- Visualizar Momentos de Avaliação
- Validar / Cancelar Validação de Momentos de Avaliação
- Inscrever / Cancelar Inscrições em Momentos de Avaliação
- Marcar / Cancelar Marcação de Novos Momentos de Avaliação

### **2.1.3 Características do Utilizador**

Os utilizadores da aplicação serão alunos, docentes e coordenadores dos cursos leccionados na ESTIG. A sua utilização esporádica e em periodos de tempo espaçados faz com que em termos de usabilidade se tenha especial atenção à facilidade de aprendizagem no uso das interfaces.

## Capítulo 3

# Diagrama de Casos de Uso

Como é visível na Figura 3.1, para este trabalho, foram definidos 4 casos de uso, sendo um para cada tipo de actores e um que, apesar de ter características específicas para cada tipo de actor é comum aos três.

Todos os tipos de utilizador podem efectuar login/logout, bem como listar as suas avaliações por datas e visualizar os detalhes das mesmas num dia escolhido.

O teacher pode também marcar ou cancelar uma avaliação de uma disciplina por ele leccionada, enquanto que o student pode inscrever-se ou cancelar a inscrição numa avaliação de uma disciplina em que esteja matriculado. Por fim o coordinator pode validar ou cancelar a validação de uma avaliação de um curso que coordene.

### 3.1 EFECTUAR LOGIN

#### 3.1.1 Descrição

Todos os utilizadores do sistema podem efectuar login ou logout do sistema.

#### 3.1.2 Actores

User

#### 3.1.3 Cenário Principal

##### Login

Figura 6.2

O Utilizador:

1. Ao aceder ao site o sistema verifica se o utilizador está logado.
2. Caso o utilizador esteja logado:

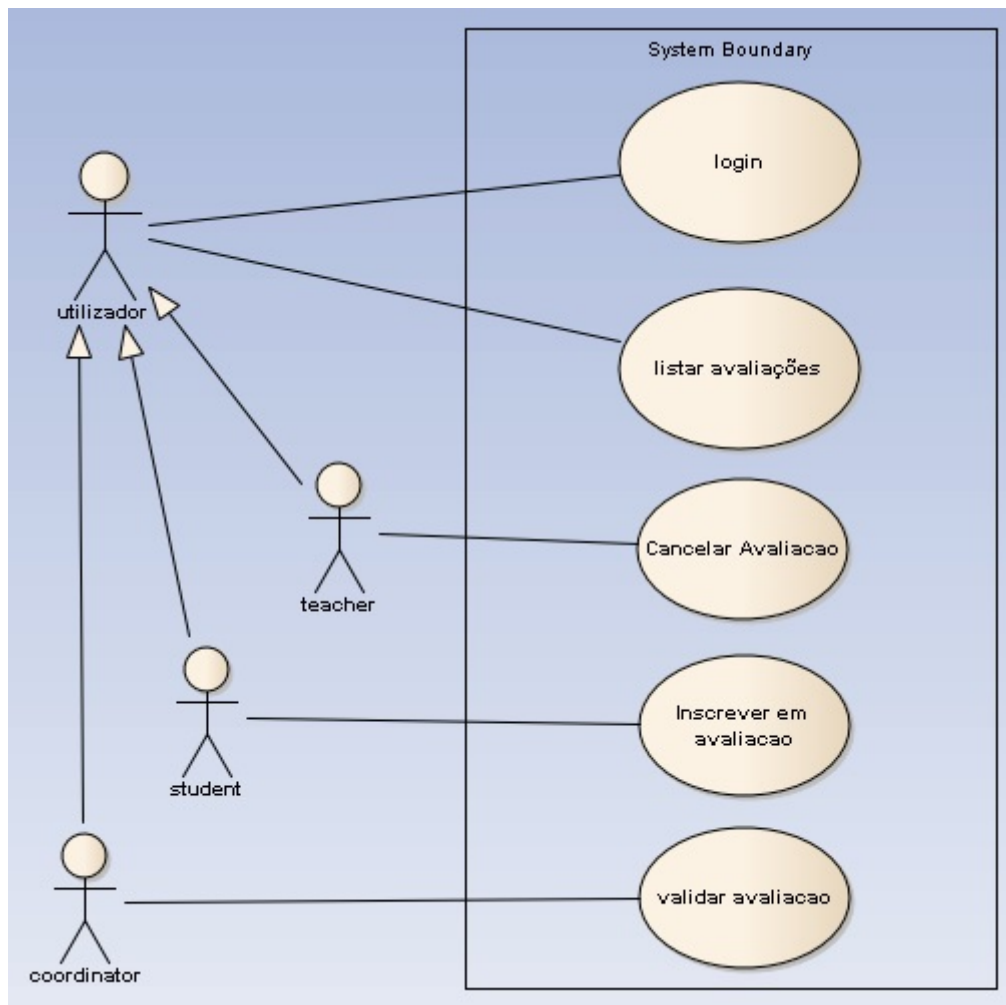


Figura 3.1: Diagrama: Casos de Uso

2.1 O sistema consulta os cursos associados ao utilizador, agrupados pelo papel desempenhado no mesmo

2.2 O sistema consulta as disciplinas associadas ao utilizador para cada papel desempenhado e curso

2.3 O sistema consulta as avaliações associadas a cada disciplina do utilizador

### 3.1.4 Extensões ou Variações

#### Logout

Figura 6.3

O Utilizador:

1. Escolhe efectuar logout.
2. O sistema limpa os dados do utilizador na sessão actual.

3. O sistema limpa os cursos associados ao utilizador como coordinator.
4. O sistema limpa os cursos associados ao utilizador como teacher.
5. O sistema limpa os cursos associados ao utilizador como student.

## **3.2 LISTAR AVALIAÇÕES**

### **3.2.1 Descrição**

A cada acesso ao sistema, o mesmo consulta as avaliações associadas ao utilizador para sempre apresentadas no calendário no intervalo de datas definido, bem como os detalhes das avaliações do dia seleccionado.

### **3.2.2 Pré-condições**

O utilizador deverá estar logado e apenas terá acesso a avaliações a ele associadas como student, teacher ou coordinator.

### **3.2.3 Actores**

User

### **3.2.4 Cenário Principal**

#### **Listar avaliações entre datas**

Figura 6.4 O user:

1. Acede ao sistema, já logado.
2. O sistema obtém a lista de cursos a ele associados, passando pelos mesmos passos já descritos no login.
3. O sistema obtém o intervalo de datas definido.
4. O sistema obtém a lista de avaliações definidas no intervalo de datas.
5. É apresentado ao utilizador a lista de avaliações no calendário.

### **3.2.5 Extensões ou Variações**

#### **Listar detalhes de avaliações**

Figura 6.5

O user:

1. Ao escolher um dia no calendário.
2. O sistema verifica o dia escolhido.
3. O sistema obtém a lista de avaliações associadas ao user para esse dia.
4. O sistema apresenta ao utilizador os detalhes das avaliações.

## **3.3 VALIDAR AVALIAÇÕES**

### **3.3.1 Descrição**

O Coordenador de Curso pode validar ou cancelar a validação das avaliações das disciplinas do curso a ele associado.

### **3.3.2 Pré-condições**

O coordenador deverá estar logado e apenas terá acesso a avaliações de disciplinas do seu curso.

### **3.3.3 Actores**

Coordenador

### **3.3.4 Cenário Principal**

#### **Validar Avaliação**

Figura 6.10 O coordenador:

1. Ao visualizar a avaliação, escolhe validar.
2. O sistema valida a avaliação.
3. O sistema actualiza o calendario
4. O sistema apresenta o calendario

### **3.3.5 Extensões ou Variações**

#### **Cancelar Validação**

Figura 6.11

O Coordenador:

1. Ao visualizar uma disciplina, escolhe cancelar a sua validação.
2. O sistema cancela a validação
3. O sistema actualiza o calendário
4. O sistema apresenta o calendário

## **3.4 MARCAR AVALIAÇÃO**

### **3.4.1 Descrição**

O docente pode criar, alterar e cancelar momentos de avaliação para as disciplinas em que seja docente.

### 3.4.2 Pré-condições

Apenas é possível criar momentos de avaliação para as disciplinas em que o docente esteja associado.

### 3.4.3 Actores

Docente

### 3.4.4 Cenário Principal

Figura 6.8

#### Novo Momento de Avaliação

O Docente:

1. Escolhe criar um novo momento de avaliação.
2. Caso tenha já preenchido o formulário:
  - 2.1 O sistema cria a nova avaliação
  - 2.2 O sistema actualiza o calendário
  - 2.3 O sistema apresenta o calendário
3. Caso não tenha preenchido o formulário:
  - 3.1 O sistema obtém a lista de disciplinas leccionadas pelo docente
  - 3.2 O sistema apresenta o formulário para que o docente crie a nova avaliação para uma das disciplinas por ele leccionadas

### 3.4.5 Extensões ou Variações

Figura 6.9

#### Cancelar Momento de Avaliação

O Docente:

1. Escolhe o dia em que está marcada a Avaliação.
2. O sistema apresenta os detalhes da avaliação.
3. O Docente escolhe cancelar avaliação.
4. O sistema cancela a avaliação.
5. O sistema actualiza o calendário
6. O sistema apresenta o calendário actualizado

## **3.5 INSCREVER EM AVALIAÇÃO**

### **3.5.1 Descrição**

O Aluno pode inscrever-se ou cancelar a inscrição nas avaliações das disciplinas a ele associado.

### **3.5.2 Pré-condições**

O aluno deverá estar logado e apenas terá acesso a avaliações das disciplinas onde está matriculado.

### **3.5.3 Actores**

Aluno

### **3.5.4 Cenário Principal**

#### **Inscriver em Avaliação**

Figura 6.6 O aluno:

1. Ao visualizar a avaliação, escolhe inscrever-se
2. O sistema inscreve o aluno
3. O sistema actualiza o calendario
4. O sistema apresenta o calendario

### **3.5.5 Extensões ou Variações**

#### **Cancelar Inscrição**

Figura 6.7

O aluno:

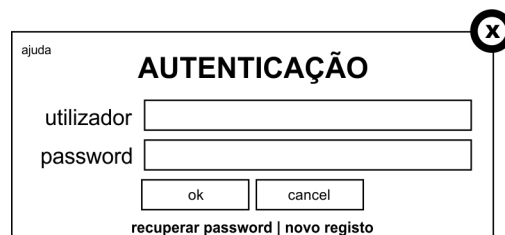
1. Ao visualizar uma disciplina, escolhe cancelar a sua inscrição
2. O sistema cancela a inscrição
3. O sistema actualiza o calendário
4. O sistema apresenta o calendário



## Capítulo 4

# Modelação de Interfaces

### 4.1 Efectuar Login



The diagram shows a login window titled "AUTENTICAÇÃO". In the top-left corner, there is a link labeled "ajuda". In the top-right corner, there is a close button represented by a circle with an "x". The main area contains two labels, "utilizador" and "password", each followed by a text input field. Below these fields are two buttons labeled "ok" and "cancel". At the bottom of the window, there is a link labeled "recuperar password | novo registo".

Figura 4.1: Interface: Login

## 4.2 Inscrever em Momento de Avaliação

[ajuda](#)[logout](#)

# GESTÃO DE AVALIAÇÕES

JOÃO PACIÊNCIA  
Coordenador de Curso  
Engenharia Informática

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

16/09/2013

PCR

teste

peso da avaliação

50

sala(s)

cancelar inscrição

16/09/2013

HA

exame

peso da avaliação

90

sala(s)

inscrever

Figura 4.2: Interface: Efectuar ou Cancelar Inscrição em Momento de Avaliação

## 4.3 Marcar e Cancelar Momento de Avaliação

[ajuda](#)[logout](#)

# GESTÃO DE AVALIAÇÕES

JOÃO PACIÊNCIA  
Coordenador de Curso  
Engenharia Informática

**INSERIR NOVA AVALIAÇÃO**

Tipo de avaliação:

Data: 15/10/2013

Peso:

Sala:

Observações:

Disciplina:

16/09/2013  
**PCR**  
teste

peso da avaliação

sala(s)

50

16/09/2013  
**HA**  
exame

peso da avaliação

sala(s)

90

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Figura 4.3: Interface: Marcar Novo Momento de Avaliação

#### 4.4 Validar e Cancelar Validação de Momentos de Avaliação

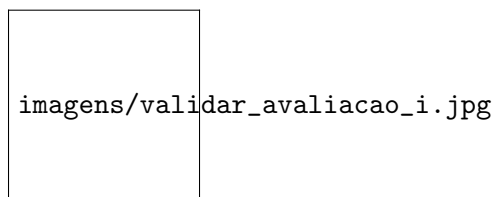


Figura 4.4: Interface: Validar ou Cancelar Validação de Momento de Avaliação

## **4.5 Listar Momentos de Avaliação**

## Capítulo 5

# Modelação da Base de Dados

Resumidamente a base de dados está representada da seguinte forma:

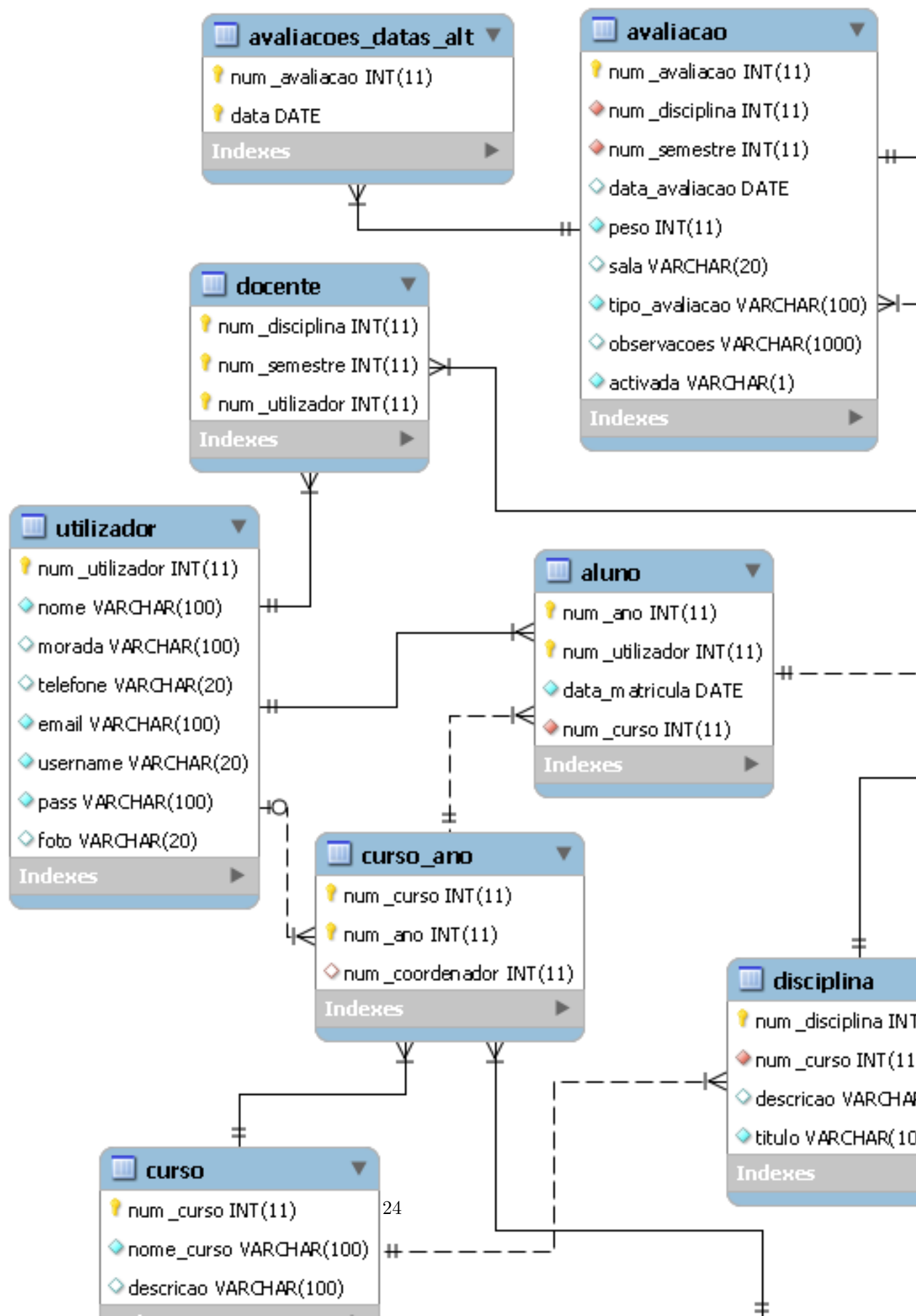
### TABELAS

- **curso** - Representa os cursos leccionados
- **ano\_lectivo** - Representa um ano lectivo
- **utilizador** - todos os utilizadores ficam registados nesta tabela
- **curso\_ano** - Representa um curso leccionado num determinado ano, nesta tabela fica representado o coordenador do curso nesse ano associando o id do respectivo user na tabela utilizador
- **semestre** - Representa um semestre de um determinado ano lectivo
- **disciplina** - Representa uma disciplina
- **disciplina\_semestre** - Faz referencia a uma disciplina leccionada num determinado semestre
- **Docente** - Representa um ou mais utilizadores designados como docentes de uma determinada disciplina
- **aluno** - Representa uma matrícula de um utilizador como aluno num determinado curso
- **matricula\_disciplina** - Representa a matricula numa disciplina de um determinado aluno
- **avaliacao** - Representa uma avaliacao marcada para uma disciplina leccionada num determinado semestre
- **avaliacao\_datas\_alt** - Nesta tabela ficam registadas as datas alternativas escolhidas pelo docente para que o coordenador possa ter a possibilidade de trocar caso aconteça um peso demasiado excessivo nas avaliações para os alunos.

- **avaliacao\_aluno** - Representa a inscrição de um aluno numa avaliação e é onde fica registada a sua nota

## **STORED PROCEDURES**

- **get\_coordinator\_evaluations( user\_num, num\_discipline )** - Obter lista de avaliações com o perfil de coordenador de um determinado utilizador
- **get\_teacher\_evaluations( user\_num, num\_discipline )** - Obter lista de avaliações com o perfil de docente de um determinado utilizador
- **get\_student\_evaluations( user\_num, num\_discipline )** - Obter lista de avaliações com o perfil de aluno de um determinado utilizador
- **get\_cursos\_user( user\_num )** - Obter lista de cursos a que utilizador esteja associado
- **get\_coordinator\_disciplines( user\_num, num\_course )** - Obter lista de disciplinas com o perfil de coordenador de um determinado utilizador
- **get\_teacher\_disciplines( user\_num, num\_course )** - Obter lista de disciplinas com o perfil de docente de um determinado utilizador
- **get\_student\_disciplines( user\_num, num\_course )** - Obter lista de disciplinas com o perfil de aluno de um determinado utilizador
- **add\_evaluation( user\_num, discipline\_id, date, weight, classroom, type, observations )** - Adicionar uma avaliação a uma determinada disciplina





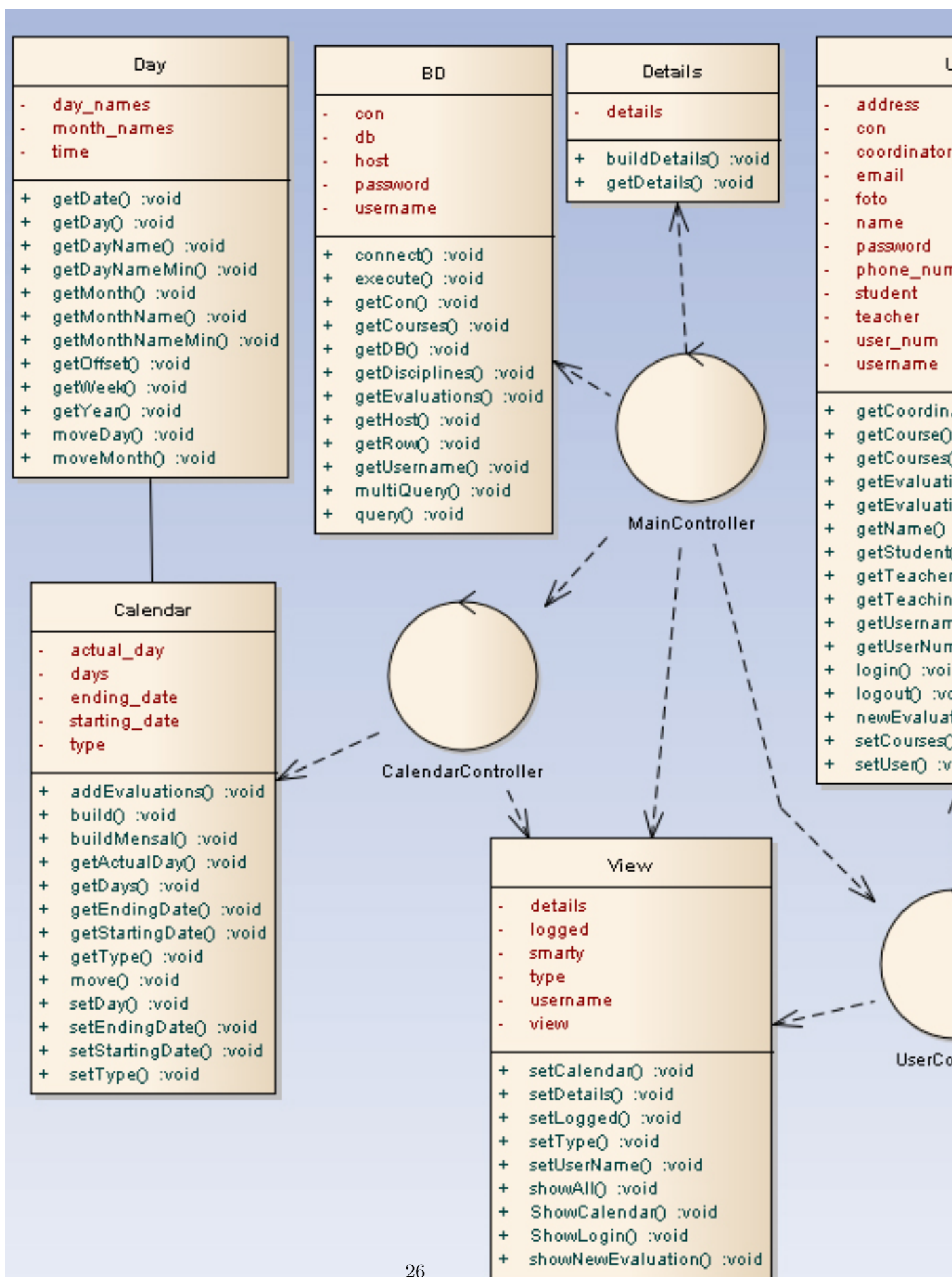
## **Capítulo 6**

# **Modelação UML**

### **6.1 Diagrama de Classes**

### **6.2 Diagramas de Sequência**

#### **6.2.1 Efectuar Login**



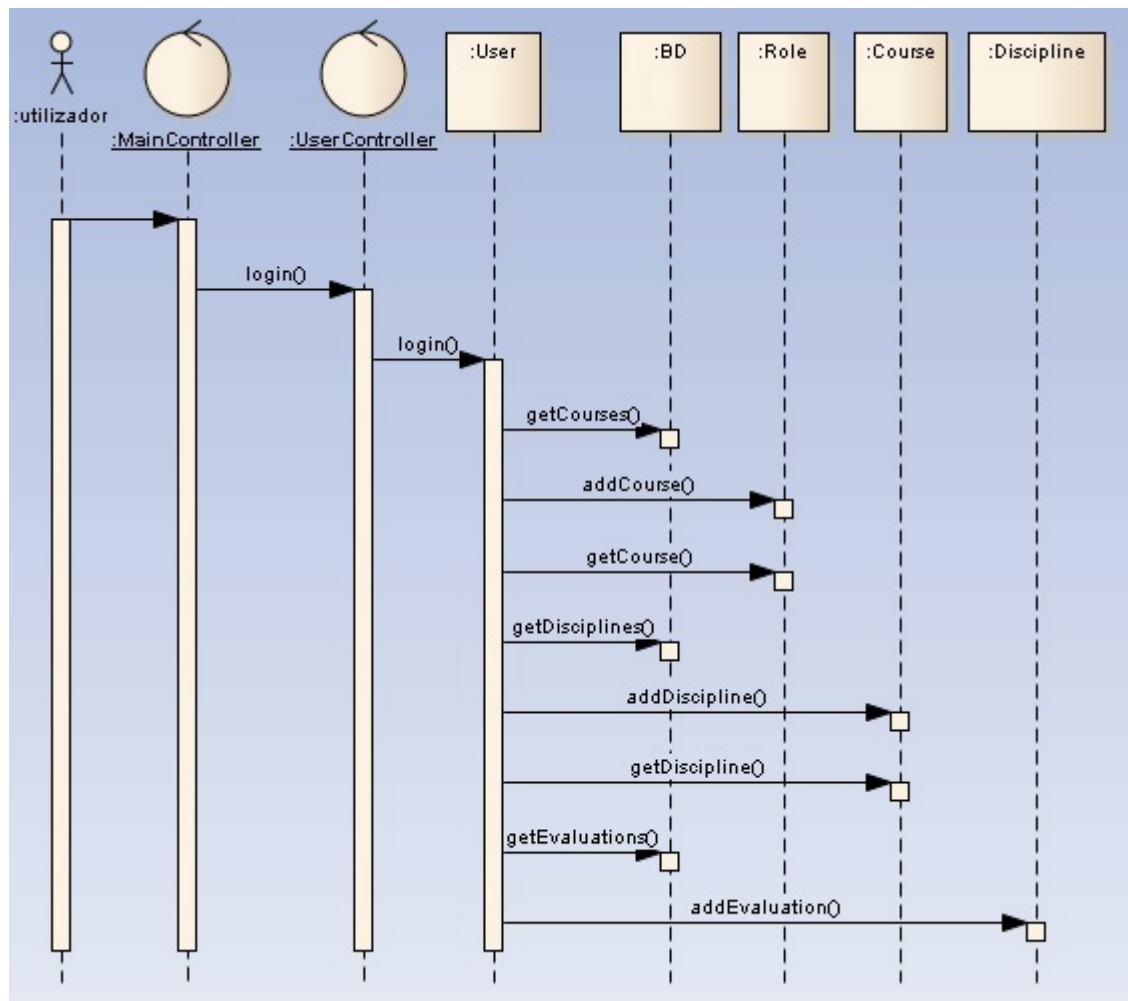


Figura 6.2: Diagrama de Sequência: Login

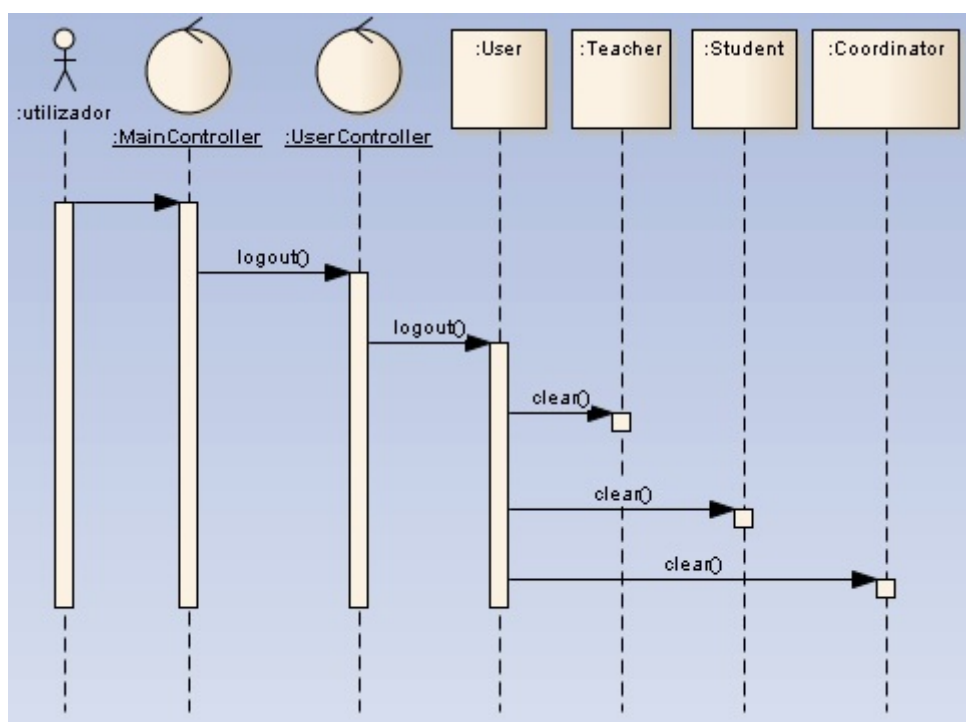


Figura 6.3: Diagrama de Sequência: Logout

## 6.2.2 Listar Avaliações

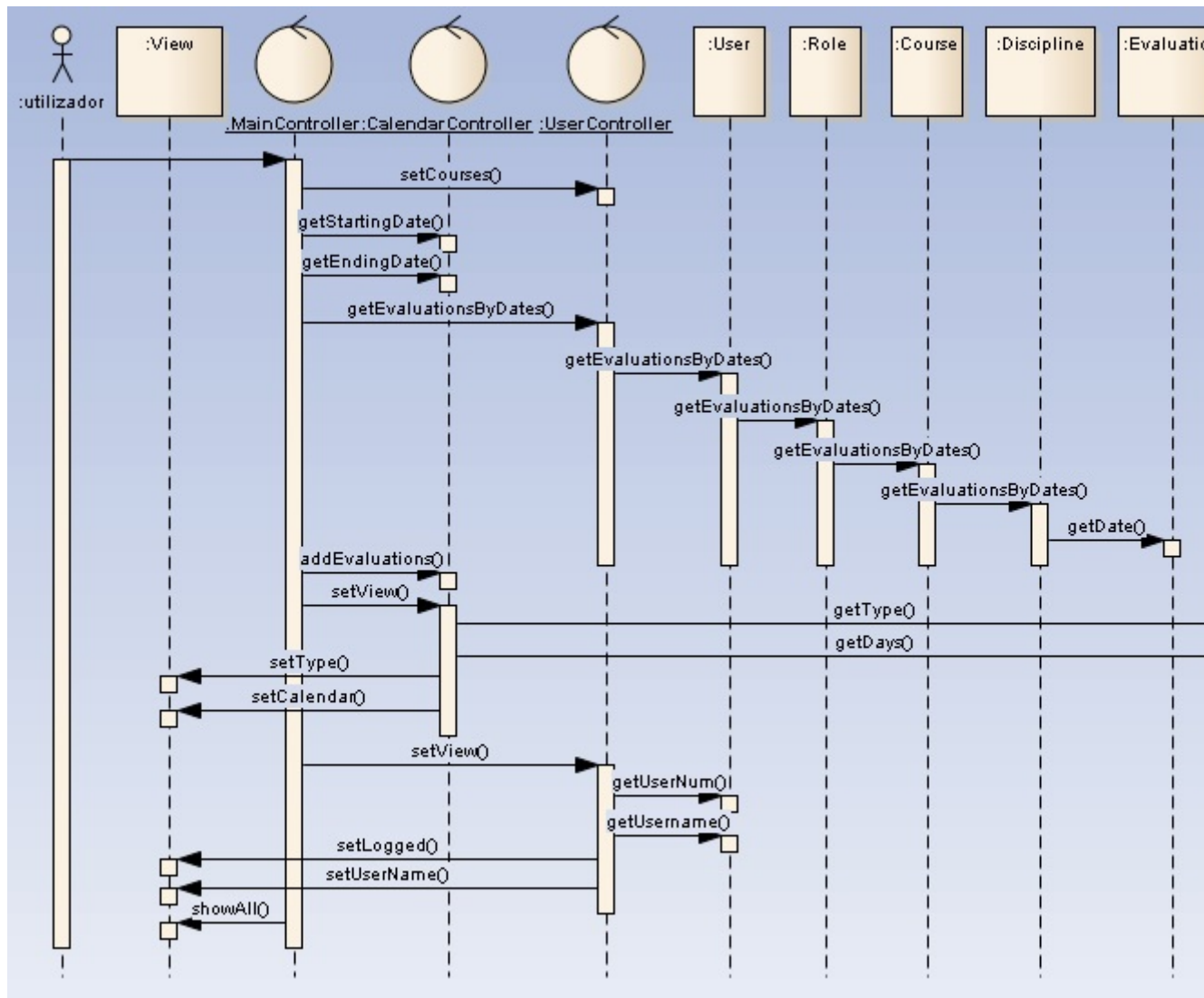


Figura 6.4: Diagrama de Sequência: Listar Avaliações Entre Datas

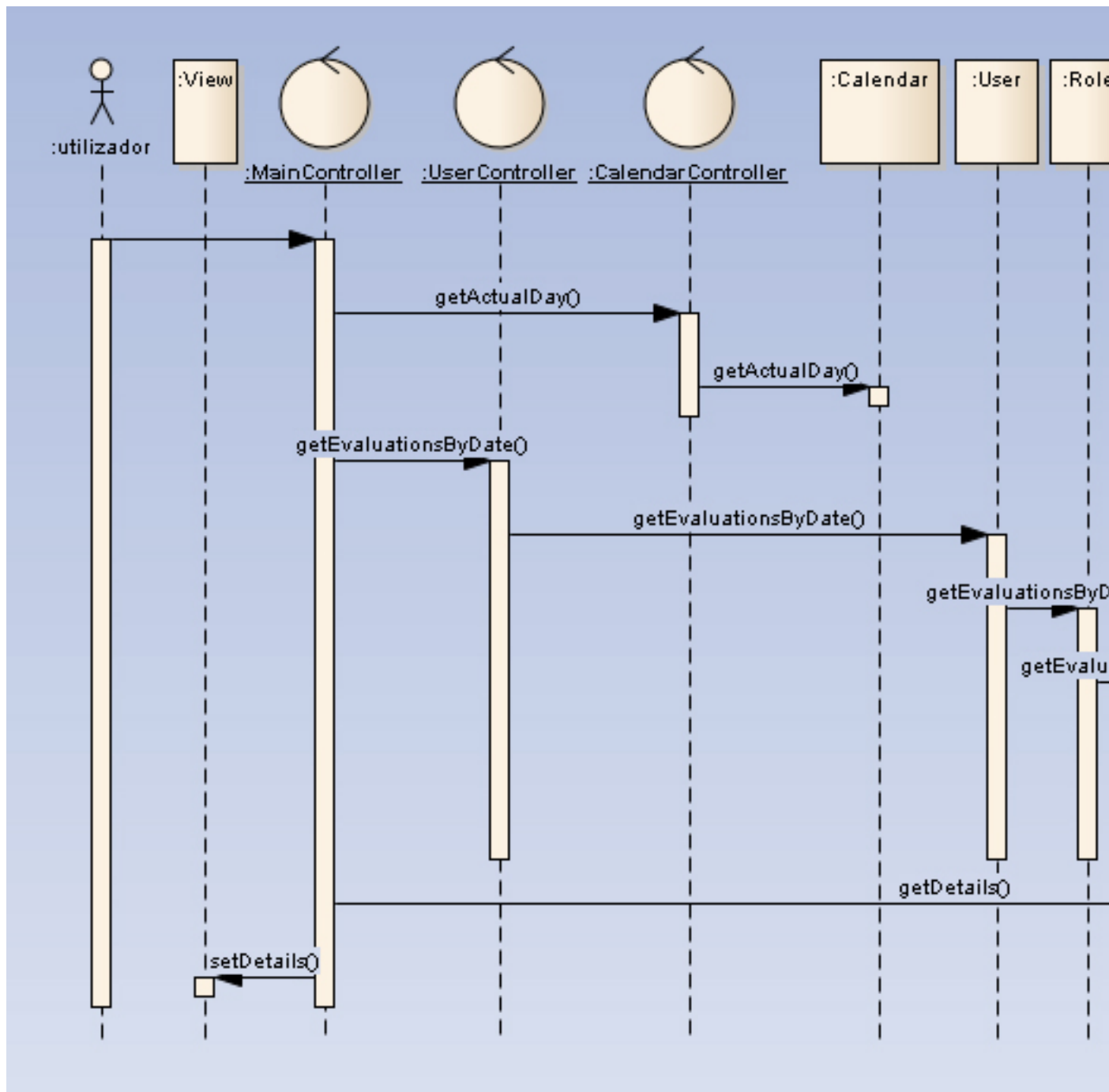


Figura 6.5: Diagrama de Sequência: Listar Detalhes de Avaliação

### 6.2.3 Inscrição em Avaliação

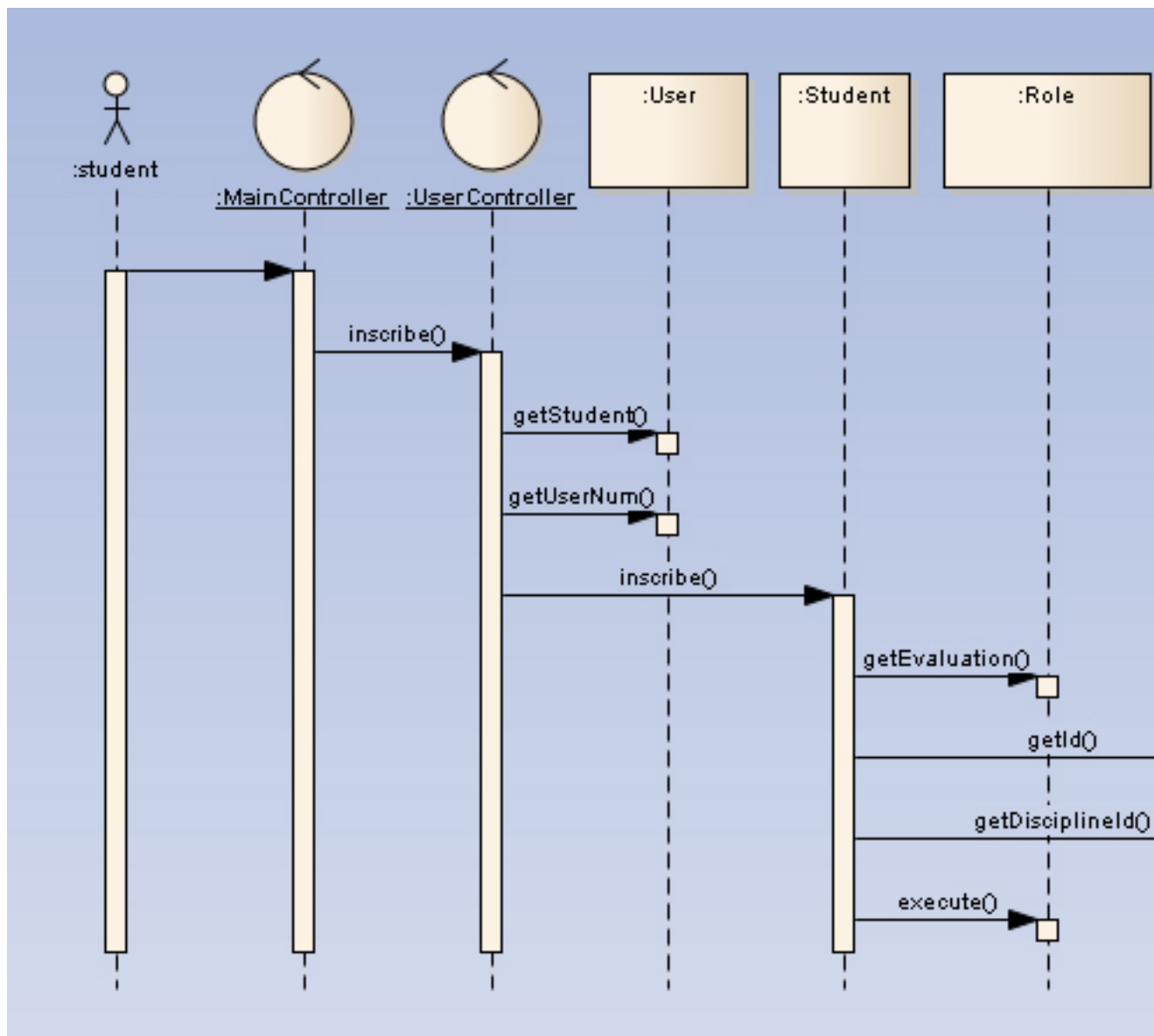


Figura 6.6: Diagrama de Sequência: Efectuar Inscrição em Momento de Avaliação

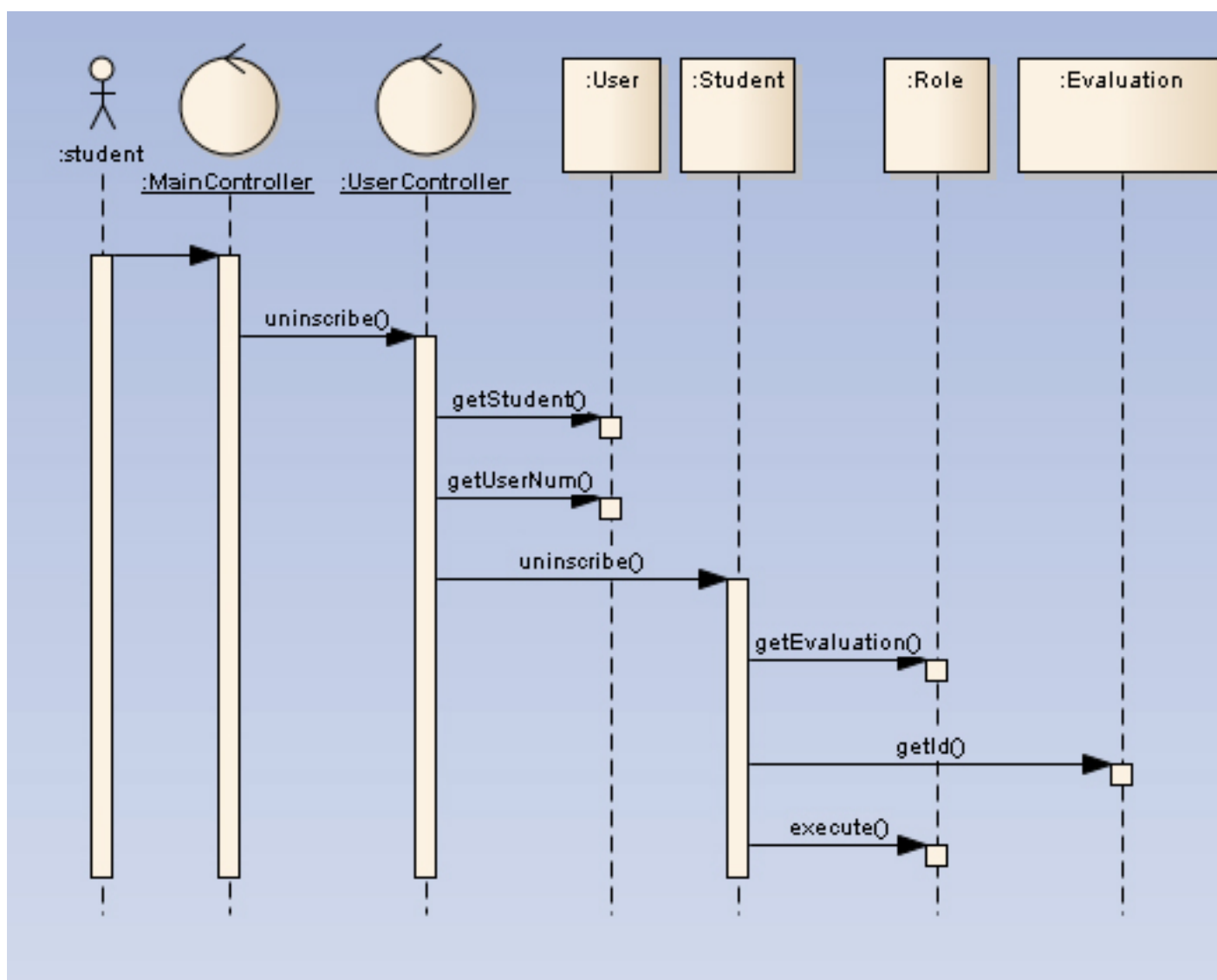


Figura 6.7: Diagrama de Sequência: Cancelar Inscrição de Momento de Avaliação



## 6.2.4 Marcar Avaliação

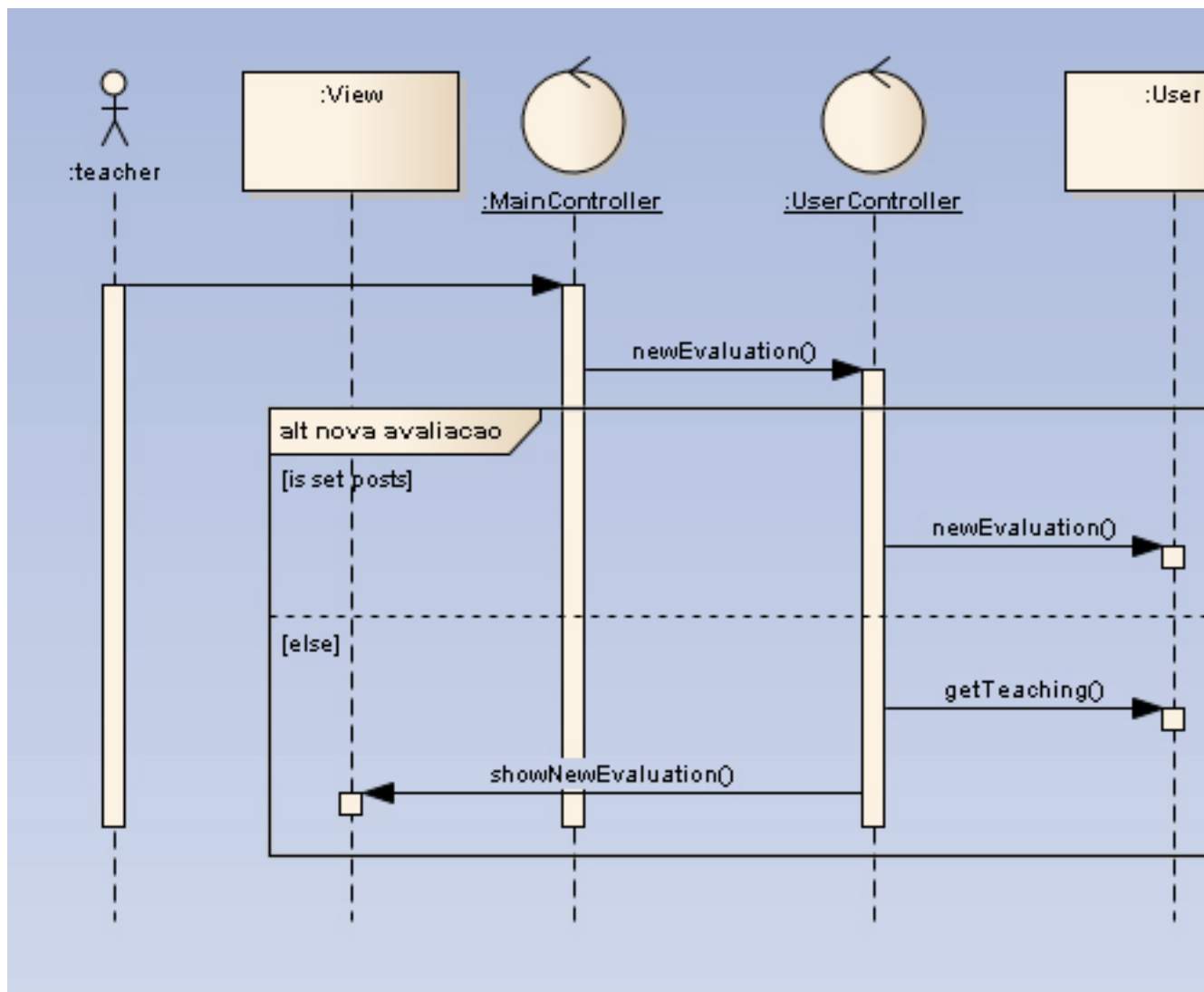


Figura 6.8: Diagrama de Sequência: Novo Momento de Avaliação

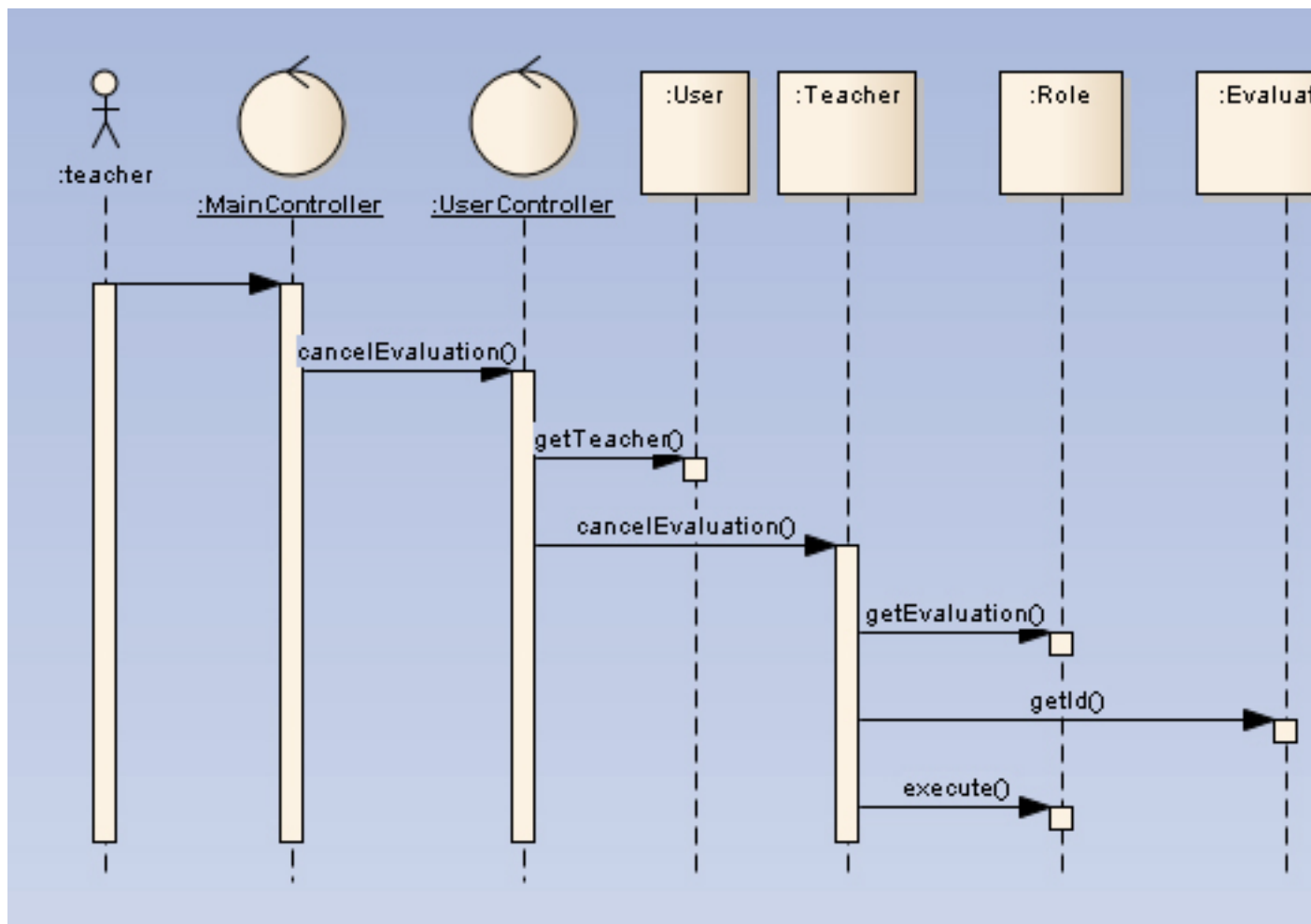


Figura 6.9: Diagrama de Sequência: Cancelar Momento de Avaliação

### 6.2.5 Validar Avaliações

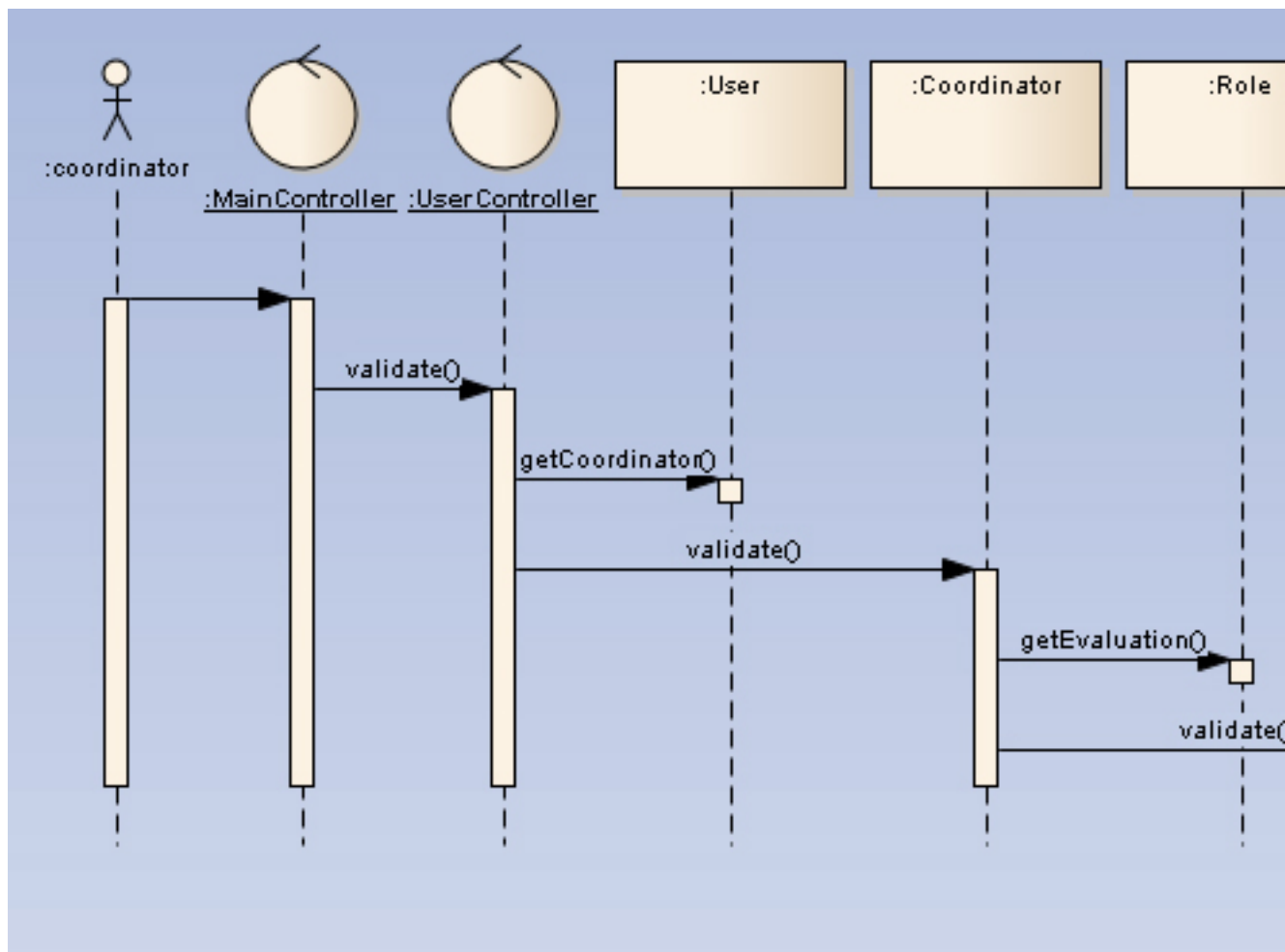


Figura 6.10: Diagrama de Sequência: Validar Avaliação

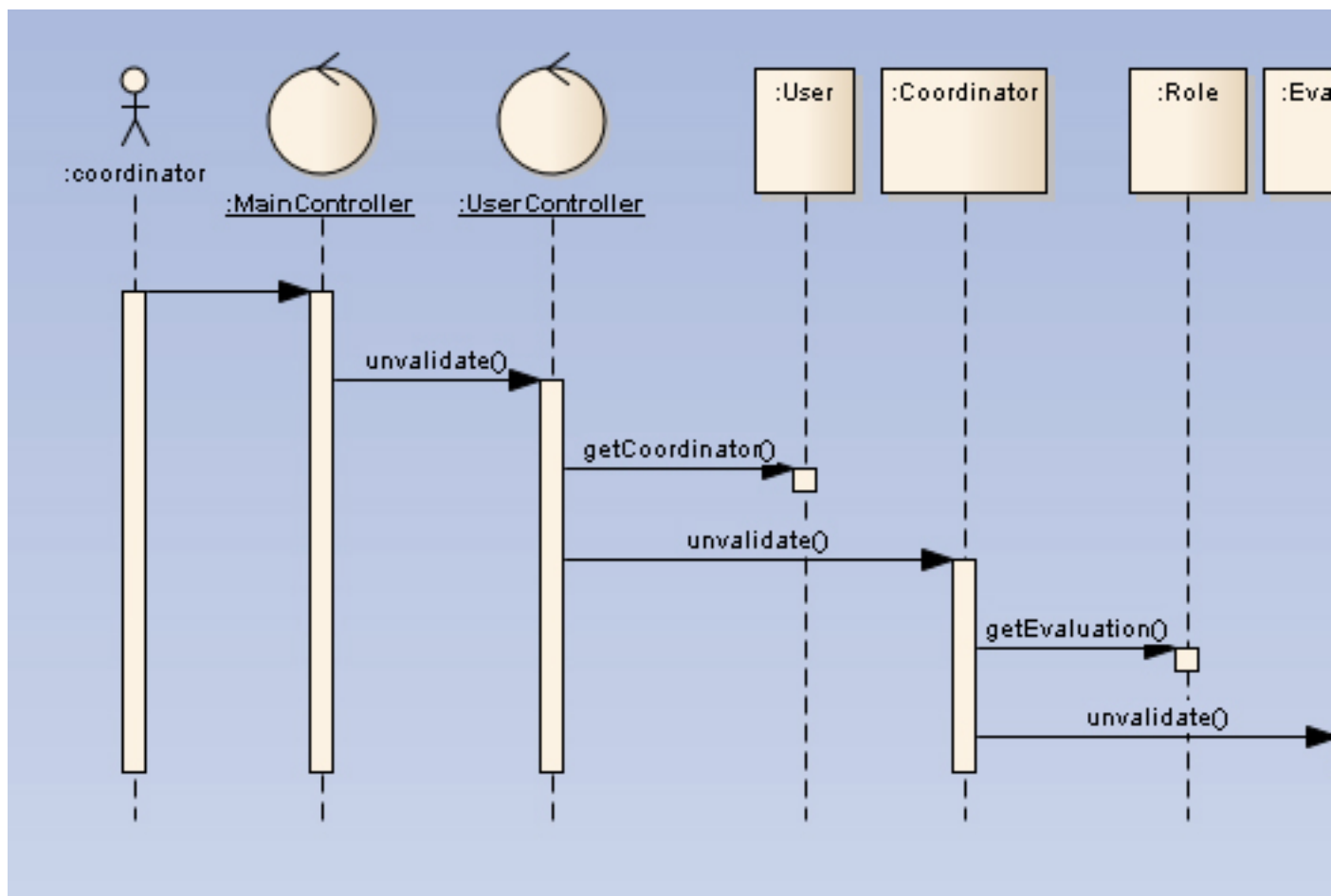


Figura 6.11: Diagrama de Sequência: Cancelar Validação de Momento de Avaliação

## Capítulo 7

# Implementação

### 7.1 Introdução

Enquadramento geral do projecto, objectivos a atingir e identificação das tarefas que foram objecto de implementação.

### 7.2 Decisões Globais de Implementação

#### 7.2.1 Tecnologias Utilizadas

Para este projecto foi escolhida a linguagem de programação PHP com ligação a base de dados MySQL, é utilizado o princípio de programação orientada a objectos com um modelo de arquitectura Model-View-Controller (MVC).

Para a ligação entre o PHP e o mySQL é utilizada a extensão MySQLi. Para a criação das páginas a apresentar ao utilizador foi implementado o sistema de gestão de templates para PHP - Smarty (versão 3.1). E também o API de Javascript - jQuery (versão jquery-1.3.1).

#### 7.2.2 Armazenamento dos Dados

Os dados utilizados pela aplicação são armazenados na base de dados local (mySQL) e quando um utilizador acede ao site é criada uma sessão em PHP que guarda, temporariamente, os dados necessários para manter o utilizador logado.

#### 7.2.3 Hierarquia de Ficheiros

Os ficheiros com os controladores estão colocados na directoria “*controller*”, os do modelo na directoria “*model*”, da vista na directoria “*view*”. As folhas de estilo (“CSS”) dentro da directoria “*css*”, os elementos gráficos são colocados dentro de “*imagens*”. Todos os ficheiros com as bibliotecas javascript ou funções criadas para este site ficam dentro da directoria “*js*”. Por fim os ficheiros utilizados pelo Smarty para apresentação de html ficam dentro da directoria

*“templates”*.

As classes utilizadas no site estão fisicamente localizadas em ficheiros de extensão PHP com o mesmo nome da respectiva classe.

## 7.3 Decisões de Implementação Específicas

### 7.3.1 Ligação à Base de Dados

O controlador principal do site está no ficheiro `MainController.php`, no seu construtor é criado um objecto que servirá de ligação à base de dados. esse objecto é da classe de modelo BD. Esta classe aceita no seu construtor o endereço do servidor onde a base de dados está armazenada, o nome de utilizador, password e o nome da base de dados. Com esses dados o sistema efectua a ligação.

Quando a execução do código termina, esta classe fecha a conexão com a base de dados através do seu método `__destruct`.

A classe para ligação à base de dados contém vários métodos dos quais se podem aqui destacar o método `multiQuery` que permite obter resultados que uma chamada de uma stored procedure que retorne mais que uma linha. Para consultas “normais” existem os métodos `query` que retorna os resultados num array, `execute` para executar queries que não retornem resultados e o `getRow` que apenas retorna uma linha.

### 7.3.2 Funcionamento do controlador principal

O controlador principal `MainController` começa por criar uma sessão PHP no servidor, criar os objectos para a ligação à base de dados, vista e dois objectos para controladores “secundários”: `UserController` e `CalendarController`.

De seguida executa a acção pretendida pelo utilizador através do método `actions`, este método verifica no array `$_GET` as variáveis recebidas sendo que o primeiro será a classe a utilizar e os seguintes serão os métodos com o respectivo parâmetro.

Por exemplo, para um acesso com `?Calendar&setDay=20131023`, é executado o método `setDay(20131023)` da classe `CalendarController`.

### 7.3.3 Login

Sempre que o utilizador acede ao site é criado um objecto da classe `UserController` que recebe no seu construtor o objectos da ligação à base de dados e da vista. É então criado o `user` da classe de modelo `User` e executado o método `login`. Este método verifica se foi submetido o formulário de login, sendo que o tenha sido executa o método `login` do `user` com o username e password recebidos como parâmetros, caso não tenha sido submetido o formulário o método é chamado sem parâmetros.

Os objectos da classe `User` contêm três objectos das classes de modelo `Coordinator`, `Teacher` e `Student`, o seu construtor executa também o método da própria classe `login`. Este método, verifica se recebe parâmetros, caso não tenha recebido, executa o método `setUser`, caso tenha recebido, executa um query para confirmar a existência de um utilizador com os dados recolhidos.

Após estar definido, ou não, o utilizador é executado o método `setCourses`.

Este método começa por chamar o método `getCourses` do objecto da classe `BD` que por sua vez executa a stored procedure `get_cursos_user` que retorna a lista de cursos a que o utilizador está associado, para cada um deles será adicionado ao objecto correspondente ao papel do utilizador em questão o respectivo curso.

Para cada curso é obtida a lista de disciplinas que está associada ao utilizador com o papel do mesmo. A partir da lista obtida é adicionada ao curso a mesma.

Por fim, para cada disciplina obtida são consultadas as avaliações da mesma, associadas ao utilizador e são adicionadas ao objecto da `Disciplina` criado.

#### **7.3.4 Logout**

Quando um utilizador efectua o logout, todas as variáveis do objecto da classe modelo `User` e executa o método `clear` de cada uma das classes dos papéis possíveis. Estas classes herdam da classe `Role` que contem os métodos comuns a todas. Este método esvazia o array com os cursos.

#### **7.3.5 Listar Avaliações**

O sistema após ter feito as tarefas de verificar que utilizador está logado e quais os cursos, disciplinas e avaliações a ele estão associados, executa as acções solicitadas pelo método `GET`. Terminadas estas tarefas o `MainController` executa de novo o método `setCourses`.

Após isso solicita ao controlador `CalendarController` quais as datas inicial e final definidas e solicita ao `UserController` as avaliações contidas nesse intervalo de datas.

#### **7.3.6 Ver Detalhes de Avaliação**

#### **7.3.7 Marcar Avaliação**

#### **7.3.8 Cancelar Avaliação**

#### **7.3.9 Validar Avaliação**

#### **7.3.10 Cancelar Validação de Avaliação**

#### **7.3.11 Inscrever em Avaliação**

#### **7.3.12 Cancelar Inscrição em Avaliação**

## Capítulo 8

# Conclusão

No trabalho efectuado sobre o tema em na disciplina de Bases de Dados 2, optámos por uma forte política de segurança ao nível da base de dados. Visto que o trabalho actual irá ser implementado em mySQL ao invés de T-SQL, poderão algumas implementações não ser compatíveis com o novo tipo de base de dados. Nomeadamente a criação de um login no servidor e acesso à bd limitando o acesso aos dados de acordo com o perfil. Limitando o acesso directo às tabelas, optando por execução de stored procedures e algumas funcionalidades dos triggers como por exemplo o envio de emails.

Nos casos em que não for possível implementar algumas destas funcionalidades ao nível da base de dados, terão que ser implementadas pelo PHP ou C# nas respectivas classes.

É possível também que durante a implementação, dadas estas restrições, se chegue à conclusão que é necessário a criação de mais classes representando algumas tabelas da base de dados.

No geral com será possível o coordenador de um curso validar, cancelar ou alterar a data de uma avaliação. O docente tem a possibilidade de criar, cancelar ou alterar um momento de avaliação. E o aluno pode Inscrever-se ou cancelar a inscrição numa avaliação.

Todos os utilizadores poderão ver os momentos de avaliação relacionados com o seu perfil e, também, alterar os seus dados pessoais, ou recuperar os dados de acesso ao sistema.

É possível um novo utilizador criar um novo registo, no entanto não terá acesso a nenhuma avaliação, visto não estar prevista a implementação da gestão de perfis de utilizadores. Tal funcionalidade terá de ficar para trabalho futuro.