



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Projeto Facultativo de Desenvolvimento de  
Sistemas de Software

Catarina Machado (a81047)

6 de Outubro de 2018

## **Resumo**

O presente relatório descreve de forma sucinta o projeto facultativo realizado no âmbito da UC de Desenvolvimento de Sistemas de Software, no terceiro ano do Mestrado Integrado em Engenharia Informática, na Universidade do Minho. O objetivo do projeto foi desenvolver um sistema capaz de gerir as quotas dos sócios do centro de estudantes do curso (CeSIUM).

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Organização dos Dados- Model</b>	<b>3</b>
2.1	Classe Sócio . . . . .	3
2.2	Classe Quota . . . . .	3
2.3	Classe GestaoSocios . . . . .	3
2.4	Classe SocioNullException . . . . .	3
<b>3</b>	<b>Interface Gráfica</b>	<b>4</b>
<b>4</b>	<b>Controller</b>	<b>4</b>
<b>5</b>	<b>Métodos Essenciais</b>	<b>4</b>
<b>6</b>	<b>Manual de Utilização</b>	<b>4</b>
<b>7</b>	<b>Conclusões</b>	<b>8</b>

# 1 Introdução

O relatório descreve a forma como os dados foram organizados, as classes que foram utilizadas e inclui também um manual de utilização para a aplicação.

Para a realização deste projeto segui o modelo MVC (Model-view-controller). O trabalho foi realizado utilizando o IntelliJ IDEA.

O objetivo da aplicação é que seja possível visualizar uma lista com todos os sócios do centro de estudantes bem como visualizar os detalhes de um determinado sócio (dados pessoais e quotas pagas). Também será possível adicionar novos sócios ao sistema e registar o pagamento de novas quotas por parte dos sócios.

Na Secção 2 encontram-se as classes que fazem parte do "Model", na secção 3 estão enumeradas as classes que constituem a "View" do programa e na secção 4 encontra-se uma breve descrição da classe que constitui o "Controller". Na secção 5 são apresentados os métodos essenciais para o funcionamento do programa e na secção 6 está um manual de utilizador para a plataforma.

## 2 Organização dos Dados- Model

Para desenvolvermos este trabalho adotamos as seguintes classes:

### 2.1 Classe Sócio

```
private int numero;  
private String nome;  
private String curso;  
private int ano;  
private String morada;  
private List<Quota> quotas;
```

Esta classe armazena toda a informação necessária relativa a um sócio, nomeadamente o seu **número** de sócio, **nome**, **curso**, **ano**, **morada** e uma **lista com todas as quotas** já pagas por ele.

### 2.2 Classe Quota

```
private double valor;  
private LocalDate data;
```

A classe Quota armazena a informação relativa ao pagamento de uma quota, nomeadamente o **valor** e a **data** em que foi realizada.

### 2.3 Classe GestaoSocios

```
private Map<Integer, Socio> socios;
```

Esta classe é a classe "principal" do sistema uma vez que guarda a **informação de todos os sócios**. Esta informação é guardada num map, onde a chave é o número de sócio.

### 2.4 Classe SocioNullException

Classe de exceção- quando o sócio a que tentamos aceder não existe (null).

### 3 Interface Gráfica

Para a camada de apresentação do sistema, foram necessárias 6 classes:

- *InitialView* - painel inicial da aplicação, com os botões para cada uma das funcionalidades principais do sistema.
- *VisualizarSociosView* - encarregue de mostrar uma tabela com todos os sócios.
- *SocioDetailsView* - apresenta os detalhes relativos a um determinado sócio.
- *QuotasPagasView* - responsável por mostrar ao utilizador uma tabela com todas as quotas pagas por ele.
- *NovoSocioView* - formulário para a criação de um novo sócio.
- *PagarQuotaView* - responsável por mostrar um formulário que permite que um sócio pague uma quota de um determinado valor.

### 4 Controller

A classe "GestaoSociosController", está responsável por coordenar a troca de informação entre a interface gráfica e a camada lógica da aplicação, por exemplo, preenche as tabelas da camada de apresentação com os dados da camada lógica, "pega" nos dados introduzidos pelo utilizador aquando da criação de um novo sócio e chama os métodos necessários para que essa informação seja devidamente inserida no sistema, entre outros.

Esta classe também alimenta a aplicação com três sócios, meramente ilustrativos.

### 5 Métodos Essenciais

Para a concretização do problema foram criados alguns métodos com o intuito de responder às diferentes necessidades do sistema:

1. Visualizar sócios (Figura 2)- método **getSocios**, da classe GestaoSocios.
2. Visualizar detalhes de um sócio (Figura 3)- método **getSocio**, da classe GestaoSocios.
3. Alterar informações de um sócio (Figura 3)- método **alterarInfoSocio**, da classe GestaoSocios.
4. Visualizar as quotas pagas por um determinado sócio (Figura 3)- método **getQuotas**, da classe Sócio.
5. Adicionar um novo sócio (Figura 5)- método **addSocio**, da classe GestaoSocios.
6. Pagamento de uma nova quota (Figura 6)- método **pagarQuota**, da classe GestaoSocios, que utiliza o método addQuota da classe Sócio.

### 6 Manual de Utilização

Quando a aplicação é aberta, o menu que é apresentado é o seguinte:

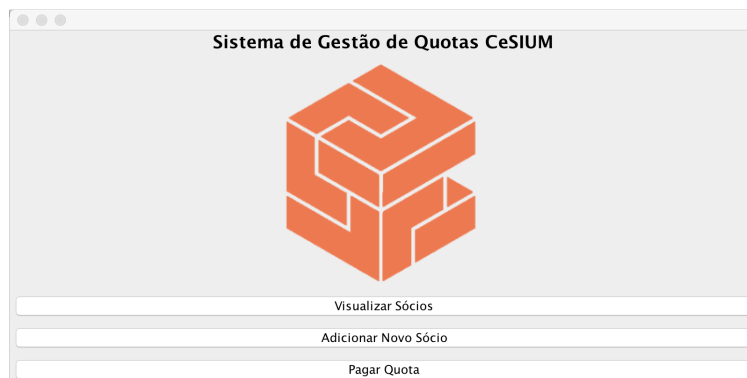


Figura 1: Menu inicial da aplicação.

Carregando no botão "Visualizar sócios", é apresentada uma tabela com as informações de todos os sócios, tal como se pode ver em seguida:

Número	Nome	Curso	Ano	Morada
1776	Catarina	MIEI	3	Braga
1970	Eva	MIEI	2	Braga
1890	Lucas	MIEI	4	Braga

Figura 2: Visualizar todos os sócios do CeSIUM.

Carregando num dos sócios apresentado, é aberto um novo menu. Por exemplo, carregando no sócio "Catarina" é aberto o seguinte painel:

Figura 3: Detalhes de um sócio.

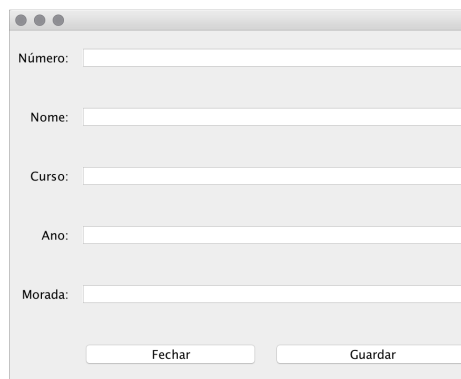
Aqui, o utilizador pode alterar as informações deste sócio. Tem também a possibilidade de ver as quotas pagas por este sócio. Carregando então no botão "Ver Quotas Pagas" aparece o seguinte menu:



Data	Valor
2018-10-06	10.0
2018-10-06	15.0

Figura 4: Visualizar quotas pagas por um determinado sócio.

Olhando novamente para a Figura 1, e escolhendo agora o botão "Adicionar Novo Sócio", é apresentado ao utilizador o formulário em seguida:



Número:

Nome:

Curso:

Ano:

Morada:

Figura 5: Adicionar um novo sócio.

Por último, pressionando o último botão do menu inicial (Figura 1) o utilizador tem ainda a possibilidade de registar o pagamento de uma nova quota. Este é o painel apresentado caso esta opção seja seleccionada:



Pagar Quota

Número de Sócio:

Valor:

Figura 6: Pagamento de uma nova quota.

Para além destes menus, é também de salientar que existe feedback por parte da aplicação quando existe input por parte do utilizador, fazendo assim com que a UX seja melhorada.

Por exemplo, caso seja adicionado com sucesso um novo sócio ao sistema aparece a seguinte mensagem no ecrã:

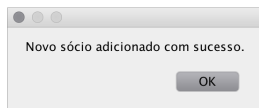


Figura 7: Feedback positivo novo sócio.

Caso haja insucesso nesta operação, por exemplo, se não foram inseridos apenas caracteres numéricos no ano e no número de sócio aparece a seguinte janela.

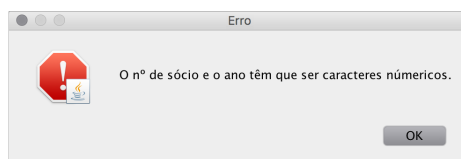


Figura 8: Feedback negativo novo sócio.

Caso uma nova quota seja efetuada com sucesso, ou seja, o valor inserido seja um caracter numérico e o número de sócio seja um número que já existe é apresentada a seguinte mensagem:

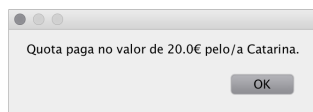


Figura 9: Feedback pagamento de uma quota.

Caso, por exemplo, o número de sócio inserido não corresponda a nenhum sócio no sistema é apresentada a seguinte janela:

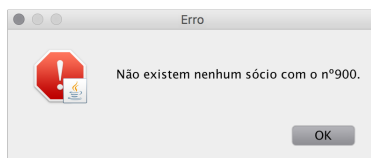


Figura 10: Feedback negativo nova quota.

Quando os dados de um sócio são alterados, aparece a seguinte mensagem:

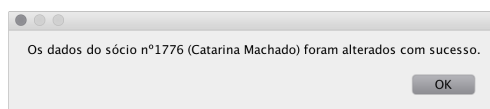


Figura 11: Feedback Alterar Dados.

Esses são apenas alguns exemplos de feedback que a aplicação fornece.



## 7 Conclusões

A realização deste mini projeto foi bastante positiva nesta altura do semestre uma vez que me deu a oportunidade de relembrar alguns conceitos de Java e de praticar um pouco mais esta linguagem de programação. Ajudou ainda a praticar o padrão de arquitetura de software MVC.

Gostei do resultado final da aplicação no que diz respeito à UI e também à forma como as classes se encontram implementadas. No entanto, gostava de ter povoado a aplicação com mais funcionalidades, por exemplo, haver um menu com uma tabela com a informação de todas as quotas pagas por todos os sócios, haver uma notificação dos sócios que têm quotas em atraso ou até dar mais informações estatísticas como por exemplo o número total de sócios do centro de estudantes.