

Universidade do Minho
Licenciatura em Engenharia Informática

Disciplina de Desenvolvimento de Sistemas de Software
Trabalho Prático
Fase 2

Ano Lectivo de 2009/10



escola de engenharia



departamento de
informática

GereComSaber

Grupo 15

Cláudio Manuel Rigueiro Pires - 49347
Luciano Constantino Bessa Alves - 49378
Tomás Cardoso Peixoto Rito - 49411

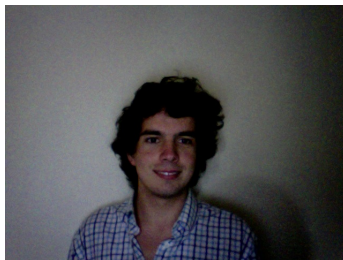
18 de Dezembro de 2009



Claudio Pires - 49347



Luciano Alves - 49378



Tomás Rito - 49411

Conteúdo

1	Introdução	7
1.1	Contextualização	7
1.2	Apresentação do Caso de Estudo	7
1.3	Objectivos	7
2	Análise de requisitos	8
2.1	Procedimentos	8
2.1.1	Proposta inicial do sistema GereComSaber	8
2.1.2	Proposta revista do sistema GereComSaber	9
2.2	Use Cases	9
2.2.1	Sistema GereComSaber	9
2.2.2	Subsistema Online (Descontinuado)	11
2.3	Descrição textual dos Use Cases	12
2.4	Modelo do domínio	16
2.4.1	Modelo domínio inicial	16
2.4.2	Modelo domínio revisto	17
2.5	Diagramas de sequência	18
2.6	Diagramas de classes	25
2.6.1	Diagrama de classes conceptual	26
2.6.2	Especificação das classes	26
3	Conclusão	27

Lista de Figuras

2.1	Sistema GereComSaber	8
2.2	Use Case do sistema escritório	10
2.3	Use Case gerir contas	10
2.4	Use Case gerir fornecedores	10
2.5	Use Case gerir serviços	11
2.6	Use Case gerir clientes	11
2.7	Use Case do sistema online	12
2.8	Descrição textual do Use Case 'Consultar serviços'	12
2.9	Descrição textual do Use Case 'Adicionar serviço'	13
2.10	Descrição textual do Use Case 'Remover serviço'	13
2.11	Descrição textual do Use Case 'Adicionar cliente'	13
2.12	Descrição textual do Use Case 'Consultar clientes'	14
2.13	Descrição textual do Use Case 'Editar dados cliente'	14
2.14	Descrição textual do Use Case 'Visão geral'	14
2.15	Descrição textual do Use Case 'Ver extracto'	14
2.16	Descrição textual do Use Case 'Estatísticas'	14
2.17	Descrição textual do Use Case 'Autenticar utilizador'	15
2.18	Descrição textual do Use Case 'Consultar fornecedores'	15
2.19	Descrição textual do Use Case 'Consultar serviços cliente'	15
2.20	Descrição textual do Use Case 'Adicionar fornecedor'	15
2.21	Descrição textual do Use Case 'Remover fornecedor'	16
2.22	Modelo do domínio inicial	17
2.23	Modelo do domínio revisto	18
2.24	Diagrama de Sequência do Use Case 'Consultar serviços'	18
2.25	Diagrama de Sequência do Use Case 'Adicionar serviço'	19
2.26	Diagrama de Sequência do Use Case 'Remover serviço'	20

2.27 Diagrama de Sequência do Use Case 'Adicionar cliente'	21
2.28 Diagrama de Sequência do Use Case 'Consultar clientes'	21
2.29 Diagrama de Sequência do Use Case 'Editar dados cliente'	22
2.30 Diagrama de Sequência do Use Case 'Visão geral'	22
2.31 Diagrama de Sequência do Use Case 'Ver extracto'	22
2.32 Diagrama de Sequência do Use Case 'Estatísticas'	23
2.33 Diagrama de Sequência do Use Case 'Autenticar utilizador'	23
2.34 Diagrama de Sequência do Use Case 'Consultar fornecedores'	24
2.35 Diagrama de Sequência do Use Case 'Consultar serviços cliente'	24
2.36 Diagrama de Sequência do Use Case 'Adicionar fornecedor'	25
2.37 Diagrama de Sequência do Use Case 'Remover fornecedor'	25
2.38 Diagrama de classes conceptual	26
2.39 Especificação das classes do sistema	26

Resumo

Este relatório tem como objectivo descrever de forma completa todo o processo de desenvolvimento do sistema GereComSaber. Este sistema pretende ser um sistema de gestão do serviço em condomínios fechados, de uma empresa encarregada de assegurar diversos serviços em casa dos condóminos. Para desenvolver este sistema foi seguido o processo RUP, Rational Unified Process, ou seja, todo o processo de captura de requisitos e de documentação irá ser efectuado usando a linguagem de modelação UML enquanto toda a fase de concepção irá ter uma abordagem orientada a objectos. O projecto em si será dividido em três fases, e no final de cada uma delas terão que ser cumpridos determinados objectivos para que se possa seguir para a próxima fase de forma correcta e para se verificar que o projecto está no bom caminho. Tem-se assim uma abordagem ao modelo incremental que é típico do RUP. O próprio relatório será construído ao longo das fases de maneira que na última fase, este deverá estar completamente finalizado.

1 Introdução

1.1 Contextualização

No âmbito da disciplina Desenvolvimento de Sistemas de Software, 3º ano do curso de Engenharia Informática da Universidade do Minho foi-nos pedido que desenvolvêssemos um projecto para auxílio de uma empresa - GereComSaber.

1.2 Apresentação do Caso de Estudo

O GereComSaber é um sistema que tem como objectivo ajudar uma empresa na gestão de um condomínio, para isso esta empresa necessita de interagir directamente com outro tipo de empresas: empresas de jardinagem, limpeza de interiores, cozinha, etc. A GereComSaber, de forma a satisfazer os seus clientes ao máximo, deverá ser capaz de articular os diversos serviços disponibilizados pelas variadas empresas e pô-los à disponibilidade dos clientes consoante as suas exigências, e as suas disponibilidades financeiras.

1.3 Objectivos

Como principal objectivo deste projecto é dinamizar o a realização de novos trabalhos nos nossos futuros empregos de modo que encaremos esta primeira fase, análise de requisitos, como uma das fases mais importantes da realização de qualquer projecto, algo que até agora não tem acontecido. Em segundo plano temos plena noção que uma boa fase de análise nos levará a uma muito mais fácil e correcta implementação do projecto em si.

2 Análise de requisitos

2.1 Procedimentos

2.1.1 Proposta inicial do sistema GereComSaber

Para esta primeira fase foi necessário investigarmos todas as operações presentes dentro do sistema GereComSaber, para isso analisamos cuidadosamente o enunciado do projecto para recolhermos a informação mais relevante e essencial ao software. Depois de reunirmos a maior parte da informação sobre este tipo de sistemas o grupo reuniu-se e de forma a decidir o que seria de facto o sistema GereComSaber e estruturar os requisitos que o nosso futuro software teria de seguir. Concluído isto, iniciou-se um novo projecto no Software Visual Paradigm e estruturamos os requisitos anteriormente formulados através de use cases com as respectivas descrições textuais. Decidimos organizar o sistema de informação em 2 subsistemas: Online e Escritório. Fizemos isto para que cada um dos sistemas seja independentemente do outro com o objectivo de assegurar sempre um número significativo de serviços mesmo que um subsistema esteja interrompido.

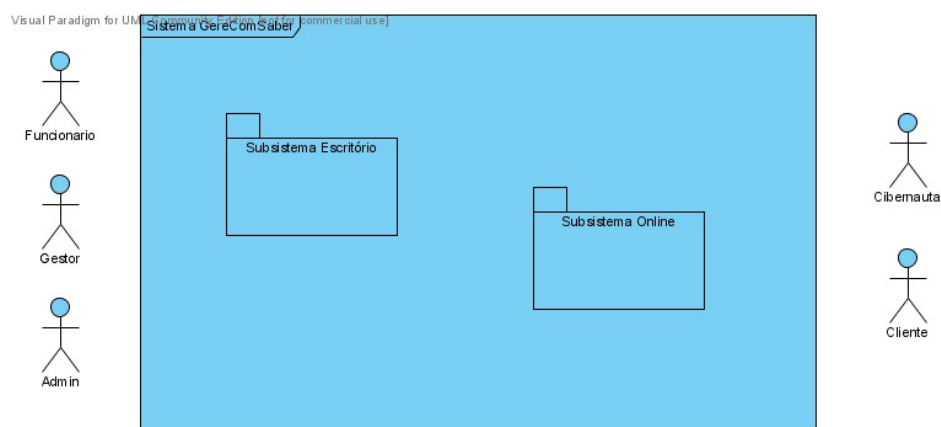


Figura 2.1: Sistema GereComSaber

2.1.2 Proposta revista do sistema GereComSaber

Decidimos efectuar nesta segunda fase do projecto, alterações no sistema. Apesar de ser interessante possuir um sistema online para os clientes gerirem os seus serviços remotamente através de suas casas, isto iria implicar algumas falhas no sistema ao nível da segurança e também fiabilidade. Visto que não estamos a contar usar bases de dados relacionais mas sim recorrer a ficheiros para armazenar todas as informações, seria muito complicado arranjar uma solução razoável para ambos os sistemas. Esta decisão de usar ficheiros deve-se ao facto do projecto em questão ser focado em todo o planeamento e análise de requisitos do sistema e não no desenho de uma base de dados relacional. Isto iria implicar muitas tabelas diferentes, normalização das mesmas, servidor de base de dados, assim como relembrar a sintaxe dos comandos SQL. Optámos por isso, pelo uso de ficheiros, muito provavelmente através do mecanismo de ObjectStreams do Java de guardar e obter objectos. Assim, os requisitos do sistema são cumpridos na mesma e a implementação será muito mais acessível, tendo apenas o sistema escritório, que passaremos a designar de sistema GereComSaber.

2.2 Use Cases

Um Use Case descreve a funcionalidade proposta para um novo sistema de software, que está a ser projectado. Segundo Ivar Jacobson, podemos dizer que um Use Case é um "documento narrativo que descreve a sequência de eventos de um actor que usa um sistema para completar um processo". Podemos então dizer que um Use Case é uma espécie de classe manifestada por sequências de mensagens intercambiáveis entre os sistemas e um ou mais actores. Para representar um Use Case utilizamos uma elipse com o nome do Use Case lá dentro, ou abaixo dela. Um Use Case é, então, uma unidade significativa de um trabalho, como o 'login', 'registo no sistema', etc.

Vamos então expôr todos os diagramas de Use Case do sistema GereComSaber e também do subsistema online que será descontinuado, isto é, não será implementado.

2.2.1 Sistema GereComSaber

A empresa possui um software que auxilia e assegura que os serviços em casa dos condóminos são efectuados. Através deste os funcionários da empresa podem gerir os clientes e serviços e os gestores ficam encarregues das contas e dos fornecedores da empresa. Todos estes actores terão um login que lhes permite aceder ao sistema e apenas às operações necessárias, não sendo possível portanto, que por exemplo um funcionário consiga ver as contas da empresa com o seu login ou um gestor conseguir ver gerir clientes e serviços.

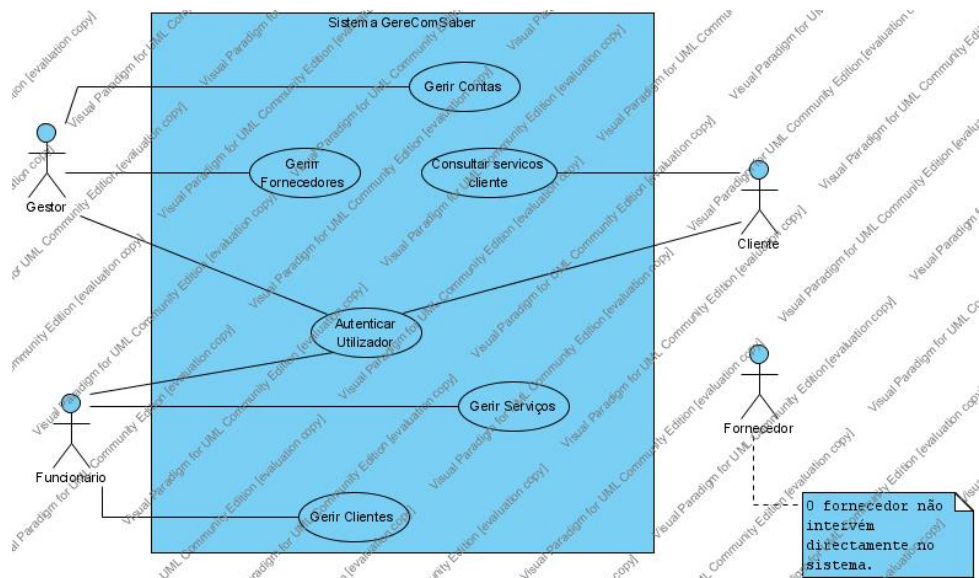


Figura 2.2: Use Case do sistema escritório

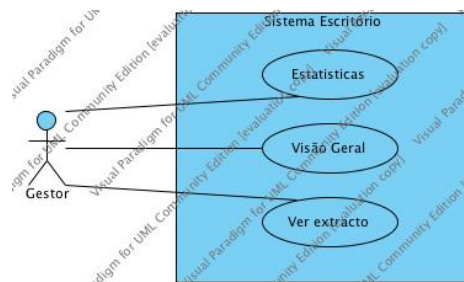


Figura 2.3: Use Case gerir contas

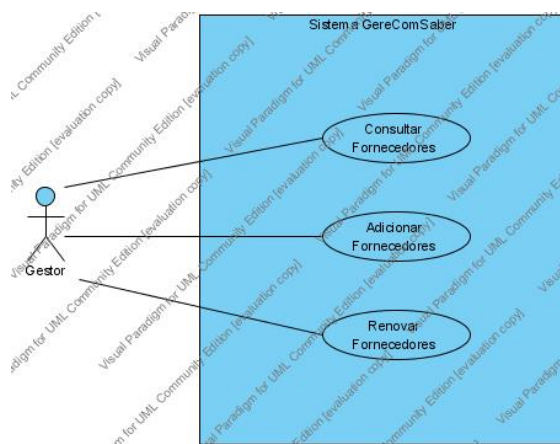


Figura 2.4: Use Case gerir fornecedores

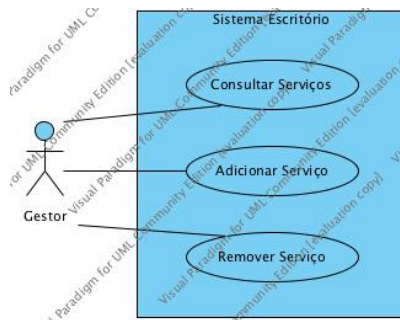


Figura 2.5: Use Case gerir serviços

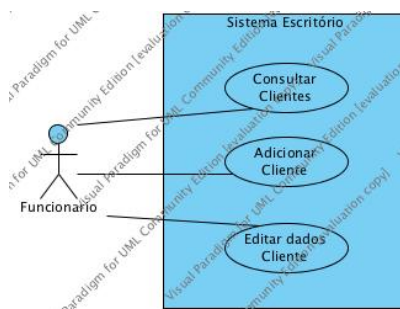


Figura 2.6: Use Case gerir clientes

2.2.2 Subsistema Online (Descontinuado)

Os clientes da empresa GereComSaber tem oportunidade de gerir e o seu serviço remotamente, isto é, sem a necessidade de deslocação ao escritório da empresa, pois esta disponibiliza um software para esse fim. Através deste os clientes podem não só consultar os serviços contratados mas também acrescentar ou remover serviços. Para isso é necessário que introduzam o login correctamente disponibilizado pela empresa quando se tornaram clientes desta. Embora seja uma ferramenta útil, existem certas operação que apenas poderão ser executadas no escritório. Como já explicamos anteriormente, decidimos excluir este subsistema, por isso, não será implementado. O diagrama é o seguinte:

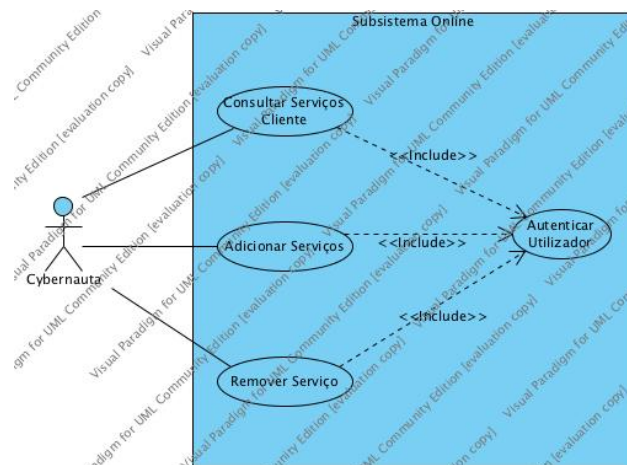


Figura 2.7: Use Case do sistema online

2.3 Descrição textual dos Use Cases

Estas descrições resultam de tomadas de decisão conjuntas entre clientes, utilizadores e analistas, sendo documentos que fixam responsabilidades funcionais do sistema a desenvolver e servem para planear o projecto (tempo, dinheiro, entre outros), sendo ainda mais úteis na concepção e até mesmo na implementação. Na descrição textual de todos os diagramas de Use Cases usámos o formato standard do Visual Paradigm, pois é bastante explícito e compreensível (mais que puro texto), e também não existe nenhum padrão "oficial" para esta tarefa.

Super Use Case		
Author		
Date		
1/Dez/2009 2:35:18		
Brief		
Description		
Consultar lista serviços por categoria e respectivos fornecedores e preços		
Preconditions		
Estar autenticado como funcionário		
Existirem serviços no sistema		
Post-conditions		
Flow of Events	Actor Input	System Response
	1 Consultar serviços	
	2	Mostra categorias de serviços
	3 Selecciona categoria	
	4	Filtra serviços correspondentes à categoria
	5	Mostra serviços filtrados
	6 Selecciona serviço	
	7	Mostra fornecedores e que prestam esse mesmo serviço com os respectivos preços

Figura 2.8: Descrição textual do Use Case 'Consultar serviços'

Author		
Date	2/Dez/2009 2:42:53	
Brief Description	Adicionar um serviço a um cliente	
Preconditions	Existirem serviços no sistema Existirem clientes no sistema Estar autenticado como cliente, funcionario	
Post-conditions	Sucesso: Um novo serviço é adicionado à lista de serviços de um dado cliente Insucesso: Não são adicionados serviços	
Flow of Events		Actor Input System Response
	1	Adicionar Serviço
	2	
	3	Introduz nome ou código do cliente desejado
	4	
	5	Verifica se existe cliente
	6	Mostra categorias de serviços
	7	
	8	Introduz categoria desejada
	9	
	10	Mostra serviços correspondentes à categoria introduzida
	11	
	12	Escolhe serviço
	13	Mostra fornecedores e respectivos preços do serviço
	14	
	10	Escolhe fornecedor pretendido
	11	Escolhe duração do serviço
	12	
	13	Calcula e apresenta montante a pagar
	14	
	13	Efectua pagamento
	14	
		Adiciona Serviço
Exception: 4		Actor Input System Response
Cliente não existe	1	
		Volta ao passo 3

Figura 2.9: Descrição textual do Use Case 'Adicionar serviço'

Author		
Date	2/Dez/2009 3:19:18	
Brief Description	Remover um determinado serviço de um cliente	
Preconditions	Existirem serviços no sistema Existirem clientes no sistema Estar autenticado como cliente, funcionario	
Post-conditions	Sucesso: Serviço pretendido é removido da lista de serviços do cliente Insucesso: Lista de serviços do cliente mantém-se	
Flow of Events		Actor Input System Response
	1	Remover Serviço
	2	
	3	Introduz nome ou código cliente desejado
	4	
	5	Verifica se existe cliente
	6	Mostra todos os serviços do cliente
	7	
	8	Escolhe serviço a remover
	9	
	10	Calcula e mostra montante a receber
	11	Adiciona o montante calculado na conta do cliente
	12	
	13	Remove serviço da lista de serviços do cliente
	14	
Exception: 4		Actor Input System Response
Cliente não existe	1	
		Volta para o passo 3

Figura 2.10: Descrição textual do Use Case 'Remover serviço'

Super Use Case	Gerir Clientes	
Author		
Date	1/Dez/2009 3:41:17	
Brief Description	Adicionar novo cliente ao sistema	
Preconditions	Estar autenticado como funcionario	
Post-conditions	Sucesso: Cliente é adicionado na lista de clientes do sistema Insucesso: Lista de clientes do sistema mantém-se inalterada	
Flow of Events		Actor Input System Response
	1	Adicionar Cliente
	2	
	3	Introduz dados nas caixas respectivas
	4	
	5	Mostra caixas de texto a pedir dados
	6	
	7	Valida dados
	8	Verifica existencia do cliente
	9	Cliente é adicionado à lista de clientes
Exception: 5		Actor Input System Response
Cliente já existe	1	
		Volta para passo 2
Exception: 4		Actor Input System Response
Dados introduzidos incorrectamente	1	
		Volta para passo 2

Figura 2.11: Descrição textual do Use Case 'Adicionar cliente'

Super Use Case	Gerir clientes	
Author		
Date	1/Dez/2009 3:50:32	
Brief Description	Consultar lista de clientes e detalhes	
Preconditions	Estar autenticado como funcionario	
Post-conditions		
Flow of Events		Actor Input
	1	Consultar clientes
	2	
	3	
	4	Selecciona um cliente específico
Exception: 2a-2 Nao existem clientes	5	
		System Response
		Carrega lista clientes
		Mostra lista clientes
		Mostra detalhes desse cliente
		System Response
	1	Manda mensagem de erro

Figura 2.12: Descrição textual do Use Case 'Consultar clientes'

Super Use Case	Gerir clientes	
Author		
Date	1/Dez/2009 4:06:20	
Brief Description	Alterar ou corrigir os dados de um cliente	
Preconditions	Estar autenticado como funcionario	
Post-conditions	Sucesso: Dados alterados Insucesso: Dados inalterados	
Flow of Events		Actor Input
	1	Editar dados cliente
	2	
	3	Introduz dados cliente
	4	
	5	
	6	Altera os campos desejados
Exception: 4 Cliente não existe	7	
		System Response
		Pede dados cliente
		Procura cliente na lista
		Apresenta caixas de texto com todos os campos
		Guarda alterações
		System Response
	1	Volta para passo 2

Figura 2.13: Descrição textual do Use Case 'Editar dados cliente'

Super Use Case	Gerir contas	
Author		
Date	1/Dez/2009 4:58:20	
Brief Description	Mostrar visão geral das contas da gereComSaber	
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	Visão geral
	2	
	3	
	4	
		System Response
		Acede aos campos da contabilidade
		Calcula resultados
		Mostra resultados

Figura 2.14: Descrição textual do Use Case 'Visão geral'

Super Use Case	Gerir contas	
Author		
Date	1/Dez/2009 5:17:52	
Brief Description	Extracto detalhado de todos os movimentos	
Preconditions	Estar autenticado como gestor	
Post-conditions		
Flow of Events		Actor Input
	1	Ver extracto
	2	
	3	
		System Response
		Acede ao extracto na contabilidade
		Mostra todos os movimentos do extracto

Figura 2.15: Descrição textual do Use Case 'Ver extracto'

Super Use Case	Gerir contas	
Author		
Date	1/Dez/2009 4:43:47	
Brief Description	Mostra algumas estatísticas uteis recorrendo a graficos	
Preconditions	Estar autenticado como gestor	
Post-conditions		
Flow of Events		Actor Input
	1	Consultar estatísticas
	2	
	3	
	4	
		System Response
		Acede as estatísticas na contabilidade
		Calcula e constroi valores
		Mostra gráficos

Figura 2.16: Descrição textual do Use Case 'Estatísticas'

Author			
Date	1/Dez/2009 4:14:15		
Brief Description	Autenticar utilizador no sistema		
Preconditions			
Post-conditions	Sucesso: Utilizador autenticado Insucesso: Ninguém autenticado		
Flow of Events		Actor Input	System Response
	1	Autenticar utilizador	
	2		Pede dados utilizador
	3	Introduz dados	
	4		Valida dados
Exception: 4 Falhou a autenticar		Actor Input	System Response
	1		Mostra mensagem de boas vindas
	2		Mostra mensagem de erro
			Volta para passo 2

Figura 2.17: Descrição textual do Use Case 'Autenticar utilizador'

Author			
Date	1/Dez/2009 4:28:08		
Brief Description	Consulta detalhada dos fornecedores da GereComSaber		
Preconditions	Estar autenticado como gestor Existirem fornecedores no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Consultar fornecedores	
	2		Aceder a lista de serviços
	3		Criar lista de fornecedores a partir da lista de serviços
	4		Mostrar fornecedores
	5	Escolhe fornecedor	
	6		Agrupa e mostra serviços prestados pelo fornecedor detalhadamente
	7		Mostra os fornecedores da lista

Figura 2.18: Descrição textual do Use Case 'Consultar fornecedores'

Super Use Case	Gerir clientes		
Author			
Date	18/Dez/2009 6:19:20		
Brief Description	Consulta os serviços requisitados por um cliente		
Preconditions	Estar autenticado como cliente ou funcionario Existirem clientes		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Consultar serviços cliente	
	2		Pede username
	3	Introduz username	
	4		Valida username
	5		Agrupa lista com serviços do cliente com o username introduzido
	6		Mostra a lista
Exception: 4		Actor Input	System Response
Erro ao validar	1		Volta ao passo 2

Figura 2.19: Descrição textual do Use Case 'Consultar serviços cliente'

Super Use Case	Gerir fornecedores		
Author			
Date	17/Dez/2009 17:36:04		
Brief Description	Adiciona fornecedor à GereComSaber		
Preconditions	Estar autenticado como gestor		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Adicionar fornecedor	
	2		Pedir dados
	3	Introduz nome fornecedor e dados do serviço	
	4		Valida dados introduzidos
	5		Cria novo serviço com esses dados
	6		Adiciona serviço à lista de serviços
Exception: 5 Dados inválidos		Actor Input	System Response
	1		Volta para passo 2

Figura 2.20: Descrição textual do Use Case 'Adicionar fornecedor'

Super Use Case	Gerir fornecedores	
Author		
Date	17/Dez/2009 18:25:54	
Brief Description	Remove um fornecedor e todos os seus serviços prestados	
Preconditions	Estar autenticado como gestor Existirem serviços no sistema	
Post-conditions		
Flow of Events		Actor Input
	1	Remover fornecedor
	2	
	3	Introduz nome fornecedor
	4	
	5	
	6	
	7	
	8	
	9	
		System Response
		Pede nome do fornecedor
		Verifica se existe fornecedor
		Remove serviços desse fornecedor da lista de serviços
		Procura no contracto de cada cliente da lista clientes quem possui serviços do fornecedor
		Remove serviços desse fornecedor, do contracto dos clientes anteriores
		Calcula novo montante a pagar
		Actualiza montante a pagar do contracto dos clientes envolvidos
Exception: 4 Fornecedor não existe		
		Actor Input
	1	
		System Response
		Volta ao passo 2

Figura 2.21: Descrição textual do Use Case 'Remover fornecedor'

2.4 Modelo do domínio

O termo domínio denota um conjunto de sistemas ou áreas funcionais dos vários sectores organizacionais de actividade, onde a existência de uma terminologia (sintaxe ou termo) deve estar inequivocamente associada a certos conceitos, o que muito simplifica a compreensão e, em consequência, a correcta comunicação entre quem tem que "definir contractos de informatização" e quem tem que "realizar tais contractos" no âmbito (domínio) de tais sistemas, áreas, etc. Deve-se modelar um domínio organizando e definindo todo o tipo de vocabulário fundamental do domínio do problema, em especial o que sobressai da análise de requisitos com os interlocutores numa tabela semântica, ou seja, um dicionário terminológico aceite e assinado por ambas as partes. As classes, atributos e relacionamentos servem em UML para modelar o domínio do sistema, tal como, com mais detalhe, servem para modelar os elementos de software. Por vezes confunde-se modelo de domínio com diagrama de classe, no entanto, o primeiro é conceptual enquanto que o segundo já corresponde a uma especificação/implementação com vista à codificação.

2.4.1 Modelo domínio inicial

Depois de agrupar um conjunto de palavras-chave do problema, procedemos à construção do modelo:

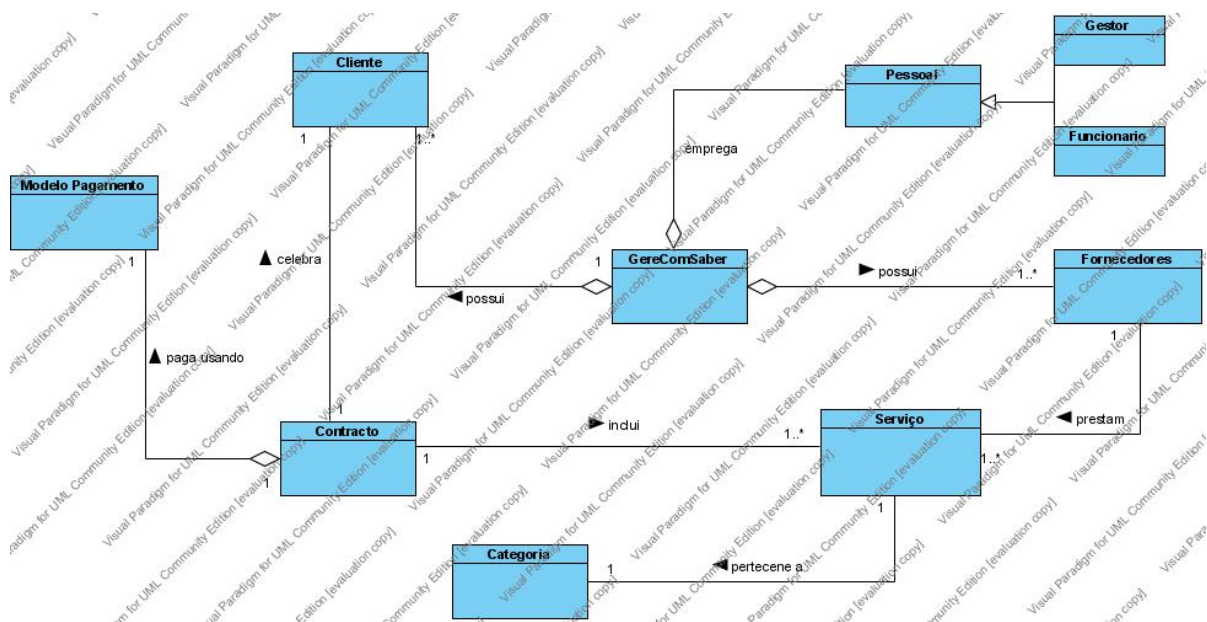


Figura 2.23: Modelo do domínio revisto

2.5 Diagramas de sequência

Muito importante para um correcto trabalho na linguagem de UML são os diagramas de sequência - diagramas que representam, de uma forma eficiente e interactiva, a sequência de processos, mais precisamente as mensagens passadas entre objectos num determinado contexto. Para o programador do futuro software o diagrama de sequência serve para dar uma certa lógica temporal à troca de mensagens entre os objectos existentes no sistema. No nosso projecto construímos os seguintes diagramas de sequência (um para cada Use Case):

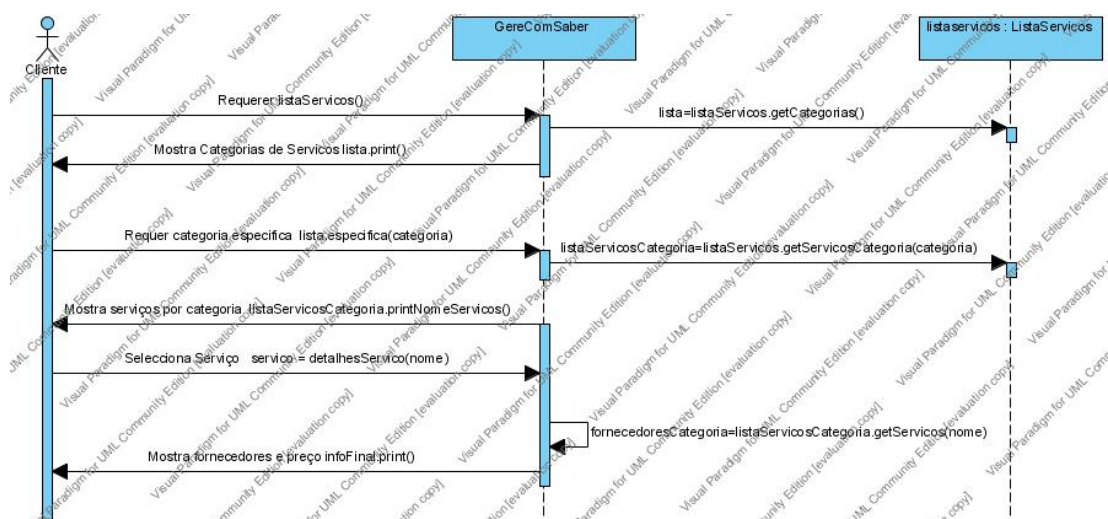


Figura 2.24: Diagrama de Sequência do Use Case 'Consultar serviços'

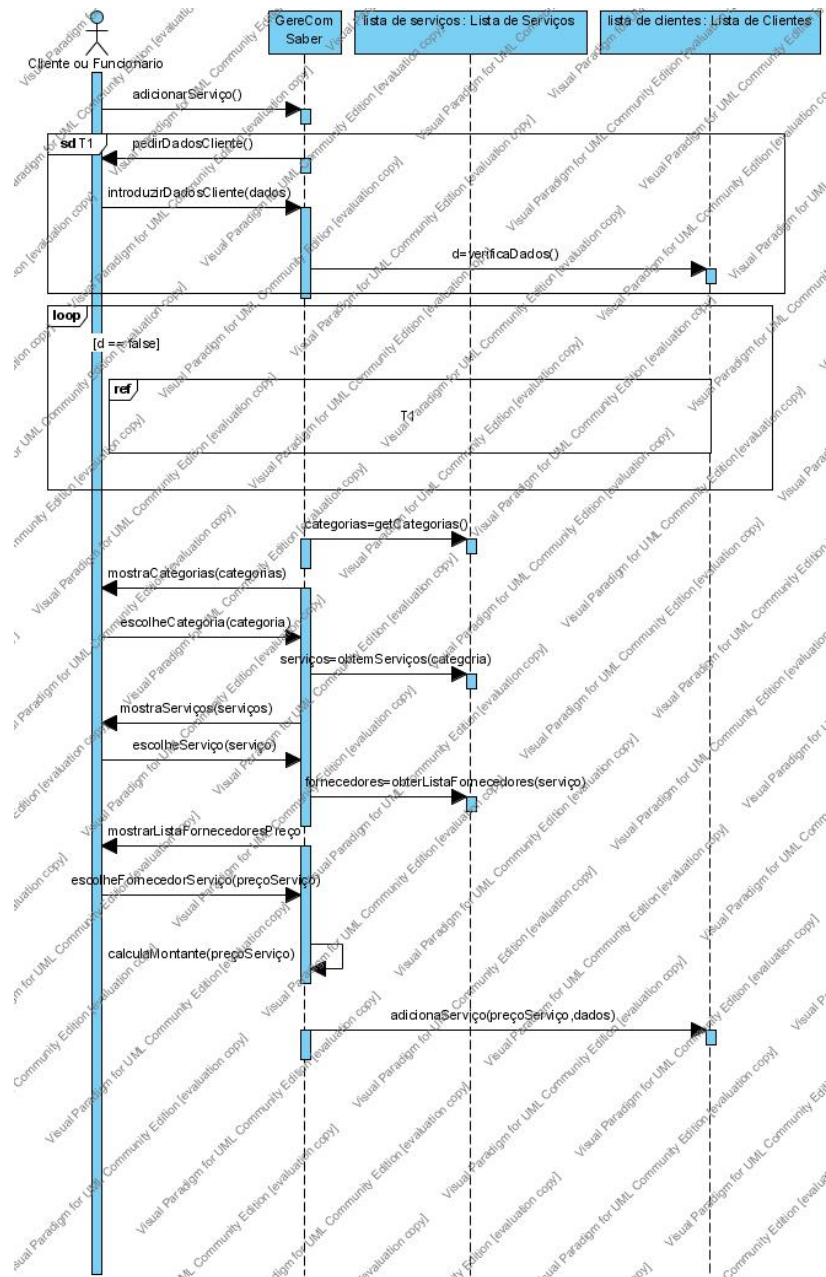


Figura 2.25: Diagrama de Sequência do Use Case 'Adicionar serviço'

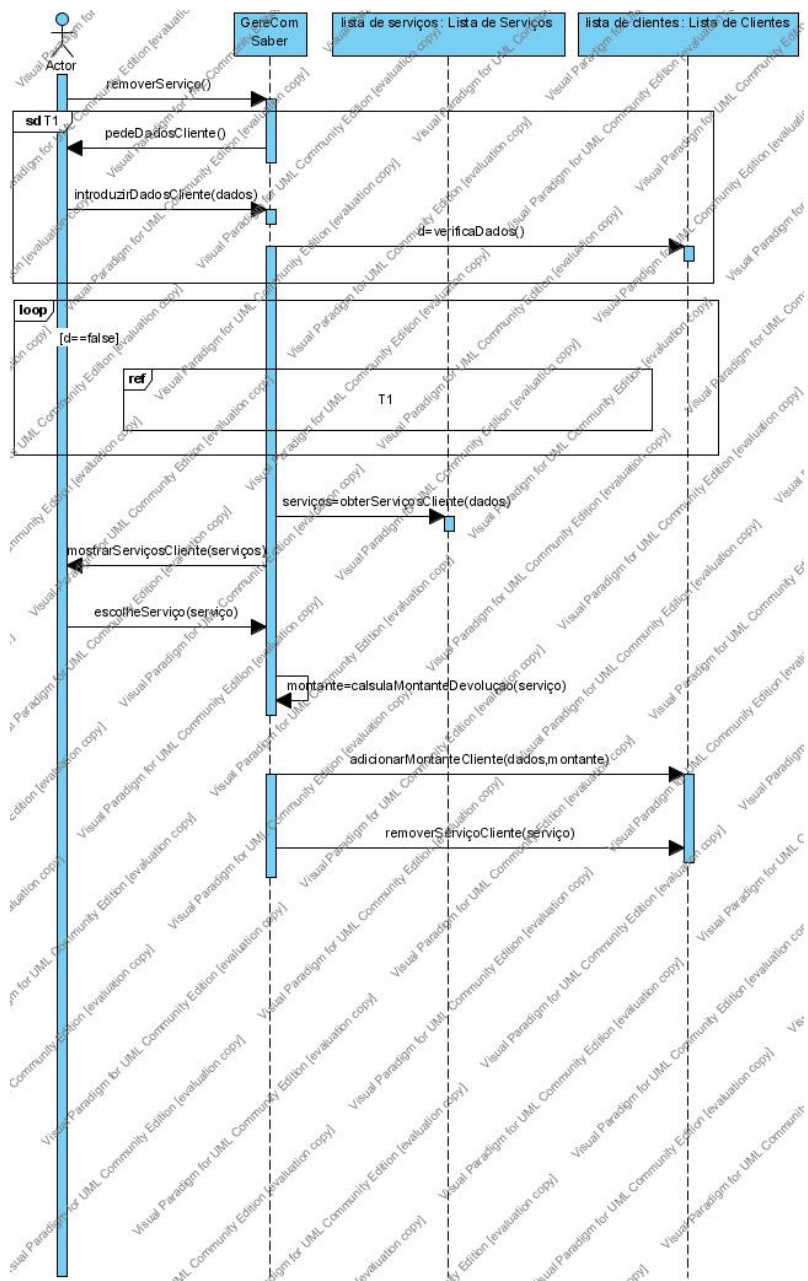


Figura 2.26: Diagrama de Sequência do Use Case 'Remover serviço'

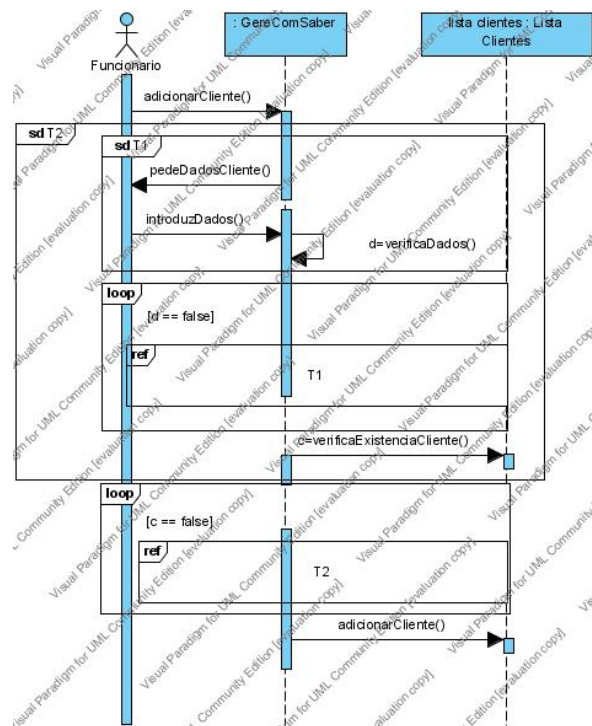


Figura 2.27: Diagrama de Sequência do Use Case 'Adicionar cliente'

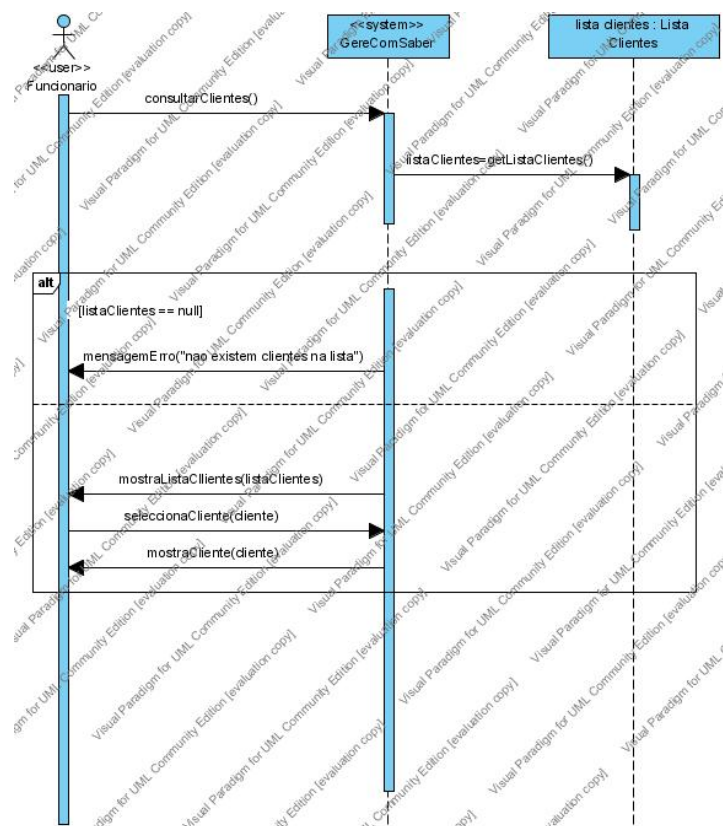


Figura 2.28: Diagrama de Sequência do Use Case 'Consultar clientes'

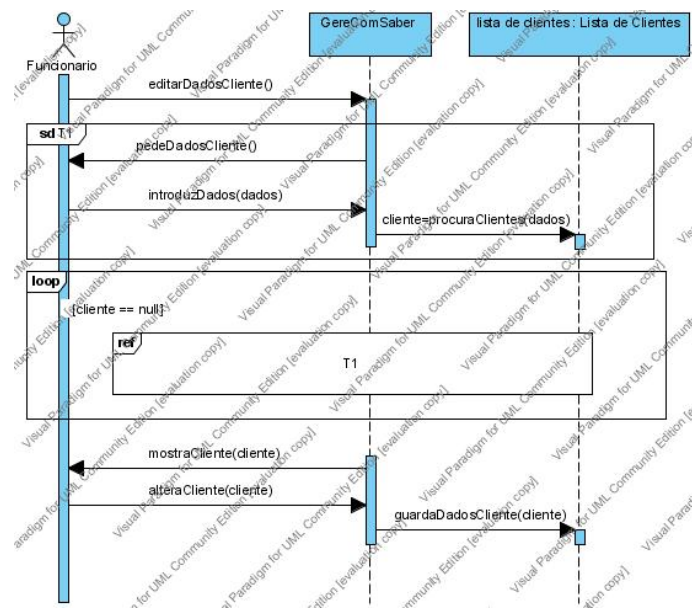


Figura 2.29: Diagrama de Sequência do Use Case 'Editar dados cliente'

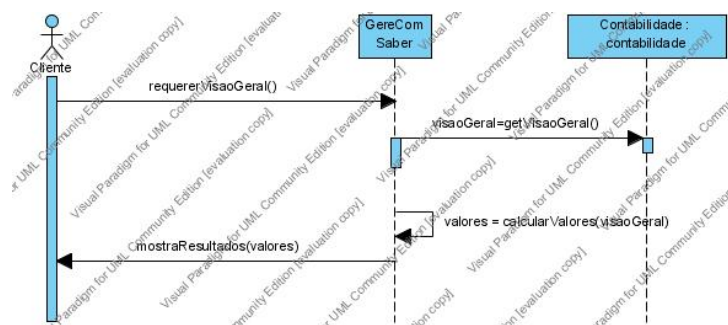


Figura 2.30: Diagrama de Sequência do Use Case 'Visão geral'

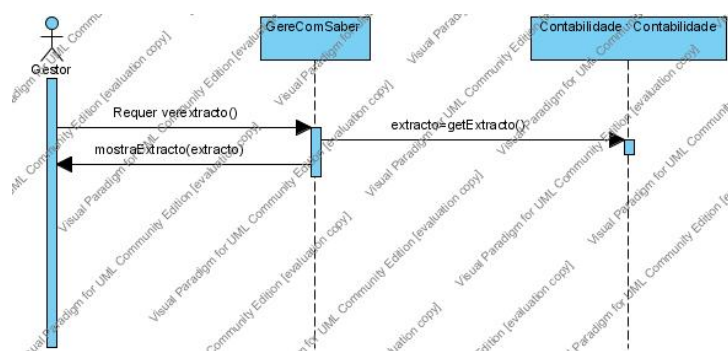


Figura 2.31: Diagrama de Sequência do Use Case 'Ver extracto'

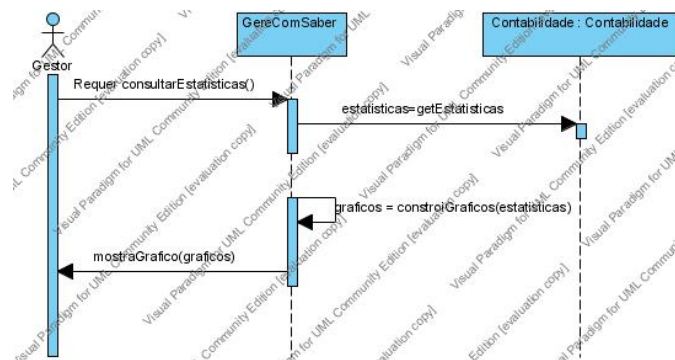


Figura 2.32: Diagrama de Sequência do Use Case 'Estatísticas'

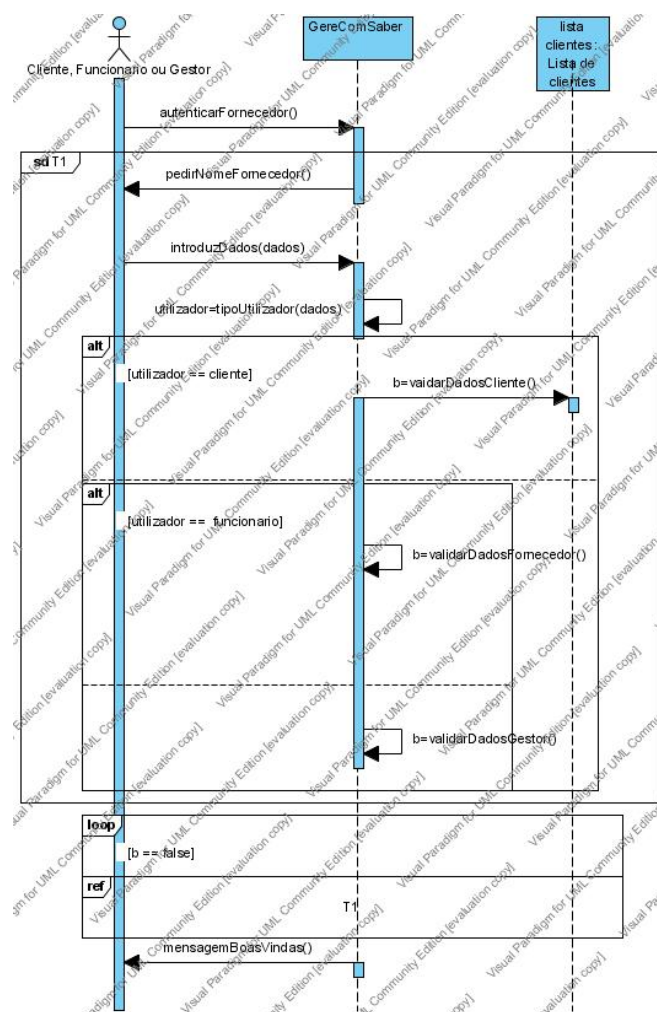


Figura 2.33: Diagrama de Sequência do Use Case 'Autenticar utilizador'

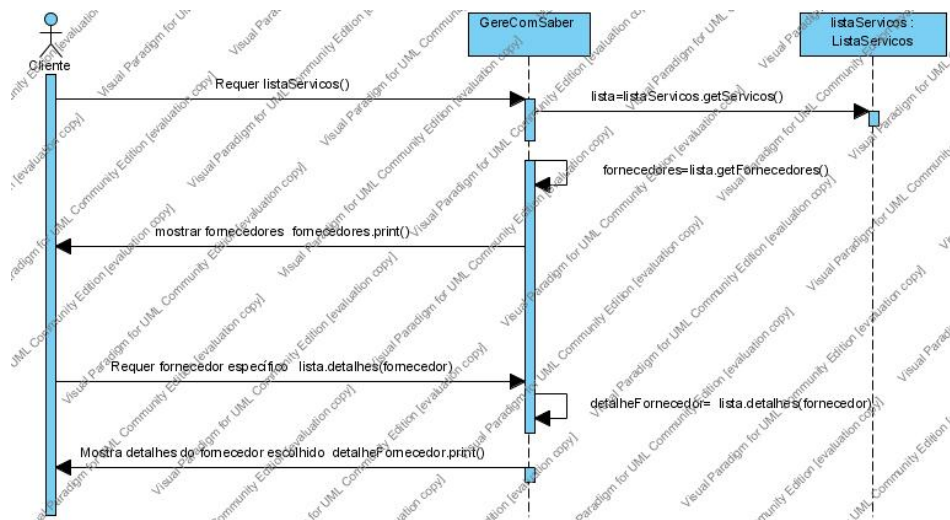


Figura 2.34: Diagrama de Sequência do Use Case 'Consultar fornecedores'

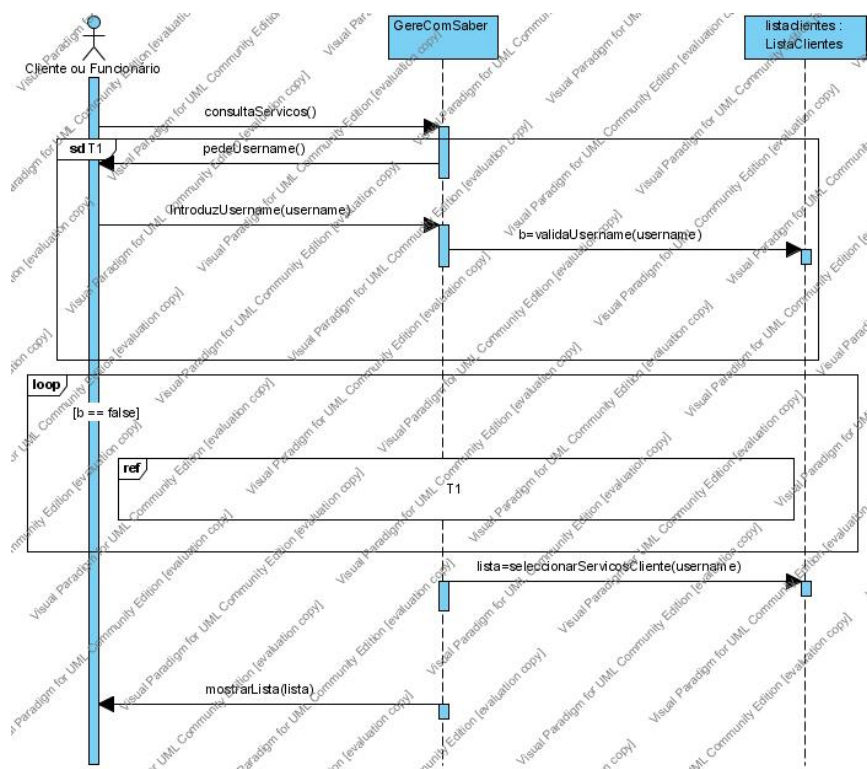


Figura 2.35: Diagrama de Sequência do Use Case 'Consultar serviços cliente'

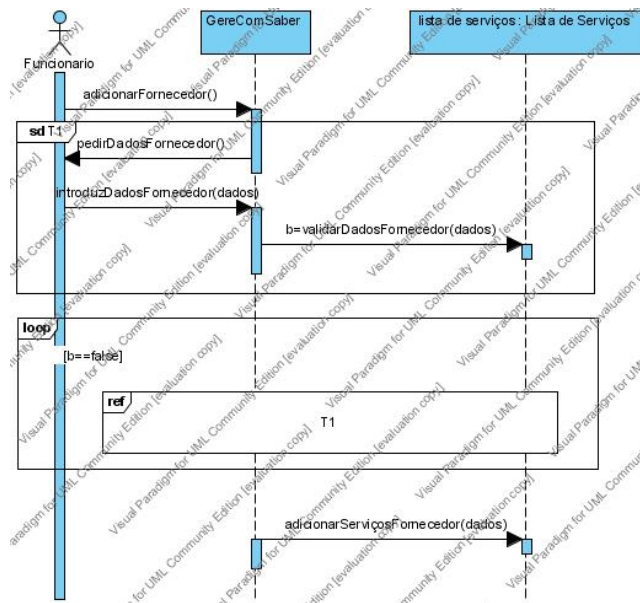


Figura 2.36: Diagrama de Sequência do Use Case 'Adicionar fornecedor'

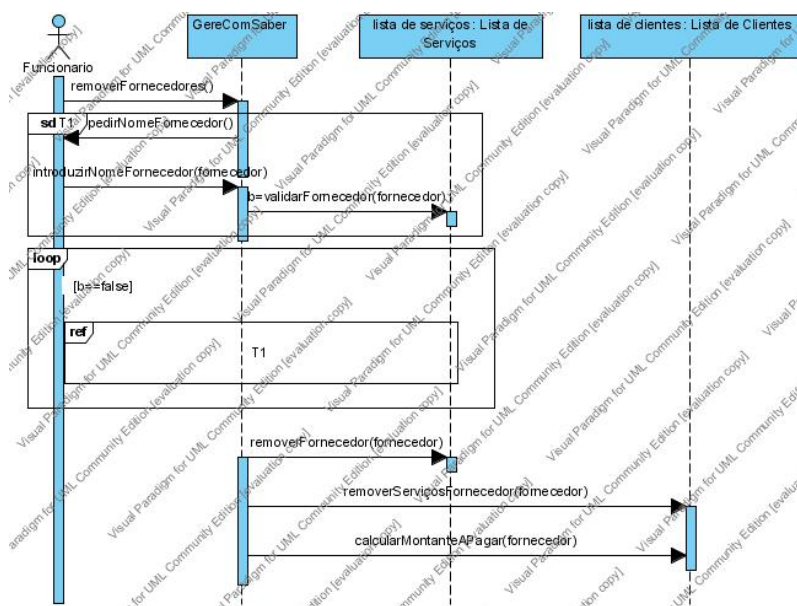


Figura 2.37: Diagrama de Sequência do Use Case 'Remover fornecedor'

2.6 Diagramas de classes

Em UML um diagrama de classes é uma representação de toda a estrutura do sistema e das relações das classes desse sistema. Um diagrama de classes bem modelado é fundamental para o sucesso do sistema. Para cada objecto do sistema, um diagrama de classes descreve a sua identidade, os seus relacionamentos com outros objectos, os seus atributos e as suas operações.

Diagramas de Classes que se tornam complexos devem ser divididos, e funcional ou semanticamente estruturados em Packages, que agrupam classes com tais afinidades. No entanto não achamos necessário fazer tal divisão no nosso sistema.

2.6.1 Diagrama de classes conceptual

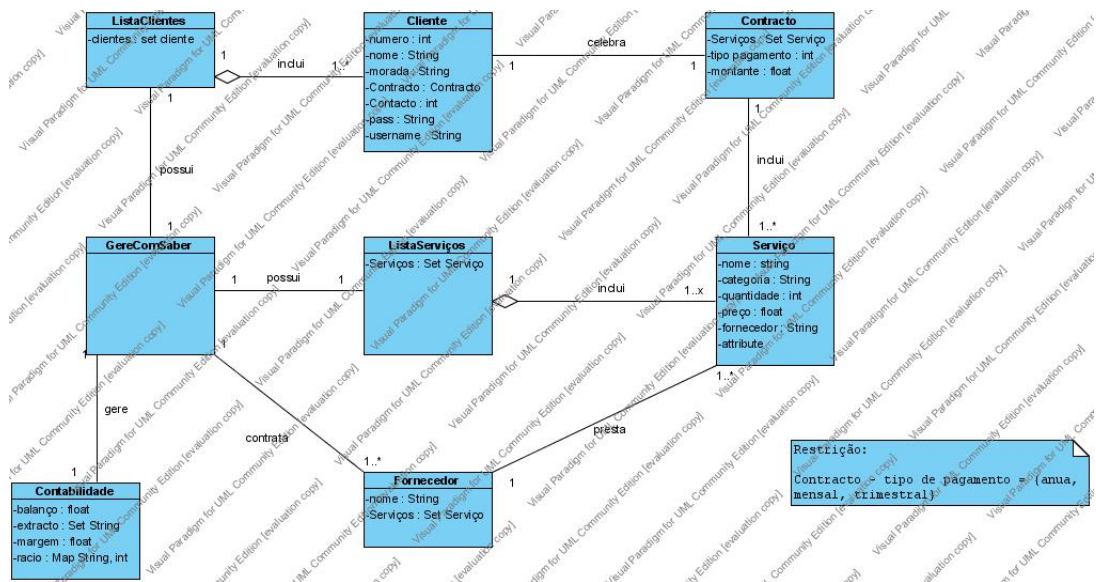


Figura 2.38: Diagrama de classes conceptual

2.6.2 Especificação das classes

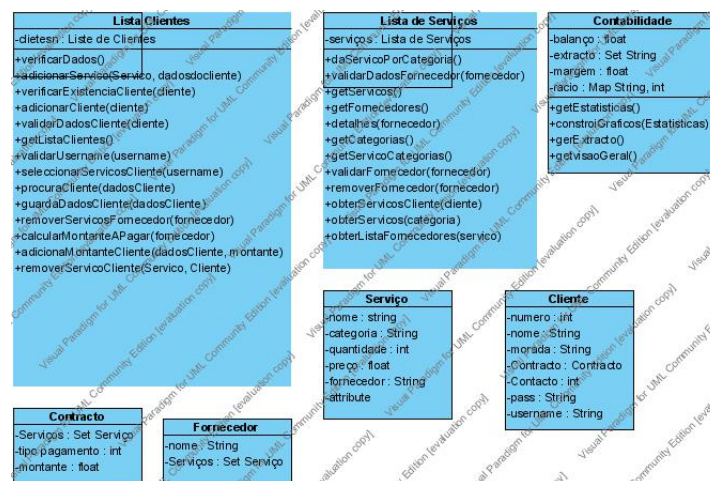


Figura 2.39: Especificação das classes do sistema

3 Conclusão

No fim da primeira fase de criação do sistema GereComSaber, concluímos que para tal tivemos de adquirir conhecimento, recorrendo a muito estudo e pesquisa no âmbito da modelação de linguagens, nomeadamente UML (Unified Modeling Language). A realização foi posta em prática no programa Visual Paradigm. O modelo foi cuidadosamente pensado, planeado e implementado, de forma a que futuramente, nas próximas fases, o sistema evolua de forma proveitosa. Pensamos ter cumprido os objectivos da primeira fase tal como estar no caminho certo para tratamento dos objectivos futuros. No entanto temos noção que este projecto é um projecto continuo, passível de ser optimizado posteriormente. Depois de terminar a segunda fase do projecto e melhorarmos a anterior, concluímos que este tipo de trabalhos, em UML, tem que ser feitos de uma forma progressiva, ou seja, é sempre necessário proceder à algumas alterações e refinamentos de etapas anteriores. Concluímos também a importância da análise de requisitos e de toda esta fase de planeamento, e acreditamos que estamos num bom caminho para uma rápida e mais fácil implementação em linguagem Java. No entanto, este modelo poderá ainda sofrer pequenos ajustes.