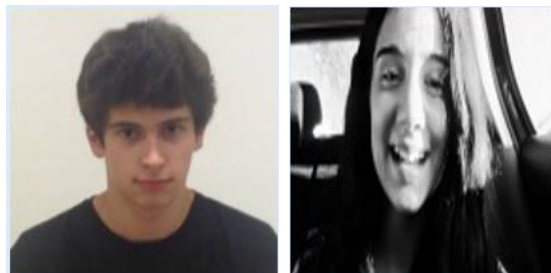


# Desenvolvimento de Sistemas de Software Gestão Apartamentos

André Rodrigues Freitas A74619  
Sofia Manuela Gomes de Carvalho A76658

13 de Novembro de 2016



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Desenvolvimento</b>	<b>4</b>
2.1	Modelo de Domínio . . . . .	5
2.2	Modelo de Use Case . . . . .	6
2.2.1	Autenticação . . . . .	7
2.2.2	AcrescentarDespesas . . . . .	8
2.2.3	PagarDespesas . . . . .	8
2.2.4	ConsultarDespesas . . . . .	9
2.2.5	ConsultarPagamentosEfetuados . . . . .	9
2.2.6	AcrescentarMorador . . . . .	10
2.2.7	RemoverMorador . . . . .	11
<b>3</b>	<b>Conclusão</b>	<b>12</b>

# 1 Introdução

Este trabalho pretende desenvolver um sistema de suporte à partilha de despesas num apartamento. A aplicação desenvolvida suporta o registo das várias despesas e da sua gestão a nível de pagamentos por parte de cada um dos moradores registados nesse apartamento. Cada morador tem a sua própria conta.

Este sistema tem em conta que moradores habitam o apartamento num determinado momento, tendo apenas estes permissão para fazer operações e visualizar as despesas relativas ao apartamento em causa. Para poder fazer qualquer tipo de operação neste sistema, o morador precisa de estar autenticado. Os dados para autenticação num certo apartamento (nome de utilizador e palavra-passe) são fornecidos pelo senhorio, aquando da conclusão do acordo de arrendamento.

## 2 Desenvolvimento

No nosso sistema, o senhorio é o único autorizado a remover um morador, ou seja, apenas este pode remover uma conta associada a um morador de um apartamento, apartamento este que o morador deixou de habitar. Cada senhorio pode, assim, estar associado a vários apartamentos mas cada morador só pode estar associado a uma e uma só habitação de cada vez.

Da mesma forma que o senhorio pode remover moradores, este pode também acrescentar moradores a um dado apartamento.

A consulta de despesas pode ser efetuada pelo senhorio e pelos moradores. Assim, um morador pode ver apenas as suas despesas e um senhorio pode ver todas as despesas de todos os moradores num determinado apartamento.

A consulta dos pagamentos efetuados funciona de forma semelhante à consulta de despesas, sendo que aqui apenas se vêem as despesas já pagas.

Da parte do morador, este tem de se autenticar para fazer qualquer operação na aplicação. Ele tem a possibilidade de acrescentar despesas ao sistema, sejam estas recorrentes (água, luz, eletricidade, gás, renda e condomínio) ou extraordinárias (por exemplo: obras no apartamento ou compras de mobiliário, entre outras).

Outra funcionalidade prevista é pagar despesas. Também aqui o morador terá de indicar o tipo de despesa e depois se vai pagar essa despesa no mês correspondente ou se quer fazê-lo apenas e só no mês seguinte, com juros de 10%. Terá, por isso, de pagar despesas, no máximo, com um mês de atraso.

## 2.1 Modelo de Domínio

Para o Modelo De Domínio criamos várias entidades:

- Morador: esta entidade corresponde a um morador que habita o apartamento alugado;
- Senhorio: o senhorio é o proprietário do apartamento, sendo que cada senhorio pode possuir um ou mais apartamentos;
- Apartamento: é ao qual se vão associar despesas, moradores e pagamentos.
- DespesaRecorrente: vai corresponder às despesas pagas todos os meses tais como eletricidade, água, gás, entre outras. Estas poderão variar de apartamento para apartamento visto que em alguns poderão haver, por exemplo, despesas com pacotes de Internet e televisão que poderão não haver noutros.
- DespesaExtra: são o tipo de despesas extraordinárias, aquelas despesas que não constam do contrato feito. Podem resultar da necessidade de reparar algo no apartamento, entre outros.

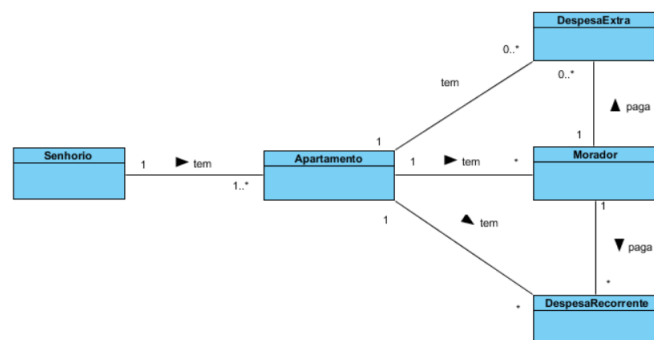


Figura 1: Modelo De Domínio

## 2.2 Modelo de Use Case

Na definição de requisitos funcionais, identificamos dois atores: o Morador e o Senhorio, pois são aqueles que vão interagir com o sistema. Já na identificação dos Use Case, definimos sete Use Case, sendo dois deles comuns aos dois atores que vão interagir com o sistema.

Os comportamentos do sistema, da parte do Morador, são:

- Autenticação
- AcrescentarDespesas
- ConsultarDespesas
- ConsultarPagamentosEfetuados
- PagarDespesas

Da parte do Senhorio, o sistema permite:

- AcrescentaMorador
- ConsultarDespesas
- ConsultarPagamentosEfetuados
- RemoveMorador

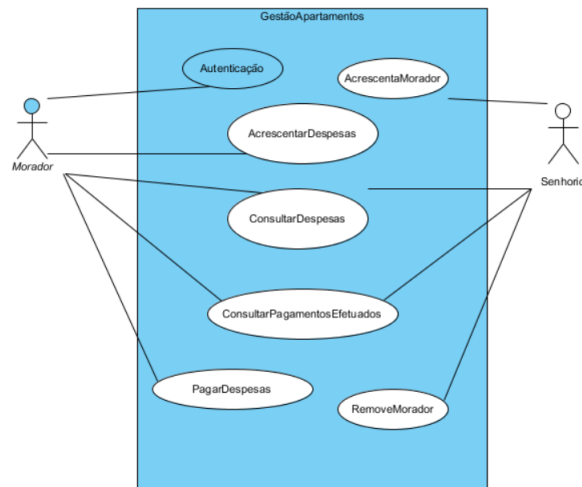


Figura 2: Diagrama de Use Case

### 2.2.1 Autenticação

Este Use Case vai permitir ao Morador fazer a sua autenticação no sistema, para assim poder ter acesso às suas contas, ao pagamentos efetuados e às despesas por pagar, entre outros. Para tal, é necessário que não esteja ainda nenhum Morador autenticado. Assim, o Morador escreve o seu email e a respetiva password e o sistema valida o acesso e apresenta as várias opções disponíveis. Caso o email ou a password estejam incorretas, é lançado um aviso a dizer que estas estão incorretas e volta a pedir o email e a password.

Autenticação		
Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References Description		
Agency FB		
Main		
Super Use Case		
Author		
Date		
Brief Description	Morador autentica-se na aplicação	
Preconditions	Nenhum morador autenticado	
Post-conditions	Morador fica autenticado	
Flow of Events		Actor Input System Response
	1	Apresenta email e password
	2	Valida acesso
	3	Apresentar opções
Comp. Alternativo 1 [email ou password inválida] (passo 2)		Actor Input System Response
	1	Avisa sobre email ou password inválida
	2	Regressa ao passo 1

Figura 3: Especificação Tabular do Use Case Autenticação

## 2.2.2 AcrescentarDespesas

AcrescentarDespesas exige que o Morador esteja autenticado. Quando tal acontece, este indica o valor da despesa, a sua data de emissão e a data limite de pagamento e o sistema regista esta despesa.

AcrescentarDespesas		
Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References Description		
Main	Agency FB 8	
Super Use Case		
Author		
Date		
Brief Description	Acrescenta despesas de um certo tipo	
Preconditions	Utilizador autenticado	
Post-conditions	Despesa adicionada	
Flow of Events		
	1	Indica valor da despesa, data de emissão e data limite de pagamento
	2	Regista despesa

Figura 4: Especificação Tabular do Use Case AcrescentarDespesas

## 2.2.3 PagarDespesas

Este Use Case exige que o Morador esteja autenticado, sendo que o sistema vai mostrar o tipo de despesas e o ator seleciona de seguida o tipo de despesa. Se não tiver despesas desse tipo por pagar, o sistema volta a mostrar os tipos de despesas. Caso contrário, o ator pode escolher as despesas que quer pagar e o sistema irá gerar as referências bancárias para que isso possa ser feito.

PagarDespesas		
Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References Description		
Main	Agency FB 8	
Super Use Case		
Author		
Date		
Brief Description	Morador recebe referência de pagamento	
Preconditions	Está autenticado	
Post-conditions	Referências bancárias geradas	
Flow of Events		
	1	Mostra tipos de contas
	2	Seleciona tipo de contas
	3	Mostra contas por pagar
	4	Seleciona contas que quer pagar
	5	Fornecer uma referência bancária
Comp.Alternativo1 (passo 3) (sem contas daquele tipo por pagar)		
	1	Regressa a 1

Figura 5: Especificação Tabular do Use Case PagarDespesas



### 2.2.4 ConsultarDespesas

Este Use Case é comum aos dois atores do sistema. Permite consultar a lista de um dado tipo de despesas. Para tal, o ator indica que quer consultar a lista de despesas e o sistema fornece os tipos de despesas. De seguida, o ator seleciona o tipo de despesa que quer consultar e é apresentada uma lista de despesas do tipo selecionado. Se, ao selecionar um tipo que não tem nenhuma despesa, o sistema indica que não há nenhuma despesa desse tipo na lista e volta a fornecer os tipos de despesa para, quem sabe, nova escolha de tipo.

ConsultarDespesas		
Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References Description		
Agency FB 8		
Man	Super Use Case	
	Author	
	Date	
	Brief Description	
	Preconditions	
	Post-conditions	
	Despesas consultadas	
Flow of Events	Actor Input	
	System Response	
	1	Indica que quer consultar a lista de despesas
	2	Fornecer os tipos de despesas
Comp.Alternativo (passo 4) [não há lista de despesas]	Actor Input	
	System Response	
	1	Indica que não há nenhuma despesa na lista
	2	Regressa a 2

Figura 6: Especificação Tabular do Use Case ConsultarDespesas

### 2.2.5 ConsultarPagamentosEfetuados

É outro Use Case comum aos dois atores. É necessário ou ser um Morador autenticado ou ser Senhorio para poder consultar os pagamentos já efetuados. O ator indica que quer consultar a lista de pagamentos efetuados e o sistema devolve o tipo de despesas para o ator selecionar. Este seleciona o tipo de pagamento que quer consultar e é devolvida pelo sistema a respetiva lista de pagamentos. Contudo, o sistema pode divulgar uma mensagem sobre o facto de não existir nenhum pagamento efetuado do tipo selecionado e volta a apresentar os tipos de despesas.

ConsultarPagamentosEfetuados		
Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References Description		
Agency FB 8		
Main		
Super Use Case		
Author		
Date		
Brief Description		
Preconditions		
Post-conditions		
Flow of Events	Actor Input	
	System Response	
	1	Indica que quer consultar a lista de pagamentos já efetuados
	2	Fornecer os tipos de despesas
Comp. Alternativo (passo 5)	Actor Input	
	System Response	
Inã há lista de pagamentos efetuados	1	Indica que não há nenhum pagamento efetuado do tipo selecionado na lista
	2	Regressa a 3

Figura 7: Especificação Tabular do Use Case ConsultarPagamentosEfetuados

2.2.6 AcrescentarMorador

Este Use Case permite adicionar um morador ainda não existente. O Senhorio indica em que apartamento quer acrescentar o novo morador e o sistema apresenta os detalhes do apartamento. De seguida, o Senhorio fornece o nome, o NIF e o contacto do novo morador e o sistema regista-o.

AcrescentaMorador		
Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References Description		
Agency FB 8		
Main		
Super Use Case		
Author		
Date		
Brief Description		
Preconditions		
Post-conditions		
Flow of Events	Actor Input	
	System Response	
	1	Indica apartamento (morada)
	2	Apresenta detalhes do apartamento
	Actor Input	
	System Response	
	3	Indica nome, NIF e contacto
	4	Regista morador

Figura 8: Especificação Tabular do Use Case AcrescentaMorador

### 2.2.7 RemoveMorador

Apenas o Senhorio pode aceder a este Use Case que remove um morador. Para tal, indica o apartamento sobre o qual quer remover o morador e o sistema apresenta os detalhes do apartamento. O Senhorio indica o NIF do morador a remover e o sistema remove-o.

RemoveMorador		
Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References Description		
Agency FB 8		
Man		
Super Use Case	Author	
	Date	
	Brief Description	Remove um morador
	Preconditions	Feito pelo Senhorio
	Post-conditions	Morador removido
Flow of Events	1	Indica apartamento (morada)
	2	Apresenta detalhes do apartamento
	3	Indica NIF
	4	Remove morador associado ao NIF escolhida

Figura 9: Especificação Tabular do Use Case RemoveMorador

### 3 Conclusão

Nesta primeira fase do trabalho, optamos por implementar desta forma o sistema embora nos tenhamos, contudo, deparado com várias opções alternativas às que implementamos. Chegamos a equacionar a divisão da entidade `DespesaRecorrente` em várias que representariam despesas mais específicas como despesas com luz, água e eletricidade. No entanto, apercebemo-nos que isso iria restringir bastante a flexibilidade da aplicação e não teria em conta as possíveis diferenças nos apartamentos.

Em relação aos Use Case, optamos por ter um ator `Senhorio` que funcionasse um pouco como um possível administrador, acima de tudo, por causa da gestão de quem está a habitar um dado apartamento num certo momento. O facto de poder consultar as despesas e os pagamentos já efetuados permite-lhe ter a noção do que se está a passar no apartamento, mesmo que algumas das despesas não necessitem totalmente de lhe ser comunicadas. Por isso, foi necessário dividir os atos de consultar despesas e pagar em dois Use Case visto que num deles apenas tem acesso o `Morador`, mas no outro têm acesso ambos os atores. As despesas já pagas num Use Case à parte estão destinadas a fornecer uma forma rápida ao `Senhorio` de saber quem já pagou e o que pagou. São também úteis para os moradores para que estes possam verificar rapidamente o que já pagaram. Apesar disso, julgamos que seria também possível num só Use Case consultar e pagar as despesas e será algo que deverá ser analisado tendo em conta a segunda parte deste projeto.