



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

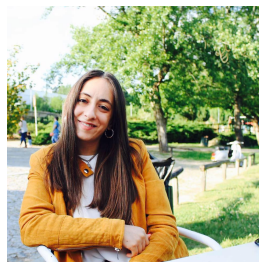
Fase Final



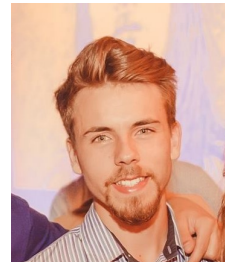
Célia Fi-
gueiredo
a67637



Gil
Gonçalves
a67738



Márcia
Costa
a67672



Daniel
Rodrigues
a67634



Ricardo
Lopes
a72062

Braga, 27 de Dezembro de 2016

Conteúdo

1	Introdução	2
1.1	Apresentação do Caso de Estudo	2
2	Requisitos do Sistema	4
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
3	Arquitetura da Aplicação	8
3.1	Modelo de Dominio	8
3.2	Diagrama de Classes	9
3.3	Modelo de Use Case	10
3.3.1	Diagrama de Use Case	10
3.3.2	Subdiagrama Gerir Despesas	17
3.3.3	Subdiagrama Administrar Contas	22
3.3.4	Subdiagrama Interação com os Utilizadores	28
4	Implementação e Instalação do Sistema	32
4.1	Diagrama de Instalação	32
5	Interface do Sistema Gestão de Despesas - SGD	34
5.1	Máquinas de Estado	34
5.1.1	Efetuar Login	34
5.1.2	Criar Conta	34
5.1.3	Convidar Utilizadores	35
5.1.4	Remover Utilizadores	35
5.1.5	Inserir Fatura	35
5.1.6	Deposita Dinheiro	35
5.2	Mockups	36
6	Conclusão	41

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 concepção 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.

Nesta 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. Será necessária a criação de

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.
- Cada pagamento efetuado não pode ultrapassar o valor da fatura
- 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. A modelação concetual consiste no processo de construção de um modelo de dados a partir de uma análise detalhada dos requisitos. Este modelo é completamente independente de todas as considerações físicas, ou seja é independente dos detalhes de implementação. De seguida está apresentado o esquema concetual relativo ao projeto.

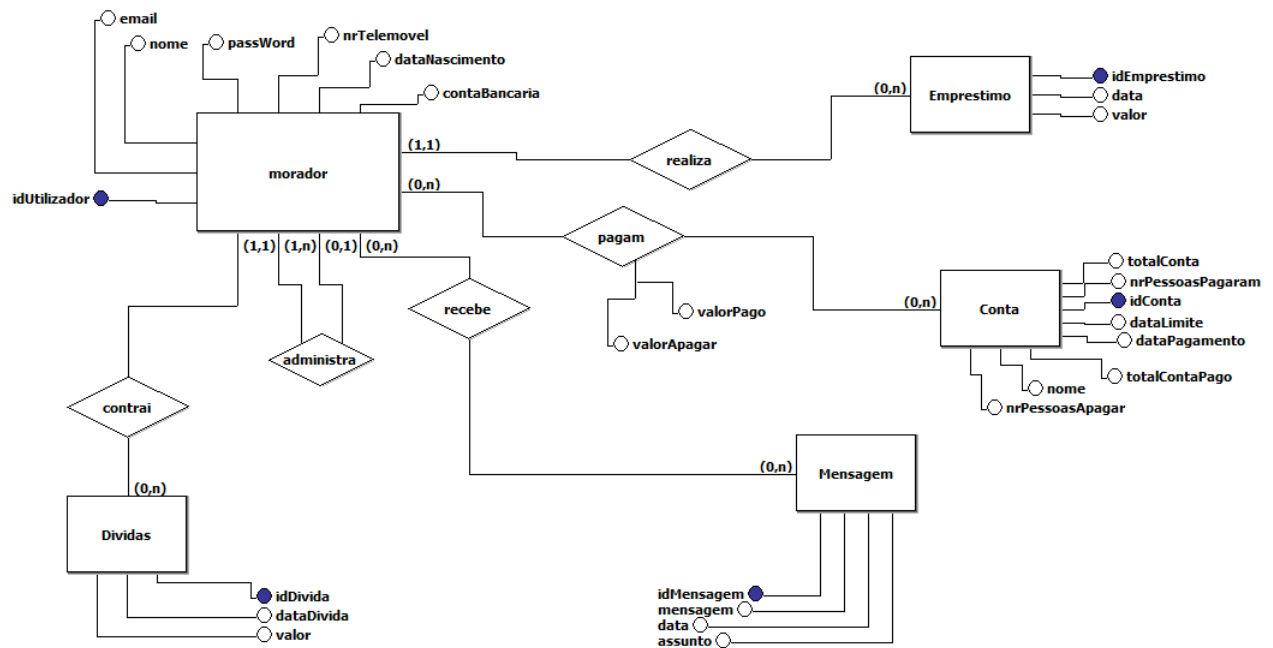


Figura 2.1: Modelo Concetual

Terminada a modelação concetual, procedemos à fase de modelação lógica. Esta consiste na tradução do modelo concetual para o lógico capaz de representar os requisitos definidos, assim como a validação do mesmo. Este modelo está representado na figura abaixo:

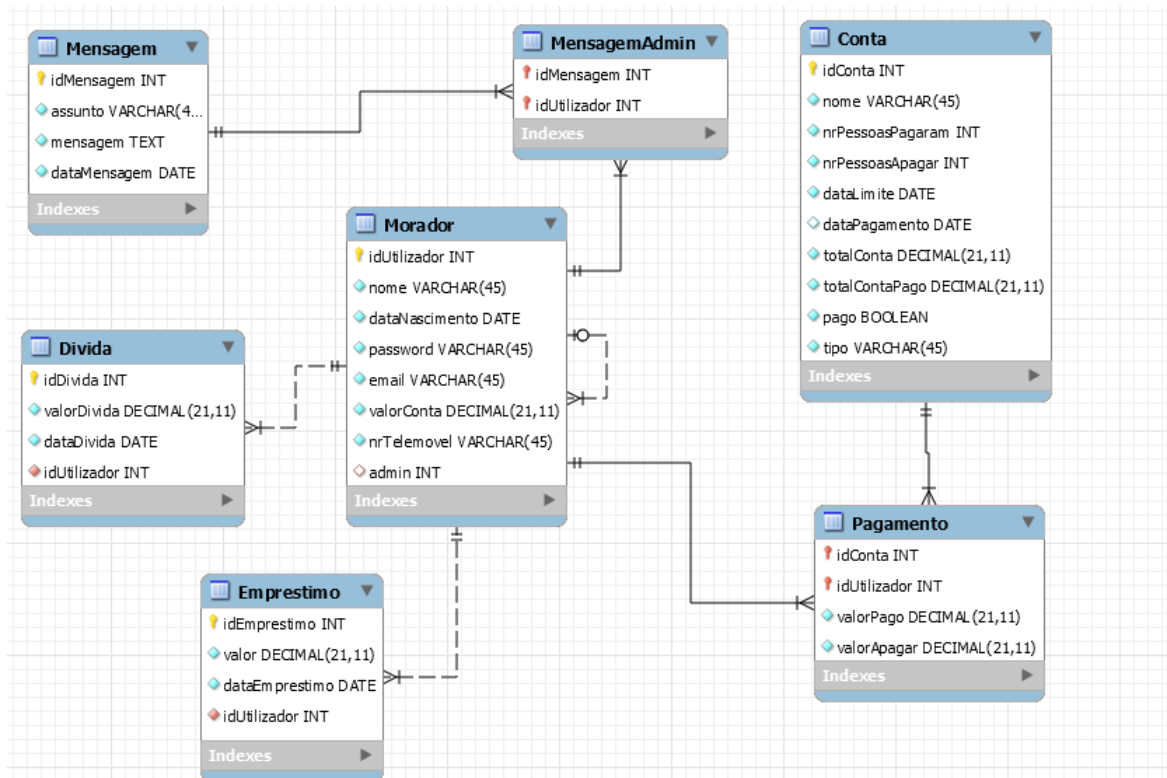


Figura 2.2: Modelo Lógico

[illegible]

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.

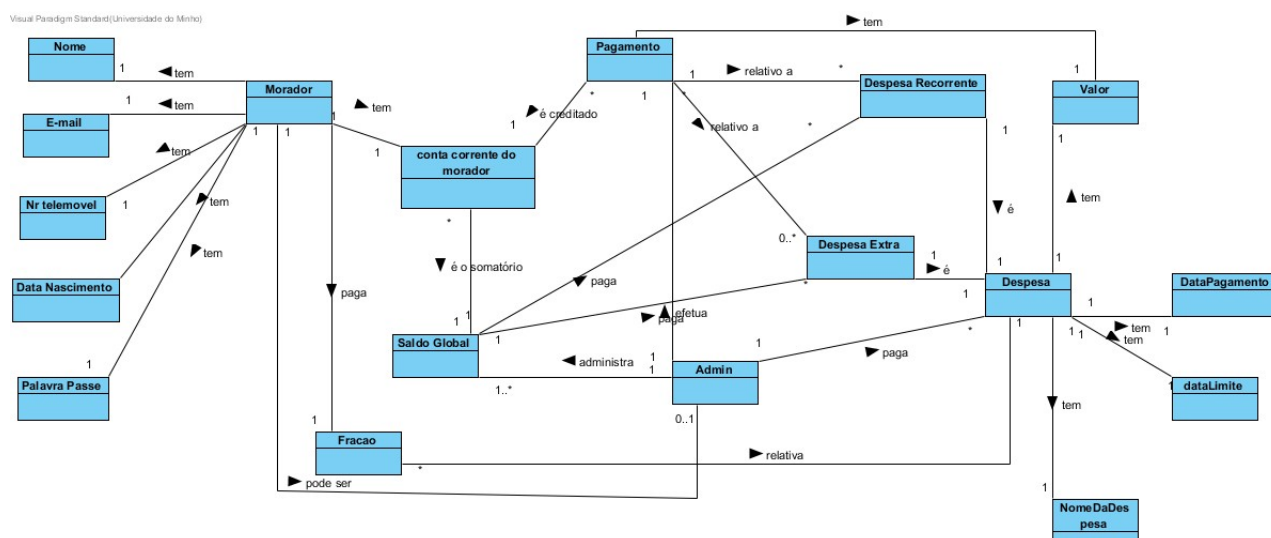


Figura 3.1: Modelo Dominio

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

Para efetuar login será necessário inserir o e-mail e a palavra passe.

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 Diagrama de Classes

A metodologia de implementação das boas práticas do UML no desenvolvimento de qualquer projecto, especialmente em projectos de media e pequena escala, para possíveis implementações de ideias da camada de negócio e de esquemas conceptuais, que podem ser validados e discutidos pelo grupo de desenvolvimento e analisados sem que seja necessário perder tempo com testes e builds. No caso específico do diagrama de classes aquilo que concluímos é que permite de uma forma simples e transparente partilhar e discutir como irá ser, ou pelos menos como se pretende que seja, os esquemas conceptuais do modelo de domínio e esqueleto da nossa implementação, ou seja, da estrutura básica das classes. O modelo abaixo apresentado é portanto uma tradução daquilo que pretendíamos com o modelo de domínio, mas nesse caso, com maior detalhe técnico relativos à linguagem de programação escolhida, i.e. o Java. Temos portanto o seguinte diagrama de classes:

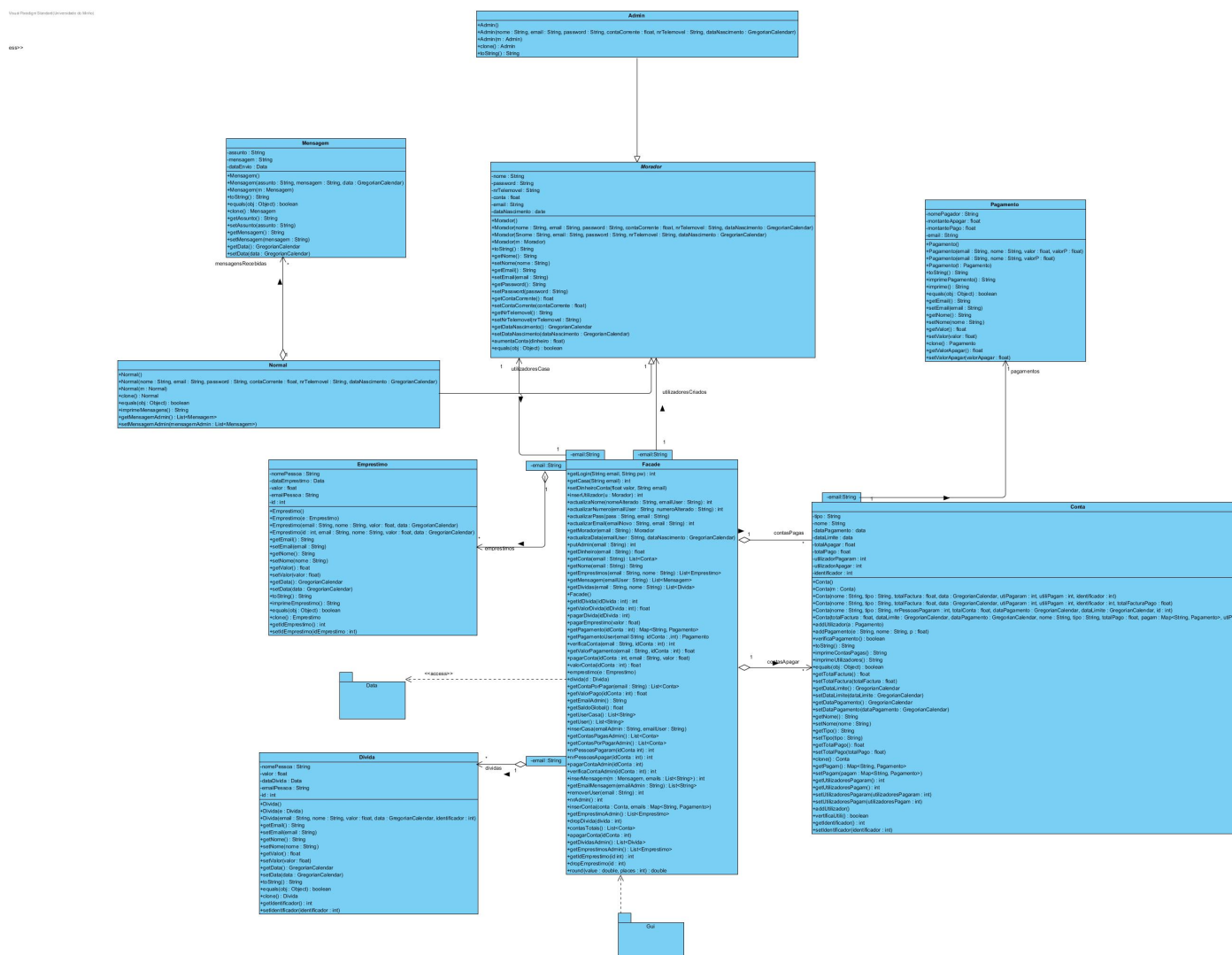


Figura 3.2: Diagrama de Classes

3.3 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.

3.3.1 Diagrama de Use Case

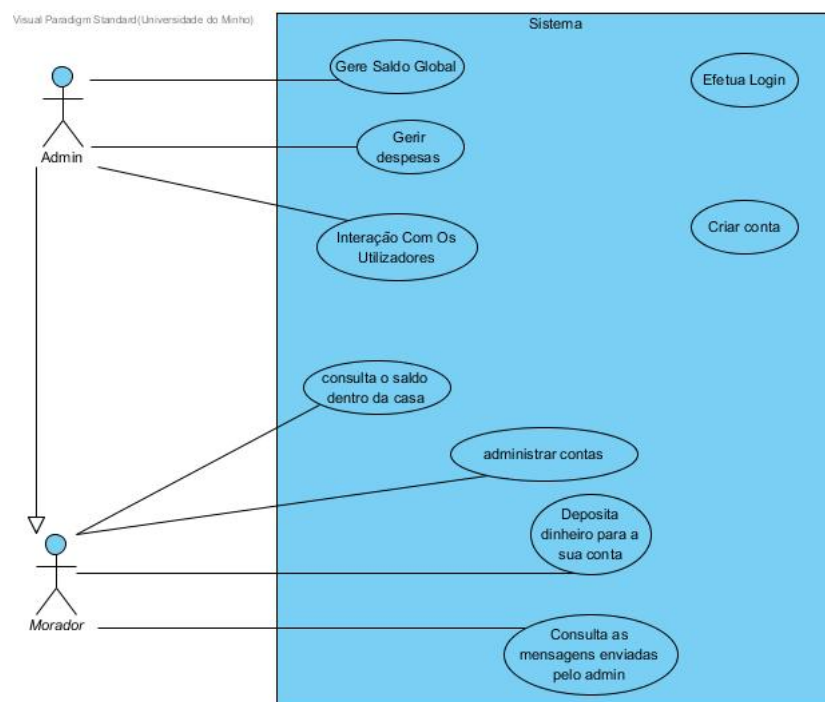


Figura 3.3: Modelo Use Case

Especificação: Gere Saldo Global

Esta interação pressupõe que o administrador tenha o login previamente efetuado e desta forma, a visualização do saldo global ao Admin será simples e clara. É então desta forma que o Admin tem acesso ao valor que consta no saldo global, onde esse valor é simplesmente dado a conhecer pelo sistema. O saldo global a que nos referimos é o resultado do somatório de todas as contas correntes de cada morador do apartamento em questão. Este Use Case apenas dá a conhecer ao Admin esse valor.

O use case em formato tabular do Gere saldo Global e o seu respectivo diagrama de sequência é o seguinte:

Date	2016		
Brief Description			
Preconditions	Admin tem login efetuado		
Post-conditions			
Flow of Events		Actor Input	System Response
	1		o sistema apresenta o dinheiro do saldo
	2	o admin vê o dinheiro	
	3		

Figura 3.4: Especificação do Use Case: Gere Saldo Global

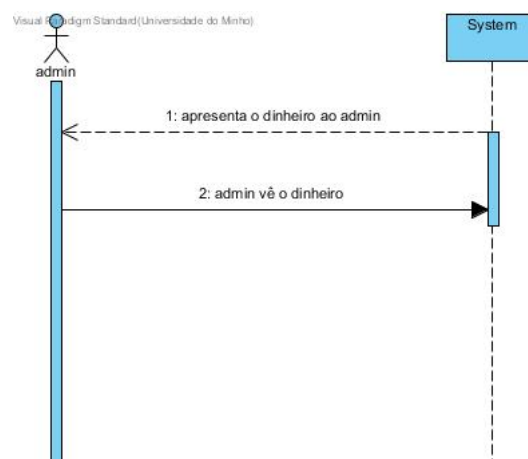


Figura 3.5: Diagrama de Sequência: Gere Saldo Global

Especificação: Efetua Login

Cada morador tem que efetuar Login na aplicação, para tal, é necessário que o indivíduo tenha conta previamente criada. Como em qualquer aplicação, este passo é fundamental e não poderia deixar de o ser no nosso sistema. Para se efetuar o registo do login é solicitado pelo sistema o endereço de email do utilizador em questão para sua validação. Após introduzir o email o sistema requer também o preenchimento da respetiva password. Optamos por colocar um endereço de email ao invés de um username ou um número de telemóvel porque assim poderá existir a possibilidade de futuramente serem enviados para o email dos utilizadores da aplicação, notificações relativas às despesas e ou pagamentos.

Preconditions	Ter criado conta	
Post-conditions	Ter efectuado o login	
Flow of Events		Actor Input
	1	
		System Response
		o sistema pede que introduza o email
	2	o utilizador introduz o email
	3	
		o sistema valida o email
Exceção 1 [email invalido] (passo 3)	4	
		o sistema pede que introduza a palavra passe
	5	o utilizador introduz a palavra passe
	6	
		o sistema valida a palavra passe
Exceção 2 [palavra passe incorreta] (passo 6)	7	
		login efectuado com sucesso

Figura 3.6: Especificação do Use Case: Efetua Login

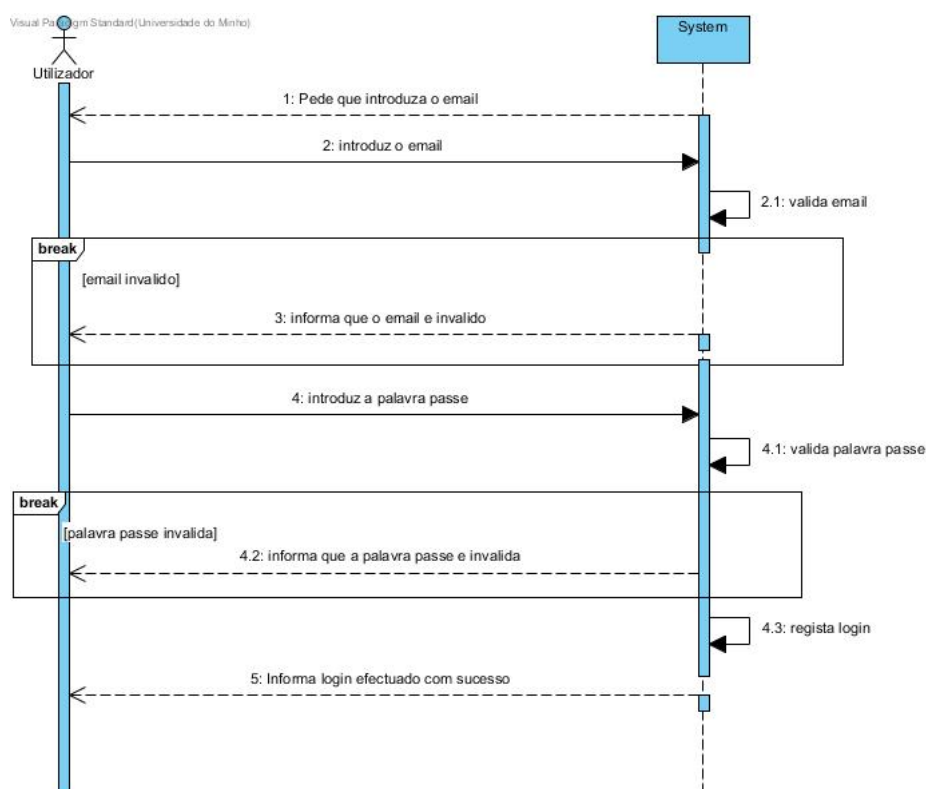


Figura 3.7: Diagrama de Sequência: Efetua Login

Especificação: Criar Conta

Esta interação é uma interação bastante vulgar no sentido em que em praticamente todas as aplicações o ato de “criar conta” é quase obrigatório. No nosso sistema este passo é fundamental para os utilizadores poderem utilizar a nossa aplicação. Para um utilizador criar conta tem que mencionar determinados dados ao sistema e este tem aceitar estes mesmos dados. Esses dados são simples e optamos por apenas solicitar o nome, data de nascimento, e-mail, número de telemóvel e password.

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
Excecção 1 (passo 7) [E-mail inválido]		Actor Input	System Response
	1		sistema informa que mail é inválido
Excecção 2 (passo 10) [password não cumpra requisitos]		Actor Input	System Response
	1		informa que password não cumpre requisitos minimos

Figura 3.8: Especificação do Use Case: Criar Conta

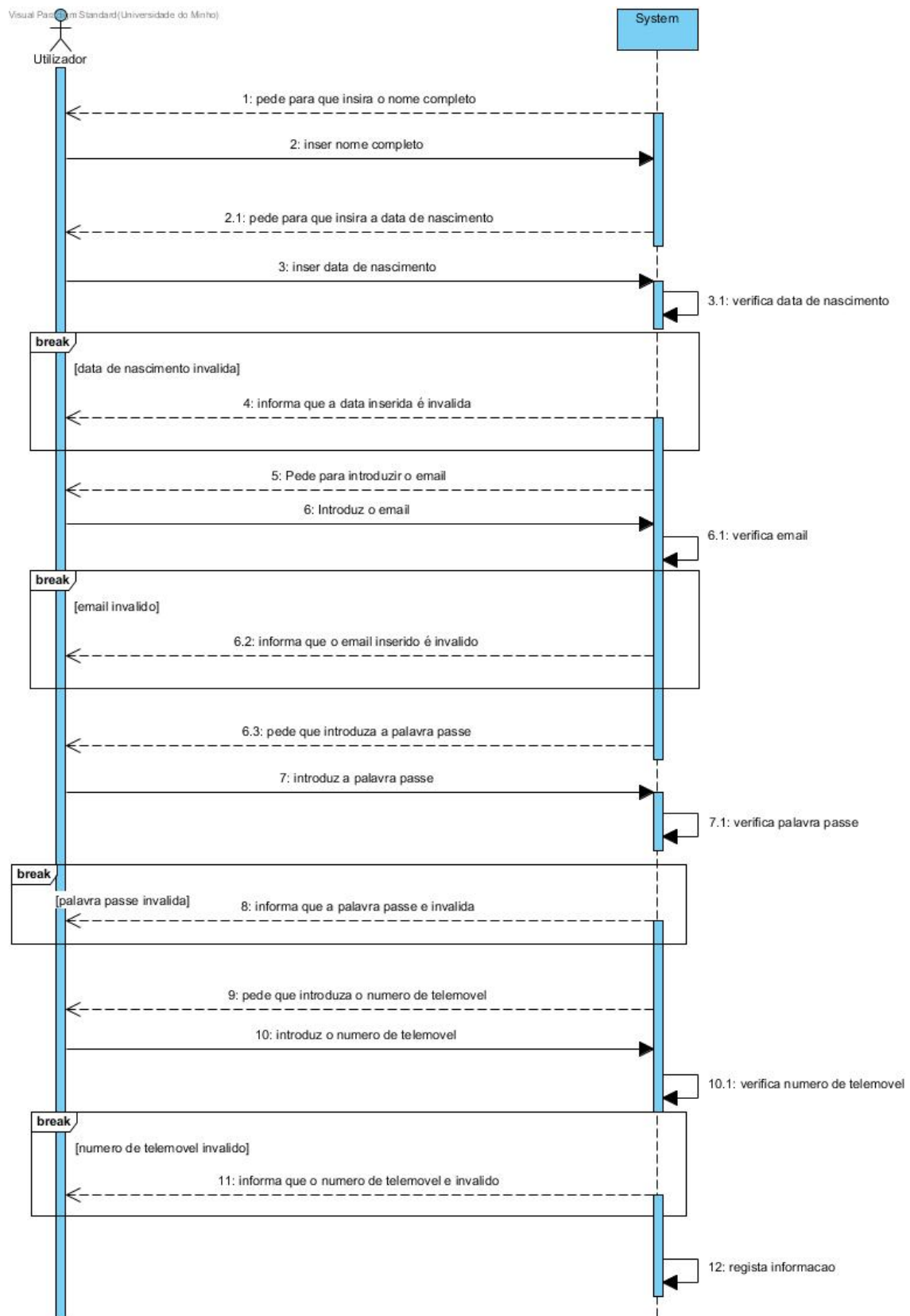


Figura 3.9: Diagrama de Sequência: Criar Conta

Especificação: Consulta Saldo dentro de casa

Este use case apenas é útil para a consulta do saldo dentro do apartamento, ou seja, existe a possibilidade de consultar o valor do saldo que consta no apartamento. Cada utilizador que tenta aceder a este saldo, tem que ter o login feito.

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 3.10: Especificação do Use Case: Consulta Saldo dentro de Casa

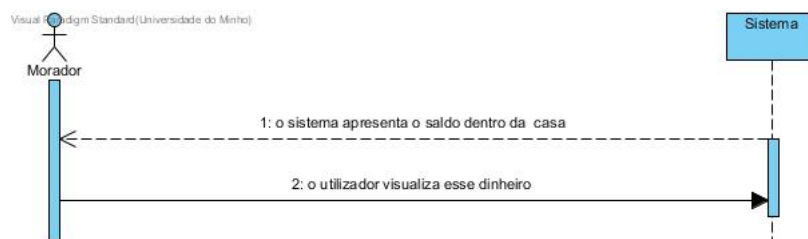


Figura 3.11: Diagrama de Sequência: Consulta saldo de casa

Especificação: Deposita Dinheiro para a sua Conta

Esta especificação foi feita a pensar no depósito da conta corrente de cada morador/utilizador da aplicação, assim como, a atualização da conta corrente. Basicamente, o sistema solicita que o utilizador adicione determinada quantia na conta corrente e após este procedimento ser efetuado com sucesso a conta corrente surge atualizada. Este processo requer que o utilizador tenha o login efetuado, pois só desta forma é possível aceder a estas especificações.

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

Figura 3.12: Especificação do Use Case: Deposita Dinheiro para a sua conta

Especificação: Consulta Mensagens enviadas pelo Administrador

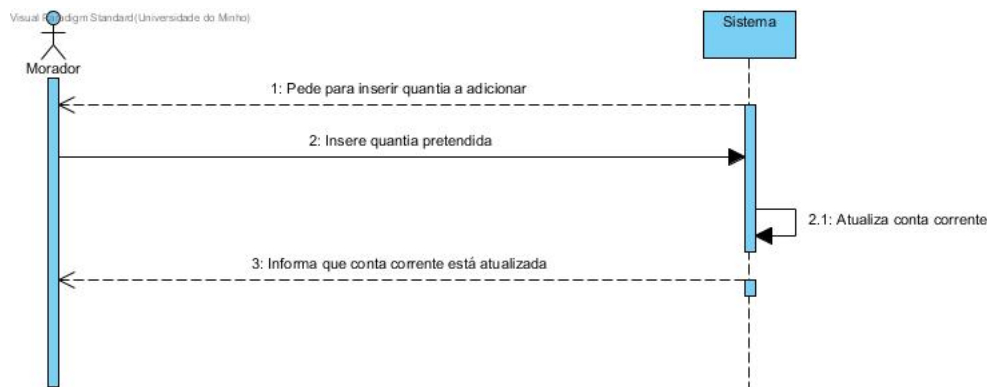


Figura 3.13: Diagrama de Sequência: Deposita Dinheiro na Conta

Date	23/out/2016 21:54:07	
Brief Description		
Preconditions	Ter o login efectuado	
Post-conditions	Consegue consulta as mensagens recebidas	
Flow of Events		
	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]		
	1	O sistema informa que não existem mensagens para apresentar

Figura 3.14: Especificação do Use Case: Consulta Mensagens enviadas pelo Administrador

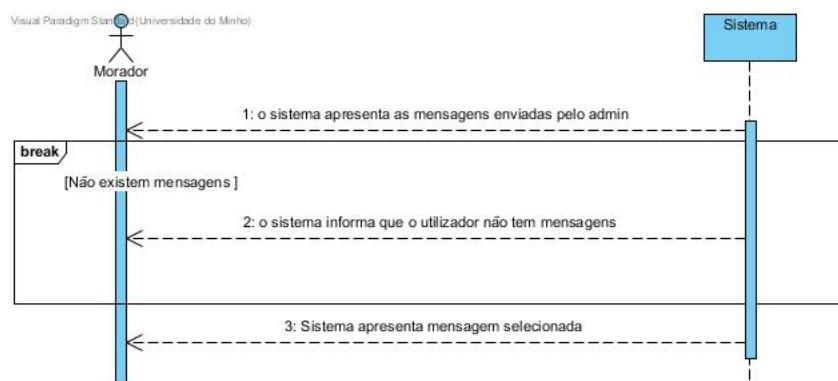


Figura 3.15: Diagrama de Sequência: Consulta Mensagens

3.3.2 Subdiagrama Gerir Despesas

Especificação: Efetua pagamento da despesa

De forma a organizar a nossa aplicação apresentamos as despesas que estão por pagar por lista de onde são visualizadas as despesas que estão por pagar e assim fazendo um clique é possível aceder a uma determinada fatura seleccionada pelo utilizador. A despesa em falta é automaticamente descontada ao saldo global do apartamento em causa. Deste modo, sabemos sempre se existem faturas por pagar e o saldo é atualizado no imediato. Claro está, que é necessário que o valor do saldo seja sempre igual ou superior aos valores indicados nas faturas.

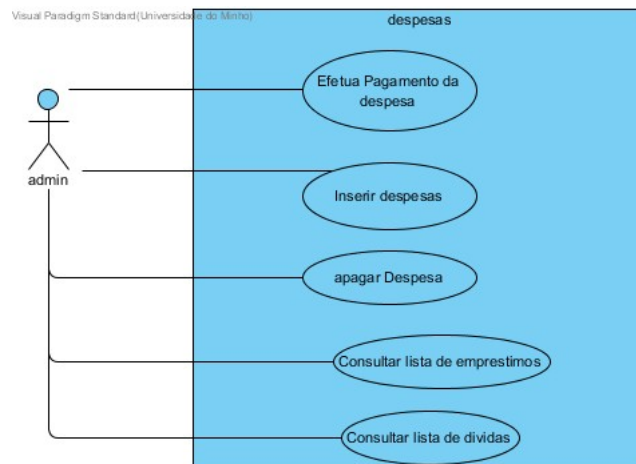


Figura 3.16: Sub-Diagrama Gerir Despesas

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		
Excecao (passo 1) [Nao existe faturas a pagar]		

Figura 3.17: Especificação do Use Case:Efetua Pagamento da despesa

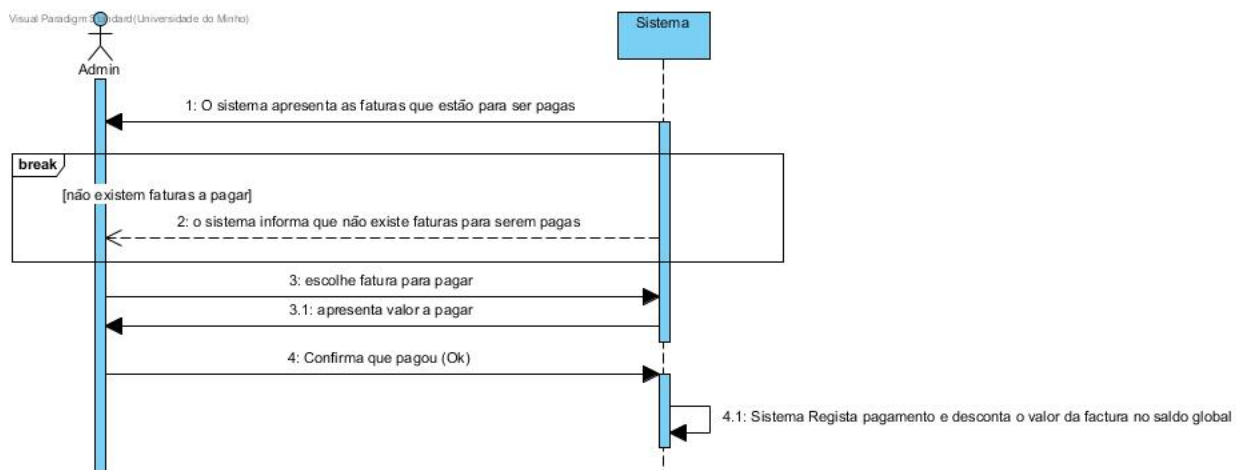


Figura 3.18: Diagrama de Sequência: Administrador efetua pagamento da despesa

Especificação: Inserir Despesa

Cabe ao administrador fazer as inserções das despesas no sistema. Para que cada inserção seja feita com sucesso têm que ser bem introduzidas e aceites pelo sistema, colocando a data,

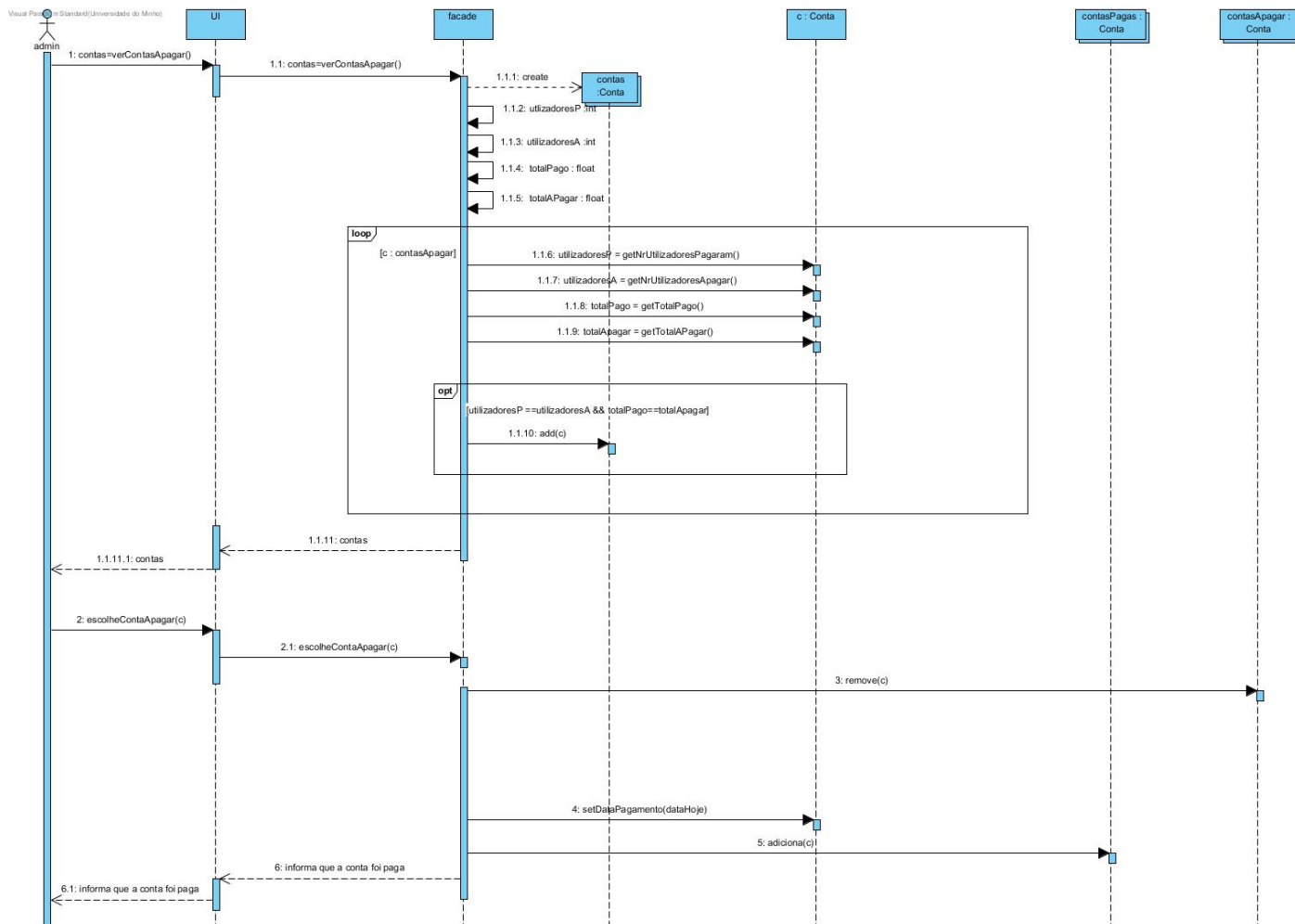


Figura 3.19: Diagrama de Iteração: Administrador efetua pagamento

o nome e os valores de cada fatura válidos. É também tarefa do administrador decidir que percentagem cada morador registado tem que pagar de cada fatura/despesa.

Preconditions	o admin está autenticado	
Post-conditions	despesa inserida	
Flow of Events		Actor Input
	1	
	2	Inserir nome da fatura
	3	
	4	o utilizador introduz a data
	5	
	6	
	7	Inserir valor da fatura
	8	
	9	
	10	o utilizador escolhe o tipo normal
	11	
	12	
	13	
Excecao(passo 4 [O sistema não valida a data])		Actor Input
	1	
Exceção (passo 7) [o sistema não valida o valor introduzido]		Actor Input
	1	
Comp.Alternativo (passo 2) [Inser despesa extra]		Actor Input
	1	
	2	o admin escolhe os utilizadores
	3	
	4	o admin introduz a percentagem que cada um tem de pagar
	5	

Figura 3.20: Especificação do Use Case: Inserir Despesas

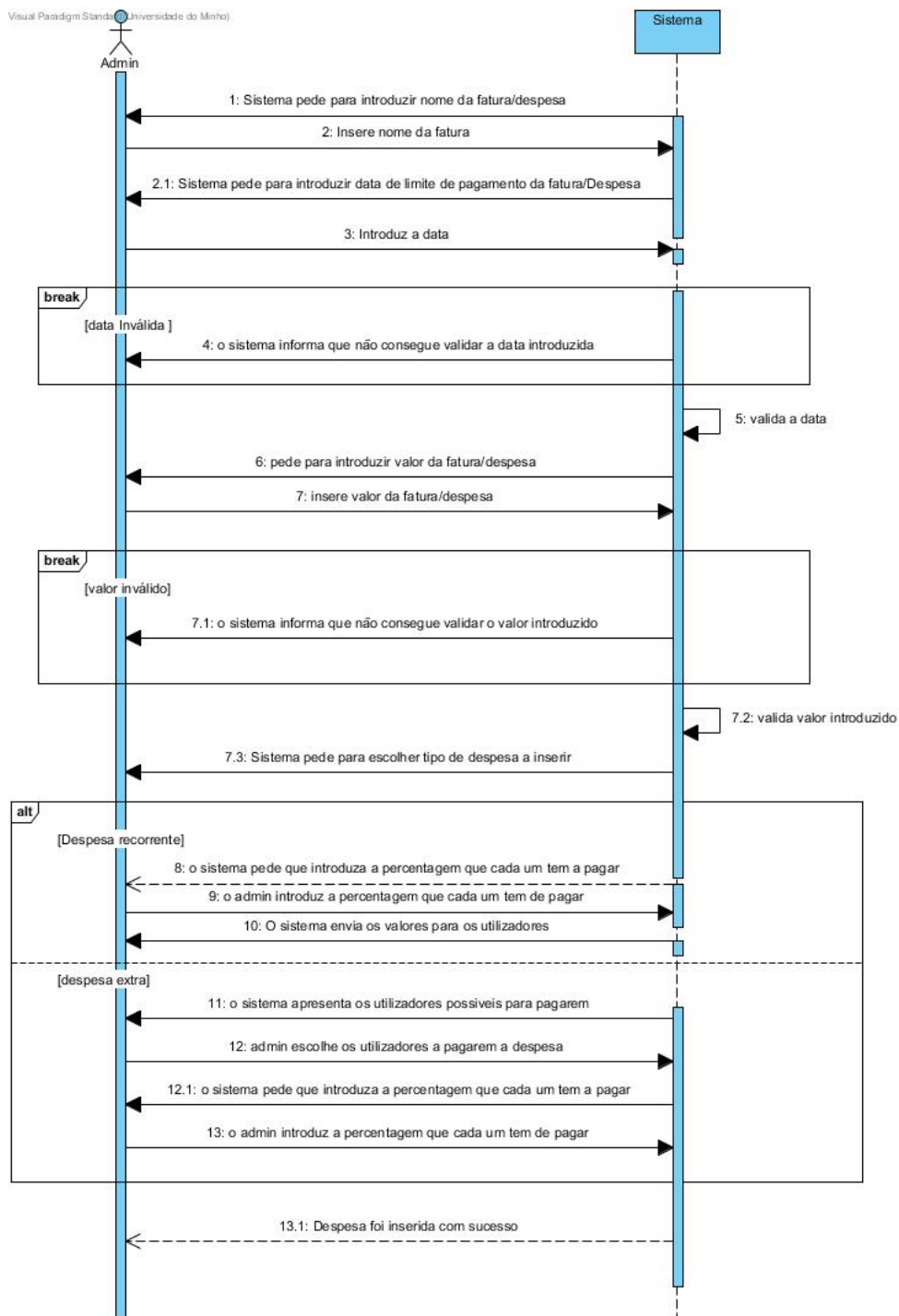


Figura 3.21: Diagrama de Sequência: Inserir Despesa

Especificação: Apagar Despesa

O administrador tanto insere as despesas no sistema como as elimina. Temos apenas que ter em consideração que é condição necessária que o admin tenha o login efetuado e a partir daqui o sistema apresenta a listas das despesas por pagar o administrador apenas selecciona aquela que efetivamente pretende que seja eliminada.

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
Execução (passo 1) [Não existem despesas por pagar]		Actor Input
	1	o sistema informa que não existe despesa por pagar

Figura 3.22: Especificação do Use Case: Apagar Despesa

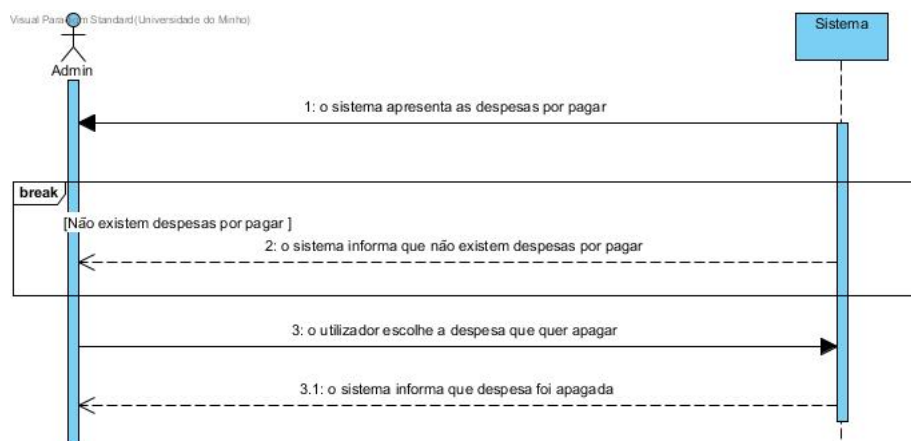


Figura 3.23: Diagrama de Sequência: Apagar Despesa

3.3.3 Subdiagrama Administrar Contas

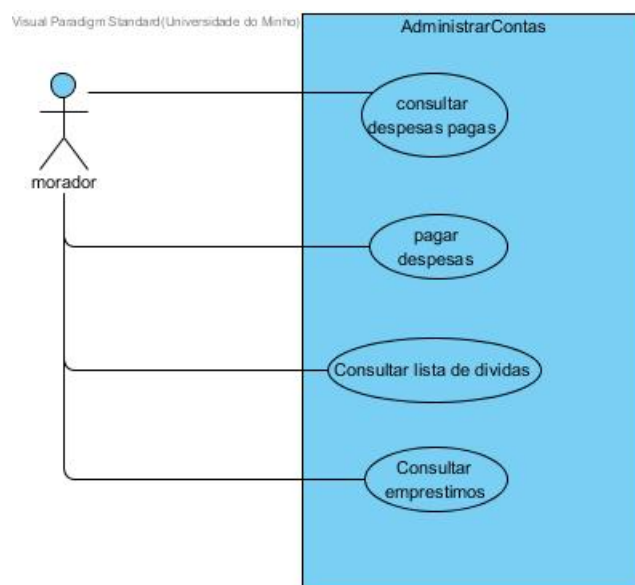


Figura 3.24: Subdiagrama: Administrar Contas

Especificação: Consultar despesas pagas

Esta é uma consulta feita pelo administrador. Fica possível a consulta de cada uma das despesas que já foram pagas. Esta interação apenas serve para consultar informação. O sistema apresenta todos os detalhes da fatura previamente seleccionada pelo admin. Como o próprio nome do subdiagrama indica, trata-se de administrar contas.

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

Figura 3.25: Especificação do Use Case: Consultar Despesas Pagas

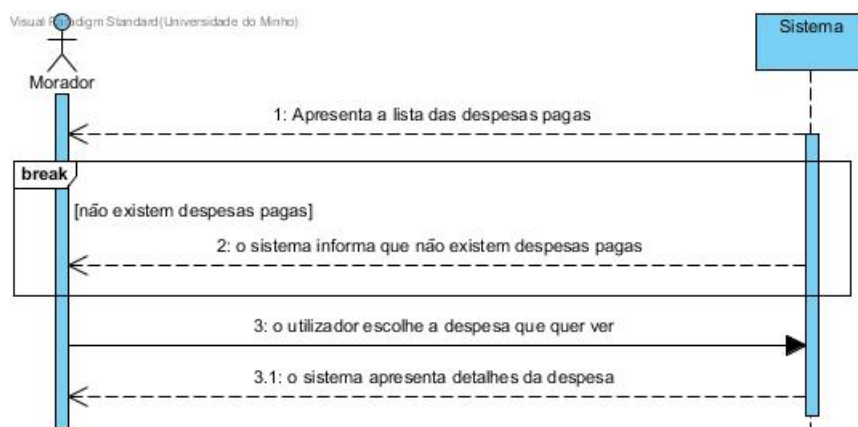


Figura 3.26: Diagrama de Sequência: Consultar Despesas pagas

Especificação: Pagar Despesas

É nesta especificação que está descrito como é que todos os pagamentos da nossa aplicação são efetivamente pagos. Cada utilizador, previamente registado e com login feito selecciona a conta que pretende pagar e procede ao pagamento em questão. O sistema após validar este pagamento atualiza a informação e coloca-a visível.

Preconditions	o utilizador tem o login efectuado		
Post-conditions			
Flow of Events		Actor Input	System Response
	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
	6		o sistema regista a informacao
Comp.Alternativo(passo4)[utilizador, paga a menos]		Actor Input	System Response
	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
Execcao(passo 5)[Dinheiro Insuficiente]		Actor Input	System Response
	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]		Actor Input	System Response
	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
Execcao(passo 5)[Valor introduzido a mais]		Actor Input	System Response
	1		o sistema informa que o valor introduzido é demasiado para pagar a conta

Figura 3.27: Especificação do Use Case: Pagar Despesas

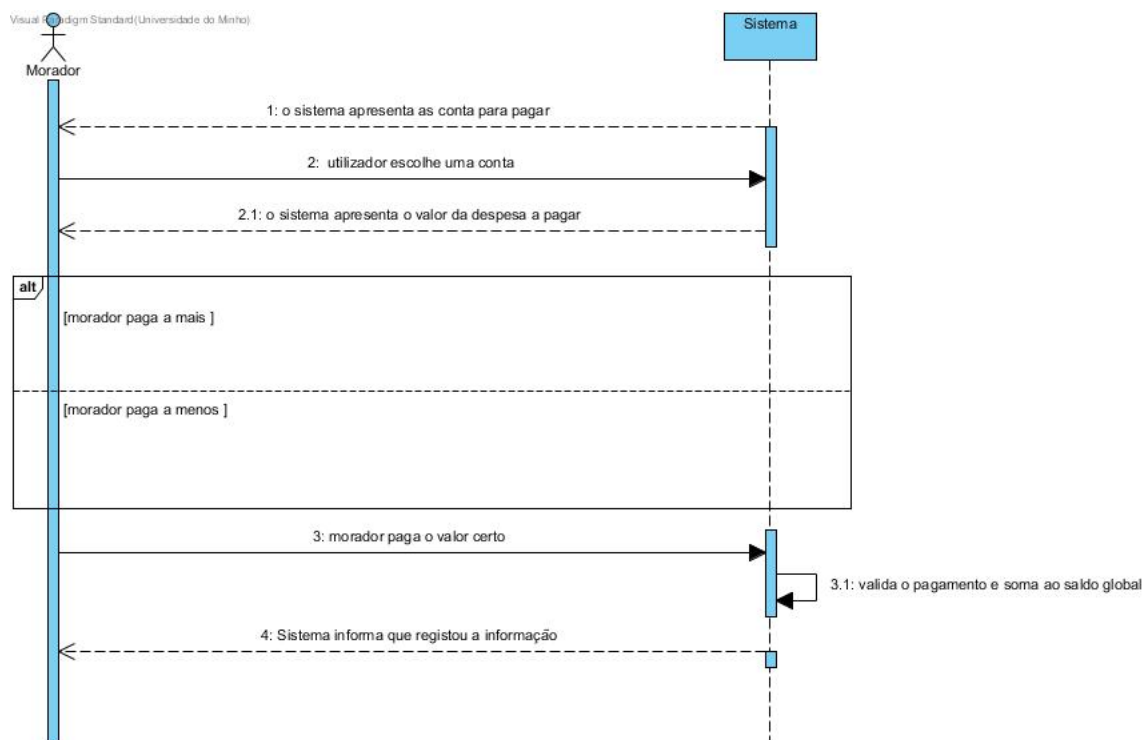


Figura 3.28: Diagrama de Sequência: Utilizador paga despesa

Especificação: Consultar Empréstimos

Nesta especificação o utilizador tem a vantagem de aceder à lista de empréstimos, ou seja, consegue através da aplicação visualizar os valores que emprestou a outros moradores do apartamento, previamente registados na aplicação.

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
Excecao(passo 1) [não fez empréstimos]		Actor Input
	1	o sistema informa que o utilizador não fez empréstimos

Figura 3.30: Especificação do Use Case: Consultar Empréstimos

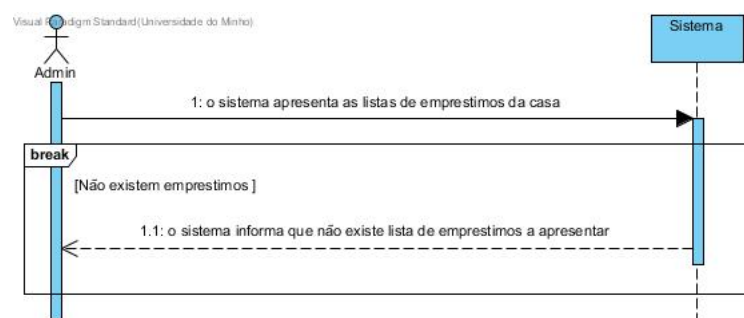


Figura 3.31: Diagrama de Sequência: Consulta lista de empréstimos

Especificação: Consultar Lista de Dívidas

Paralelamente ao que foi referido na especificação do use case de “consultar lista de empréstimos”, nesta secção é possível ter acesso à lista de dívidas. Por dívidas entende-se dinheiro que o morador deve à casa, neste caso a alguém que colocou o dinheiro por ele. A consulta de dívidas permite que cada utilizador tenha acesso à lista de dívidas e valores por pagar, desta forma tem sempre as contas com os valores exatos em dívida sem ter que fazer cálculos e arredondamentos.

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

Figura 3.32: Especificação do Use Case: Consultar lista de dividas

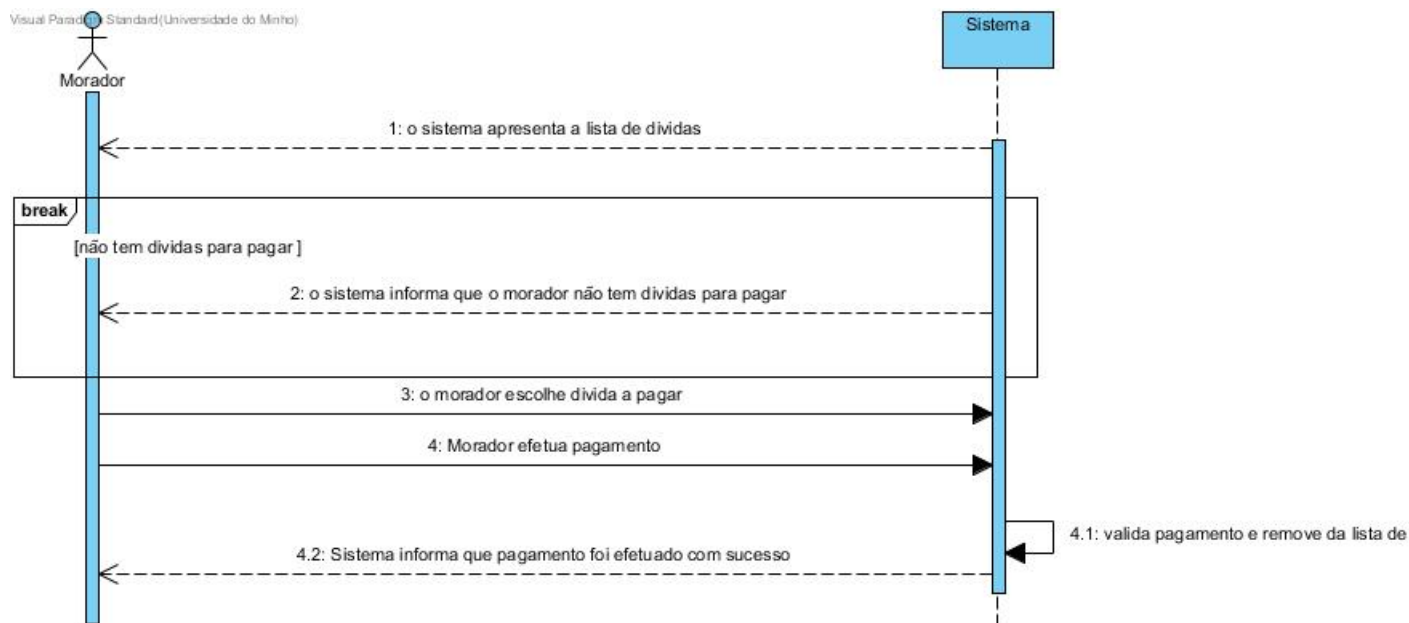


Figura 3.33: Diagrama de Sequência: Consulta lista de dividas

3.3.4 Subdiagrama Interação com os Utilizadores

A interação com os utilizadores da aplicação torna a experiência mais interessante e motivadora.

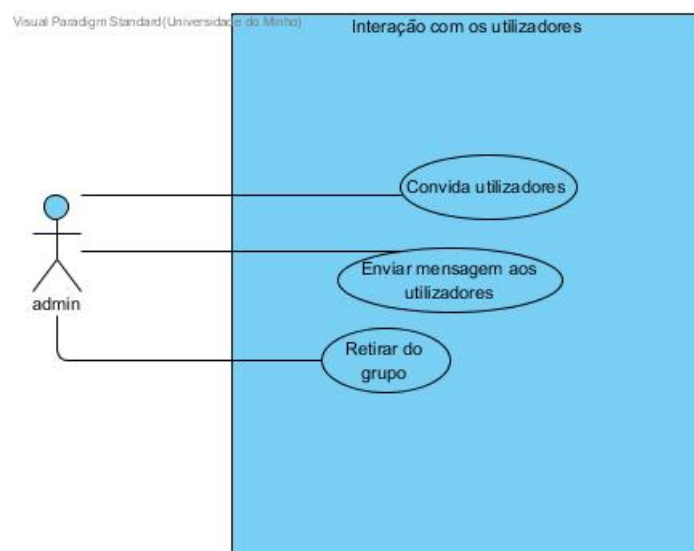


Figura 3.34: Subdiagrama: Interação com os utilizadores

Especificação: Convida Utilizadores

É através do email que um utilizador previamente registado convida outros a pertencerem ao “grupo” e a começarem a participar nas despesas do apartamento. O novo membro é notificado e de imediato pode proceder ao seu registo e efetuar login.

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

Figura 3.35: Especificação do Use Case: Convida Utilizadores

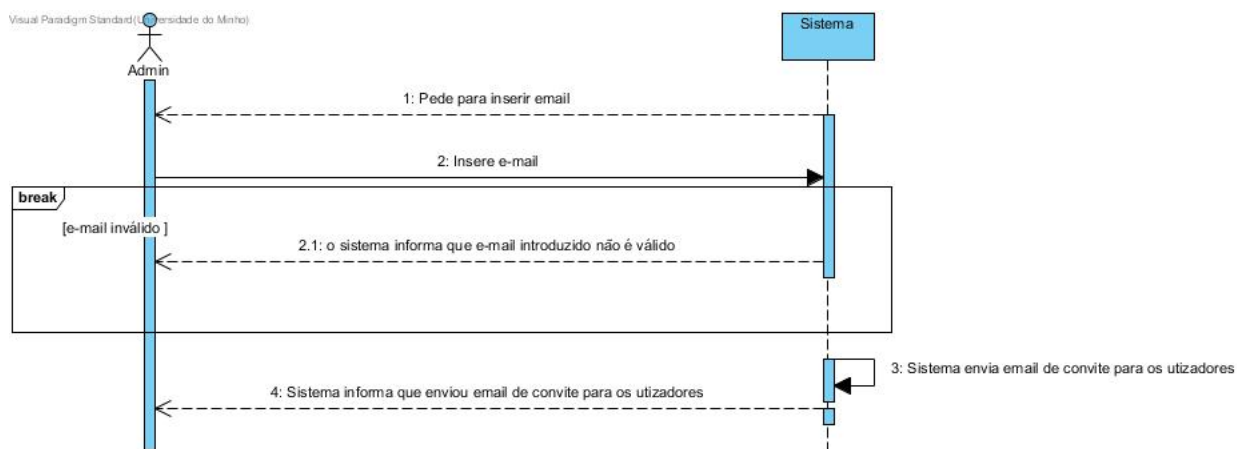


Figura 3.36: Diagrama de Sequência: Convida Utilizadores

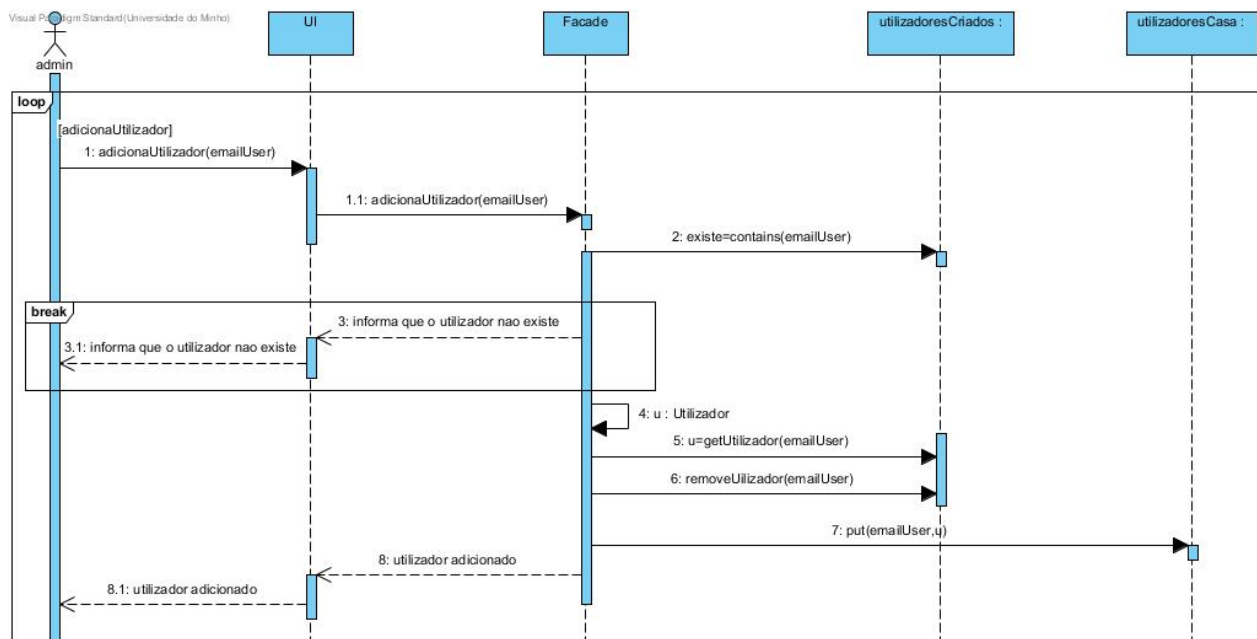


Figura 3.37: Diagrama de Iteração: Convida Utilizadores

Especificação: Envia Mensagens aos Utilizadores

O administrador da aplicação tem acesso à lista de utilizadores que estão aptos a receber mensagens. O mesmo escolhe a quem pretende enviar uma mensagem e envia. O sistema apenas confirma o envio. Este processo pode ser enviado aos utilizadores que o administrador quiser.

Date	1/nov/2016 23:28:15	
Brief Description		
Preconditions	Ter o login efectuado	
Post-conditions		
Flow of Events		Actor Input
		System Response
	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 3.38: Especificação do Use Case: Envia Mensagens aos Utilizadores

Especificação: Retirar do grupo

O administrador consegue remover da aplicação os membros que de uma forma ou de outra já não pertencem ao grupo. O sistema apresenta a lista dos moradores registados e apenas é seleccionado o membro a excluir.

Date	23/out/2016 19:45:13	
Brief Description		
Preconditions	o admin ter o login efectuado	
Post-conditions	o admin consegue remover o utilizador	
Flow of Events		Actor Input
		System Response
	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
Exceção [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 3.39: Especificação do Use Case: Retirar do grupo

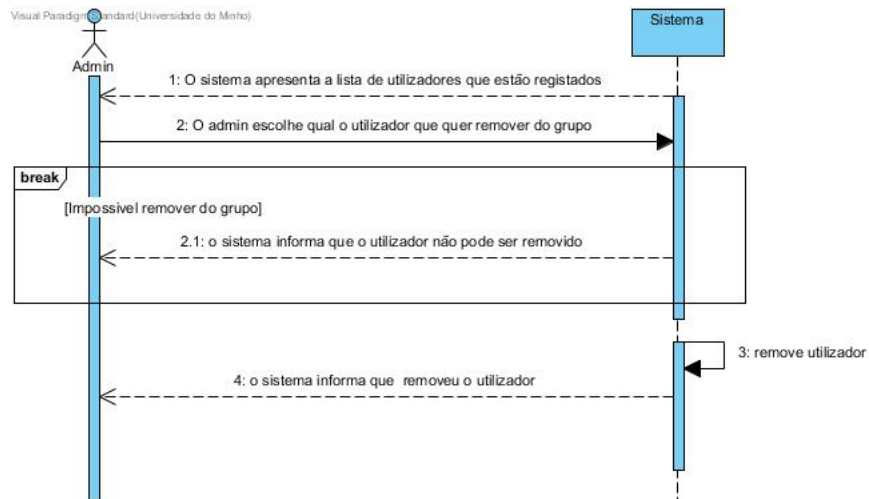


Figura 3.40: Diagrama de Sequência: Administrador Remove do Grupo

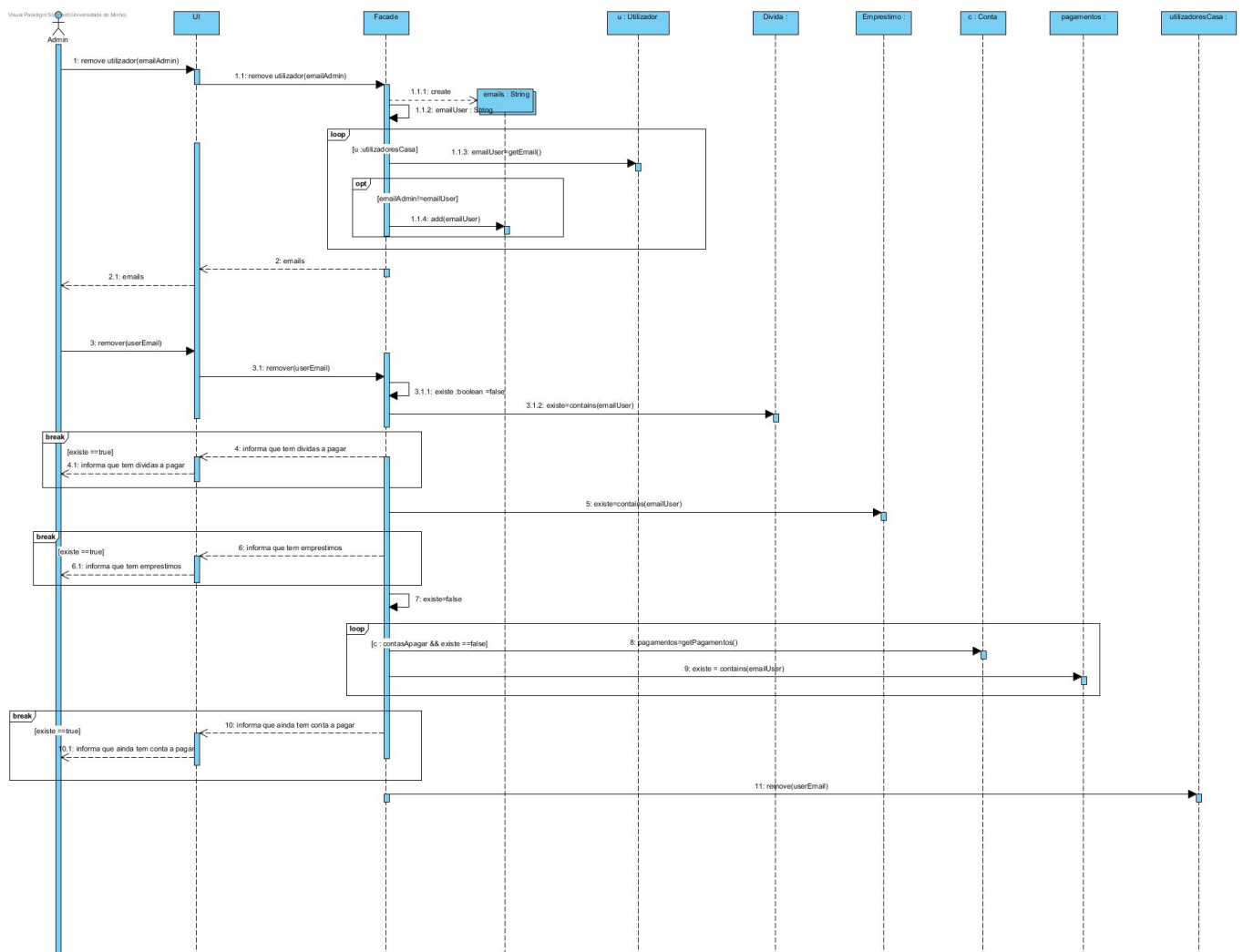


Figura 3.41: Diagrama de Iteração: Administrador Remove do Grupo

4. Implementação e Instalação do Sistema

4.1 Diagrama de Instalação

Os dois grandes subsistemas da aplicação desenvolvida são o computador do utilizador e o servidor de base de dados. A comunicação é estabelecida por TCP/IP. O programa é uma aplicação Java, e para a apresentação gráfica ao utilizador, é usado JavaSwing. Para ser possível a comunicação entre a aplicação em Java e a base de dados MySQL, recorre-se à interface JDBC¹.

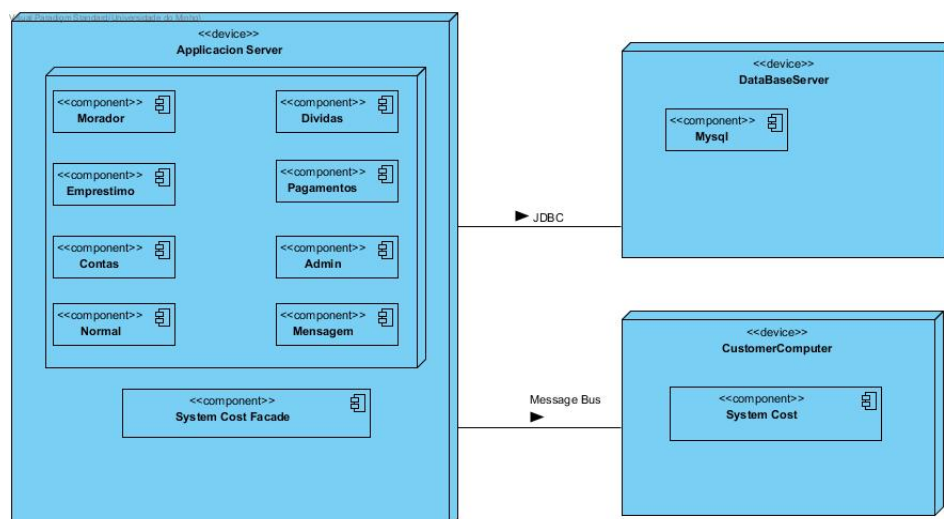


Figura 4.1: Diagrama de Instalação ou Deployment Diagram

¹Java Database Connectivity

Os três principais pacotes definidos são *Presentation*, que trata da apresentação gráfica, em *Java Swing*; *Business*, que contém toda a lógica de negócio; e *Data*, responsável pela ligação à base de dados. A camada de apresentação comunica com a camada de lógica de negócio a partir da classe SGD (Sistema de Gestão de Despesas), que desempenha a função de facade. Não existe qualquer tipo de comunicação feita diretamente entre a camada de apresentação e a camada de dados. A comunicação entre o package *Business* e a persistência de Dados é feita recorrendo às várias classes *DAO* implementadas no *Package Data*, que comunicam com a base de dados *MySQL*.

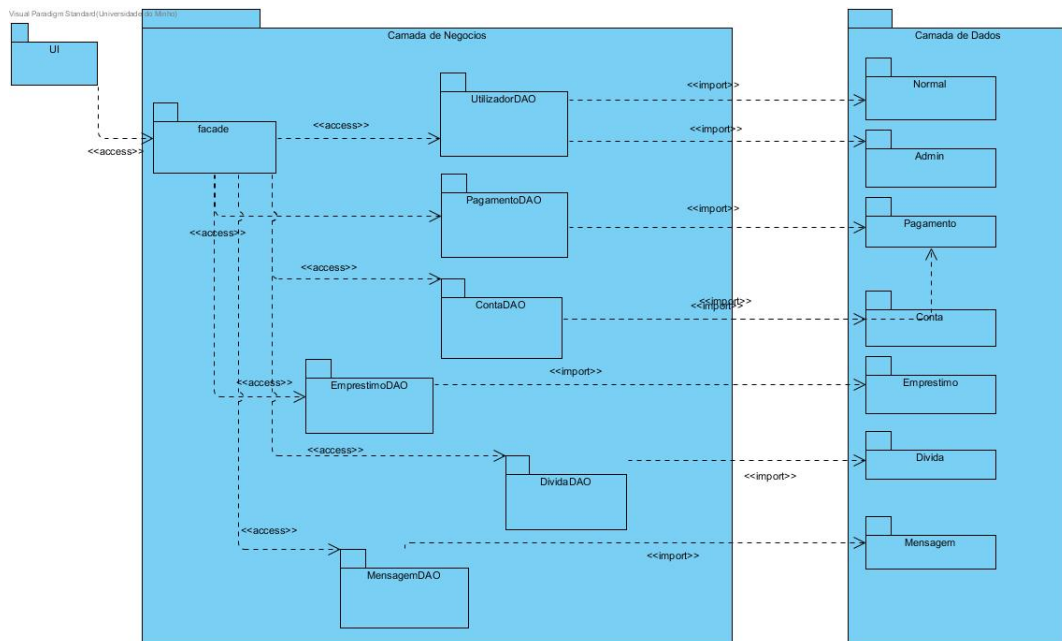


Figura 4.2: Diagrama de Pacotes

5. Interface do Sistema Gestão de Despesas - SGD

5.1 Máquinas de Estado

5.1.1 Efetuar Login

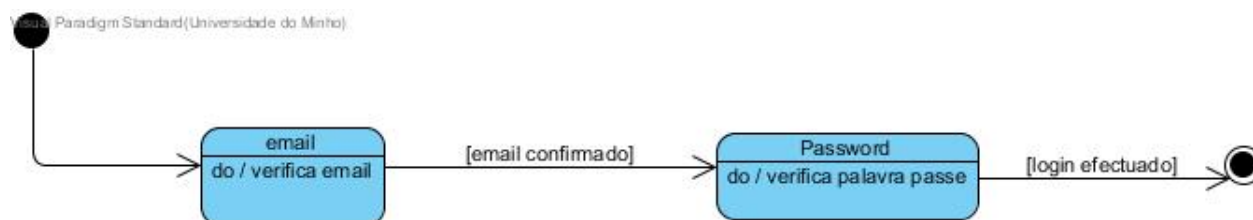


Figura 5.1: Máquina de Estados: Efetuar Login

5.1.2 Criar Conta

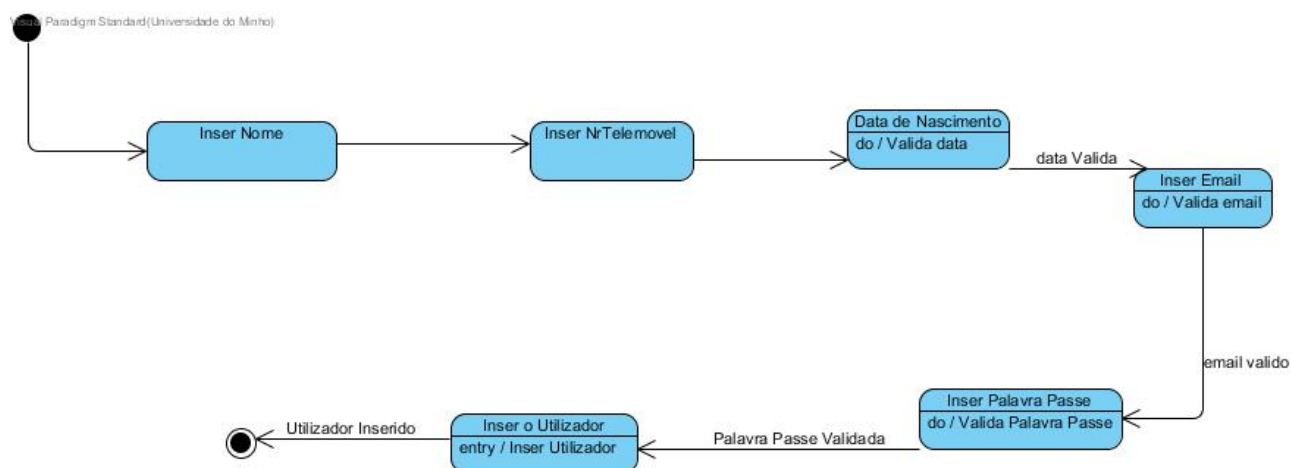


Figura 5.2: Máquina de Estados: Criar Conta

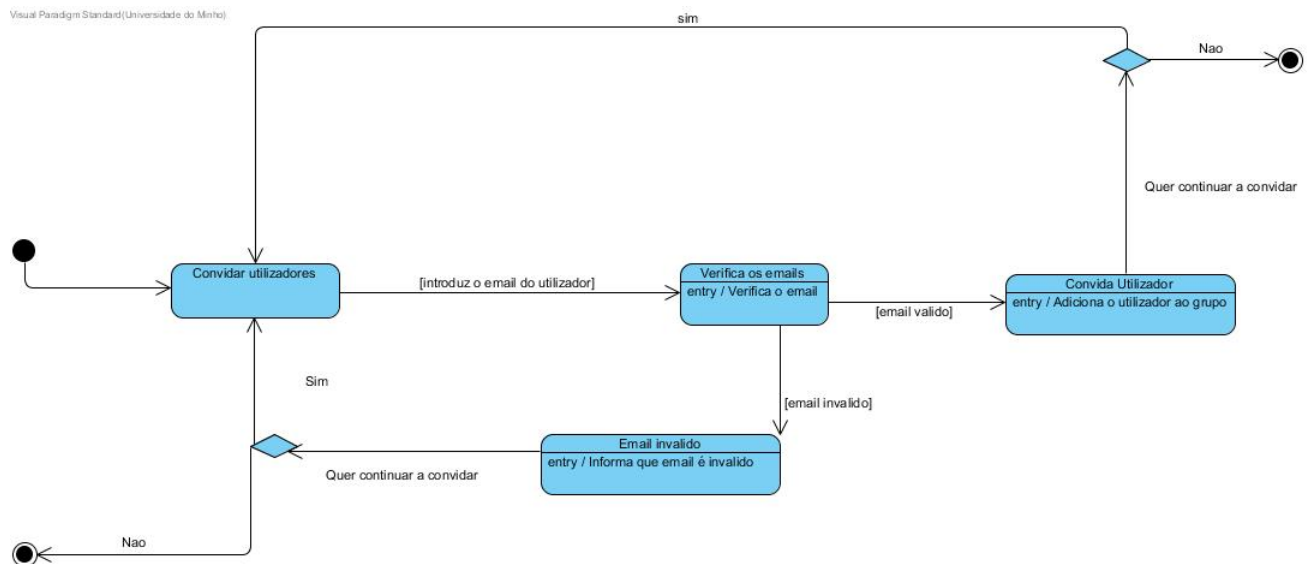


Figura 5.3: Máquina de Estados: Convidar Utilizadores

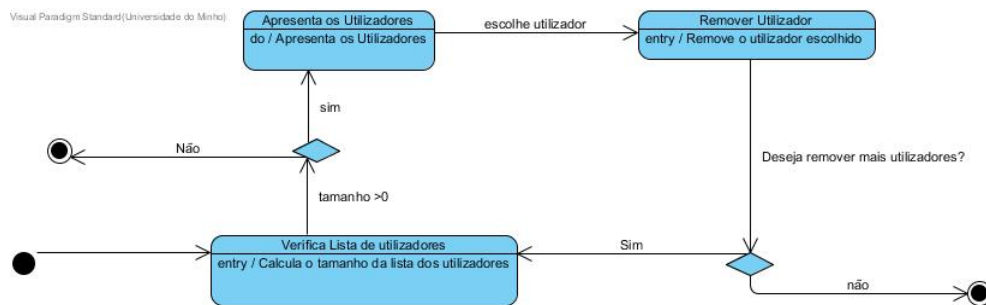


Figura 5.4: Máquina de Estados: Remover Utilizadores

5.1.3 Convidar Utilizadores

5.1.4 Remover Utilizadores

5.1.5 Inserir Fatura

5.1.6 Deposita Dinheiro

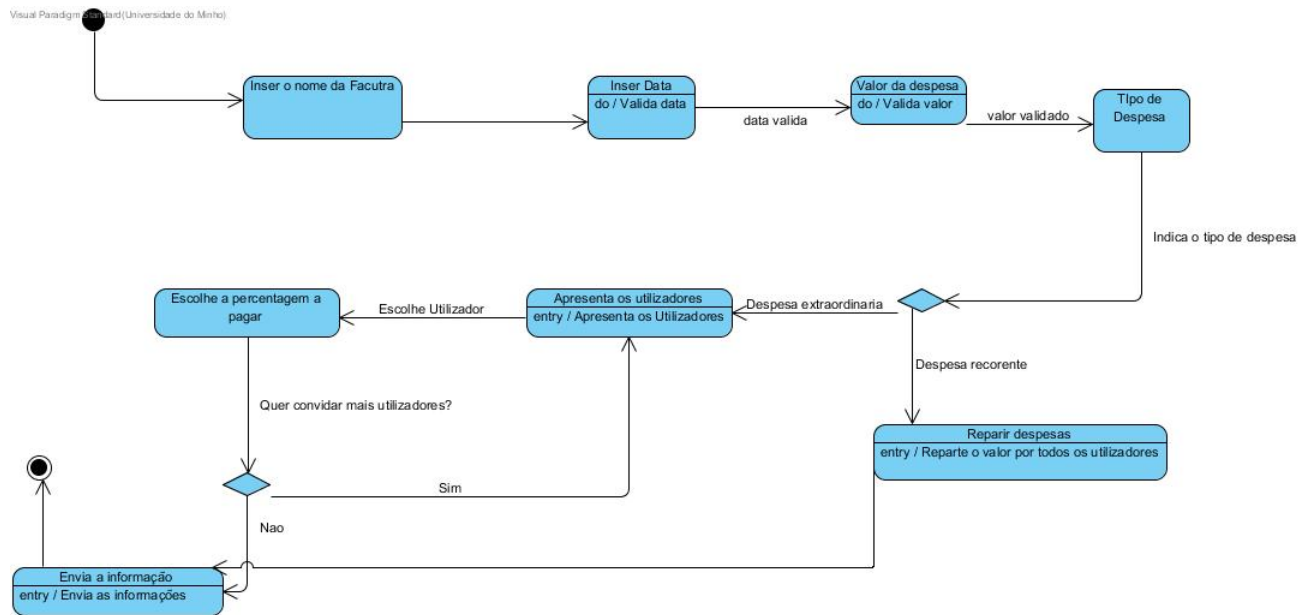


Figura 5.5: Máquina de Estados: Inserir Fatura

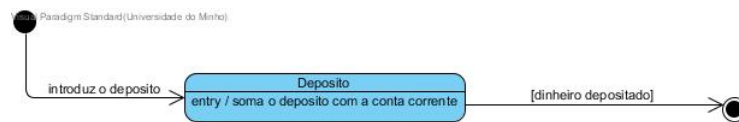


Figura 5.6: Máquina de Estados: Deposita Dinheiro

5.2 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 referimos, para o utilizador efetuar o login necessita de se registar previamente, fornecendo alguns dados que o identifiquem.

Figura 5.7: Criar nova Conta

Esta será a janela para os moradores e administrador efetuarem login na aplicação.



Figura 5.8: Login

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

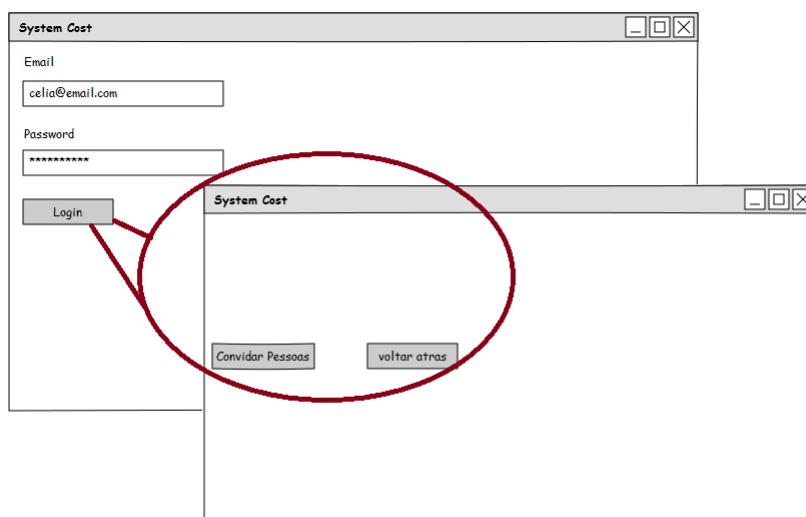


Figura 5.9: Login quando não existem grupos criados

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 apresentada uma lista com as pessoas já existentes e a possibilidade de convidar mais membros.

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.

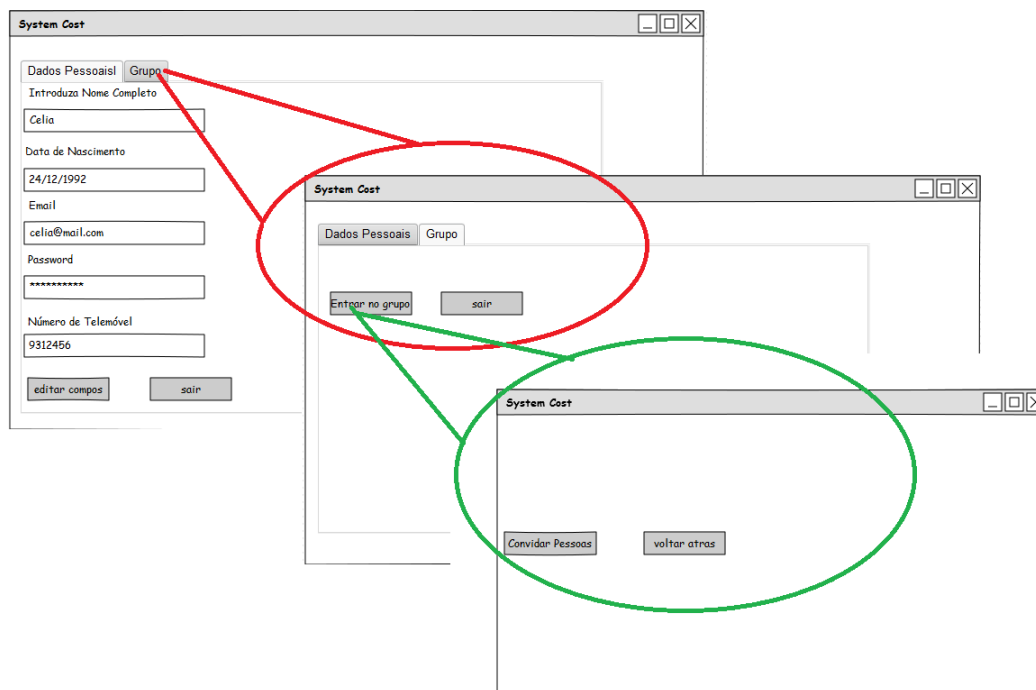


Figura 5.10: Visualização/Alteração dos dados

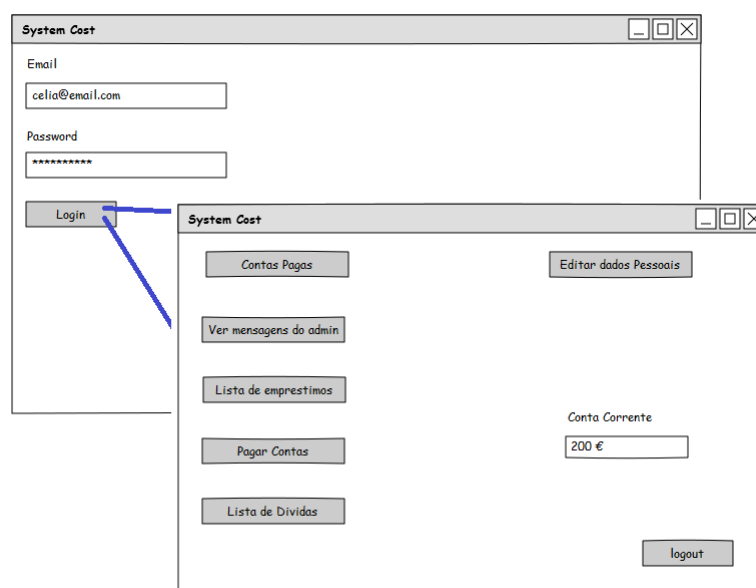


Figura 5.11: Login /página inicial morador

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

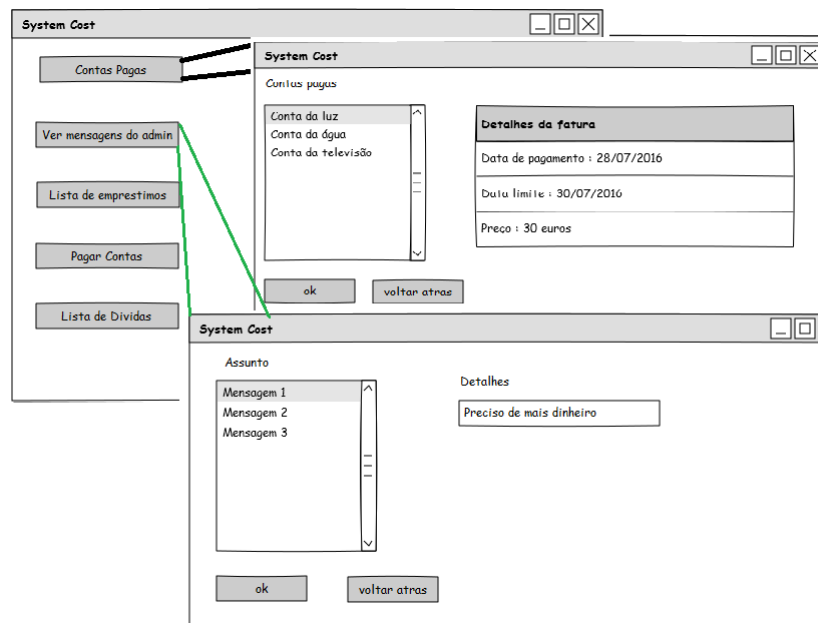


Figura 5.12: Opções do utilizador

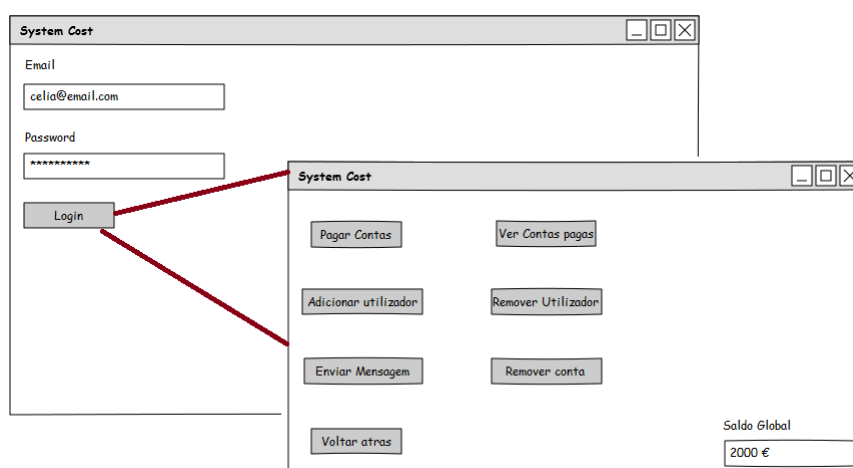


Figura 5.13: Login/Privilégios de administrador

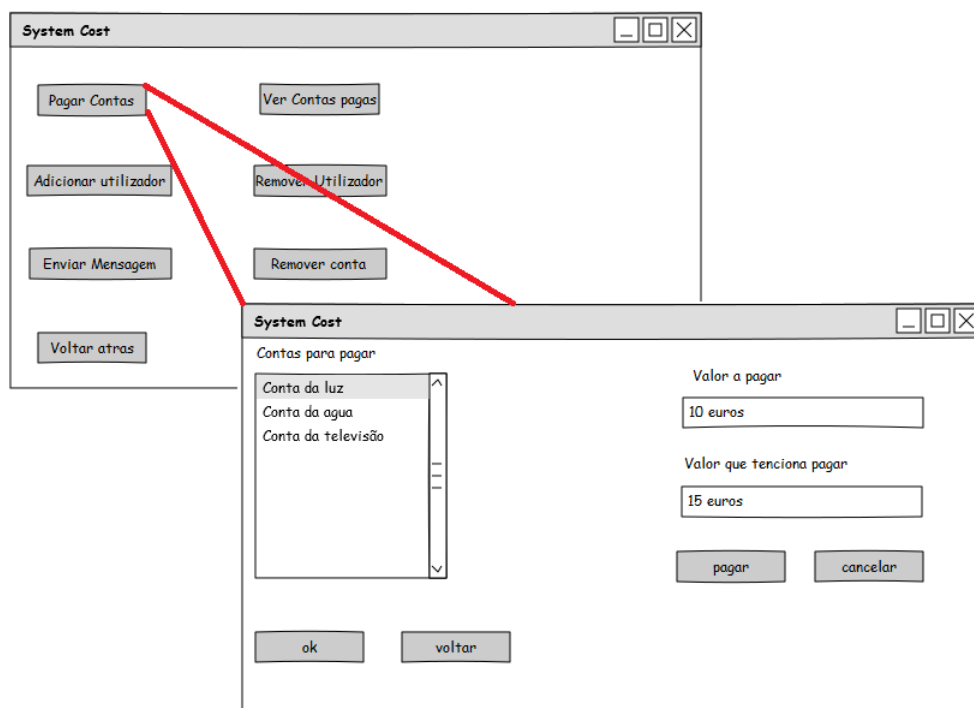


Figura 5.14: Interface do administrador, janela pagar contas

6. Conclusão

Para concluir esta primeira fase de modelação do projeto que consiste em desenvolver um sistema de suporte à partilha de despesas num apartamento, foi-nos proposto enquadrar e descrever da forma mais detalhada possível o sistema a ser desenvolvido. Para isso fizemos uma descrição do processo de análise de requisitos construindo assim o modelo de domínio. Do modelo de domínio e requisitos do sistema foi possível desenvolver os diagramas de 'Use Case' e a posterior especificação de cada um deles.

Depois de todos estes elementos procedemos à fase de pensar de como seria a nossa aplicação fisicamente falando e então para tal utilizamos o programa 'Pencil' e desta forma desenvolvemos a nossa primeira proposta de interface com o utilizador. Construimos também os diagramas de máquinas de estado de acordo com a interface pensada. Porém estes dois pontos ainda têm muitas arestas para limar, pois ainda não temos uma interface completa com todas as possíveis funcionalidades.

Um dos principais problemas que encontramos foi a modelação do Modelo de Dominio, pois estavam sempre a surgir novos requisitos e a serem eliminados outros assim como nas especificações dos 'Use Case'. Podemos concluir que esta é uma etapa que requer uma análise muito cuidada, pois estão sempre a surgir novas maneiras de abordar o projeto.