



Universidade do Minho

Conselho de Cursos de Engenharia

Licenciatura em Engenharia Informática

3ºAno

Disciplina de Desenvolvimento de Sistemas de Software

Ano Lectivo de 2009/2010

Relatório do projecto - 1º Fase

Grupo 30

Dezembro, 2009

Identificação do grupo:

Grupo 30

Nome: Ivo Madeira Macedo

Número: 47120

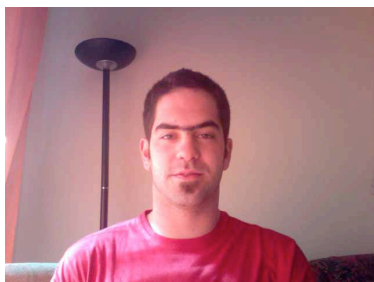
Fotografia:



Nome: José Filipe Figueiredo

Número: 47046

Fotografia:



Resumo

Considere-se a área da actuação das empresas que fazem gestão do serviço em condomínios fechados.

Pretende-se com este trabalho, dar uma proposta de um sistema de software, em que fica encarregue de assegurar a gestão dos diversos trabalhos a realizar, em casa dos condóminos.

O sistema a desenvolver, designado por GCS, deverá efectuar toda a gestão de serviços, desde o cliente, a empresa GereComSaber e os seus fornecedores.

Para satisfazer as necessidades do cliente, a empresa GereComSaber, possui uma carteira de fornecedores que subcontrata, e que são os responsáveis pela realização dos serviços.

É possível que para um determinado tipo de serviço exista mais do que um fornecedor disponível. Cabe à empresa GereComSaber escolher qual o fornecedor que indica ao cliente

A proposta de modelação e respectivo desenvolvimento na linguagem de programação orientada aos objectos – Java.

Área de Aplicação: Modelação e Desenvolvimento de Sistemas de Software

Palavras-Chave: Unified Modeling Language.

Índice:

1. Introdução	pag.3
2. Motivação e objectivos	pag.4
3. Diagramas de Use Cases's	pag.5
4. Modelo de Domínio	pag.10
5. Conclusões e trabalho futuro	pag.11
6. Anexos	pag.12

1. Introdução:

No âmbito da disciplina de Desenvolvimento de Sistemas de Software da Licenciatura em Engenharia Informática da universidade do Minho, criámos uma proposta de um sistema de software para gestão de serviços em condomínios fechados.

Para a modelação do nosso projecto usámos o UML (Unified Modeling Language) que é uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objectos. Sintetiza os principais métodos existentes, sendo considerada uma das linguagens mais expressivas para modelagem de sistemas orientados a objectos. Por meio dos seus diagramas é possível representar sistemas de software sob diversas perspectivas de visualização. Facilita a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um sistema - gerentes, coordenadores, analistas, desenvolvedores - por apresentar um vocabulário de fácil entendimento.

2. Motivação e objectivos:

Este projecto, será constituído por várias fases. A primeira fase (fase actual) é de levantamento de requisitos, com recurso à identificação dos use cases e respectiva descrição textual, e também o modelo de domínio; A segunda fase consistirá na realização da arquitectura do sistema proposto para satisfazer os requisitos identificados na primeira parte, através dos diagramas de sequência para as interacções descritas nos use cases, e um refinamento dos diagramas de classe). Por fim, apresentamos a implementação da solução proposta.

3. Diagramas de Use Cases

Os diagramas de Use Case permitem definir os requisitos funcionais de um sistema, isto é :

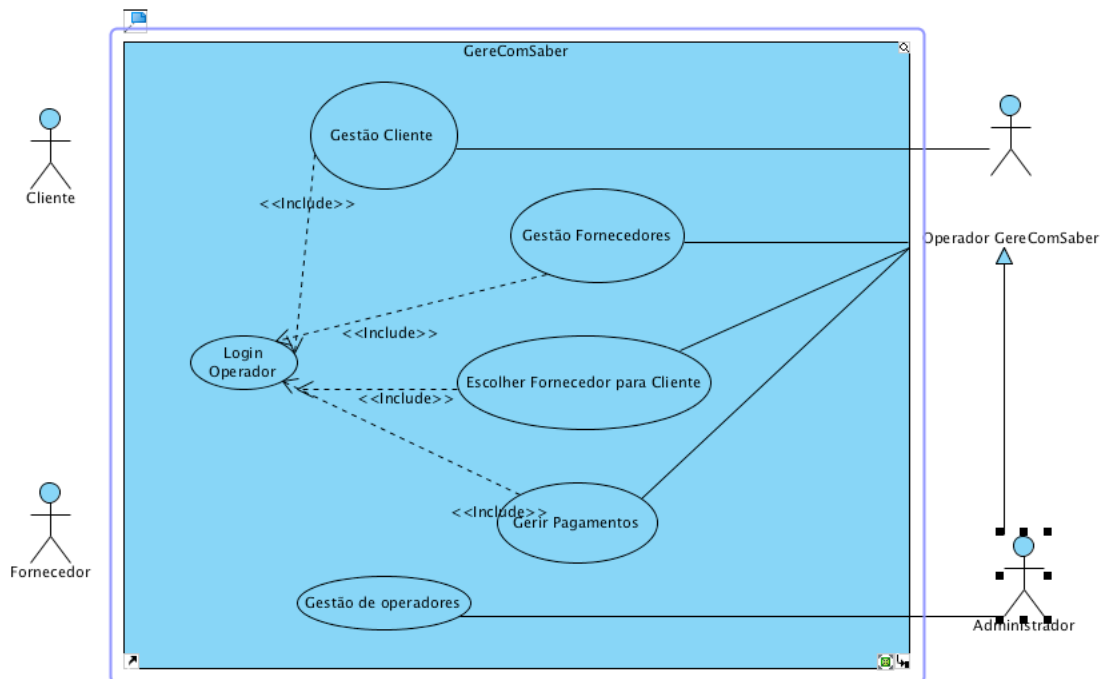
- que serviços deve fornecer;
- a quem os deve fornecer.

Utilizando diagramas de use case, clientes e equipa de desenvolvimento podem chegar a um acordo sobre qual o sistema a desenvolver, facilitando a resolução de alterações nos requisitos funcionais.

No entanto os diagramas de use case não suportam a captura de requisitos não funcionais.

Em seguida, apresentamos uma breve explicação sobre os principais requisitos da nossa aplicação:

Use Case GereComSaber:

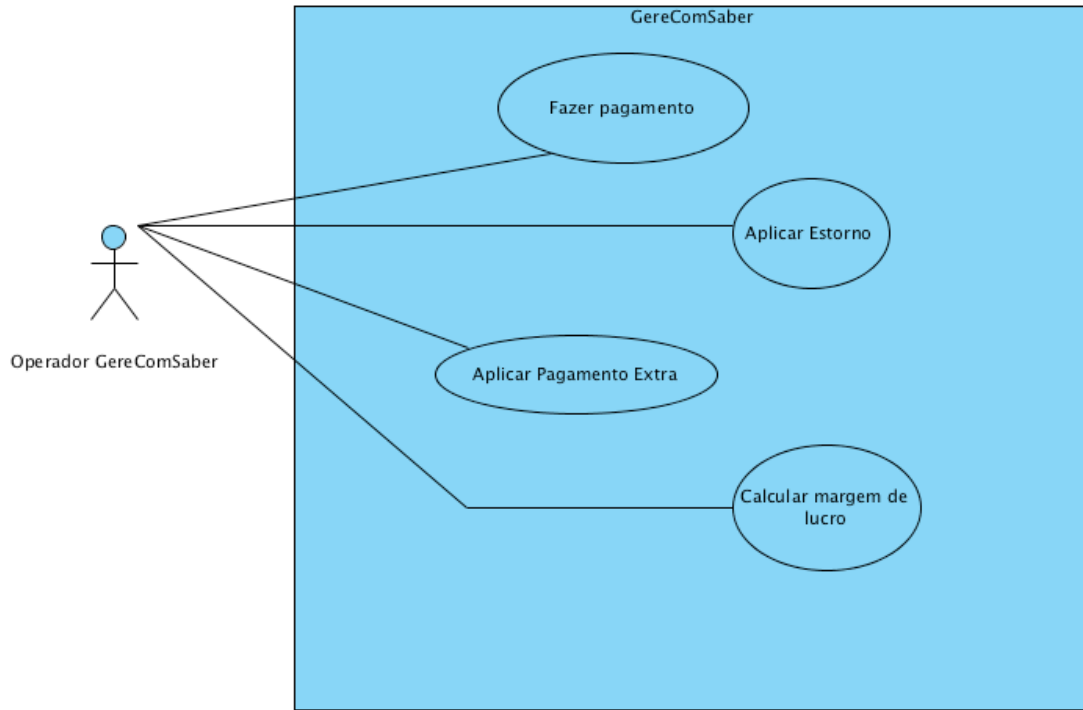


•Diagrama de Use Case GereComSaber

Neste caso tanto o Cliente como o Fornecedor não intervêm directamente no sistema mas uma grande parte das actividades descritas neste diagrama são desencadeadas por ambos, sendo o Operador **GereComSaber** o único actor que interage directamente com o sistema.

Use Case Gerir Pagamentos:

Neste diagrama representamos os requisitos que o operador da GereComSaber deve ter, relativamente à Gestão de Pagamentos.



•Diagrama de Use Case Gerir Pagamentos

No Use case **Fazer Pagamento** , por cada contracto do cliente, calcula o montante que tem de ser debitado aos fornecedores, tal como podemos verificar no diagrama textual:

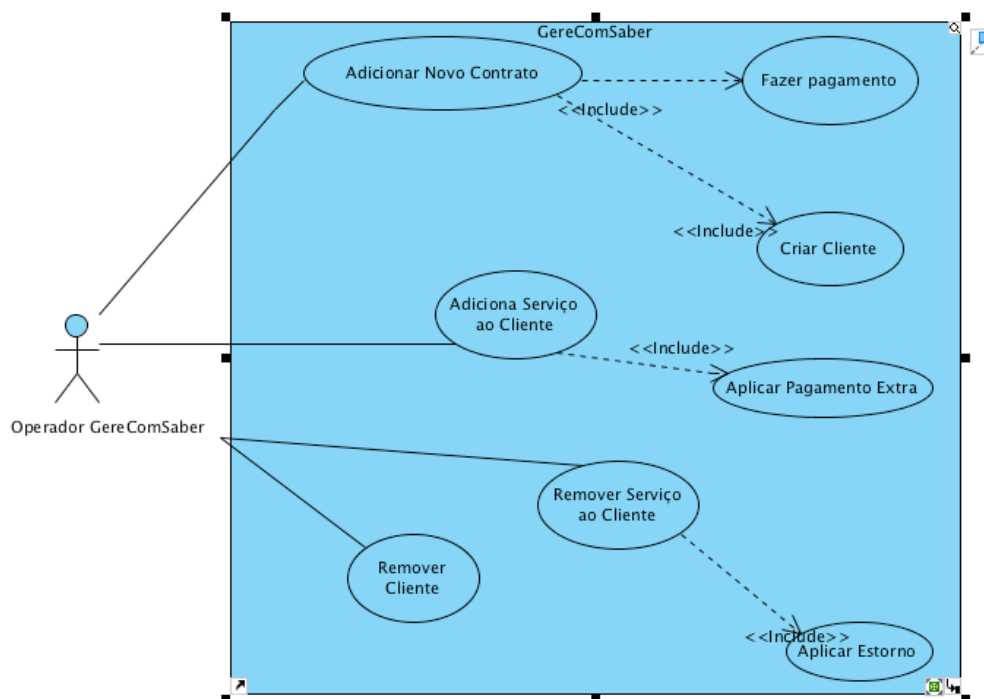
Brief Description	A GereComSaber faz o pagamento do valor especificado em determinado contrato. (um ou mais fornecedores)	
Preconditions		
Post-conditions		
Flow of Events	Actor Input	System Response
	1 Actor escolhe o Contrato de Cliente.	
	2	Sistema avalia a validade do contrato.
	3	Sistema debita o montante respectivo a cada fornecedor especificado no contrato.
4	Sistema notifica o Actor da transacao e guarda-a em historico.	
Alternative 1 contrato invalido(cliente inexistente, servicos desactualizados)	Actor Input	System Response
	1	Notificacao ao Actor.
2	Volta ao passo 1.	
Exception 1 saldo insuficiente	Actor Input	System Response
	1	Notificar Actor.

•Especificação textual do Use Case Fazer Pagamento

Os Use Cases's Aplicar Estorno e Pagamento Extra, fazem parte dos acertos ao valor do contrato do cliente, isto é, quando o cliente quer aumentar ou diminuir os serviços realizados.

Por último, o UC Calcular Margem de Lucro, informa o lucro total que a empresa GereComSaber teve ao longo do tempo, que correspondem às comissões que a empresa teve por cada cliente.

Seguidamente vamos explicitar o diagrama **Gerir Clientes**.



- Diagrama de Use Case Gerir Clientes

Neste diagrama é especificado todo o comportamento dos Use Case's que estão de algum modo relacionado com acções relativas a Clientes. Passaremos a explicitar os Use Case's : **Adicionar Novo Contrato** e **Remover Serviço ao Cliente**.

Super Use Case			
Author	macbook		
Date	30/11/2009 12:25:28		
Brief Description	Cria um novo Contrato.		
Preconditions			
Post-conditions			
Flow of Events		Actor Input	System Response
	1	<<include>> Criar Cliente.	
	2		Sistema disponibiliza a listagem de fornecedores e serviços.
	3	Actor designa quais os serviços/actividades a adicionar ao contrato.	
	4		Sistema associa os serviços/actividades ao cliente e respectivo contrato.
	5		<<include>> Fazer Pagamento
	6		Notificacao ao actor.
	7		Registo em historico.

- Descrição textual do Use Case **Adicionar Novo Contrato**

Como podemos verificar, na criação de um novo contrato, é primeiramente criado um novo Cliente com as devidas informações, é preenchido o contrato do Cliente de acordo com os serviços e/ou actividades escolhidos e no fim, e de acordo com o enunciado é activado o comportamento do Use Case **Fazer Pagamento** através da sua inclusão (<<include>>).

Super Use Case			
Author	macbook		
Date	30/11/2009 12:07:58		
Brief Description	Actor remove um serviço da lista de serviços do contrato de um cliente.		
Preconditions			
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Actor designa um cliente.	
	2		Sistema disponibiliza ao Actor todos os serviços presentes no contrato desse cliente.
	3	Actor designa um serviço.	
	4		Sistema remove o serviço da lista de serviços do cliente.
	5		<<include>> Aplicar Estorno
	6		Sistema notifica o Actor.
7		Alteração registada em historico.	
Alternative 1 cliente não existe		Actor Input	System Response
	1		Notificação ao Actor.
	2		Voltar ao passo 1.

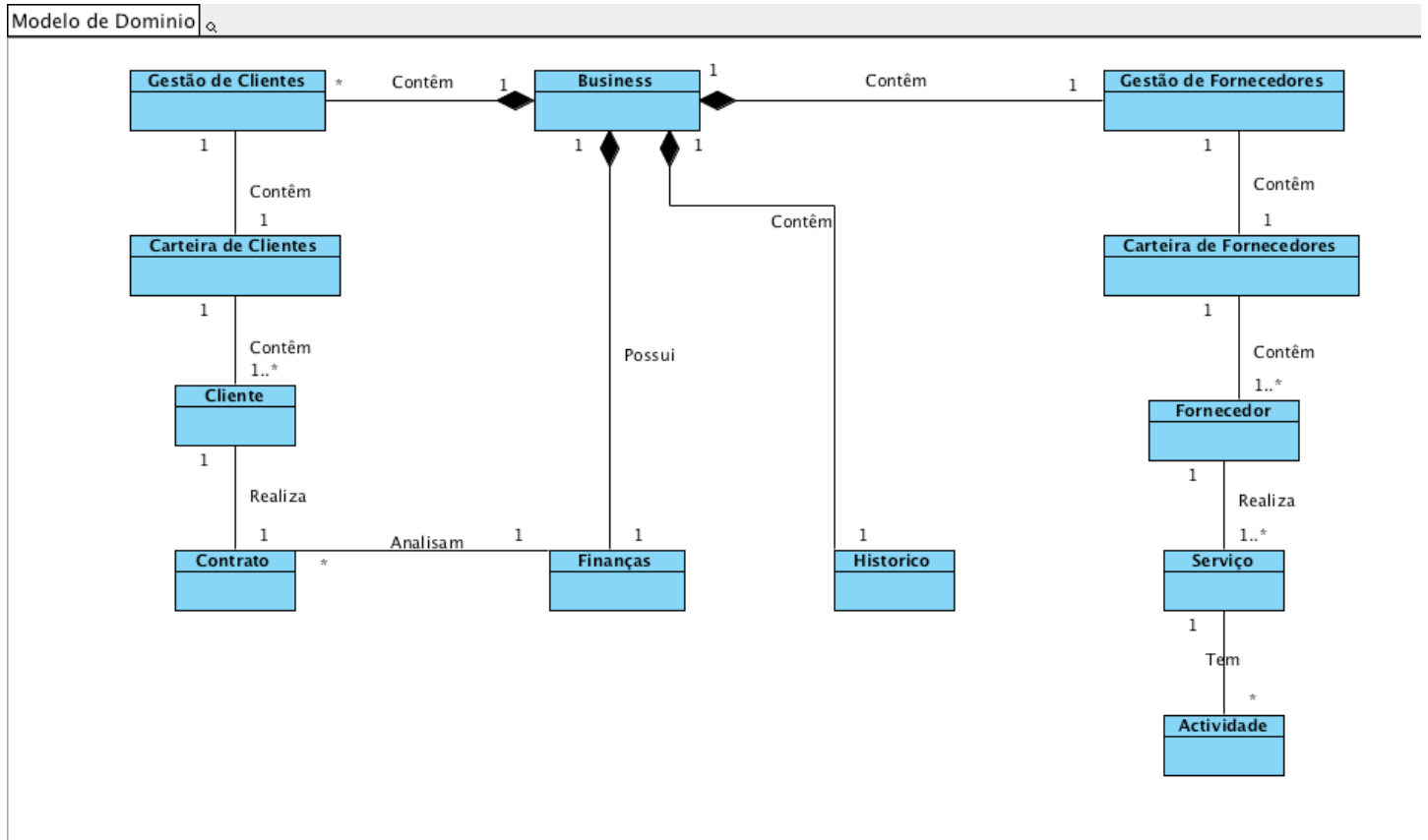
•Especificação textual do Use Case **Remover Serviço ao Cliente**

No contexto das alterações na quantidade de serviços de determinado Cliente ao longo de cada período, optámos por sempre que a lista de serviços contratados é alterada o acerto é efectuado imediatamente. Por exemplo, ao observarmos o comportamento deste Use Case podemos verificar que quando um serviço é removida do contrato do Cliente, o montante a pagar a partir dessa alteração terá necessariamente de diminuir. Este acerto é feito a partir da inclusão do Use Case **Aplicar Estorno**.

Optámos por este método ao invés de fazer os acertos no fim de cada período.

4. Modelo de Domínio:

O modelo de domínio ilustra as classes conceptuais significativas, no domínio do problema real, não sendo uma descrição dos objectos (ou classes) do software. O modelos de domínio que melhor se adequa ao nosso problema é o seguinte:



- Modelo de Domínio GCS

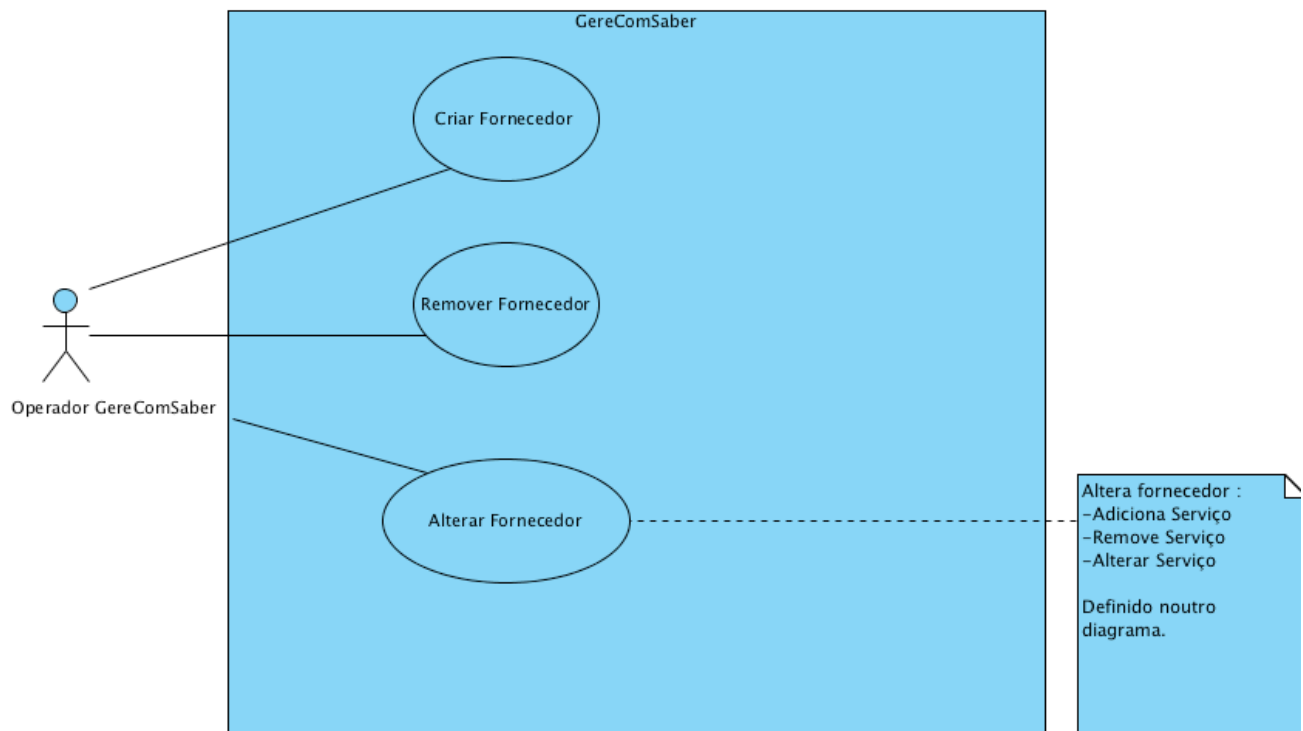
5. Conclusões e trabalho futuro:

A base para um produto final rigoroso e preciso tendo em vista a satisfação do cliente, será uma modelação cuidada e rigorosa, desde o levantamento de requisitos, a sua implementação e até ao produto final. Isto será possível com a ajuda dos diagramas UML que a aplicação Visual Paradigm fornece.

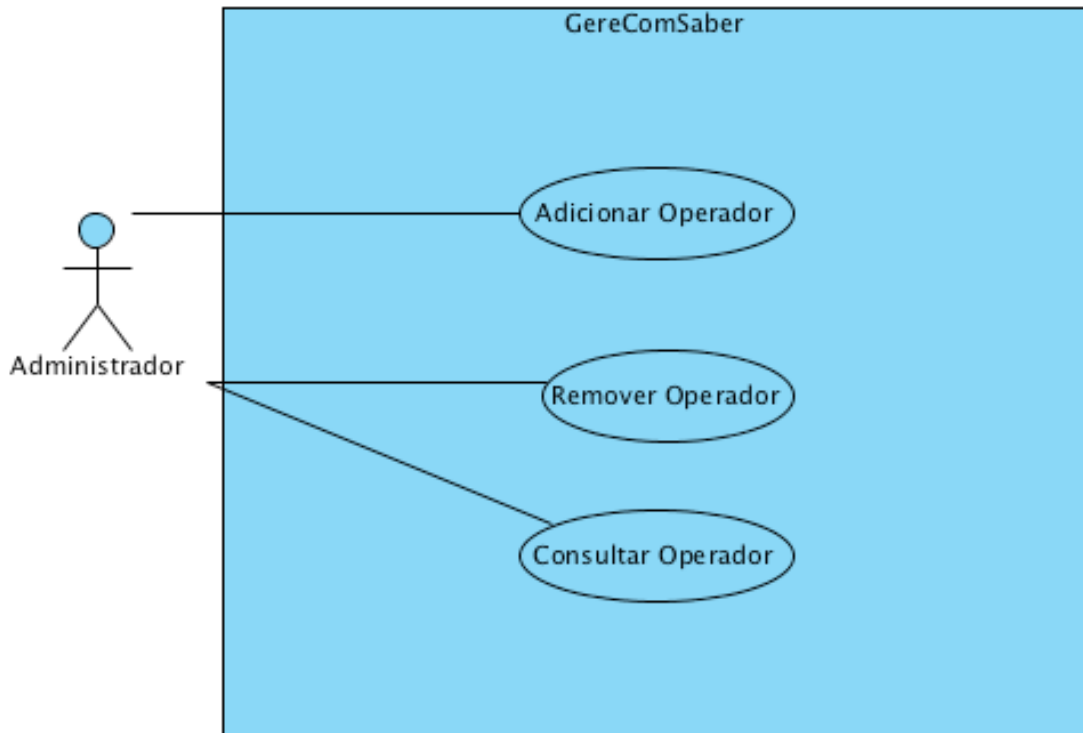
A principal dificuldade que encontra-mos inicialmente, foi a utilização e manuseamento da aplicação proposta – Visual Paradigm, mas foi ultrapassada satisfatoriamente através da prática e desenvolvimento do nosso sistema.

Anexos:

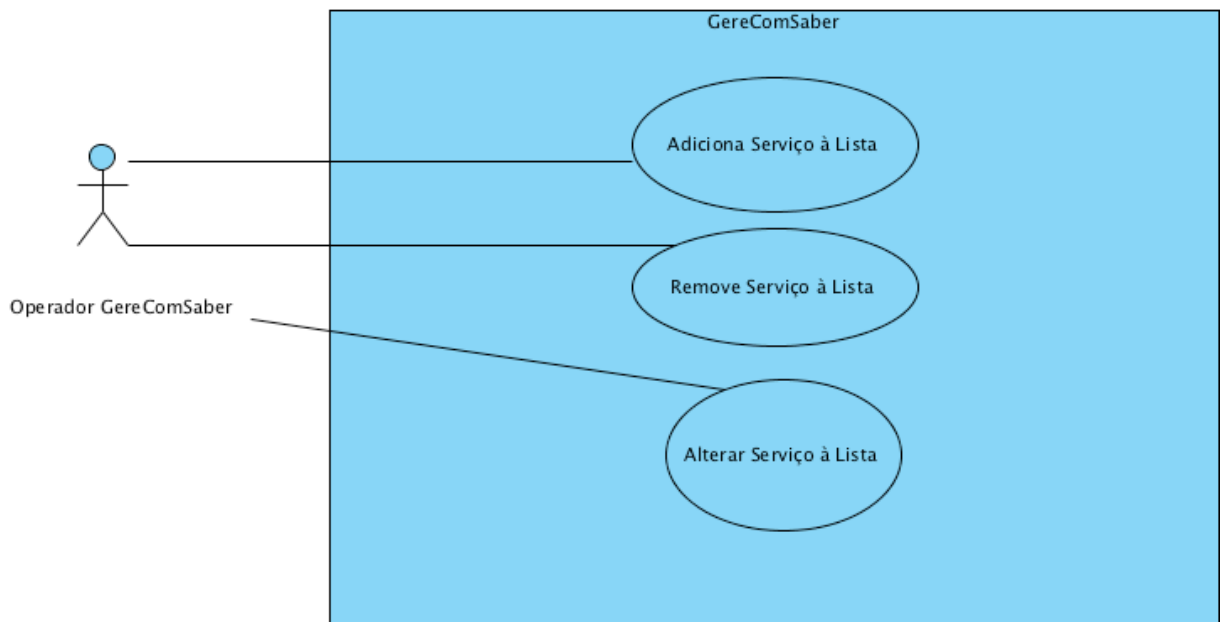
Apresentam-se aqui os Use Cases (e especificação textual) mais relevantes para o projecto:



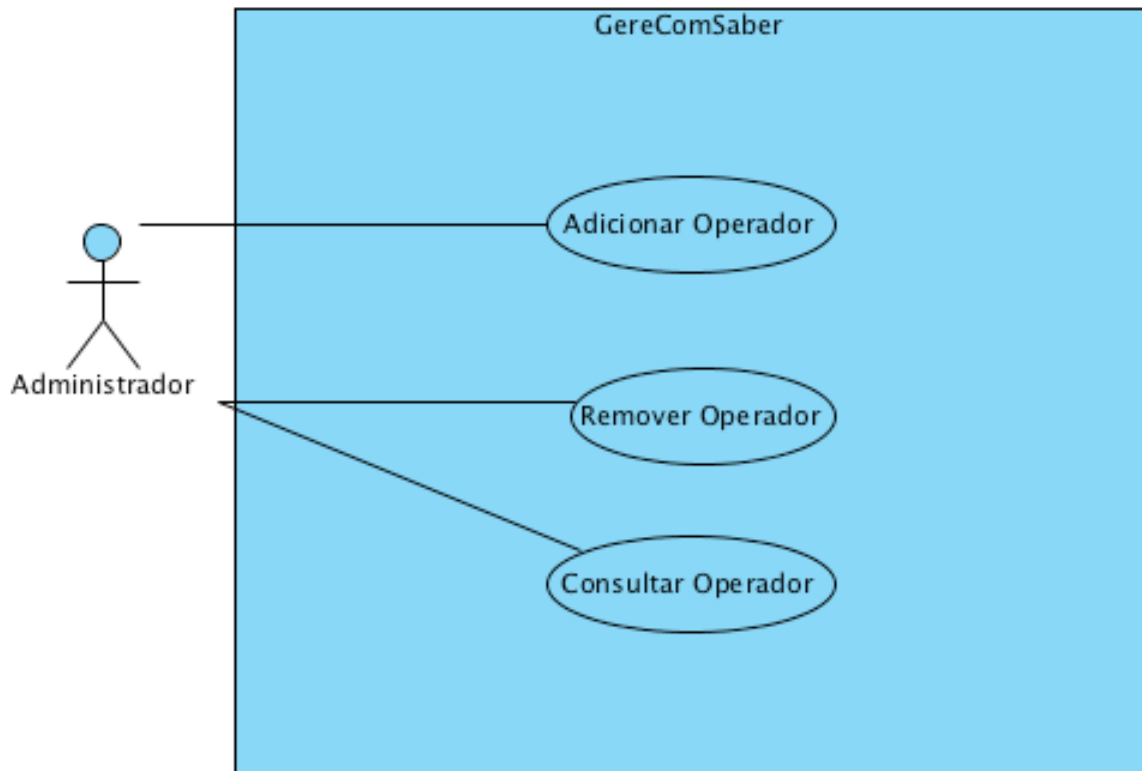
• Diagrama de Use Case **Gerir Fornecedor**



• Diagrama de Use Case **Gestão de Operadores**



• Diagrama de Use Case **Alterar Fornecedor**



• Diagrama de Use Case **Gerir Operador**

Post-conditions	Sucesso: A lista de serviços e actualizada com o novo serviço. Insucesso: Lista de serviços inalterada.	
Flow of Events		Actor Input
	1	Actor designa um serviço a ser adicionado.
	2	Actor designa um cliente.
	3	
	4	
	5	
		System Response
		Sistema adiciona o serviços a lista de serviços do cliente.
		Notificacao do autor.
		Alteracao registada em historico.
Alternative 1 servico nao disponivel		Actor Input
	1	
	2	
		System Response
		Notificacao do Sistema.
		Voltar ao passo 1.
Alternative 2 cliente nao existente		Actor Input
	1	
	2	
		System Response
		Notificacao do Sistema.
		Voltar ao passo 2.
Alternative 3 servico ja existente no contrato do cliente		Actor Input
	1	
	2	
		System Response
		Notificacao do Sistema.
		Voltar ao passo 1.

• Especificação Textual do Use Case **Adicionar Serviço ao Cliente**

Brief Description	GereComSaber devolve um montante (estorno) ao Cliente.	
Preconditions		
Post-conditions	Sucesso : Saldo GereComSaber diminui consoante o estorno. Insucesso : Saldo permanece Inalterado.	
Flow of Events		Actor Input
		System Response
	1	Sistema calcula o valor do montante a retornar ao Cliente.
	2	Debito do montante, no saldo do Sistema.
	3	Notificacao do Sistema.
4	Registo da movimentacao no historico.	
Exception 1 saldo insuficiente		Actor Input
	1	Notificacao ao Actor.

• Especificação Textual do Use Case **Aplicar Estorno**

Brief Description	Actor adiciona um novo cliente a carteira de Clientes da GereComSaber.	
Preconditions		
Post-conditions	Sucesso : Carteira de clientes actualizada com novo cliente. Insucesso : Carteira de clientes inalterada.	
Flow of Events		Actor Input
		System Response
	1	Actor introduz informacoes do novo cliente.
	2	Sistema cria um Cliente.
	3	Sistema avalia o Cliente.
	4	Sistema adiciona o Cliente a carteira de clientes do sistema.
	5	Notificacao ao actor.
6	Registo em historico.	
Alternative 1 dados invalidos		Actor Input
	1	Voltar ao passo 1.
Alternative 2 cliente ja existe na base de dados		Actor Input
		System Response
	1	Notificacao ao actor.
2	Voltar ao passo 1.	

• Especificação Textual do Use Case **Criar Cliente**

Brief Description	Actor remove um <u>serviço da lista de serviços do contrato de um cliente.</u>		
Preconditions			
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Actor <u>designa um cliente.</u>	
	2		<u>Sistema disponibiliza ao Actor todos os serviços presentes no contrato desse cliente.</u>
	3	Actor <u>designa um serviço.</u>	
	4		<u>Sistema remove o serviço da lista de serviços do cliente.</u>
	5		<u><<include>> Aplicar Estorno</u>
	6		<u>Sistema notifica o Actor.</u>
	7		<u>Alteração registada em historico.</u>
Alternative 1 <u>cliente não existe</u>		Actor Input	System Response
	1		<u>Notificação ao Actor.</u>
	2		<u>Voltar ao passo 1.</u>

- Especificação Textual do Use Case **Remover Serviço ao Cliente**