



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

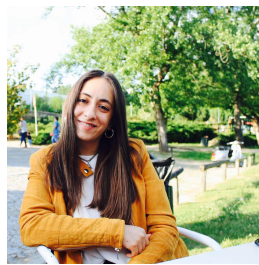
### **1ª Fase**



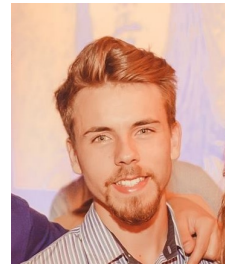
Célia Fi-  
gueiredo  
a67637



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 . . . . .	5
<b>3</b>	<b>Arquitetura da Aplicação</b>	<b>6</b>
3.1	Modelo de Dominio . . . . .	6
3.2	Modelo de Use Case . . . . .	7
3.2.1	Diagrama de Use Case . . . . .	7
3.2.2	Subdiagrama Gerir Despesas . . . . .	11
3.2.3	Subdiagrama Administrar Contas . . . . .	13
3.2.4	Subdiagrama Interação com os Utilizadores . . . . .	16
3.3	Máquinas de Estado . . . . .	17
3.3.1	Efetuar Login . . . . .	17
3.3.2	Criar Conta . . . . .	18
3.3.3	Convidar Utilizadores . . . . .	18
3.3.4	Remover Utilizadores . . . . .	19
3.3.5	Inserir Fatura . . . . .	19
3.3.6	Deposita Dinheiro . . . . .	19
3.4	Mockups . . . . .	20
<b>4</b>	<b>Conclusão</b>	<b>24</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 conseguirmos 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 número de telefone/telemóvel 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), assim como relativamente às restantes despesas (água, luz, gás, internet, ou até mesmo despesas extraordinárias).

## 2. Requisitos do Sistema

### 2.1 Análise e Levantamento dos Requisitos

A aplicação a desenvolver deverá suportar o registo das despesas e a gestão do seu pagamento por parte de moradores registados.

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

- Cada morador deve efetuar um registo fornecendo o seu nome, e-mail, número de telemóvel e data de nascimento.
- O morador assim como o utilizador necessitam de efetuar login na aplicação
- O utilizador que convidar os restantes será considerado administrador do sistema.

#### 2.1.2 Gestão das contas

- Após o registo será associada uma conta corrente a cada utilizador.
- Cada conta corrente permitirá a gestão do saldo corrente (estão incluídas dívidas) de cada morador.
- O morador deverá efetuar um pagamento.
- Cada pagamento será creditado na conta corrente do morador.
- O pagamento pode ser igual ou superior à quantia necessária para pagar as despesas em causa.
- Existirá um saldo global da casa, este que é o somatório de todas as contas correntes dos moradores
- O saldo global 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 (i.e. água, luz, cadeiras, etc)

## **2.2 Base de dados**

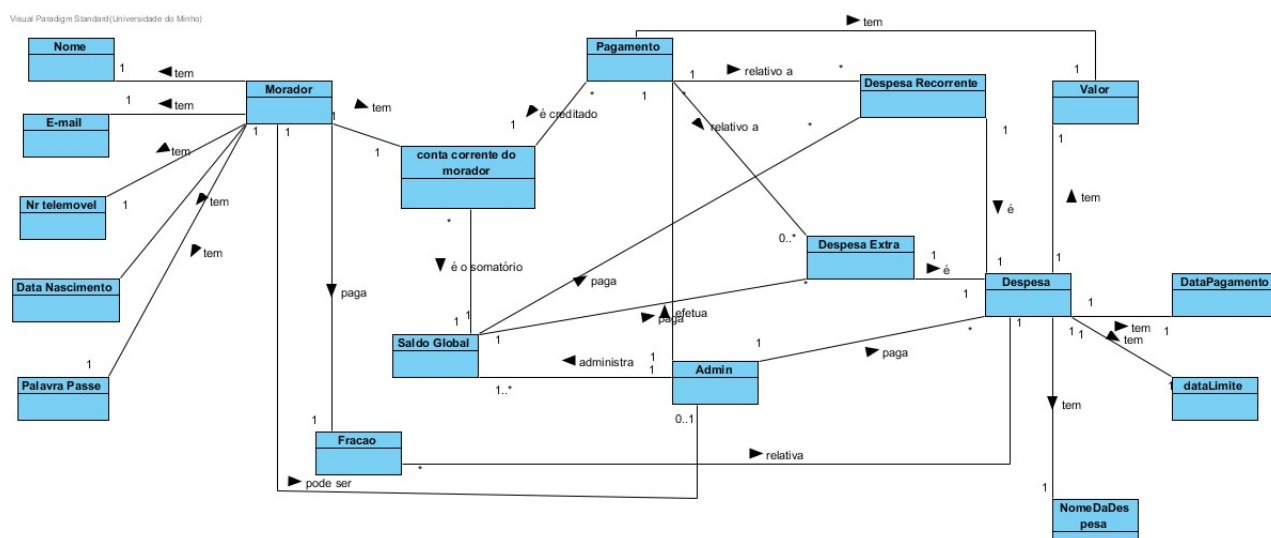
A aplicação necessitará de uma implementação de uma base de dados para gerir os elementos que pertencem ao grupo assim como as despesas efetuadas pelos moradores.

## 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.

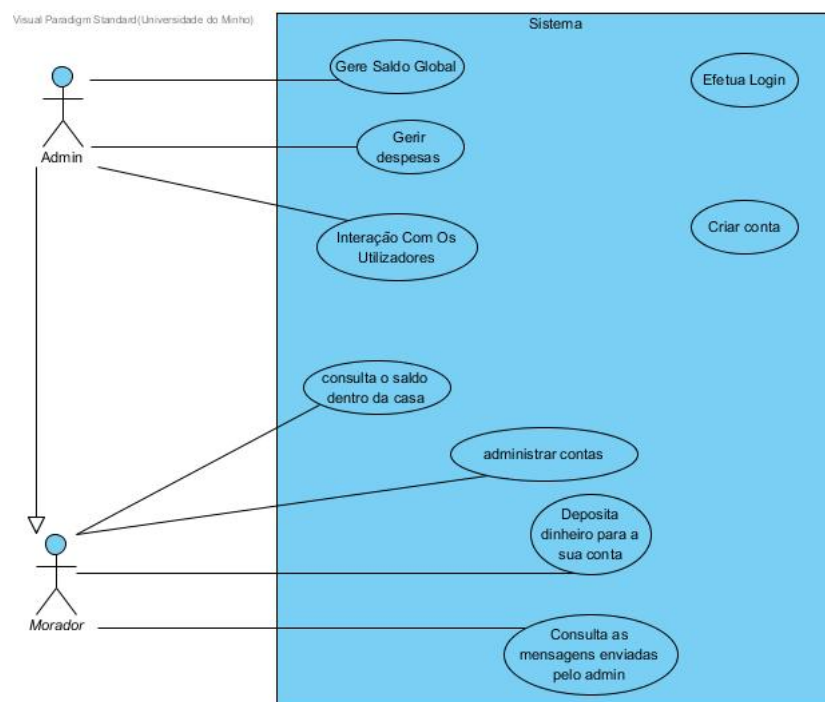
Como se pode observar na figura o morador efetua pagamento relativo a despesa recorrente ou extra, assim como paga uma fração da despesa, essa fração é relativa a um tipo de despesa.

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



## Especificação: Gere Saldo Global

Preconditions	Admin tem login efetuado	
Post-conditions		
Flow of Events		Actor Input
	1	
	2	o admin vê o dinheiro
	3	O admin volta ao menu inicial
	4	
Comp. Alternativo (passo 3) [Utilizador manda mensagens aos utilizadores]		System Response
	1	o sistema apresenta o dinheiro do saldo
	2	
	3	o sistema pergunta para quem quer enviar a mensagem
	4	o sistema envia a mensagem

Figura 3.3: Especificação do Use Case: Gere Saldo Global

## Especificação: Efetua Login

Preconditions	Ter criado conta	
Post-conditions	Ter efectuado o login	
Flow of Events		Actor Input
	1	
	2	o utilizador introduz o email
	3	
	4	
	5	o utilizador introduz a palavra passe
	6	
Exceção 1 [email inválido] (passo 3)		System Response
	1	o sistema pede que introduza o email
Exceção 2 [palavra passe incorreta] (passo 6)		System Response
	1	o sistema valida o email
		System Response
		o sistema pede que introduza a palavra passe
		System Response
		o sistema valida a palavra passe
		System Response
		login efectuado com sucesso

Figura 3.4: Especificação do Use Case: Efetua Login

## Especificação: Criar Conta

Post-conditions	Conta Criada		
Flow of Events		Actor Input	System Response
	1		Inserir Nome Completo
	2	Utilizador insere nome	
	3		Inserir data de nascimento
	4	Utilizador insere data de nascimento	
	5		Inserir email
	6	insere mail	
	7		verifica e-mail
	8		pede para inserir password
	9	insere password	
	10		verifica que password
	11		pede para inserir nr de telemovel
	12	insere nr de telemovel	
	13		Regista dados
Excecao 1 (passo 7) [E-mail inválido.]		Actor Input	System Response
	1		sistema informa que mail é inválido
Excecao 2 (passo 10) [password não cumpra requisitos.]		Actor Input	System Response
	1		informa que password não cumpre requisitos minimos

Figura 3.5: Especificação do Use Case: Criar Conta

## Especificação: Consulta Saldo dentro de casa

Date	23/out/2016 23:05:17		
Brief Description			
Preconditions	O utilizador ter feito o login		
Post-conditions			
Flow of Events		Actor Input	System Response
	1		o sistema apresenta o saldo dentro da casa
	2	o utilizador visualiza esse dinheiro	

Figura 3.6: Especificação do Use Case: Consulta Saldo dentro de Casa

## Especificação: Deposita Dinheiro para a sua Conta

Date	17/Out/2016 16:16:28	
Brief Description		
Preconditions	login efetuado	
Post-conditions	conta corrente atualizada	
Flow of Events		<b>Actor Input</b>
		<b>System Response</b>
	1	Pede para inserir quantia a adicionar
	2	insere quantia
	3	Atualiza conta corrente

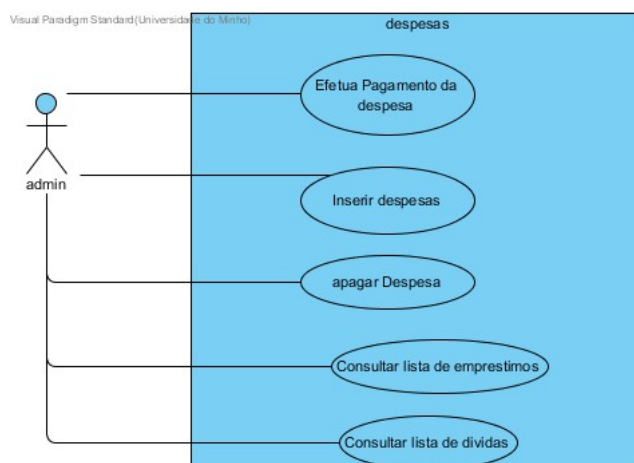
**Figura 3.7:** Especificação do Use Case: Deposita Dinheiro para a sua conta

## Especificação: Consulta Mensagens enviadas pelo Administrador

Date	23/out/2016 21:54:07	
Brief Description		
Preconditions	Ter o login efectuado	
Post-conditions	Consegue consulta as mensagens recebidas	
Flow of Events		<b>Actor Input</b>
		<b>System Response</b>
	1	o sistema apresenta as mensagens enviadas pelo admin
	2	o utilizador escolhe a mensagem que quer ler
	3	o sistema apresenta a mensagem seleccionada
Exceção(passo 1) [Não existem mensagens]		<b>Actor Input</b>
	1	O sistema informa que não existem mensagens para apresentar

**Figura 3.8:** Especificação do Use Case: Consulta Mensagens enviadas pelo Administrador

### 3.2.2 Subdiagrama Gerir Despesas



**Figura 3.9:** Sub-Diagrama Gerir Despesas

#### Especificação: Efetua pagamento da despesa

Author	Célia Figueiredo		
Date	17/Out/2016 16:17:24		
Brief Description			
Preconditions	Ter dinheiro suficiente para pagar a despesa e ter efectuado o login		
Post-conditions	Pagamento efectuado		
Flow of Events		Actor Input	System Response
	1		O sistema apresenta as faturas que estão para ser pagas
	2	escolhe fatura para pagar	
	3		Apresenta valor a pagar
	4	Confirma que pagou (Ok)	
	5		Sistema Regista pagamento e desconta o valor da factura no saldo global
Excecao (passo 1) [Nao existe faturas a pagar]		Actor Input	System Response
	1		o sistema informa que não existe faturas para serem pagas

**Figura 3.10:** Especificação do Use Case:Efetua Pagamento da despesa

## Especificação: Inserir Despesa

<b>Preconditions</b>	o admin está autenticado	
<b>Post-conditions</b>	despesa inserida	
<b>Flow of Events</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	O sistema pede que introduza o nome da fatura
	2	Inserir nome da fatura
	3	O sistema pede que introduza a data da fatura
	4	O utilizador introduz a data
	5	O sistema valida a data
	6	O sistema pede que introduza o valor da fatura
	7	Inserir valor da fatura
	8	O sistema valida o valor
	9	O sistema apresenta os tipos possíveis da fatura
	10	O utilizador escolhe o tipo normal
	11	O sistema envia os valores para os utilizadores
	12	a fatura é introduzida com sucesso
	13	
<b>Exceção (passo 4 [O sistema não valida a data])</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	O sistema informa que não consegue validar a data introduzida
	2	
<b>Exceção (passo 7) [o sistema não valida o valor introduzido]</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	O sistema informa que o valor introduzido é inválido
<b>Comp. Alternativo (passo 2) [Inser despesa extra]</b>		<b>Actor Input</b>
		<b>System Response</b>
	1	O sistema apresenta os utilizadores possíveis para pagarem
	2	O admin escolhe os utilizadores
	3	O sistema pede que introduza a percentagem a pagar
	4	O admin introduz a percentagem que cada um tem de pagar
	5	regressa 12

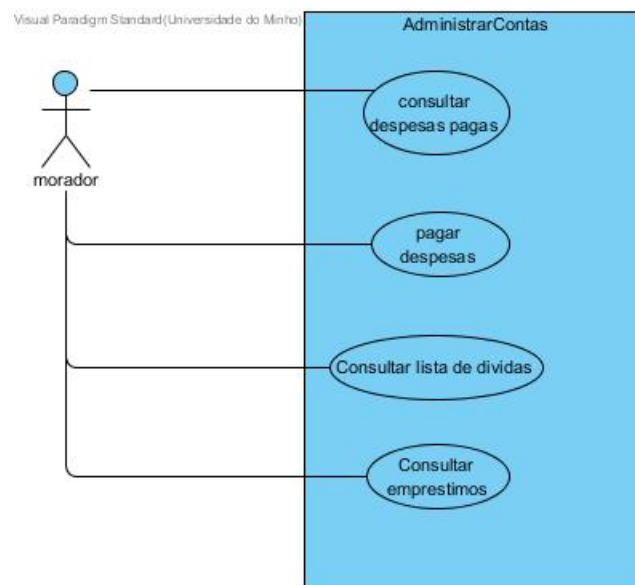
**Figura 3.11:** Especificação do Use Case: Inserir Despesas

## Especificação: Apagar Despesa

Date	8/Nov/2016 1:00:26	
Brief Description		
Preconditions	o admin tem o login efectuado	
Post-conditions		
Flow of Events		Actor Input
		System Response
	1	o sistema apresenta as despesas por pagar
	2	o utilizador escolhe a despesa que quer apagar
	3	o sistema apaga a despesa
Excecao(passo1)[Não existem despesas por pagar]		Actor Input
	1	o sistema informa que não existe despesa por pagar

**Figura 3.12:** Especificação do Use Case: Apagar Despesa

### 3.2.3 Subdiagrama Administrar Contas



**Figura 3.13:** Subdiagrama: Administrar Contas

## Especificação: Consultar despesas pagas

Date	23/out/2016 22:45:49	
Brief Description		
Preconditions	Ter efectuado o login	
Post-conditions	Conseguir visualizar a fatura pretendida	
Flow of Events		<b>Actor Input</b>
		<b>System Response</b>
	1	Apresenta a lista das despesas pagas
	2	o utilizador escolhe a fatura que quer ver
Exceção (passo1) [Não existem faturas pagas]	3	o sistema apresenta detalhes da fatura
		<b>Actor Input</b>
		<b>System Response</b>
	1	o sistema informa que não existem faturas pagas
	2	

Figura 3.14: Especificação do Use Case: Consultar Despesas Pagas

## Especificação: Pagar Despesas

Preconditions	o utilizador tem o login efectuado	
Post-conditions		
Flow of Events		<b>Actor Input</b>
		<b>System Response</b>
	1	o sistema apresenta as conta para pagar
	2	o utilizador escolhe uma conta
	3	o sistema apresenta o valor a pagar
	4	o utilizador paga o valor
	5	o sistema valida o pagamento
Comp.Alternativo(passo4)[utilizador paga a menos]	6	o sistema regista a informacao
		<b>Actor Input</b>
		<b>System Response</b>
	1	o utilizador paga menos que o valor pretendido
	2	o sistema insere o utilizador na lista de dividas e o valor que esta a dever e desconta o valor paga na fatura
	3	regressa ao passo 5
Excecao(passo 5)[Dinheiro Insuficiente]		<b>Actor Input</b>
	1	o sistema informa que não tem dinheiro suficiente na conta para pagar o que pretende e desconta o valor paga na fatura
Comp.Alternativo(passo4)[utilizador paga a mais]		<b>Actor Input</b>
		<b>System Response</b>
	1	o utilizador paga a mais do que o valor pretendido
	2	o sistema insere o utilizador na lista de emprestimos e o valor que ele emprestou
	3	regressa ao passo 5
Excecao(passo 5)[Valor introduzido a mais]		<b>Actor Input</b>
	1	o sistema informa que o valor introduzido é demasiado para pagar a conta

Figura 3.15: Especificação do Use Case: Pagar Despesas

## Especificação: Consultar Lista de Dívidas

Preconditions	o utilizador tem de ter efectuado o login		
Post-conditions			
Flow of Events		Actor Input	System Response
	1		o sistema apresenta a lista de dividas
	2	o utilizador escolhe uma para pagar	
	3	o utilizador efectua o pagamento	
	4		o sistema valida o pagamento
	5		o sistema remove a divida da lista
Excecção (passo 4) [Dinheiro introduzido insuficiente]		Actor Input	System Response
	1		o sistema informa que o utilizador não tem dinheiro suficiente para pagar a divida
Excecção (passo 4) [Dinheiro introduzido a menos]		Actor Input	System Response
	1		o sistema informa que o dinheiro introduzido não chega para pagar a divida
Excecção (passo 4) [Dinheiro introduzido a mais]		Actor Input	System Response
	1		o sistema informa que o utilizador introduziu dinheiro a mais
Excecção (passo 1) [o utilizador não tem dividas para pagar]		Actor Input	System Response
	1		o sistema informa que o utilizador não tem dividas para pagar

Figura 3.16: Especificação do Use Case: Consultar lista de dívidas

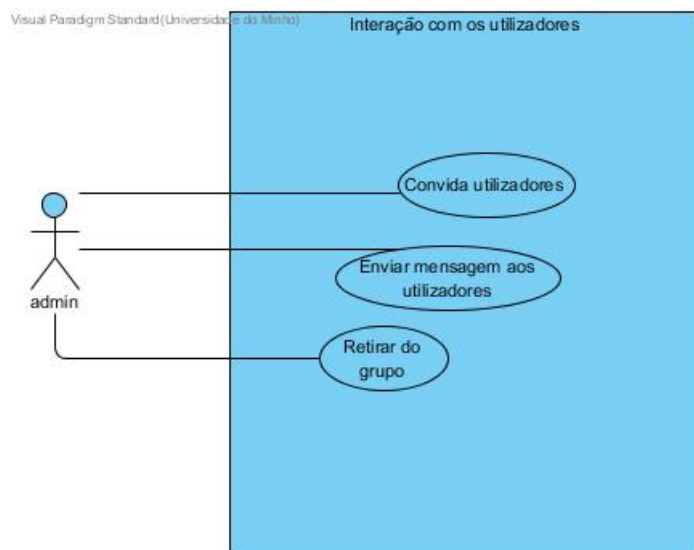
## Especificação: Consultar Empréstimos

Date	8/nov/2016 0:12:52		
Brief Description			
Preconditions			
Post-conditions			
Flow of Events		Actor Input	System Response
	1		o sistema apresenta a lista de empréstimos
	2	o utilizador escolhe um e vê o dinheiro que emprestou	
Execução(passo 1) [não fez empréstimos]		Actor Input	System Response
	1		o sistema informa que o utilizador não fez empréstimos

Figura 3.17: Especificação do Use Case: Consultar Empréstimos



### 3.2.4 Subdiagrama Interação com os Utilizadores



**Figura 3.18:** Subdiagrama: Interação com os utilizadores

### Especificação: Convida Utilizadores

Date	17/Out/2016 16:44:57	
Brief Description		
Preconditions	login efetuado	
Post-conditions	convite efetuado	
Flow of Events		Actor Input
	1	
	2	Insere e-mail
	3	
	4	
Exceção 1 (passo 3) [e-mail inválido]		System Response
	1	pede para inserir e-mail
		envia e-mail para os utilizadores
Exceção 1 (passo 3) [e-mail inválido]		Actor Input
	1	
Exceção 1 (passo 3) [e-mail inválido]		System Response
		informa que mail não é válido

**Figura 3.19:** Especificação do Use Case: Convida Utilizadores

## Especificação: Envia Mensagens aos Utilizadores

Date	1/nov/2016 23:28:15	
Brief Description		
Preconditions	Ter o login efectuado	
Post-conditions		
Flow of Events		<b>Actor Input</b>
		<b>System Response</b>
	1	O sistema apresenta os utilizadores aos quais pode enviar mensagem
	2	O admin escolhe um utilizador
	3	O admin escreve a mensagem
	4	Mensagem é enviada

Figura 3.20: Especificação do Use Case: Envia Mensagens aos Utilizadores

## Especificação: Retirar do grupo

Date	23/out/2016 19:45:13	
Brief Description		
Preconditions	o admin ter o login efetuado	
Post-conditions	o admin consegue remover o utilizador	
Flow of Events		<b>Actor Input</b>
		<b>System Response</b>
	1	O sistema apresenta a lista de utilizadores que estão a viver no apartamento
	2	O admin escolhe qual o utilizador que quer remover do grupo
Excecao [Não consegue remover o utilizador] (passo 3)	3	o sistema consegue remover o utilizador
	1	o sistema informa que o utilizador não pode ser removido

Figura 3.21: Especificação do Use Case: Retirar do grupo

## 3.3 Máquinas de Estado

### 3.3.1 Efetuar Login

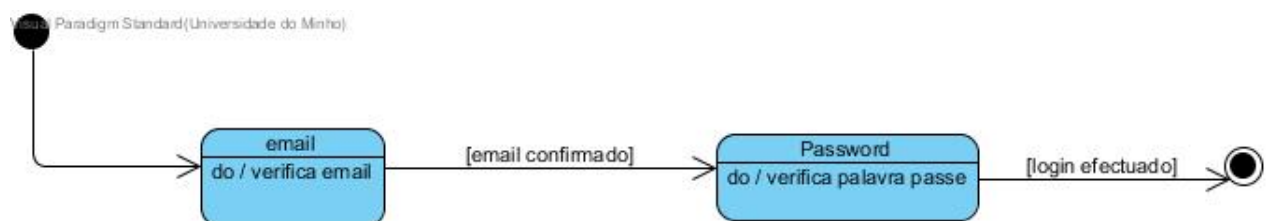


Figura 3.22: Máquina de Estados: Efetuar Login

### 3.3.2 Criar Conta

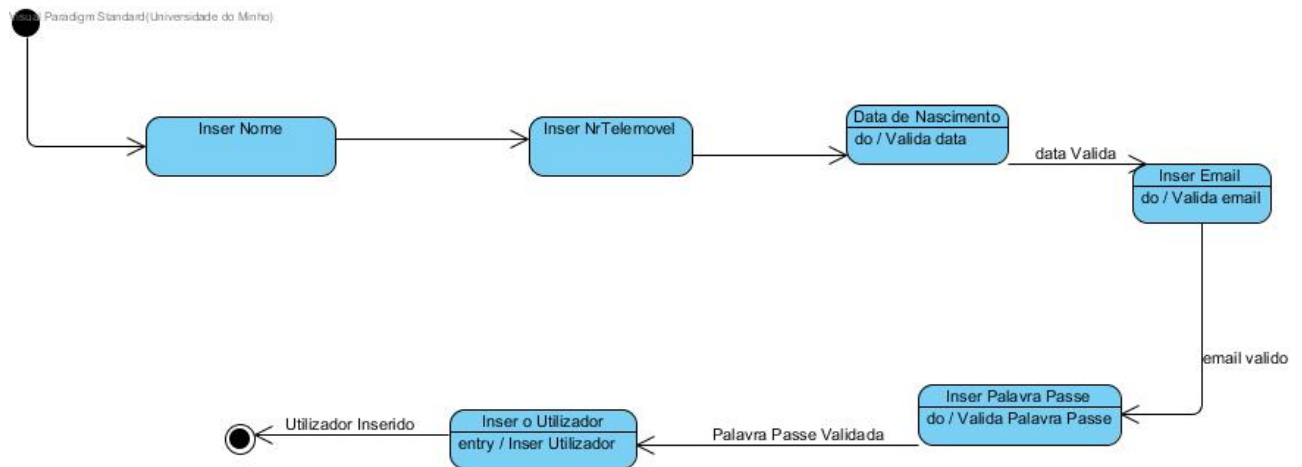


Figura 3.23: Máquina de Estados: Criar Conta

### 3.3.3 Convidar Utilizadores

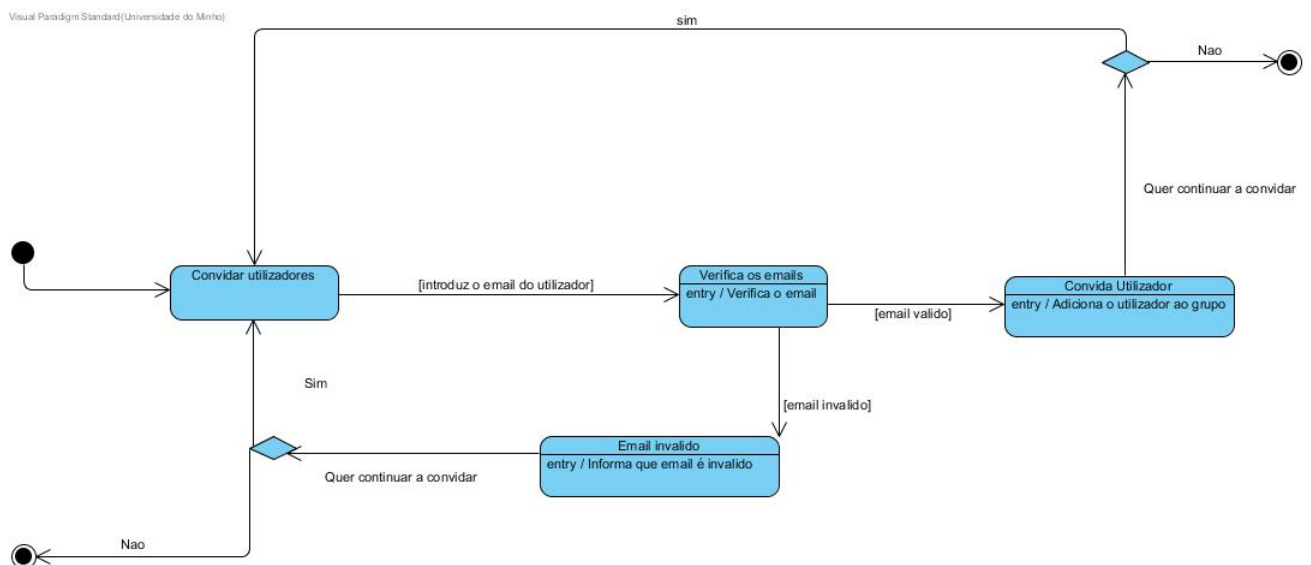


Figura 3.24: Máquina de Estados: Convidar Utilizadores

### 3.3.4 Remover Utilizadores

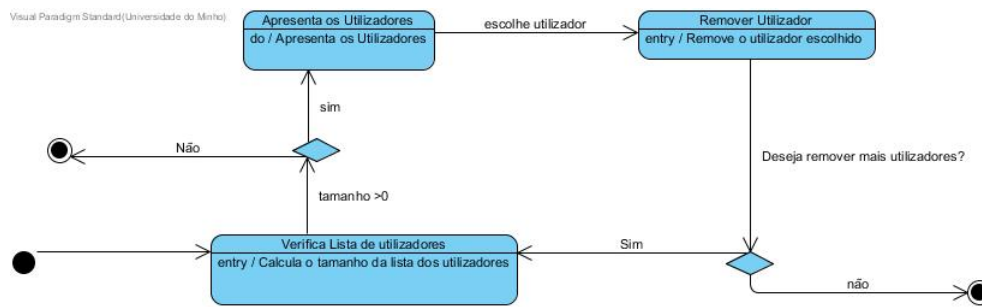


Figura 3.25: Máquina de Estados: Remover Utilizadores

### 3.3.5 Inserir Fatura

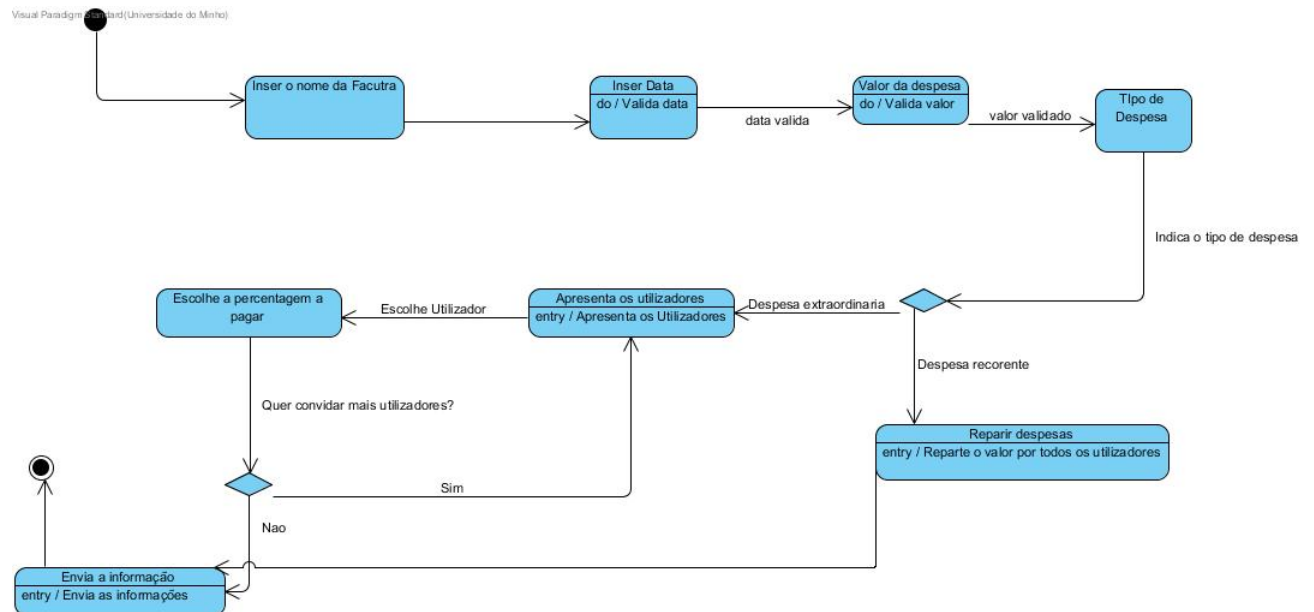


Figura 3.26: Máquina de Estados: Inserir Fatura

### 3.3.6 Deposita Dinheiro

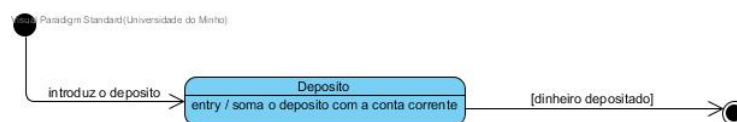
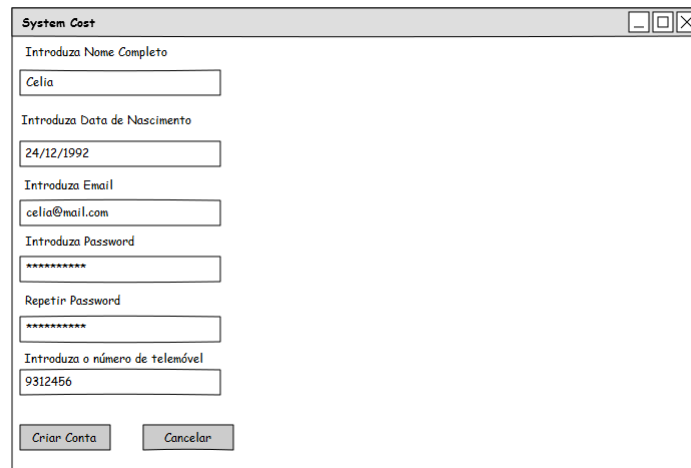


Figura 3.27: Máquina de Estados: Deposita Dinheiro

## 3.4 Mockups

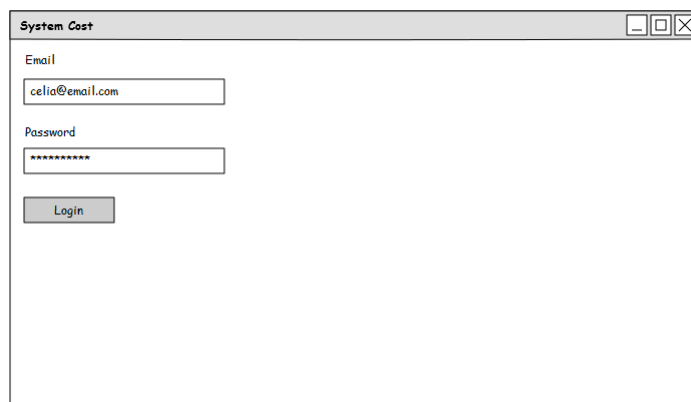
Apresentamos de seguida uma proposta de interface com o utilizador. Utilizámos o programa 'Pencil' para nos auxiliar na construção de uma possível interface com o utilizador.

Como já refirmos, para o utilizador efetuar o login necessita de se registar previamente, fornecendo alguns dados que o identifiquem.



The mockup shows a window titled 'System Cost' with standard window controls. It contains a vertical stack of input fields with labels: 'Introduza Nome Completo' (containing 'Celia'), 'Introduza Data de Nascimento' (containing '24/12/1992'), 'Introduza Email' (containing 'celia@email.com'), 'Introduza Password' (containing '\*\*\*\*\*'), 'Repetir Password' (containing '\*\*\*\*\*'), and 'Introduza o número de telemóvel' (containing '9312456'). At the bottom are two buttons: 'Criar Conta' and 'Cancelar'.

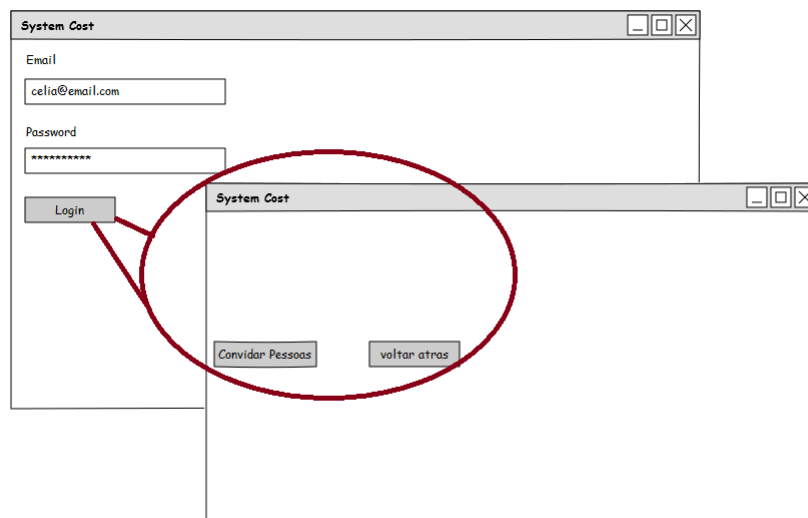
**Figura 3.28:** Criar nova Conta



The mockup shows a window titled 'System Cost' with standard window controls. It contains two input fields with labels: 'Email' (containing 'celia@email.com') and 'Password' (containing '\*\*\*\*\*'). Below these fields is a single button labeled 'Login'.

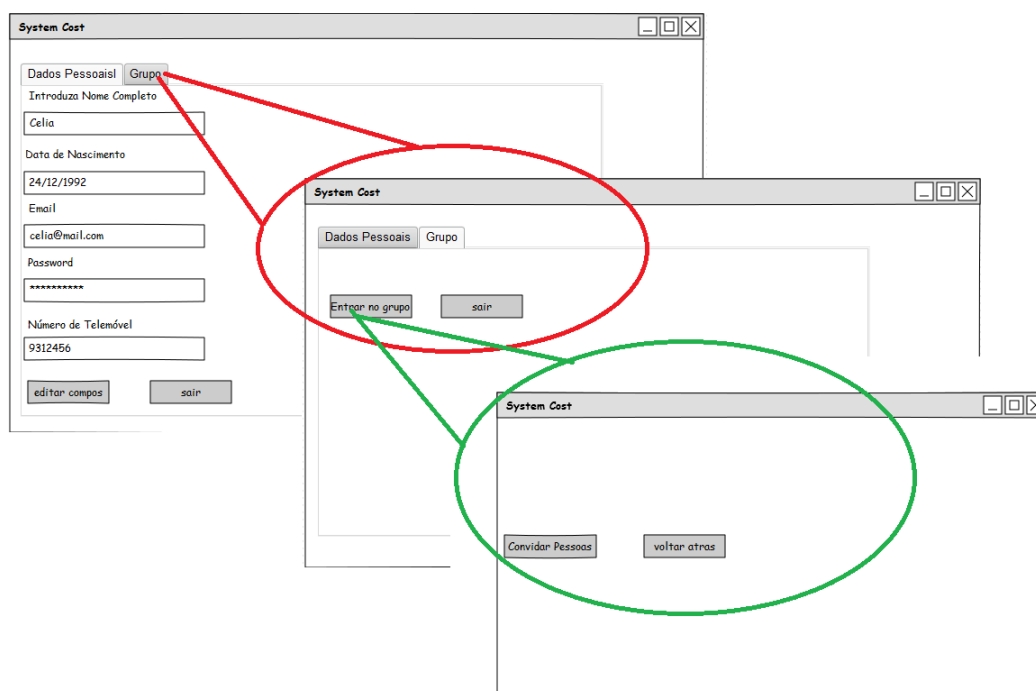
**Figura 3.29:** Login

O Administrador efetua login, mas ainda não existem grupo criado. Pode escolher a opção "Convidar Pessoas", para iniciar a formação de um grupo.



**Figura 3.30:** Login quando não existem grupos criados

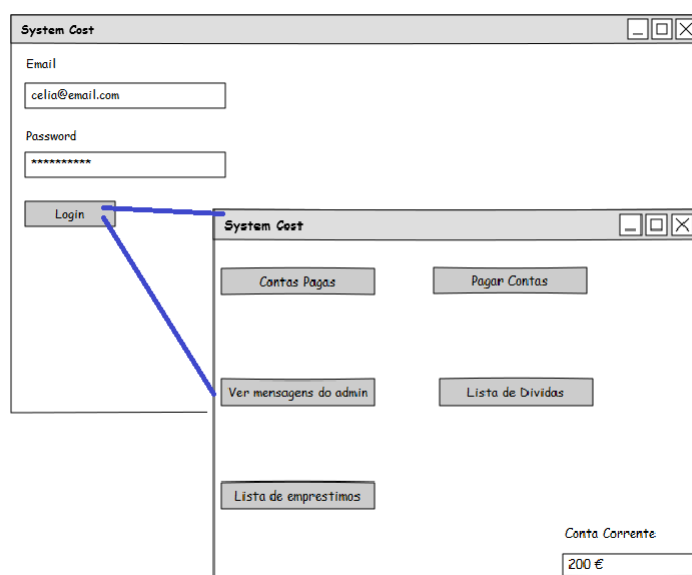
Haverá sempre a possibilidade de ver/alterar os campos preenchidos inicialmente. Carregando no botão grupo abre uma janela onde se pode entrar para grupo constituído pelos elementos da casa/apartamento. Após clicar no botão "Entrar no grupo" é-nos apresentada uma lista com as pessoas já existentes e a possibilidade de convidar mais membros.



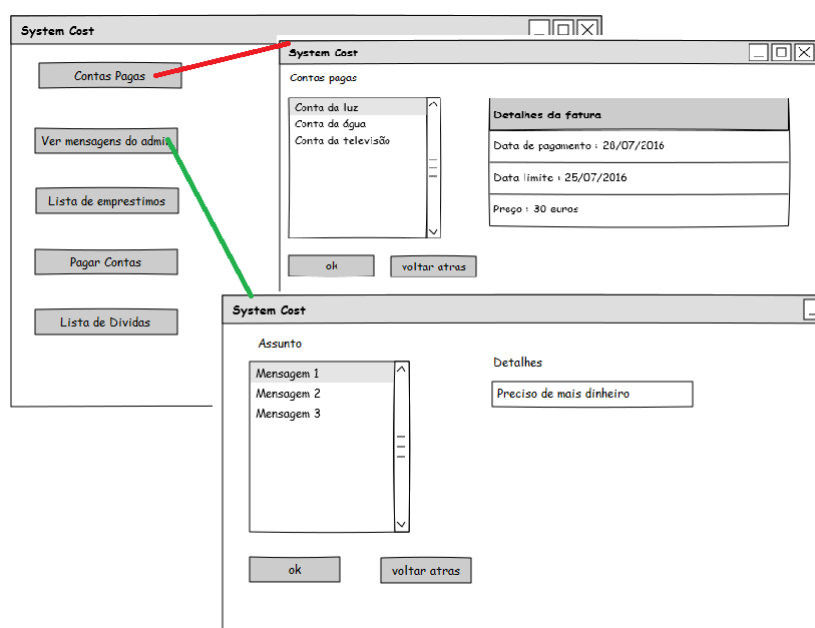
**Figura 3.31:** Visualização/Alteração dos dados

Após o utilizador efetuar o login é-lhe apresentada uma janela com as funcionalidades que a aplicação lhe oferece, como por exemplo pagar contas e acesso à lista de dividas, assim como

o valor da conta corrente.

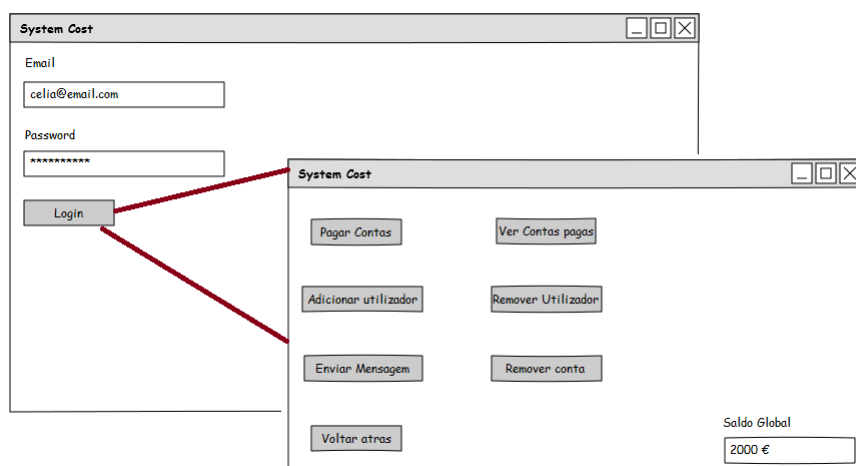


**Figura 3.32:** Login/página inicial



**Figura 3.33:** Opções do utilizador

Após o administrador efetuar o login é-lhe apresentada uma janela com as funcionalidades que a aplicação lhe oferece, como por exemplo pagar contas e adicionar/remover utilizador, enviar mensagem e verificar o saldo global



**Figura 3.34:** Login/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