

Universidade do Minho
Licenciatura em Engenharia Informática

Disciplina de Desenvolvimento de Sistemas de Software

Ano Lectivo de 2009/10



escola de engenharia



departamento de
informática

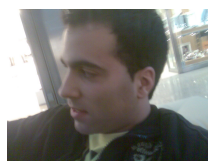
Gere Com Saber

André Barbosa - nº49357

David Leal - nº49321

José Miguel Cracel - nº49420

Tiago Oliveira - nº50129



Dezembro, 2009

Resumo

Além de conter uma introdução e uma conclusão este documento está dividido em outros quatro capítulos:

- Recursos utilizados;
- Modelo de Domínio;
- Diagrama de Use-cases;
- Diagrama de Sequência de Sistemas;
- Diagrama de Sequência de Sub-Sistemas;
- Diagrama de Classes;
- Diagrama de Sequência de Implementação;
- Decisões pertinentes.

No capítulo relativo aos recursos utilizados faz-se uma descrição da tecnologia usada (UML) bem como dos tipos de modelação levados a cabo nesta fase:

- modelo de domínio;
- diagrama de use-cases;
- diagrama de sequência;
- diagrama de classes.

No terceiro capítulo apresenta-se o modelo de domínio bem como uma explicação textual das entidades e relações que dele fazem parte.

No quarto capítulo podem-se ver os diagramas e sub-diagramas de use-cases e respectivos *flow of events*.

O quinto capítulo contém os mais simples diagramas de sequência e primeiros a serem construídos, os *Diagramas de Sequência de Sistema*.

No sexto capítulo há um avanço na especificação dos diagramas de sequência, passando agora a ser *Diagramas de Sequência de Sub-Sistemas*, onde as operações passam a ser suportadas por mais que um sistema.

O sétimo capítulo contém os *Packages* construídos para este sistema, que estão representados em *Diagramas de Classes* utilizando a notação UML.

No oitavo capítulo voltamos novamente aos diagramas de sequência, desta forma com muito mais pormenor, já utilizando funções definidas no *Diagrama de Classes* e mais sistemas para o suporte das operações. Temos assim representado o *Diagrama de Sequência de Implementação*.

O capítulo nono é sobre as decisões mais pertinentes que tomamos sobre o projecto. Estas tomadas de decisões envolvem a forma com tratamos:

- pagamentos;
- e contratos.

Conteúdo

Conteúdo	viii
1 Introdução	1
2 Recursos Utilizados	2
2.1 UML (Unified Modeling Language)	2
2.2 Modelo de Domínio	3
2.3 Diagrama de Use-Cases	3
2.4 Diagrama de Sequências	4
2.5 Diagrama de Classes	4
2.5.1 Relações entre Classes	5
3 Modelo de Domínio	6
4 Diagrama de Use-Cases	8
4.1 GCS	8
4.1.1 Gerir Funcionários	10
4.1.2 Gerir Clientes	12
4.1.3 Gerir Fornecedores	13
4.1.4 Gerir Serviços	16
4.1.5 Gerir Pagamentos	17
4.1.6 Gerir Actividades	19
4.1.7 Gerir Contratos	21
5 Diagramas de Sequência de Sistemas	24
5.1 Geral	24
5.1.1 Login	24
5.1.2 Consultar Contrato	25
5.1.3 Alterar Dados	26

5.1.4	Alterar Informação Pessoal	26
5.2	Clientes	27
5.2.1	Adicionar Cliente	27
5.2.2	Consultar Cliente	27
5.2.3	Editar Cliente	28
5.2.4	Remover Cliente	28
5.3	Fornecedores	29
5.3.1	Adicionar Fornecedor	29
5.3.2	Consultar Fornecedor	29
5.3.3	Editar Fornecedor	30
5.3.4	Remover Fornecedor	30
5.4	Funcionários	31
5.4.1	Adicionar Funcionário	31
5.4.2	Editar Funcionário	32
5.4.3	Remover Funcionário	33
5.5	Serviços	34
5.5.1	Adicionar Serviço	34
5.5.2	Remover Serviço	35
5.5.3	Consultar Lista de Serviços	35
5.6	Actividades	36
5.6.1	Adicionar Actividade	36
5.6.2	Alterar Preço de Actividade	37
5.6.3	Remover Actividade	37
5.7	Contratos	38
5.7.1	Criar Contrato	38
5.7.2	Alterar Contrato	39
5.7.3	Cancelar Contrato	40
5.7.4	Adicionar Actividade ao Contrato	40
5.7.5	Cancelar Actividade do Contrato	41
5.8	Pagamentos	41
5.8.1	Confirmar Pagamento de Contrato	41
5.8.2	Confirmar Pagamento de Acréscimo	42
5.8.3	Confirmar Pagamento de Estorno	42
6	Diagramas de Sequência de Sub-Sistemas	43
6.1	Geral	43

6.1.1	Login	43
6.1.2	Consultar Contrato	44
6.1.3	Alterar Dados	44
6.1.4	Alterar Informação Pessoal	45
6.2	Cientes	45
6.2.1	Adicionar Cliente	45
6.2.2	Consultar Cliente	46
6.2.3	Editar Cliente	46
6.2.4	Remover Cliente	47
6.3	Fornecedores	48
6.3.1	Adicionar Fornecedor	48
6.3.2	Consultar Fornecedor	48
6.3.3	Editar Fornecedor	49
6.3.4	Remover Fornecedor	50
6.4	Funcionários	51
6.4.1	Adicionar Funcionário	51
6.4.2	Editar Funcionário	52
6.4.3	Remover Funcionário	52
6.5	Serviços	53
6.5.1	Adicionar Serviço	53
6.5.2	Remover Serviço	54
6.5.3	Consultar Lista de Serviços	54
6.6	Actividades	55
6.6.1	Adicionar Actividade	55
6.6.2	Alterar Preço de Actividade	56
6.6.3	Remover Actividade	57
6.7	Contratos	58
6.7.1	Criar Contrato	58
6.7.2	Alterar Contrato	59
6.7.3	Cancelar Contrato	60
6.7.4	Adicionar Actividade ao Contrato	61
6.7.5	Cancelar Actividade do Contrato	61
6.8	Pagamentos	62
6.8.1	Confirmar Pagamento de Contrato	62
6.8.2	Confirmar Pagamento de Acréscimo	62
6.8.3	Confirmar Pagamento de Estorno	63

7	Diagramas de Classes	64
7.1	Package Serviços	65
7.2	Package Utilizadores	66
7.3	Package Contratos	67
8	Diagramas de Sequência de Implementação	68
8.1	Geral	68
8.1.1	Login	68
8.1.2	Consultar Contrato	69
8.1.3	Alterar Dados	70
8.1.4	Alterar Informação Pessoal	70
8.2	Clientes	71
8.2.1	Adicionar Cliente	71
8.2.2	Consultar Cliente	72
8.2.3	Editar Cliente	73
8.2.4	Remover Cliente	74
8.3	Fornecedores	75
8.3.1	Adicionar Fornecedor	75
8.3.2	Consultar Fornecedor	76
8.3.3	Editar Fornecedor	77
8.3.4	Remover Fornecedor	78
8.4	Funcionários	79
8.4.1	Adicionar Funcionário	79
8.4.2	Editar Funcionário	79
8.4.3	Remover Funcionário	80
8.5	Serviços	80
8.5.1	Adicionar Serviço	80
8.5.2	Remover Serviço	81
8.5.3	Consultar Lista de Serviços	81
8.6	Actividades	82
8.6.1	Adicionar Actividade	82
8.6.2	Alterar Preço de Actividade	83
8.6.3	Remover Actividade	84
8.7	Contratos	86
8.7.1	Criar Contrato	86
8.7.2	Alterar Contrato	87

8.7.3	Cancelar Contrato	88
8.7.4	Adicionar Actividade ao Contrato	89
8.7.5	Cancelar Actividade do Contrato	90
8.8	Pagamentos	91
8.8.1	Confirmar Pagamento de Contrato	91
8.8.2	Confirmar Pagamento de Acréscimo	92
8.8.3	Confirmar Pagamento de Estorno	93
9	Decisões Pertinentes	94
9.1	Pagamentos	94
9.2	Contratos	94
10	Conclusão	95

1 Introdução

No âmbito da disciplina de Desenvolvimento de Sistemas de Software, do terceiro ano da licenciatura em Engenharia Informática, foi-nos proposto modelar, numa primeira fase, um sistema relacionado com a gestão de um condomínio. Este software deverá ser capaz de oferecer aos condóminos uma vasta gama de serviços nos quais estão incluídos o acesso a serviços e actividades disponibilizadas por diversos fornecedores e a possibilidade de comprar esses serviços, estabelecendo contratos entre condóminos e fornecedores. Numa primeira fase começamos por desenhar o *Modelo de Domínio*, tendo sido este o primeiro contacto com o problema e com o cliente em questão, sendo este então um modelo muito importante pois é onde se começa a definir conceitos e o léxico relacionados com o problema. Passamos numa fase seguinte à construção do *Diagrama de Casos de Utilização*, ou *Diagrama de Use Cases*, onde são especificadas todas as acções que o sistema poderá ter com os vários utilizadores. Este é também um diagrama importante, pois é através dele que poderemos mostrar ao cliente como se comportará o software posteriormente desenvolvido, e é nesta fase que o cliente deverá avaliar o modelo e verificar se realmente é a solução pretendida para o sistema desejado. É nesta fase que se assume um compromisso real entre desenvolvedor e cliente com a finalidade de continuar a construção e a implementação do software em questão. Sendo assim vamos aqui apresentar e analisar as várias fases de desenvolvimento previstas numa primeira fase deste projecto.

Na segunda fase, são apresentados os Diagramas de Sequência desenvolvidos assim como os Diagramas de Classe. Os Diagramas de Sequência permitem a visualização do sequenciamento dos processos que o programa efectua, permitindo assim determinar de uma forma específica o comportamento que este tem. Os Diagramas de Classe servem de base para a programação dos objectos fazendo com que construção dos programas em si mais eficiente e mais rápida. Iremos mais à frente passar a uma maior especificação de todos estes processos de desenvolvimento.

2 Recursos Utilizados

2.1 UML (Unified Modeling Language)

O UML (Unified Modeling Language), é uma linguagem de modelação que nos permite facilitar o desenvolvimento de sistemas de software. O UML não define um método de desenvolvimento, ou seja, não nos diz por onde começar nem como desenhar ou pensar o nosso sistema de software, apenas nos disponibiliza um ambiente onde podemos obter uma melhor visualização e organização do sistema pensado por nós, e quais as comunicações entre objectos. Com a ajuda de uma notação gráfica, o UML dá-nos acesso a uma representação do sistema em vários diagramas padronizados (Fig.1), onde são descritos todos os pormenores de funcionamento.

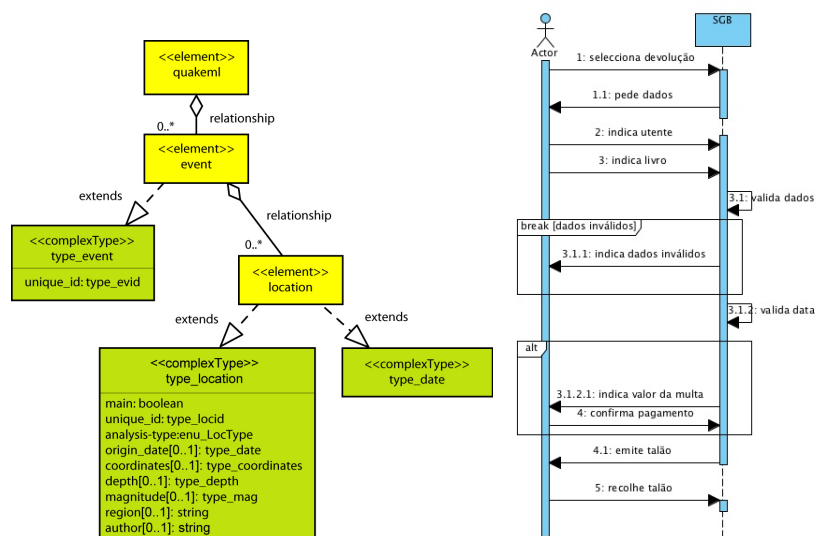


Fig.1 - Exemplos de um Diagrama de Classes e um Diagrama de Sequencia em UML.

Esta linguagem é de extrema importancia na fase de “Especificação de Requisitos”, na fase em que o diálogo com o cliente sobre o sistema de software se pode tornar confusa e resultar num mau entendimento entre desenvolvedor e cliente. O UML ajuda o cliente a definir de uma forma mais concreta aquilo que deseja para o seu sistema, podendo ter acesso a modelos e diagramas que representam o que ele realmente deseja.

2.2 Modelo de Domínio

O modelo de domínio é acima de tudo uma espécie de índice do projecto que indentifica os seus intervenientes, bem como a forma como eles se relacionam entre si. Este tipo de modelo, é construído usando a mesma notação de um Diagrama de Classes mas com todas as entidades pertencentes ao domínio do problema. Em suma, o diagrama deve conter informação relativa ao negócio, descrevendo as suas entidades e relações. Na construção do modelo de domínio deve se dar a maior relevância possível aos objectos do mundo real, pois, mais tarde, este será o ponto de partida para a construção da versões iniciais do diagrama de classes.

2.3 Diagrama de Use-Cases

Um *Diagrama de Use-Cases* ou *Diagrama de Casos de Utilização* (Fig.2) descreve todas as funcionalidades propostas para o sistema em desenvolvimento. Segundo *Ivar Jacobson* um Use Case é um “documento narrativo que descreve a sequência de eventos de um actor que usa um sistema para realizar um processo”. Um Use-Case representa apenas uma interação entre um utilizador (humano ou máquina) e o sistema. A título de exemplo, um Use-Case pode ser qualquer coisa como “Fazer Login”, “Registar no Sistema” ou “Apagar Conta”. Tipicamente todos os Casos de Utilização estão relacionados com “actores”, sendo que um actor é um humano ou uma entidade máquina que interage com o sistema com a finalidade de executar um procedimento.

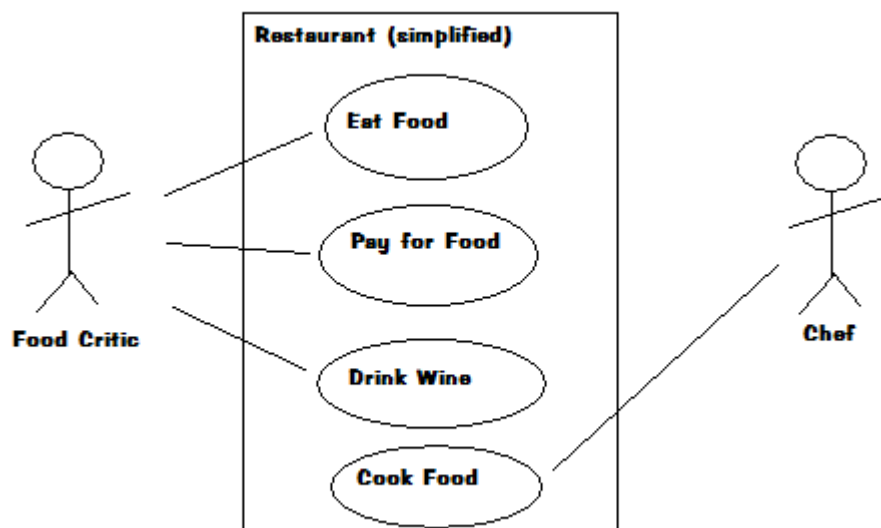


Fig.2 - Exemplo de um Diagrama de Use-Cases ou Diagrama de Casos de Utilização.

Relações entre Use-Cases

No *Diagrama de Use-Cases* podem também ser representadas as relações entre os diversos Casos de Utilização das seguintes formas:

Include Esta relação é utilizada quando um Caso de Utilização inclui outro Caso de Utilização, ou seja, admitindo que o UseCase1 inclui o UseCase2, o «*include*» define a obrigatoriedade de sempre que ocorrer o UseCase1, o UseCase2 irá ocorrer também.

Extend Esta relação é utilizada quando um Caso de Utilização “*extende*” outro Caso de Utilização, ou seja, admitindo que o UseCase1 “*extende*” o UseCase2, o «*extend*» define que sempre que ocorrer o UseCase1, o UseCase2 irá ou não ocorrer também.

2.4 Diagrama de Sequências

Os diagramas de sequência em UML representam a sequência em que os processos de cada caso de uso são executados. Tal como é habitual na modelação de sistemas de software este tipo de diagramas são desenvolvidos de forma incremental passando por várias fases:

- de Sistema
- de Sub-Sistema
- de Implementação

Os primeiros são os mais simples e são obtidos directamente da descrição textual dos use cases e contém apenas duas entidades, o actor e o sistema.

Na segunda fase já parece o sistema dividido em vários sub-sistemas mas ainda com as mensagens em forma textual tal como no diagrama anterior e no “Flow of Events”.

A última fase dos diagramas de sequência (implementação), normalmente posterior ao desenvolvimento do diagrama de classes, já identifica as várias classes do projecto (no lugar dos sub-sistemas que dão origem a packages) e as mensagens textuais são substituídas pelos métodos implementados nas classes.

2.5 Diagrama de Classes

Os diagramas de classe descrevem as classes que constituem a arquitectura do sistema e os relacionamentos entre elas. Uma classe é a base de qualquer sistema Orientado a Objectos e cada uma delas descreve um conjunto de objectos com a mesma estrutura de dados e comportamento.

Uma classe é composta por Atributos e Operações:

- **Atributos** : definem as características da classe; a visibilidade de um atributo pode ser pública, privada, protegida e package. Quando um atributo é público é acessível a partir de outras classes, privado apenas pode ser acedido pela própria classe, package apenas pode ser acedido dentro do package, protegido é acedido apenas pela própria classe e pelas suas subclasses.
- **Operações** : são funções que são requeridas a um objecto, e também possuem visibilidade como os atributos.

2.5.1 Relações entre Classes

No *Diagrama de Classes* podem também ser representadas as relações entre os diversas Classes das seguintes formas:

Generalização Relação entre uma coisa mais geral (pai/super-classe) e uma coisa mais específica (filho/sub-classe), também conhecida como herança, que representa as dependências e hierarquias.

Dependência Utiliza-se para mostrar que a origem da relação usa o destino, uma alteração no destino (o usado) pode alterar a origem (quem usa).

Associação Consiste em ter objectos de uma classe ligados a objectos de outra(s) classe(s), a navegação é bi-direccional e ilustra relações estruturais. Numa associação podemos ter:

- **Nome** : nome da associação, descreve a natureza da relação;
- **Multiplicidade** : quantos objectos participão na relação;
- **Navegação** : de onde vêm as informações da classe e para onde vão.

Agregação É um tipo de associação onde se demonstra que as informações de um objeto precisam ser complementadas de outra classe, objeto-todo e objeto-parte. (representa-se esta relação com um losango branco) Associação conhecida como "has-a"(tem um), representa uma relação forte entre as classes. Se a classe que "contém"for destruída não significa que a classe "contida"será também destruída.

Composição É um tipo de agregação, onde o objeto-parte pertence a um único objeto-todo. Associação conhecida como "owns a"(um dono), representa uma relação forte entre as classes. Se a classe que "é dona"for destruída significa que a classe "contida"será, ou seja não faz sentido que a classe-parte exista se a classe todo não existir. (representa-se esta relação com um losango preto)

3 Modelo de Domínio

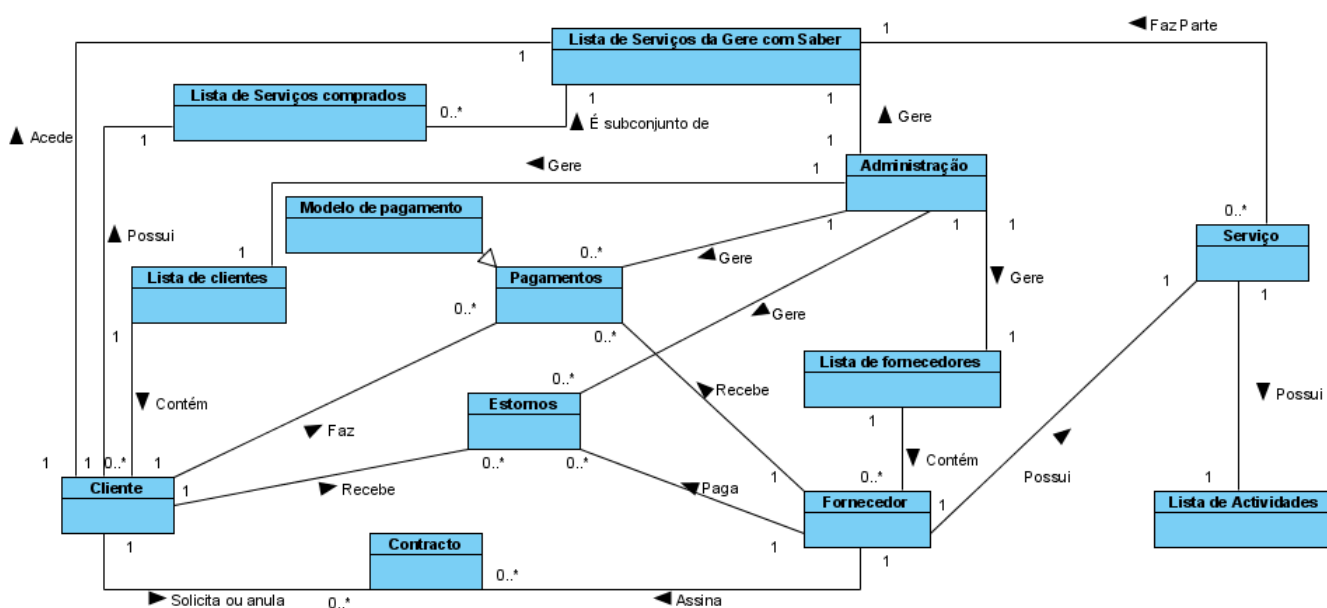


Fig.3 - Modelo de Domínio da Gere com Saber.

- A peça central do modelo é a administração que faz a gestão dos serviços, clientes, fornecedores e respectivos pagamentos e estornos.
- Os clientes, que estão organizados numa lista com todos os clientes, têm a possibilidade de aceder à lista de serviços da Gere com Saber e, se assim pretenderem, solicitar um contrato com um fornecedor de um dado serviço, passando assim a possuir uma lista de serviços comprados.
- Cada cliente pode também efectuar pagamentos e receber estornos, operações que são sempre feitas tendo a administração como intermediário.
- Os fornecedores, tal como os clientes, estão contidos numa lista gerida pela Gere com Saber.
- Cada fornecedor possui apenas um serviço que faz parte da lista de serviços da Gere com Saber, sendo que desse serviço fazem parte um lista de actividades inerentes a cada fornecedor.

- Na sua relação com os contratos, os fornecedores têm uma acção passiva pois, apenas podem realizar um contracto quando este for solicitado por um cliente.
- Os fornecedores recebem pagamentos e pagam estornos através da administração que age como intermediário no sistema.

4 Diagrama de Use-Cases

4.1 GCS

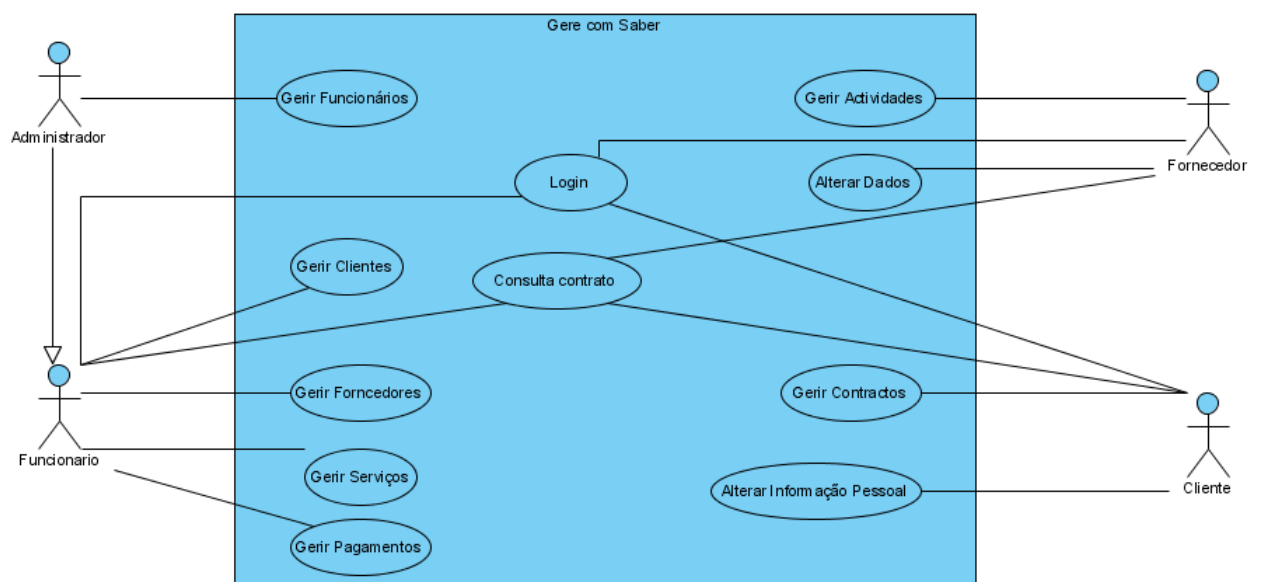


Fig.4 - Diagrama de Use-Cases da Gere com Saber.

Login

Super Use Case		
Author André		
Date Nov 30, 2009 9:41:32 PM		
Brief Description		
Preconditions O utilizador está registado no sistema.		
Post-conditions		
Flow of Events		Actor Input
	1	O utilizador pede fazer login no sistema.
	2	
	3	O utilizador insere os dados.
	4	
	5	
Alternative1		System Response
	1	
	2	
	3	
	4	
	5	O sistema valida os dados.
		Actor Input
	1	
	2	
	3	
	4	
	5	O sistema concede acesso ao utilizador.
		System Response
	1	
	2	
	3	
	4	
	5	O sistema avisa que os dados do login estão incorrectos.
		Actor Input
	1	
	2	
	3	
	4	
	5	O sistema avisa que os dados do login estão incorrectos.
		System Response
	1	
	2	
	3	
	4	
	5	Volta ao passo 2.

Consultar Contrato

Super Use Case		
Author André		
Date Nov 30, 2009 8:45:50 PM		
Brief Description		
Preconditions O contracto foi realizado.		
Post-conditions		
Flow of Events		Actor Input
	1	O utilizador pede para ver a lista dos contratos.
	2	
	3	O utilizador selecciona o contrato que deseja visualizar.
	4	
	5	
Alternative1		System Response
	1	
	2	
	3	
	4	O sistema verifica o tipo do utilizador. (executa um fluxo alternativo)
	5	O sistema apresenta os dados do contrato. Ver informa
		Actor Input
	1	
	2	
	3	
	4	
	5	
Alternative2		System Response
	1	
	2	
	3	O sistema verifica que o utilizador é do tipo funcionário.
	4	O sistema apresenta a lista completa dos contratos.
	5	Volta ao passo ao passo 3.]
		Actor Input
	1	
	2	
	3	
	4	
	5	
Alternative3		System Response
	1	
	2	
	3	O sistema verifica que o utilizador é do tipo cliente.
	4	O sistema apresenta a lista dos contratos que envolvem o cliente.
	5	Volta ao passo 3.

Alterar Informação Pessoal

Super Use Case		
Author	André	
Date	Nov 30, 2009 9:11:58 PM	
Brief Description		
Preconditions	O cliente encontra-se registado no sistema	
Post-conditions	Sucesso: Os dados do cliente são actualizados Insucesso: Não são feitas alterações no sistema	
Flow of Events		Actor Input
	1	O cliente pede para alterar as suas informações pessoais.
	2	
	3	O cliente insere os novos dados.
	4	
	5	O cliente responde afirmativamente.
	6	
Alternative 1		System Response
	1	
	2	O sistema pede os seus novos dados.
	3	
	4	O sistema pergunta se o cliente tem a certeza que deseja gravar as alterações.
	5	
	6	O sistema actualiza as informações do sistema.
		Actor Input
	1	
	2	
	3	
	4	
	5	O cliente responde negativamente.
	6	
		System Response
	1	
	2	
	3	
	4	
	5	
	6	O sistema cancela a operação.

4.1.1 Gerir Funcionários

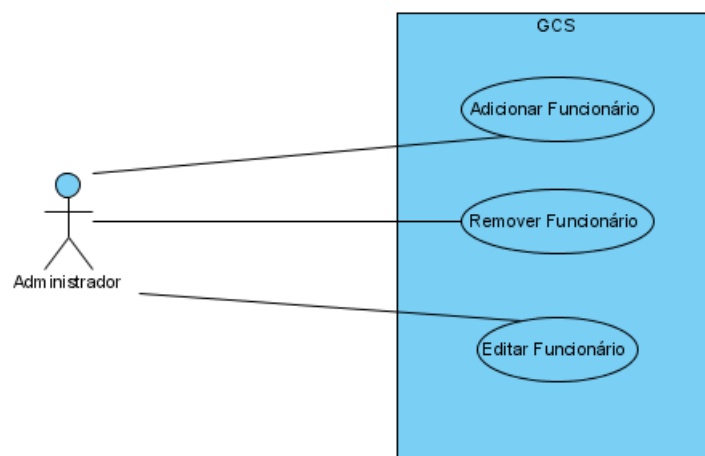


Fig.5 - Diagrama de Use-Cases de Gerir Funcionários.

Adicionar Funcionário

Super Use Case		
Author André		
Date Nov 30, 2009 4:49:28 PM		
Brief Description		
Preconditions O funcionário ainda não foi registado.		
Post-conditions O funcionário foi registado com sucesso.		
Flow of Events		Actor Input
	1	O administrador pede para adicionar funcionário.
	2	
	3	O administrador insere os dados.
	4	O sistema regista o novo funcionário.
		System Response
		O sistema pede os dados do novo funcionário (nome, nif, telefone e morada, palavra-passe).

Remover Funcionário

Super Use Case		
Author André		
Date Nov 30, 2009 4:56:35 PM		
Brief Description		
Preconditions O funcionário já foi registado.		
Post-conditions O funcionário deixa de estar registado.		
Flow of Events		Actor Input
	1	O administrador pede para remover funcionário.
	2	
	3	O administrador insere o código.
	4	
		System Response
		O sistema pede o código do funcionário (nif).
		O sistema valida o código inserido.
		O sistema remove o registo do funcionário.
Alternative 1		Actor Input
	1	
	2	
	3	
	4	
		System Response
		O sistema avisa que o código inserido não é válido.
		Volta ao passo 1.

Editar Funcionário

Super Use Case		
Author André		
Date Nov 30, 2009 5:00:41 PM		
Brief Description		
Preconditions O funcionário está registado.		
Post-conditions Registo do funcionário é actualizado.		
Flow of Events		Actor Input
	1	O administrador pede para editar um funcionário.
	2	
	3	O administrador insere o código.
	4	
	5	
	6	O administrador insere os novos dados do funcionário.
		System Response
		O sistema pede o código do funcionário (nif).
		O sistema valida o código inserido.
		O sistema pede os novos dados.
		O sistema regista as alterações.
Alternative 1		Actor Input
	1	
	2	
	3	
		System Response
		O sistema avisa que o código inserido é inválido.
		Volta ao passo 1.

4.1.2 Gerir Clientes

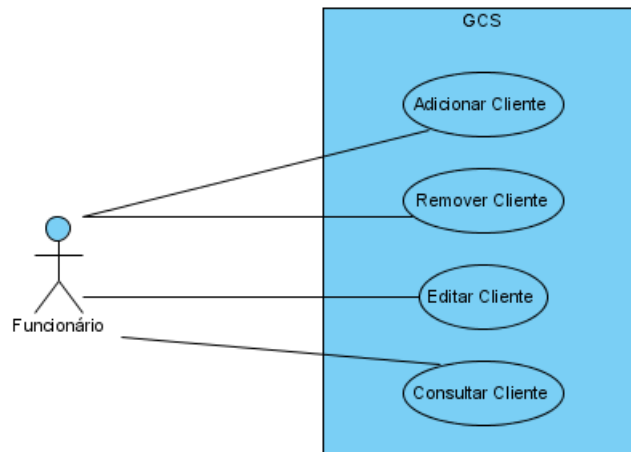


Fig.6 - Diagrama de Use-Cases de Gerir Clientes.

Adicionar Clientes

Super Use Case		
Author	André	
Date	Nov 26, 2009 11:24:48 PM	
Brief Description		
Preconditions	O cliente ainda não foi registado no sistema.	
Post-conditions	O novo cliente fica registado.	
Flow of Events	Actor Input	
	1	O funcionário pede para registar um novo cliente.
	2	
	3	O funcionário insere os dados.
	4	
	System Response	
		O sistema pede os dados do cliente (nome, nif, telefone, morada, palavra-passe).
		O sistema regista o novo cliente.

Remover Cliente

Super Use Case		
Author	André	
Date	Nov 26, 2009 11:53:31 PM	
Brief Description		
Preconditions	O cliente foi registado anteriormente.	
Post-conditions	Sucesso: O registo do cliente é apagado. Insucesso: Não são feitas alterações.	
Flow of Events	Actor Input	
	1	O funcionário pede para remover um cliente.
	2	
	3	O funcionário selecciona o cliente pretendido.
	4	
	5	O funcionário responde afirmativamente
	6	
	System Response	
		O sistema apresenta a lista dos clientes.
		O sistema pergunta se tem a certeza que deseja efectuar a operação.
		O sistema remove o registo do cliente.
Exception1	Actor Input	
	1	
	2	
	3	
	4	O funcionário responde negativamente.
	5	
	System Response	
		O sistema cancela a acção.

Editar Cliente

Super Use Case		
Author André		
Date Nov 30, 2009 5:33:29 PM		
Brief Description		
Preconditions O cliente está registado no sistema.		
Post-conditions Os dados do cliente são actualizados.		
Flow of Events		Actor Input
	1	O funcionário pede para editar um cliente.
	2	
	3	O funcionário selecciona o cliente pretendido.
	4	
	5	O funcionário insere os novos dados do cliente.
	6	
		System Response
		O sistema apresenta a lista dos clientes
		O sistema pede os novos dados.
		O sistema regista as alterações.

Consultar Cliente

Super Use Case		
Author André		
Date Nov 30, 2009 5:39:34 PM		
Brief Description		
Preconditions O cliente já se encontra registado no sistema.		
Post-conditions		
Flow of Events		Actor Input
	1	O funcionário pede para consultar um cliente.
	2	
	3	O funcionário selecciona o cliente pretendido.
	4	
		System Response
		O apresenta a lista dos clientes.
		O sistem apresenta os dados do cliente.

4.1.3 Gerir Fornecedores

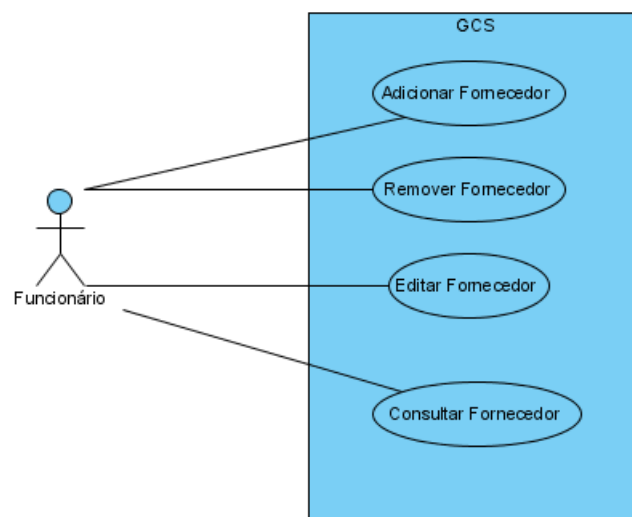


Fig.7 - Diagrama de Use-Cases de Gerir Fornecedores.

Adicionar Fornecedor

Super Use Case		
Author		
Date		
Brief Description		
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	O funcionário pede para adicionar um fornecedor.
	2	
	3	O funcionário insere os dados. O tipo de serviço é selecionado a partir da lista de serviços.
	4	O sistema regista o fornecedor.

Remover Fornecedor

Super Use Case		
Author		
Date		
Brief Description		
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	O funcionário pede para remover um fornecedor.
	2	
	3	O funcionário seleciona o fornecedor pretendido.
	4	
	5	O funcionário responde afirmativamente.
	6	O sistema remove o registo do fornecedor.
Exception1		Actor Input
	1	
	2	
	3	
	4	O funcionário responde negativamente.
		O sistema cancela a operação.

Editar Fornecedor

Super Use Case		
Author André		
Date Nov 30, 2009 6:05:16 PM		
Brief Description		
Preconditions O fornecedor está registado no sistema		
Post-conditions Sucesso: Dados do fornecedor alterados Insucesso: Sistema não faz qualquer alteração		
Flow of Events		Actor Input
	1	O funcionário pede para editar um fornecedor.
	2	
	3	O funcionário selecciona fornecedor
	4	
	5	O funcionário insere os dados
	6	
	7	O funcionário responde afirmativamente
	8	
Exception1		System Response
	1	
	2	
	3	
	4	O funcionário responde negativamente
	5	
		O sistema regista as alterações.
		O sistema apresenta lista de fornecedores
		O sistema pede os novos dados do fornecedor
		O sistema pergunta se tem a certeza que quer efectuar as alterações
		O sistema cancela operação

Consultar Fornecedor

Super Use Case		
Author André		
Date Nov 27, 2009 12:00:11 AM		
Brief Description		
Preconditions O fornecedor está registado		
Post-conditions		
Flow of Events		Actor Input
	1	O funcionário pede para consultar um fornecedor.
	2	
	3	O funcionário selecciona o fornecedor pretendido
	4	
		System Response
		O sistema apresenta a lista de fornecedores
		O sistema apresenta os dados do fornecedor

4.1.4 Gerir Serviços

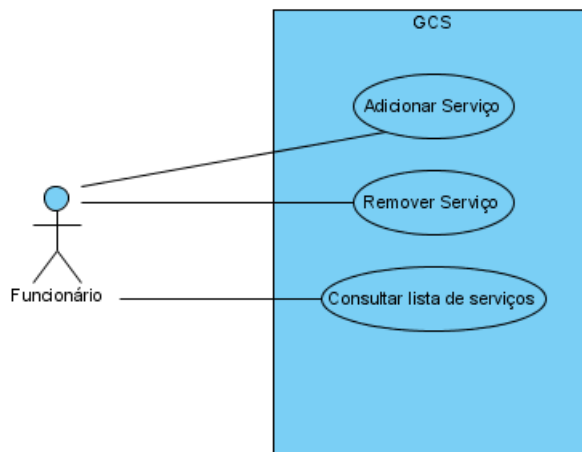


Fig.8 - Diagrama de Use-Cases de Gerir Serviços.

Adicionar Serviços

Super Use Case		
Author	André	
Date	Nov 30, 2009 6:14:07 PM	
Brief Description		
Preconditions	O serviço não está registado	
Post-conditions	O serviço fica disponível	
Flow of Events		Actor Input
	1	O funcionário pede para adicionar um serviço.
	2	
	3	O funcionário insere os dados
	4	
		System Response
		O sistema pede o nome e código do serviço
		O sistema regista o novo serviço

Remover Serviço

Super Use Case		
Author André		
Date Nov 27, 2009 4:37:14 PM		
Brief Description		
Preconditions Não existem fornecedores desse serviço no sistema.		
Post-conditions Sucesso: O registo do serviço deixa de existir no sistema. Insucesso: O sistema não faz qualquer alteração		
Flow of Events		Actor Input
	1	O funcionário pede para remover um serviço.
	2	
	3	O funcionário selecciona o o serviço pretendido
	4	
	5	O funcionário responde afirmativamente
Exception1	6	O serviço é removido do sistema
		System Response
	1	
	2	
	3	
	4	O funcionário responde negativamente
	5	O sistema cancela a operação

Consultar Lista de Serviços

Super Use Case		
Author André		
Date Nov 30, 2009 6:21:45 PM		
Brief Description		
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	O funcionário pede para consultar a lista de serviços.
	2	
		System Response
		Sistema apresenta a lista de serviços

4.1.5 Gerir Pagamentos

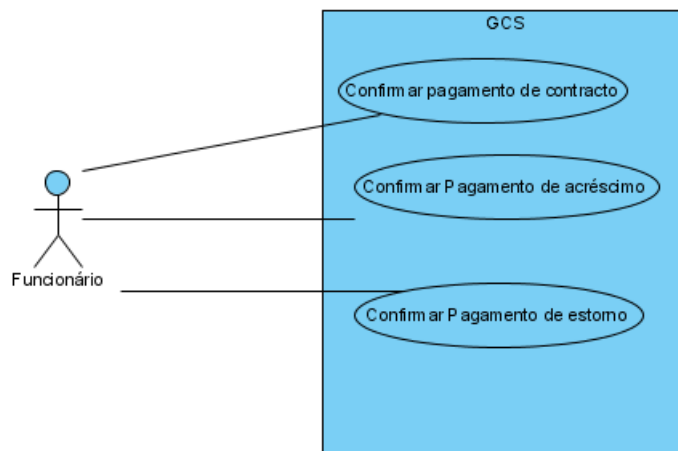


Fig.9 - Diagrama de Use-Cases de Gerir Pagamentos.

Confirmar Pagamento de Contrato

Super Use Case		
Author		
André		
Date		
Nov 30, 2009 6:22:26 PM		
Brief Description		
Preconditions		
O Contrato existe e ainda não foi pago		
Post-conditions		
O contrato foi pago		
Flow of Events		Actor Input
	1	O funcionário pede para ver lista de contratos.
	2	
	3	Funcionário seleciona o contrato
	4	
		System Response
		Sistema apresenta lista de contratos carecem de pagamento
		O sistema marca contrato como pago pelo cliente

Confirmar Pagamento de Acréscimo

Super Use Case		
Author		
André		
Date		
Nov 30, 2009 6:25:03 PM		
Brief Description		
Preconditions		
Acréscimo ainda não foi pago		
Post-conditions		
Acréscimo foi pago		
Flow of Events		Actor Input
	1	O funcionário pede para ver lista de contratos.
	2	
	3	Funcionário Seleciona contrato
	4	
		System Response
		O sistema apresenta lista de contratos que carecem de pagamento de acréscimo
		O sistema marca acréscimo como pago por cliente

Confirmar Pagamento de Estorno

Super Use Case		
Author		
André		
Date		
Nov 30, 2009 6:27:17 PM		
Brief Description		
Preconditions		
O estorno ainda não foi pago		
Post-conditions		
O estorno foi pago		
Flow of Events		Actor Input
	1	O funcionário pede para ver lista de contratos.
	2	
	3	O funcionário seleciona o contrato
	4	
		System Response
		O sistema apresenta lista de contratos que carecem de pagamento de estorno
		O sistema marca estorno como pago ao cliente

4.1.6 Gerir Actividades

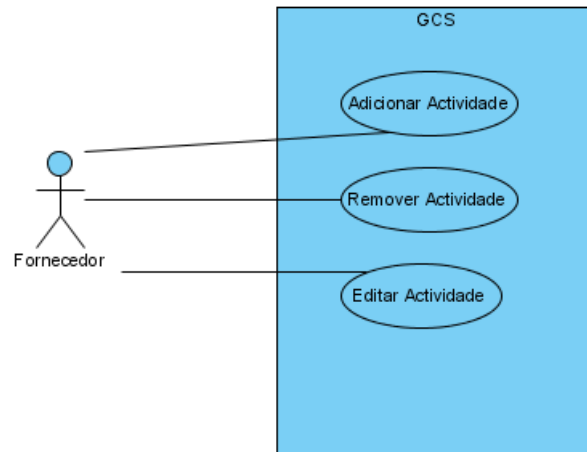


Fig.10 - Diagrama de Use-Cases de Gerir Actividades.

Adicionar Actividade

Super Use Case		
Author	André	
Date	Nov 28, 2009 5:08:26 PM	
Brief Description		
Preconditions	A actividade ainda não foi registada pelo fornecedor.	
Post-conditions	A actividade é adicionada à lista de actividades do fornecedor	
Flow of Events		Actor Input
	1	O fornecedor pede para adicionar nova actividade.
	2	
	3	O fornecedor preenche os dados solicitados
	4	
		System Response
		O sistema pede os dados da nova actividade. (descrição, preço)
		O sistema regista a nova actividade.

Remover Actividade

Super Use Case		
Author André		
Date Nov 30, 2009 8:27:32 PM		
Brief Description		
Preconditions A actividade foi registada anteriormente.		
Post-conditions Sucesso: A actividade deixa de fazer parte da lista do fornecedor. Insucesso: A actividade deixa de fazer parte da lista do fornecedor.		
Flow of Events		Actor Input
	1	O fornecedor pede para apresentar a sua lista de actividades.
	2	
	3	O fornecedor selecciona a actividade que pretende remover.
	4	
	5	O fornecedor responde afirmativamente.
	6	
Exception1		System Response
	1	
	2	
	3	
	4	
	5	O sistema pergunta se o fornecedor tem a certeza que deseja remover a actividade.
	6	O sistema remove o registo da actividade.
		Actor Input
	1	
	2	
	3	
	4	
	5	O fornecedor responde negativamente.
	6	O sistema cancela a operação.

Alterar Preço de Actividade

Super Use Case		
Author André		
Date Nov 30, 2009 8:38:25 PM		
Brief Description		
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	O fornecedor pede a lista das suas actividades.
	2	
	3	O fornecedor selecciona a actividade que pretende alterar.
	4	
	5	O fornecedor insere o novo preço.
	6	
	7	O fornecedor responde afirmativamente.
Exception1		System Response
	1	
	2	O sistema apresenta a lista das actividades.
	3	
	4	O sistema pede o novo preço da actividade.
	5	
	6	O sistema pergunta se tem a certeza que deseja efectuar a alteração.
	7	O sistema regista as alterações.
		Actor Input
	1	
	2	
	3	
	4	
	5	
	6	
	7	O fornecedor negativamente.
		System Response
	8	O sistema cancela a operação.

4.1.7 Gerir Contratos

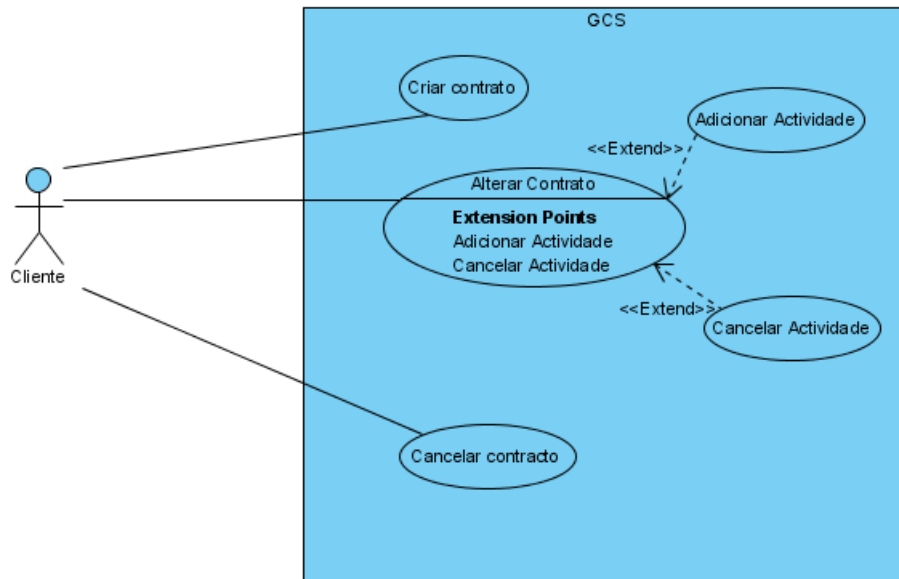


Fig.11 - Diagrama de Use-Cases de Gerir Contratos.

Criar Contracto

Super Use Case		
Author	André	
Date	Nov 30, 2009 6:29:20 PM	
Brief Description		
Preconditions	Cliente não possui contrato com os fornecedores que seleccionar	
Post-conditions	Sucesso: Cliente passa a ter contrato com o fornecedor Insucesso: O sistema não faz qualquer acção	
Flow of Events		Actor Input
	1	O cliente pede para ver lista de serviços.
	2	
	3	O cliente escolhe o serviço
	4	
	5	O cliente selecciona as actividades
	6	
	7	Cliente responde afirmativamente
	8	
Exception1		System Response
	1	
	2	O sistema apresenta lista de serviços
	3	
	4	O sistema apresenta lista de fornecedores e respectivas actividades
	5	
	6	O sistema apresenta as simulações de contratos e pergunta se o cliente aceita
	7	
	8	O sistema regista contrato de actividades com os fornecedores seleccionados
		Actor Input
	1	
	2	
	3	
	4	
	5	
	6	O cliente responde negativamente
	7	
		System Response
		O sistema cancela a operação

Alterar Contrato

Super Use Case		
Author André		
Date Nov 30, 2009 6:42:48 PM		
Brief Description		
Preconditions Cliente possui contratos		
Post-conditions		
Flow of Events		Actor Input
	1	O cliente pede para alterar contratos
	2	
	3	O cliente escolhe contrato a alterar
	4	
	5	<extend> Adicionar Actividade
Alternative1		System Response
	1	
	2	O sistema apresenta lista de contratos
	3	
	4	O sistema apresenta lista de actividades do contrato
	5	

Adicionar Actividade ao Contracto

Super Use Case		
Author André		
Date Nov 30, 2009 6:50:56 PM		
Brief Description		
Preconditions		
Post-conditions Sucesso: Nova actividade adicionada e acrescimo assinalado Insucesso: Sistema não faz qualquer alteração		
Flow of Events		Actor Input
	1	O cliente pede para mostrar a lista de actividades disponiveis
	2	
	3	O cliente seleciona nova actividade para adicionar ao seus contrato
	4	
	5	O cliente responde afirmativamente
	6	
Exception1		System Response
	1	
	2	O sistema apresenta lista de actividades do fornecedor
	3	
	4	O sistema pergunta se tem a certeza que pretende adicionar a nova actividade
	5	
	6	O sistema adiciona nova actividade ao contrato e marca acrescimo no contrato

Cancelar Actividade

Super Use Case		
Author André		
Date Nov 30, 2009 6:59:13 PM		
Brief Description		
Preconditions		
Post-conditions Sucesso: Contrato alterado e marcado com estorno Insucesso: Sistema não faz qualquer alteração		
Flow of Events		Actor Input
	1	O cliente pede para mostrar a lista de actividades do seu contrato
	2	
	3	O cliente seleciona a actividade que deseja cancelar
	4	
	5	O cliente responde afirmativamente
Exception1	6	O sistema actualiza contrato e marca contrato com estorno
		System Response
	1	
	2	
	3	
	4	
	5	O cliente responde negativamente
	6	O sistema cancela operação

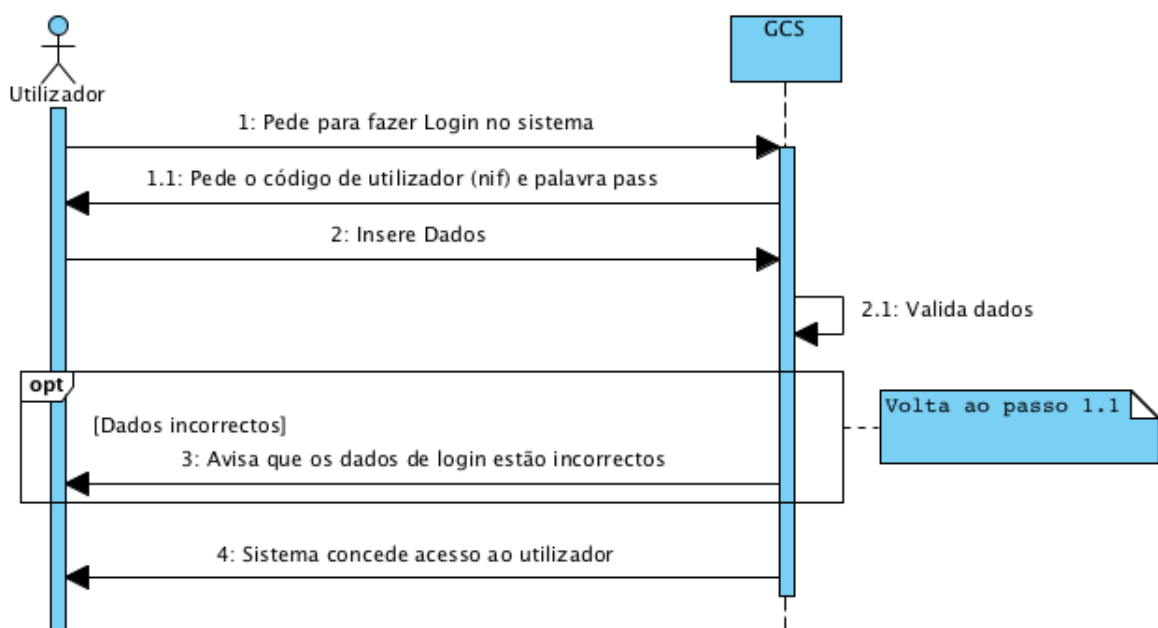
Cancelar Contracto

Super Use Case		
Author André		
Date Nov 30, 2009 7:04:25 PM		
Brief Description		
Preconditions O cliente possui pelo menos um contrato		
Post-conditions Sucesso: Contrato cancelado e marcado com estorno Insucesso: O sistema não fez alterações		
Flow of Events		Actor Input
	1	O cliente pede para mostrar lista de contratos
	2	
	3	O cliente seleciona o contrato pretendido
	4	
	5	O cliente responde afirmativamente
Exception1	6	O contrato é cancelado
		System Response
	1	
	2	
	3	
	4	
	5	O cliente responde negativamente
	6	O sistema cancela operação

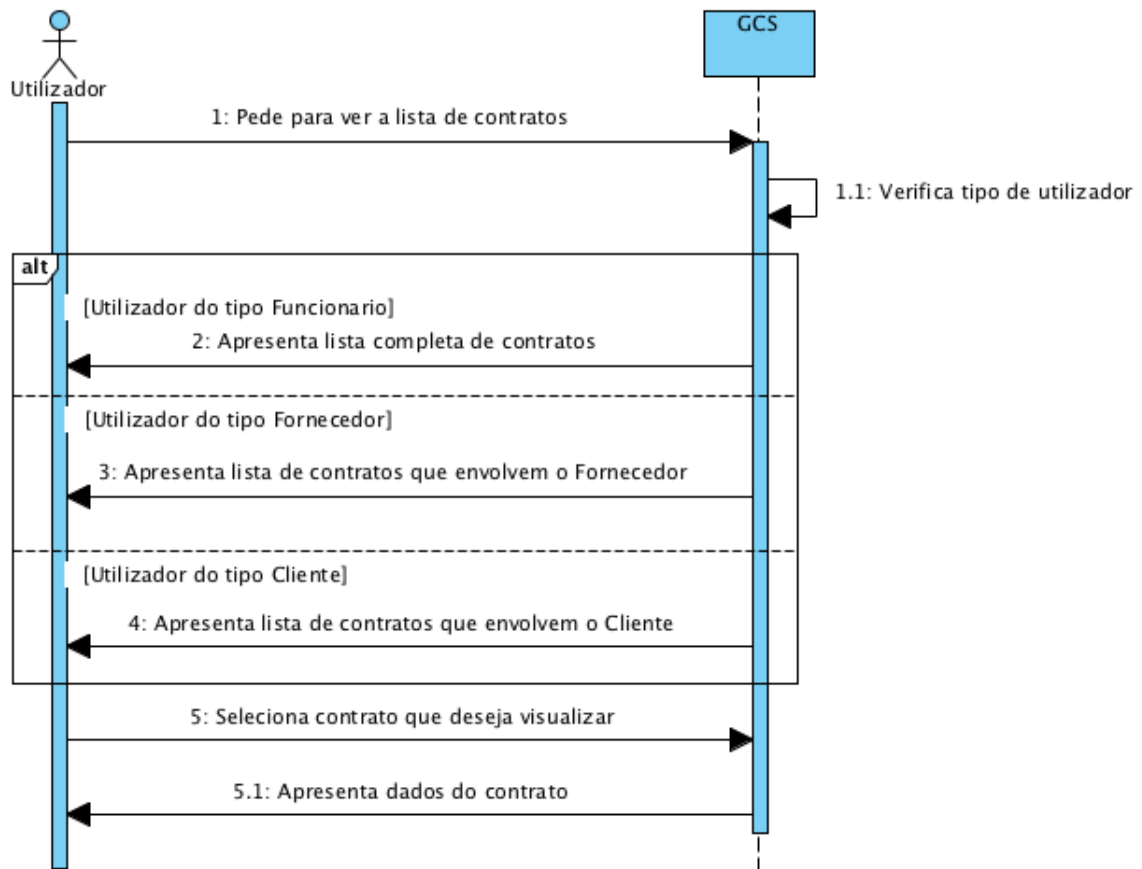
5 Diagramas de Sequência de Sistemas

5.1 Geral

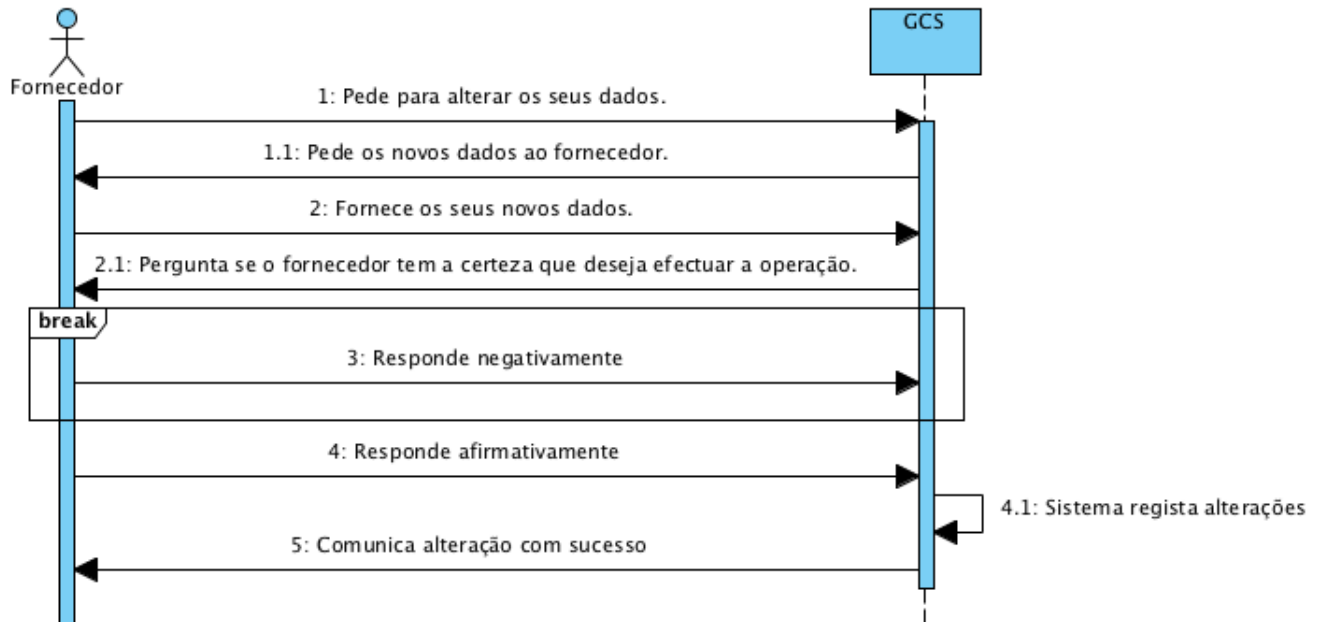
5.1.1 Login



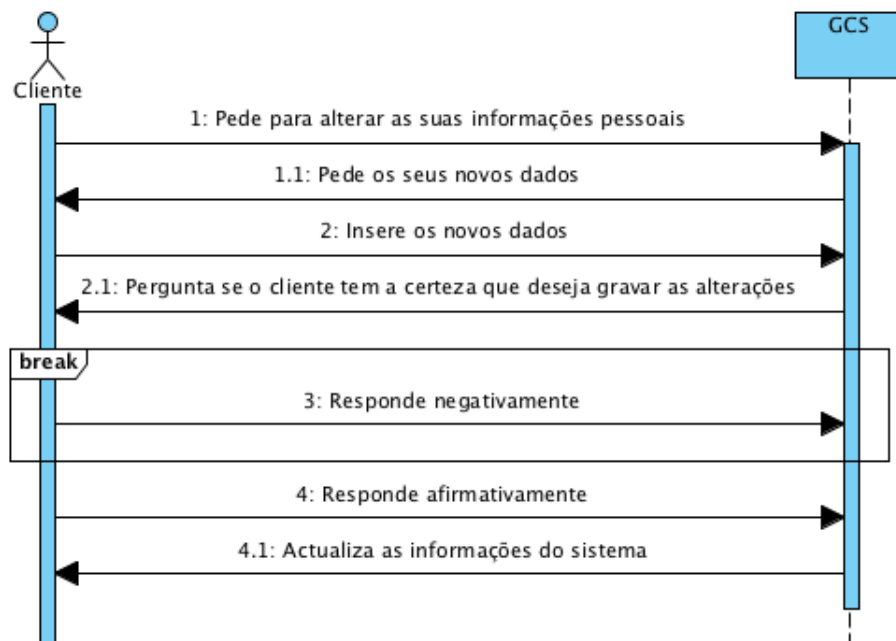
5.1.2 Consultar Contrato



5.1.3 Alterar Dados

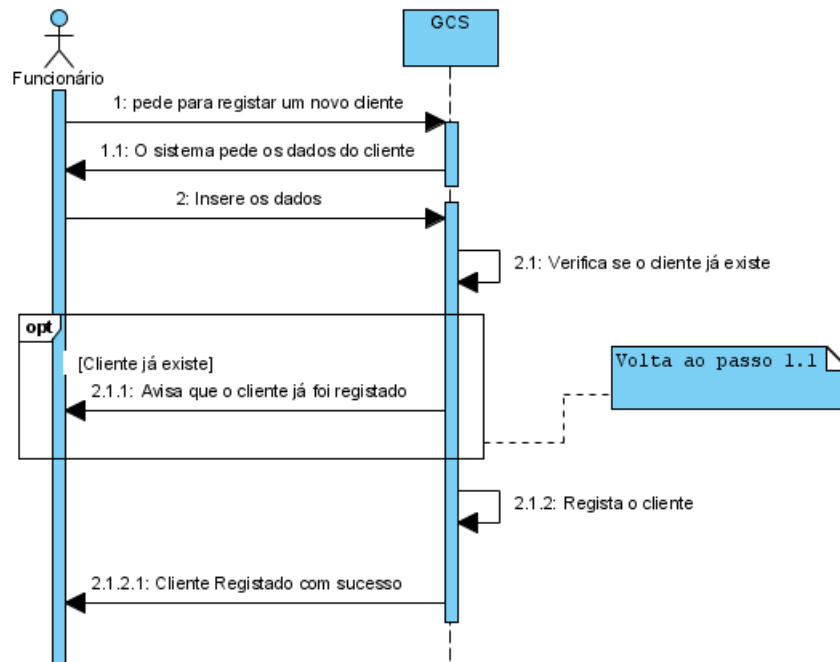


5.1.4 Alterar Informação Pessoal

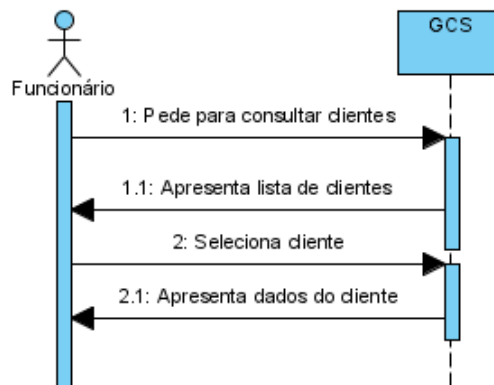


5.2 Clientes

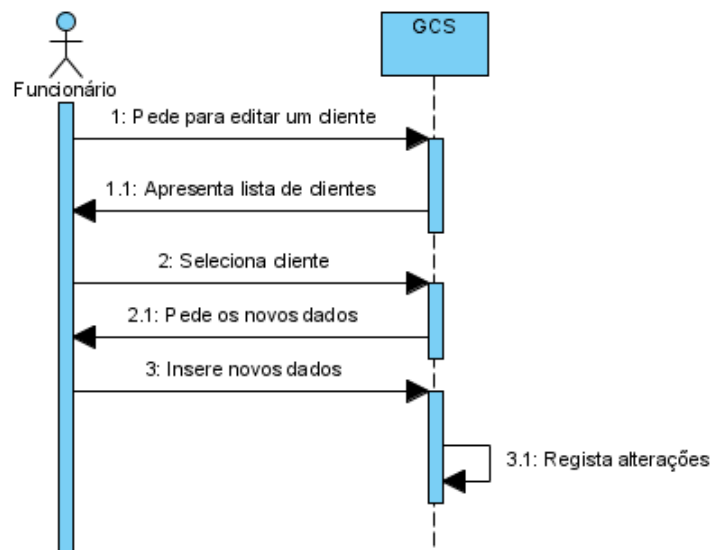
5.2.1 Adicionar Cliente



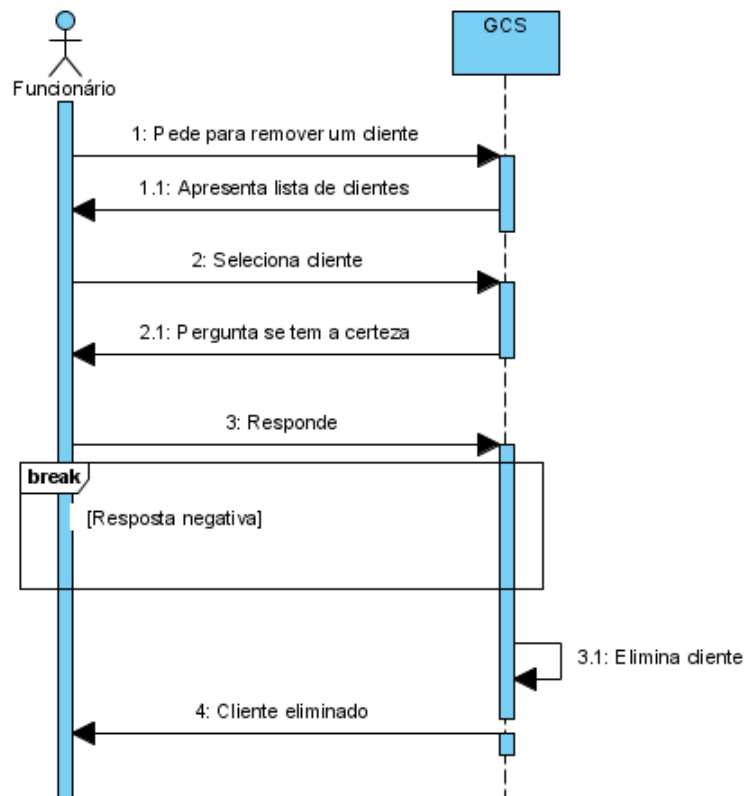
5.2.2 Consultar Cliente



5.2.3 Editar Cliente

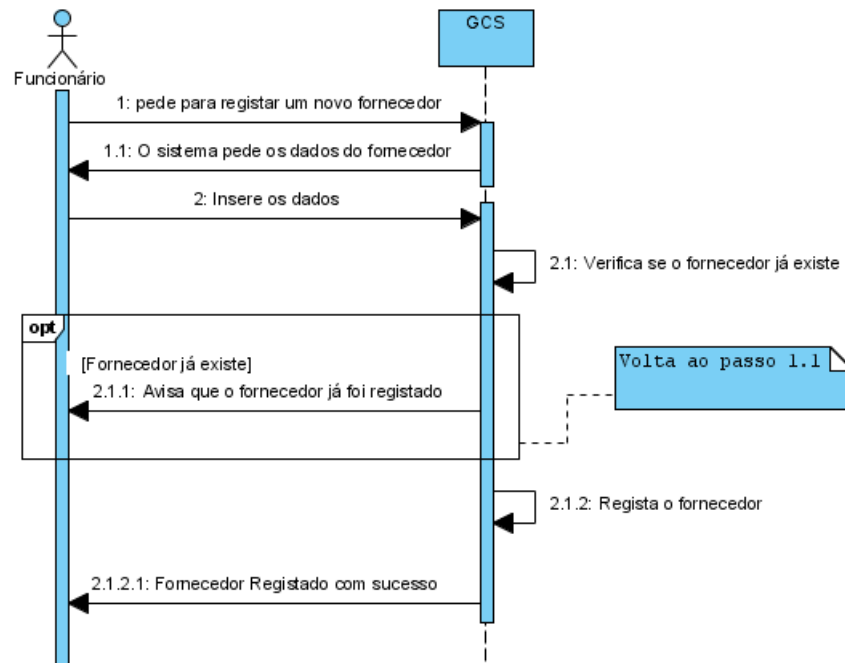


5.2.4 Remover Cliente

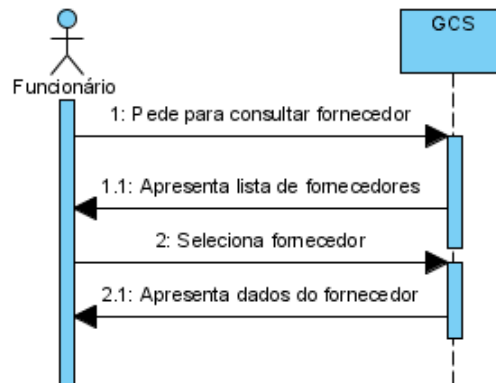


5.3 Fornecedores

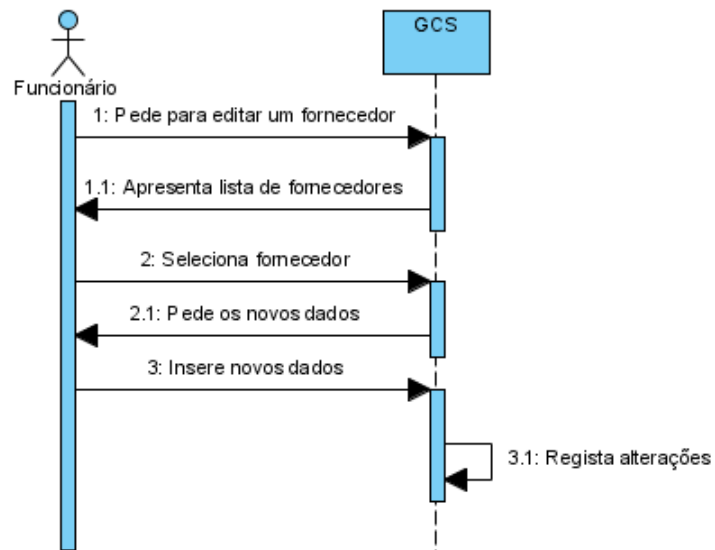
5.3.1 Adicionar Fornecedor



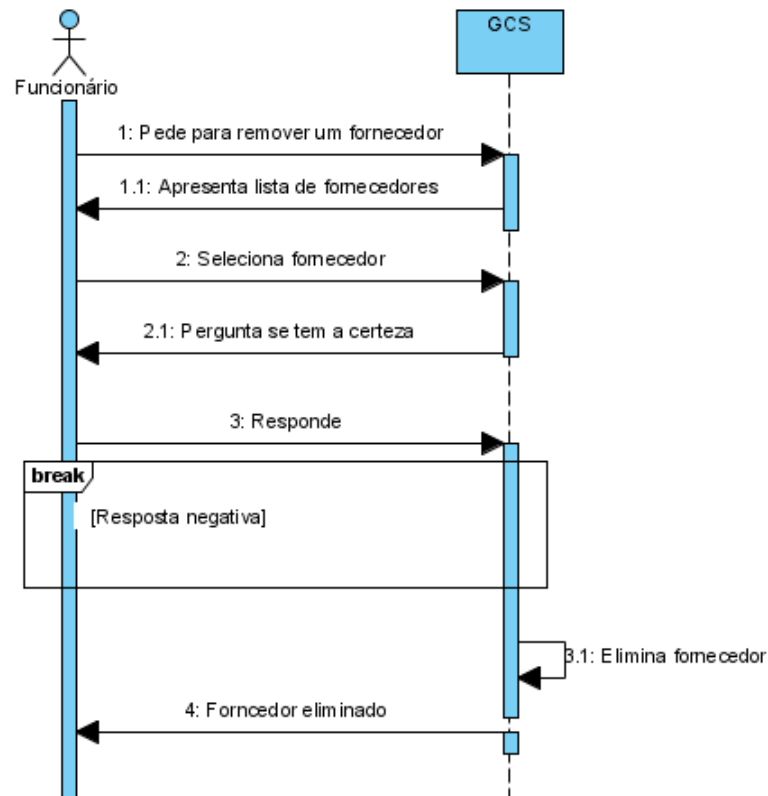
5.3.2 Consultar Fornecedor



5.3.3 Editar Fornecedor

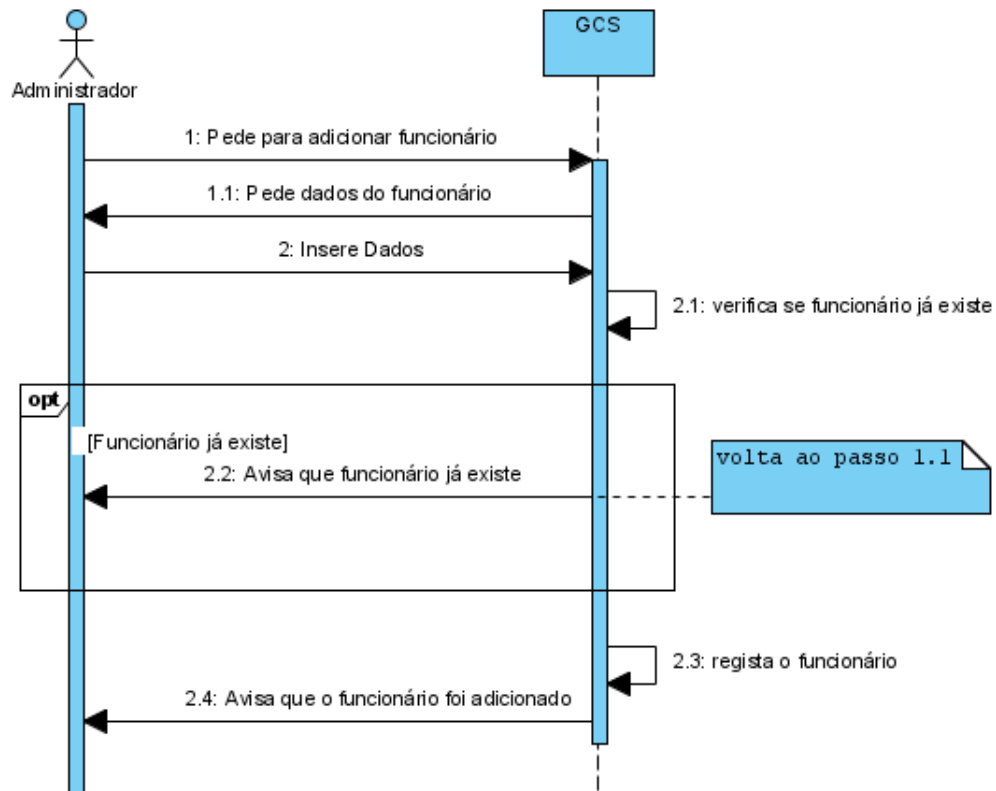


5.3.4 Remover Fornecedor

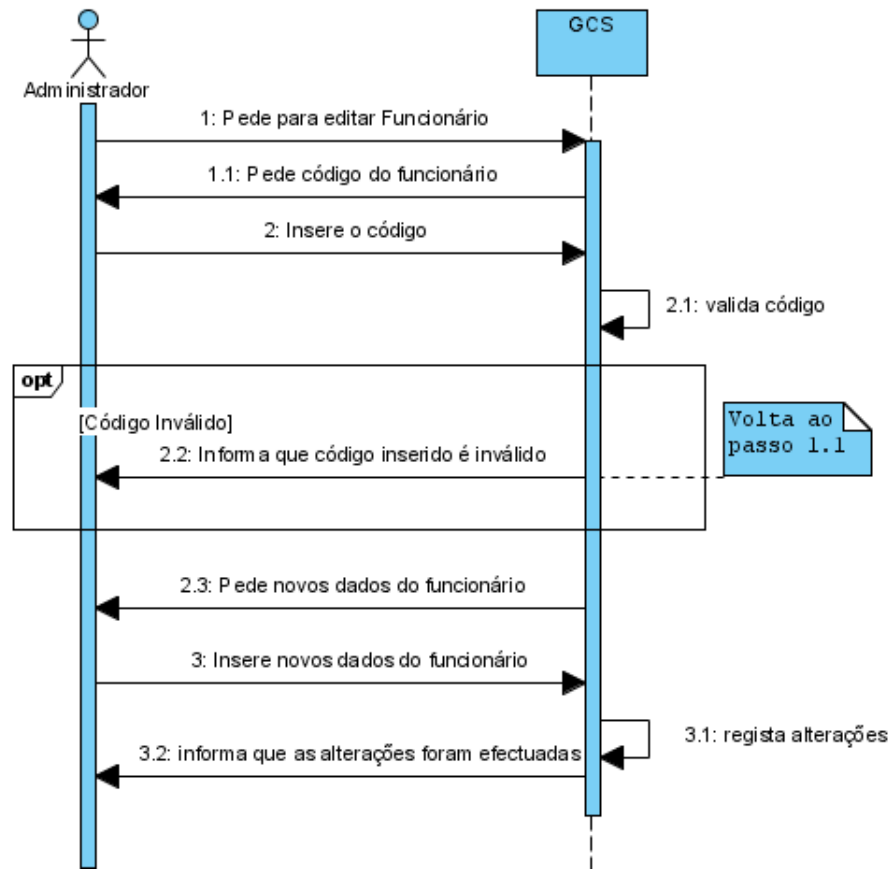


5.4 Funcionários

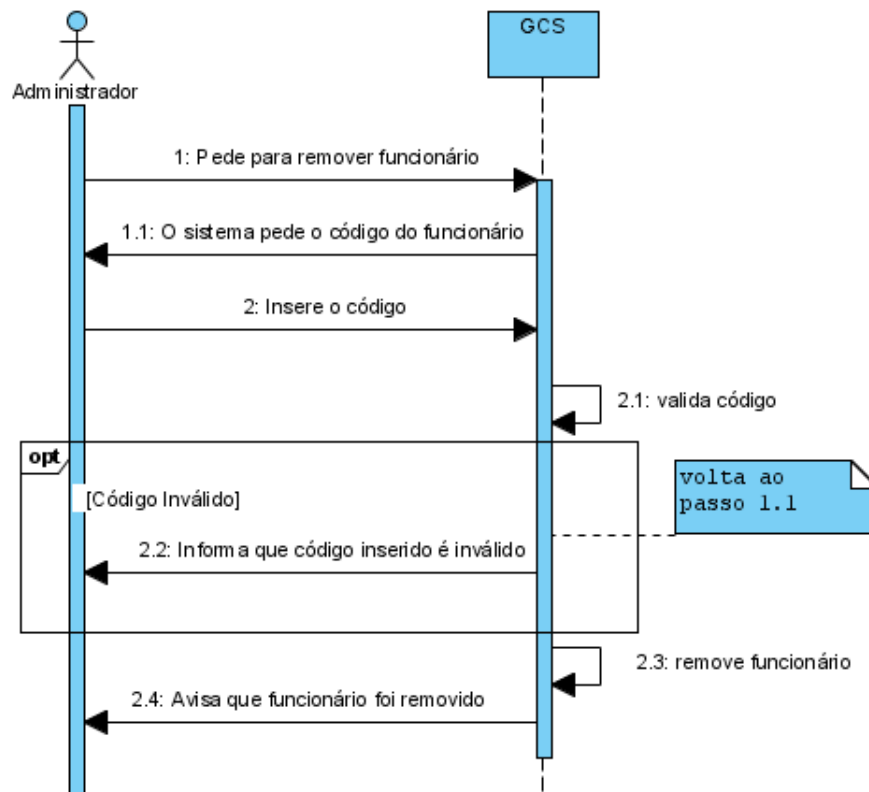
5.4.1 Adicionar Funcionário



5.4.2 Editar Funcionário

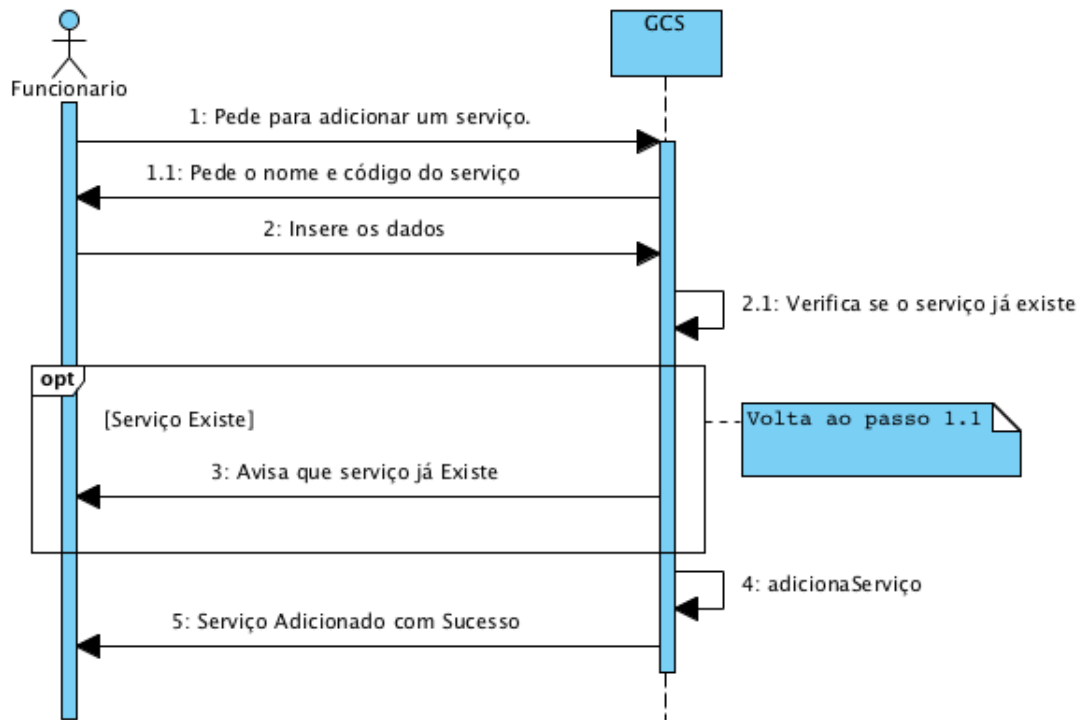


5.4.3 Remover Funcionário

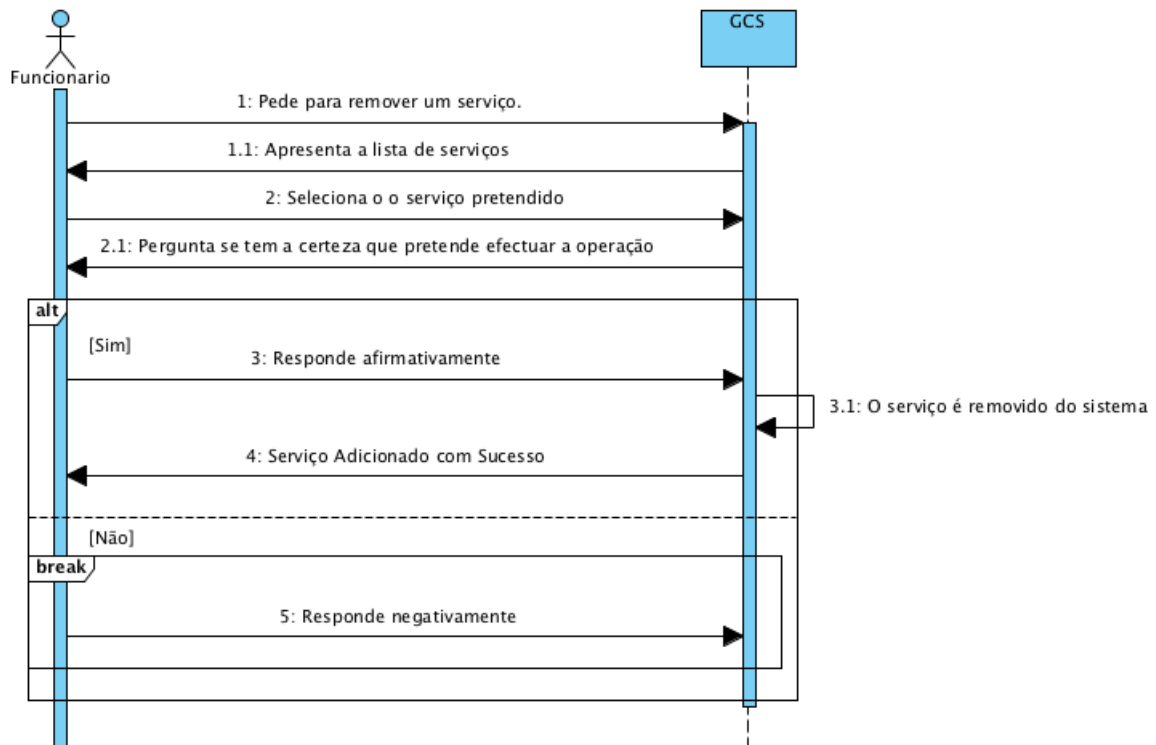


5.5 Serviços

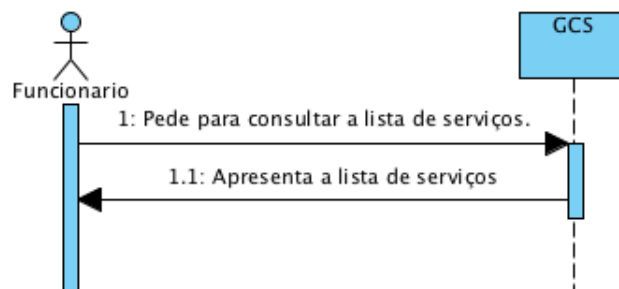
5.5.1 Adicionar Serviço



5.5.2 Remover Serviço

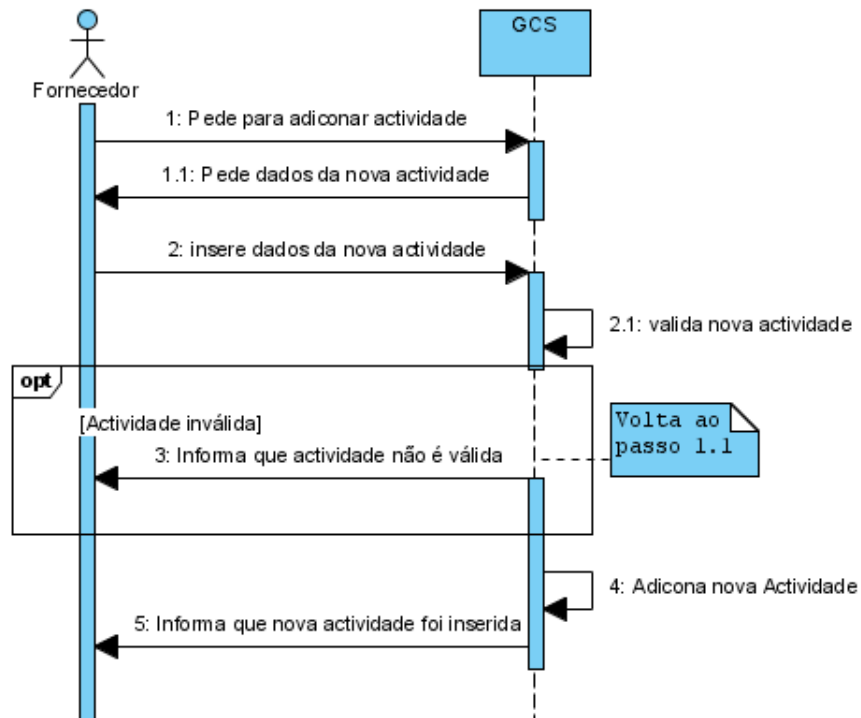


5.5.3 Consultar Lista de Serviços

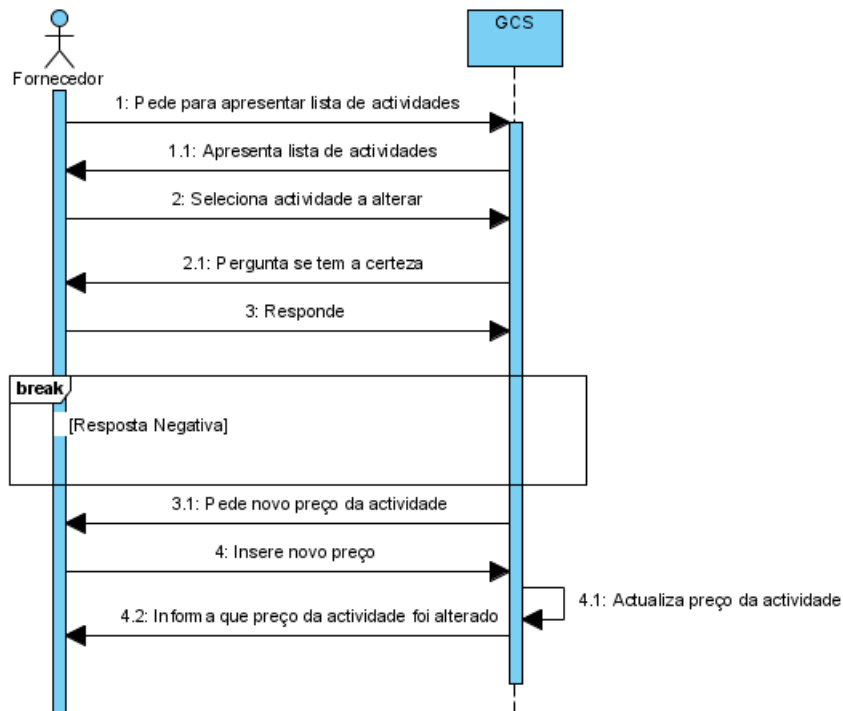


5.6 Actividades

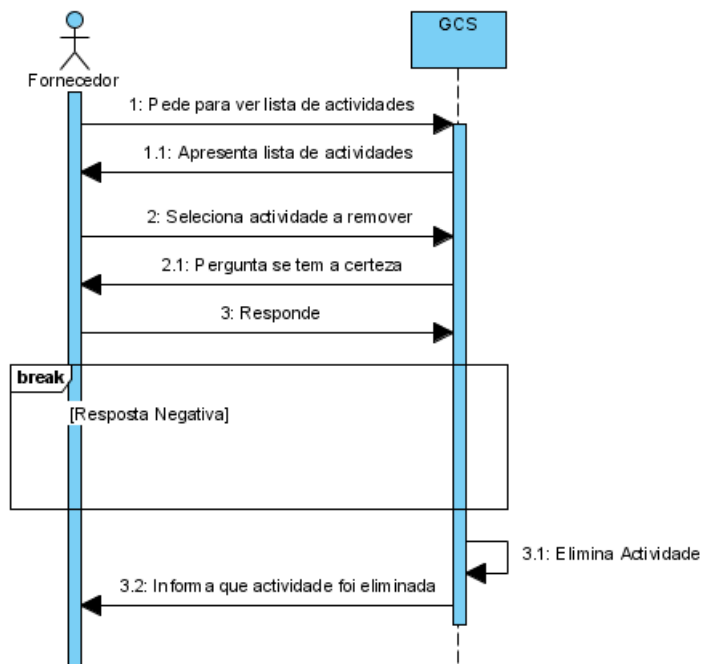
5.6.1 Adicionar Actividade



5.6.2 Alterar Preço de Actividade

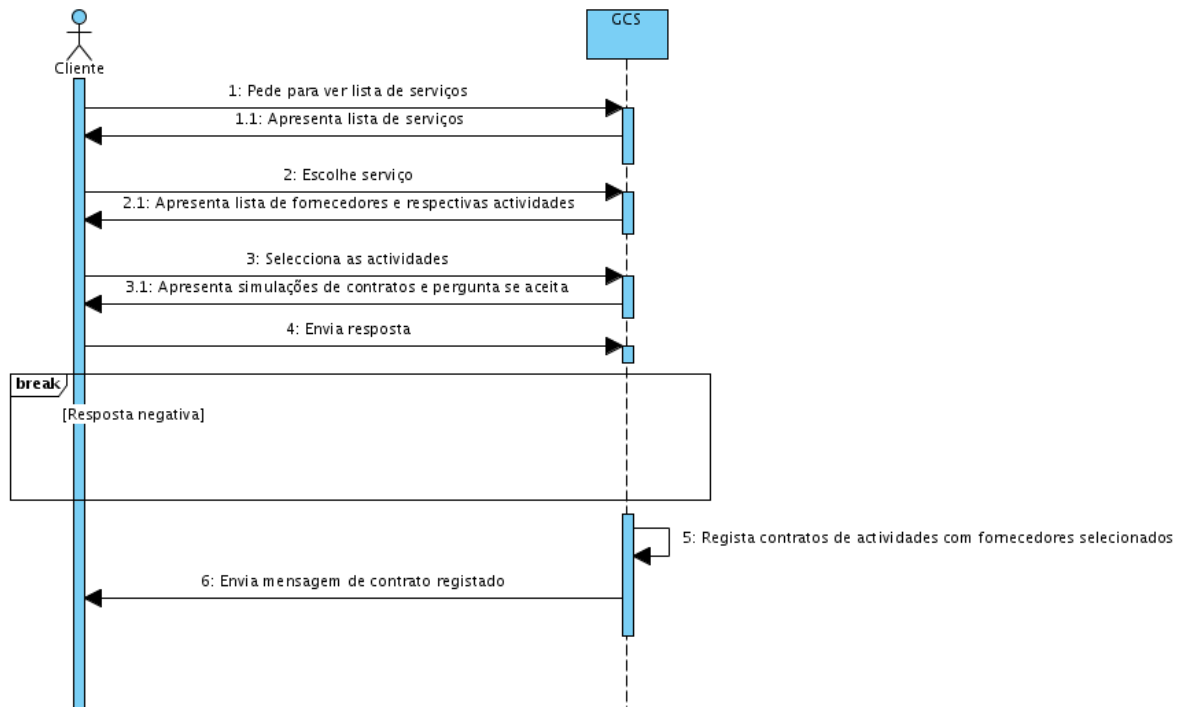


5.6.3 Remover Actividade

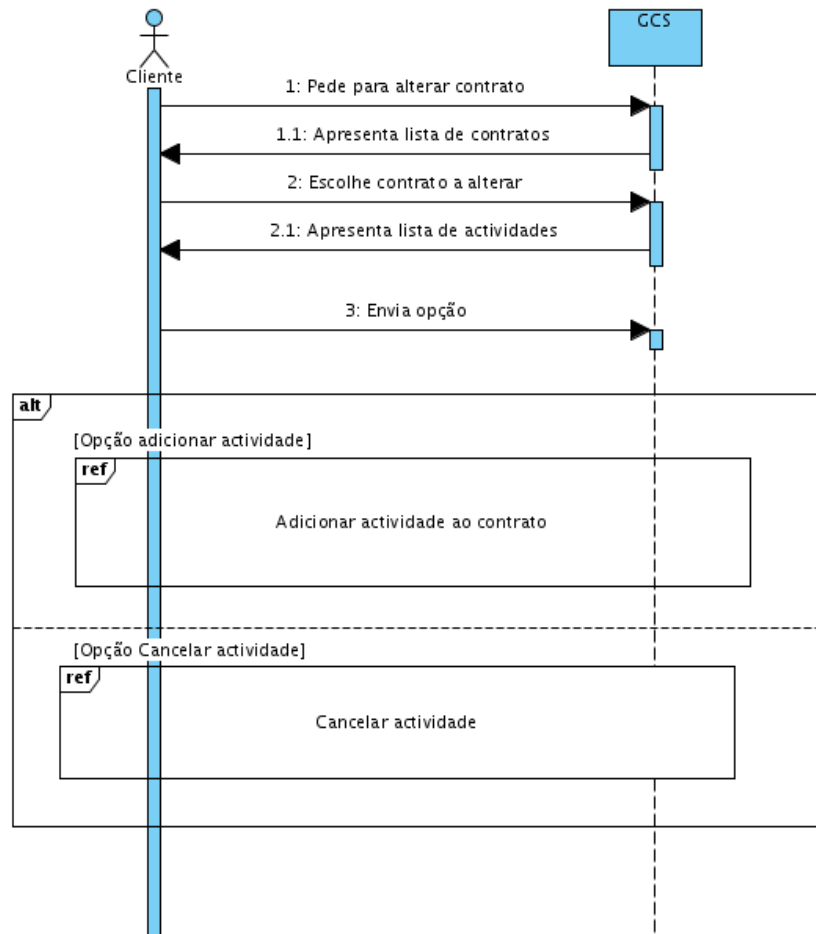


5.7 Contratos

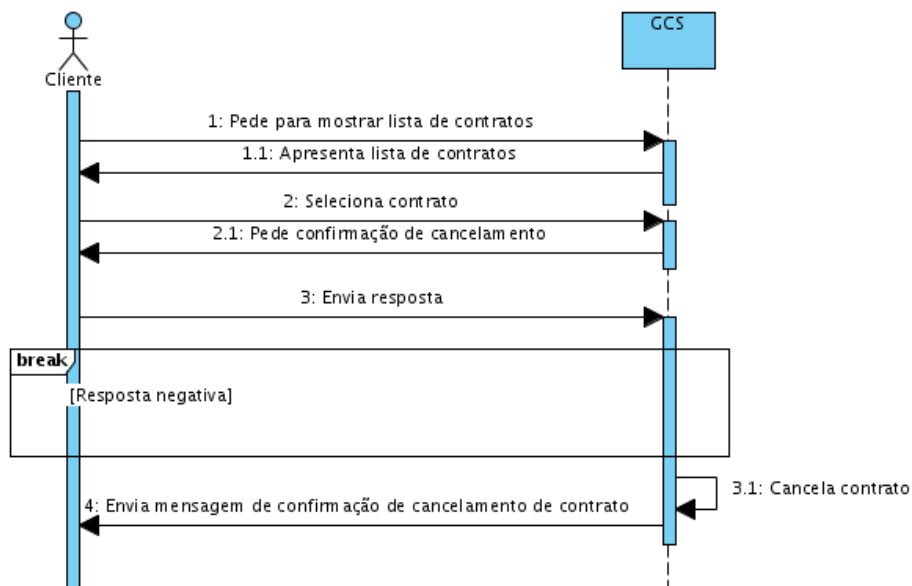
5.7.1 Criar Contrato



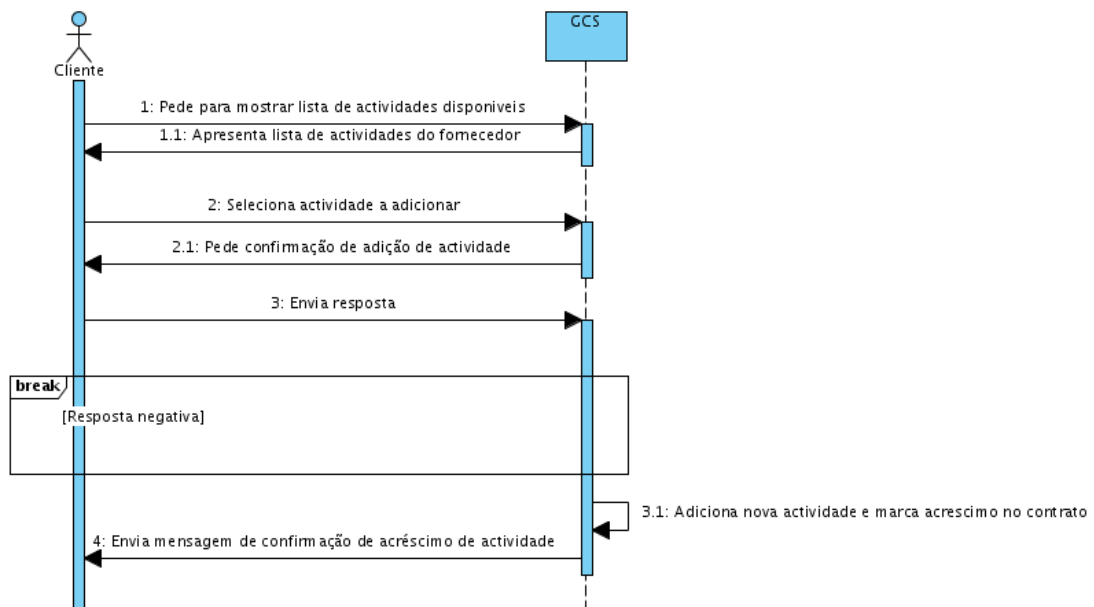
5.7.2 Alterar Contrato



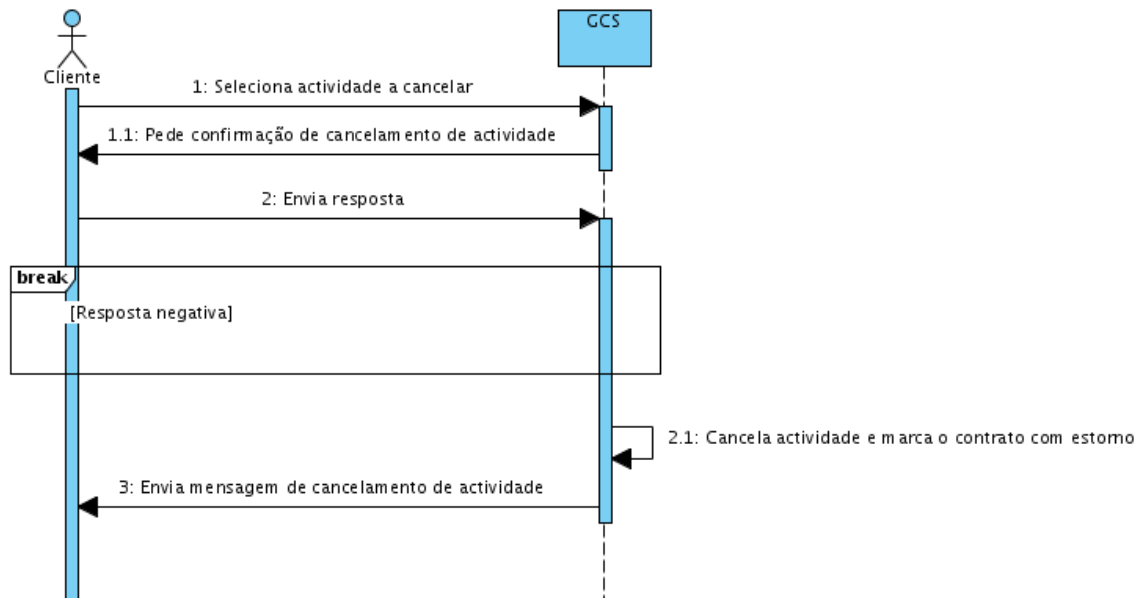
5.7.3 Cancelar Contrato



5.7.4 Adicionar Actividade ao Contrato

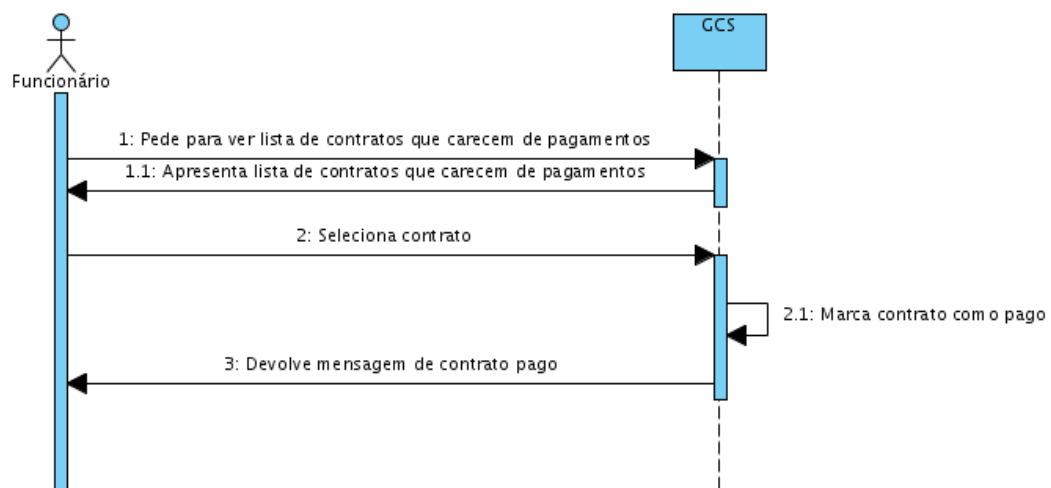


5.7.5 Cancelar Actividade do Contrato

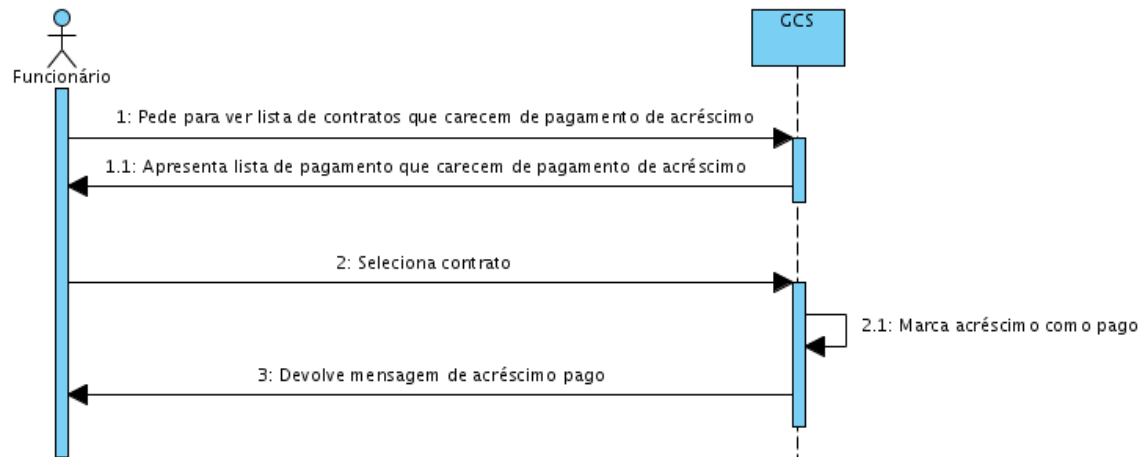


5.8 Pagamentos

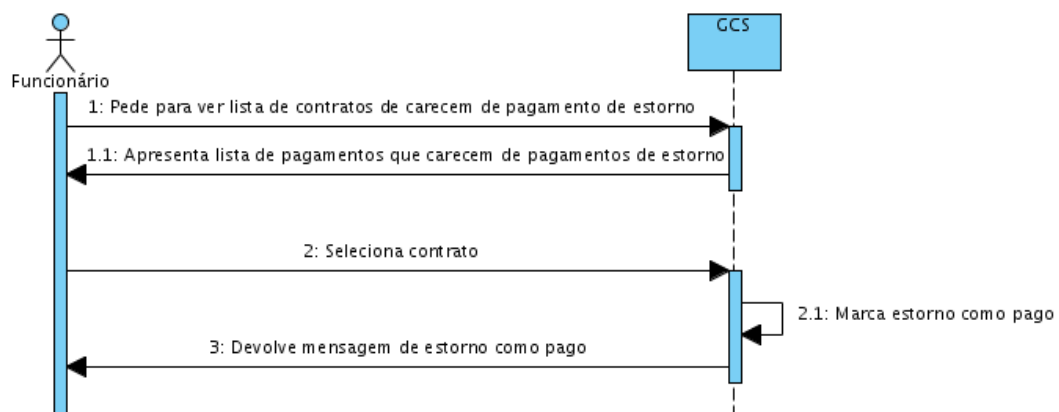
5.8.1 Confirmar Pagamento de Contrato



5.8.2 Confirmar Pagamento de Acréscimo



5.8.3 Confirmar Pagamento de Estorno

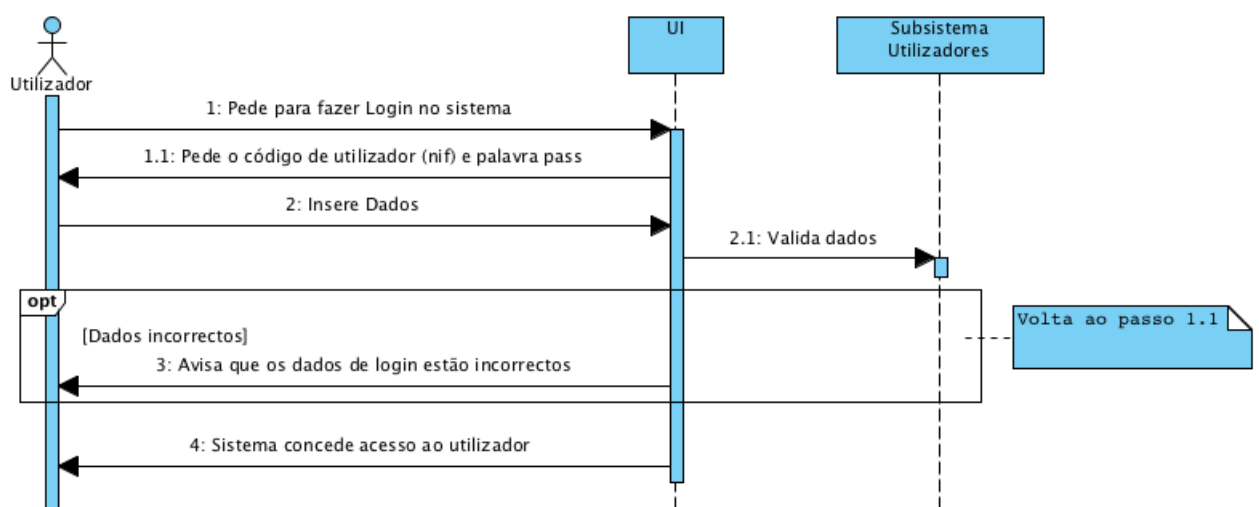


6 Diagramas de Sequência de Sub-Sistemas

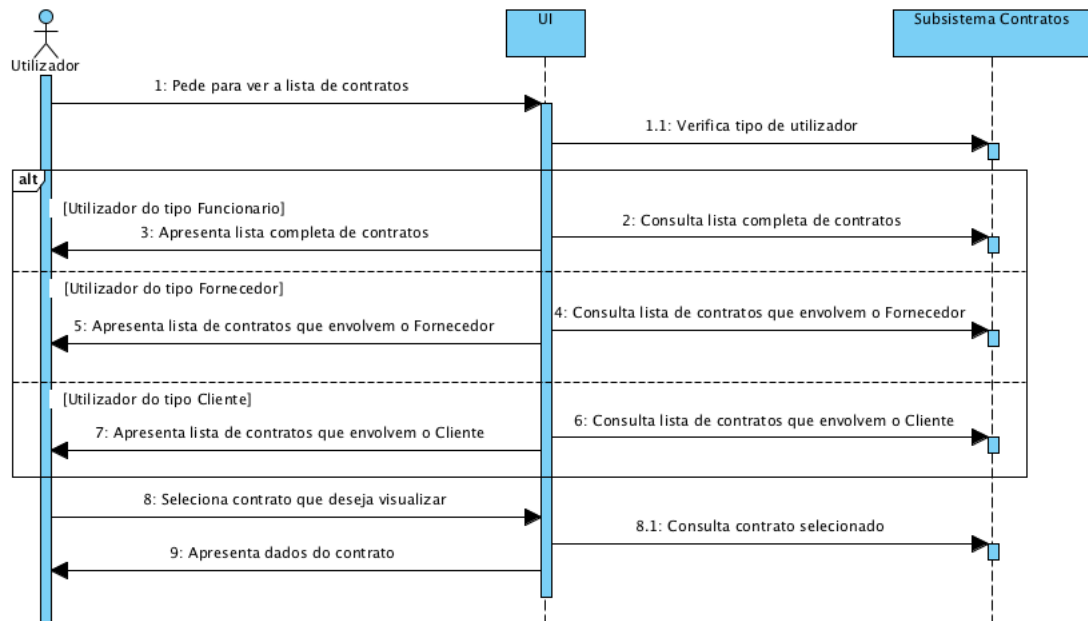
Nos diagramas de sequência de sub-sistemas identificamos três sub-sistemas sendo eles o sub-sistema de Utilizadores, de Serviços e de Contratos. Sendo assim, as operações que no diagrama de sistema tínhamos onde envolvia a gestão de clientes, funcionários e fornecedores serão agora operações que interagem com o sub-sistema de utilizadores. O mesmo acontece com as operações que envolvem actividades e serviços que agora serão realizadas com o sub-sistema serviços, assim como as operações que envolve contratos vão agora interagir com o sub-sistema Contratos. Com este refinamento que fizemos dos Diagramas de Sequência de Sistema para os Diagramas de Sequência de Sub-Sistemas aproximamo-nos mais da solução final do nosso problema, onde inicialmente todas as operações eram suportadas por um Sistema principal, e depois passaram a ser suportadas por vários sub-sistemas.

6.1 Geral

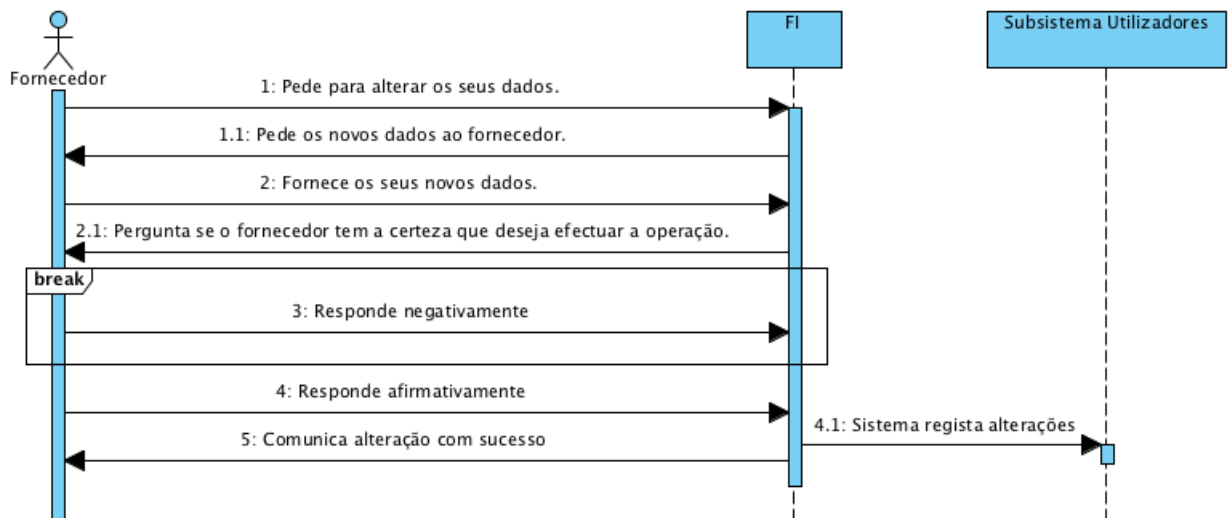
6.1.1 Login



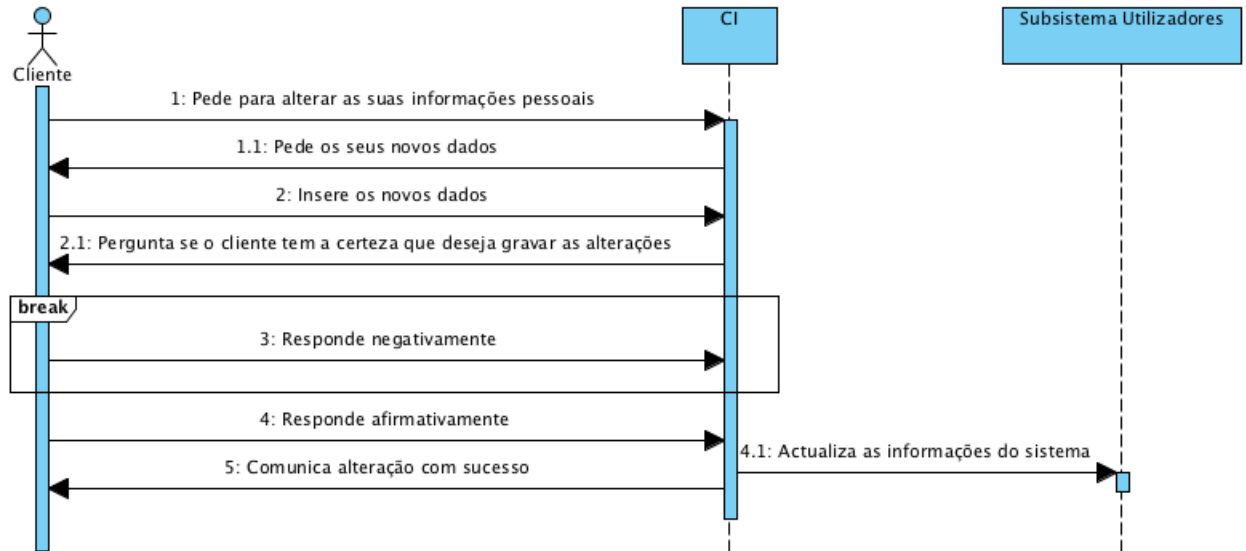
6.1.2 Consultar Contrato



6.1.3 Alterar Dados

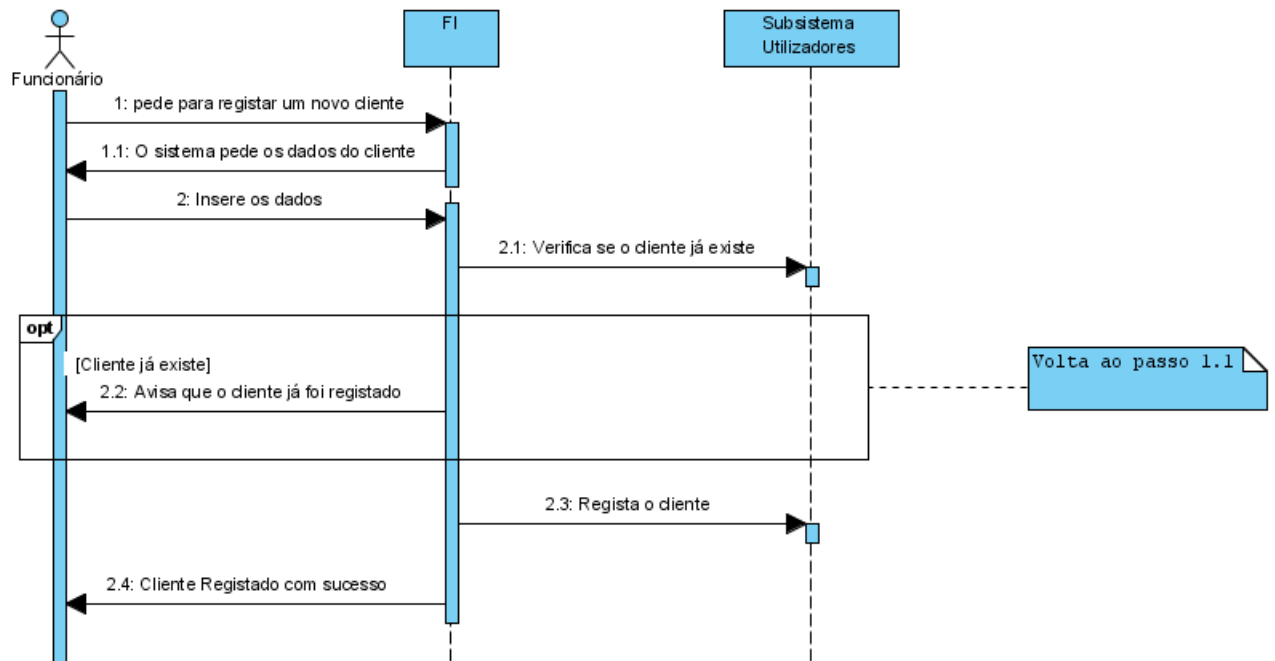


6.1.4 Alterar Informação Pessoal

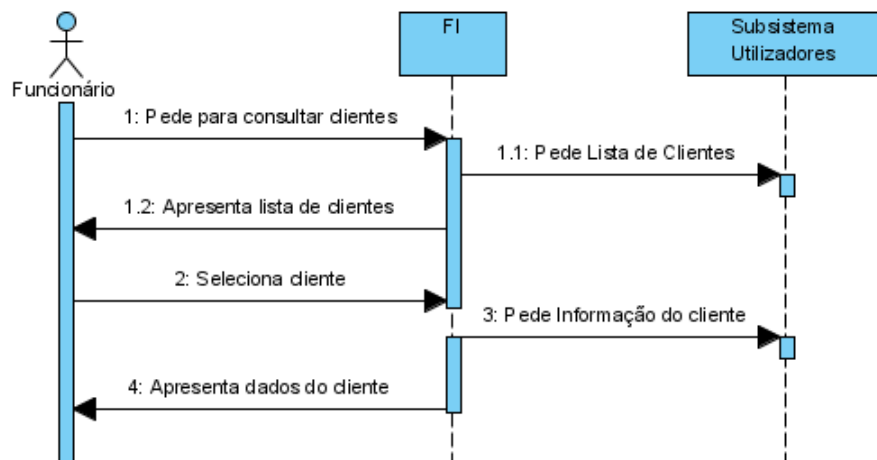


6.2 Clientes

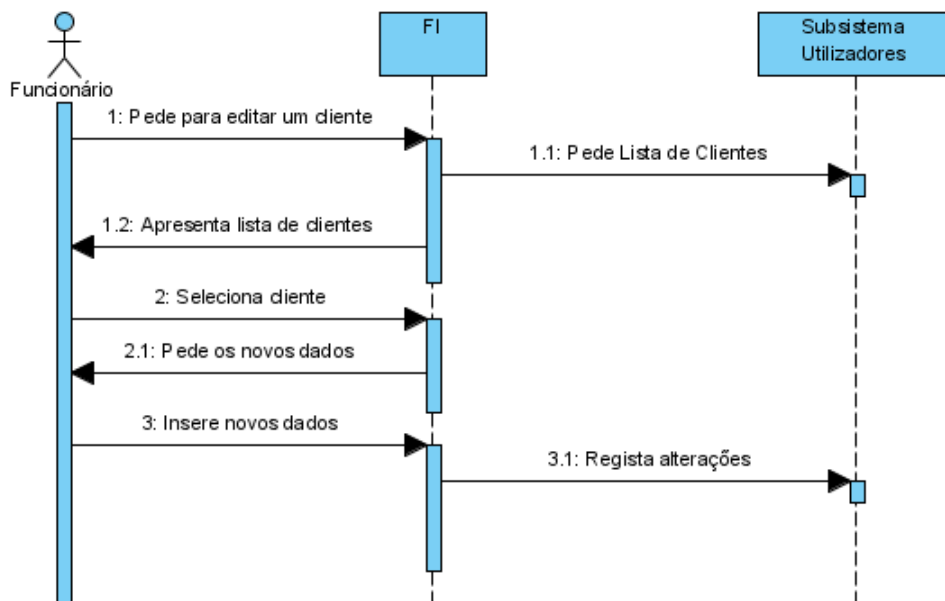
6.2.1 Adicionar Cliente



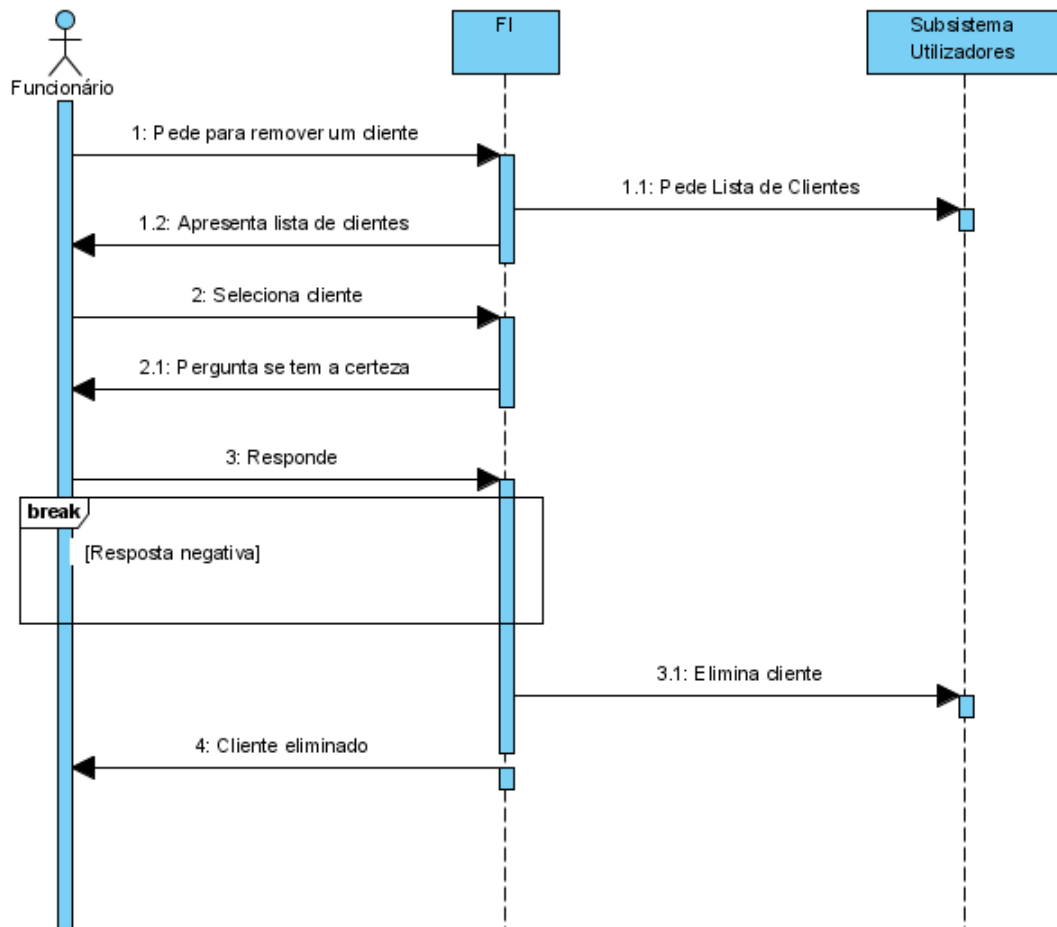
6.2.2 Consultar Cliente



6.2.3 Editar Cliente

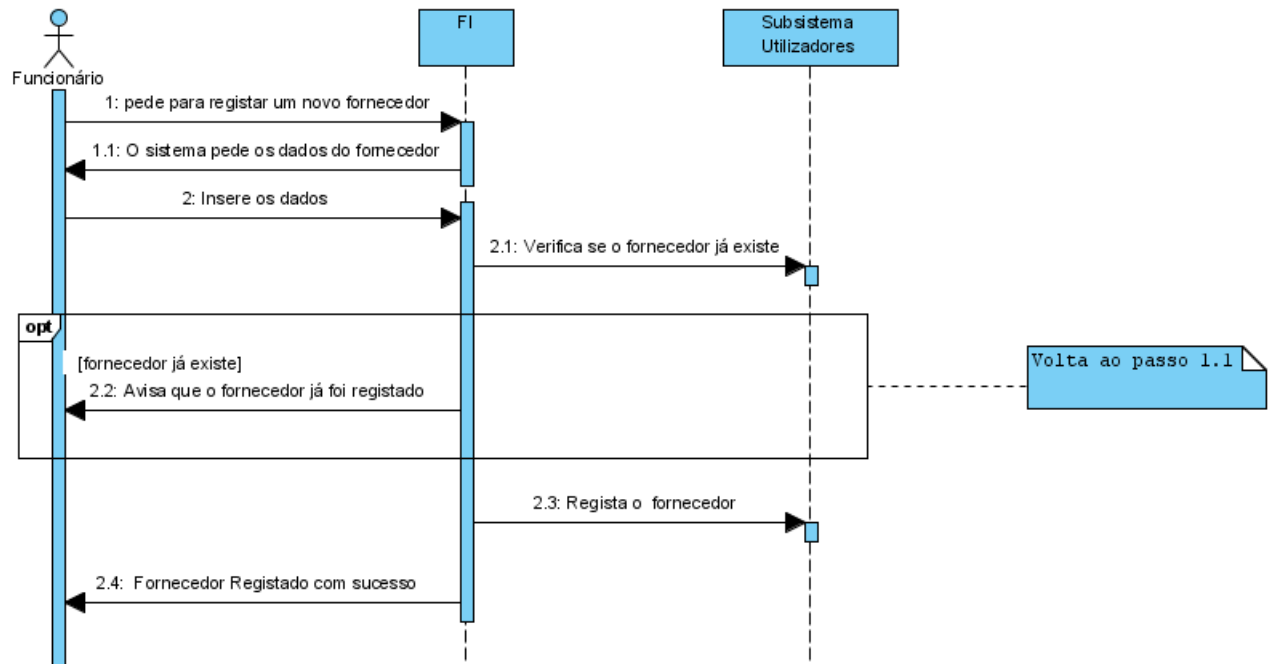


6.2.4 Remover Cliente

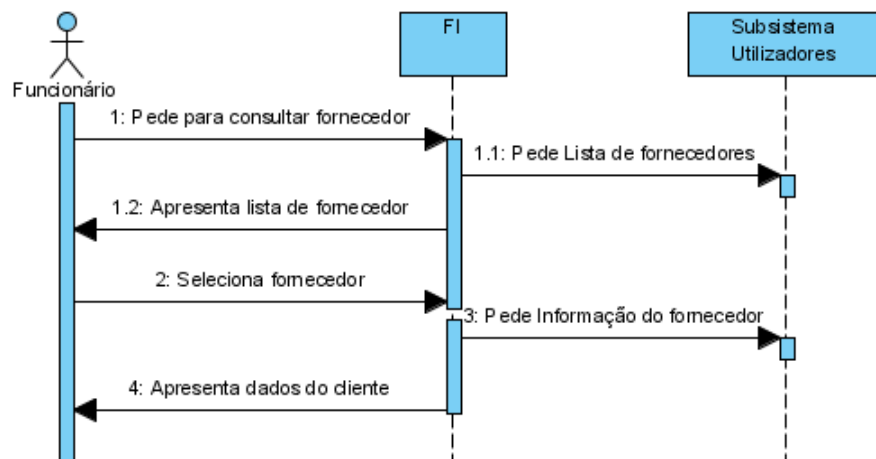


6.3 Fornecedores

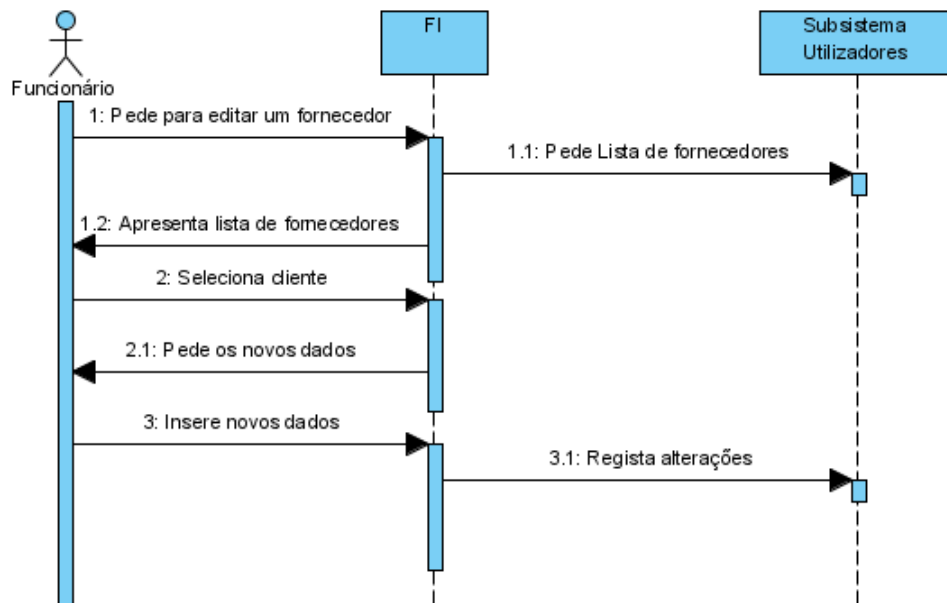
6.3.1 Adicionar Fornecedor



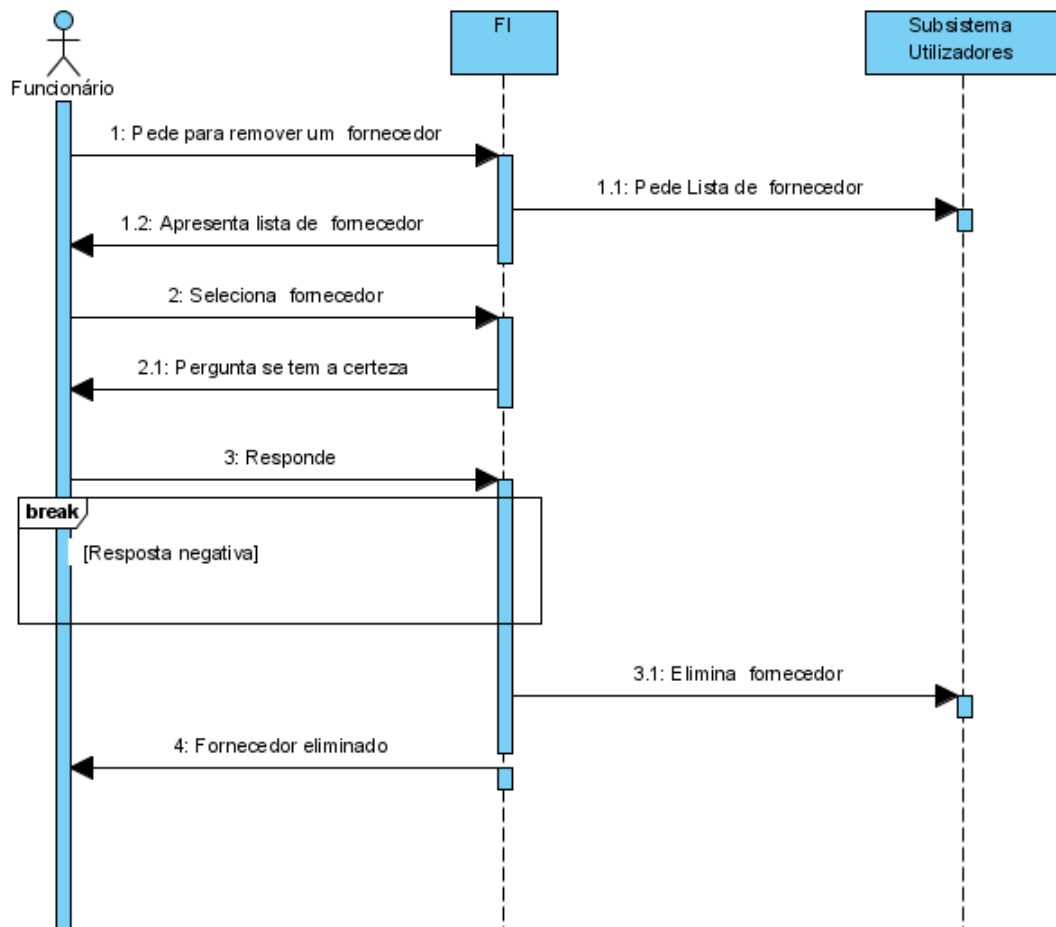
6.3.2 Consultar Fornecedor



6.3.3 Editar Fornecedor

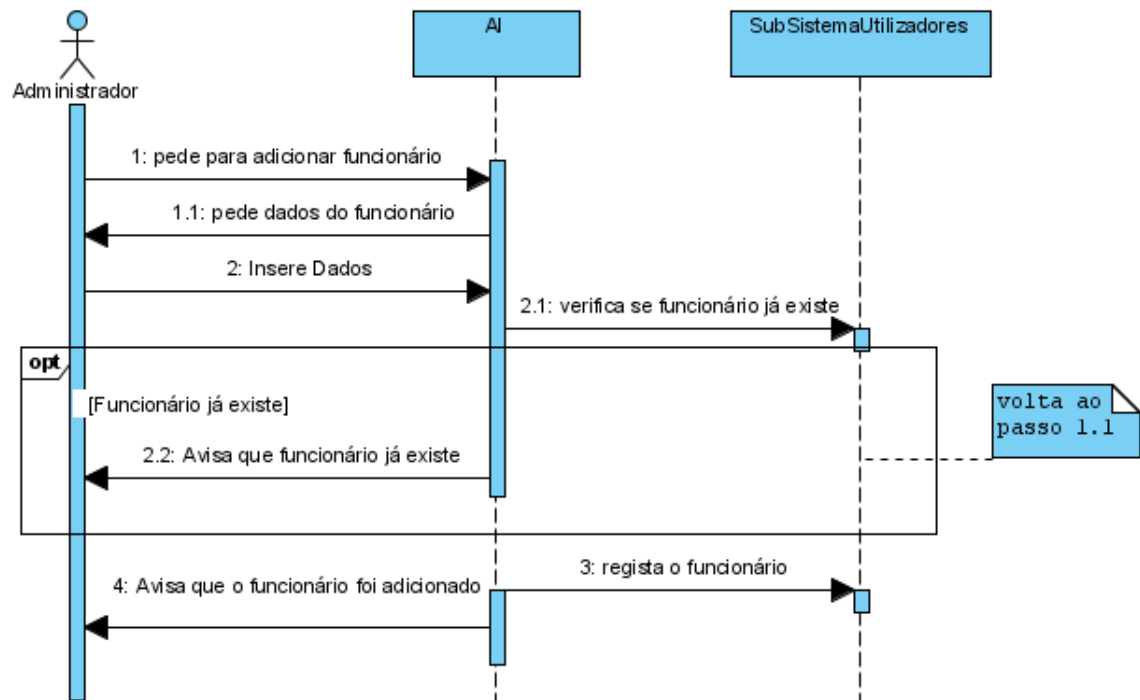


6.3.4 Remover Fornecedor

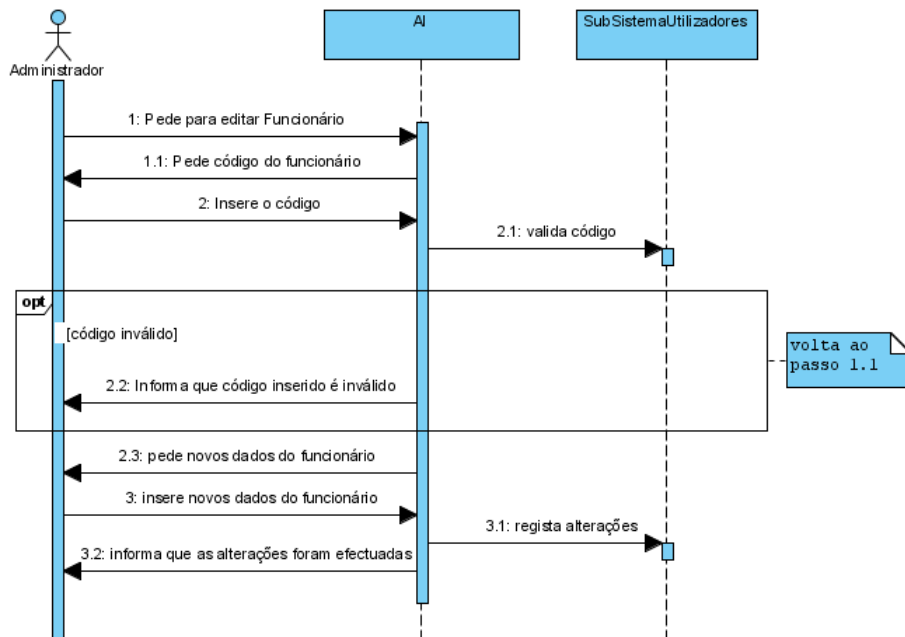


6.4 Funcionários

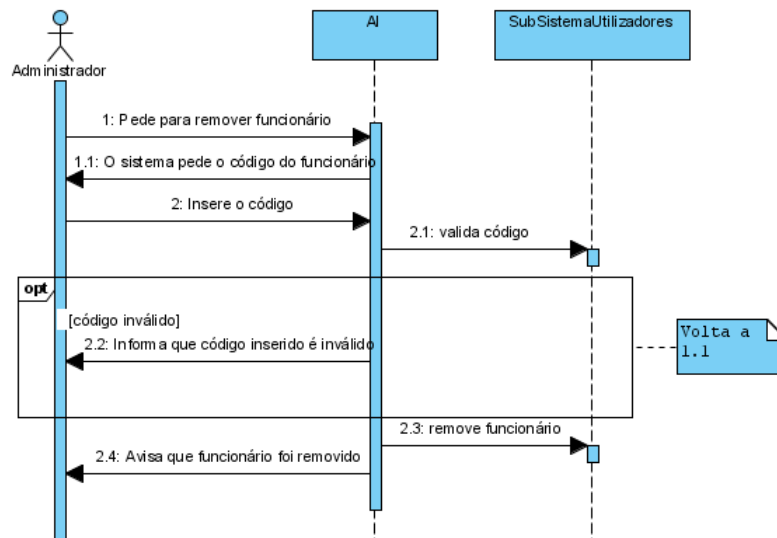
6.4.1 Adicionar Funcionário



6.4.2 Editar Funcionário

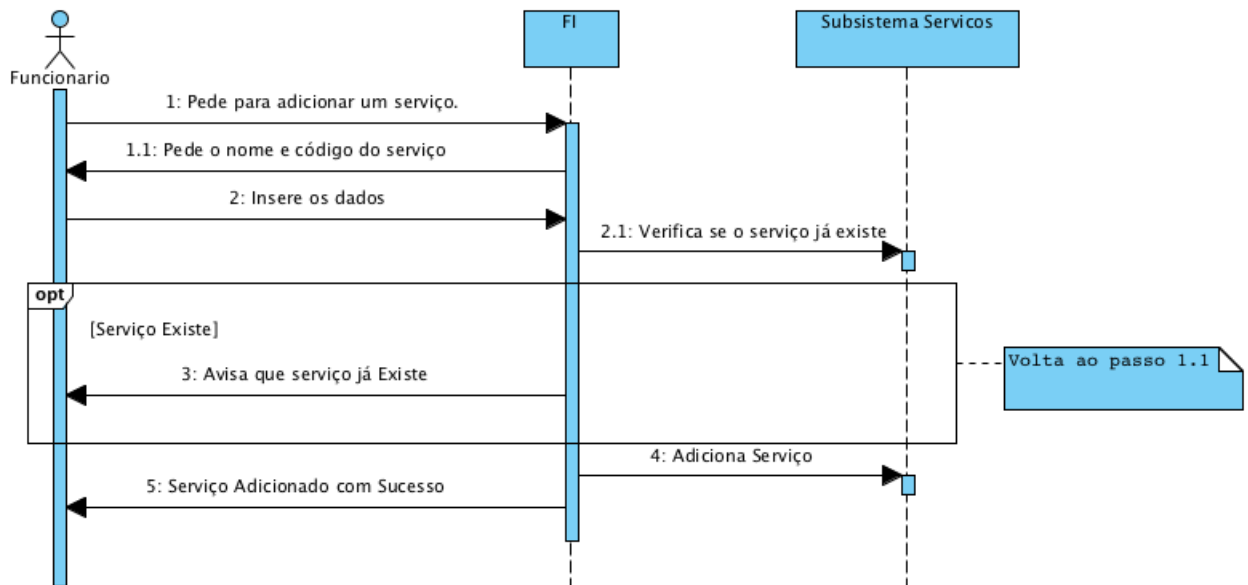


6.4.3 Remover Funcionário

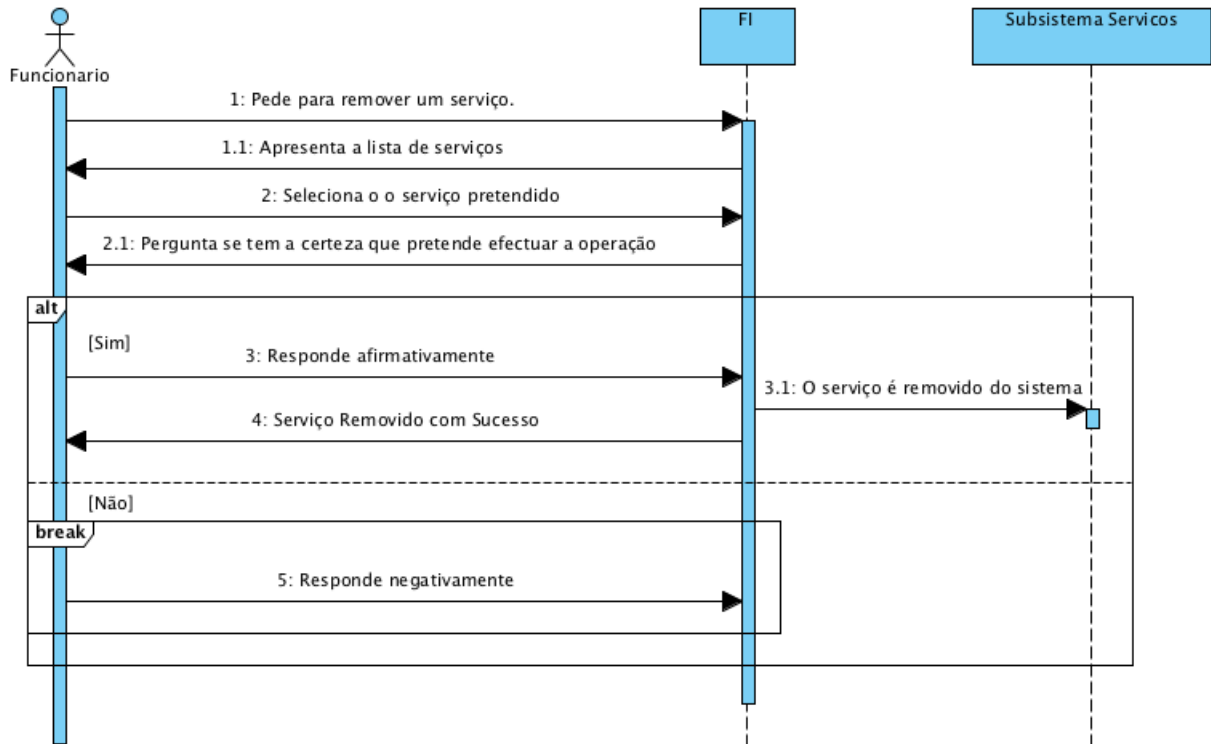


6.5 Serviços

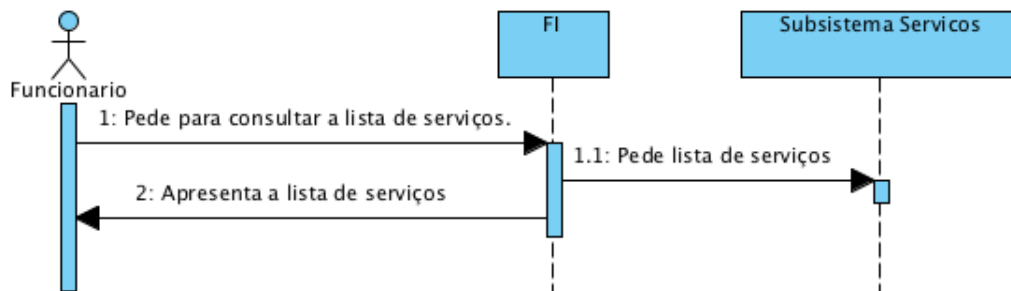
6.5.1 Adicionar Serviço



6.5.2 Remover Serviço

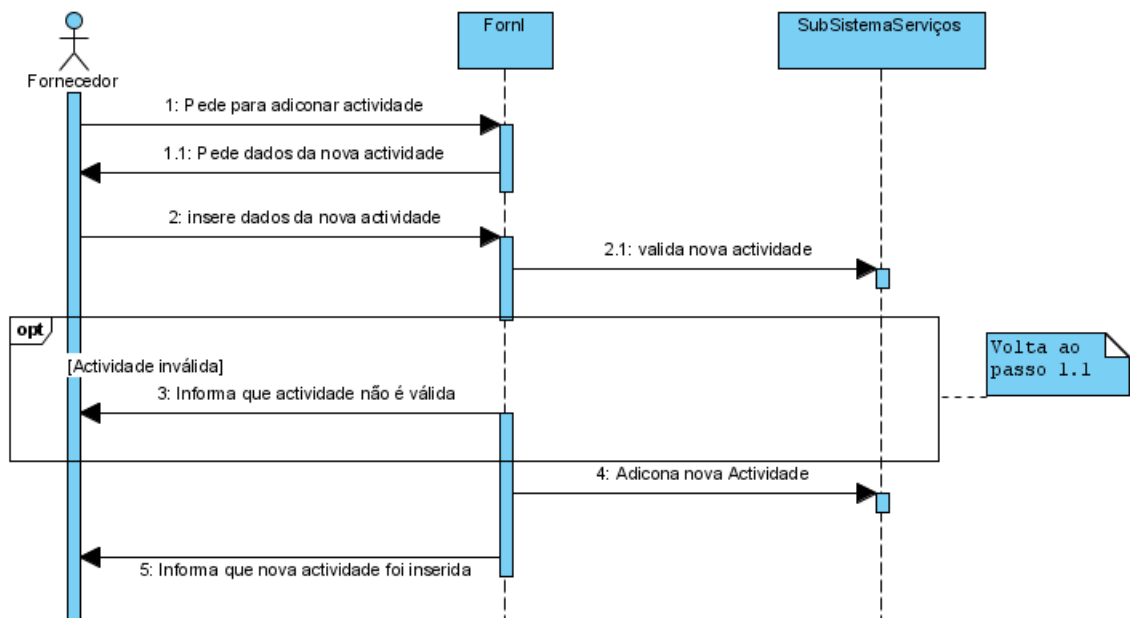


6.5.3 Consultar Lista de Serviços

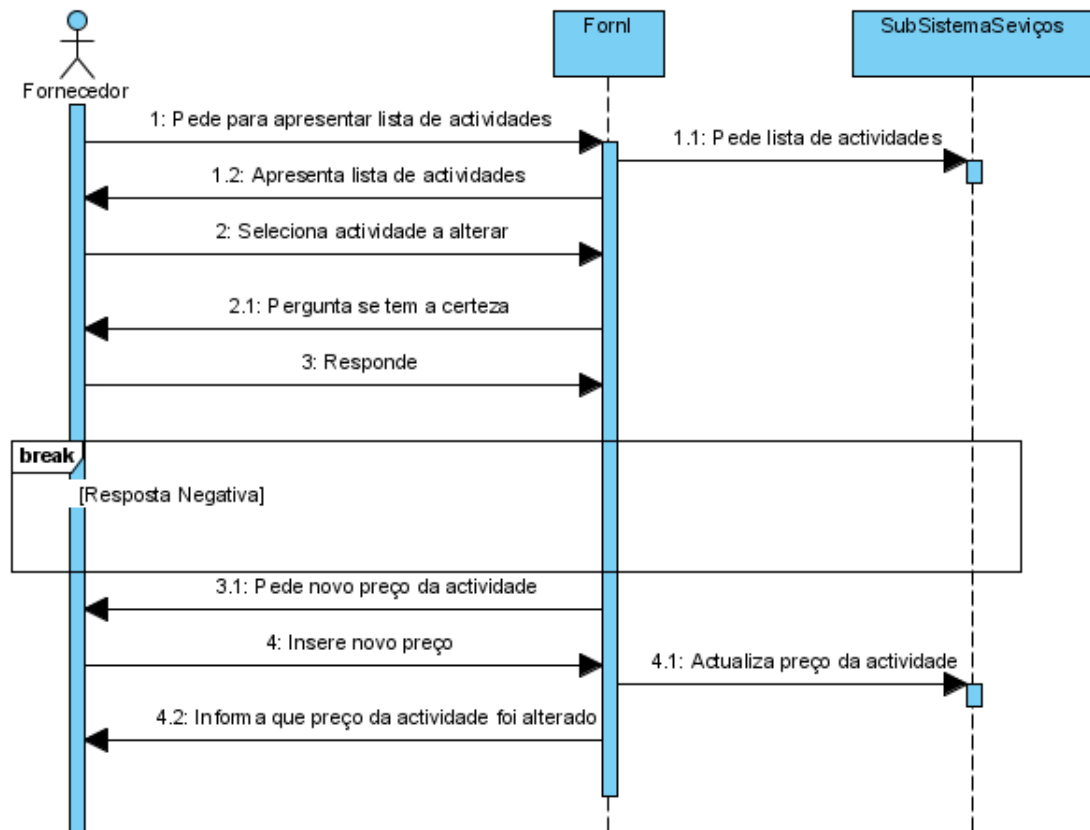


6.6 Actividades

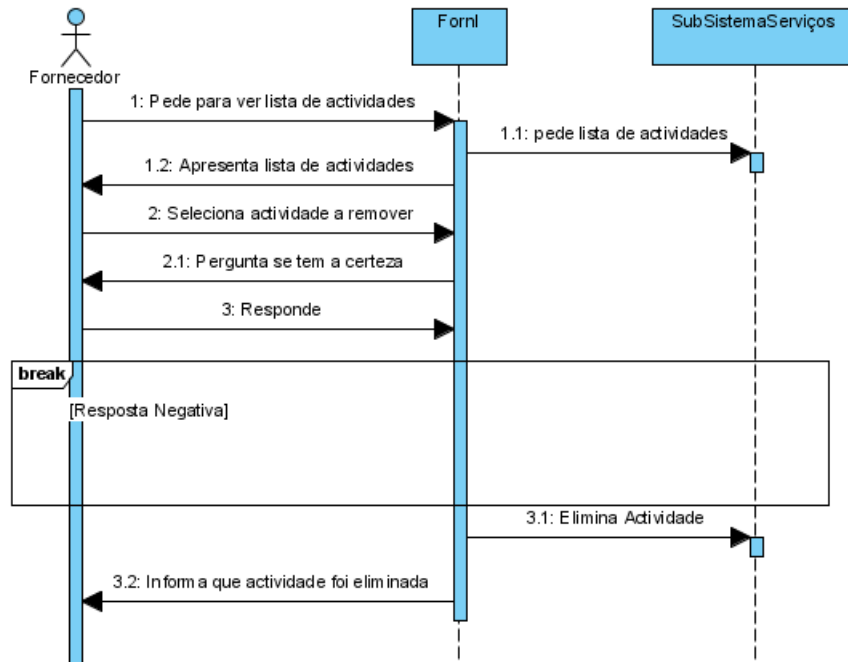
6.6.1 Adicionar Actividade



6.6.2 Alterar Preço de Actividade

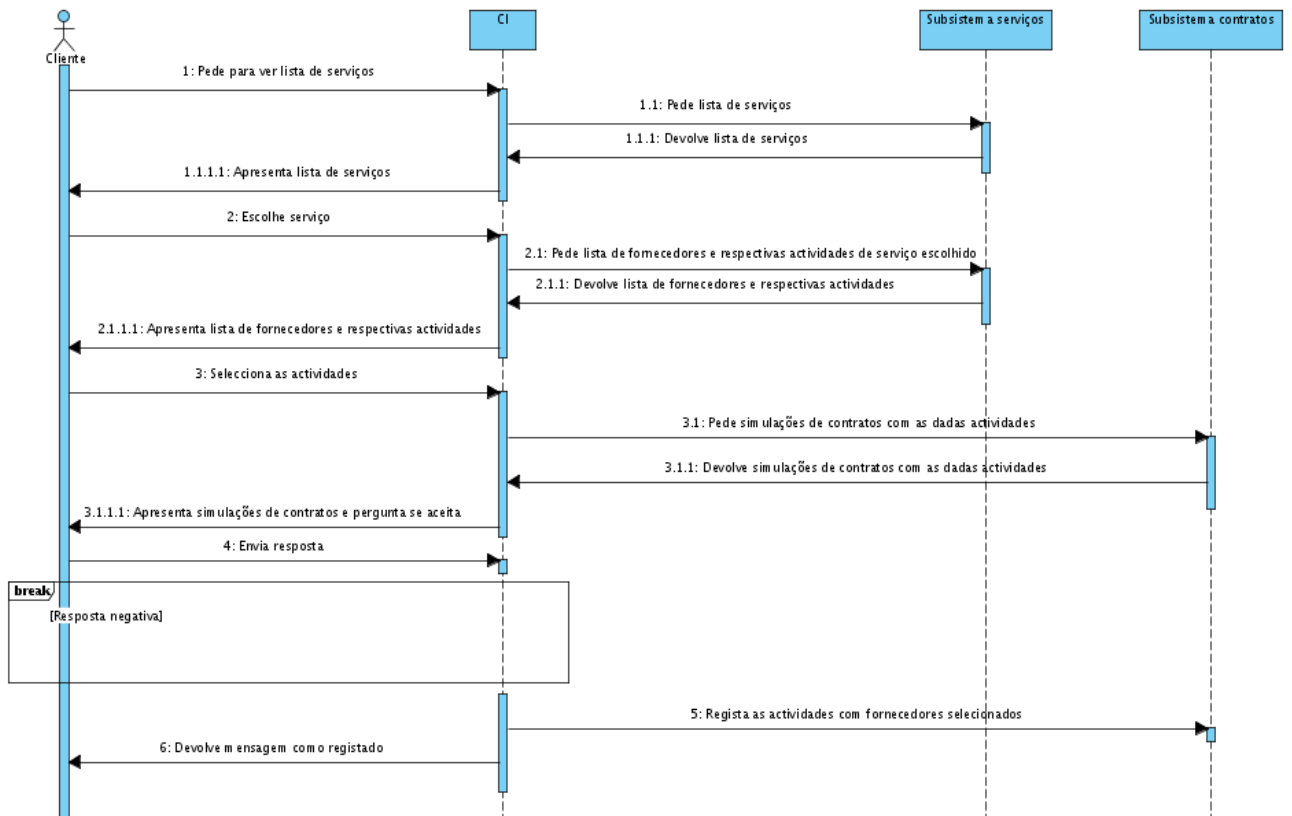


6.6.3 Remover Actividade

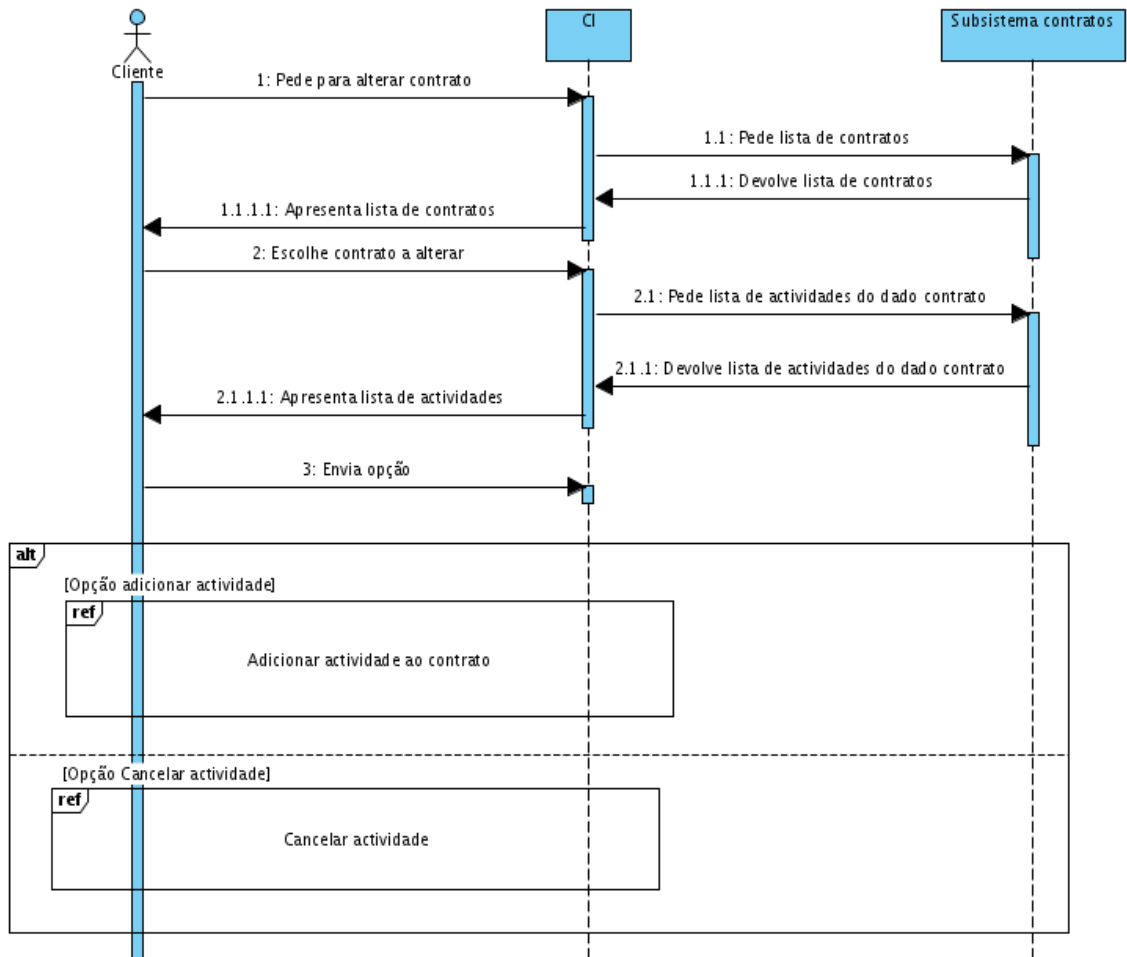


6.7 Contratos

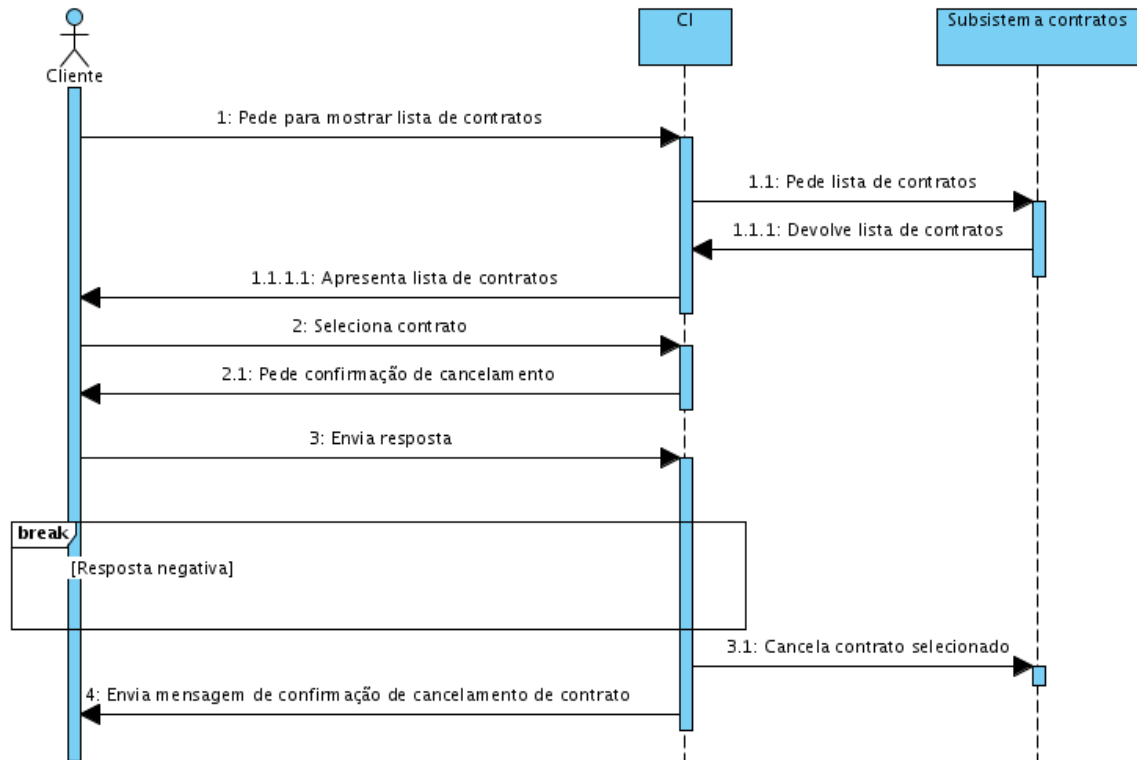
6.7.1 Criar Contrato



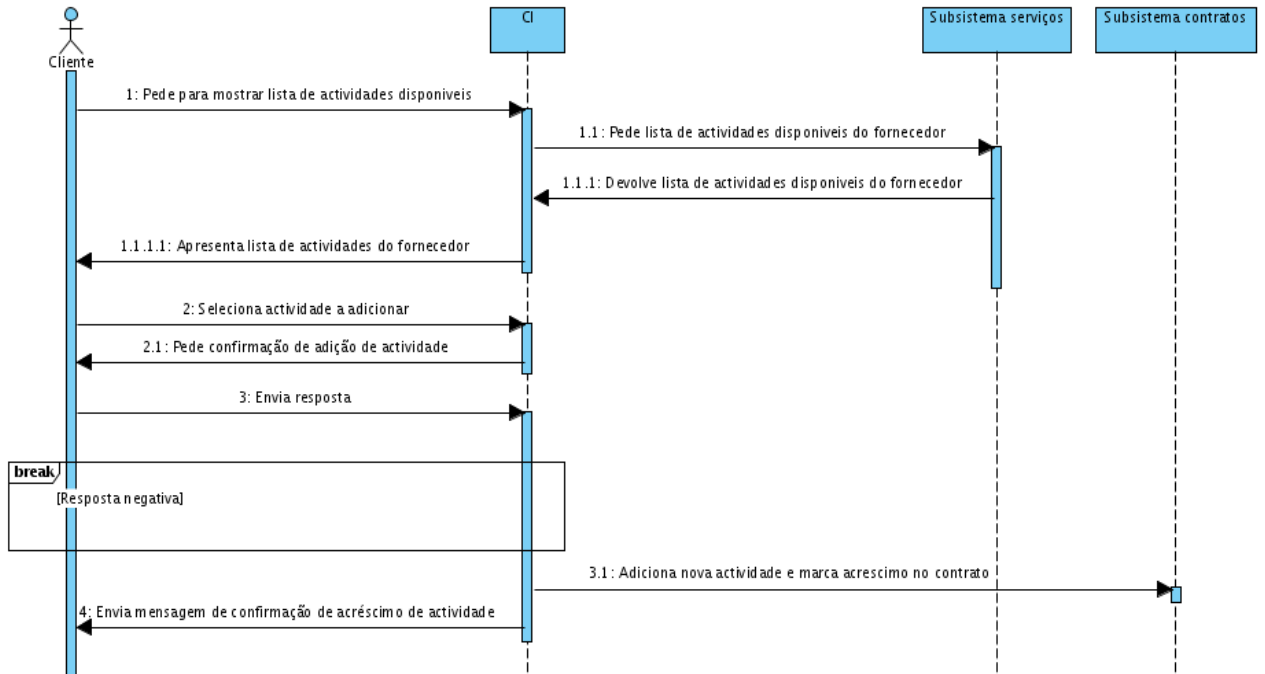
6.7.2 Alterar Contrato



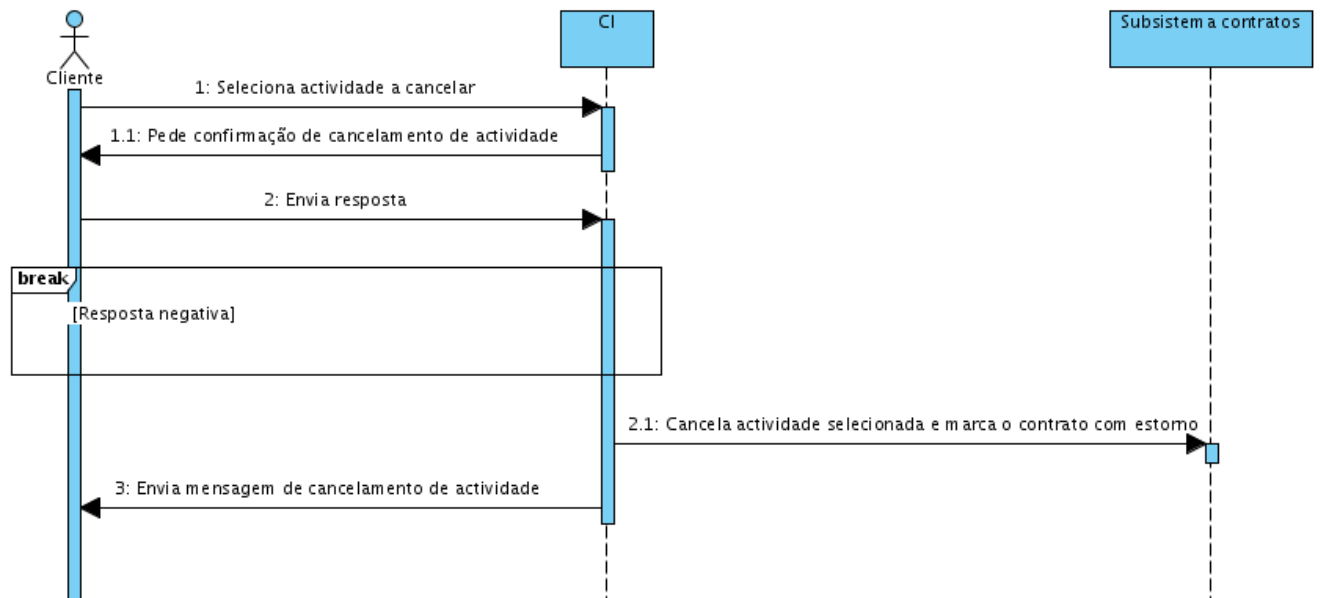
6.7.3 Cancelar Contrato



6.7.4 Adicionar Actividade ao Contrato

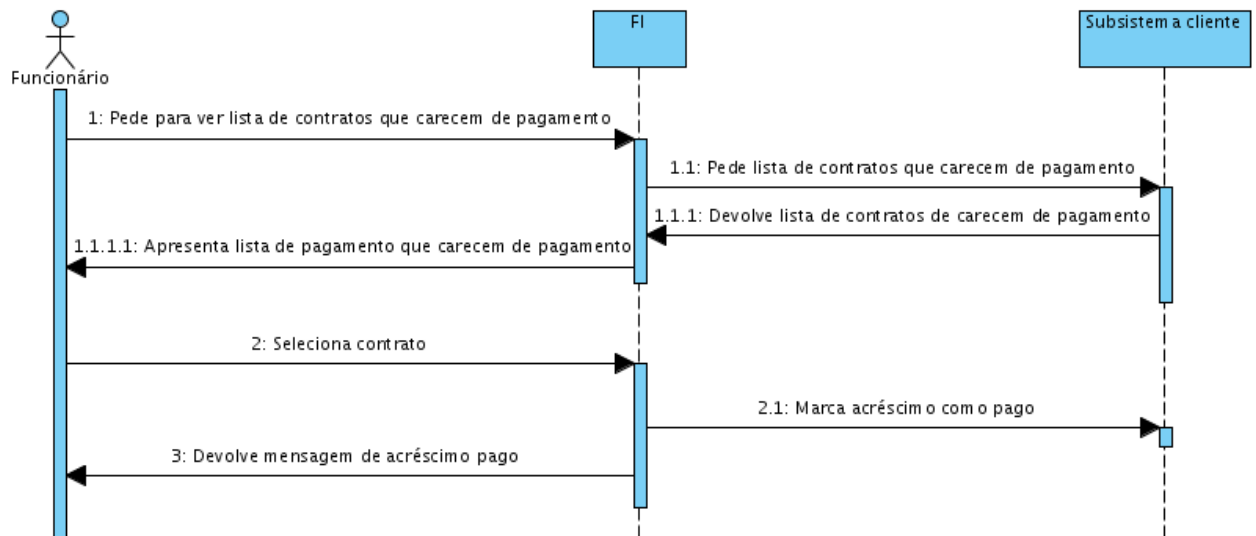


6.7.5 Cancelar Actividade do Contrato

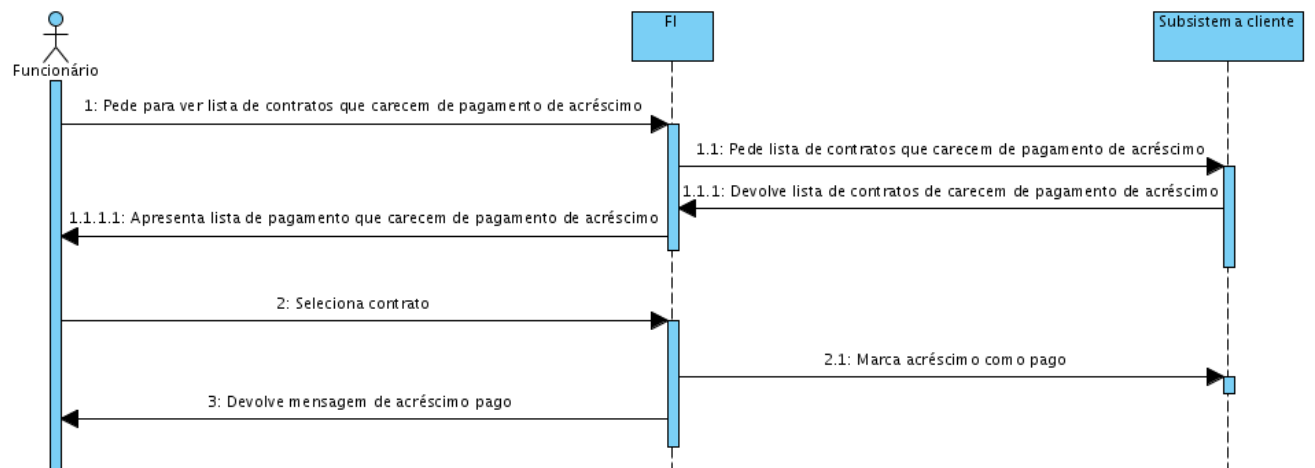


6.8 Pagamentos

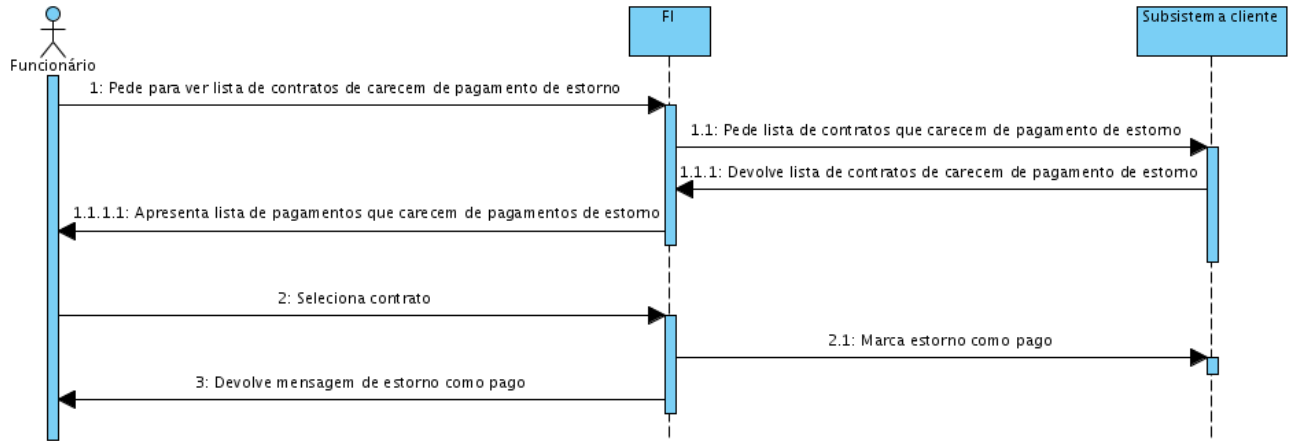
6.8.1 Confirmar Pagamento de Contrato



6.8.2 Confirmar Pagamento de Acréscimo



6.8.3 Confirmar Pagamento de Estorno



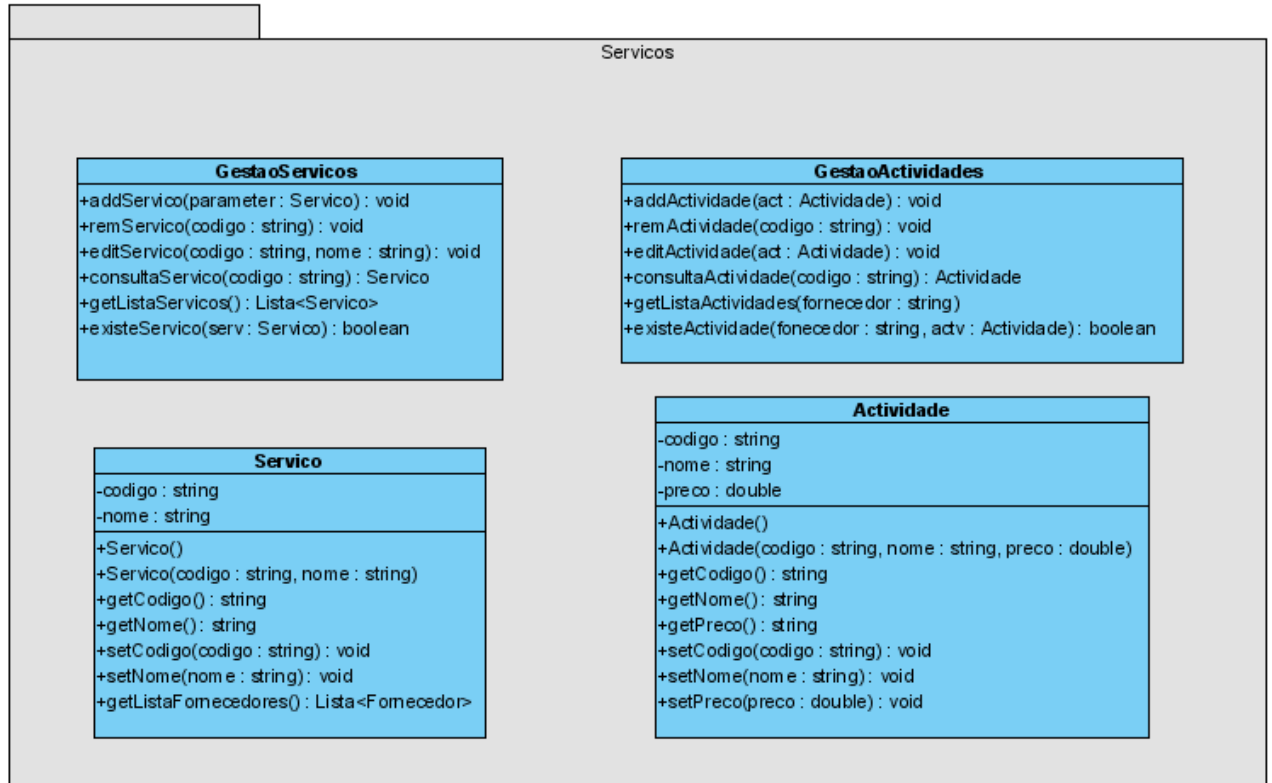
7 Diagramas de Classes

O diagrama de classes está dividido em três sub-diagramas (packages) cada um com as classes dos sub-sistemas identificados anteriormente:

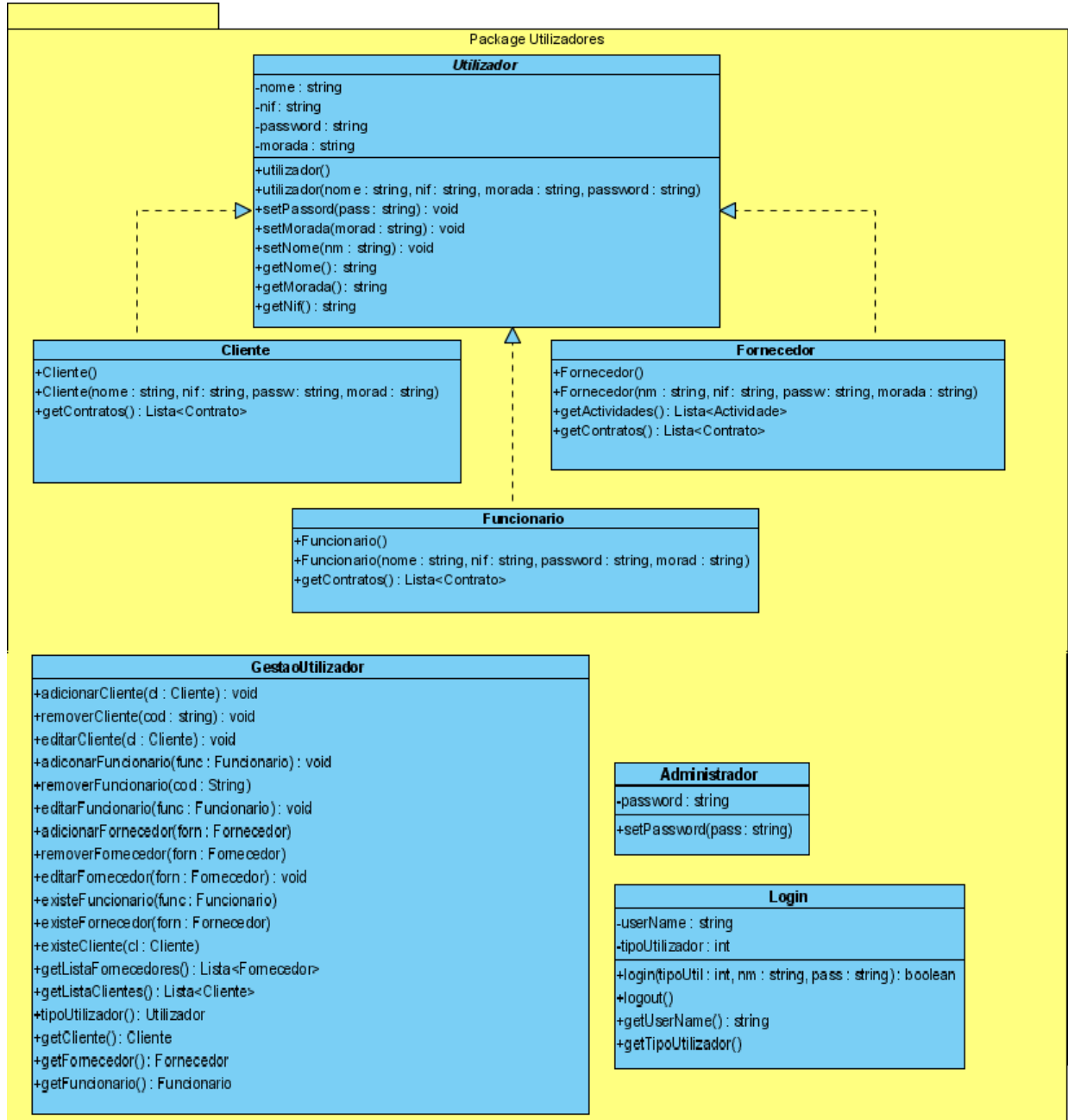
- Package Serviços;
- Package Utilizadores;
- Package Contratos.

O recurso ao uso de packages dá uma maior modularidade ao código do projecto e facilita a distribuição de tarefas pelos membros do grupo.

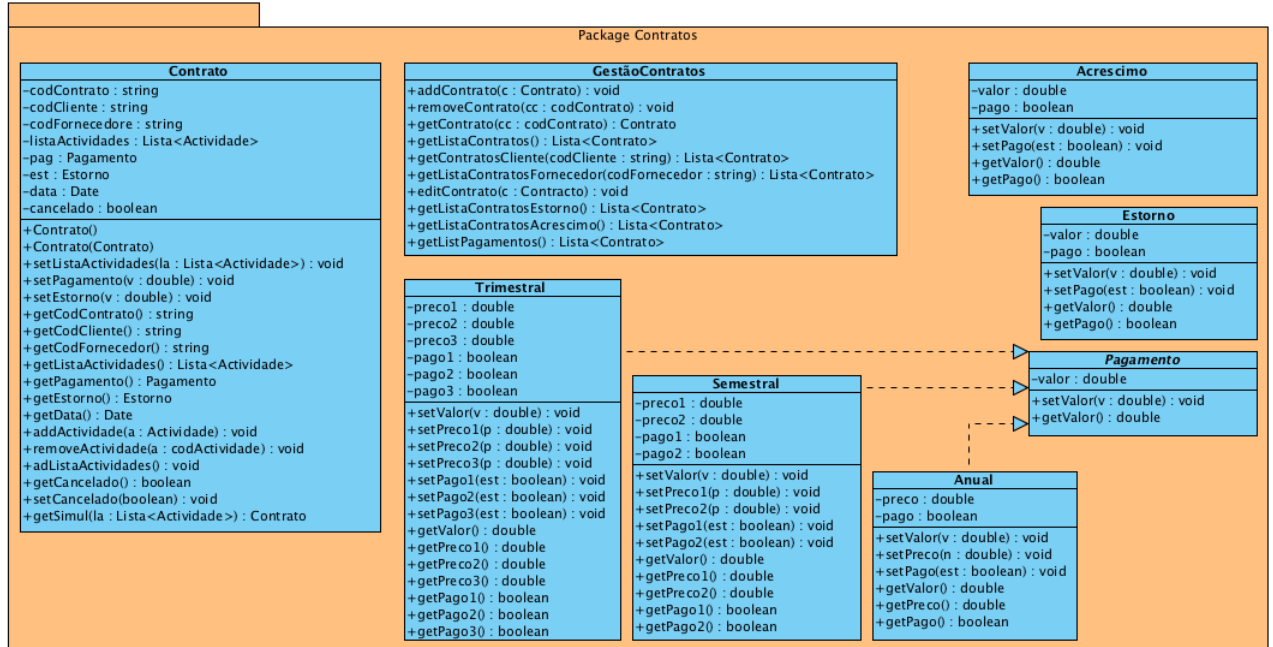
7.1 Package Serviços



7.2 Package Utilizadores



7.3 Package Contratos

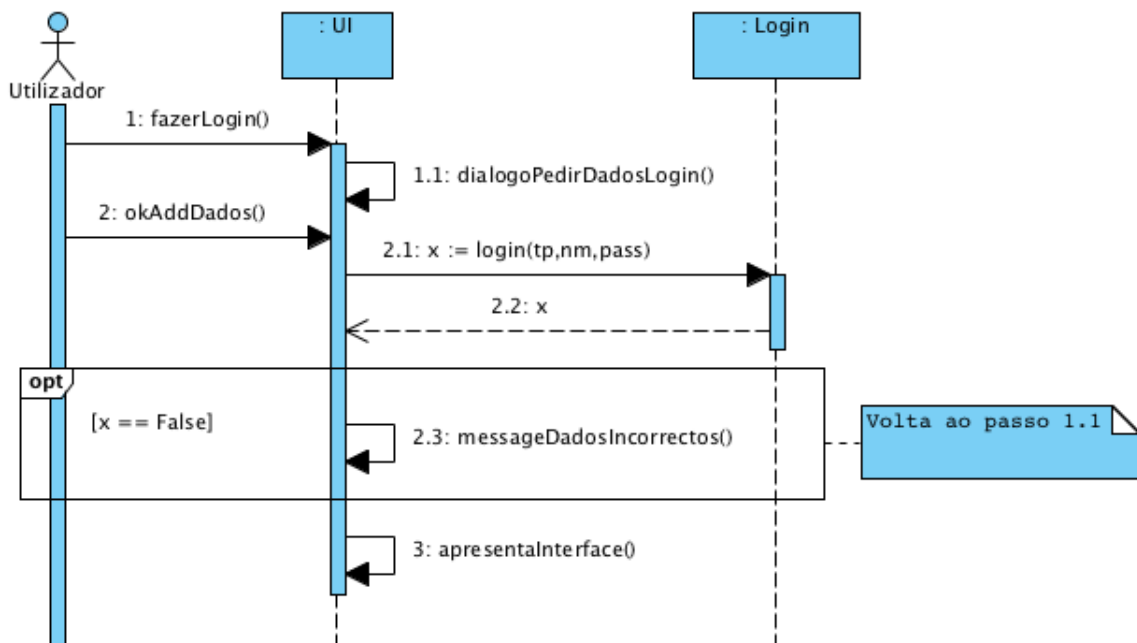


8 Diagramas de Sequência de Implementação

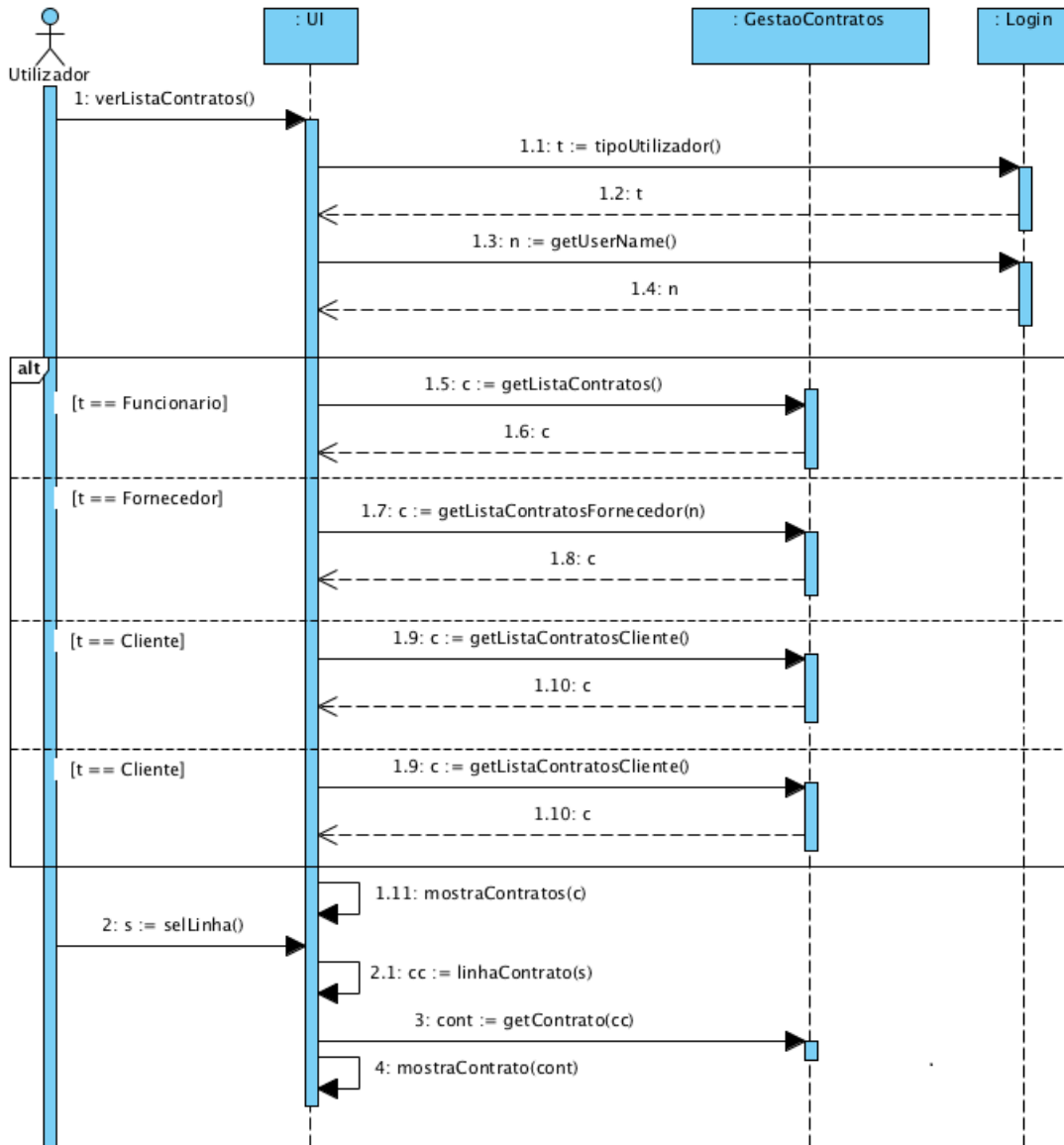
Os diagramas de sequência de implementação são os diagramas que mais se aproximam da solução final do problema. Nestes diagramas já utilizamos, em vez de sub-sistemas, classes já definidas por nós no Diagrama de Classes assim como trocamos as operações, que até agora eram descritas com linguagem corrente, por métodos que já foram definidos também no diagrama de classes, e sendo esses os métodos que mais tarde serão implementados.

8.1 Geral

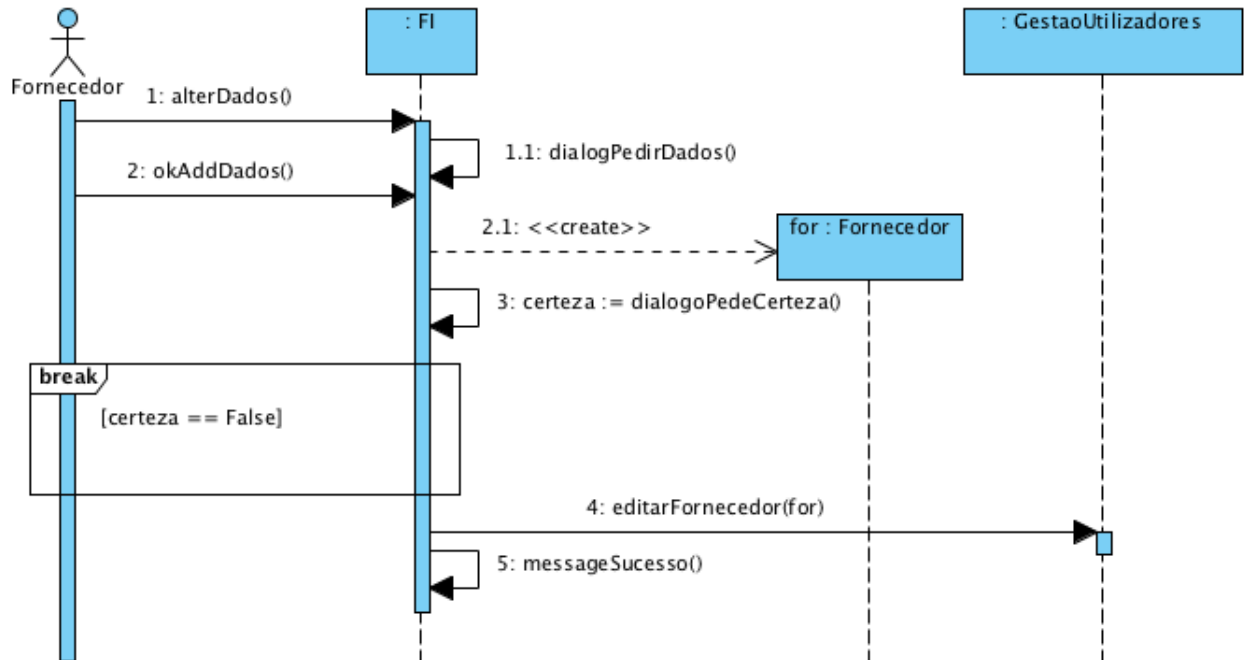
8.1.1 Login



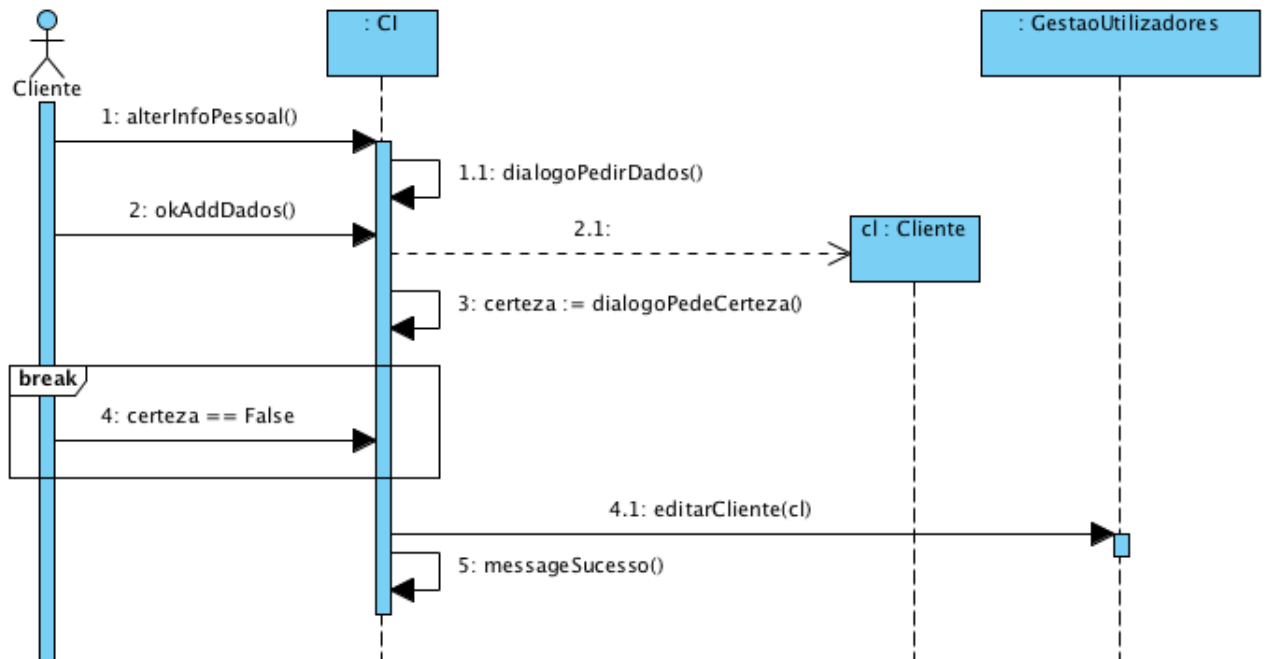
8.1.2 Consultar Contrato



8.1.3 Alterar Dados

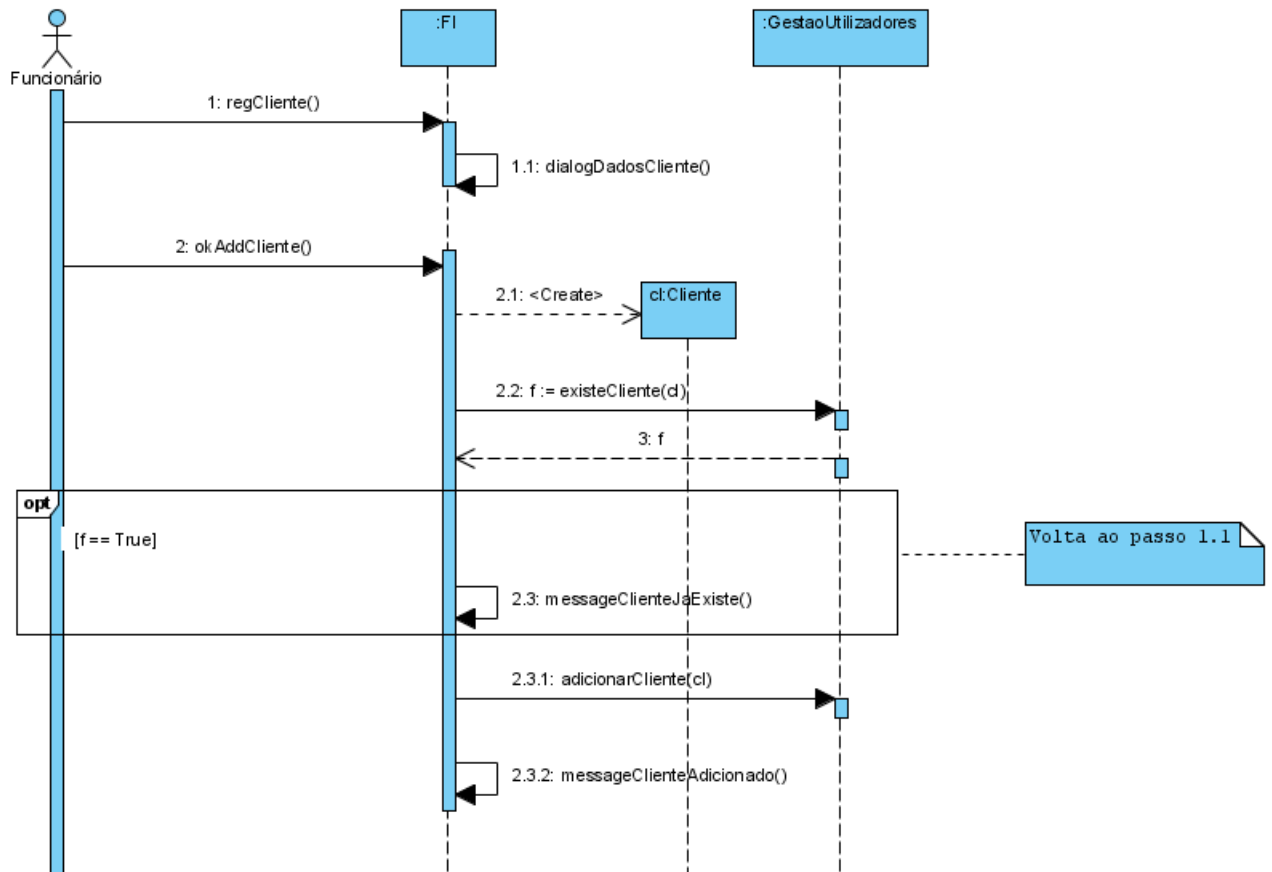


8.1.4 Alterar Informação Pessoal

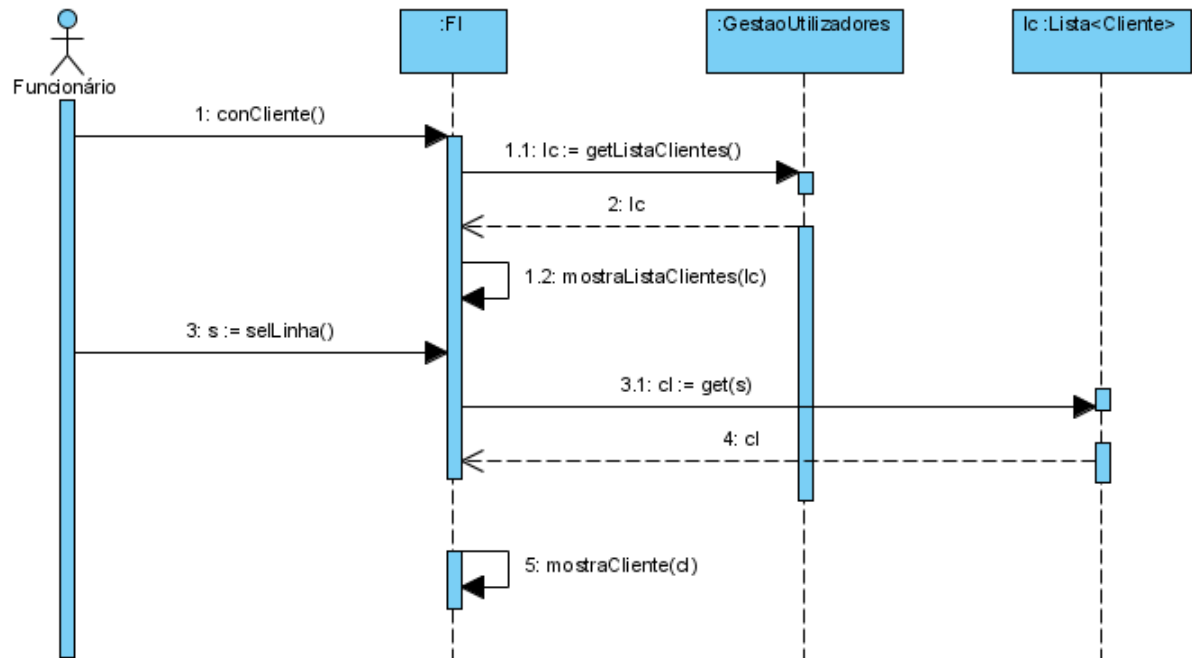


8.2 Clientes

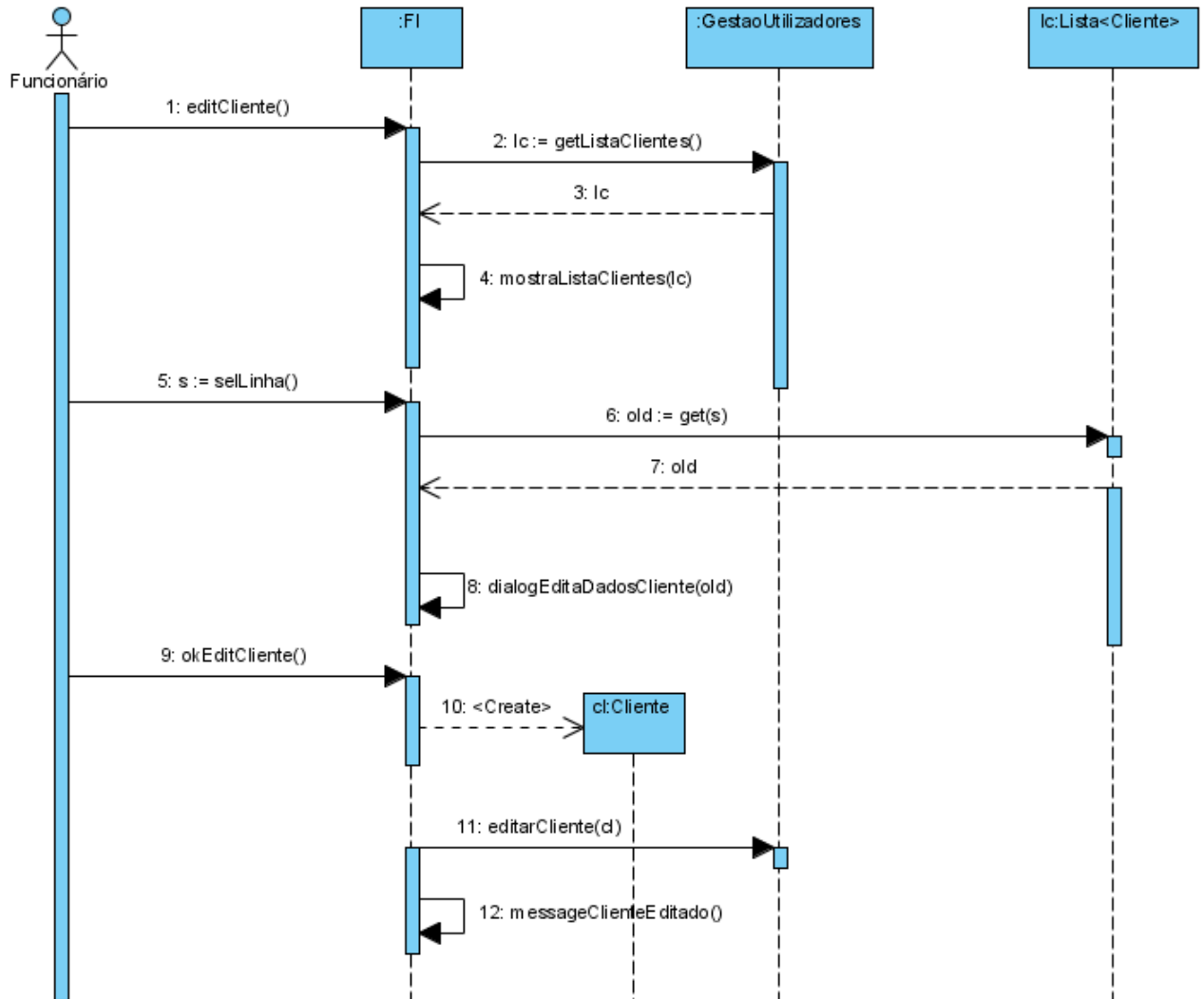
8.2.1 Adicionar Cliente



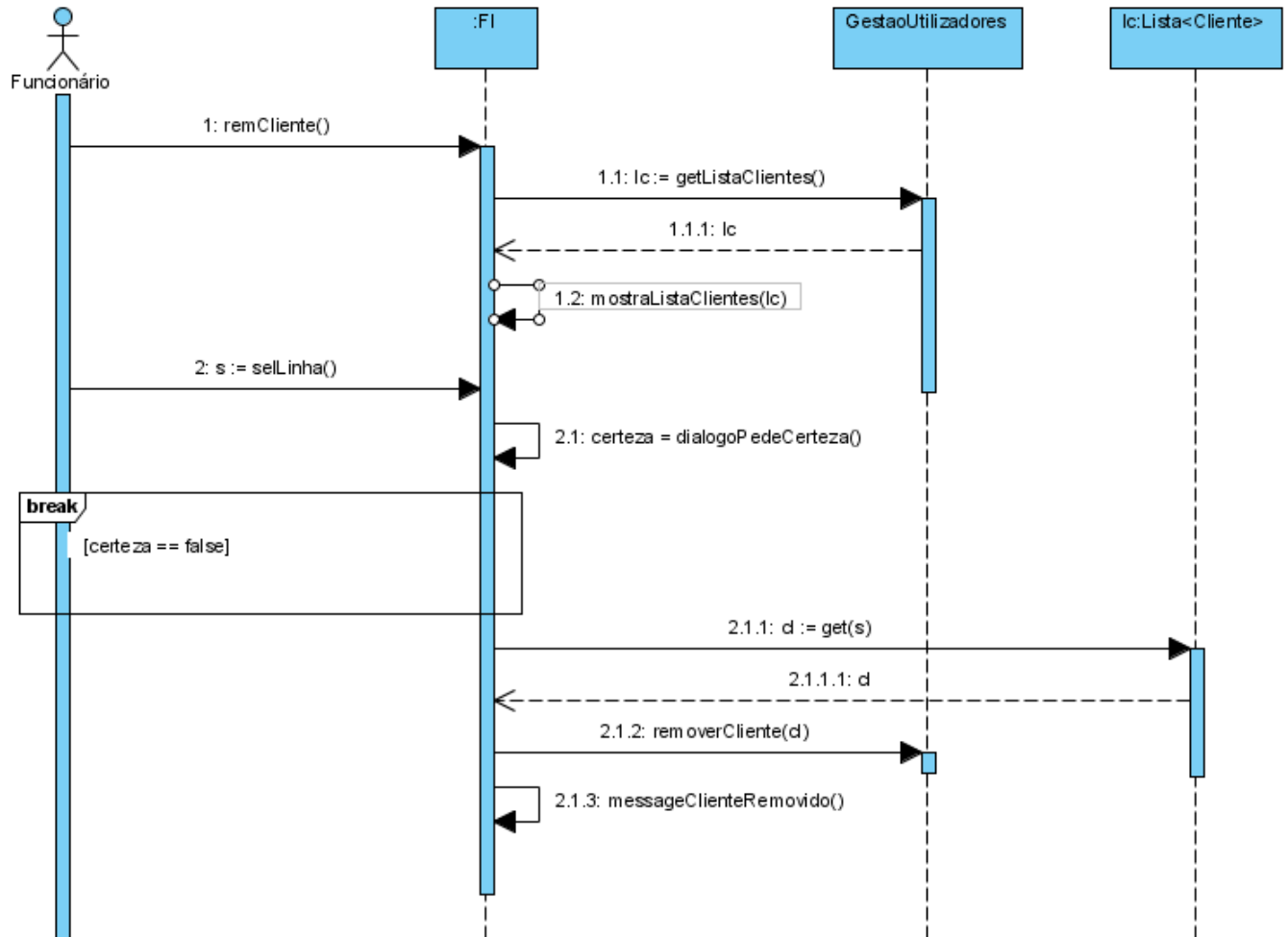
8.2.2 Consultar Cliente



8.2.3 Editar Cliente

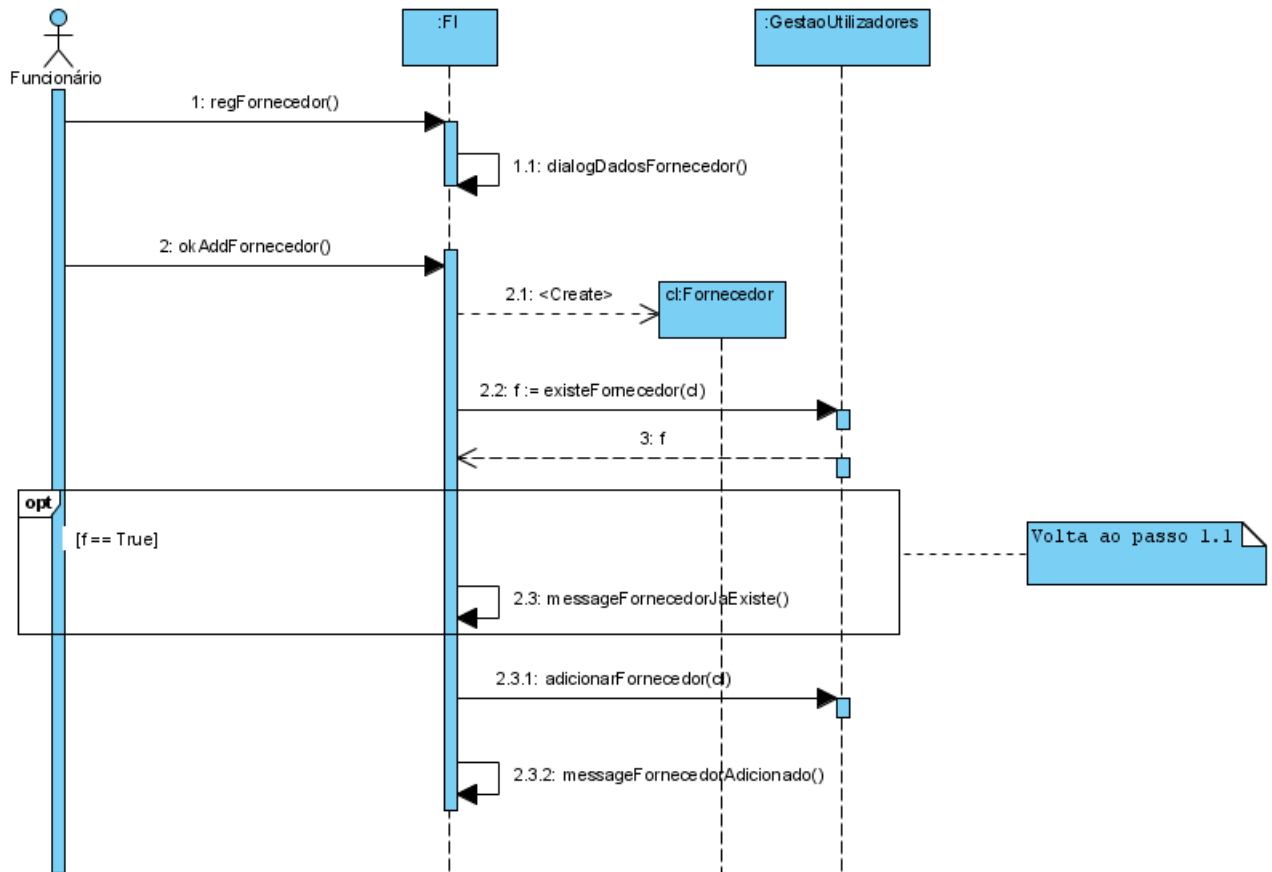


8.2.4 Remover Cliente

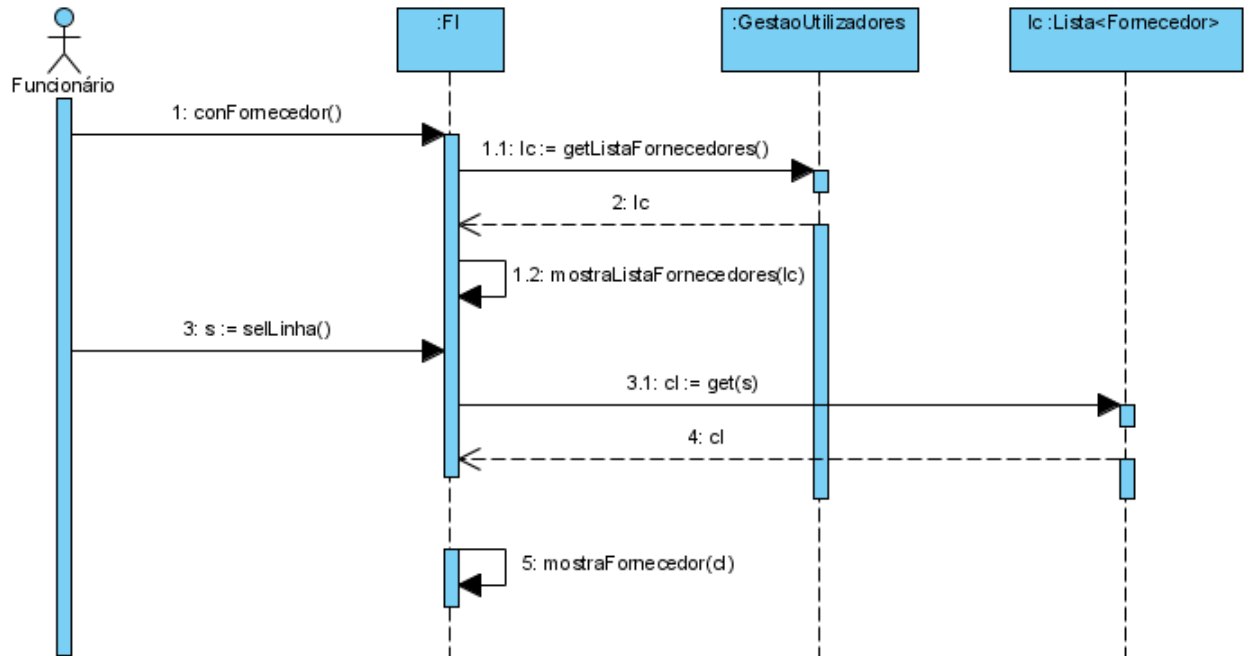


8.3 Fornecedores

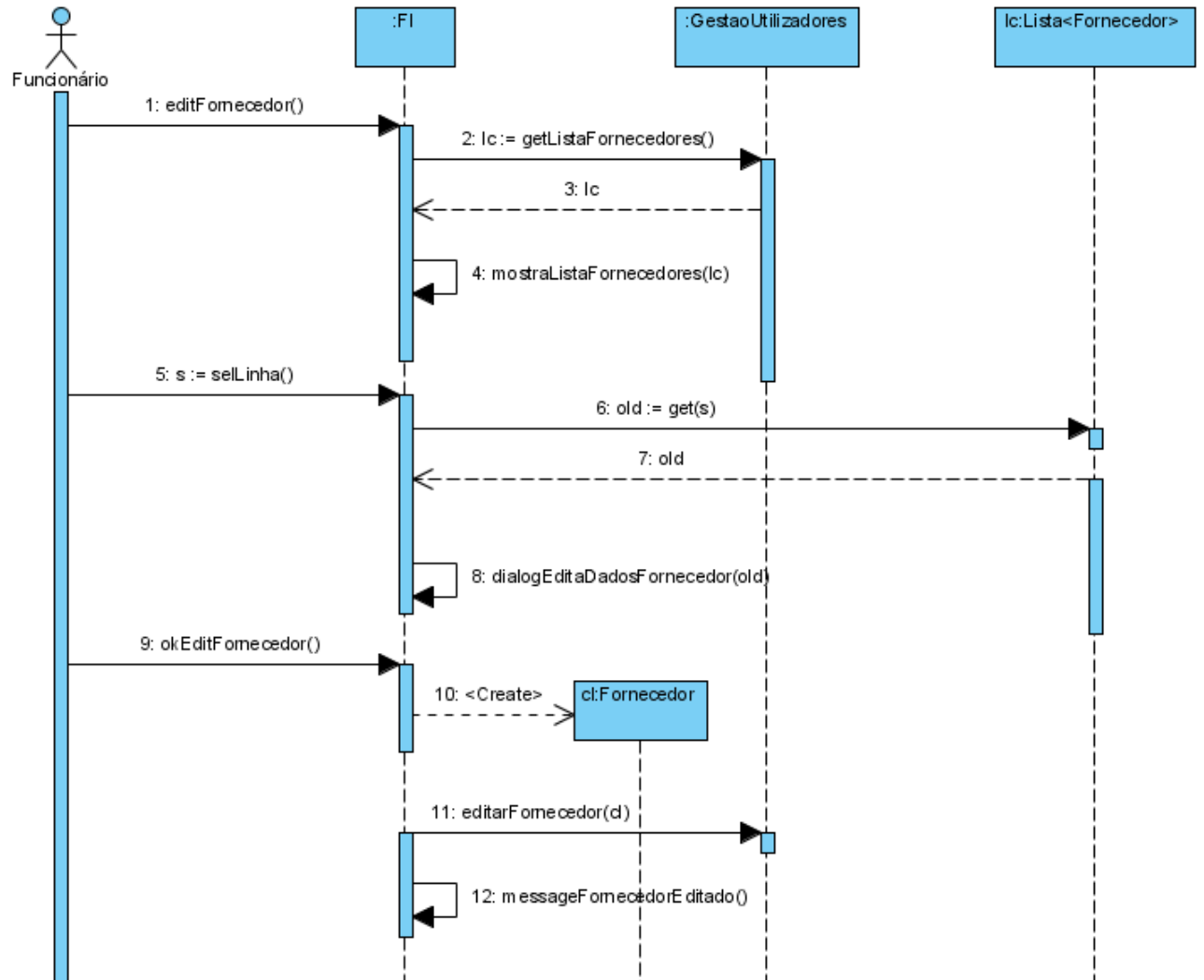
8.3.1 Adicionar Fornecedor



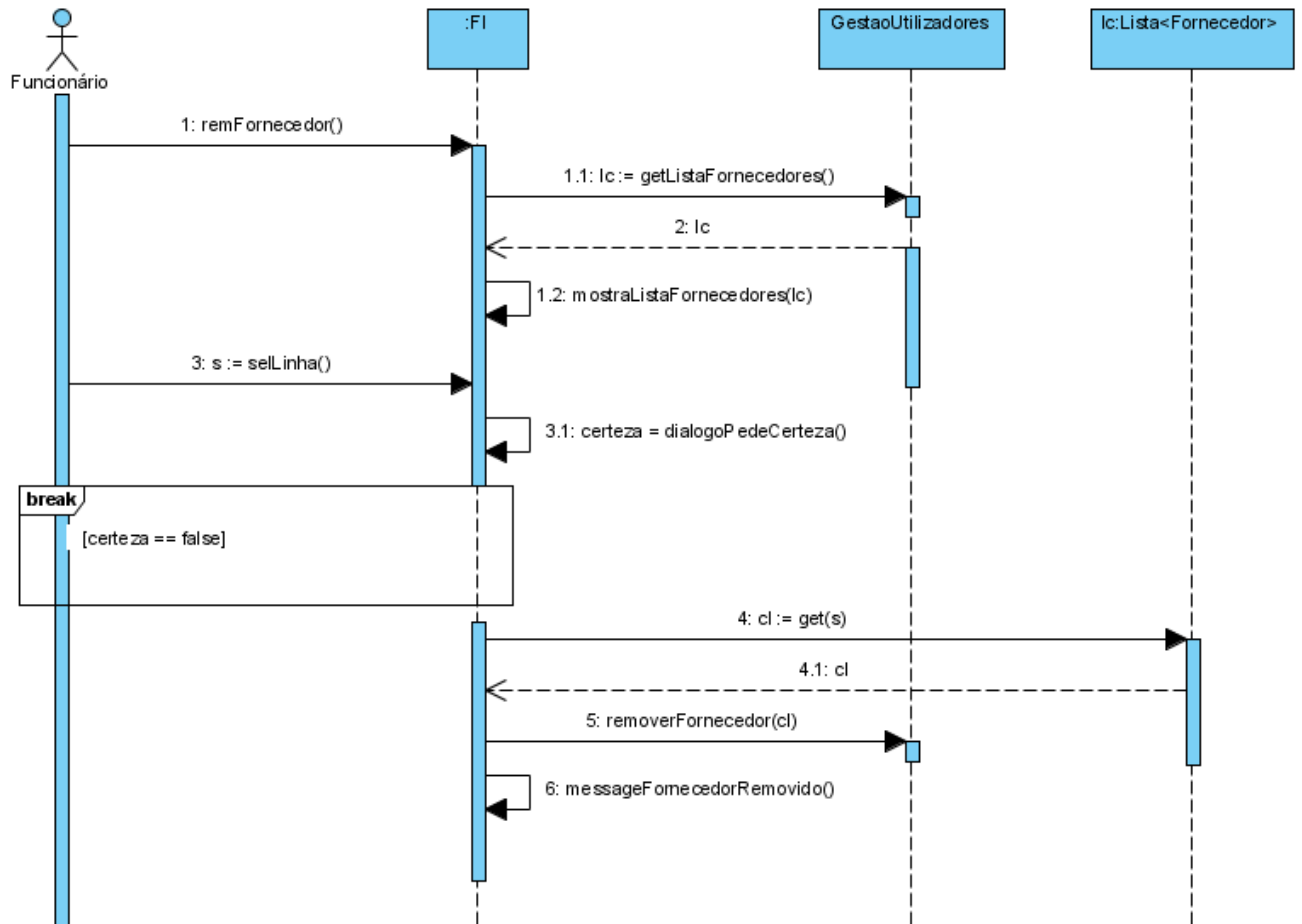
8.3.2 Consultar Fornecedor



8.3.3 Editar Fornecedor

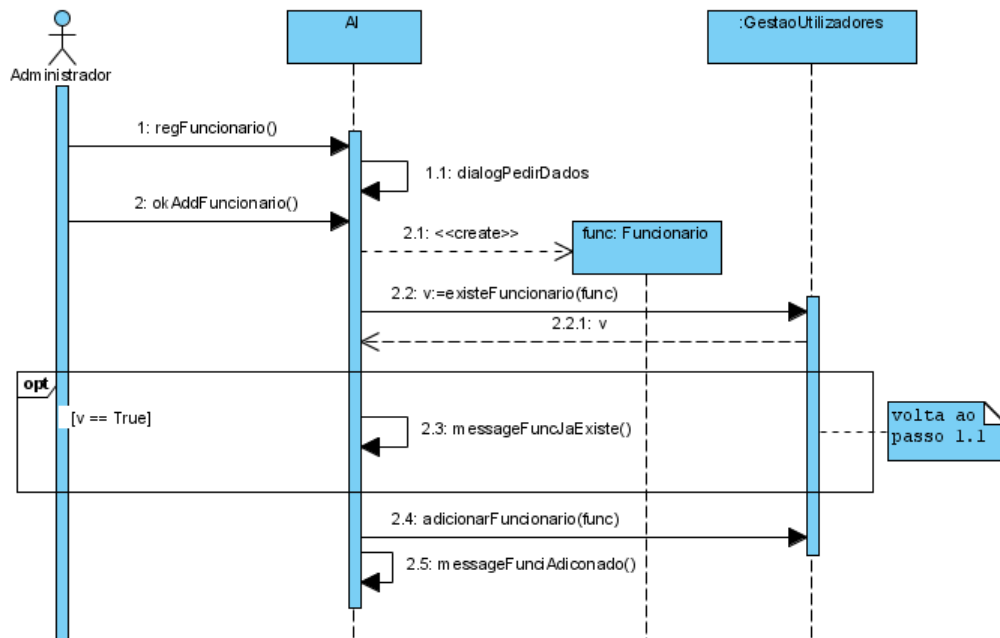


8.3.4 Remover Fornecedor

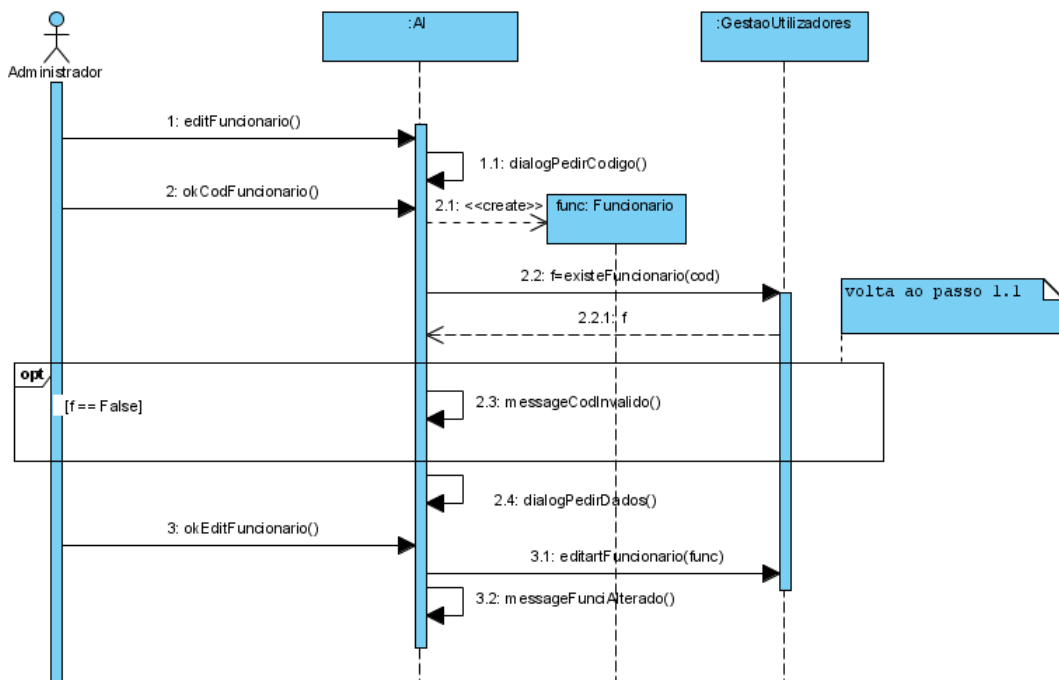


8.4 Funcionários

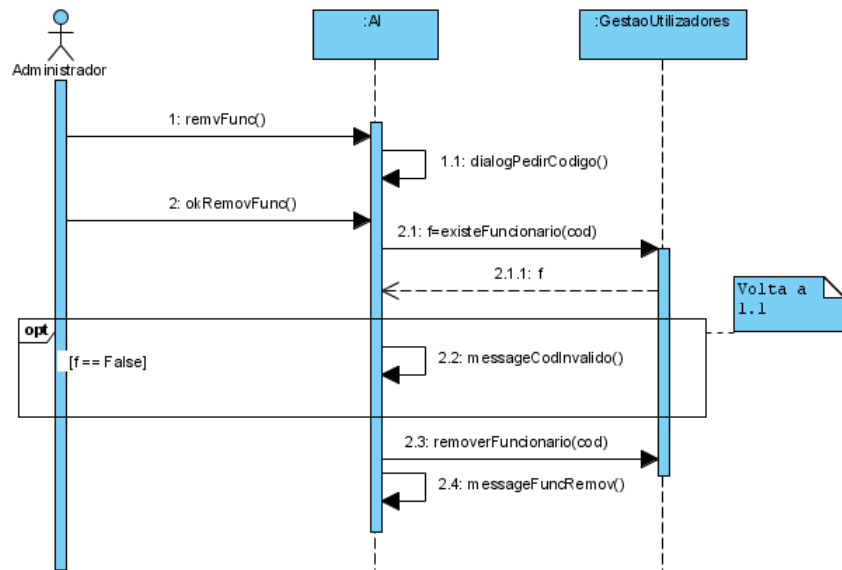
8.4.1 Adicionar Funcionário



8.4.2 Editar Funcionário

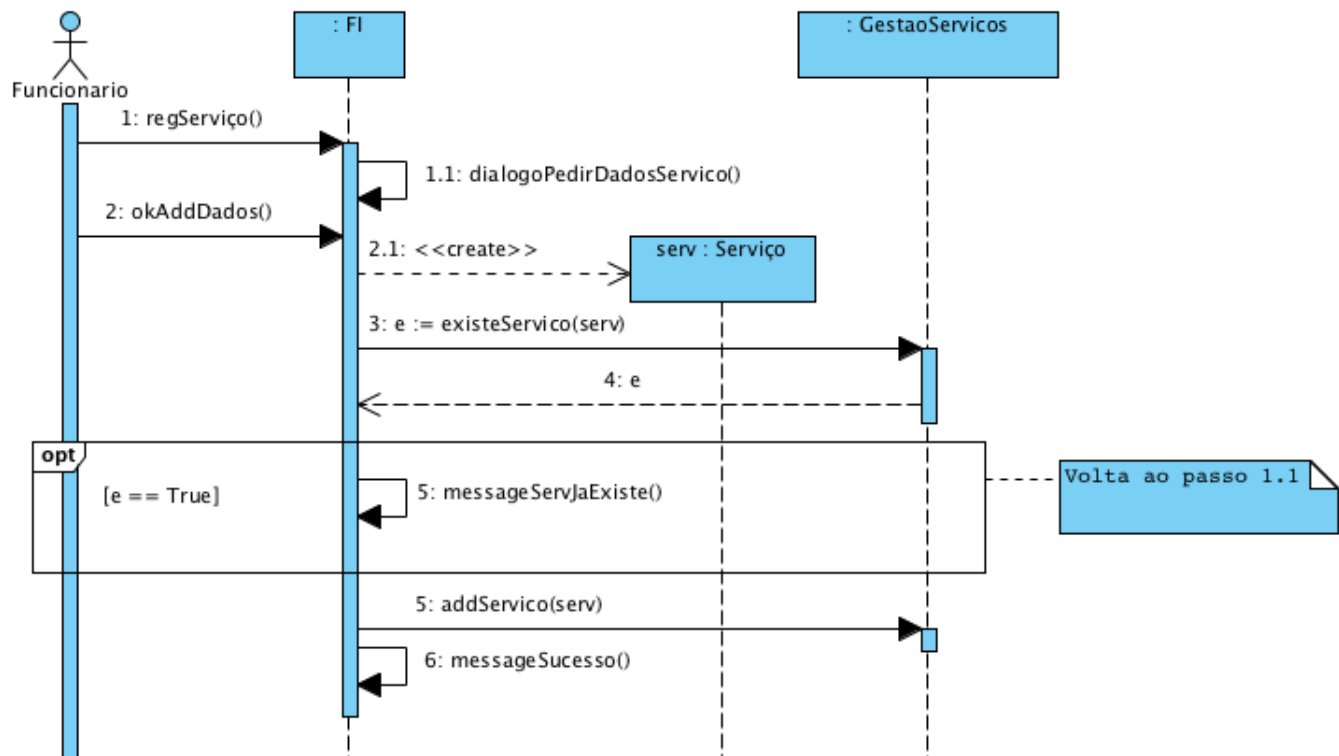


8.4.3 Remover Funcionário

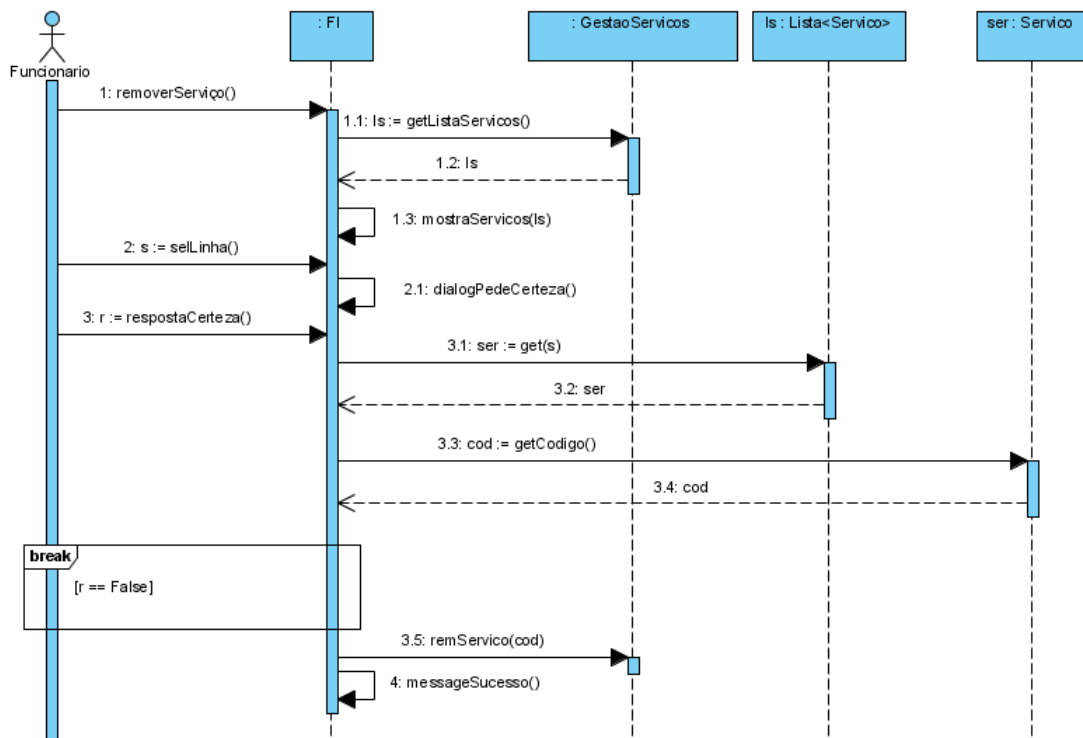


8.5 Serviços

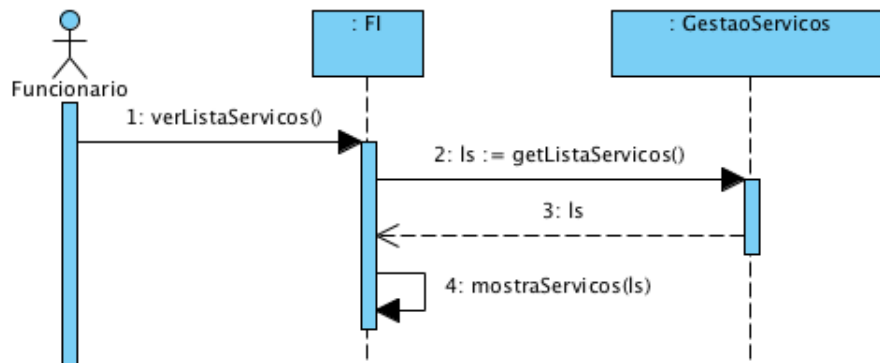
8.5.1 Acidionar Serviço



8.5.2 Remover Serviço

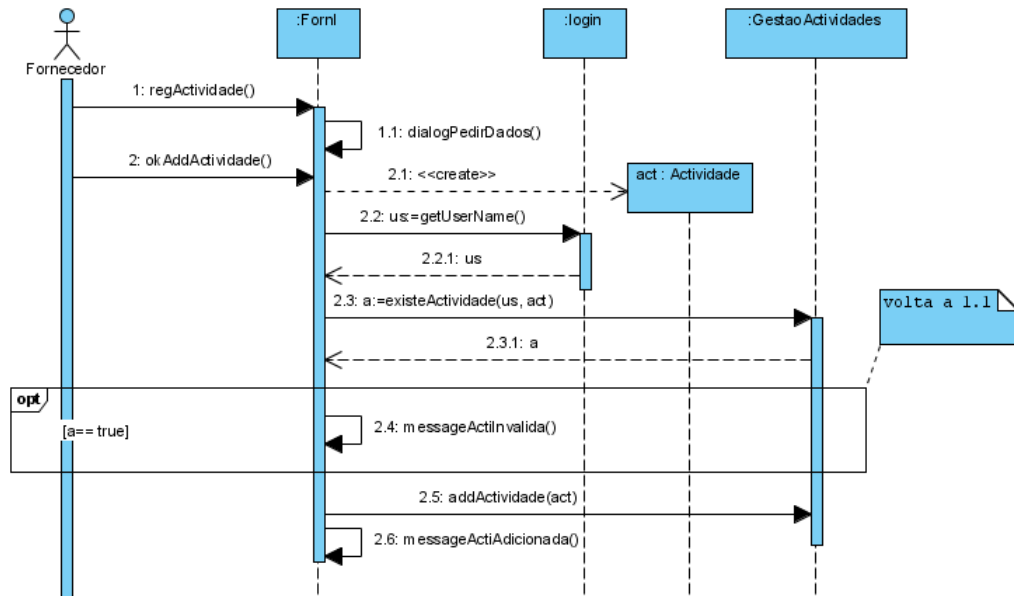


8.5.3 Consultar Lista de Serviços

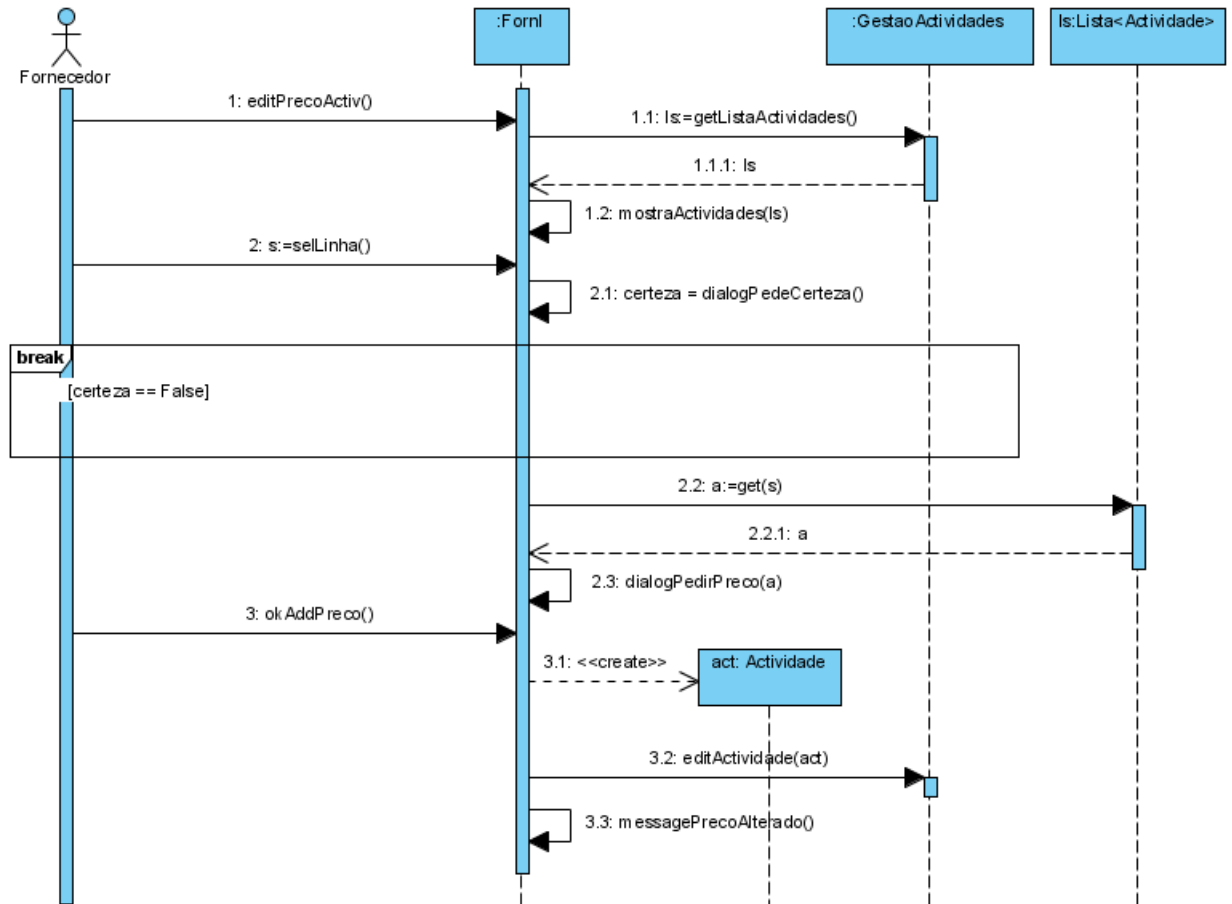


8.6 Atividades

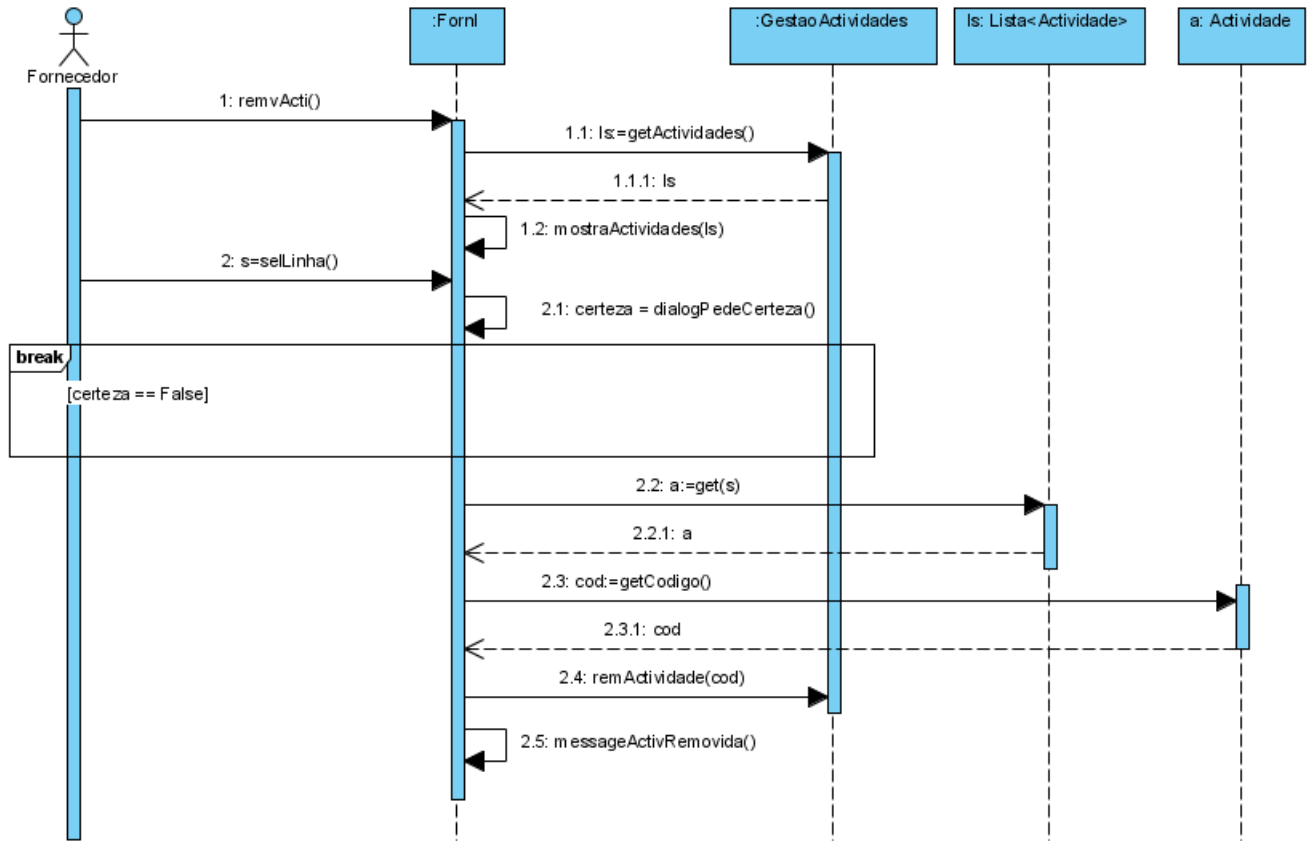
8.6.1 Adicionar Actividade



8.6.2 Alterar Preço de Actividade

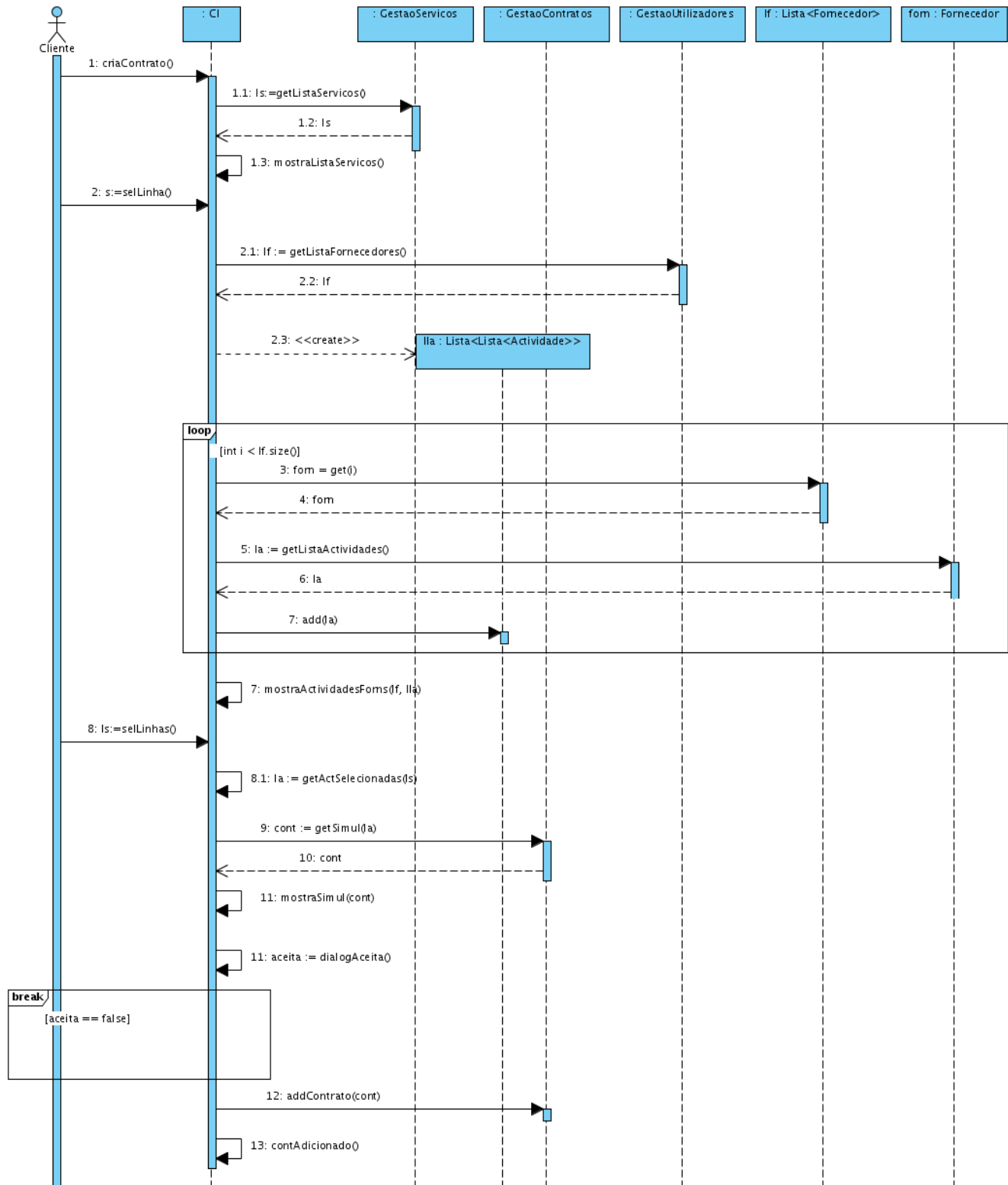


8.6.3 Remover Actividade

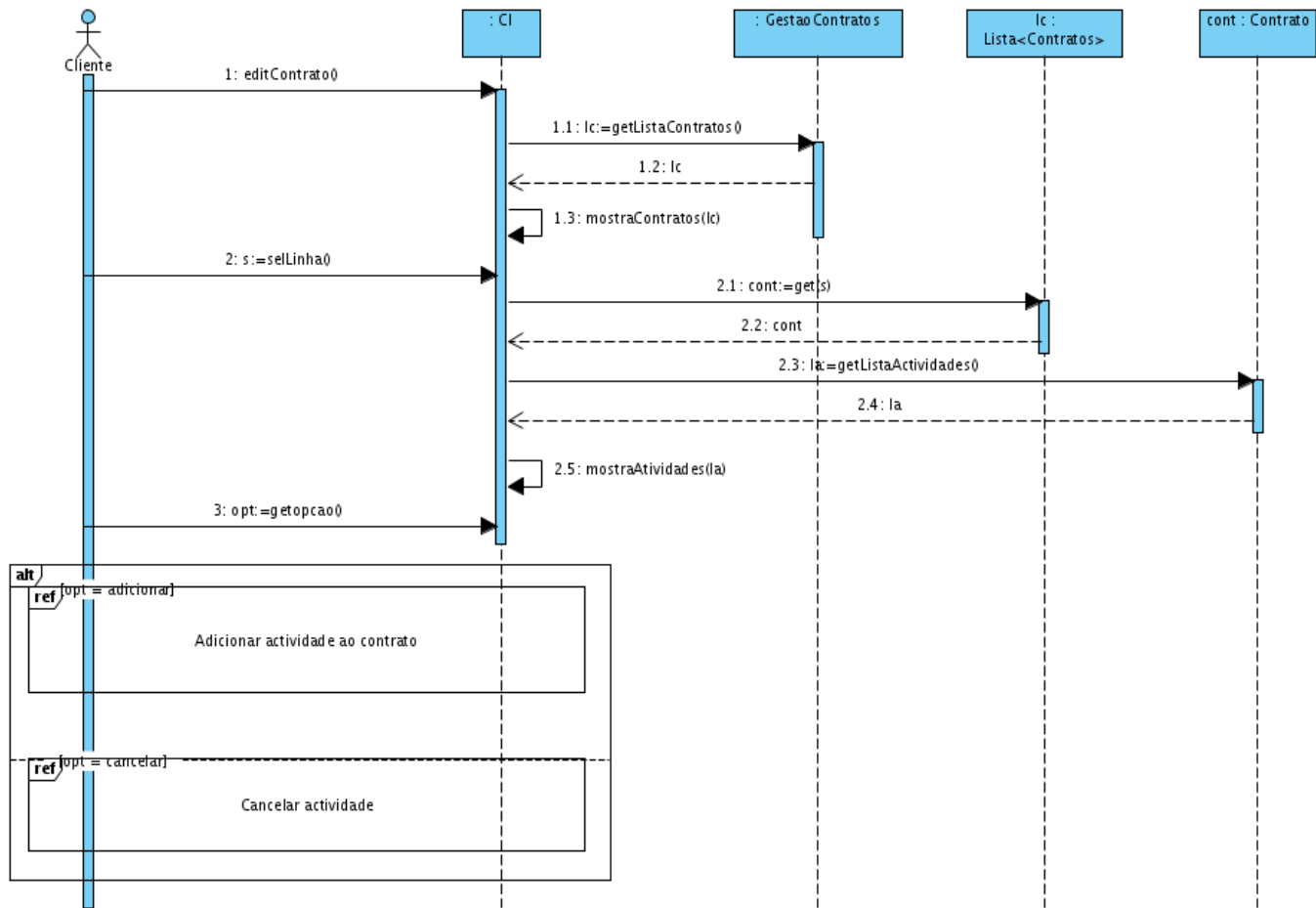


8.7 Contratos

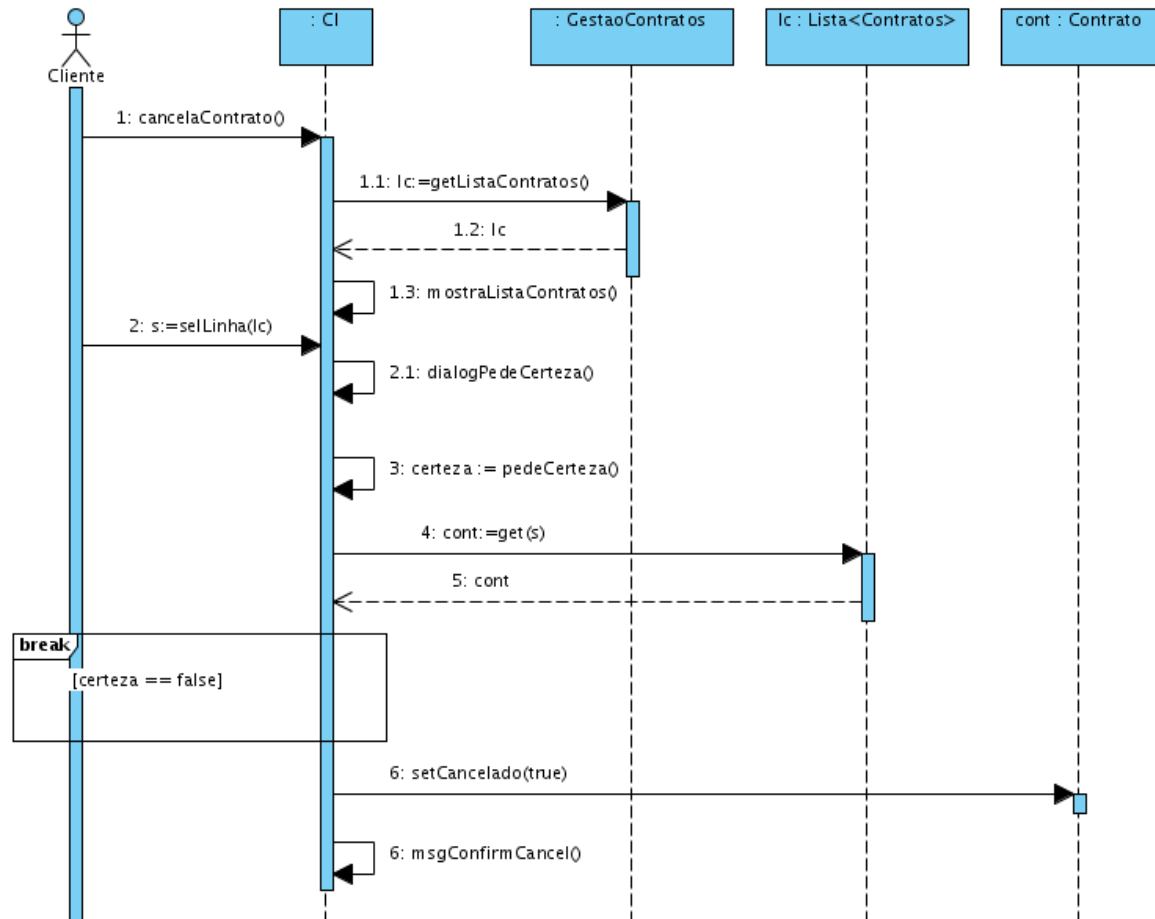
8.7.1 Criar Contrato



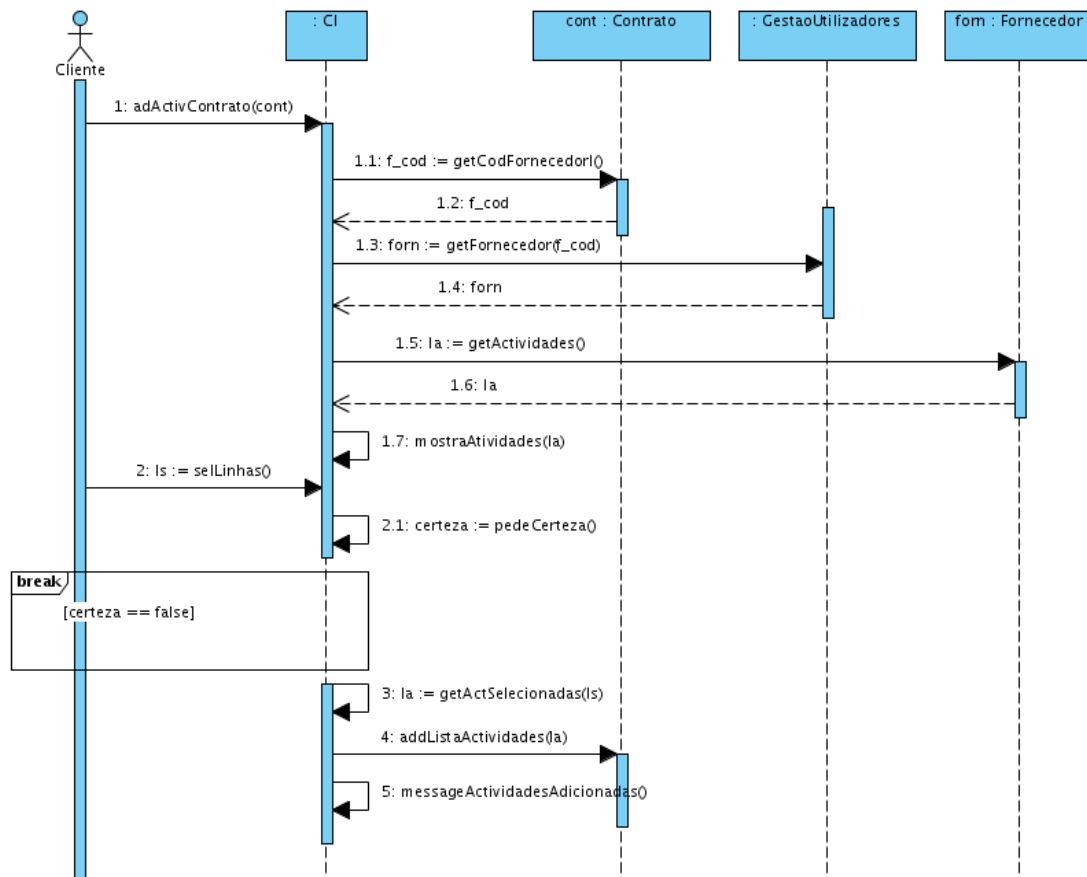
8.7.2 Alterar Contrato



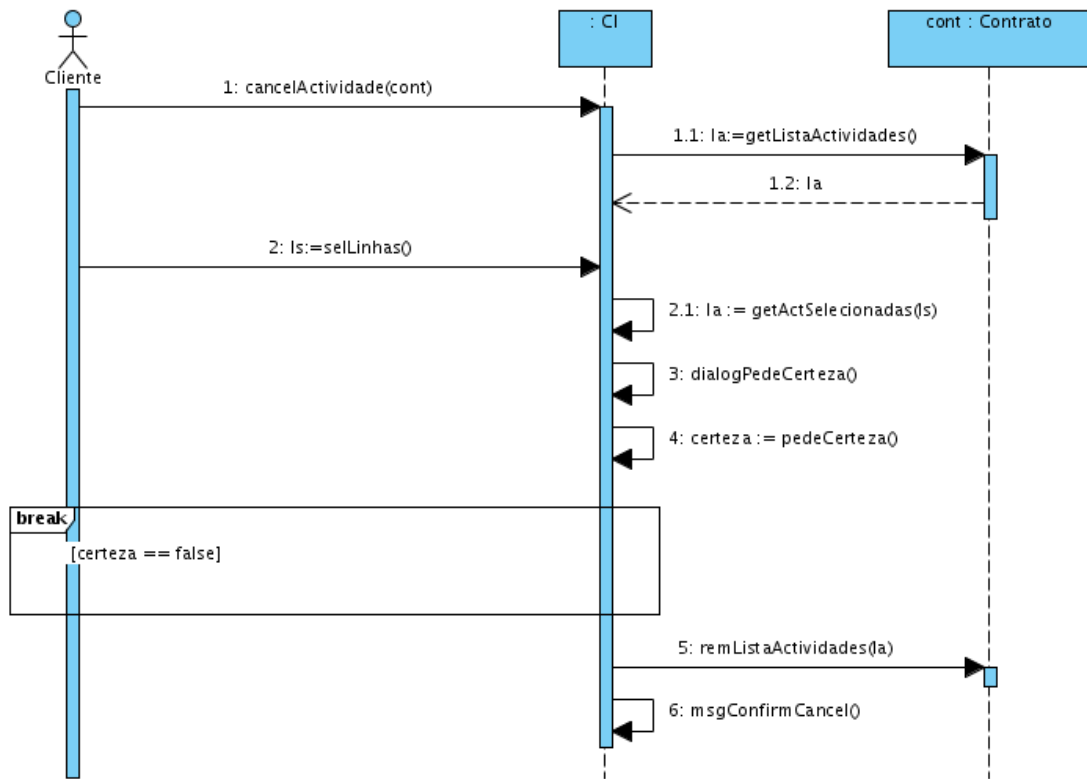
8.7.3 Cancelar Contrato



8.7.4 Adicionar Actividade ao Contrato

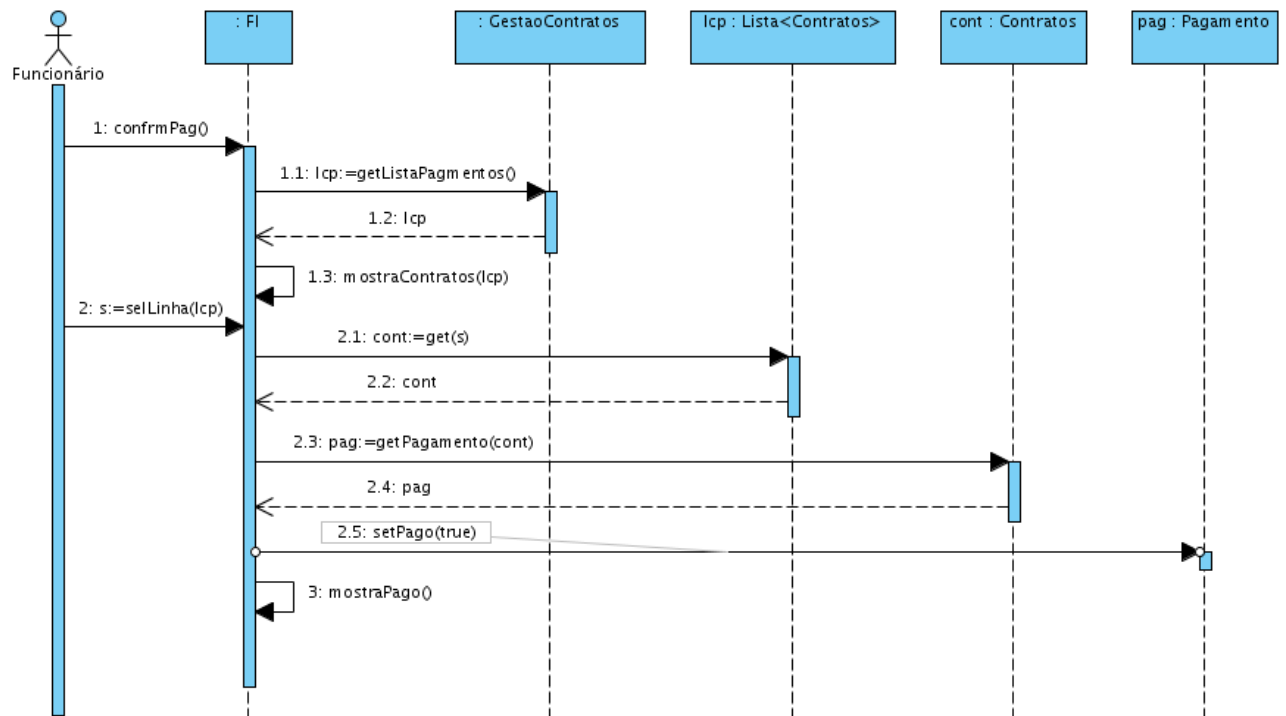


8.7.5 Cancelar Actividade do Contrato

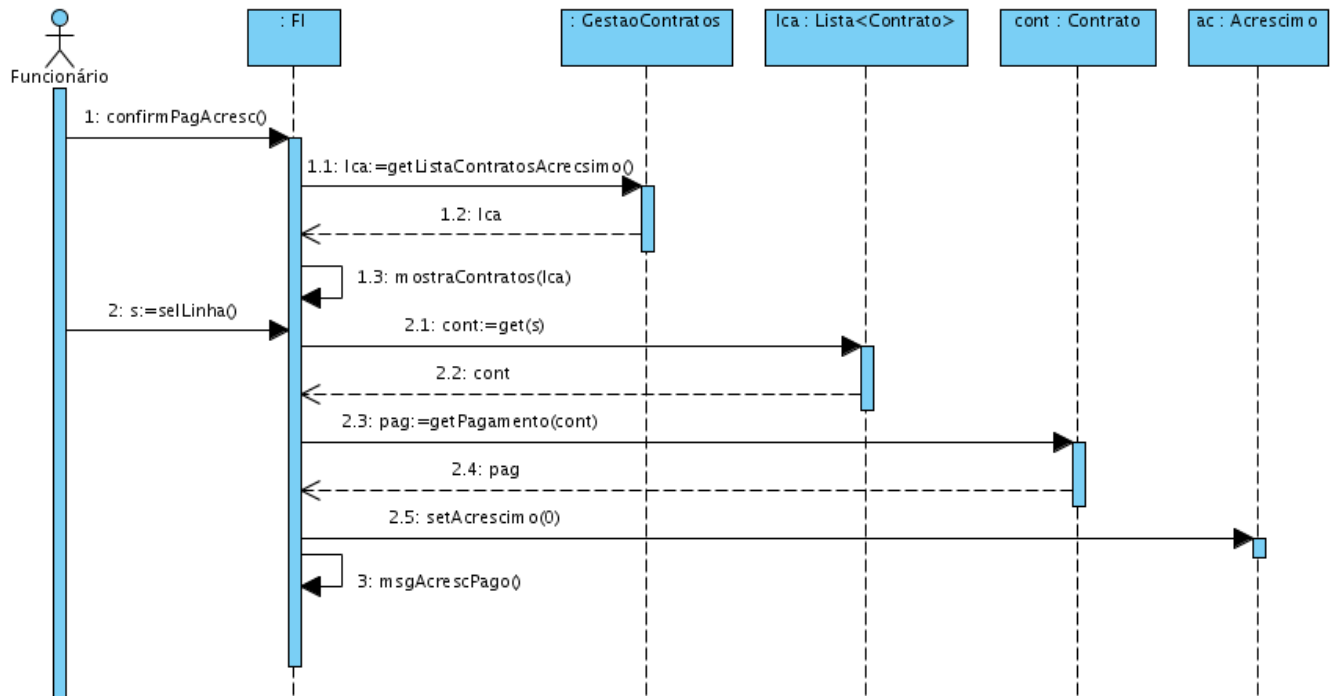


8.8 Pagamentos

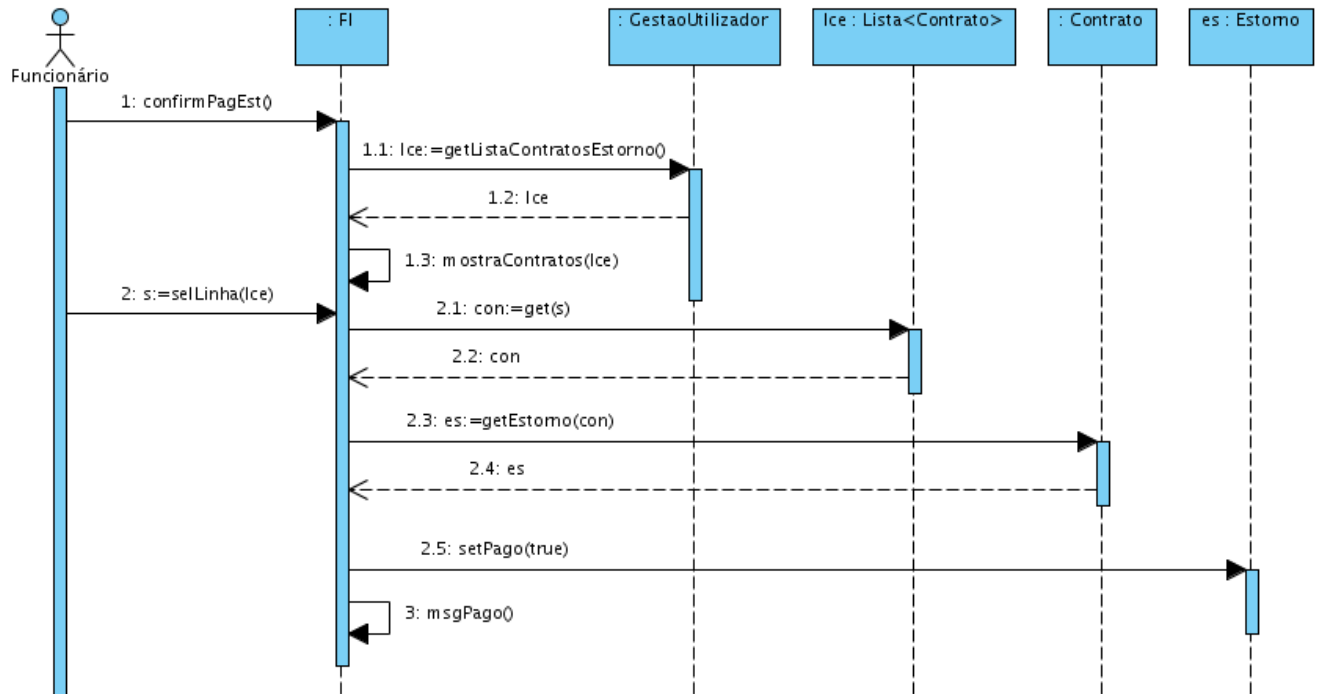
8.8.1 Confirmar Pagamento de Contrato



8.8.2 Confirmar Pagamento de Acréscimo



8.8.3 Confirmar Pagamento de Estorno



9 Decisões Pertinentes

9.1 Pagamentos

Consideramos que cada vez que um cliente efectua um pagamento relativo a qualquer serviço prestado, é da responsabilidade do funcionário da Gere com Saber confirmar no sistema o pagamento foi efectuado com sucesso. O mesmo se aplica para os pagamentos de estornos pela parte dos fornecedores.

9.2 Contratos

Cada cliente pode ter vários contratos, sendo que cada contrato corresponde a uma relação entre um cliente e um fornecedor, ou seja, um cliente tem apenas um único contrato com cada fornecedor. Os contratos podem ser alterados sempre que consoante a vontade do cliente em adquirir novos serviços ou desistir de serviços, sendo que os valores relativos a pagamentos serão repostos no final do contrato.

Todos os contratos têm a duração fixa de um ano, podendo ser cancelados a qualquer altura.

10 Conclusão

Após a construção da primeira parte do trabalho podemos concluir que o desenho do Modelo de Domínio, assim como o Diagrama de Use-Cases, foi importante para melhor compreendermos o problema em questão assim como para o diálogo com o cliente, onde com a construção destes diagramas foi possível ele analisar se estávamos a evoluir para uma solução que se encaixava nos requisitos que ele apresentou.

O diagrama de use cases também permitiu que o cliente perceba-se qual seria a interação dos utilizadores com o software que será posteriormente desenvolvido, e é um passo em frente para a construção de modelos de nível mais baixo, mais próximos da implementação final.

A construção iterativa dos diagramas de sequência permitiu-nos dividir facilmente o problema em partes mais pequenas levando-nos à construção de *Sub-Sistemas* e *Packages*, conferindo uma maior modularidade ao código final deste projecto.

A construção do *Diagrama de Classe de Sub-Sistema* ajuda-nos a identificar todos os atributos e métodos necessários na implementação de cada classe.

Estes modelos criados nesta fase do projecto irão ser certamente muito úteis na fase de implementação, onde estaremos a implementar um programa devidamente desenhado e aprovado pelo cliente.