

Universidade do Minho
Conselho de Cursos de Engenharia
Licenciatura em Engenharia Informática

Desenvolvimento de Sistemas de Software (Grupo 22)

Ano Lectivo de 2009/10



Universidade do Minho



escola de engenharia



departamento de
informática

Gestão de Condomínios

Desenvolvido por:

António Bouças (49330)
Paulo Festa (49419)
Tiago Mendes (51760)
André Magalhães (51821)

Novembro, 2009



Resumo

O projecto desenvolvido e descrito neste documento é parte integrante da cadeira de *Desenvolvimento de Sistemas de Software* do curso Licenciatura em Engenharia Informática (LEI) e visa modelar e concretizar um projecto.

Este projecto tem como principal objectivo desenvolver uma aplicação multi-utilizador que permita ao fornecedor proporcionar serviços, ao cliente requisita-los e ao administrador garantir a correcta gestão dos condomínios fechados.

Conteúdo

Conteúdo	1
Lista de Figuras	3
1 Introdução	4
1.1 Apresentação do Caso de Estudo	4
1.2 Objectivos	5
2 Modelo do Domínio	6
2.1 Glossário	7
2.2 Decisões	8
2.3 Diagrama do modelo do domínio	9
3 Use Cases	11
3.1 Use Cases multi-utilizador	11
3.1.1 Login e Logout	11
3.1.2 Pesquisar actividade	11
3.1.3 Consultar contratos	12
3.2 Administrador	12
3.2.1 Gerir Utilizadores	12
3.2.2 Análise de Resultados	12
3.2.3 Alterar Comissão	13
3.3 Fornecedor	13
3.3.1 Consultar Saldo	13
3.3.2 Editar Actividade	13
3.3.3 Anular Serviço	13
3.3.4 Anular Actividade	13
3.3.5 Inserir Serviço	14
3.3.6 Inserir Actividade	14
3.4 Cliente	15
3.4.1 Consulta Histórico	15

3.4.2	Cancelar Serviço	15
3.4.3	Cancelar Actividade	16
3.4.4	Requisitar Serviço	16
3.4.5	Requisitar Actividade	16
4	Conclusão	17

Lista de Figuras

2.1	Modelo do Domínio - Lista de Actividades	7
2.2	Modelo do Domínio - Relação entre Contrato, Cliente e Valor	8
2.3	Modelo do Domínio	10
3.1	Use Case "Inserir Actividade"(Main Flow)	14
3.2	Use Case "Inserir Actividade"(Alternative & Exception)	15
3.3	Use Cases	16

Capítulo 1

Introdução

O **UML** (Unified Modeling Language) é uma linguagem de modelação que nos permite visualizar um projecto com base em diagramas padronizados e com forte carga semântica. Projectos desenvolvidos segundo esta perspectiva requerem o levantamento de requisitos junto do cliente e conhecimento da área de negócio, na qual o projecto se baseia, antes de modelar o sistema ou mesmo desenvolver a aplicação. A implementação apenas surgirá após a completa modelação do sistema.

É na fase de modelação que surge o **Modelo do Domínio** que não é mais que um glossário com o vocabulário fundamental do contexto do problema. Relacionando esses termos iremos definir classes (e as suas ligações) que serão, com grande probabilidade, entidades ou eventos do negócio. Surgem também os **Use Cases**, que são elementos que traduzem funcionalidades do sistema.

1.1 Apresentação do Caso de Estudo

Este projecto tem como objectivo o desenvolvimento de uma aplicação informática eficiente que possibilite gerir serviços em condomínios fechados. Esses serviços são prestados por fornecedores e requisitados por clientes. A GereComSaber é uma empresa intermediária na relação cliente-fornecedor e que garante que todos os serviços são prestados e pagos.

Cada fornecedor irá disponibilizar uma lista de serviços, sendo que cada serviço tem a si associado um conjunto de actividades que podem ser contratadas individualmente ou colectivamente. O cliente poderá aceder a essas actividades e escolher as que pretende adquirir. No entanto, esta comunicação terá que se fazer através da aplicação visto que o cliente não possui qualquer contacto com os fornecedores. Por este motivo os pagamentos serão sempre realizados "à cabeça" à GereComSaber, ficando esta responsável por pagar aos fornecedores. Todas as actividades pagas pelo cliente incluem uma comissão, que será o lucro da empresa GereComSaber.

A requisição de novas actividades implica a celebração de novo contrato. Por esse mo-

tivo a aplicação terá que gerir os acréscimos no montante pago (mensalmente, trimestralmente, etc) pelo cliente. Por outro lado poderá existir cancelamento de contratos sem prejuízo para o cliente, o que desde de logo implica que haja estornos.

1.2 Objectivos

Nesta primeira fase pretende-se construir o **Modelo do Domínio**, bem como o diagrama de **Use Cases**, textualmente definidos.

Capítulo 2

Modelo do Domínio

O **Modelo do Domínio** tem um papel importante na fase inicial da modelação do projecto. Após a recolha de requisitos, por indicação do cliente e dados relativos a área onde se enquadra o problema, constrói-se um glossário com os termos mais relevantes para o domínio do problema que, normalmente, será analisado quer pelos responsáveis do projecto quer pelo cliente que encomenda o software.

Com este dicionário terminológico poderemos relacionar e organizar as entidades identificadas, assim como os eventos, de forma a obter ligações semânticas entre entidades.

De forma a facilitar a leitura das relações entre entidades (ou eventos) e visualizar o sistema, que através do seu trabalho em conjunto fornece um comportamento cooperativo, existem duas características importantes:

Multiplicidade Indica quantos objectos partilham na relação entre eventos/entidades.

Da *Figura 2.1* poderemos retirar que uma lista de actividades (multiplicidade 1) contém uma ou mais (multiplicidade 1..*) actividades e que cada uma destas contém um preço.

Navegação Indica em que sentido pode ser lida e analisada a relação. Da *Figura 2.1* podemos concluir que uma actividade possui um preço mas não poderemos dizer que um preço possui uma actividade.

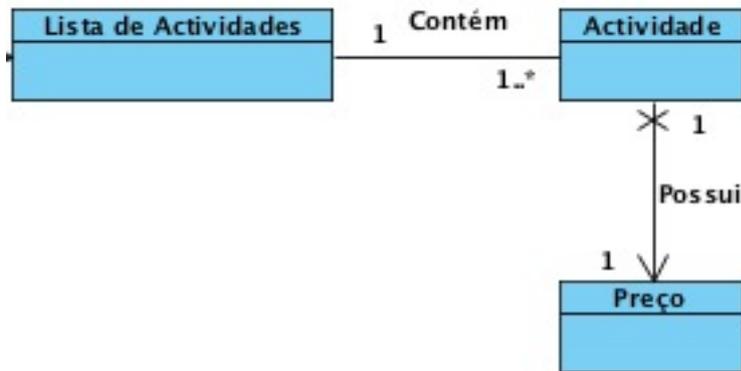


Figura 2.1: Modelo do Domínio - Lista de Actividades

2.1 Glossário

A identificação de eventos ou entidades não é um processo directo, tendo que passar por vários refinamentos numa fase inicial do projecto. Inicialmente foi fácil identificar os actores que interagem com o sistema assim como as entidades mais importantes: contrato, serviços, actividades, etc. Após alterações, algumas delas já depois de ter concebido os Use Cases, obteve-se o seguinte glossário:

Fornecedor O actor que presta serviços ao cliente, usando a GereComSaber como intermediário.

Serviço Serviço prestado por um fornecedor (Ex: Jardinagem) que inclui um conjunto de actividades.

Actividade Realização de um trabalho contratado pelo cliente (Ex: corte de relva).

Lista de Actividades Conjunto de actividades.

Preço Valor monetário associado a uma actividade e pago pelo cliente

Carteira de Fornecedores Conjunto de Fornecedores.

Contrato Documento onde consta as actividades escolhidas e o cliente que as requisitou. Cada contrato apenas pode conter um serviço (tendo este múltiplas actividades).

Contratos Caducados Contratos em que a data de término já foi ultrapassada.

Lista de Contratos Lista de contratos (caducados + em vigor).

Valor Valor especificado em cada contrato pelas actividades contratadas.

Forma de Pagamento Modo como será pago as actividades (mensal, trimestral ou semestralmente).

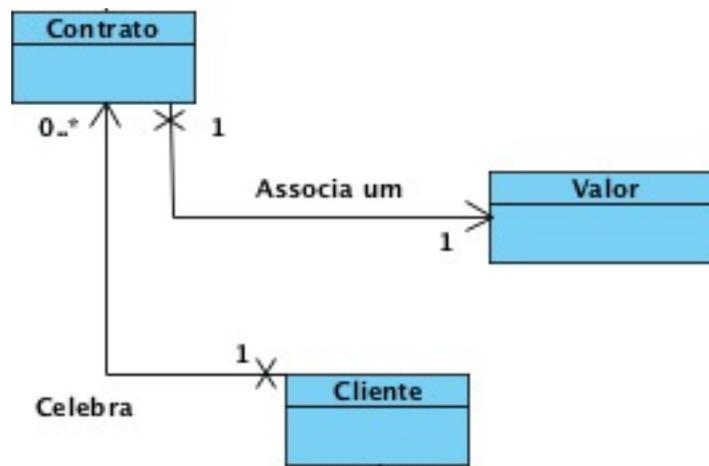


Figura 2.2: Modelo do Domínio - Relação entre Contrato, Cliente e Valor

Comissão Valor acrescentado ao preço das actividades para constar no contrato. Trata-se do lucro da GereComSaber

Cliente Actor que requisita actividades e celebra contratos.

Carteira de Cliente Conjunto de clientes.

2.2 Decisões

Sendo o modelo do domínio o primeiro contacto com o problema e com a área de negócio houve decisões que tiveram que ser tomadas. Algumas vieram-se a revelar erradas e foram corrigidas nos sucessivos refinamentos a que o modelo foi submetido. Outras prevaleceram até a fase actual, entre as quais:

- A carteira de cliente (ou fornecedor) existe mesmo que a GereComSaber não tenha qualquer cliente (ou fornecedor) associado. Daí a relação de 1 (Carteira) para * (fornecedores/clientes).
- A entidade **contrato** são os contratos em vigor, i.e., os que a data de termino é superior a data actual. É por esse motivo que a lista de contratos pode possuir (ou não) a entidade **contrato**.
- Considera-se que não sentido existe uma lista de actividades se não houver serviços.

Deve-se ainda realçar o facto de algumas relações terem sido ocultadas por uma questão de "implicações". A titulo de exemplo vejamos os pagamentos do cliente em relação ao contratos celebrados (*Figura 2.2*). Cada cliente celebra um ou mais contratos e cada contrato tem um valor associado. Por implicação cada cliente terá um ou mais valores associados.

Apesar desta conclusão, não vemos qualquer relação entre as entidades "valor" e "cliente".

2.3 Diagrama do modelo do domínio

Após esta primeira fase e tendo em consideração toda a informação a qual tivemos acesso (via cliente ou através do conhecimento adquirido na área de negócio) obtivemos o modelo presente na *figura 2.3*. Os use cases tiveram também um papel bastante importante, nomeadamente, na parte textual. Muitos dos substantivos usadas na descrição textual deram origem a entidades, restando depois apenas criar as relações existentes.

Deste modelo surgirá (nas próximas fases) um modelo de classes, altura em que haverá novo refinamento ou, por exemplo, inclusão de agregações.

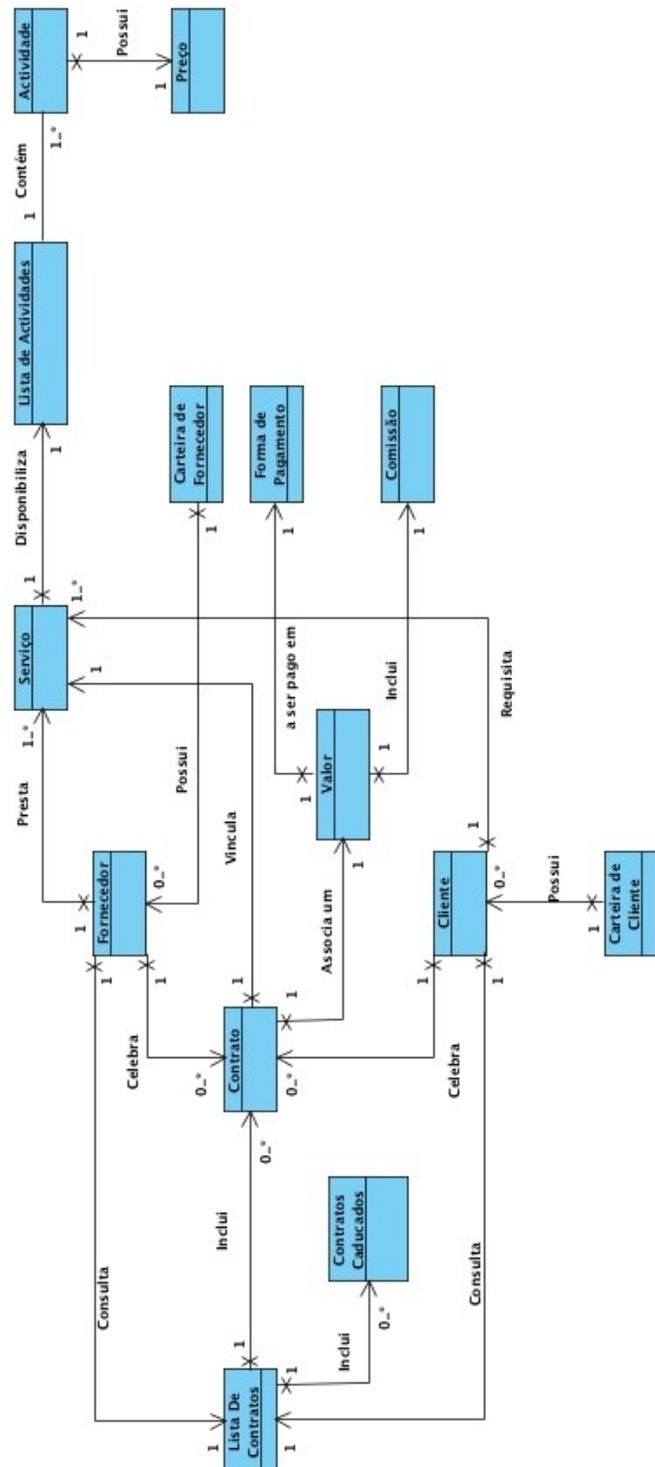


Figura 2.3: Modelo do Domínio

Capítulo 3

Use Cases

Os **Use Cases** identificam funcionalidades do nosso Sistema, que no seu completo conjunto, identificam todas as funcionalidades da GereComSaber bem como esta se deve comportar perante os diferentes tipos de utilizador. Servem também como forma de diálogo entre o Cliente e os responsáveis pelo desenvolvimento do software. Iremos de seguida falar do nosso diagrama de **Use Cases** subdividindo-os pelos actores que interagem com o Sistema.

3.1 Use Cases multi-utilizador

No nosso diagrama existem **Use Cases** que não são exclusivos de um só actor, estando disponíveis a todos.

3.1.1 Login e Logout

Entre eles surgem os de **Log In** e **Log Out**, que estão associados a todos os actores, estes são importantes para distinguir os diferentes privilégios e funcionalidades a que cada tipo de utilizador tem acesso, sendo que o **Log In** surge como pré-condição para todos os restantes **Use Cases**.

3.1.2 Pesquisar actividade

Todos os utilizadores têm a possibilidade de pesquisar por uma determinada actividade, este **Use Case** surge como uma generalização pois esta pesquisa pode ser efectuada de três formas diferentes:

Pesquisar por preço O utilizador introduz um intervalo de preço, ao qual será devolvida a lista de actividades cujo preço está compreendido nesse intervalo.

Pesquisar por código O utilizador introduz um código, sendo-lhe devolvida a actividade correspondente a esse código (se existir):

Pesquisar por descrição O utilizador introduz uma keyword ou um conjunto de keywords e é-lhe mostrado todas as actividades que na sua descrição esteja(m) a(s) keyword(s) fornecidas.

3.1.3 Consultar contratos

Outro **Use Case** comum a todos os utilizadores é o Consulta de Contratos, o qual surge também como uma generalização, pois damos a possibilidade de efectuar uma pesquisa por contratos em vigor ou por contratos finalizados. Esta separação surge pelo facto de em muitos casos ser muito mais eficaz apresentar apenas um dos casos. Vejamos, como exemplo, o caso em que o administrador após consultar os resultados parciais fica com dúvida em relação ao lucro alcançado. Neste caso é muito mais rápido para o administrador analisar uma lista com contratos finalizados (ou cancelados) do que analisar todos os contratos, visto que os contratos em vigor ainda não afectaram os resultados. O mesmo se passa se quiser saber qual o fornecedor que mais serviços presta no momento (apenas analisa os contratos em vigor). Para além disso é uma forma cómoda e funcional de guardar um histórico de actividades.

3.2 Administrador

Um administrador é qualquer pessoa que tenha privilégios para remover ou adicionar utilizadores e por outro lado, possa ter acesso a informações relativas ao funcionamento da GereComSaber, informações essas confidenciais estando ao alcance de um grupo muito restrito de pessoas como por exemplo as comissões.

3.2.1 Gerir Utilizadores

O administrador tem poder para decidir quem pode aceder ao sistema, podendo adicionar ou remover o acesso de um determinado utilizador. Por exemplo, ao adicionar um novo fornecedor à carteira de fornecedores, este passará a ter acesso ao sistema assim como um fornecedor que seja retirado da carteira de fornecedores deixará de ter acesso ao sistema. Por esta razão, temos aqui uma generalização que se decompõe em Adicionar Acesso e Remover Acesso.

3.2.2 Análise de Resultados

Para analisar os resultados da GereComSaber, o Administrador poderá querer ver os resultados de todo o período de vida da empresa ou apenas num determinado período, imaginemos que por exemplo, poderá querer ver apenas do mês anterior para elaborar um Balancete de Contabilidade. Como a Análise de Resultados, dá duas alternativas ao Administrador surge aqui novamente uma generalização.

3.2.3 Alterar Comissão

O administrador pode eventualmente querer alterar o valor da comissão de uma actividade, para tal, tem esta funcionalidade disponível que lhe permite alterar a comissão, isto vai implicar um aumento do preço para a contratação da actividade por parte do cliente, no entanto, a percentagem correspondente à comissão ganha pela GereComSaber será invisível ao cliente.

3.3 Fornecedor

Um Fornecedor que esteja ligado à GereComSaber disponibiliza uma lista de Serviços que presta, Serviços esses que possuem a si associados uma lista de Actividades. Por exemplo, o Serviço de Jardinagem fornecido pelo fornecedor X pode ter diversas actividades associadas como Cortar Relva, Aparar Cantos e Arranjar Buxos. Já o fornecedor Z, também possuidor do Serviço de Jardinagem já só fornece a opção de Cortar Relva.

3.3.1 Consultar Saldo

De forma a verificar o lucro que está a obter ao prestar os seus serviços aos clientes da GereComSaber e se está de facto a ser compensatório para ele a sua ligação com a GereComSaber, pode consultar o seu saldo que é dado desde que o Fornecedor está ligado à GereComSaber e também do ano corrente.

3.3.2 Editar Actividade

Pode haver necessidade por parte do Fornecedor em alterar um ou mais atributos de uma determinada actividade (exceptuando o código) e a aplicação concede esse privilégio ao Fornecedor.

3.3.3 Anular Serviço

A partir do momento em que um Fornecedor decidir deixar de prestar um determinado serviço tem a possibilidade de o fazer. A este **Use Case** está associado ao Anular Actividade com um estereótipo de include pois um serviço tem uma lista de actividades e se anulamos um serviço, então todas as actividades terão também de ser efectivamente anuladas portanto sempre que o Fornecedor anule um serviço, as actividades também terão sempre de ser anuladas.

3.3.4 Anular Actividade

Esta funcionalidade oferecida ao Fornecedor poderá ser chamada de duas formas distintas. Uma se o Fornecedor decidir explicitamente remover 1 ou mais actividades de um

	Actor Input	System Response
Flow of Events	1	Verifica se foi o fornecedor que activou directamente o Inserir Actividade.
	2	Inserir o código do serviço onde quer anexar as novas actividades.
	3	Certifica-se que o serviço existe.
	4	Disponibiliza os campos da nova actividade para o cliente preencher.
	5	Inserir os dados relativos a nova actividade.
	6	Confirma a inserção da actividade.
	7	Cria e insere a nova actividade na base de dados.
	8	Questiona o fornecedor para saber se este pretende inserir mais actividades para o actual serviço.
	9	Informa que não quer inserir mais actividades.
	10	Informa o fornecedor que o processo de inserção de actividade(s) foi concluído com sucesso.

Figura 3.1: Use Case "Inserir Actividade"(Main Flow)

determinado serviço ou então caso decida remover um serviço e aí serão removidas as actividades associadas a esse serviço.

3.3.5 Inserir Serviço

Está disponível para o Fornecedor a opção de inserir um novo serviço, o qual terá também de ter as suas respectivas actividades, esta relação é exprimida com um include porque Inserir Serviço chamará sempre o **Use Case** de Inserir Actividades.

3.3.6 Inserir Actividade

Esta funcionalidade tem uma particularidade quando comparada com as até aqui analisadas. Por um lado poderá ser executada pelo fornecedor de forma directa, i.e., sempre que há interesse de inserir uma nova actividade (num serviço já existente). Por outro lado existe a possibilidade de este Use Case ser executado por invocação do "Inserir actividade", i.e., sempre que o cliente insere um novo serviço e vai anexar actividades.

Na primeira situação não será necessário pedir o código do serviço dentro do "Inserir actividade"(visto que será inserido dentro do novo serviço) enquanto na segunda situação já é necessário. Esta questão de fluxo foi resolvida usando um caminho alternativo, no caso do Use Case "Inserir actividade"ter sido invocado por um outro Use Case, que nos faltar saltar o pedido de código de serviço.

Por outro lado o cliente poderá inserir múltiplas actividades para o mesmo serviço

Alternative 1: Não foi o fornecedor que o invocou.		Actor Input	System Response
	1		Salta para 4 (Main Flow).
Alternative 3: O serviço não existe.		Actor Input	System Response
	1		Informa o fornecedor que o código inserido não está associado a qualquer serviço.
	2		Volta para 2 (Main Flow).
Exception 6: O fornecedor cancela.		Actor Input	System Response
	1		Informa o fornecedor que o processo de inserção de actividade foi cancelado com sucesso.
Exception 7: Não foi possível inserir a actividade.		Actor Input	System Response
	1		Informa o fornecedor que o processo de inserção de actividade não foi concluído, indicando o motivo.
Alternative 9: O cliente indica que quer inserir mais actividades.		Actor Input	System Response
	1		Volta para 4 (Main Flow).

Figura 3.2: Use Case "Inserir Actividade"(Alternative & Exception)

sem que o Use Case tenha que ser invocado novamente. Este facto deve-se ao facto de o sistema questionar, no final de cada inserção, o cliente para saber se este pretende inserir mais actividades. Em caso afirmativo (e devido a um caminho alternativo) o use case voltará a ser invocado, excepção feita para o pedido de código do serviço, visto que será no mesmo anteriormente seleccionado. Estamos, de certa forma, na presença de um ciclo que só acaba quando o cliente acaba as inserções.

Na figura 3.1 é possível ver o fluxo, enquanto que na figura 3.2 visualiza-se os caminhos alternativos

3.4 Cliente

3.4.1 Consulta Histórico

O cliente tem acesso à visualização de todo o seu histórico na GereComSaber, ou seja, pode ver todos os contratos estabelecidos quer estejam em vigor ou já tenham expirado, quanto já gastou em serviços contratados, etc.

3.4.2 Cancelar Serviço

Se um cliente entender já não precisar das actividades contratadas relativas a um serviço, pode fazê-lo com esta funcionalidade. A relação de include com o **Use Case** de Cancelar Actividade está presente, pois, cancelando um serviço, todas as actividades subjacentes terão de ser canceladas também.

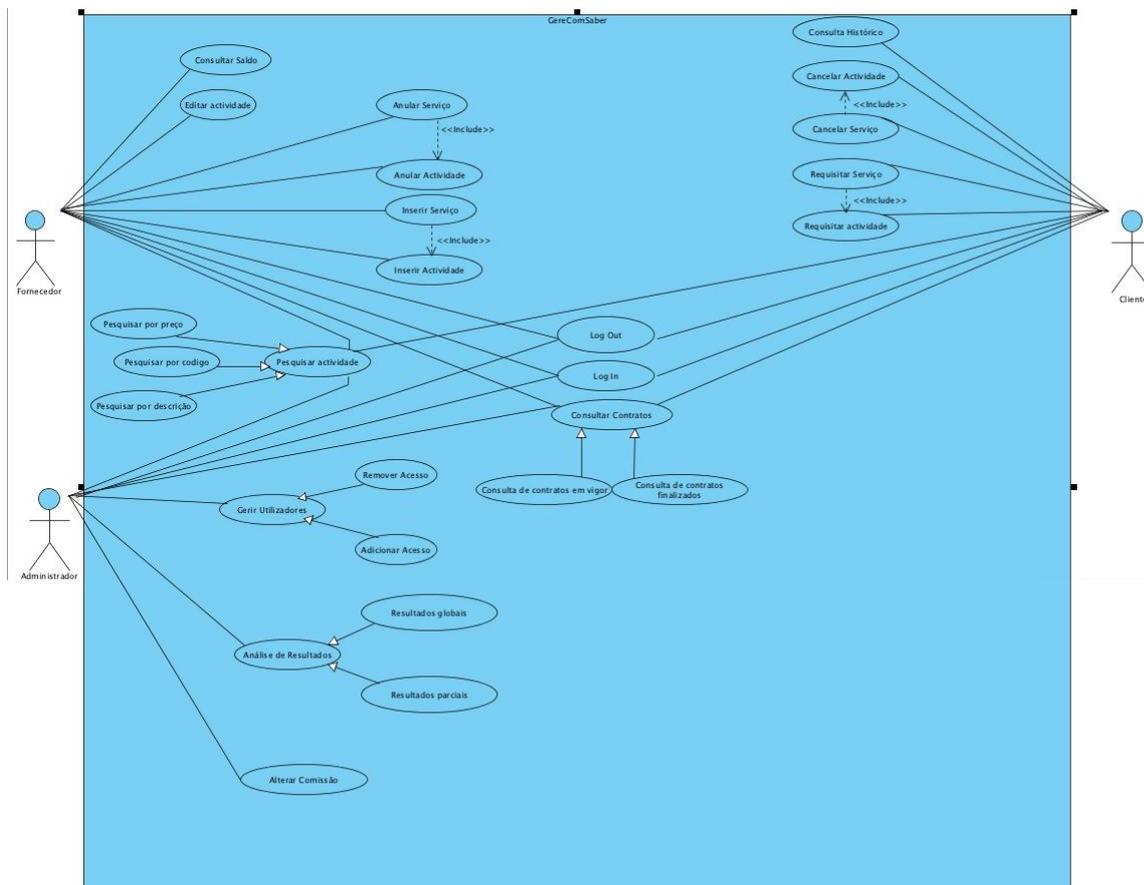


Figura 3.3: Use Cases

3.4.3 Cancelar Actividade

Para além deste **Use Case** poder ser invocado caso um cliente decida cancelar um serviço, outra alternativa seria decidir apenas remover uma ou mais actividades desde que já tenham sido contratadas e que o contrato ainda se mantenha em vigor.

3.4.4 Requirir Serviço

O cliente pode, através da aplicação, requisitar um novo serviço e ao fazê-lo terá também de escolher as actividades dentro desse serviço que pretende contratar. Esta relação implica que haja uma relação de include com o Requirir Actividade.

3.4.5 Requirir Actividade

Para além da situação acima descrita em que o cliente pode chegar a este **Use Case**, o cliente pode também requisitar uma actividade que ainda não tenha contratado sendo em ambos os casos celebrado um novo contrato.

Capítulo 4

Conclusão

Para este projecto os objectivos são claros e foram, de uma forma geral e para esta primeira etapa, cumpridos. A única dificuldade passou pelo facto de nunca termos tido, no seio do grupo, qualquer contacto com o UML ou com este tipo de modelação de sistemas. Nesse sentido as aulas práticas foram uma mais valia, apesar de não serem por si só suficientes.

Podemos, desde de já, tirar ilações em relação ao Modelo do Domínio e aos Use Cases. Em relação ao primeiro concluímos que o mesmo tem uma importância muito elevada no desenvolvimento de um projecto, coisa que não era de facto perceptível antes do início do mesmo. A construção de um dicionário de termos, onde muitos deles serão com grande probabilidade (e só agora começamos a tomar conhecimento disso) classes do nosso projecto informático, e as relações entre esse dicionário ajudaram a ganhar uma visão global sobre o sistema. Este último aspecto é relevante na altura da idealização e construção dos Use Cases.

Os Use Cases vêm-nos ajudar na visualização das funcionalidades do sistema e na indicação de quem são as pessoas (actores) que as podem usar. O principal obstáculo foi o facto de termos que garantir um fluxo coerente em Use Cases onde havia pelo menos um **include**, situação onde quer um determinado Use Case fosse invocado directamente pelo utilizador quer fosse invocado por um outro Use Case o fluxo teria que ser sequencialmente correcto. Excepção feita para este ponto o desenvolvimento desta etapa decorreu sem sobressaltos.

Nas próximas fases serão implementados o diagrama de classes (evolui do modelo do domínio) e os diagramas de sequência (provêm da análise da descrição textual dos Use Cases).