



Escola de Engenharia  
**Universidade do Minho**

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA  
**Mestrado Integrado em Engenharia Informática**  
*Desenvolvimento Sistemas de Software*

## System Cost – Partilha de Despesas num Apartamento

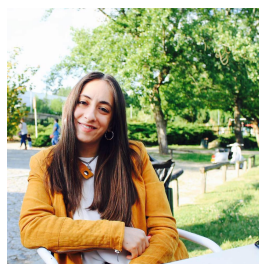
### 1ª Fase



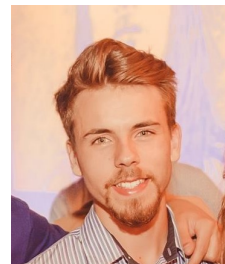
Célia Fi-  
gueiredo  
a67637



Gil  
Gonçalves  
a67738



Márcia  
Costa  
a67672



Daniel  
Rodrigues  
a67634



Ricardo  
Lopes  
a72062

Braga, 9 de Novembro de 2016

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Apresentação do Caso de Estudo . . . . .	2
<b>2</b>	<b>Requisitos do Sistema</b>	<b>4</b>
2.1	Análise e Levantamento dos Requisitos . . . . .	4
2.1.1	Criação do grupo com os elementos da casa/apartamento . . . . .	4
2.1.2	Gestão das contas . . . . .	4
2.2	Base de dados . . . . .	4
<b>3</b>	<b>Arquitetura da Aplicação</b>	<b>5</b>
3.1	Modelo de Dominio . . . . .	5
3.2	Modelo de Use Case . . . . .	6
3.2.1	Diagrama de Use Case . . . . .	6
3.2.2	Subdiagramas . . . . .	6
3.3	Mockups . . . . .	7
<b>4</b>	<b>Conclusão</b>	<b>10</b>

# 1. Introdução

No âmbito da unidade curricular de Desenvolvimento de Sistemas de Software do 3º ano do curso de MIEI, foi proposto o desenvolvimento de um projeto que visa a concepção de um sistema que serve de suporte à partilha de despesas num apartamento.

Neste relatório é descrito o processo de análise, modelação e conceção de um sistema que serve de suporte à partilha de despesas num apartamento, foi-nos proposto desenvolver uma aplicação que fosse capaz de fazer o registo das despesas que são geradas num apartamento, assim como a gestão do pagamento feita por cada um dos moradores do apartamento em questão. Decidimos que seria interessante desenvolver um sistema que permitisse aos utilizadores usufruírem do controlo de terem as suas despesas devidamente divididas, onde os pagamentos das mesmas fossem o mais breve possível, organizado e ainda a facilidade de acesso através de um smartphone, tablet ou pc para a consulta desta mesma aplicação.

O trabalho será dividido em duas fases que se completam uma à outra. Na primeira fase será descrito o processo de análise de requisitos suportada pelo modelo de domínio do sistema, casos de uso e respetivas especificações e uma possível interface com o utilizador. Tudo isto de forma a enquadrar e descrever da forma mais detalhada possível o sistema a ser desenvolvido. Serão expostos os desenhos de planificação das interfaces e da sua correlação com as funcionalidades a serem implementadas no sistema. Na segunda fase faremos com que a nossa aplicação ganhe vida e desta forma conseguiremos fazer chegar ao público alvo aquilo que seria o nosso produto final. Desta forma faremos a junção de uma base de dados com o sistema por nós já implementado.

## 1.1 Apresentação do Caso de Estudo

A aplicação terá como objetivo desenvolver um sistema de despesas num apartamento capaz de suportar o registo de despesas e a gestão do pagamento dessas mesmas despesas por parte dos moradores. Este é um sistema que proporciona aos seus utilizadores a possibilidade de efetuarem os pagamentos das suas despesas, sejam estas recorrentes (por exemplo, água ou eletricidade) ou extraordinárias (por exemplo, necessidade de realizar alguma reparação no apartamento) fazendo com que o controlo de dívidas entre moradores estejam sempre atualizadas e visíveis para todos os utilizadores da aplicação. Neste sistema existe um morador previamente registado na aplicação necessitando de fornecer o nome, data de nascimento, e-mail e nr de telefone e uma password para efetuar o registo. Terá a ele associada uma conta corrente que será uma espécie de fundo do qual mensalmente (ou quando necessário) é creditado o pagamento, ou seja, a quantia necessária afeta as despesas correspondentes. A conta corrente de cada morador contribui para o saldo global, ou seja, no nosso sistema existe um saldo que resulta do somatório de todas as contas correntes e que corresponde ao montante total que o apartamento tem como despesa nesse mês.

O saldo global é administrado por um Administrador que é responsável por verificar se

o montante deste mesmo está completo e depois dessa forma, pagar a despesa. A todas as despesas está associado um valor assim como cada pagamento creditado da conta corrente de cada morador.

Ainda de salientar que a cada morador está associada uma fração que varia consoante o tipo de morador (exemplo: moradores que partilham quarto, terão uma fração do valor da renda menor relativamente a um morador com quarto individual), relativamente às restantes despesas (água, luz, gás, internet, ou até mesmo despesas extraordinárias).

## **2. Requisitos do Sistema**

### **2.1 Análise e Levantamento dos Requisitos**

#### **2.1.1 Criação do grupo com os elementos da casa/apartamento**

A aplicação a desenvolver deverá suportar o registo das despesas e a gestão do seu pagamento por parte de moradores registados. Para tal é necessário que cada morador efetue um registo fornecendo o seu nome, e-mail, número de telemóvel e data de nascimento.

O utilizador que convidar os restantes será considerado administrador do sistema.

Após o registo será associada uma conta corrente, esta que permitirá a gestão do saldo (estão incluídas dívidas) de cada morador.

#### **2.1.2 Gestão das contas**

O morador deverá efetuar um pagamento, que será creditado na conta corrente do morador.

O pagamento pode ser efetuado com valores relativamente grandes ao qual é debitado consoante é necessário ou depósitos da quantia necessária para pagar a/s despesa/s em causa.

Existirá um saldo global da casa, este que é o somatório de todas as contas correntes dos moradores, este saldo é administrado por um Administrador.

Existem dois tipos de despesas distintos: a despesa recorrente referente a despesas mensais como a água, luz, renda, etc; a despesa extra referente a por exemplo arranjos de material na casa.

Cada morador deverá pagar uma fração relativa à despesa.

Cada despesa a pagar tem a si associado um tipo.

### **2.2 Base de dados**

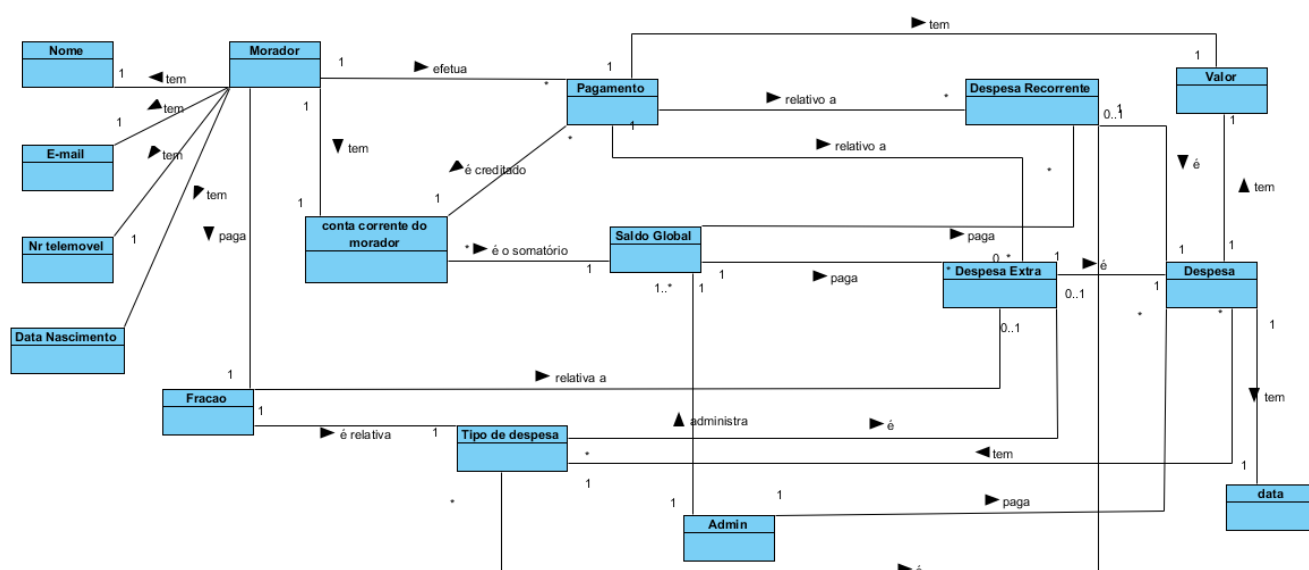
A aplicação necessitará de uma implementação de uma base de dados para gerir os elementos que pertencem ao grupo assim como as despesas efetuadas pelos moradores.

## 3. Arquitetura da Aplicação

### 3.1 Modelo de Dominio

Todo e qualquer projeto possui um domínio específico. O modelo de domínio deve capturar os seguintes pontos: as entidades, os relacionamentos entre as entidades e o vocabulário de domínio do problema. Para além disso também deve ser uma visão estática do problema onde é possível representar as regras de negócio invariantes no tempo. Ou seja, o modelo de domínio é a base para a análise de requisitos.

No que diz respeito à aplicação, como é dito na introdução, queremos desenvolver uma aplicação capaz de suportar o registo das despesas e a gestão do seu pagamento por parte dos moradores registados.



**Figura 3.1:** Modelo Dominio

O morador necessita de fornecer o nome, e-mail, número de telemovel e data de nascimento, para efetuar o login.

Como se pode observar na figura o morador efetua pagamento relativo a despesa recorrente ou extra, assim como paga uma fração da despesa, essa fração é relativa a um tipo de despesa.

O administrador administra o saldo global da casa/apartamento.

## 3.2 Modelo de Use Case

A segunda parte da análise de requisitos corresponde à definição dos use cases, com o objetivo de os aplicar nesta primeira fase deste trabalho prático. Nos use cases, queremos primeiramente, identificar os atores, que serão quem interagirá com o sistema. Posterior à identificação dos atores, passamos então à identificação dos use cases, isto é, o que se pretende do sistema. No último ponto da visão orientada aos use cases, procedemos à identificação das classes de suporte à realização dos mesmo, que corresponde à especificação da funcionalidade a ser implementada. Neste sentido, quando definimos um use case, para além de ser uma espécie de documentação, temos de ter em conta que se trata de uma unidade coerente de funcionalidade, um serviço. Define também um comportamento do sistema, sem revelar a estrutura interna, divulgando desta forma, a comunicação entre os atores e o sistema. O conjunto de todos os use cases acaba por definir pela íntegra, toda a funcionalidade do sistema que decorre na sua essência, do diálogo entre o sistema e os atores, e a responsabilidade de resposta funcional do sistema.

### 3.2.1 Diagrama de Use Case

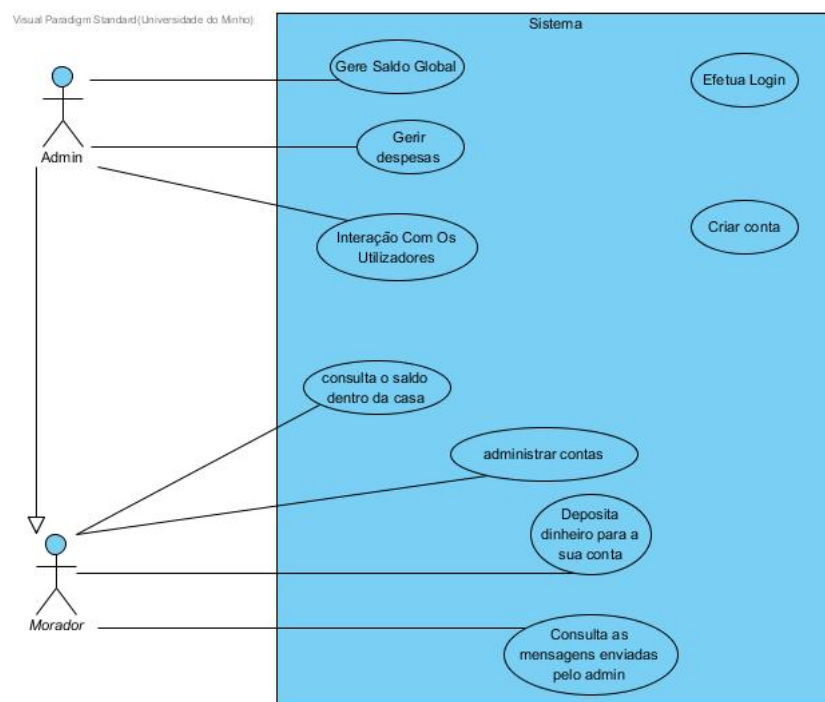
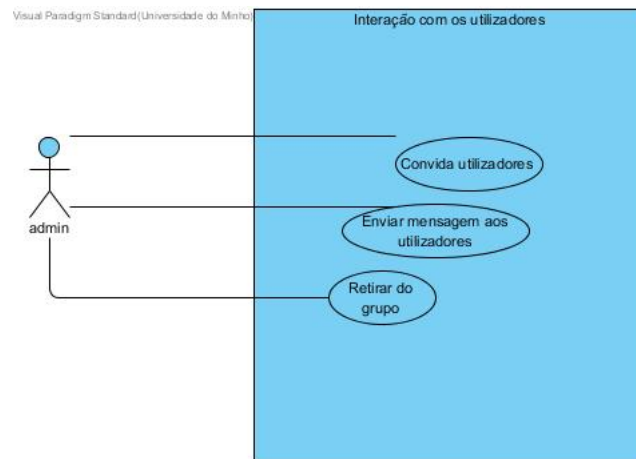
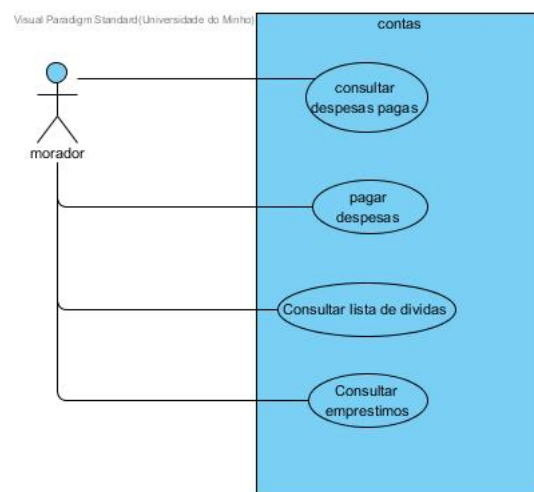


Figura 3.2: Modelo Use Case

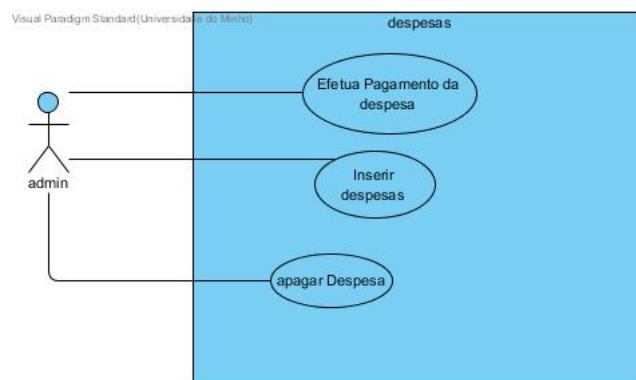
### 3.2.2 Subdiagramas



**Figura 3.3:** Sub-Diagrama Interação com os Utilizadores



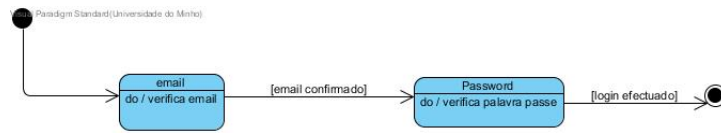
**Figura 3.4:** Sub-Diagrama Despesas



**Figura 3.5:** Sub-Diagrama gerir despesa

### 3.3 Mockups





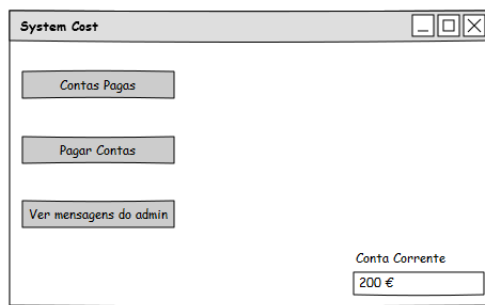
**Figura 3.6:** Login

The screenshot shows a window titled "System Cost" with standard window controls (minimize, maximize, close). The form contains the following fields and labels:

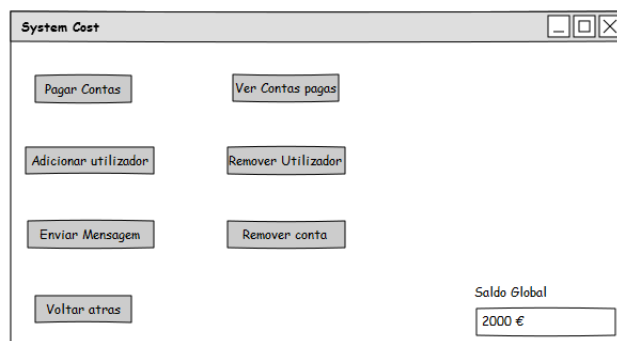
- Introduza Nome Completo:
- Introduza Data de Nascimento:
- Introduza Email:
- Introduza Password:
- Repetir Password:
- Introduza o número de telemóvel:

At the bottom of the form are two buttons: "Criar Conta" and "Cancelar".

**Figura 3.7:** Criar nova Conta



**Figura 3.8:** Utilizador Normal



**Figura 3.9:** Privilégios de administrador

## 4. Conclusão

Uma vez que se tratou de um trabalho de uma dimensão já considerável comparando com o que estávamos habituados envolveu utilização de técnicas particulares e tivemos sempre como objetivo que este trabalho fosse concebido de modo a que seja facilmente modificável, e seja, apesar da complexidade, o mais optimizado possível a todos os níveis.

o modelo de dominio as especificações de uses cases