

Desenvolvimento de Sistemas de Software

2ª Fase do Projecto

Grupo G5

Abel Matos Nº: 51776



João Amorim Nº: 51771



João Guedes Nº: 51755



Luis Oliveira Nº: 51801



Pedro Reis Nº: 51829



Índice

Introdução.....	4
Diagrama de Use Cases	5
Diagrama de Classes.....	7
Diagramas de Sequência de Sistema.....	10
Registrar Cliente	11
Iniciar Contrato	12
Adicionar Actividade a Contrato	14
Remover Actividade de Contrato.....	16
Calcular Lucro.....	18
Calcular Fluxos.....	19
Adicionar Actividade	20
Remover Actividade	22
Adicionar Fornecedor.....	23
Adicionar Serviço.....	25
Conclusão	28

Introdução

Nesta segunda fase do projecto era pedida a continuação da modelação da aplicação que iria servir de base à gestão de condomínios por parte da empresa GereComSaber. Depois de na primeira fase ter sido desenvolvido o Modelo de Domínio e Diagrama de Use Cases, era agora necessária a criação do Diagrama de Classes e Diagramas de Sequência de Sistema.

Através de uma análise mais profunda do problema, e tendo por base a modelação previamente feita, iremos tentar ser o mais específico e claro possível, respeitando sempre os requisitos pedidos. Caso seja necessário, haverá alterações na modelação anterior de forma a tornar a modelação o mais correcta possível.

Diagrama de Use Cases

Em relação ao diagrama original, foram feitas algumas alterações. Estas alterações basearam-se na remoção de alguns <<include>> e <<extends>>, que passaram a ser incorporados no use-case “pai”. Procedemos a esta alteração, pois as operações às quais os use cases que foram incorporados diziam respeito eram muito simples, e não era necessário haver um use case específico para essa operação.

Tudo o resto mantém-se igual. O actor Cliente, que pode representar um cliente ou um operador a agir a mando dum cliente, tem acesso às operações Registar Cliente, Iniciar Contrato, Adicionar Actividade a Contrato e Remover Actividade de Contrato. O actor Gestor, que representa o gestor do sistema, tem acesso às operações Calcular Lucro, Calcular Fluxos, Adicionar Actividade, Remover Actividade, Adicionar Fornecedor e Remover Fornecedor. Optámos por não oferecer directamente as opções Adicionar e Remover Serviço, pois consideramos que seria mais adequado adicionar um Serviço quando é adicionado um fornecedor que efectue esse serviço, e remover um serviço quando forem removidos todos os fornecedores que fornecem esse serviço.

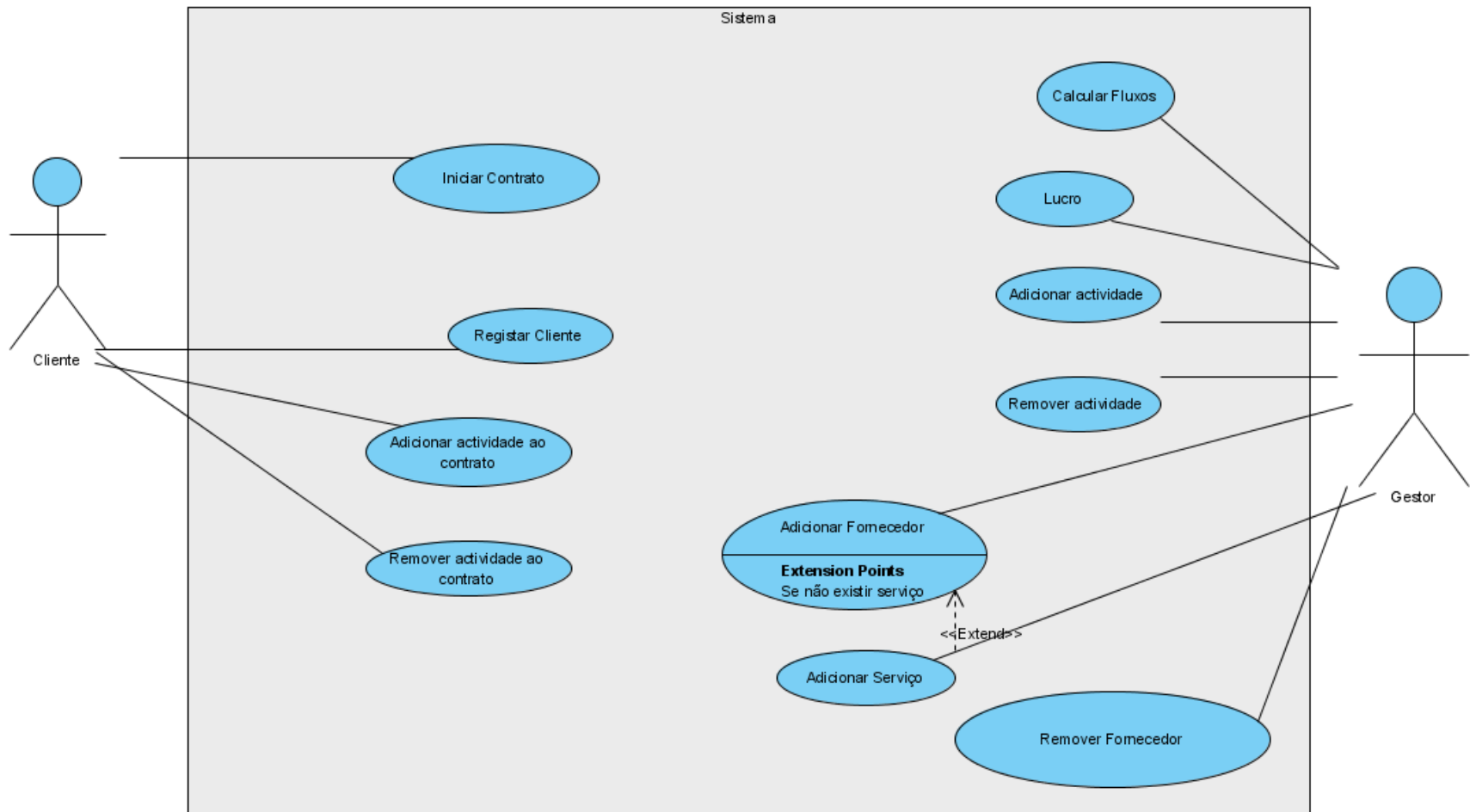
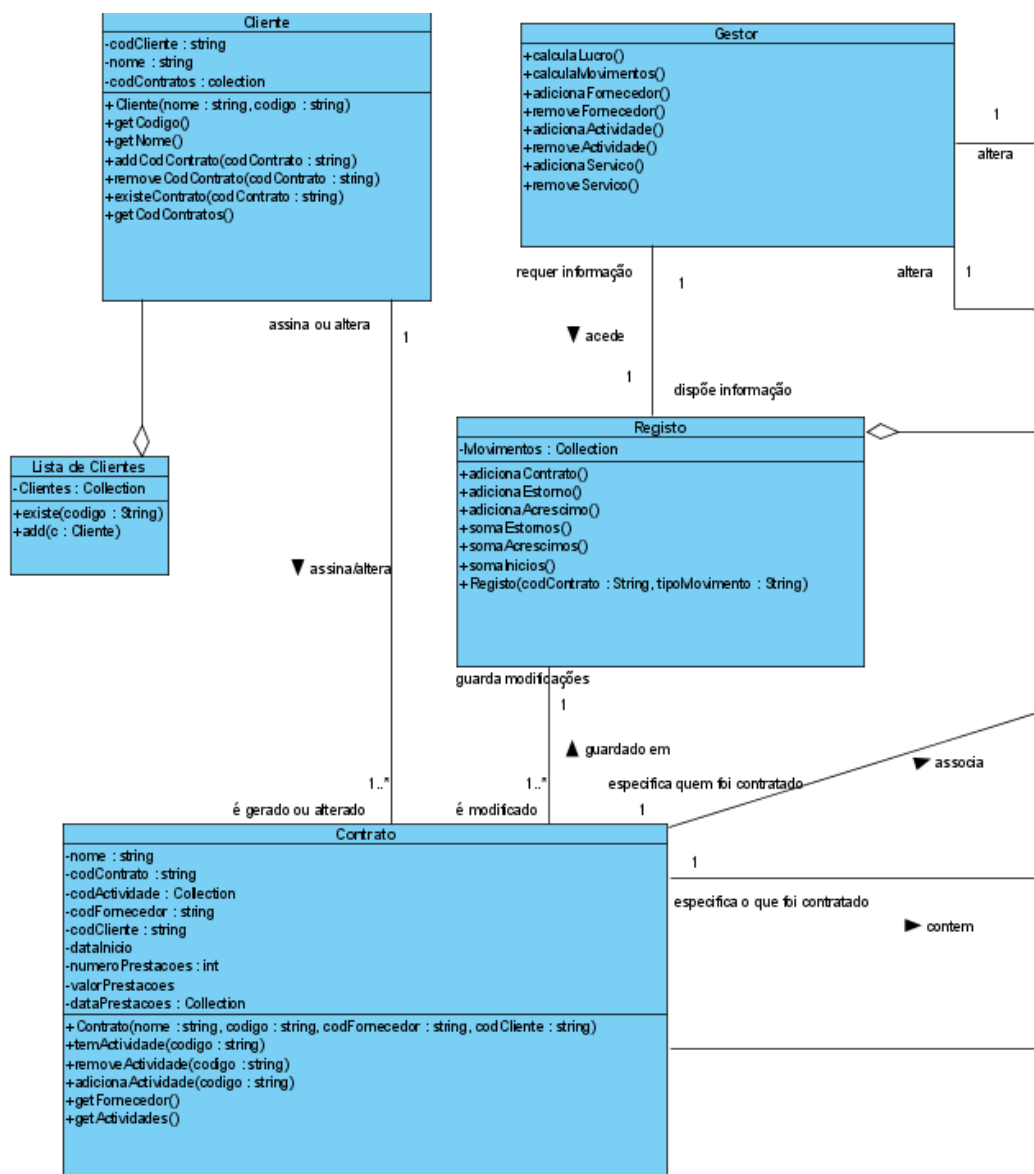
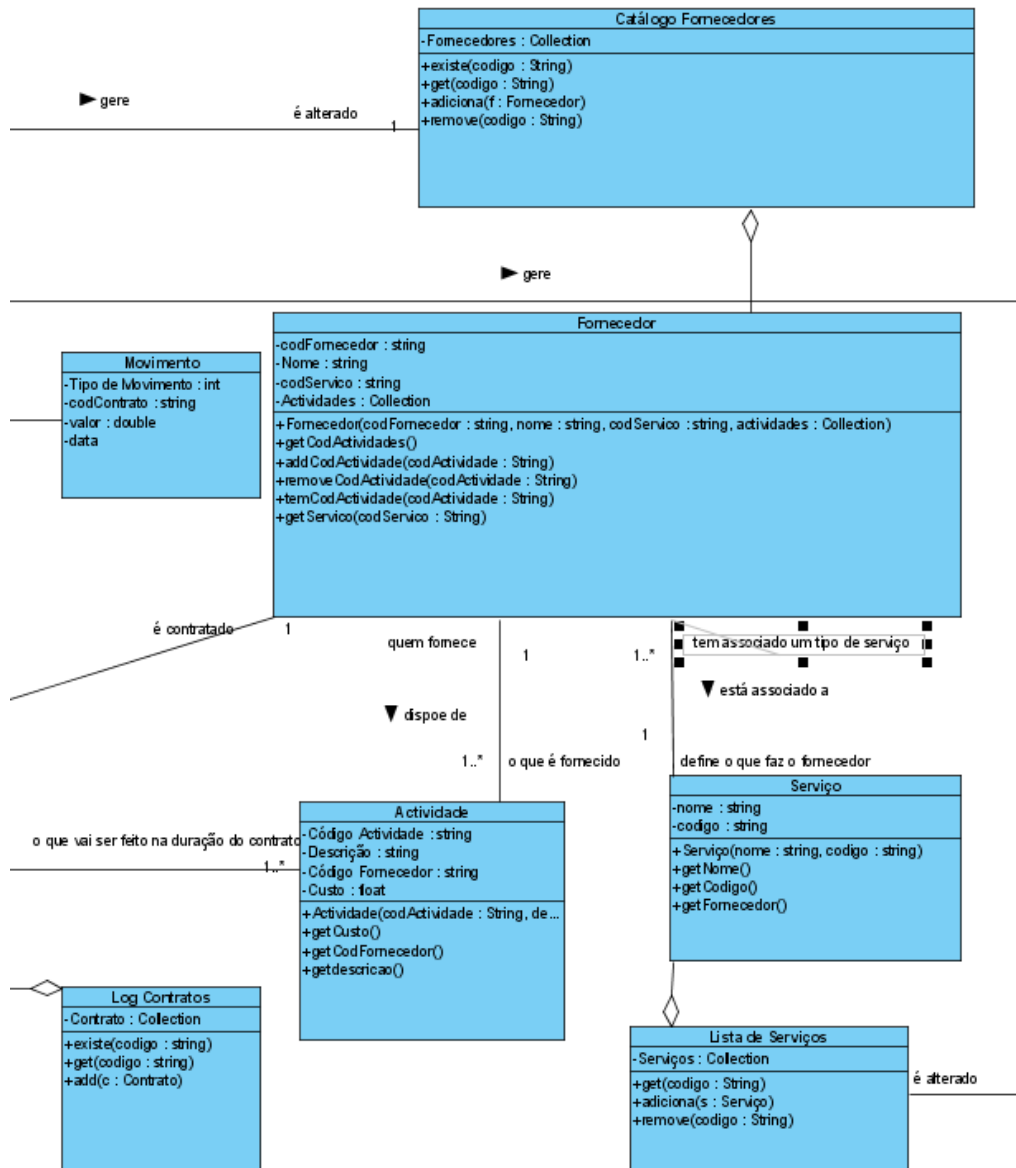


Diagrama de Classes

O seguinte diagrama de Classes é baseado no diagrama de Use Cases e no diagrama de modelo de domínio definido anteriormente. Este espelha as interações entre as Classes do software que a empresa GereComSaber pretende adquirir para facilitar a sua gestão de condomínios.





A figura anterior mostra o diagrama de classes da aplicação. Cada caixa do diagrama representa uma Classe e esta dividida em 3 secções: nome da Classe, atributos da classe e métodos da Classe.

A Classe Cliente representa um cliente que utiliza a empresa para contratar serviços. Cada instância do Cliente é guardada na Lista de Clientes.

A Classe Gestor representa o gestor do sistema. A suas operações são aquelas que permitem a manutenção do sistema.

A Classe Contrato representa um contrato que é assinado pelo Cliente com um Fornecedor. Em cada contrato são especificados o Fornecedor, o Cliente, e as Actividades que vão ser fornecidas, bem como as datas, e valor das prestações a pagar, e a data de Inicio do Contrato. Os contratos são guardados no Log de Contratos.

A Classe Registo engloba todos os inícios e alterações de Contrato feitos através do sistema. Cada início/alteração é uma instância da Classe Movimento. O Registo guarda todos os Movimentos.

A Classe Fornecedor representa um fornecedor que está associado à empresa. Cada fornecedor dispõe de uma lista de Actividades que presta, e está associado a um tipo de Serviço. Os fornecedores são guardados no Catálogo de Fornecedores.

A Classe Actividade representa uma Actividade prestada por um Fornecedor, como por exemplo, "Cortar a relva". Cada Actividade tem associado o Fornecedor que a presta e o seu custo.

A Classe Serviço representa os vários tipos de Serviços prestados pelos Fornecedores associados à GereComSaber, como por exemplo, "Jardinagem". A cada Serviço estão associados os vários fornecedores que dispõem desse tipo de serviço. Os Serviços são guardados na Lista de Serviços.

Todas as classes estão ligadas a uma outra classe por uma linha que representa o tipo de interacção entre elas. Por exemplo entre a Classe Contrato e a Classe Fornecedor existe uma associação (1 ... 1), que representa uma ligação de "um para um", ou seja, a cada Contrato apenas esta ligado um Fornecedor. Existem também ligações de 1 para 1 ou mais (1 ... 1..*), representada, por exemplo, entre a Classe Cliente e a Classe Contrato. Isto diz-nos que um Cliente pode assinar ou alterar 1 ou mais contratos. Nesta situação não faz sentido ter uma ligação 1 ... * pois este tipo de ligação diz-nos que a cada Cliente pode assinar ou alterar zero ou mais contratos. Esta ligação não teria lógica, pois se o Cliente não assinar nenhum contrato, não faz sentido guarda-lo na base de dados, e se não fizer nenhuma alteração em nenhum contrato não faz sentido relatar essa operação.

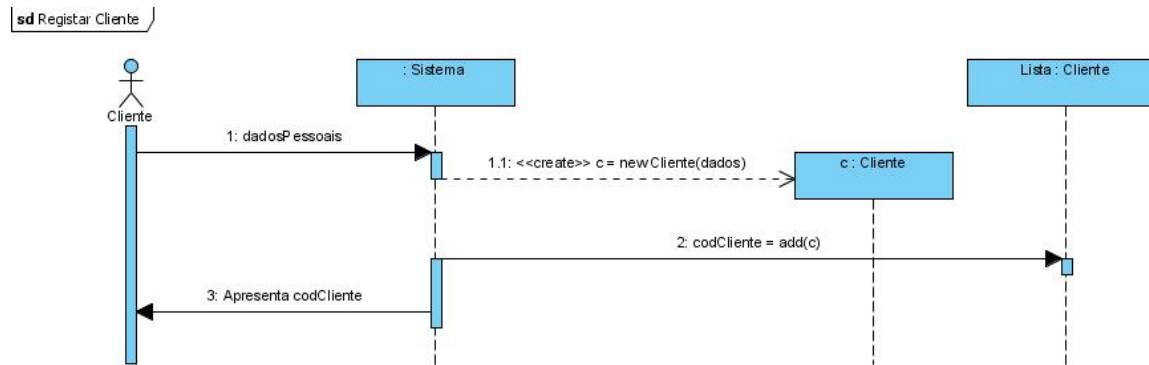
Existem neste modelo também outro tipo de ligação , que diz respeito a uma agregação, ou seja, diz que uma Classe é um conjunto de instâncias de uma outra Classe. Por exemplo a ligação entre Cliente e Lista de Clientes é uma agregação, pois a Lista de Cliente é formada por uma conjunto de Clientes. O mesmo acontece com as Classes Registo e Movimento, pois o Movimento é um registo individual das operações realizadas na aplicação, e o Registo é um "agregado" desses registos.

Diagramas de Sequência de Sistema

Neste capítulo serão apresentados os diagramas de sequência de sistema, juntamente com a descrição textual do Use Case ao qual está associado o diagrama. Existe um diagrama para cada um dos Use Cases, e em cada um deles é mostrada a sequência temporal de eventos que acontecem nas várias classes do programa quando o Use Case é executado.

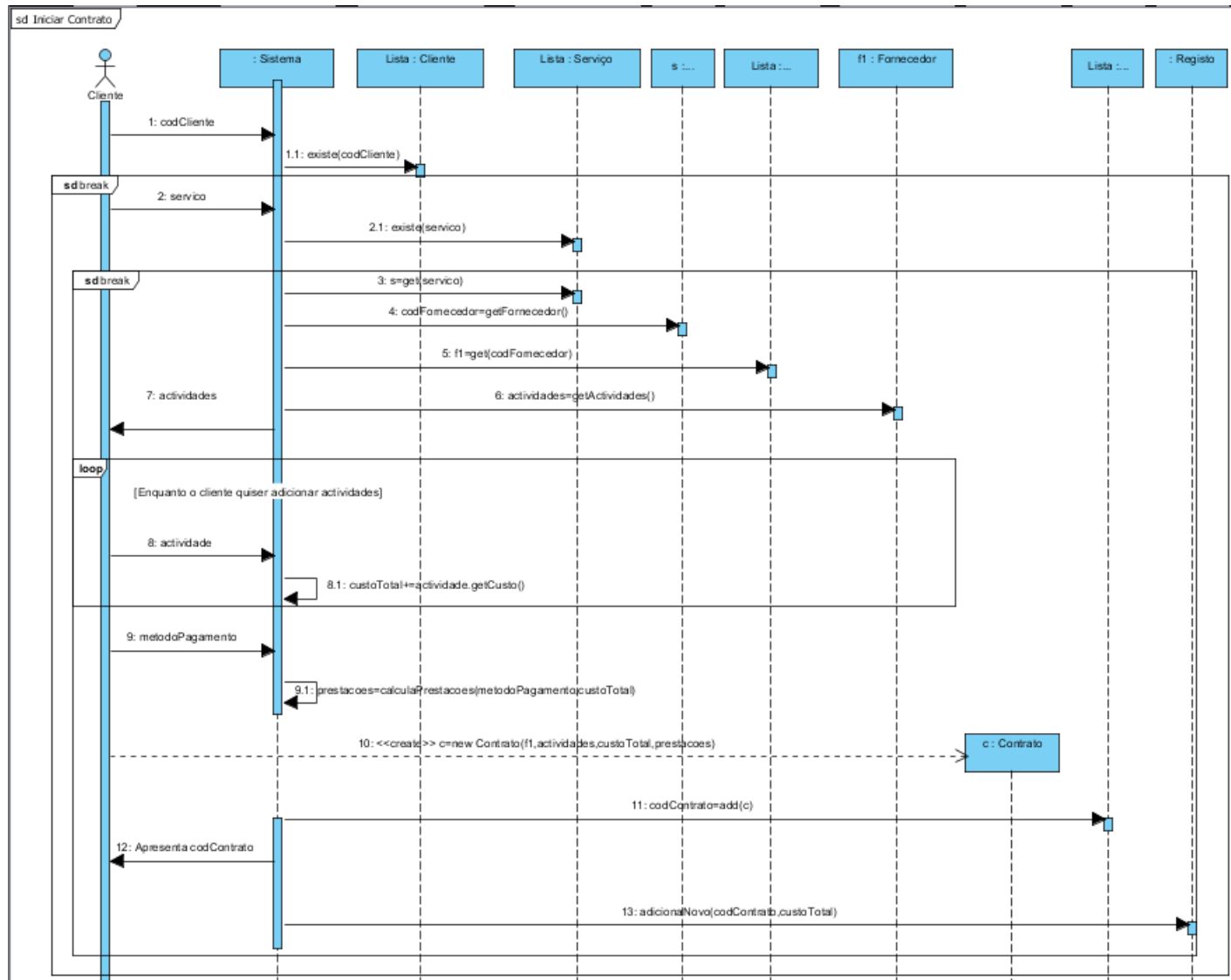
Algumas das descrições textuais dos Use Cases foram alteradas em relação à primeira fase, de forma a serem mais adequados ao contexto do problema e da aplicação.

Registrar Cliente



Super Use Case		
Author	Ze Camafeu	
Date	1/Dez/2009 18:05:06	
Brief Description		
Actor	Cliente	
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	Cliente insere dados pessoais
	2	
	3	
	4	
		System Response
		Sistema gera um Código de Cliente
		Sistema apresenta Código ao Cliente
		Sistema adiciona Cliente à lista de Clientes

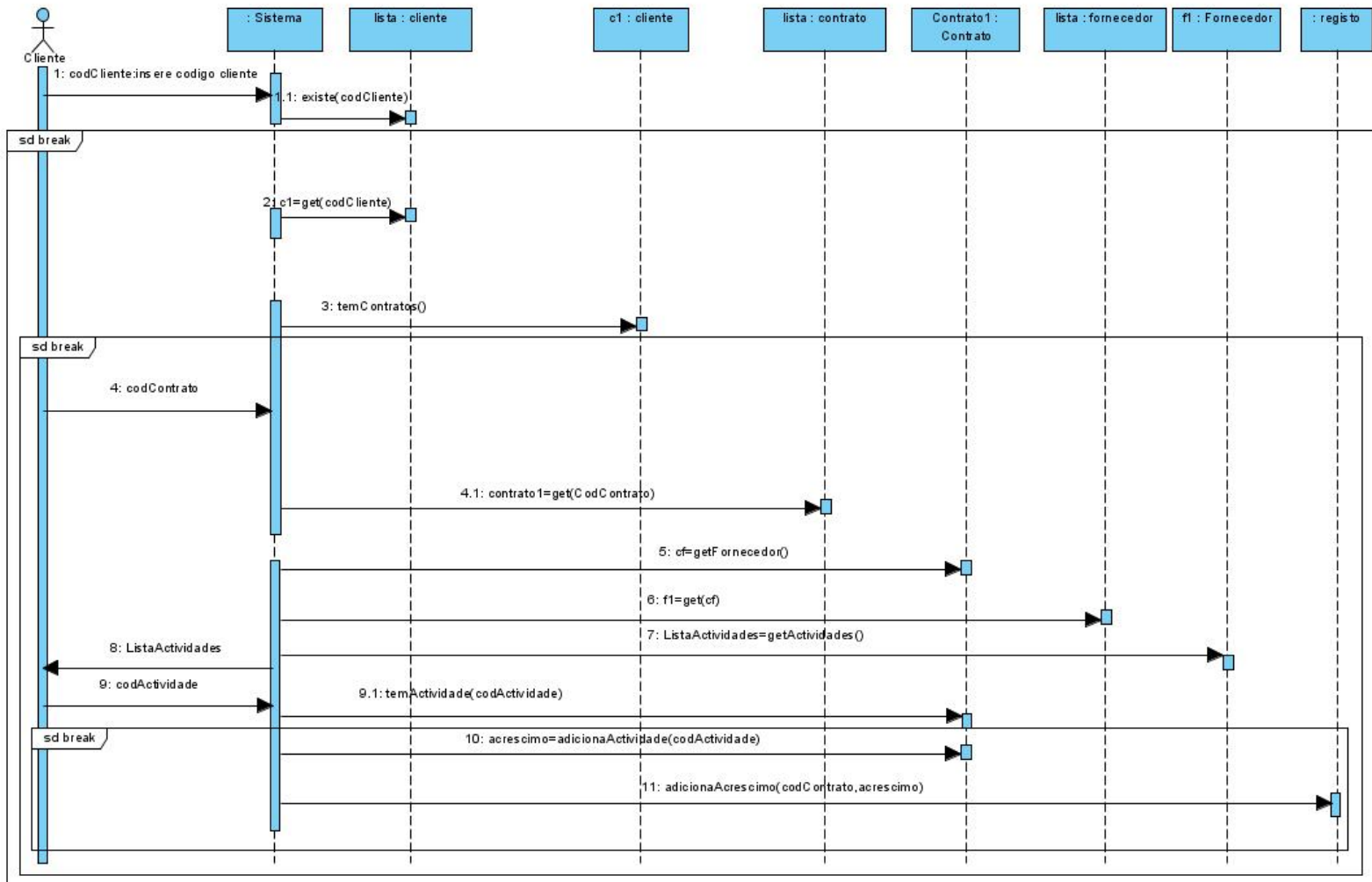
Iniciar Contrato



Super Use Case		
Author	Ze Camafeu	
Date	1/Dez/2009 17:54:15	
Actor	Cliente	
Preconditions		
Post-conditions	Sucesso: Contrato adicionado ao Registo	
Flow of Events		Actor Input
	1	Cliente insere Código de Cliente
	2	
	3	Cliente insere Serviço que pretende contratar
	4	
	5	
	6	
	7	Cliente escolhe Actividade
	8	
	9	Cliente confirma Actividades
	10	
	11	Cliente escolhe Método de Pagamento
	12	
	13	
	14	
Alternative Flow of Events: 2a - Código inválido		Actor Input
	1	
Alternative Flow of Events: 4a - Serviço não existe		Actor Input
	1	
Alternative Flow of Events: 9a - Cliente deseja adicionar mais Actividades		Actor Input
	1	
		System Response
		Sistema verifica Código
		Sistema verifica se Serviço existe
		Sistema escolhe Fornecedor
		Sistema apresenta lista de Actividade do Fornecedor
		Sistema adiciona Actividade ao Contrato
		Sistema calcula custo total
		Sistema calcula datas e valores das prestações
		Sistema adiciona Contrato à lista de contratos
		Sistema adiciona custo total ao Registo
		Voltar ao passo 1
		Voltar ao passo 3
		Voltar ao passo 6

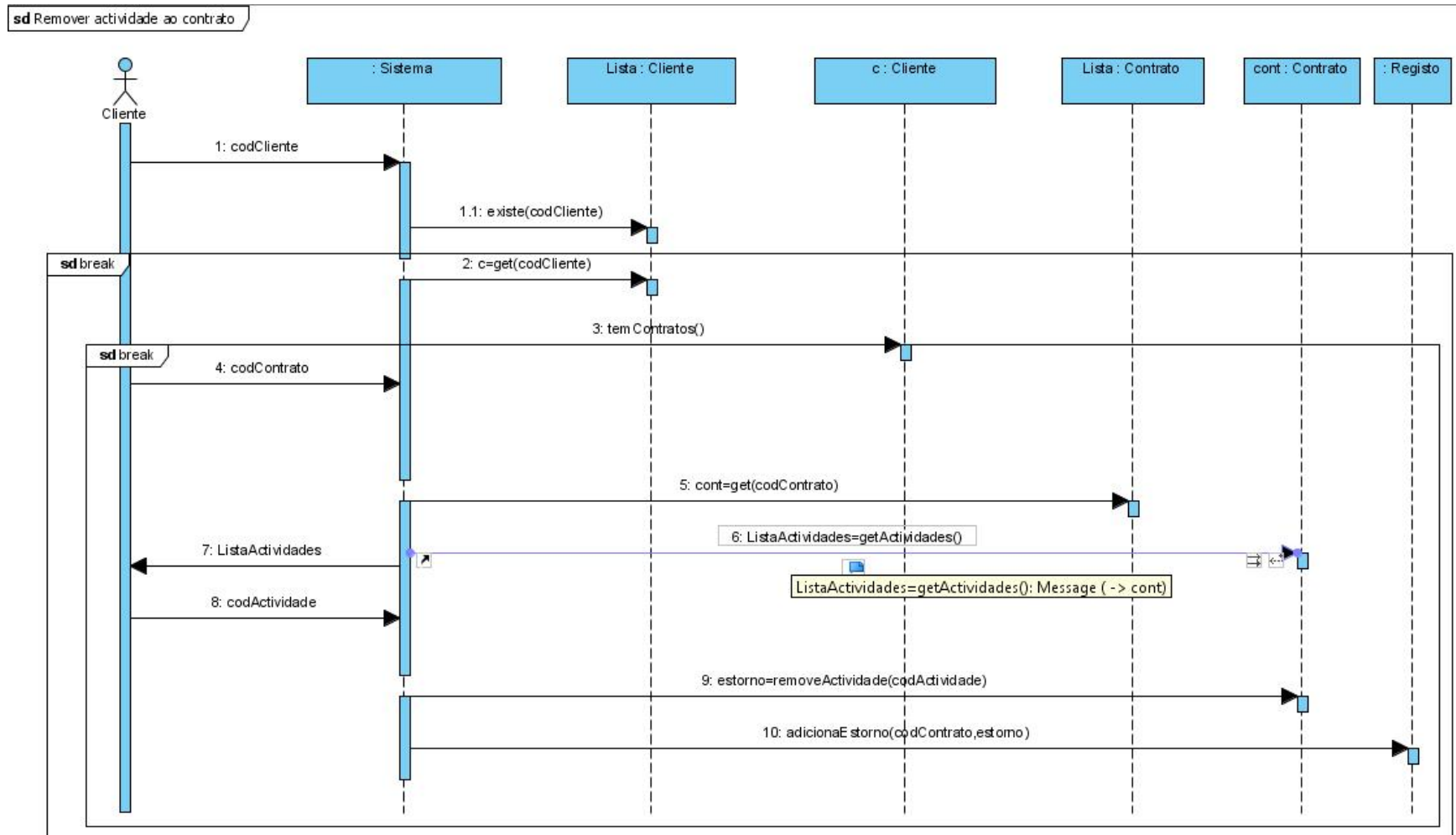
Adicionar Actividade a Contrato

sd Adicionar actividade ao contrato



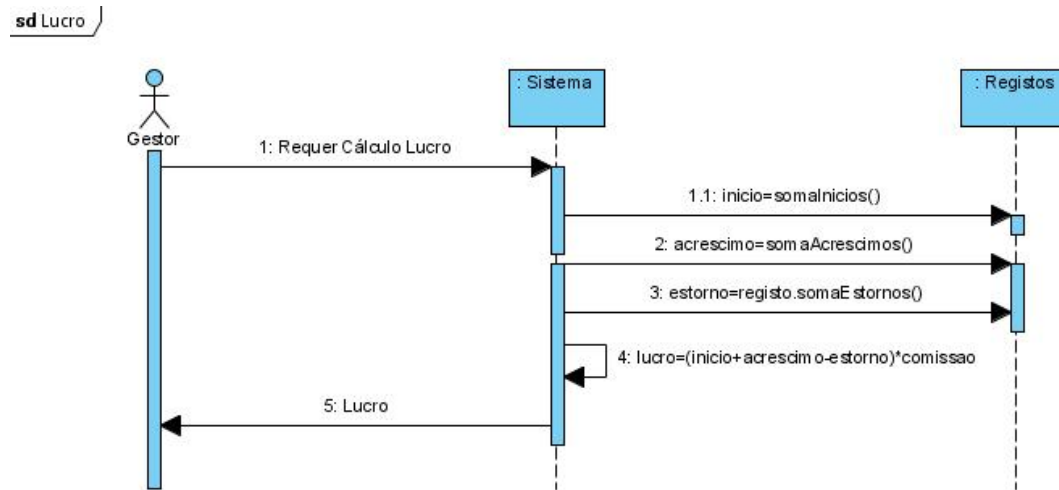
Super Use Case		
Author Ze Camafeu		
Date 1/Dez/2009 17:29:53		
Brief Description		
Actor Cliente		
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	Cliente insere Código de Cliente
	2	
	3	
	4	Cliente insere Código de Contrato
	5	
	6	Cliente insere nova Actividade a adicionar ao Contrato
	7	
	8	
	9	
	10	
Alternative Flow of Events: 2a - Código de Cliente inválido		Actor Input
	1	
Alternative Flow of Events: 3a - Cliente não tem contratos		System Response
	1	Sistema informa Cliente que não tem contratos
	2	Terminar Use Case
Alternative Flow of Events: 7a - Actividade já está no Contrato		Actor Input
	1	

Remover Actividade de Contrato



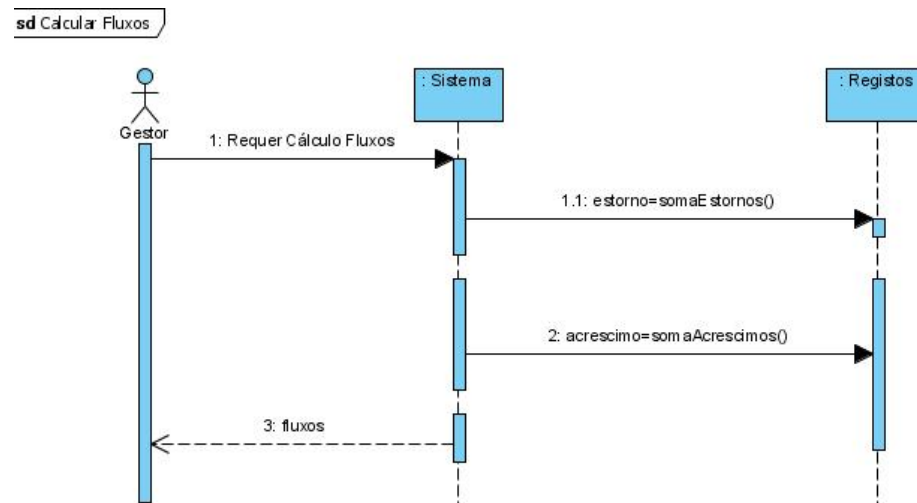
Super Use Case			
Author	Ze Camafeu		
Date	1/Dez/2009 17:45:42		
Brief Description			
Actor	Cliente		
Preconditions			
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Cliente insere Código de Cliente	
	2		Sistema verifica Código
	3		Sistema verifica se Cliente tem Contratos
	4	Cliente insere Código de Contrato	
	5		Sistema verifica Código
	6	Cliente insere Actividade a remover	
	7		Sistema verifica se Actividade está no Contrato
	8		Sistema calcula estorno a devolver
9		Sistema insere estorno no registo	
Alternative Flow of Events: 2a - Código inválido		Actor Input	System Response
	1		Voltar ao passo 1
Alternative Flow of Events: 3a - Cliente não tem contratos		Actor Input	System Response
	1		Sistema informa Cliente
	2		Terminar Use Case
Alternative Flow of Events: 5a - Código inválido		Actor Input	System Response
	1		Voltar ao passo 4
Alternative Flow of Events: 7a - Actividade não está presente no Contrato		Actor Input	System Response
	1		Voltar ao passo 6

Calcular Lucro



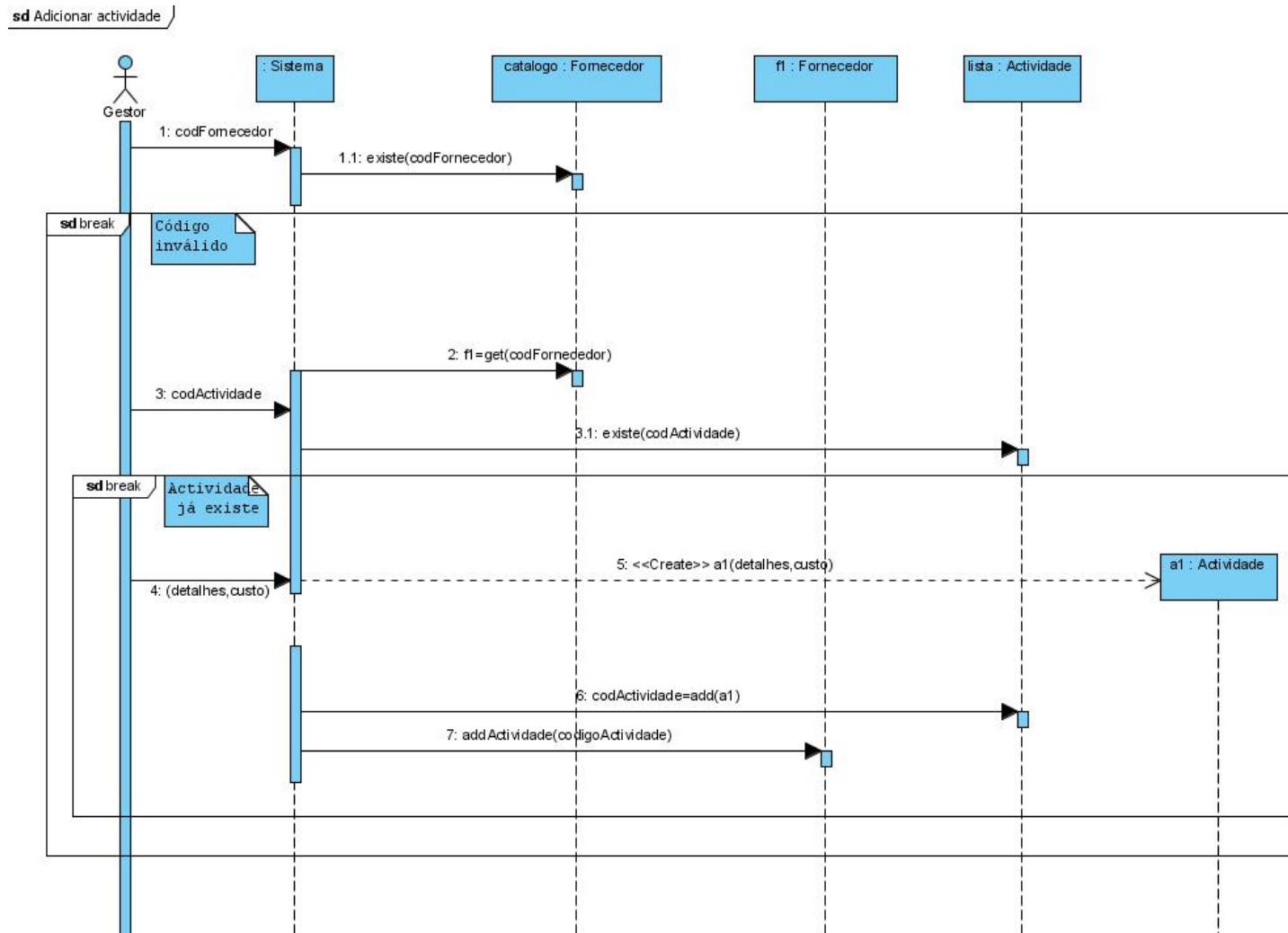
Super Use Case		
Author	Ze Camafeu	
Date	1/Dez/2009 17:19:34	
Brief Description		
Actor	Gestor	
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	Gestor requer cálculo do lucro
	2	
	3	
	4	
	5	
	6	
		System Response
		Sistema acede ao Registo
		Sistema soma Inícios de Contrato
		Sistema soma Alterações de Contrato
		Sistema calcula lucro
		Sistema apresenta lucro ao Gestor

Calcular Fluxos



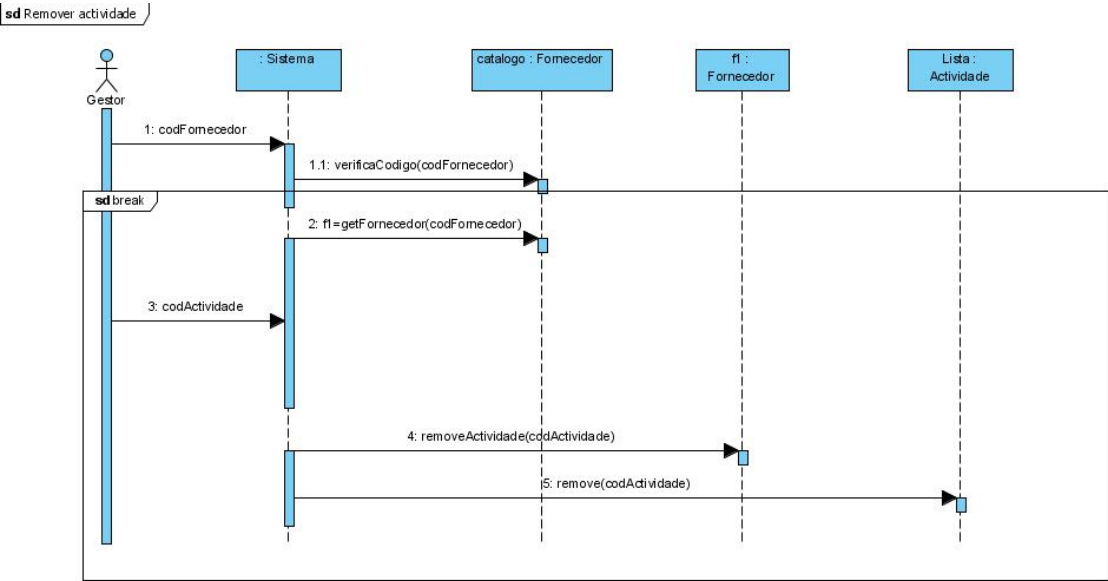
Super Use Case		
Author		
Date		
Brief Description		
Actor		
Preconditions		
Post-conditions		
Flow of Events		Actor Input
	1	Gestor requer calculo de fluxos
	2	Sistema acede ao Registo
	3	Sistema soma os valores de todas as operações do tipo "Adicionar Actividade a Contrato" e "Remover Actividade a Contrato"
	4	Sistema apresenta as informações ao Gestor

Adicionar Actividade



Super Use Case		
Author	Ze Camafeu	
Date	1/Dez/2009 17:06:24	
Brief Description		
Actor	Gestor	
Preconditions		
Post-conditions	Sucesso: Actividade adicionada a um fornecedor Insucesso: Sistema mantém-se	
Flow of Events		Actor Input
	1	Gestor insere código do fornecedor
	2	
	3	Gestor insere código da actividade
	4	
	5	Gestor adiciona detalhes da actividade
	6	
Alternative Flow of Events: 2a- Fornecedor não existe		System Response
	1	Voltar ao passo 1
Alternative Flow of Events: 4a- Actividade já existe		System Response
	1	Voltar ao passo 3

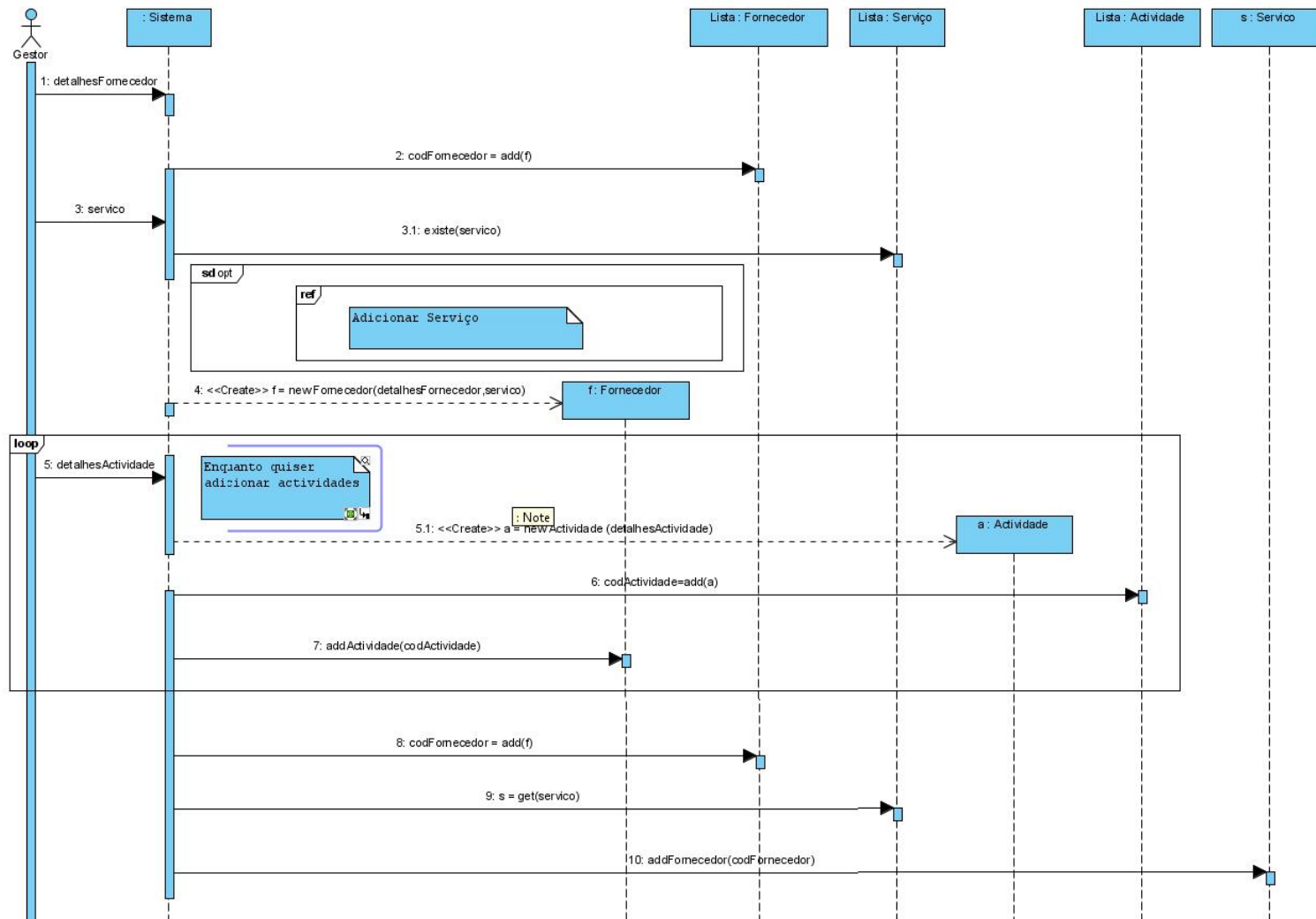
Remover Actividade



Super Use Case		
Author	flp05	
Date	Dec 1, 2009 6:59:32 PM	
Brief Description		
Preconditions		
Post-conditions	Sucesso: Actividade removida a um fornecedor Insucesso: Sistema inalterado	
Flow of Events	Actor Input	System Response
	1 Gestor insere código do fornecedor	
	2	Sistema verifica se fornecedor existe
	3 Gestor insere código da actividade	
	4	Sistema remove actividade da lista de actividades do fornecedor, e da lista de Actividades
Alternative Flow of Events: 2a- Fornecedor não existe	Actor Input	System Response
	1	Voltar ao passo 1
Alternative Flow of Events: 4a- Actividade não existe	Actor Input	System Response
	1	Voltar ao passo 3

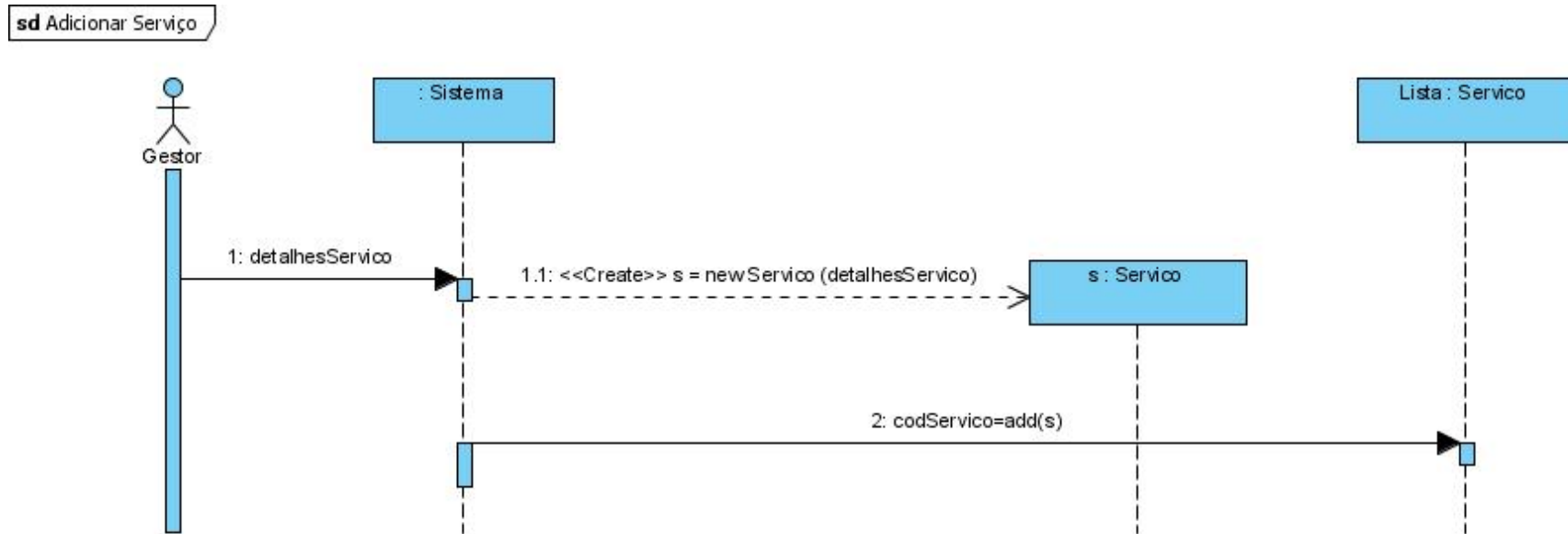
Adicionar Fornecedor

sd Adicionar Fornecedor



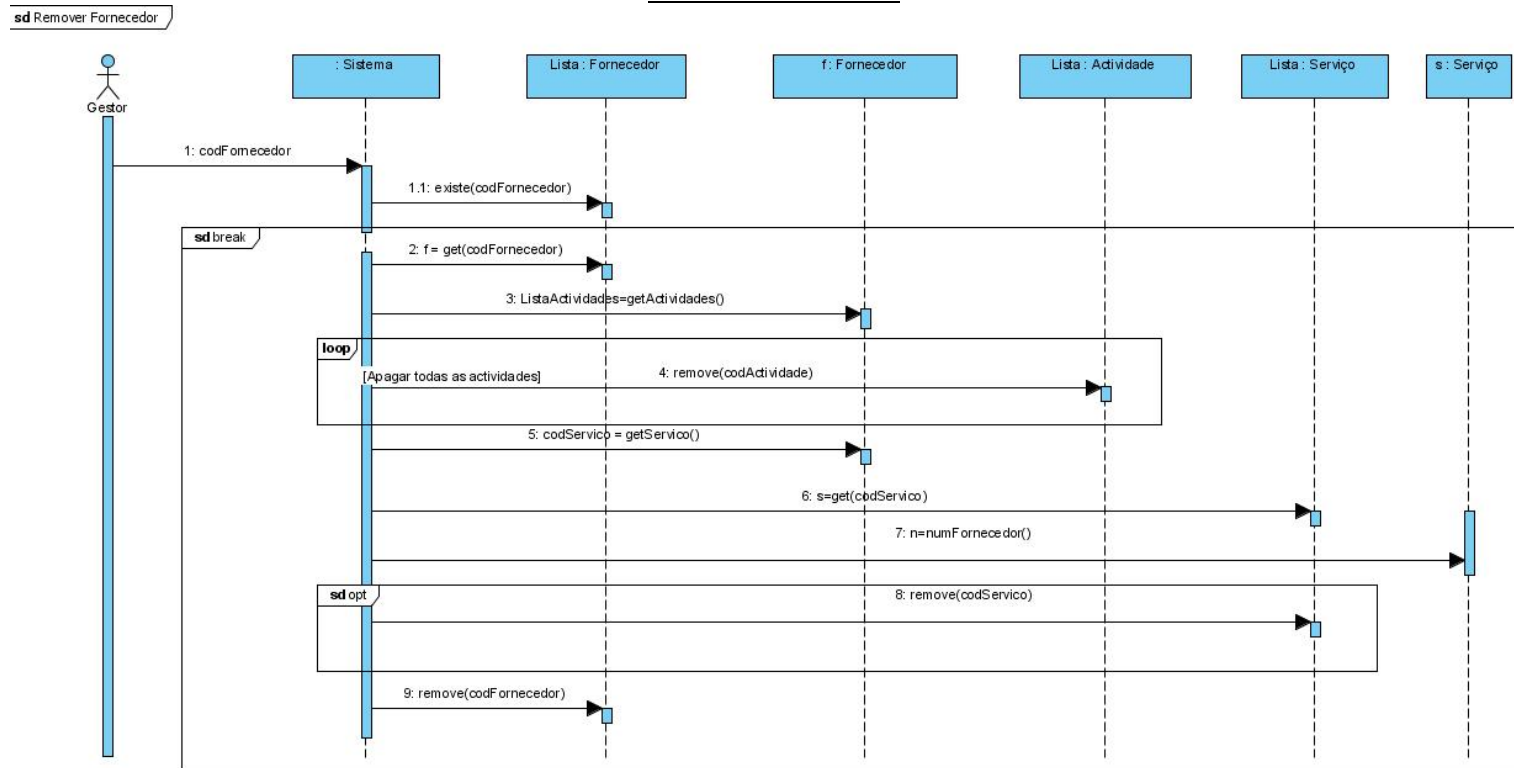
Super Use Case		
Author	Ze Camafeu	
Date	1/Dez/2009 16:51:22	
Brief Description		
Actor	Gestor	
Preconditions		
Post-conditions	Sucesso: Fornecedor adicionado ao Sistema Insucesso: Sistema mantém-se inalterado	
Flow of Events		Actor Input
	1	Gestor insere detalhes do fornecedor
	2	Gestor insere Serviço que o Fornecedor presta
	3	
	4	Gestor insere Actividade do Fornecedor
	5	
	6	
	7	Gestor confirma Actividade
	8	
	9	
Alternative Flow of Events: 3a - Extension Point Não existe serviço		Actor Input
	1	
Alternative Flow of Events: Gestor pretende inserir mais actividades		System Response
		extended by: Adicionar Serviço
		Actor Input
	1	
		System Response
		Voltar ao passo 4

Adicionar Serviço



Super Use Case		
Author	Ze Camafeu	
Date	1/Dez/2009 17:16:30	
Brief Description		
Actor	Gestor	
Preconditions	Serviço a adicionar não existe	
Post-conditions	Serviço adicionado ao Sistema	
Flow of Events		Actor Input
	1	Gestor insere detalhes do Serviço
	2	
		System Response
		Sistema actualiza lista de Serviços

Remover Fornecedor



Use Case ID		
Super Use Case		
Primary Actor	Gestor	
Secondary Actor(s)		
Brief Description	O Gestor pretende remover um fornecedor	
Preconditions		
Flow of Events		Actor Input
	1	O gestor insere o código do fornecedor a remover
	2	
	3	
	4	
	5	
		System Response
		O sistema valida o código do fornecedor
		Sistema remove Actividades do fornecedor
		Verificar se outro fornecedor fornece este serviço
		O sistema remove o fornecedor dos registos
Post-conditions	Sucesso: Fornecedor removido do sistema Insucesso: Sistema não foi alterado	
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	flp05	
Date	Dec 1, 2009 6:24:58 PM	
Alternative Flow of Events: 2a - Código inválido		Actor Input
	1	
		System Response
		Voltar ao passo 1
Extension Point 4a - Nenhum outro fornecedor fornece este serviço		Actor Input
	1	
		System Response
		Sistema remove Serviço da lista de Serviços

Conclusão

Na modelação dos novos diagramas, verificamos que era necessário proceder a alterações nos diagramas anteriores. Assim conseguimos fazer com que os modelos estejam mais próximos daquilo que consideramos ser ideal para a aplicação final.

O desenvolvimento de diagramas de sequência de sistema torna mais fácil a compreensão e execução do programa a ser feito. Estes, em conjunto com o diagrama de Use Cases e o diagrama de Classes, dão uma grande ideia de como se irá tornar a aplicação, e facilitam em muito o desenvolvimento do código.

Sentimos que atingimos os objectivos a que nos propusemos até aqui, na medida em que já temos uma noção completa de como será o resultado final, e que respeitará todos os requisitos impostos.