



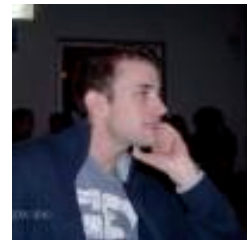
Universidade do Minho
Conselho de Cursos de Engenharia
Licenciatura em Engenharia Informática
3ºAno

Disciplina de Desenvolvimento de Sistemas de Software
Ano Lectivo de 2009/20010
Dezembro, 2009

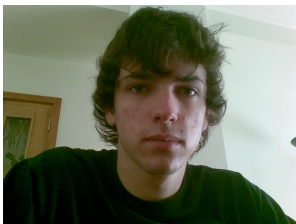
GereComSaber – Gestão de Condomínios
Grupo 14



Sandro Machado Nº 47113



João Peixoto Nº 51757



Fábio Vaz Nº 51748



Celso Coutinho Nº47037



Nuno Ferreira Nº 47426

Resumo

No presente relatório pretende-se detalhar toda a fase de desenvolvimento do sistema de gestão de condomínios GereComSaber, realizado no âmbito da Unidade Curricular de Desenvolvimento de Sistemas de *Software* (DSS) do 3º ano da Licenciatura em Engenharia Informática da Universidade do Minho. Na primeira parte este documento contém um desenho da solução, assim como todas as decisões tomadas e descrição da concepção do modelo de Domínio e de *Use Cases* para a primeira fase de desenvolvimento. Na segunda parte contém os diagramas de sequência (Sistema, Sub-Sistema e Implementação) relativos à primeira parte e o diagrama de classes do sistema.

Índice

- 1.** Introdução
- 2.** Desenho da Solução
- 3.** Decisões de Implementação
- 4.** Diagrama de Modelo de Domínio
- 5.** Diagramas Use Case
- 6.** Descrição textual dos diagramas Use Case
- 7.** Diagramas de Sequência
- 8.** Diagrama de Classes
 - 8.1.** Descrição das Classes
- 9.** Conclusão

1. Introdução

Numa sociedade em que as pessoas tendem a assumir um ritmo de vida que lhes consome quase todo o tempo livre, as tarefas domésticas surgem em segundo plano relativamente ao trabalho. Como tal, é necessário arranjar uma maneira de lidar com essas tarefas de modo a poupar o máximo de tempo possível. É neste contexto que surgem empresas que fazem a gestão de serviços em condomínios fechados. Os condóminos escolhem as tarefas que desejam ver realizadas ao longo de um determinado período, e a empresa garante que um fornecedor do serviço escolhido o irá realizar.

A empresa GereComSaber é uma destas empresas. Os condóminos escolhem um ou mais serviços do conjunto de serviços que esta empresa oferece, sendo em seguida listadas todas as actividades que podem ser encomendadas, assim como os respectivos fornecedores e preços.

Seguindo as directivas da unidade curricular de DSS, foi proposto o desenvolvimento do sistema informático da GereComSaber. Este sistema requer um bom planeamento e desenvolvimento, para que o produto final satisfaça os clientes, os administradores da empresa e os fornecedores. O desenvolvimento deste sistema em particular segue uma linhagem RUP (*Rapid Unified Process*), tendo assim como base o UML (*Unified Modeling Language*), que funciona como a ponte entre o cliente e a equipa de desenvolvimento.

Esta linguagem é provida de uma notação gráfica muito rica para representar os sistemas, que foi criada pela OMG (*Object Management Group*). Foca-se na modelação de aplicações que utilizam uma linguagem de programação orientada aos objectos.

A UML oferece um grafismo muito intuitivo para ser possível projectar a aplicação. São oferecidos elementos como actores, processos, componentes actividades, vários tipos de diagramas, etc. Entende-se então que esta linguagem é muito poderosa para planificar uma aplicação com uma linguagem orientada aos objectos.

De modo a ajudar a uma melhor compreensão do nosso trabalho, sugere-se o presente relatório, que se encontra dividido nas secções que se apresentam de seguida. Numa primeira fase falar-se-á de como foi desenhada a solução apresentada. De realçar que tal desenho foi efectuado ainda numa fase prematura de desenvolvimento. Seguidamente serão abordadas as decisões de implementação mais pertinentes e merecedoras de referência. Depois desta elaboração teórica do problema, são apresentados os diagramas de Use Cases, Sequência e de Classes com as respectivas descrições textuais. Finalmente será apresentada uma pequena conclusão sobre o estado actual do desenvolvimento.

2. Desenho da Solução

Para conceber o desenho da solução foi efectuada uma pequena análise aos serviços existentes na área de gestão de condomínios, para perceber a complexidade do problema, e até que nível de detalhes e áreas de aplicação o sistema deve abranger.

Como tal, e após efectuada essa análise, tornou-se claro que o sistema deverá disponibilizar uma variedade de serviços, com diversas actividades, por cada fornecedor que seja adequado à exigência de cada condómino. Este passo incorpora também a possibilidade da concretização ou término de qualquer contracto existente.

Os fornecedores também terão acesso ao sistema de forma a poder consultar os serviços que têm a realizar.

Além disso o sistema terá um administrador que irá gerir os utilizadores autorizados no sistema (nomeadamente os condóminos e os fornecedores), atribuindo a cada um uma credencial de acesso. O administrador vai também gerir os serviços, actividades e fornecedores a disponibilizar aos condóminos, assim como todos os métodos de pagamentos existentes e um histórico de todas as operações realizadas.

De assinalar que em cada contracto criado, a empresa GereComSaber irá cobrar uma taxa de lucro fixa. Um contracto nunca poderá ser editado, ou seja, se um condómino pretender mais um serviço após algum tempo, terá que realizar novo contracto, sendo o anterior cancelado e feito o devido acerto. De referir também que estas suposições se baseiam no tipo de serviços disponibilizados por terceiros e que, depois de adoptada uma panorâmica enquadrada com o enunciado do problema, se revelaram como funcionalidades 'a ter' no sistema.

3. Decisões de Implementação

Quando se começou a implementar o diagrama de *Use Cases*, feita a análise de requisitos e após os primeiros esboços, verificou-se que o problema teria de ser mais detalhado do que aquilo que estava a ser feito. Começou-se por analisar o que cada utilizador poderia fazer no sistema, tendo-se decidido criar *packages* com os diferentes *Use Cases* de cada utilizador: Condómino, Admin e Fornecedor. Aqui surgiu o primeiro problema pois havia *Use Cases* comuns ao Condómino e ao Admin. Para resolver tal problema, criou-se uma dependência entre estes *packages*, colocando os *Use Cases* descritos no *package* Admin, pois é o *package* de maior importância. No *package* Condómino é feita referência ao uso destes *Use Cases*.

Na realização dos vários *Use Cases* verificou-se que muitos deles necessitavam das opções Seleccionar Serviço, Seleccionar Fornecedor e Seleccionar Actividade. Assim, estas opções foram criadas como *Use Cases*, e sempre que eram necessárias noutros *Use Cases*, foi feito o *include*. Evitou-se assim a repetição de instruções nos detalhes de vários *Use Cases*. Um caso particular tratado depois, foi o de Seleccionar Actividade, que necessitava de Seleccionar Serviço e Seleccionar Fornecedor *a priori*. Assim, Seleccionar Actividade faz *include* destes dois *Use Cases*.

Para além desta simplificação, o *include* destes *Use Cases* por parte de outros *Use Cases*, reduziram ainda mais os detalhes destes, pois já não foi preciso tratar de alguns *Alternative Flow of Events*.

Um outro problema que levou a vários motivos de alteração, foi o dos contractos poderem sofrer acertos, ou seja, o Condómino pode cancelar os contractos sem estes terem acabado. Assim, cada Condómino possuirá um Histórico para no final do período contabilístico, poder ser feito o ajuste da factura desse Condómino. De notar que um contracto nunca pode ser alterado.

Nesta segunda fase de desenvolvimento melhorou-se o modelo de domínio devido à existência de erros, nomeadamente as relações entre entidades. Foram corrigidos alguns *Use Cases* com falhas de implementação e adicionaram-se também os diagramas de sequência correspondentes ao Sistema, Sub-Sistema e Implementação. O diagrama de Sistema apresenta a interacção básica entre o actor e o sistema GereComSaber. No diagrama de Sub-Sistema implementou-se a interacção entre o actor e todos os outros sub-sistemas considerados necessários para completar o *Use Case* correspondente. O diagrama de Implementação é baseado no diagrama de Sub-Sistema, onde a descrição da correspondência entre sub-sistemas é substituído pelos métodos que estão representados no diagrama de classes. De notar que os métodos já contidos nas bibliotecas principais de JAVA não são considerados.

4 .Diagrama de Modelo de Domínio

Visual Paradigm for UML Standard Edition(Universidade do Minho)

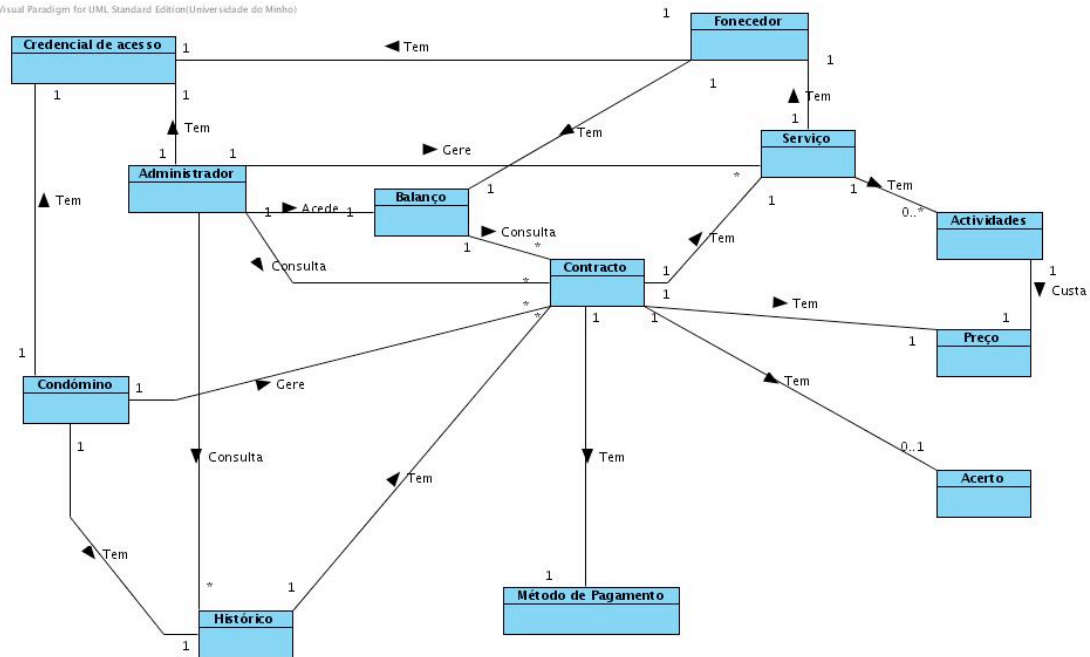


fig. Modelo de Domínio

Name:

Parent:  

Visibility: 

Documentation:

HTML  **B** *I* u                                  

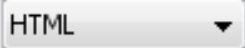
O acerto é usado sempre que o contracto é alterado pelo utilizador . Os resultados dos acertos são guardados no histórico do contracto.

Name:

Parent:  

Visibility: 

Documentation:

 **B** *I* u      **F** *F*         

As actividades estão directamente ligadas aos serviços, pois cada serviço pode ter várias actividades diferentes. Por exemplo, no serviço de Jardinagem, podemos ter as actividades de cortar a relva, cortar arbustos, plantar flores, ... Cada actividade tem um preço diferente.

Name:

Parent:  

Visibility: 

Documentation:

HTML  **B** *I* u                              

Name:

Parent:  

Visibility: 

Documentation:

HTML  **B** *I* u                        

Class Specification

Tagged Values

Constraints

Diagrams

References

Comments

Class Code Details

Java Annotations

ORM Query

Stereotypes

General

Attributes

Operations

Relations

Template Parameters

Name:

Balanço

Parent:

Modelo domínio

...

Visibility:

public

▼

Documentation:

HTML

▼

B *I* u      **F** **Ff**                    

Class Specification

Tagged Values Constraints Diagrams References Comments

Class Code Details Java Annotations ORM Query Stereotypes

General Attributes Operations Relations Template Parameters

Name: Condómino

Parent: Modelo domínio

Visibility: public

Documentation:

HTML **B I u** [list icons] **F F** [other icons]

O condómino é um dos utilizadores do programa. Este utilizador é muito limitado, pois apenas pode gerir os seus contractos. Ou seja, o objectivo é simplesmente o condómino poder subscrever aos serviços que deseja para a sua casa. Este irá guardar um histórico de contractos e os contractos actuais.

Class Specification

Tagged Values

Constraints

Diagrams

References

Comments

Class Code Details

Java Annotations

ORM Query

Stereotypes

General

Attributes

Operations

Relations

Template Parameters

Name:

Contracto

Parent:

Modelo domínio

...

Visibility:

public

▼

Documentation:

HTML

B

I

u

≡

≡

≡

≡

≡

F

F

↶

↷

↶

↷

↶

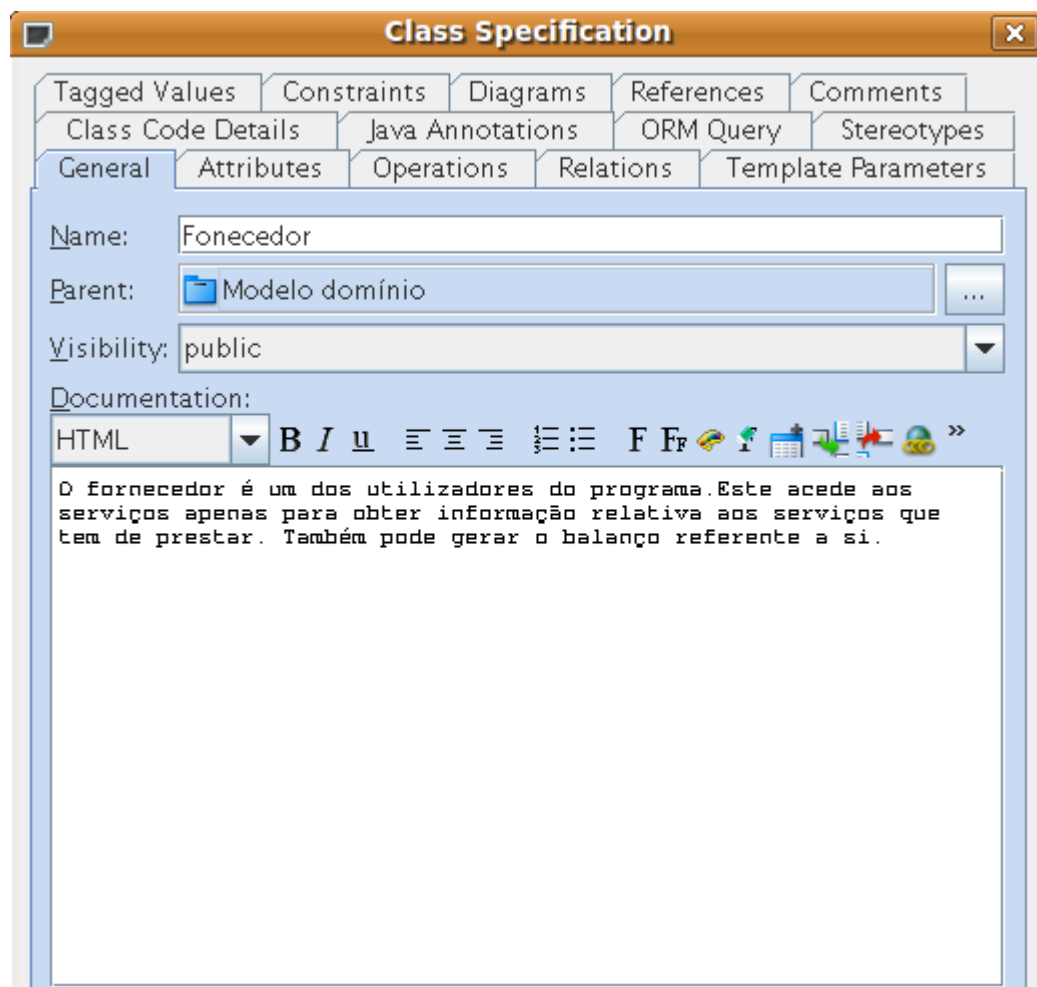
↷

↶

↷

»

Um contrato é feito por um cliente quando este subscreve um serviço. Este guarda todos os dados necessários: serviço contratado, preço e método de pagamento.



Class Specification

Tagged Values

Constraints

Diagrams

References

Comments

Class Code Details

Java Annotations

ORM Query

Stereotypes

General

Attributes

Operations

Relations

Template Parameters

Name:

Histórico

Parent:

Modelo domínio

...

Visibility:

public

▼

Documentation:

HTML

B

I

u

≡

≡

≡

≡

≡

F

F

»

Cada Condómino irá ter um histórico com os contractos já expirados ou cancelados.

Name:

Parent:  

Visibility: 

Documentation:

HTML  **B** *I* u                      

Name:

Parent:  

Visibility: 

Documentation:

HTML  **B** *I* u                                 

Name:

Parent:  

Visibility: 

Documentation:

HTML  **B** *I* u      **F** *F*            

Aqui estão englobados todos os tipos de serviço que a GereComSaber disponibiliza aos seus clientes através dos seus fornecedores. Cada tipo de serviço pode ter várias actividades. Para o cliente subscrever um serviço tem de ser feito um contracto.

Name:

Parent:  

Visibility: 

Documentation:

 **B** *I* u      **F** *F*         

Cada contracto vai ter associada uma taxa de lucro, pois o fornecedor dá um preço à GereComSaber e esta ganha uma percentagem desse valor, por ser intermediária do condómino.

5. Diagramas de Use Case

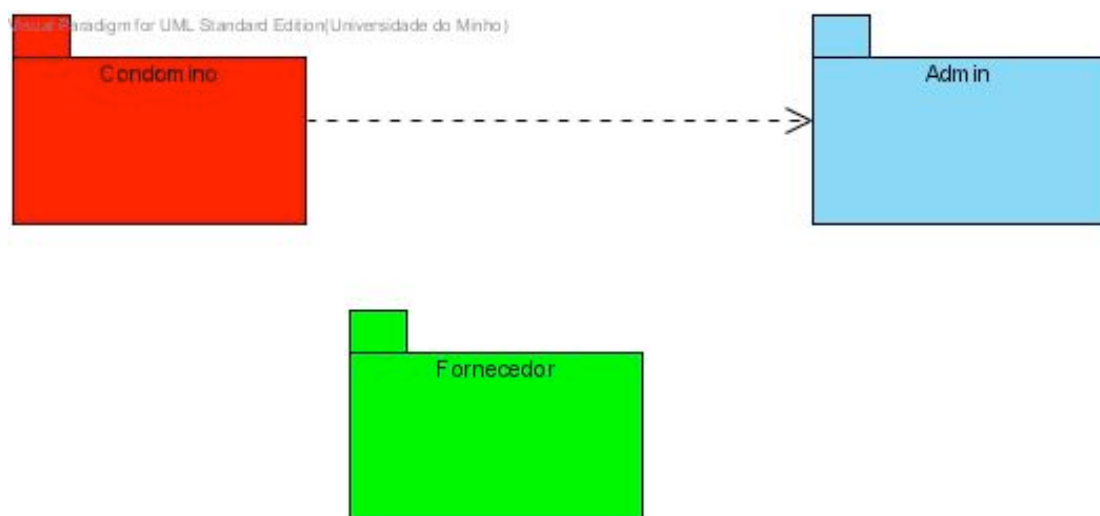


fig. GereComSaber

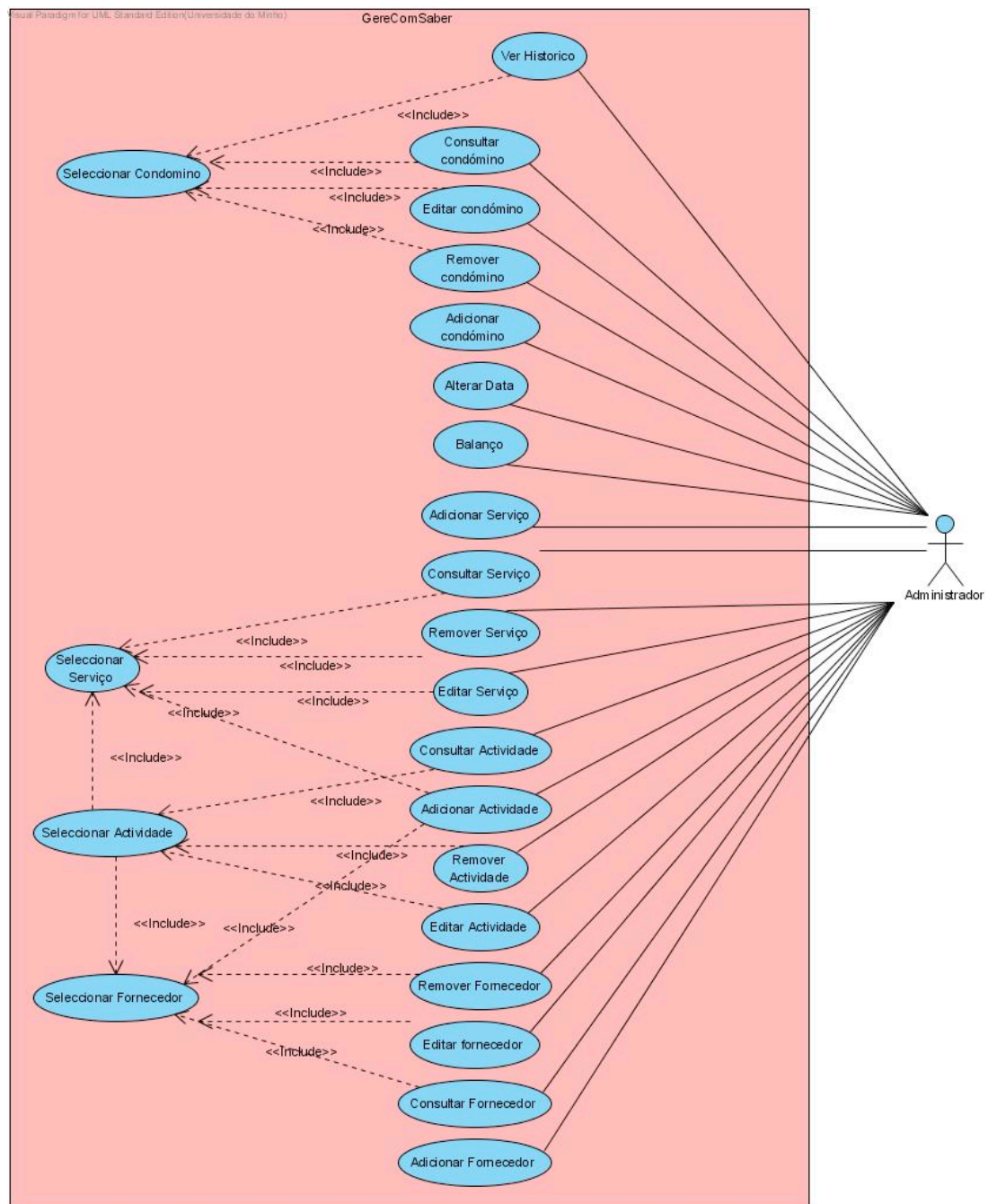


fig. Use Case Administrador

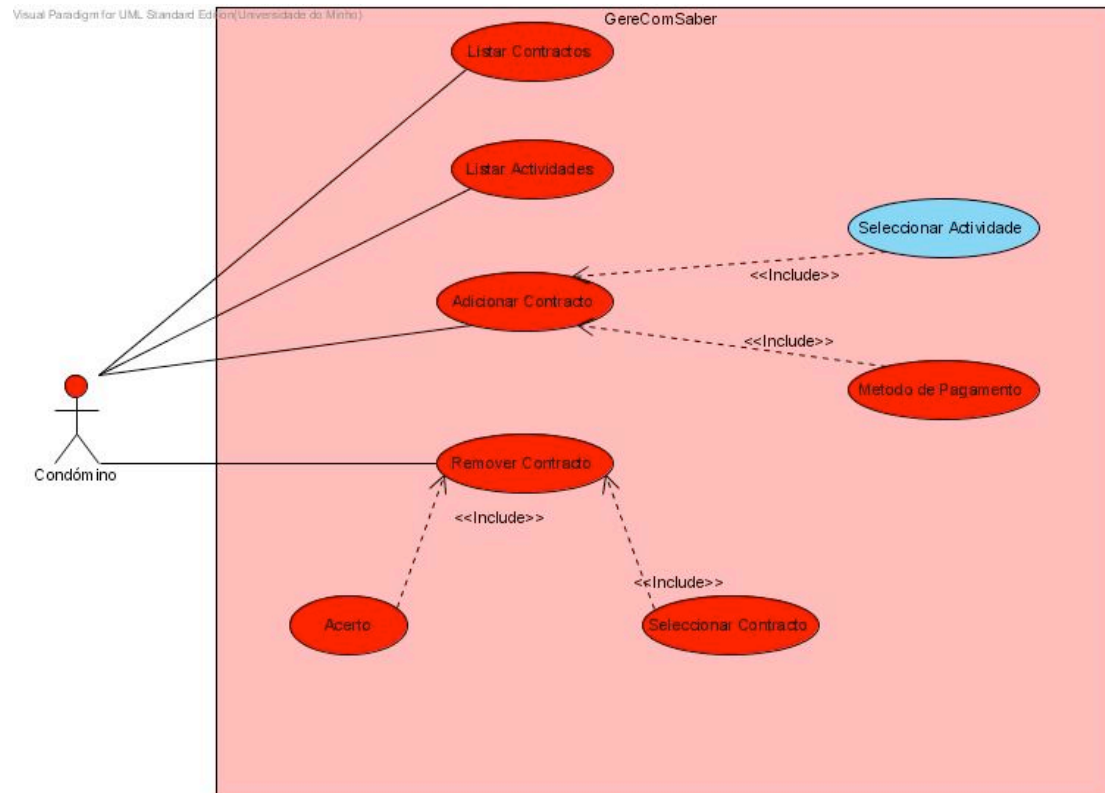


fig. Use Case Condómino

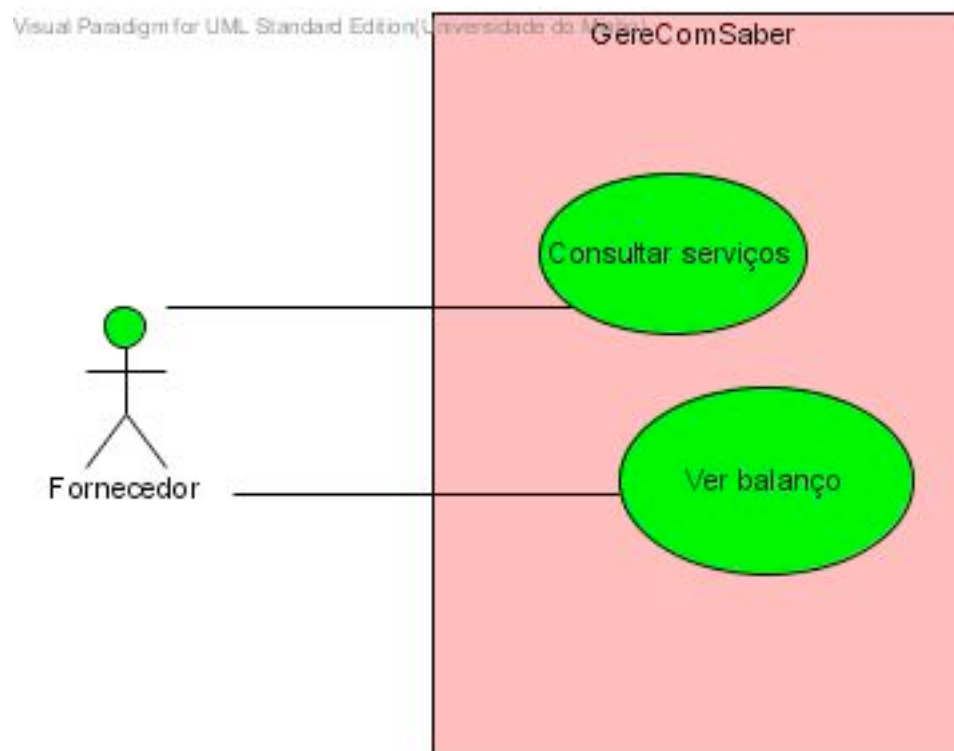


fig. Use Case Fornecedor

7. Descrição textual dos diagramas de Use Case

Use Case ID	Adicionar Actividade		
Super Use Case			
Primary Actor	Administrador		
Secondary Actor(s)			
Brief Description	O Administrador pretende adicionar uma Actividade num dado Serviço de um dado Fornecedor		
Preconditions			
Flow of Events		Actor Input	System Response
	1	Include: Seleccionar Fornecedor	
	2	Include: Seleccionar Serviço	
	3		Pedir os dados da Actividade
	4	Introduzir os dados da Actividade	
	5		Verificar os dados inseridos
	6		Introduzir a Actividade no Serviço oferecido pelo Fornecedor
	7		Confirmar a inserção da Actividade
Post-conditions	Uma Actividade foi introduzida		
Alternative flows and exceptions			
Alternative flow of event 6a – Dados inválidos		Actor Input	System Response
	1		Informar qual o erro
	2		Voltar ao passo 3
Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	20		
Date	30/Nov/2009 22:39:27		

Use Case ID	Adicionar Condômino	
Super Use Case		
Primary Actor	Admin	
Secondary Actor(s)		
Brief Description	O administrador vai adicionar uma novo condômino	
Preconditions		
Flow of Events		Actor Input
	1	Introduzir dados do condômino
	2	
	3	
		System Response
		Registrar novo condômino
		Atualizar dados
Post-conditions	Um condômino foi introduzido no condomínio	
Alternative flows and exceptions		
Alternative flow of event		Actor Input
2a - Condômino já existente	1	
	2	
		System Response
		Avisar que Condômino a inserir já existe
		Voltar ao passo 1
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Fábio	
Date	28/Nov/2009 18:56:32	

Use Case ID	Adicionar Fornecedor	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O Administrador pretende introduzir um novo Fornecedor	
Preconditions		
Flow of Events		Actor Input
	1	
	2	Introduzir dados do Fornecedor
	3	
	4	
	5	
		System Response
		Pedir dados do Fornecedor
		Verificar dados inseridos
		Inserir Fornecedor
		Confirmar inserção do Fornecedor
Post-conditions	Um novo Fornecedor foi introduzido no Sistema	
Alternative flows and exceptions		
Alternative flow of event 4a – Dados inválidos		Actor Input
	1	
	2	
		System Response
		Informar qual o erro
		Voltar ao passo 1
Non-behavior requirements		
Assumptions		
Issue		

Use Case ID	Adicionar Serviço	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O administrador pretende adicionar um serviço	
Preconditions		
Flow of Events		Actor Input
	1	Inserir informações do serviço
	2	
	3	
Post-conditions	Novo serviço disponível	
Alternative flows and exceptions		
Alternative flows		Actor Input
2.a - O serviço já existe	1	
Alternative flows		System Response
2.b - Dados insuficientes	1	
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Sandro	
Date	28/Nov/2009 18:49:47	

Use Case ID	Alterar data	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O administrador pretende alterar a data do programa	
Preconditions		
Flow of Events		Actor Input
	1	Inserir a data
	2	
	3	
Post-conditions		
Alternative flows and exceptions		
Alternative flows		Actor Input
2.a - A data não é válida	1	
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Sandro	
Date	28/Nov/2009 18:21:28	

Use Case ID	Balção	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O administrador pretende gerar o balanço do sistema	
Preconditions		
Flow of Events		Actor Input
	1	
	2	
		System Response
		Calcular o balanço do sistema
		Imprime no ecrã o balanço do sistema
Post-conditions		
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Sandro	
Date	28/Nov/2009 18:29:58	

Use Case ID	Consultar Actividade	
Super Use Case		
Primary Actor		
Secondary Actor(s)		
Brief Description	O Administrador pode consultar uma determinada Actividade de um determinado Serviço oferecido por um dado Fornecedor	
Preconditions		
Flow of Events		Actor Input
	1	Include: Seleccionar Actividade
	2	
		System Response
		Imprimir detalhes da actividade
Post-conditions		
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	20	
Date	30/Nov/2009 22:46:27	

Use Case ID	Consultar Fornecedor	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O Administrador pretende consultar os dados de um Fornecedor	
Preconditions		
Flow of Events		Actor Input
	1	Include: Seleccionar Fornecedor
	2	Imprimir dados do fornecedor
Post-conditions		
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	20	
Date	30/Nov/2009 23:12:19	

Use Case ID	Ver Historico	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O administrador pretende saber o historico de um determinado contracto	
Preconditions		
Flow of Events		Actor Input
	1	Include: Seleccionar condómino
	2	O sistema imprime o historico do contracto
Post-conditions		
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Sandro	
Date	28/Nov/2009 15:58:13	

Use Case ID	Consultar Serviço	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O Administrador pretende consultar os serviços existentes	
Preconditions		
Flow of Events		Actor Input
	1	Include: Seleccionar Serviço
	2	Imprimir no ecrã os detalhes do serviço
Post-conditions		
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Sandro	
Date	28/Nov/2009 18:43:13	

Use Case ID	Consultar Condómino	
Super Use Case		
Primary Actor	Admin	
Secondary Actor(s)		
Brief Description	Consultar o condómino	
Preconditions		
Flow of Events		Actor Input
	1	include: Seleccionar Condomino
	2	Dar informação do condómino
Post-conditions		
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Fábio	
Date	28/Nov/2009 18:55:55	

Use Case ID	Editar Actividade	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O Administrador pretende editar uma Actividade de um determinado Serviço oferecido por um dado Fornecedor.	
Preconditions		
Flow of Events		Actor Input
	1	Include: Seleccionar Actividade
	2	
	3	Escolher o campo a alterar
	4	
	5	Introduzir os dados da Actividade
	6	
	7	
	8	
		System Response
		Imprimir dados da Actividade
		Pedir novos dados da Actividade
		Verificar os dados inseridos
		Alterar a Actividade no Serviço oferecido pelo Fornecedor.
		Confirmar que a Actividade foi alterada
Post-conditions	Uma actividade é alterada	
Alternative flows and exceptions		
Exception		
		Actor Input
		System Response
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	20	
Date	30/Nov/2009 23:03:23	

Use Case ID	Editar condómino	
Super Use Case		
Primary Actor	Admin	
Secondary Actor(s)		
Brief Description	Alterar dados de um condómino	
Preconditions		
Flow of Events		Actor Input
	1	include: Seleccionar Condómino
	2	
	3	Editar informação do condómino
	4	
		System Response
		Dar informação do condómino
		Verificar alterações do condómino
		Actualizar dados
Post-conditions	Um condómino foi editado	
Alternative flows and exceptions		
Editar condómino		Actor Input
3.a Alterações não permitidas	1	
		System Response
		Voltar ao passo 3
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Fábio	
Date	28/Nov/2009 18:56:08	

Use Case ID	Editar Fornecedor		
Super Use Case			
Primary Actor	Administrador		
Secondary Actor(s)			
Brief Description	O admistrador pretende alterar os dados de um fornecedor		
Preconditions			
Flow of Events		Actor Input	System Response
	1	Include: Seleccionar Fornecedor	
	2		Imprimir dados do Fornecedor
	3	Escolher o campo a alterar	
	4		Pedir dados do Fornecedor
	5	Introduzir dados do Fornecedor	
	6		Verificar dados inseridos
	7		Alterar Fornecedor
	8		Confirmar a alteração do Fornecedor
Post-conditions	Um Fornecedor foi alterado		
Alternative flows and exceptions			
Exception 7a – Dados inválidos		Actor Input	System Response
	1		Mostrar mensagem de erro
Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	20		
Date	30/Nov/2009 23:03:23		

Use Case ID	Editar Serviço	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O administrador pretende editar um serviço	
Preconditions		
Flow of Events		Actor Input
	1	Include: Selecionar Serviço
	2	
	3	O administrador insere as edições
	4	
	5	
	6	
		System Response
		O sistema confirma o serviço
		O sistema verifica as edições
		O sistema edita o serviço
		Confirmar alteração de serviço
Post-conditions	O sistema vai ter um serviço modificado	
Alternative flows and exceptions		
Alternative flows		Actor Input
4.a - Os dados modificados não são permitidos	1	
		System Response
		Voltar ao passo 1
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Sandro	
Date	28/Nov/2009 19:01:47	

Use Case ID	Remover Actividade	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O Administrador pode remover uma Actividade de um determinado Serviço oferecido por um qualquer Fornecedor	
Preconditions		
Flow of Events		Actor Input
	1	Include: Seleccionar Actividade
	2	
	3	
		System Response
		Remover Actividade
		Confirmar a remoção da Actividade
Post-conditions	Uma Actividade foi removida	
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	peixoto	
Date	30/Nov/2009 22:32:10	

Use Case ID	Remover Fornecedor	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O Administrador pretende remover um Fornecedor	
Preconditions		
Flow of Events		Actor Input
	1	Include: Seleccionar Fornecedor
	2	
	3	
		System Response
		Remover o Fornecedor
		Confirmar a remoção do Fornecedor
Post-conditions	Um Fornecedor foi removido	
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	20	
Date	30/Nov/2009 22:55:24	

Use Case ID	Remover Serviço	
Super Use Case		
Primary Actor	Administrador	
Secondary Actor(s)		
Brief Description	O administrador pretende remover um serviço existente	
Preconditions		
Flow of Events		Actor Input
	1	Include: Seleccionar Serviço
	2	
	3	
		System Response
		O sistema removeu o serviço
		Confirmar remoção do serviço
Post-conditions	O sistema ficou com menos um serviço	
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Sandro	
Date	28/Nov/2009 18:55:21	

Use Case ID	Seleccionar Fornecedor	
Super Use Case		
Primary Actor		
Secondary Actor(s)		
Brief Description	O Administrador escolhe um Fornecedor	
Preconditions	Existir pelo menos um fornecedor no sistema	
Flow of Events		Actor Input
	1	Inserir o codigo do fornecedor
	2	Devolver o Fornecedor
Post-conditions		
Alternative flows and exceptions		
Alternative Flow		Actor Input
2a - O sistema nao encontra o nome do Fornecedor inserido	1	
	2	E impressa uma mensagem de erro
		Voltar ao passo 1
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	20	
Date	30/Nov/2009 22:09:18	

Use Case ID	Seleccionar Actividade	
Super Use Case		
Primary Actor		
Secondary Actor(s)		
Brief Description	O Administrador pretende selecionar uma actividade	
Preconditions	Existir pelo menos uma actividade no sistema	
Flow of Events		Actor Input
	1	Include: Selecionar Serviço
	2	Include: Selecionar Fornecedor
	3	Introduzir codigo da actividade
	4	Devolver a actividade pretendida
Post-conditions		
Alternative flows and exceptions		
Alternative flows		Actor Input
4.a - Nao existem actividades com aquele codigo	1	
	2	Imprimir Mensagem de erro
		Voltar ao passo 3
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Sandro	
Date	1/Dez/2009 17:23:15	

Use Case ID	Seleccionar Serviço	
Super Use Case		
Primary Actor		
Secondary Actor(s)		
Brief Description	O Administrador pretende seleccionar um serviço	
Preconditions	Existir pelo menos um serviço no sistema	
Flow of Events		Actor Input
	1	Introduzir o nome do Serviço
	2	Devolver o Serviço escolhido
Post-conditions		
Alternative flows and exceptions		
Alternative flow 2a - O sistema não encontra nenhuma ocorrência do Serviço escolhido		Actor Input
	1	Apresenta uma mensagem de erro
	2	Voltar ao passo 1
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	20	
Date	30/Nov/2009 22:18:48	

Use Case ID	Selecionar Condómino	
Super Use Case		
Primary Actor		
Secondary Actor(s)		
Brief Description	Seleciona um condómino dos existentes	
Preconditions	Existir pelo menos um condómino no sistema	
Flow of Events		Actor Input
	1	Inserir código do condómino
	2	Faz a verificação se existe algum condómino para esse código
Post-conditions		
Alternative Flows 2.a - Verificação falha		Actor Input
	1	Voltar para passo 1
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Sandro	
Date	28/Nov/2009 18:01:21	

Use Case ID	Remover condómino		
Super Use Case			
Primary Actor	Admin		
Secondary Actor(s)			
Brief Description	O administrador pretende remover um condómino		
Preconditions			
Flow of Events		Actor Input	System Response
	1	Include: Seleccionar Condomino	
	2		Remover condómino
	3		Confirmar a remoção do condómino
Post-conditions	Um condómino foi removido		
Alternative flows and exceptions			
Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	Fábio		
Date	28/Nov/2009 18:56:21		

Use Case ID	Listar Actividades		
Super Use Case			
Primary Actor	Condómino		
Secondary Actor(s)			
Brief Description	O condomino pretende requisitar uma lista de actividades.		
Preconditions			
Flow of Events		Actor Input	System Response
	1	O condomino selecciona listar actividades	
	2		O Sistema verifica as actividades existentes
	3		O Sistema lista todas as actividades
Post-conditions			
Alternative flows and exceptions			
Alternative flows		Actor Input	System Response
2a - Nao existem actividades	1		O Sistema devolve mensagem de aviso.
Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	semmler		
Date	Nov 26, 2009 3:48:54 PM		

Use Case ID	Acerto	
Super Use Case		
Primary Actor		
Secondary Actor(s)		
Brief Description	Calcula o total a pagar ao fornecedor devido ao termino do contracto	
Preconditions	Existencia de contracto	
Flow of Events		System Response
	1	O Sistema verifica o contracto
	2	O Sistema calcula o acerto
	3	O Sistema informa o valor do acerto.
	4	O Sistema devolve o valor do acerto.
Post-conditions		
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	semmler	
Date	Nov 26, 2009 6:21:35 PM	

Use Case ID	Adicionar Contracto	
Super Use Case		
Primary Actor	Condómino	
Secondary Actor(s)		
Brief Description	Condomino pretende criar um novo contracto	
Preconditions		
Flow of Events		System Response
	1	include: Seleccionar actividade
	2	O Sistema guarda opcao seleccionada
	3	O sistema actualiza fornecedores correspondente a actividade seleccionada.
	4	include: metodo de pagamento
	5	O Sistema guarda opcao seleccionada
	6	O Sistema disponibiliza a informacao total do contrato gerado.
	7	O Sistema pede confirmacao para adicionar. OK
Post-conditions	O Sistema é actualizado com a adicao do novo contracto.	
Alternative flows and exceptions		
Alternative Flow of Events 12a - Condomino cancela criacao de contracto		System Response
	1	O Sistema descarta todas as opcoes guardadas.
	2	Voltar ao passo 1
Alternative Flow of Events 12b - Contrato ja existente		System Response
	1	O Sistema da a informacao ao Condomino
	2	O Sistema descarta todas as opcoes guardadas
	3	Voltar ao passo 1
Non-behavior requirements		

Use Case ID	Metodo de Pagamento		
Super Use Case			
Primary Actor			
Secondary Actor(s)			
Brief Description	Mostra os diversos metodos de pagamento		
Preconditions			
Flow of Events		Actor Input	System Response
	1		O Sistema disponibiliza todos os metodos de pagamento disponiveis
	2	O Condomino selecciona metodo de pagamento pretendido	
	3		O Sistema guarda metodo de pagamento seleccionado.
	4		O Sistema calcula valor a pagar.
	5		O Sistema mostra informacao do valor. OK
	6		O Sistema devolve o metodo de pagamento seleccionado
Post-conditions			
Exception: 1 Nao existem metodos de pagamento disponiveis	1	Actor Input	System Response
			O Sistema da a informacao ao Condomino
Alternative flows and exceptions			
Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	semmler		
Date	Nov 26, 2009 6:32:30 PM		

Use Case ID	Remover Contracto		
Super Use Case			
Primary Actor	Condómino		
Secondary Actor(s)			
Brief Description	O condomino pretende remover um contracto		
Preconditions			
Flow of Events		Actor Input	System Response
	1	Include: seleccionar contracto	
	2		O Sistema verifica o contrato seleccionado
	3		include: Acerto
	4		O Sistema mostra a informacao de remocao de contracto. OK
Post-conditions	O Sistema e actualizado com a remocao do contrato especificado		
Alternative flows and exceptions			
Alternative Flow of Events 4a - Condomino cancela remocao de contracto		Actor Input	System Response
	1		O Sistema descarta todas as opcoes seleccionadas
	2		Voltar ao passo 1
Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	semmler		
Date	Nov 26, 2009 5:51:14 PM		

Use Case ID	Seleccionar Contracto	
Super Use Case		
Primary Actor		
Secondary Actor(s)		
Brief Description	Lista de contractos disponiveis para selecao	
Preconditions		
Flow of Events		Actor Input
	1	
	2	O Condomino selecciona o contrato pretendido
	3	
	4	
		System Response
		O Sistema disponibiliza todos os contractos activos existentes.
		O Sistema guarda contrato seleccionado. OK
		O Sistema devolve o contrato seleccionado pelo Condomino
Post-conditions		
Alternative flows and exceptions		
Exception: 1 Nao existem contractos disponiveis		Actor Input
	1	
		System Response
		O Sistema dá informacao ao Condomino
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	semmler	
Date	Nov 26, 2009 6:07:56 PM	

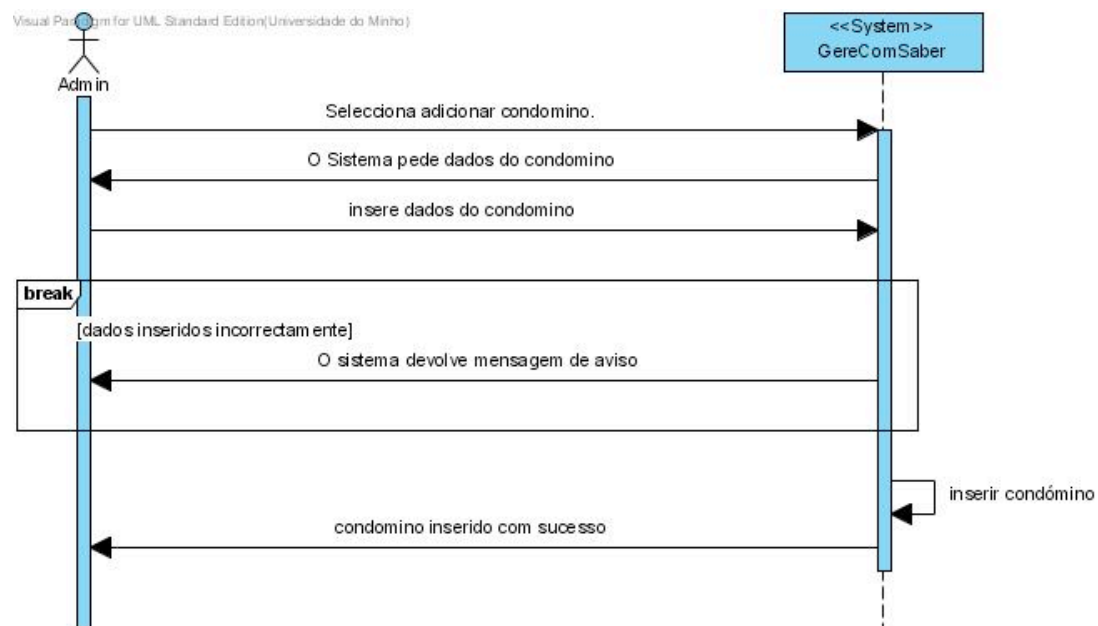
Use Case ID	Listar Contractos	
Super Use Case		
Primary Actor	Condómino	
Secondary Actor(s)		
Brief Description	O Condómino pretende requisitar uma lista de contractos a que esta vinculado.	
Preconditions		
Flow of Events		Actor Input
	1	O condómino selecciona listar contractos .
	2	
	3	O Sistema verifica os contractos existentes com o condómino.
Post-conditions		
Alternative flows and exceptions		
Alternative flows		Actor Input
2a - Nao existem contractos	1	O Sistema devolve mensagem de aviso .
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	semmler	
Date	Nov 26, 2009 2:43:53 PM	

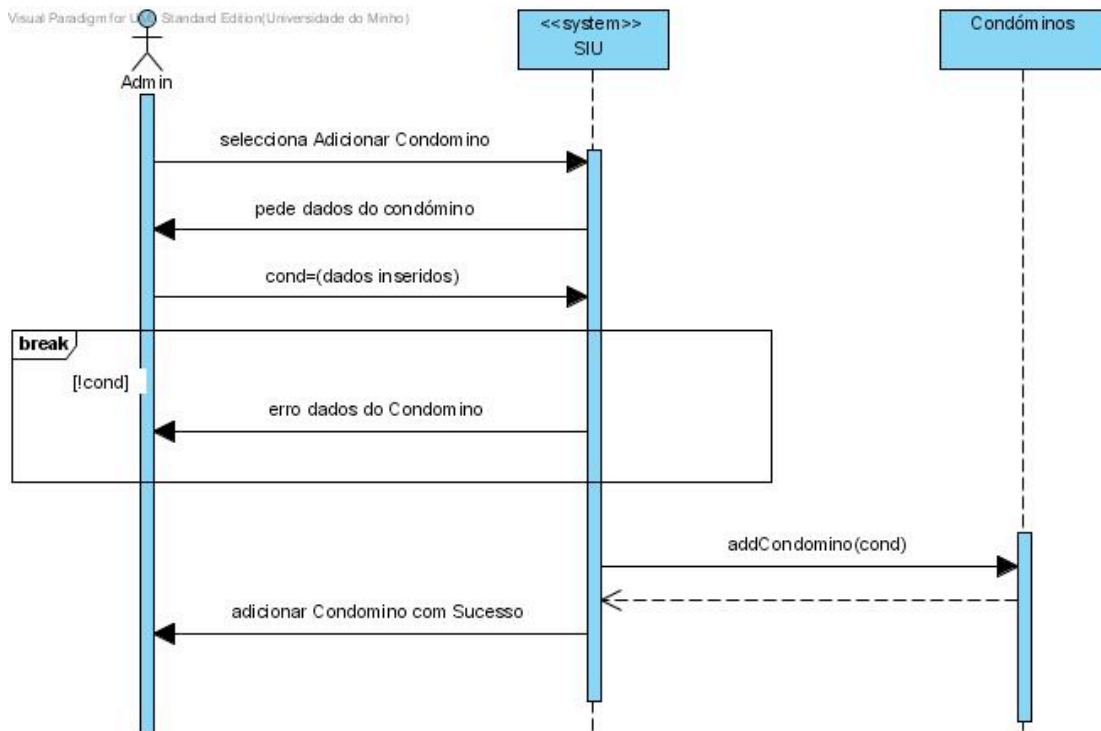
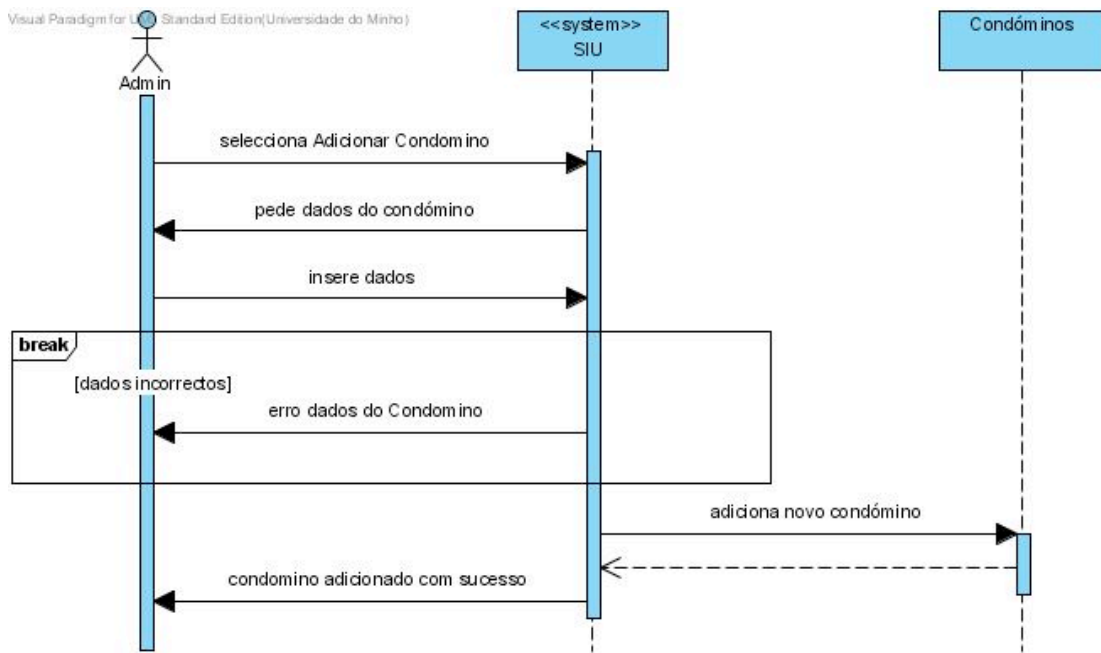
Use Case ID	Consultar serviços	
Super Use Case		
Primary Actor	Fornecedor	
Secondary Actor(s)		
Brief Description	O fornecedor pretende consultar os serviços existentes no sistema	
Preconditions	Existirem serviços no sistema	
Flow of Events		Actor Input
	1	Fornecedor solicita serviços
	2	
Post-conditions		
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Fábio	
Date	28/Nov/2009 18:19:47	

Use Case ID	Ver balanço	
Super Use Case		
Primary Actor	Fornecedor	
Secondary Actor(s)		
Brief Description	O fornecedor pretende consultar o seu balanço	
Preconditions		
Flow of Events		Actor Input
	1	O Fornecedor pede o balanço
	2	
	3	
		System Response
		O sistema calcula o balanço
		O sistema imprime
Post-conditions		
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Fábio	
Date	28/Nov/2009 18:41:39	

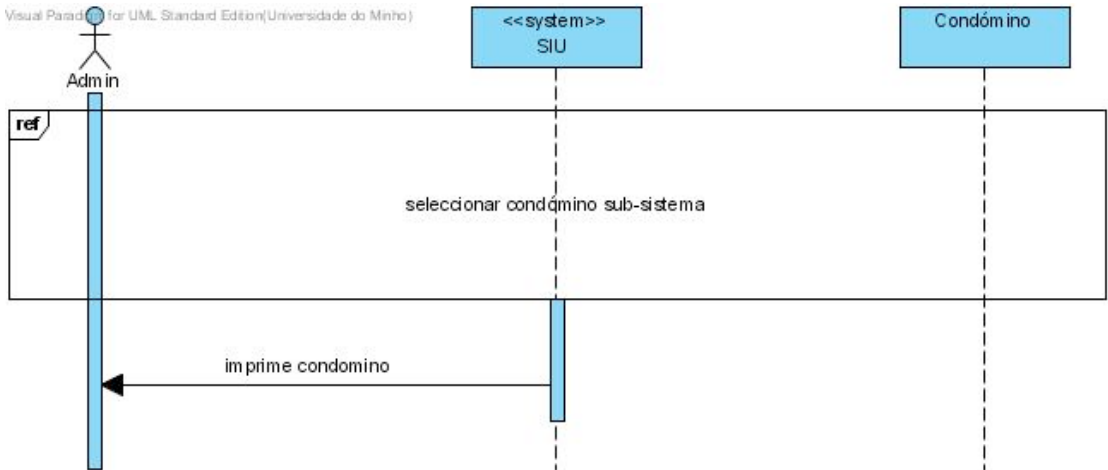
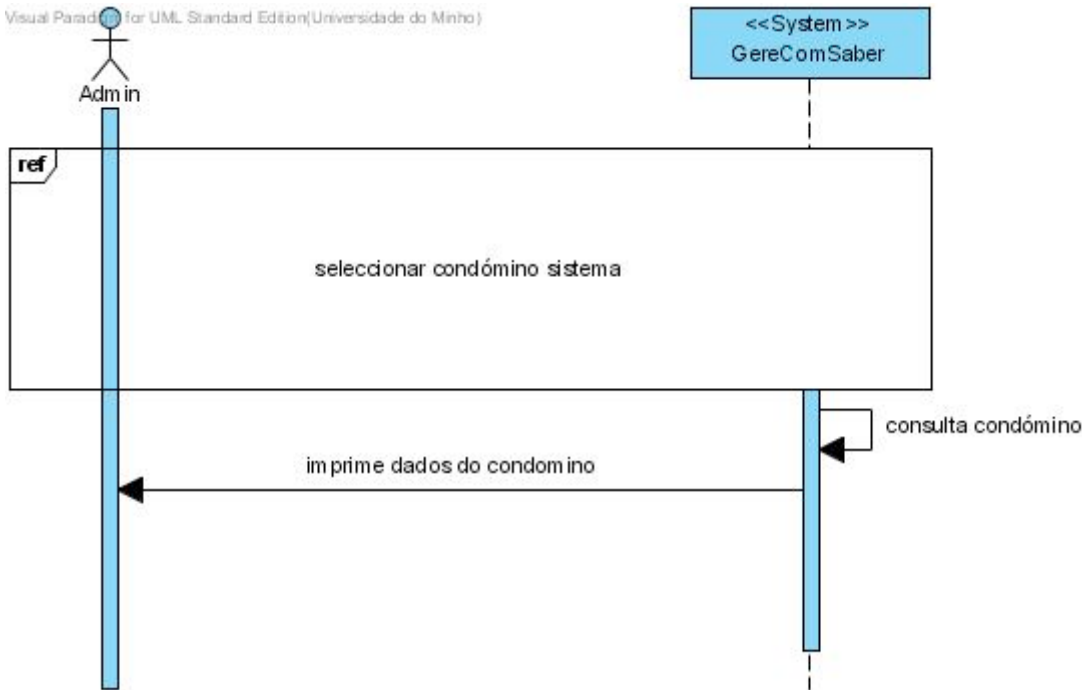
7. Diagramas de Sequência

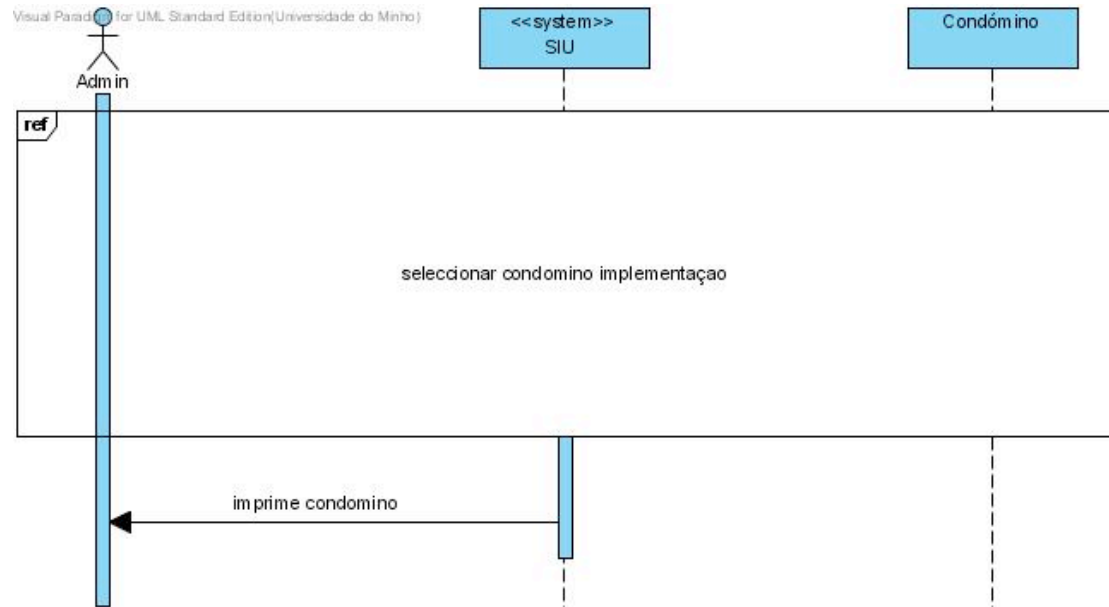
Diagramas Adicionar Condómino



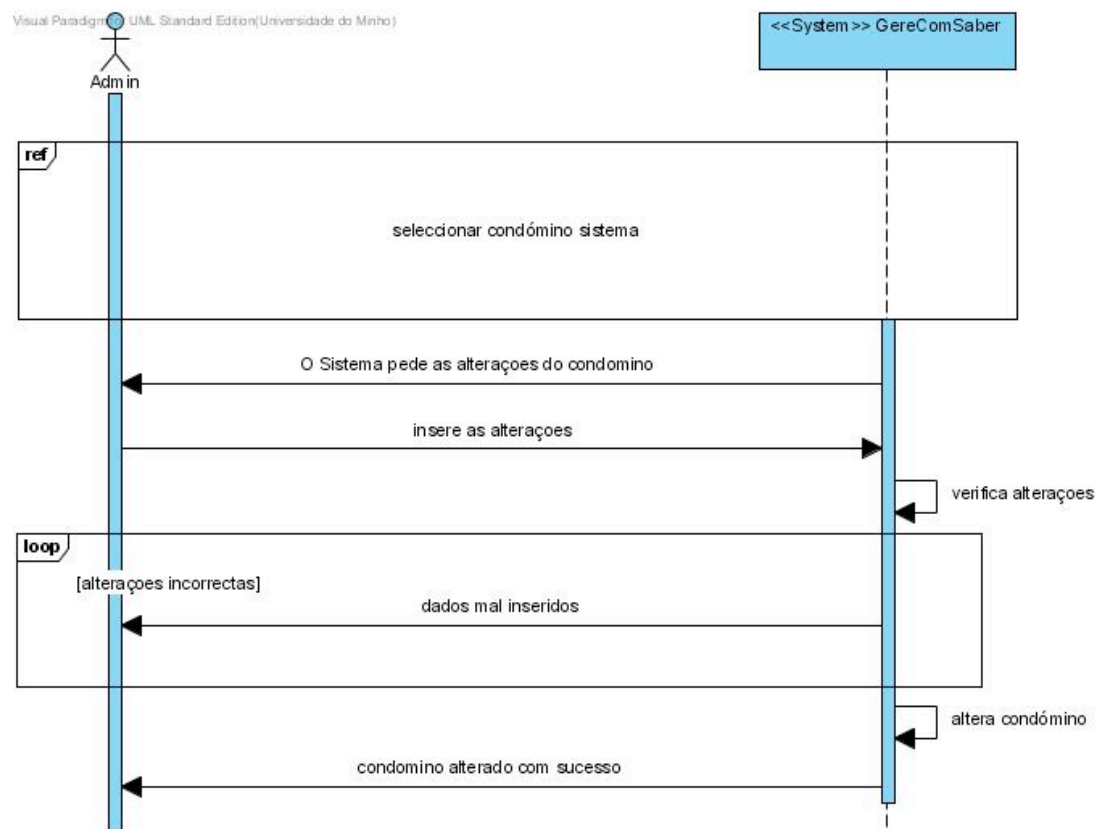


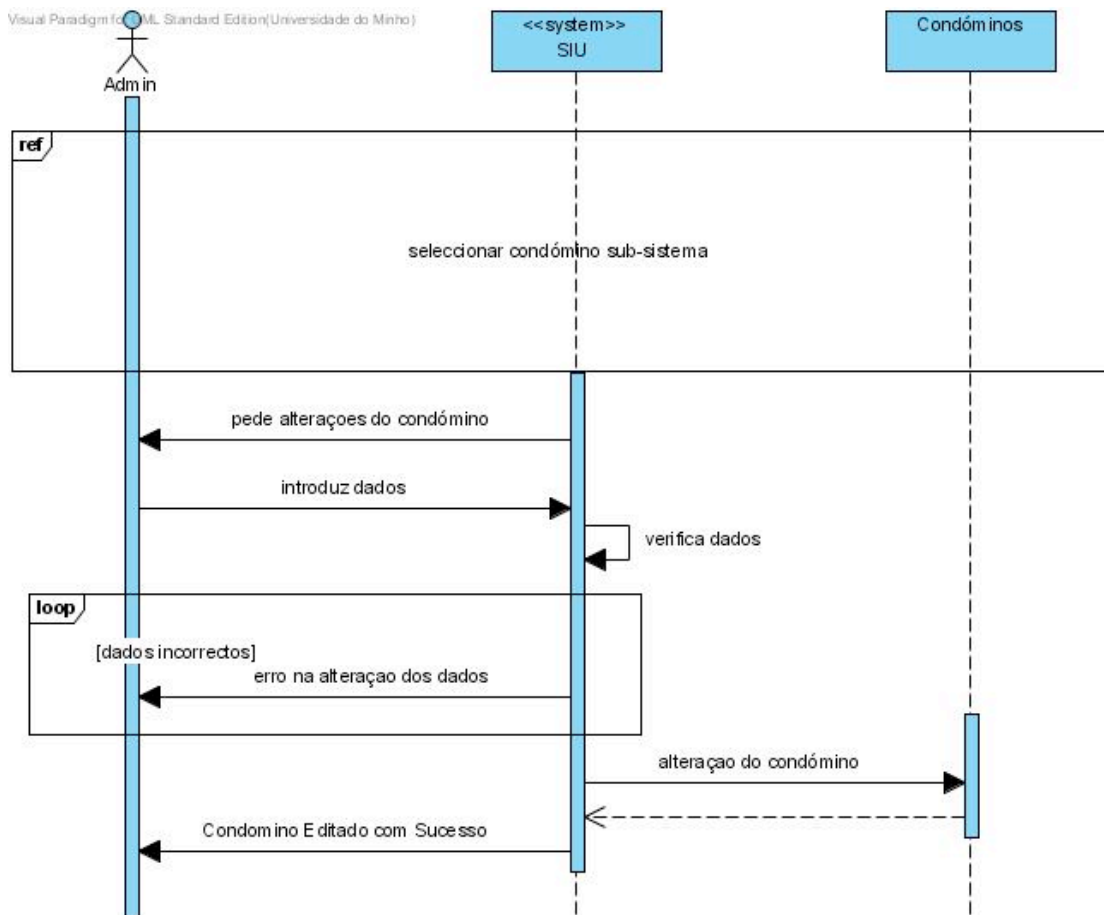
Diagramas Consultar Condómino

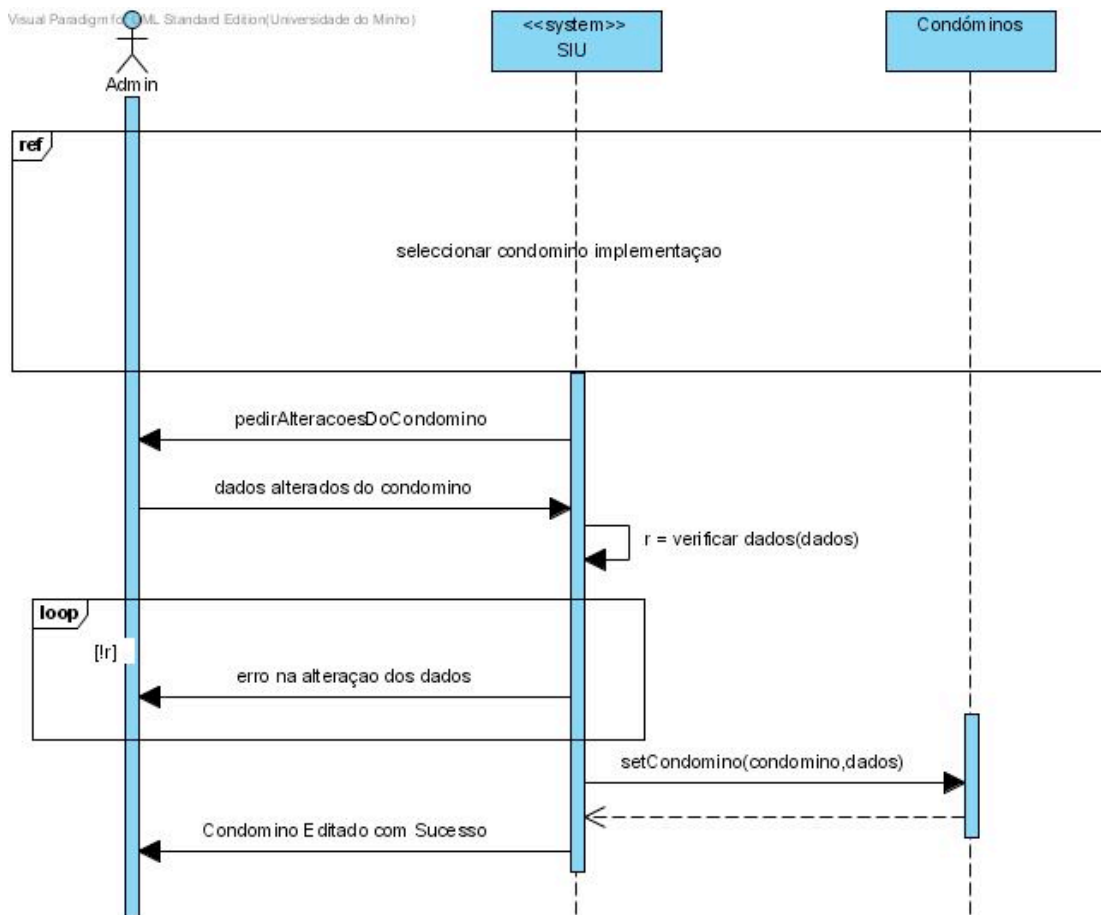




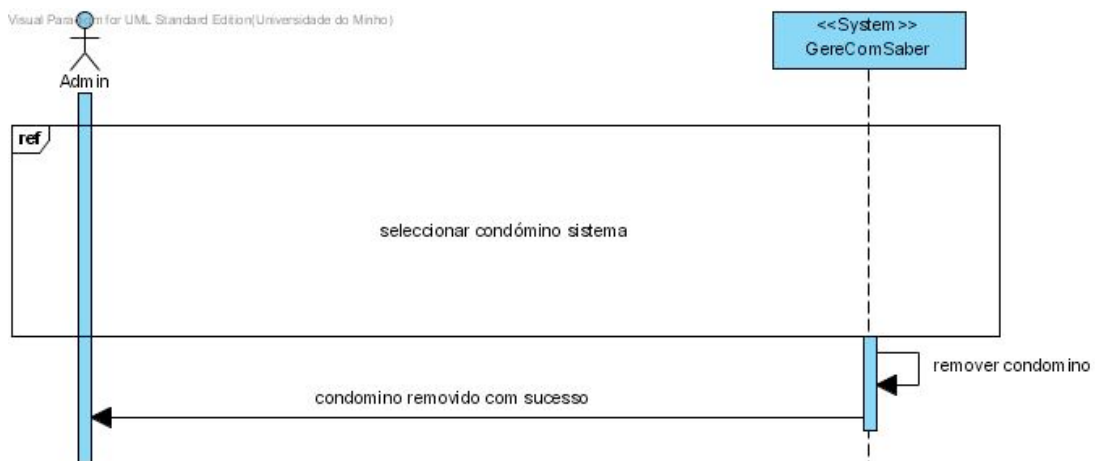
Diagramas Editar Condomínio

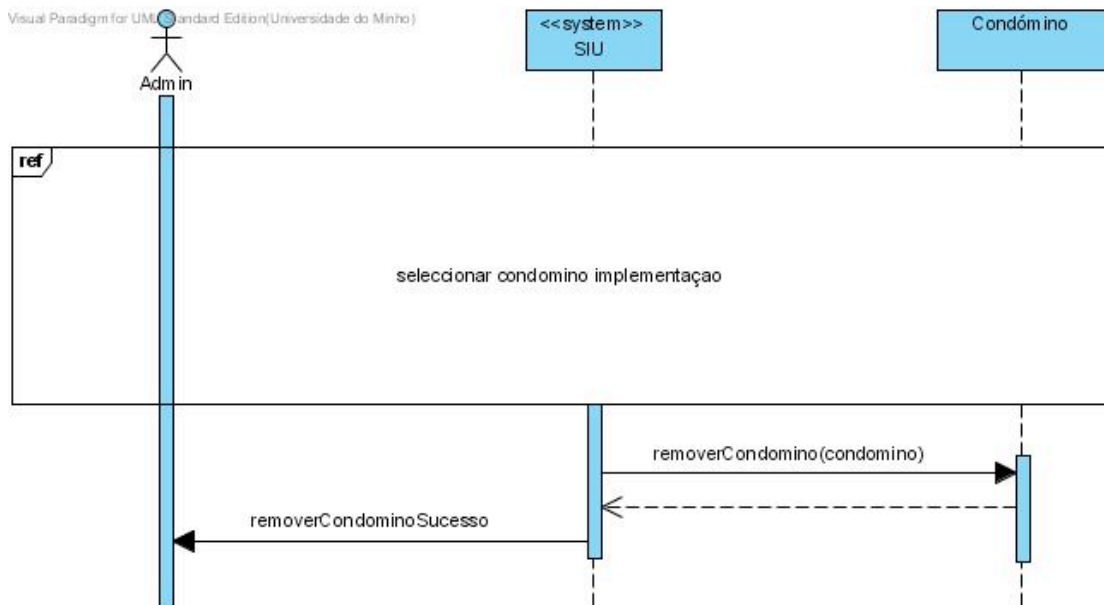
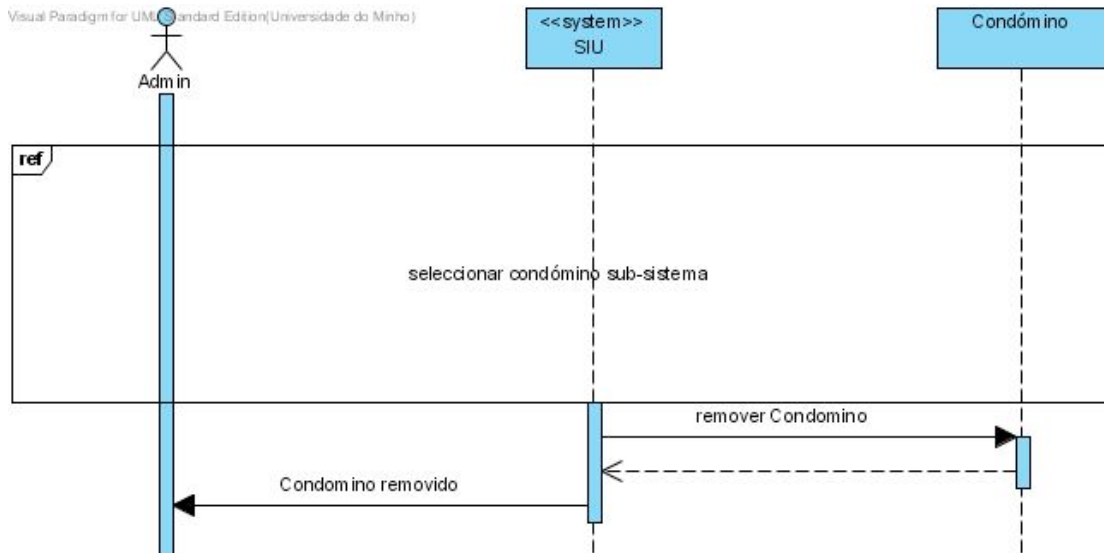




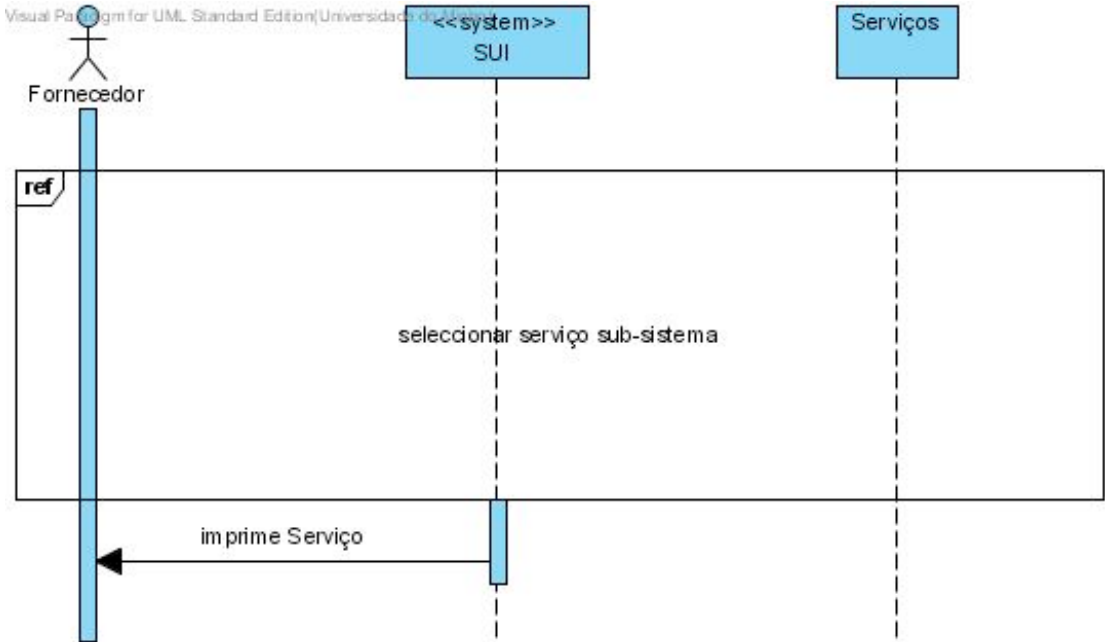
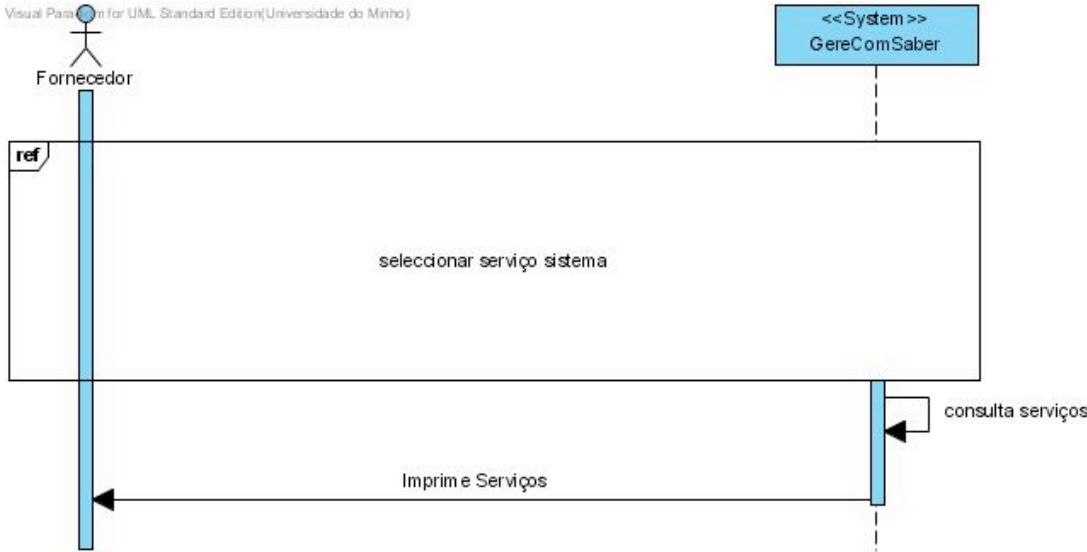


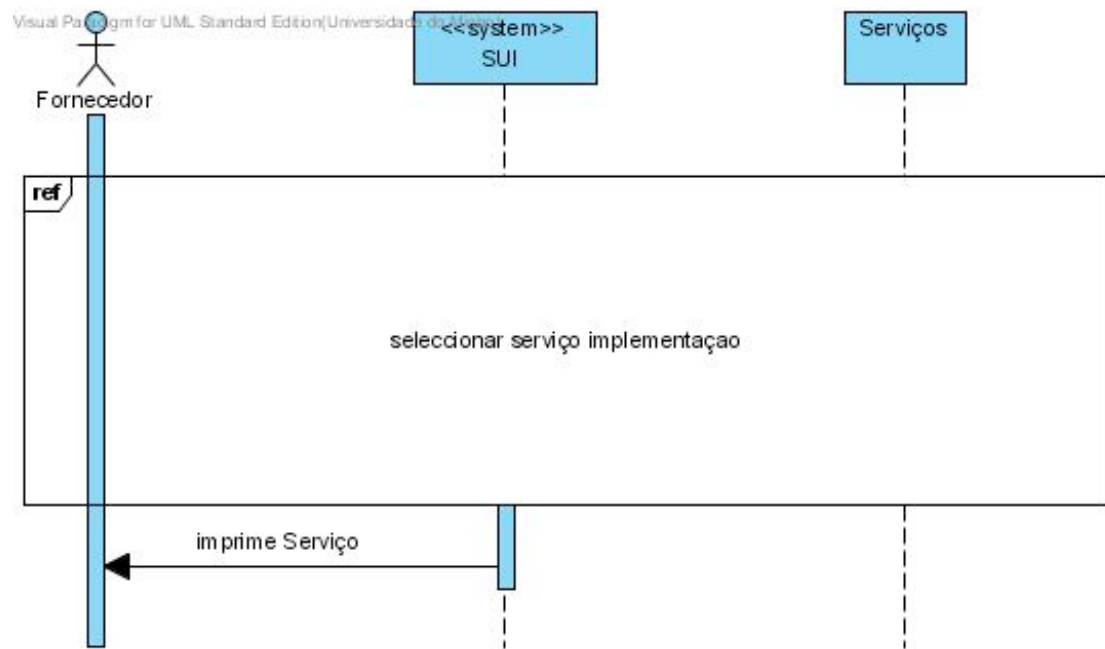
Diagramas Remove Condomínio



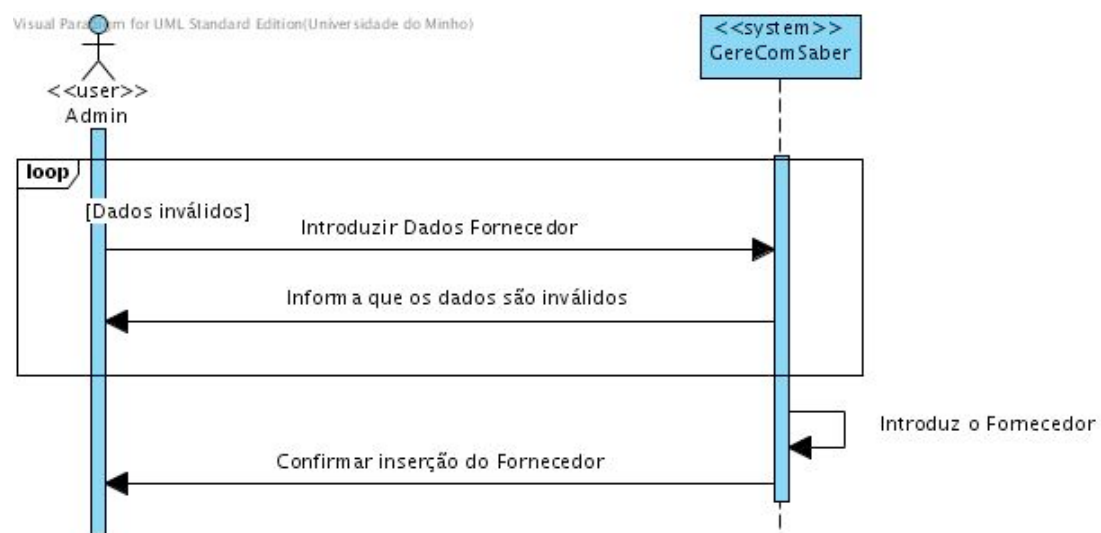


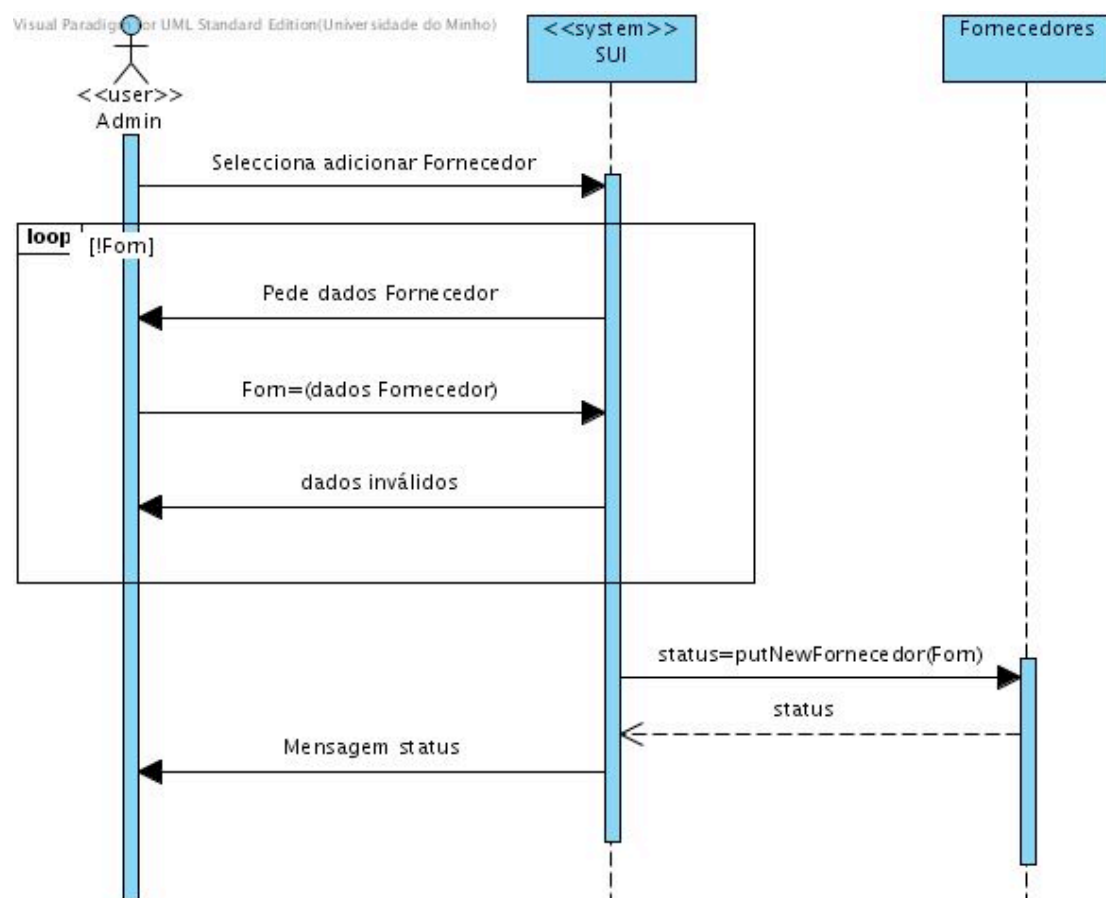
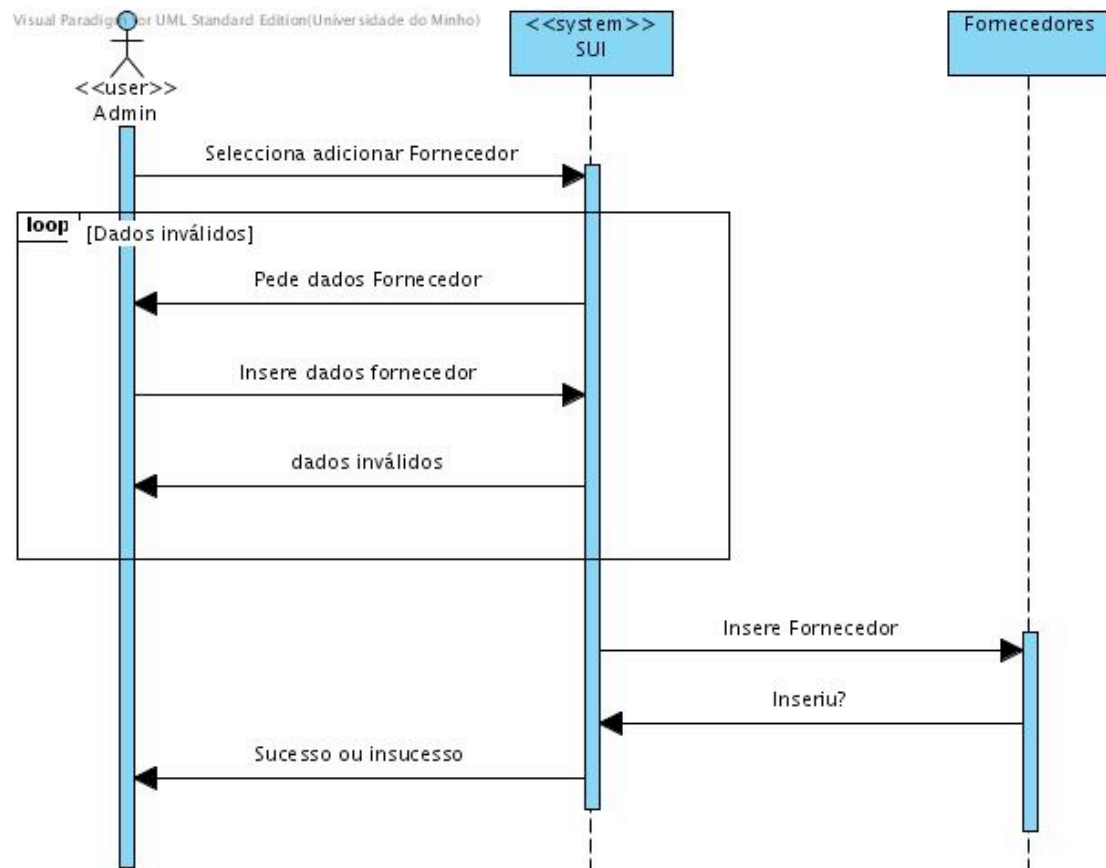
Diagramas Consultar Serviço



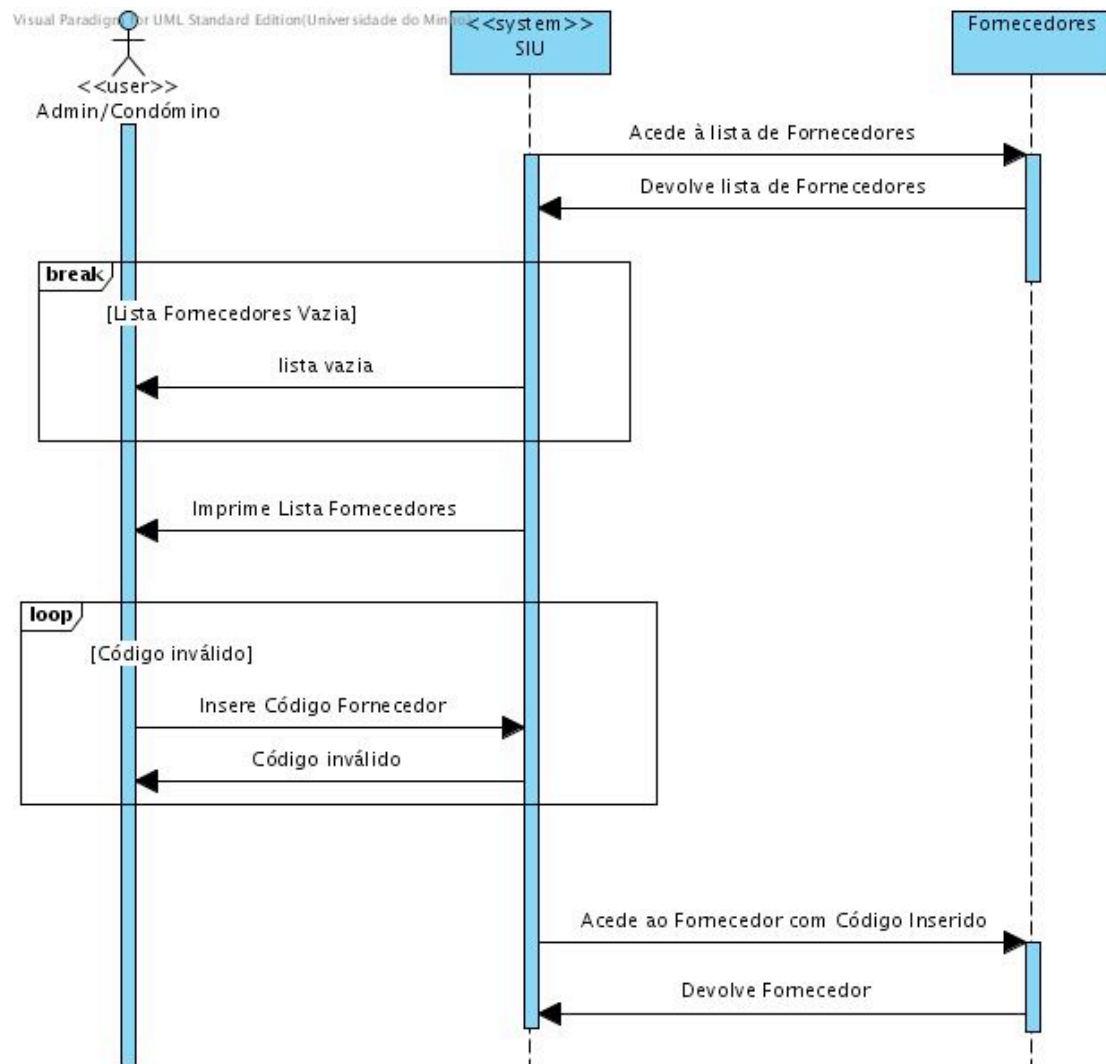
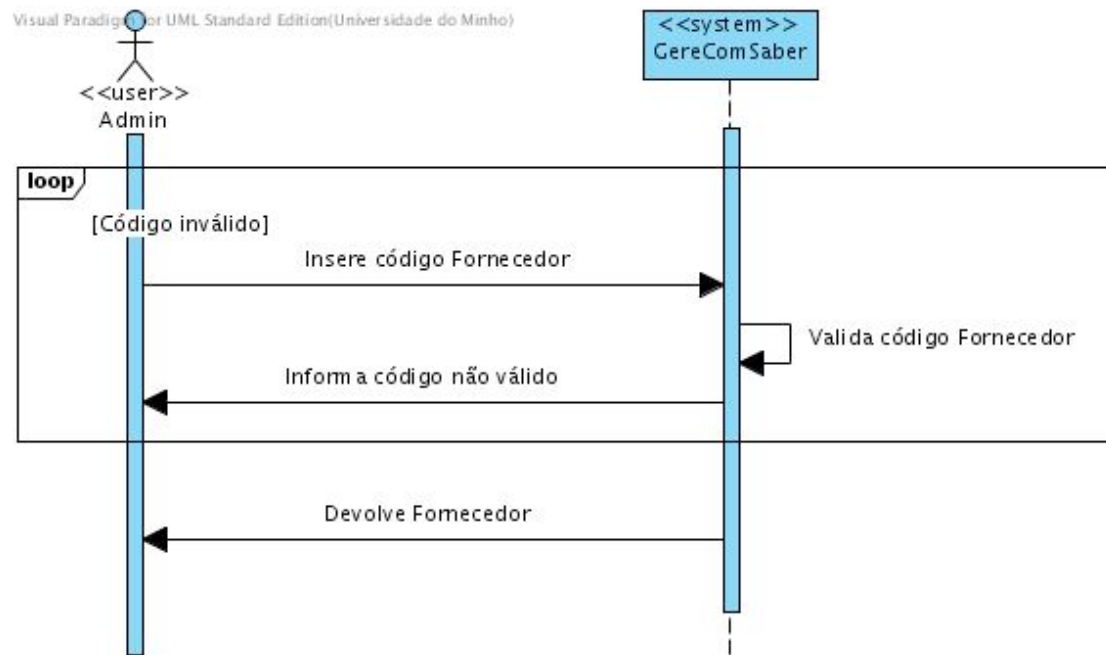


Diagramas Adicionar Fornecedor





Diagramas Seleccionar Fornecedor



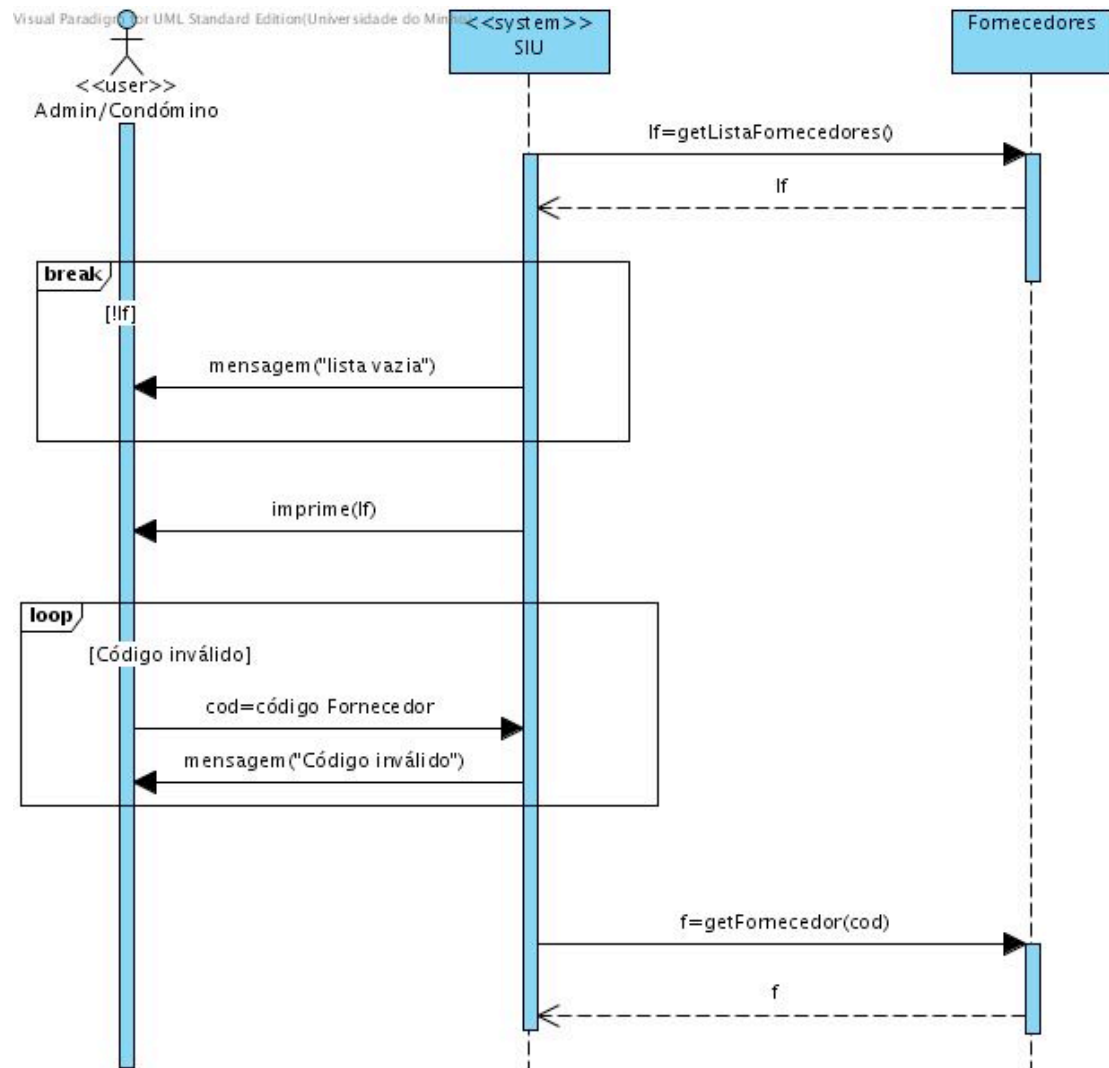
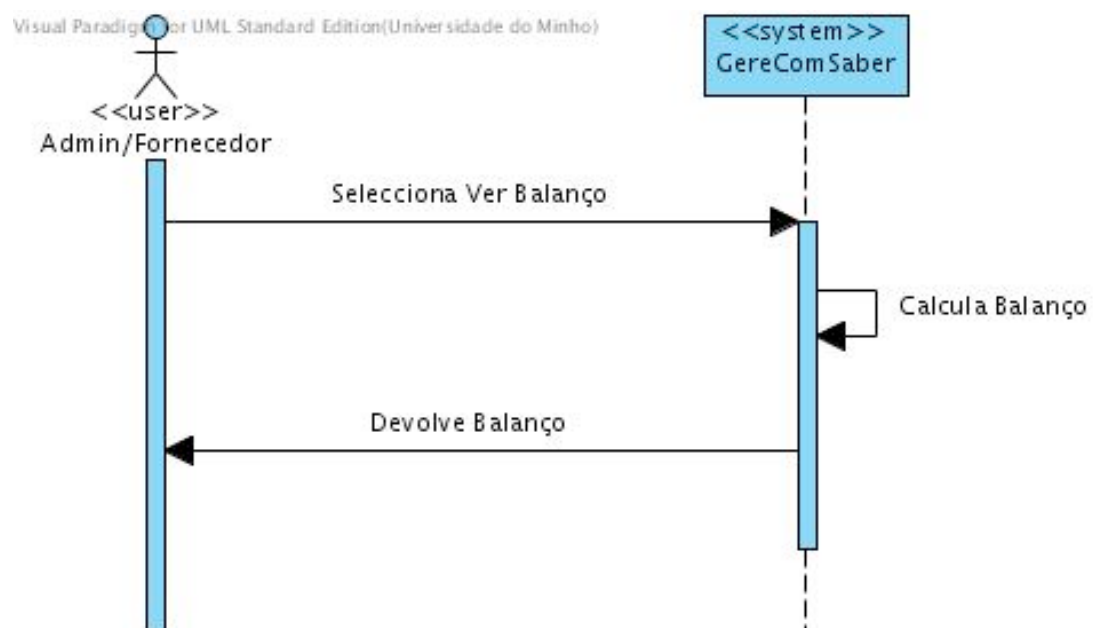
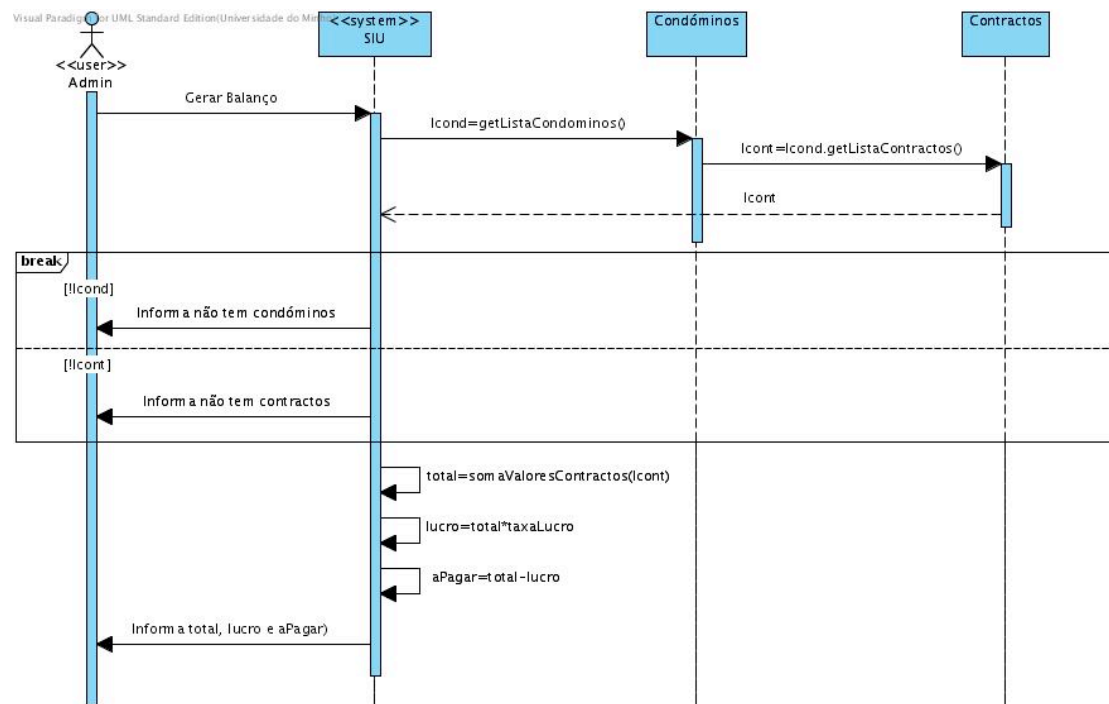
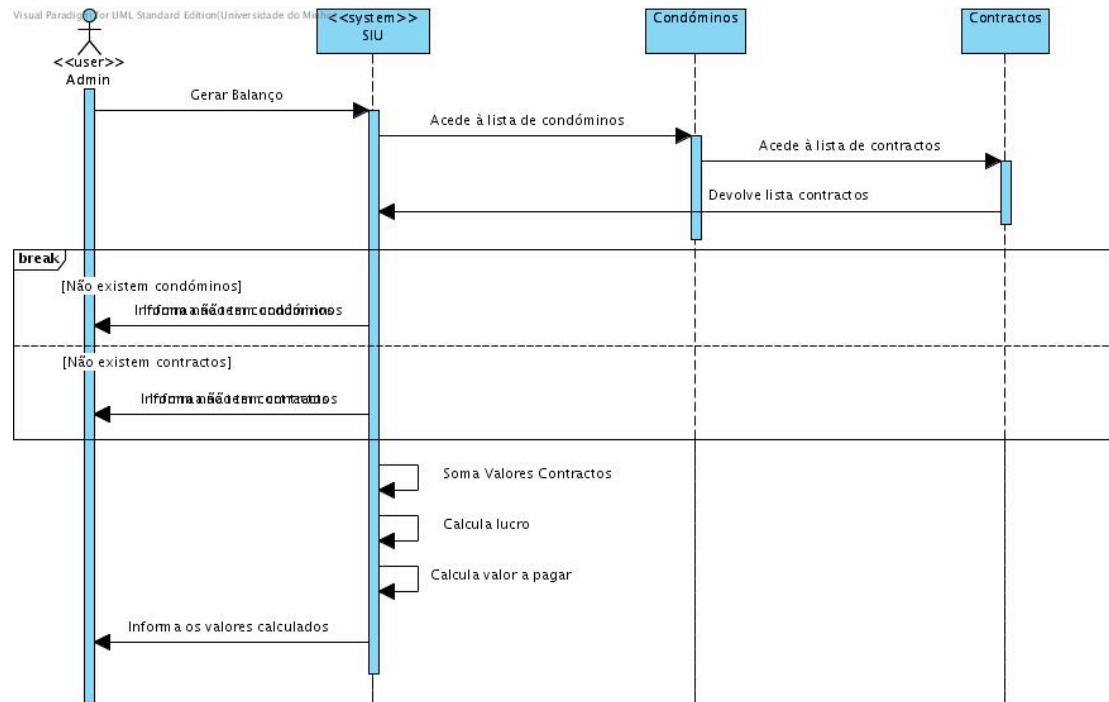


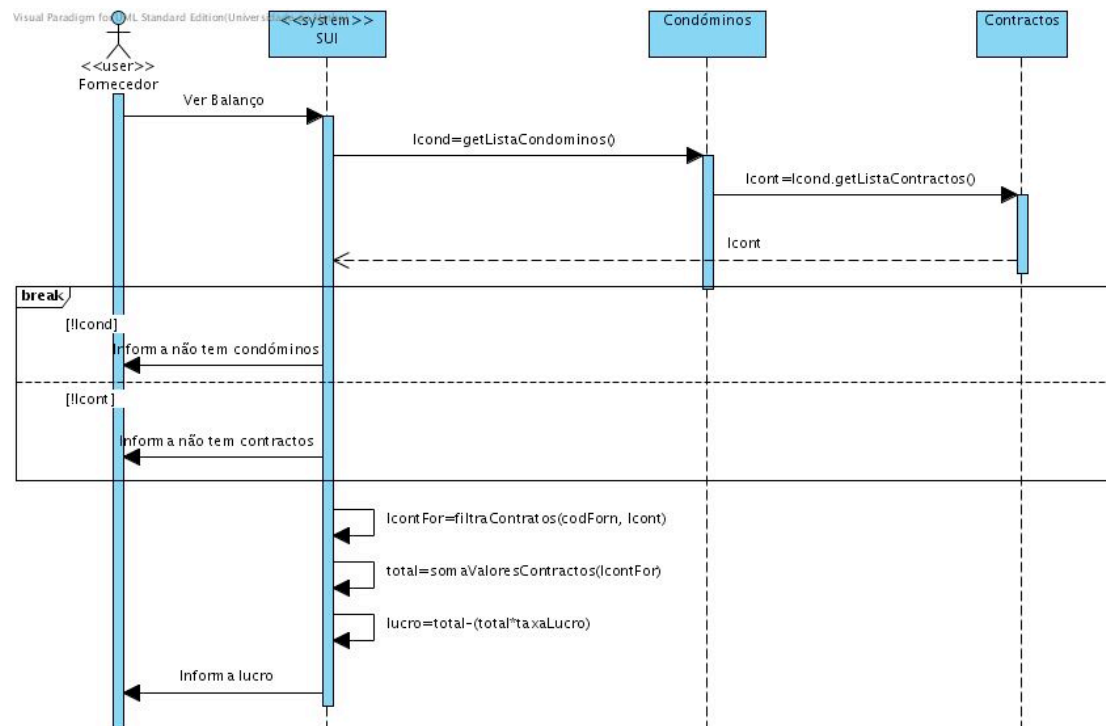
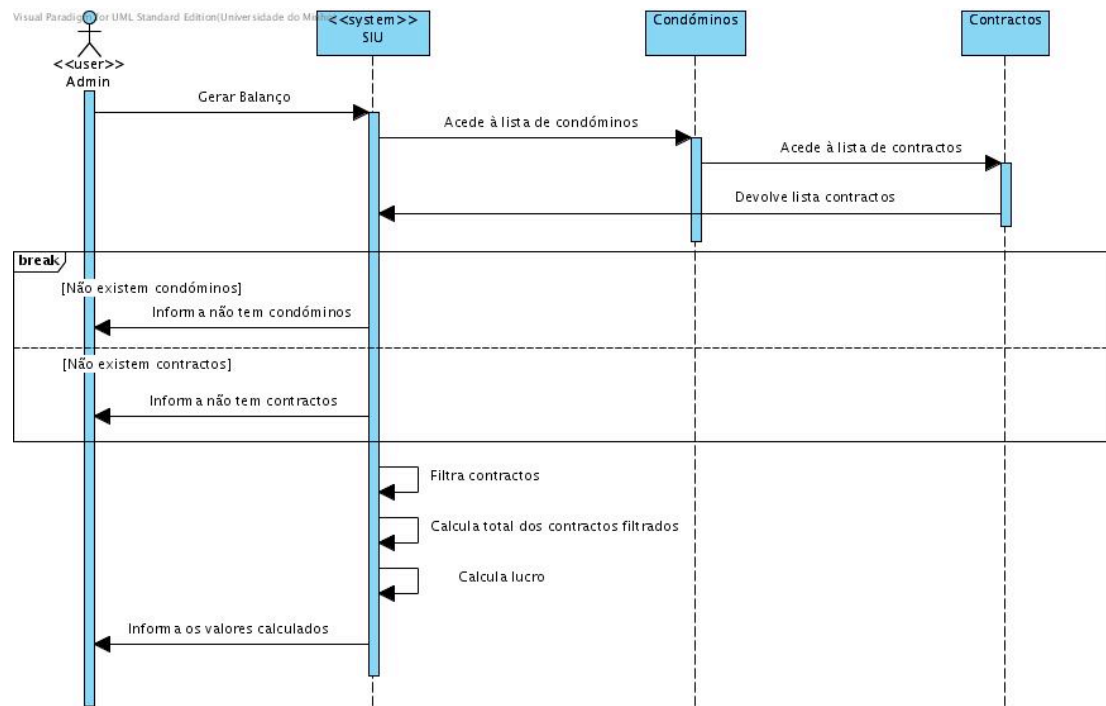
Diagrama Balanço do Sistema



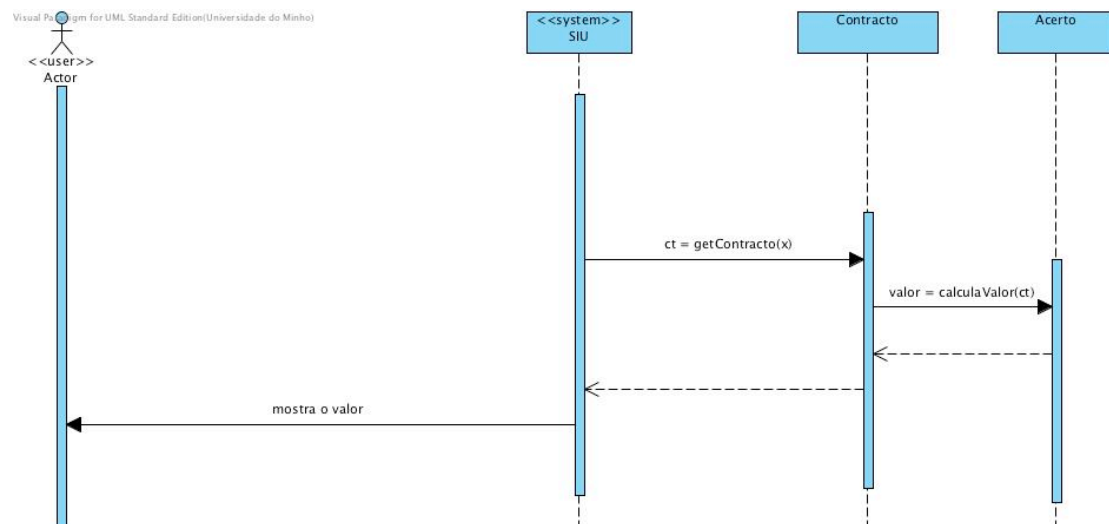
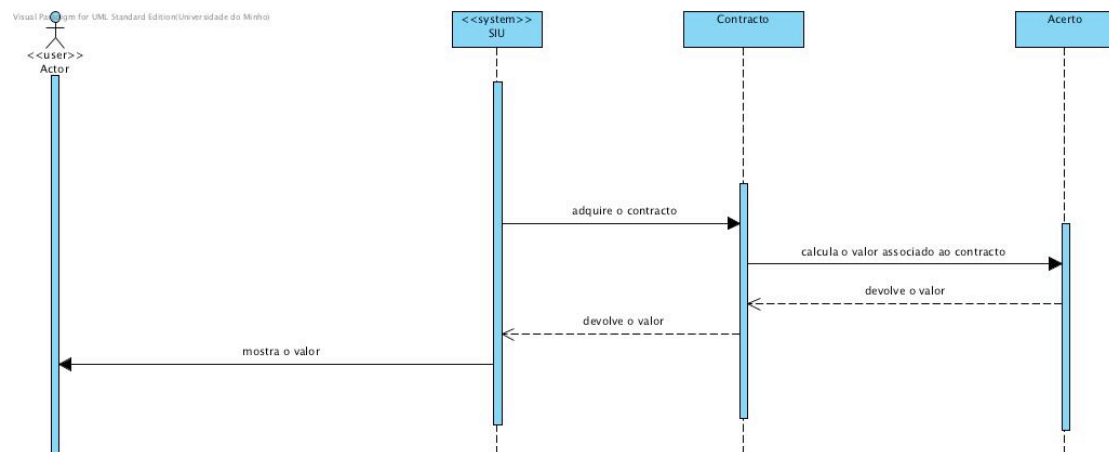
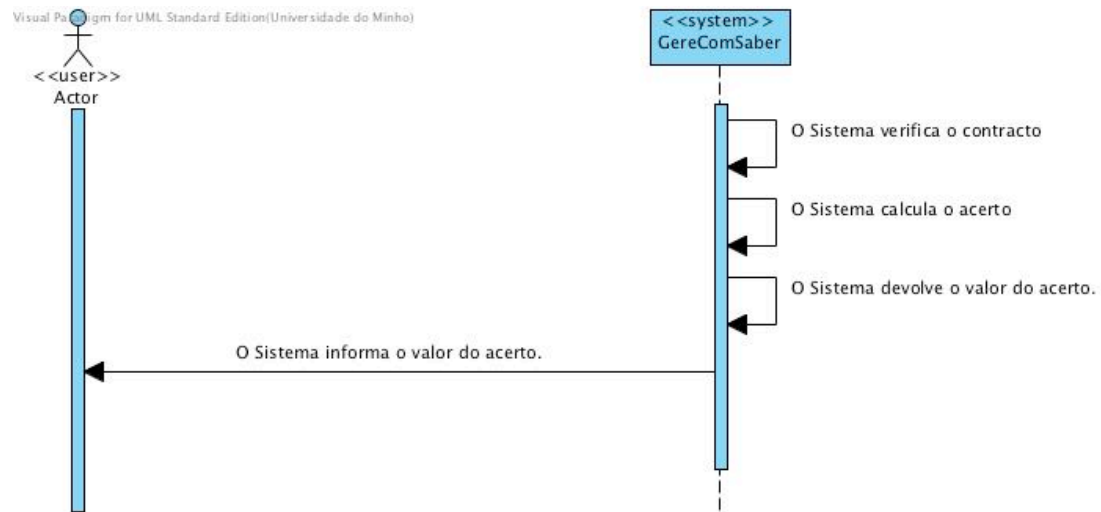
Diagramas Balanço do Admin



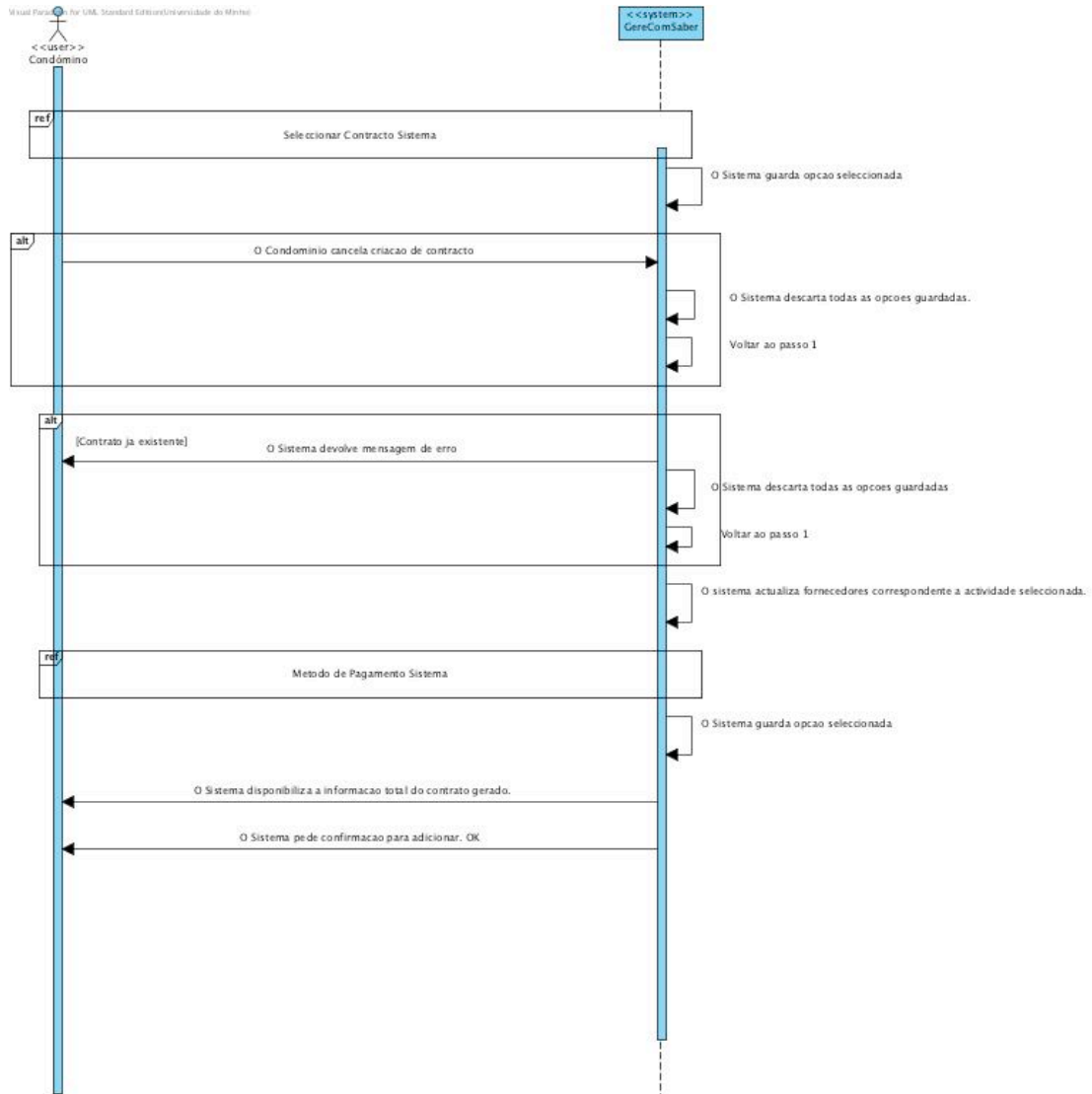
Diagramas Balanço Fornecedor

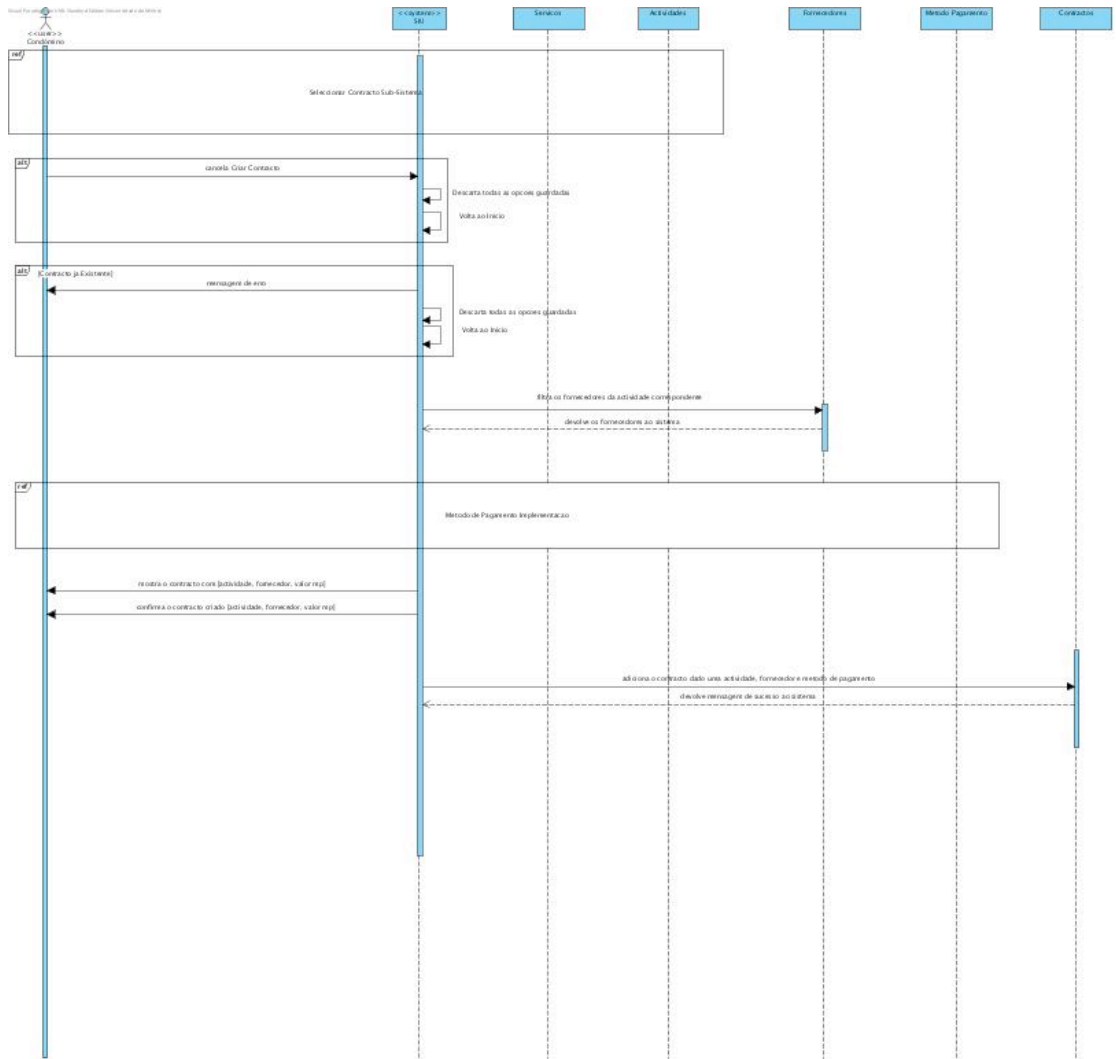


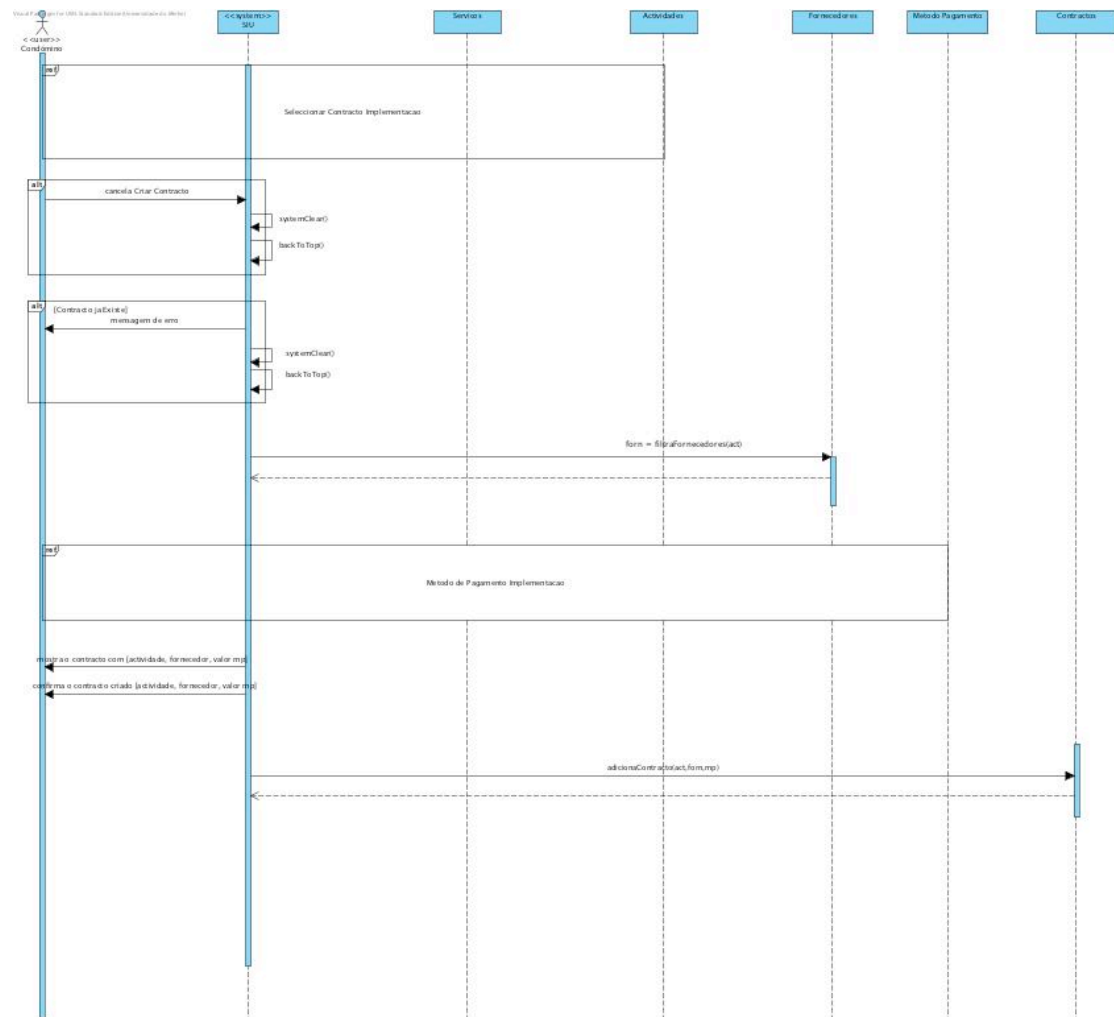
Diagramas Acerto



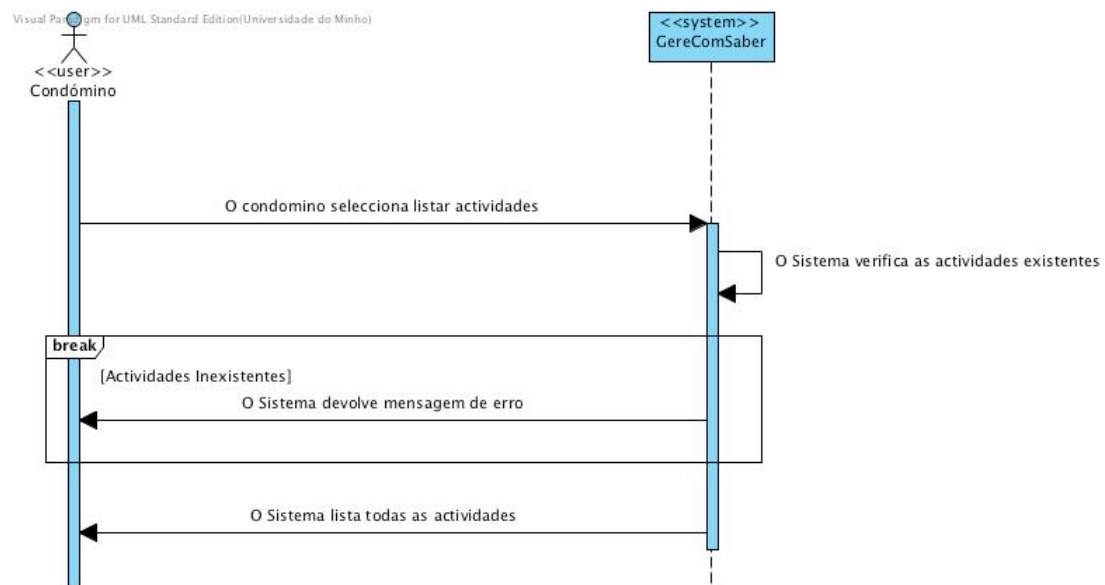
Diagramas Adicionar Contracto

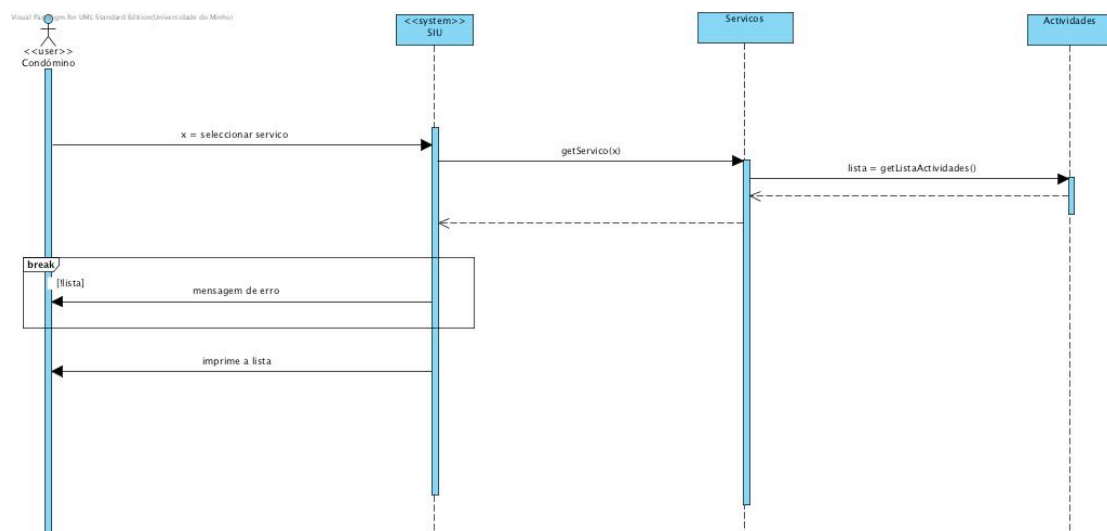
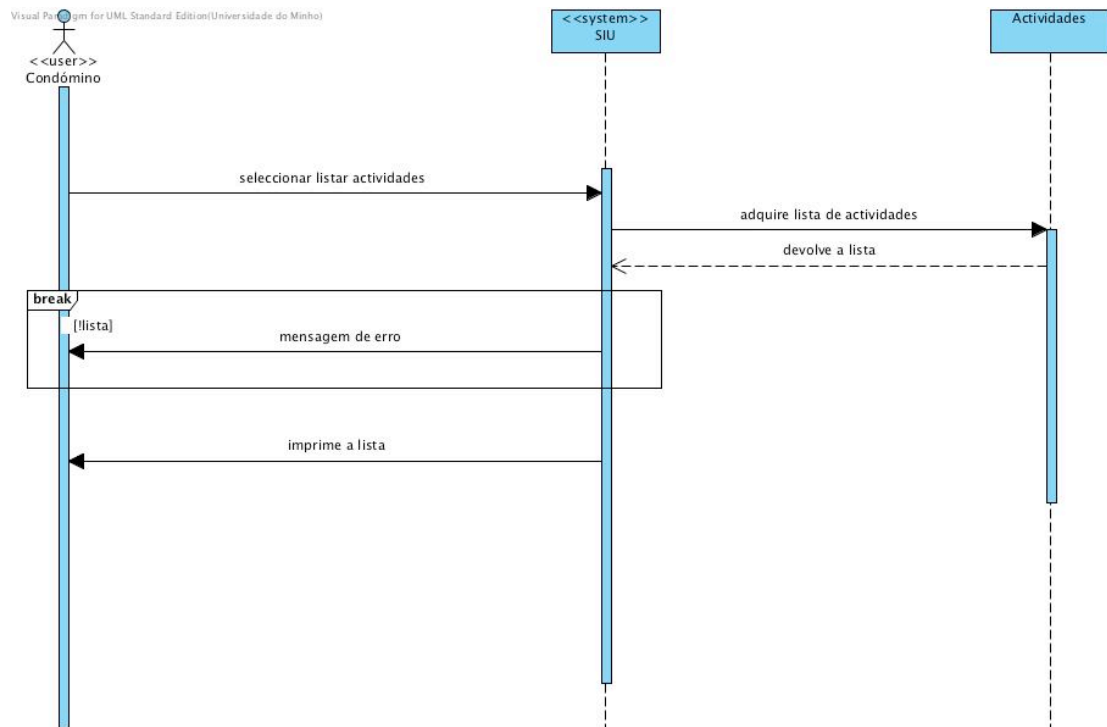




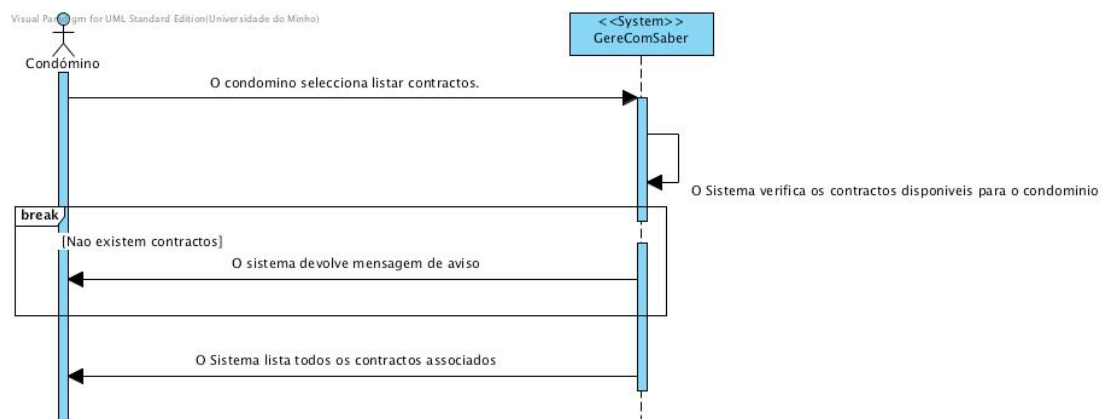


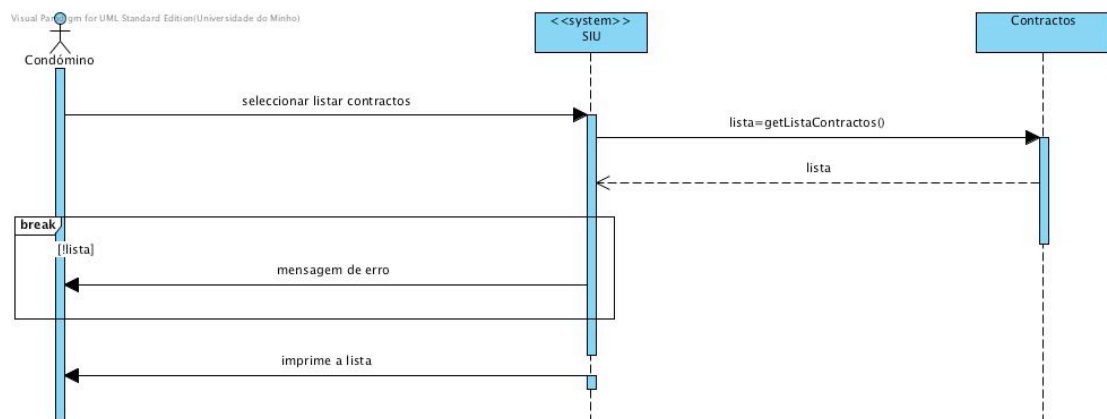
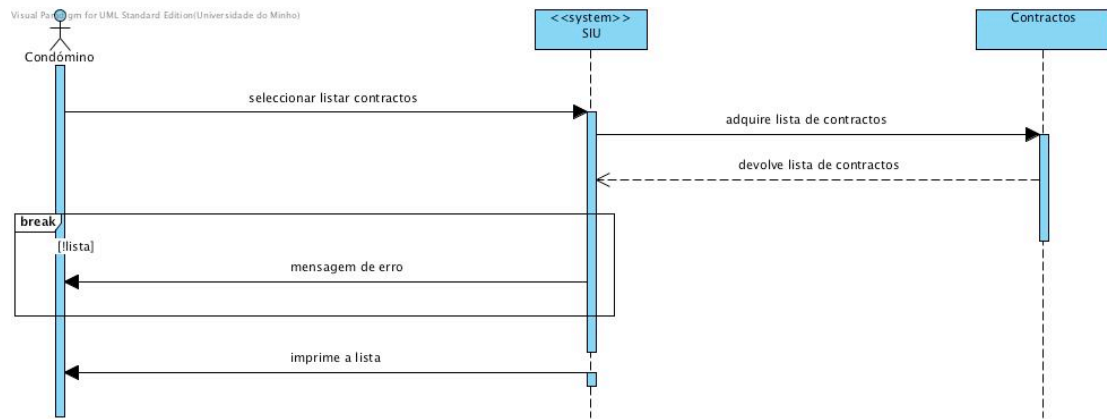
Diagramas Listar Actividades



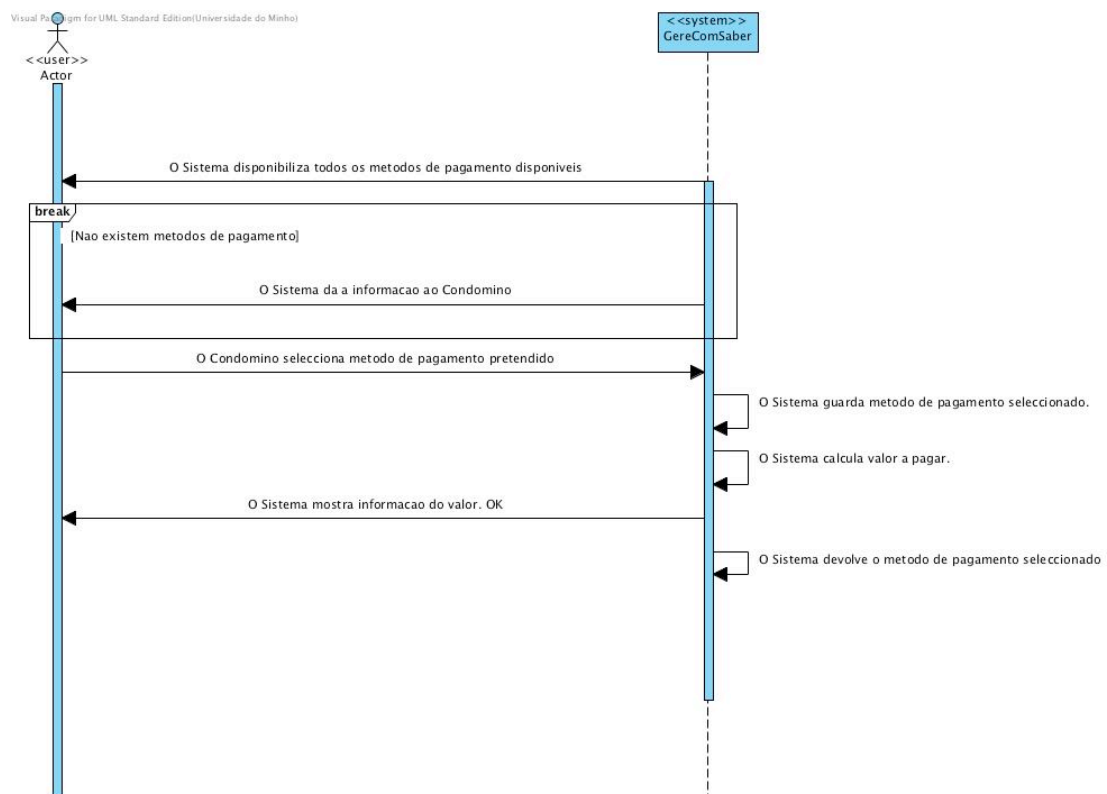


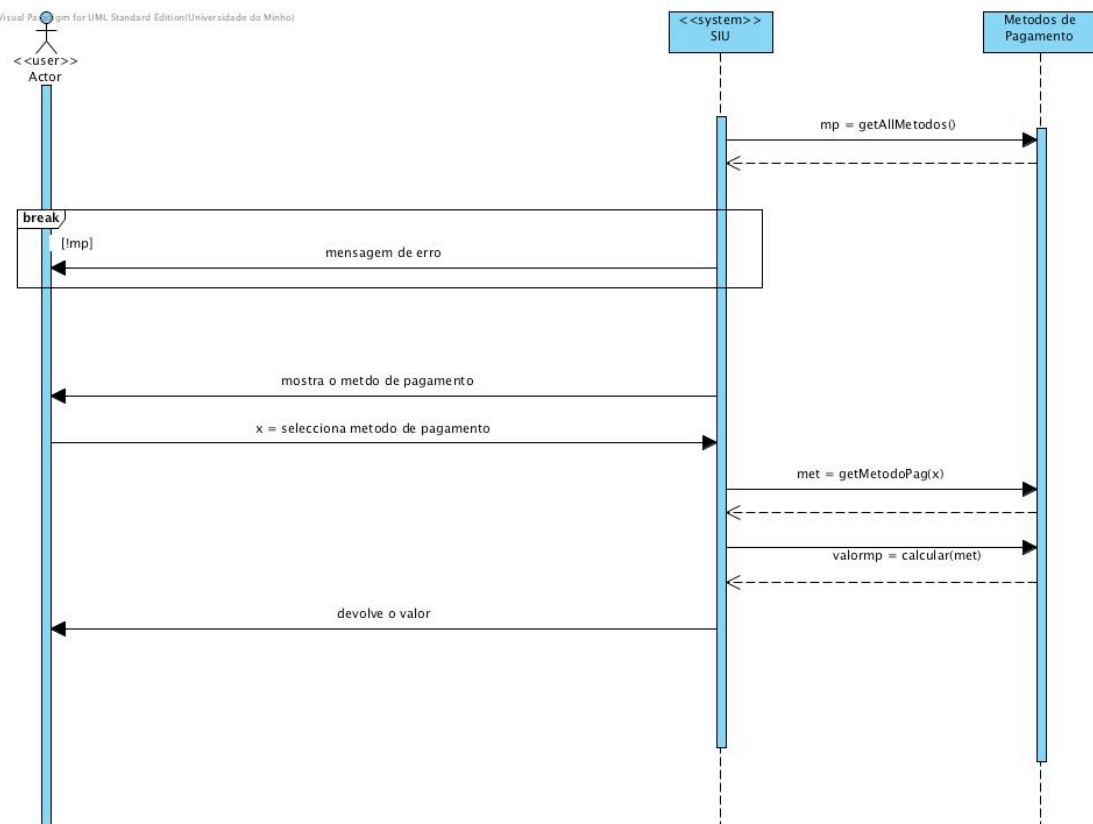
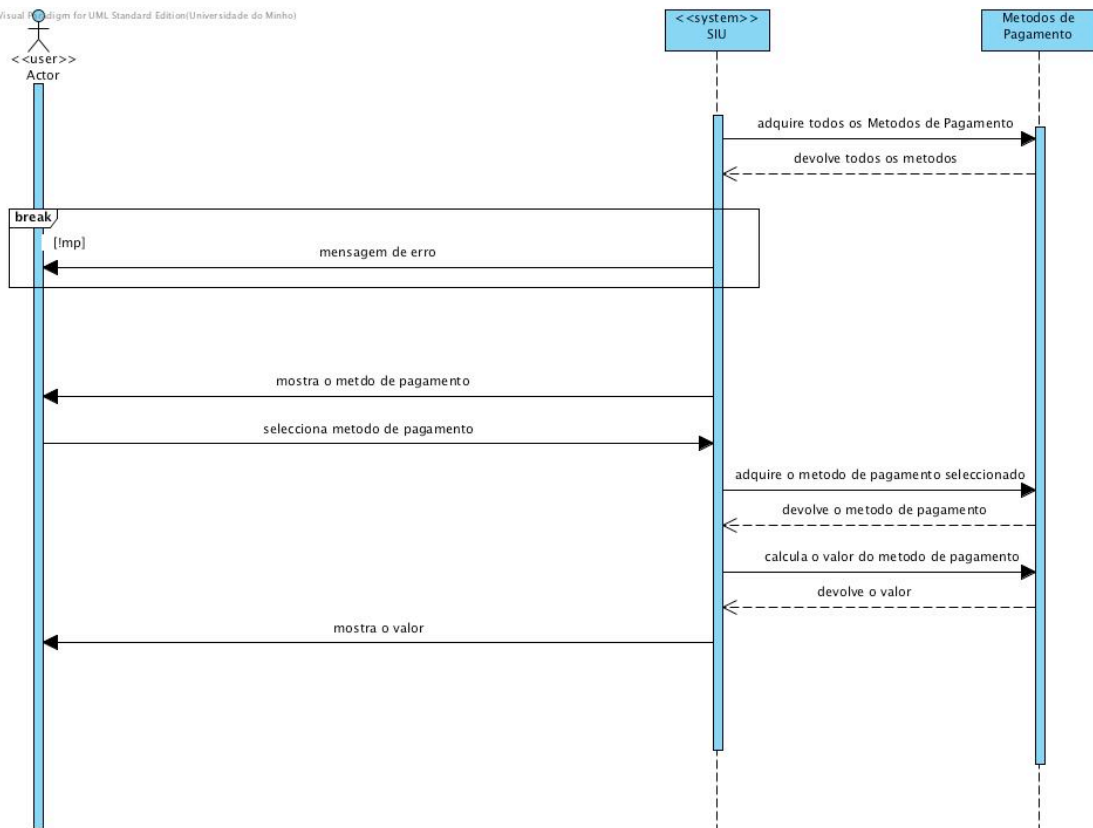
Diagramas Listar Contractos



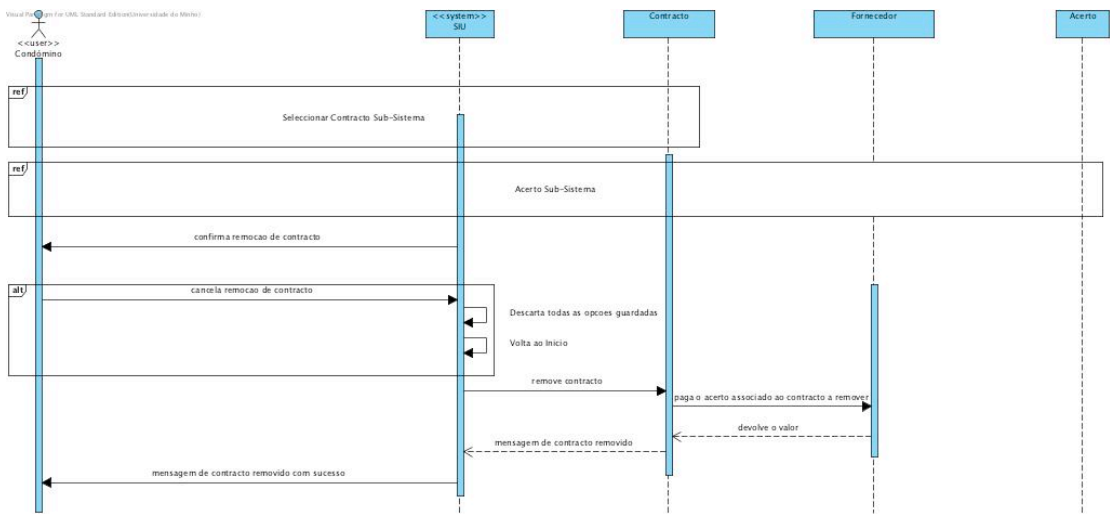
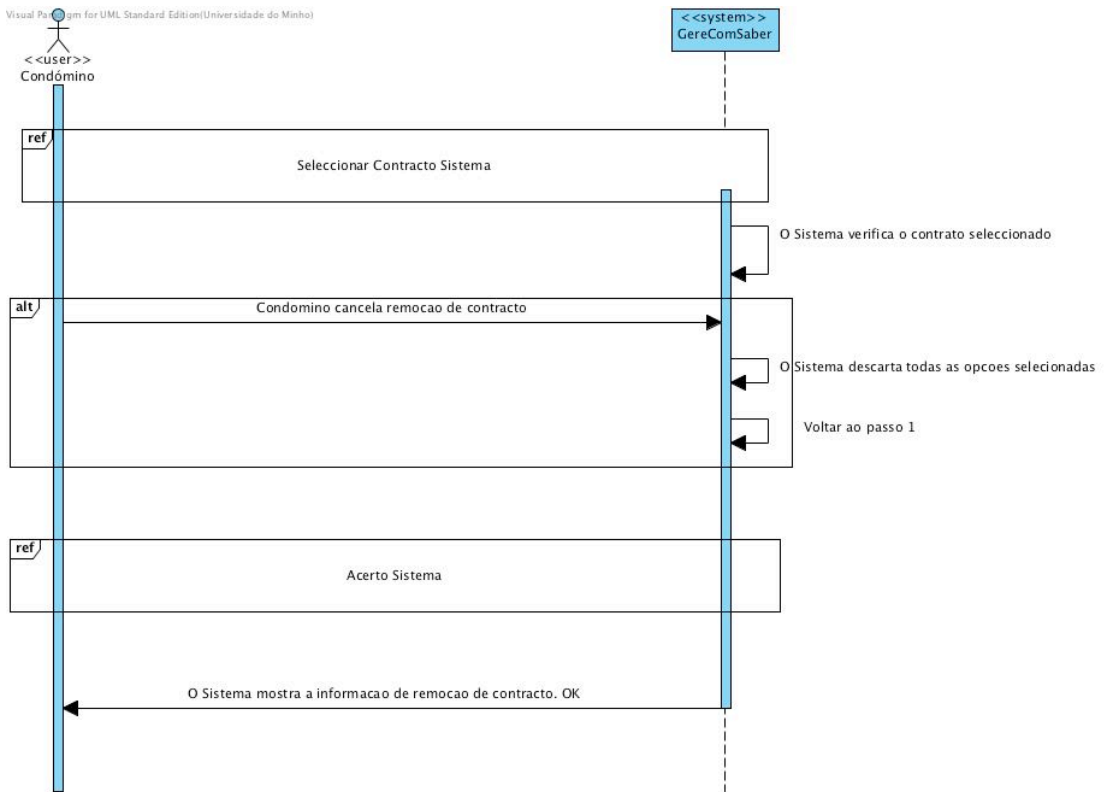


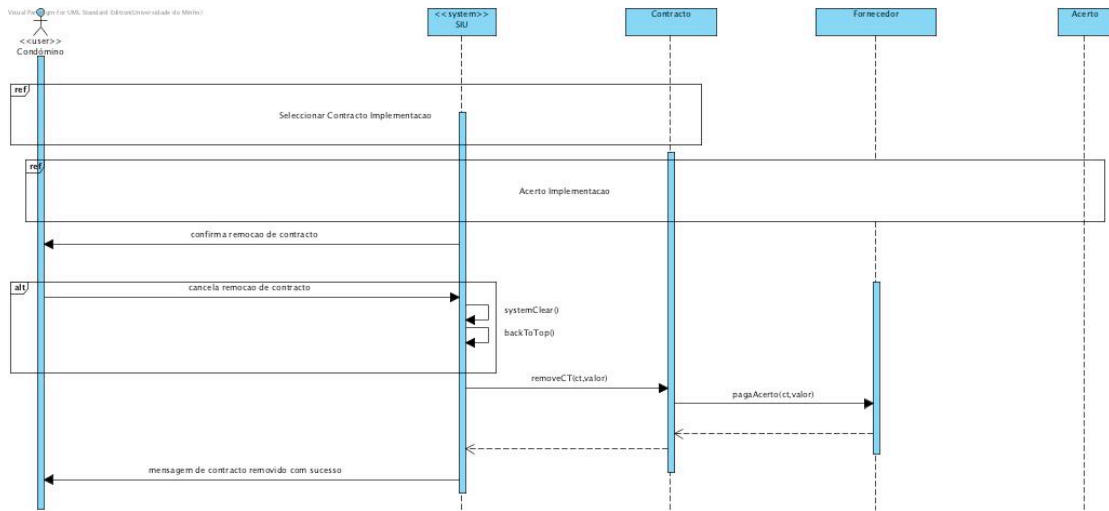
Diagramas Método de Pagamento



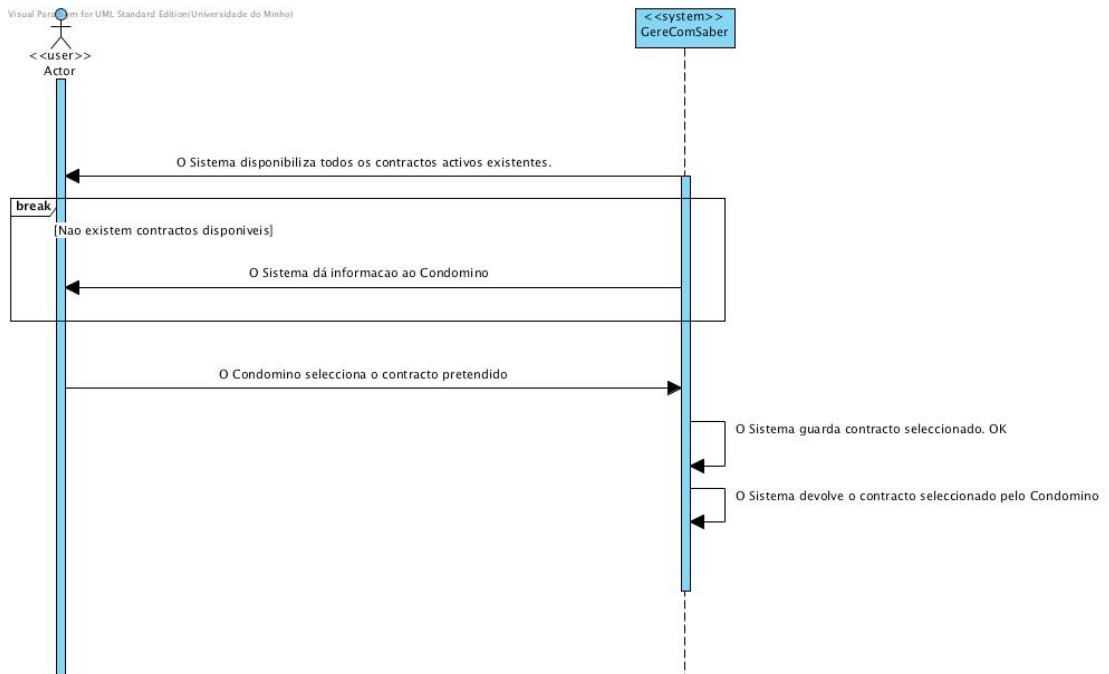


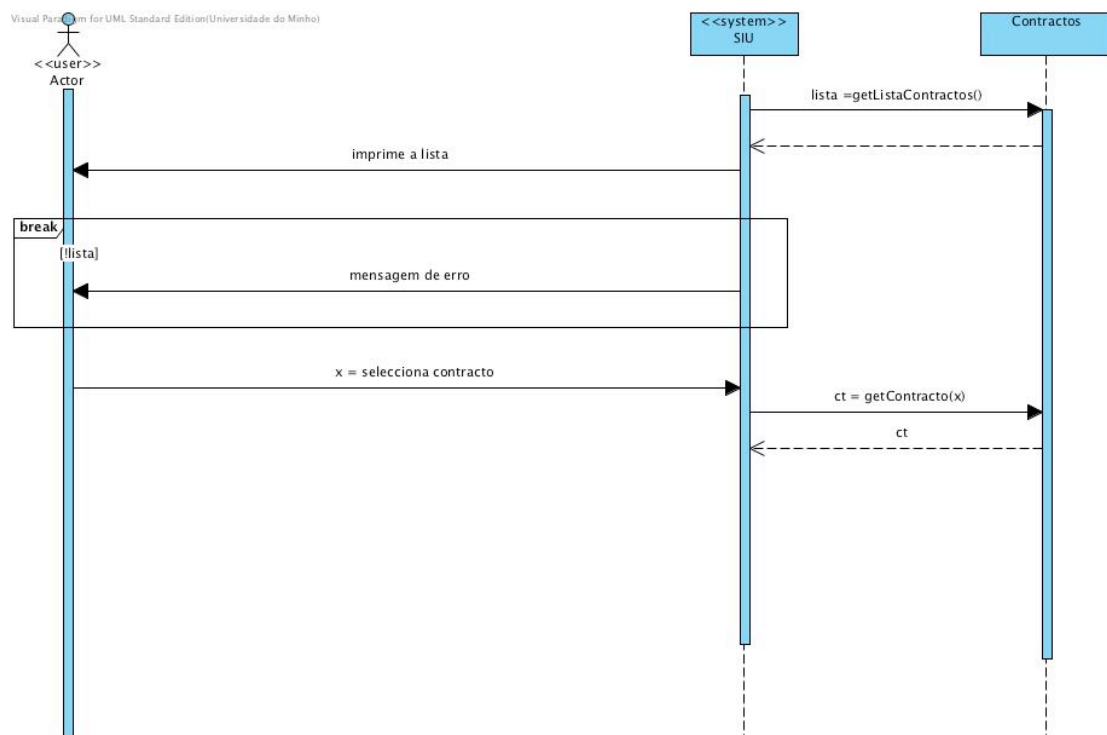
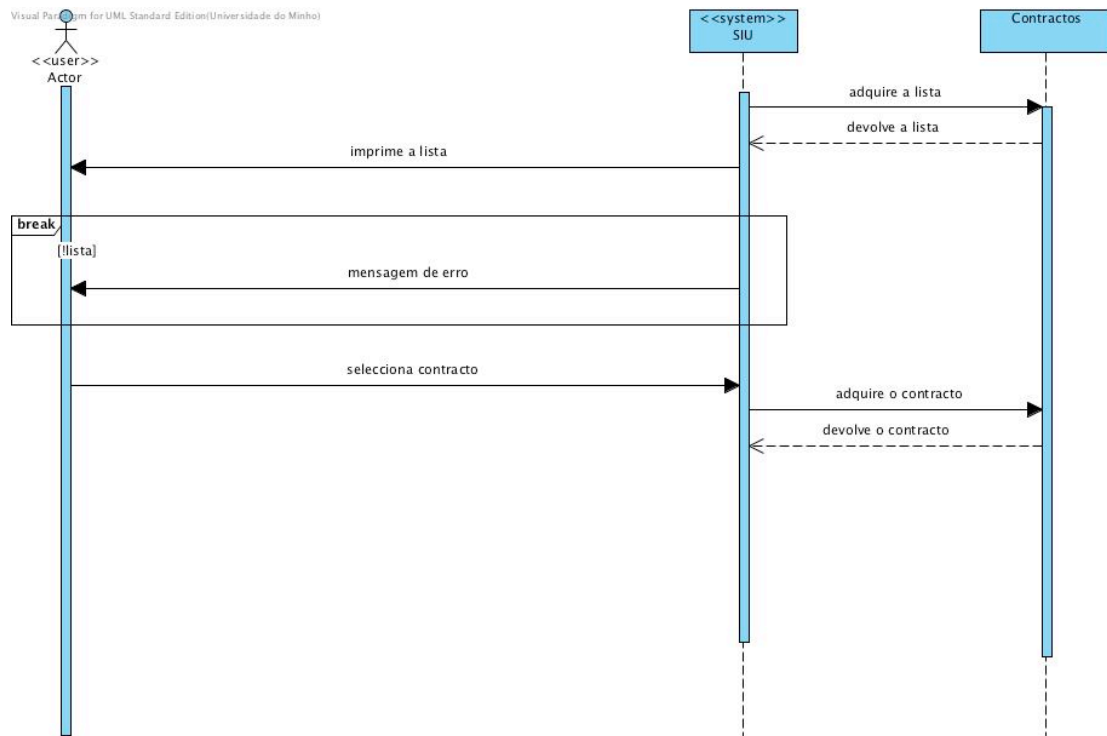
Diagramas Remover Contrato



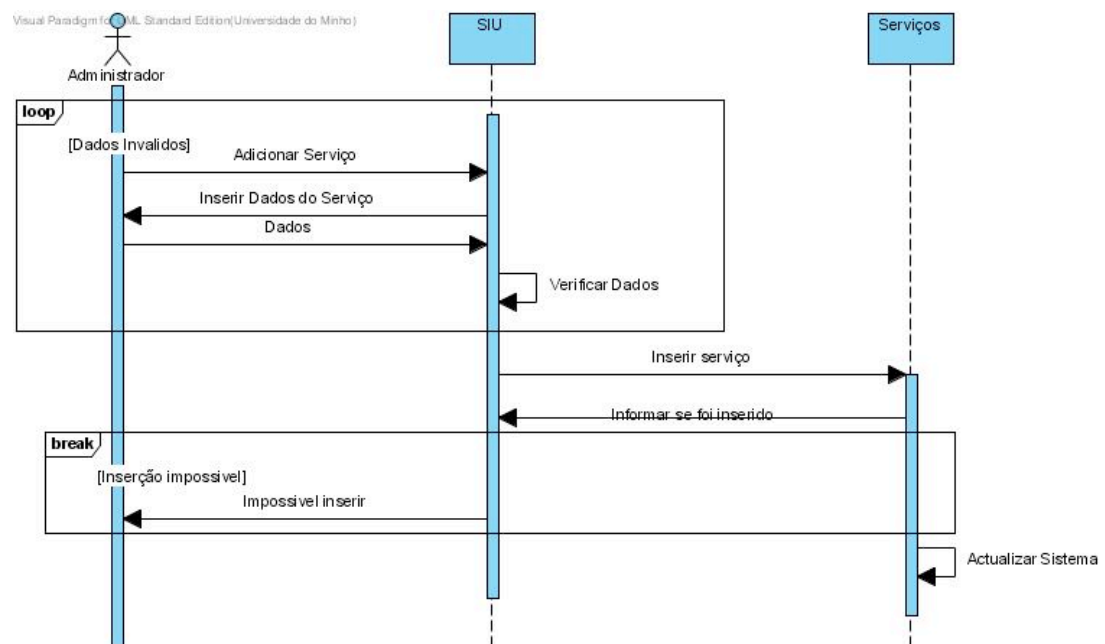
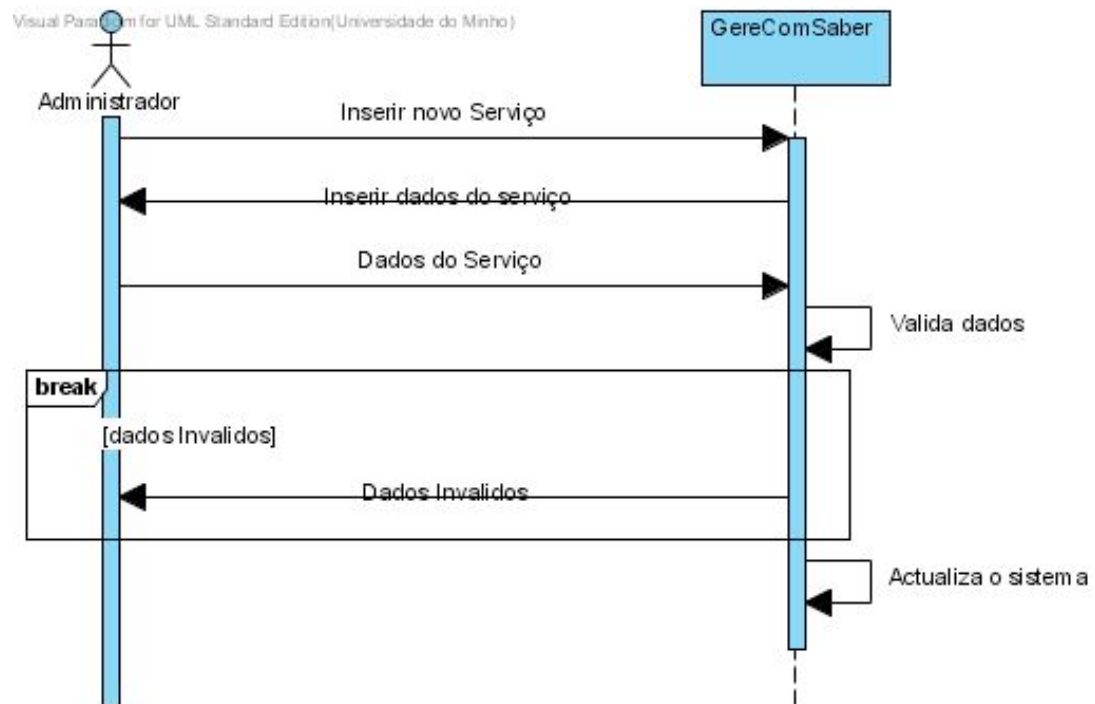


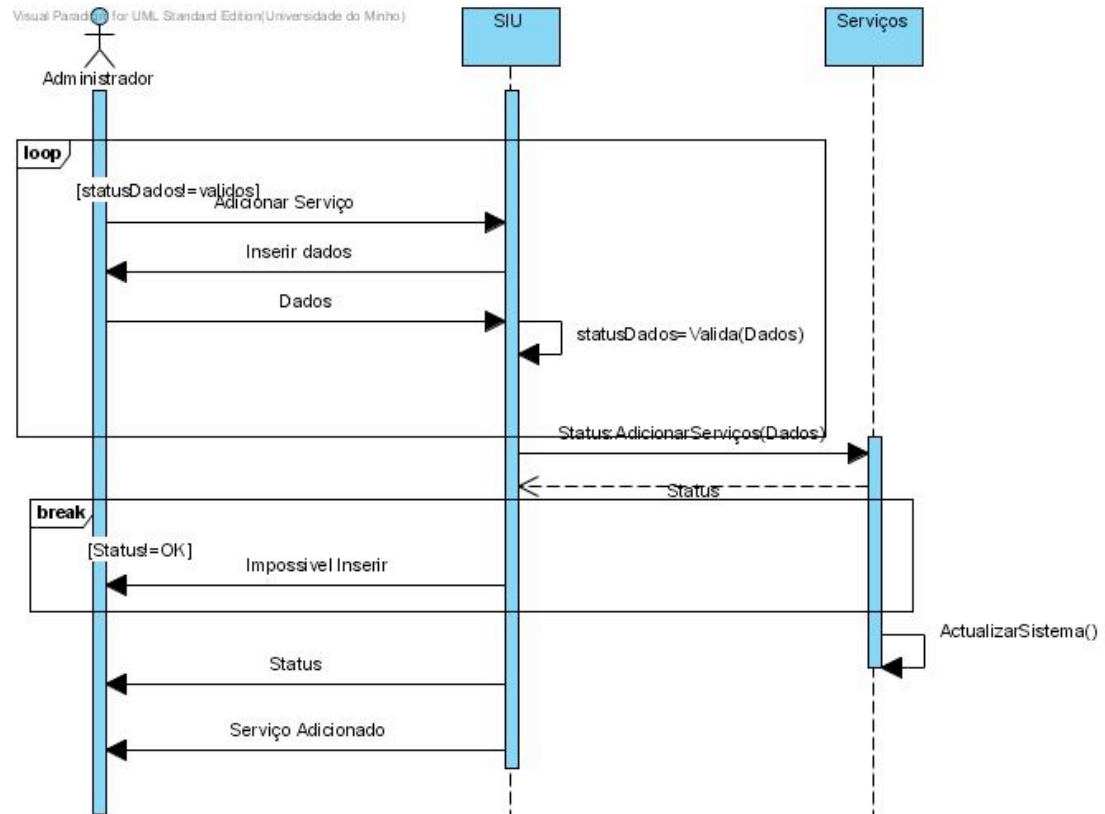
Diagramas Seleccionar Contrato



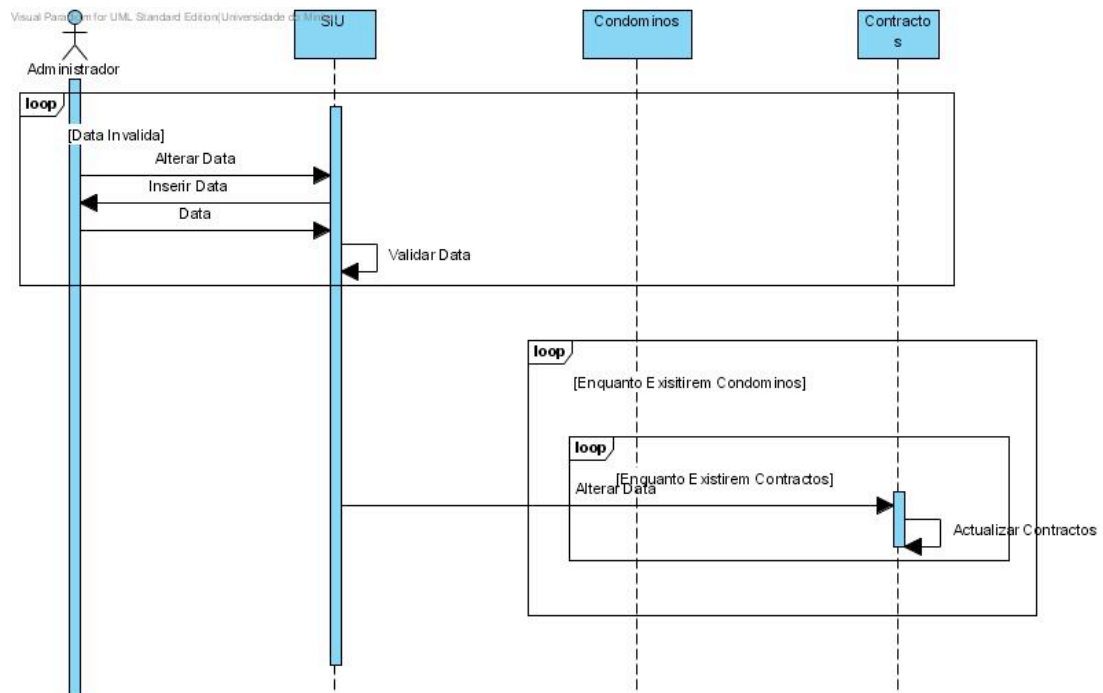
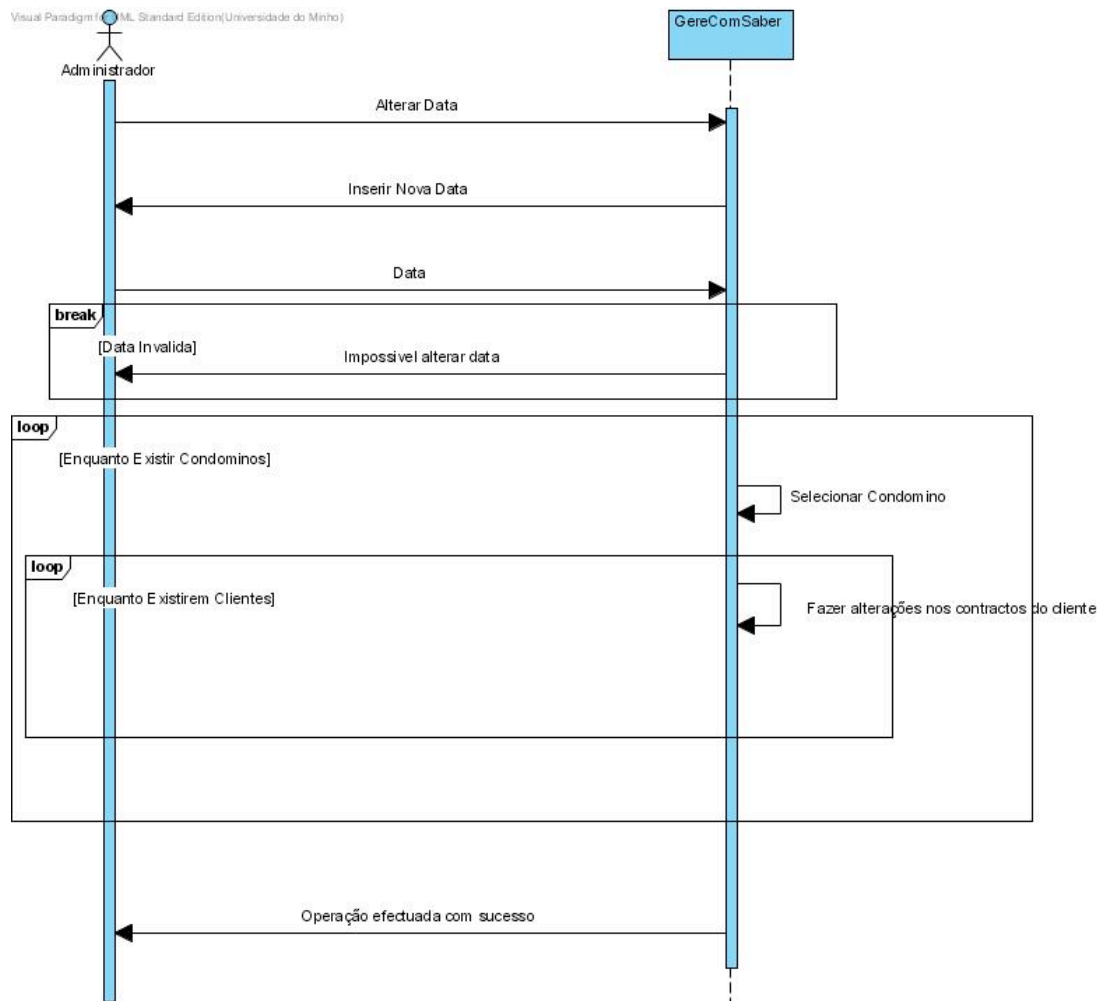


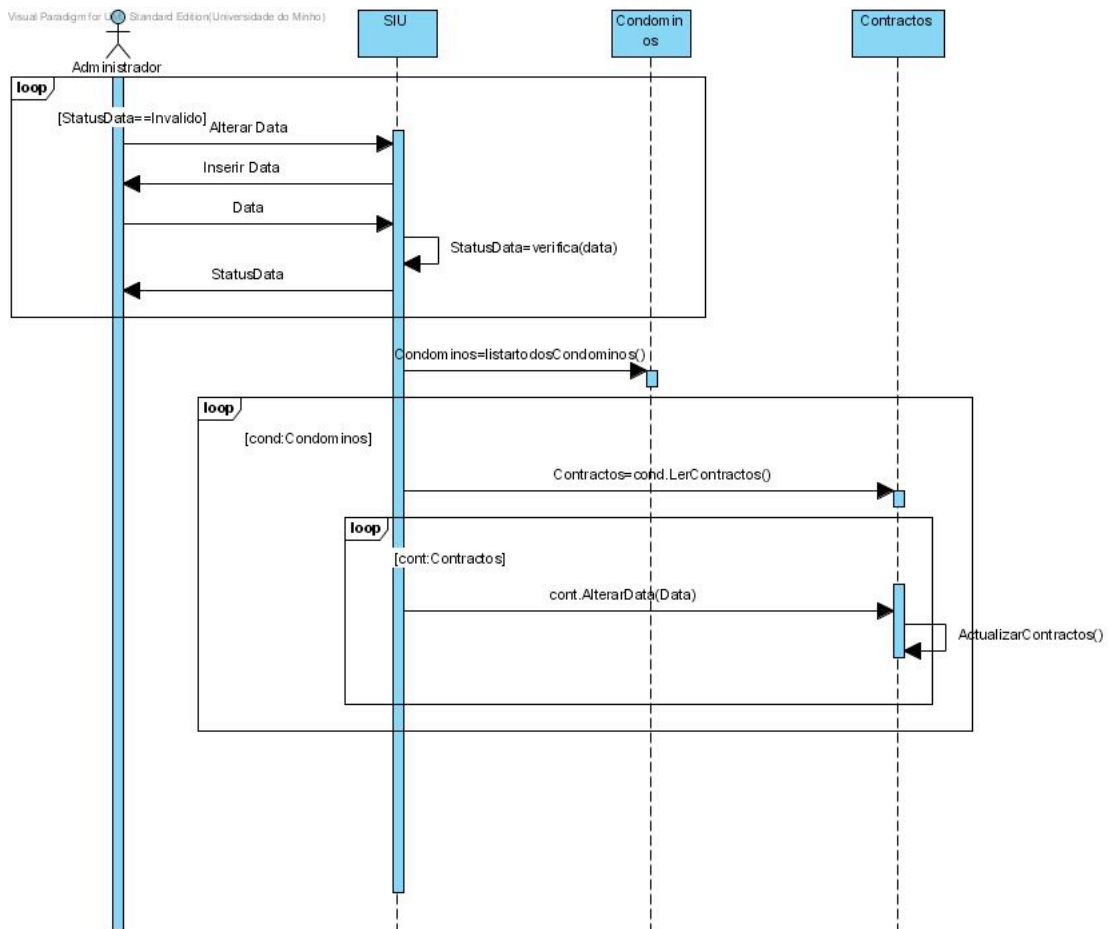
Diagramas Adicionar Serviço



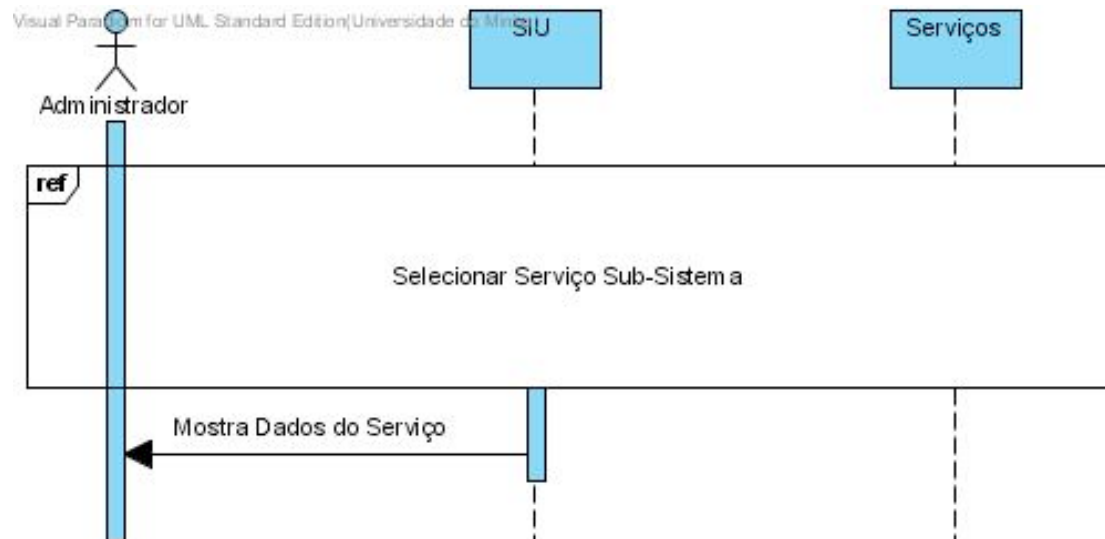
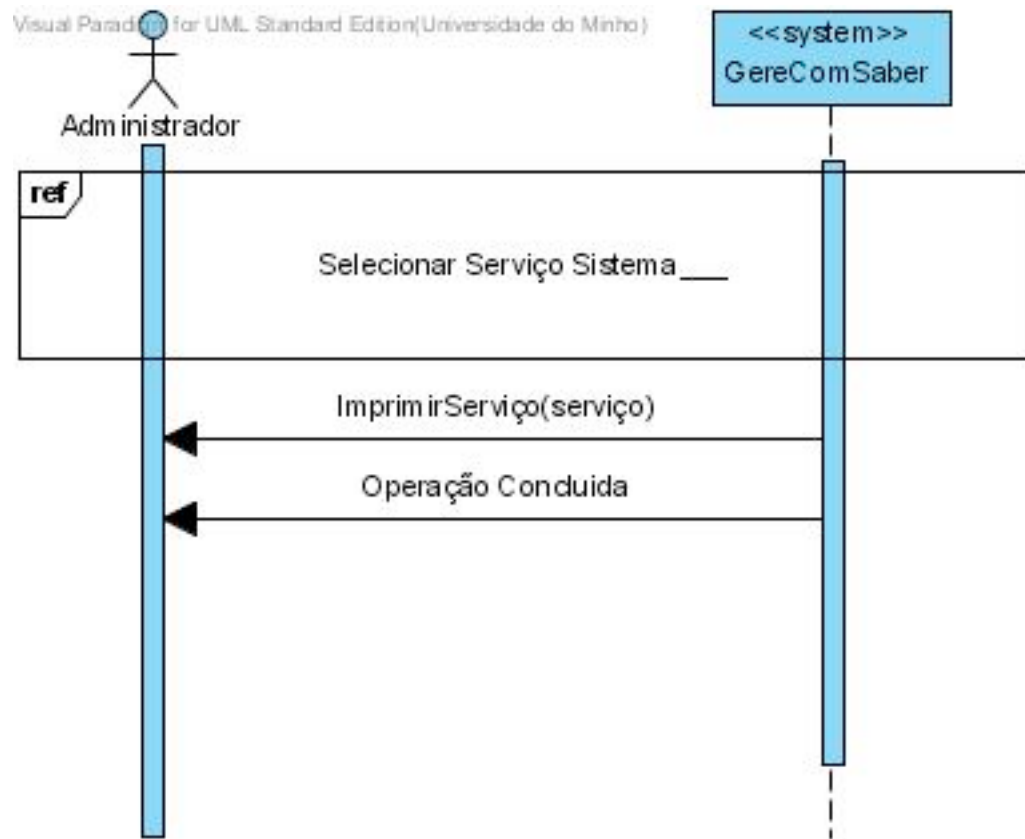


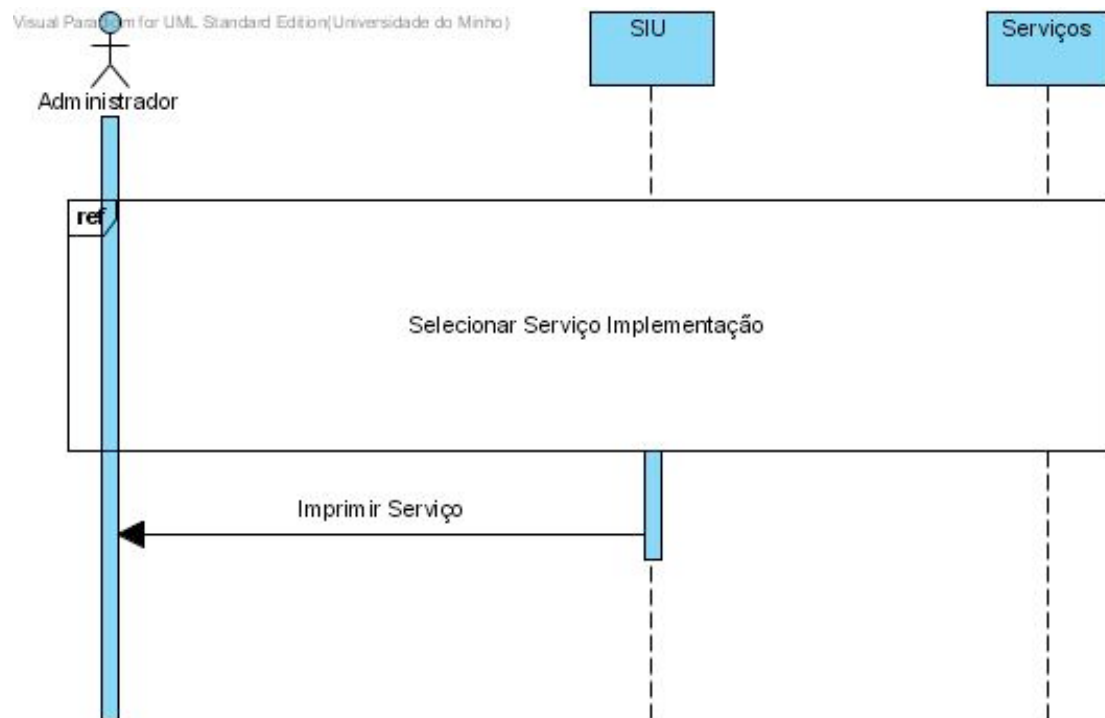
Diagramas Alterar Data



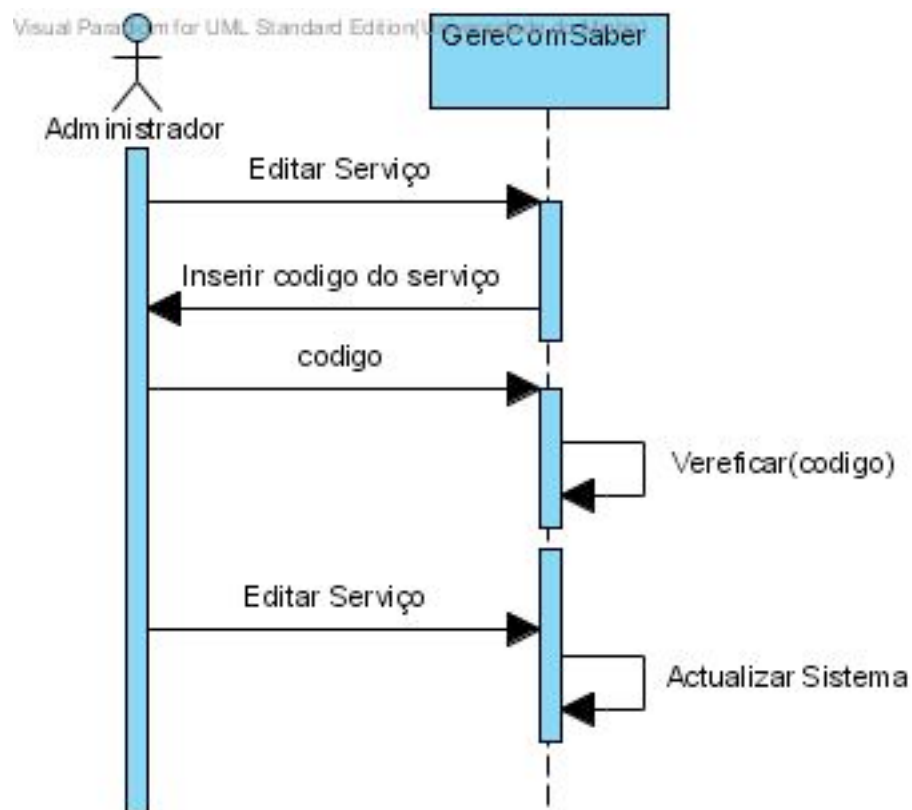


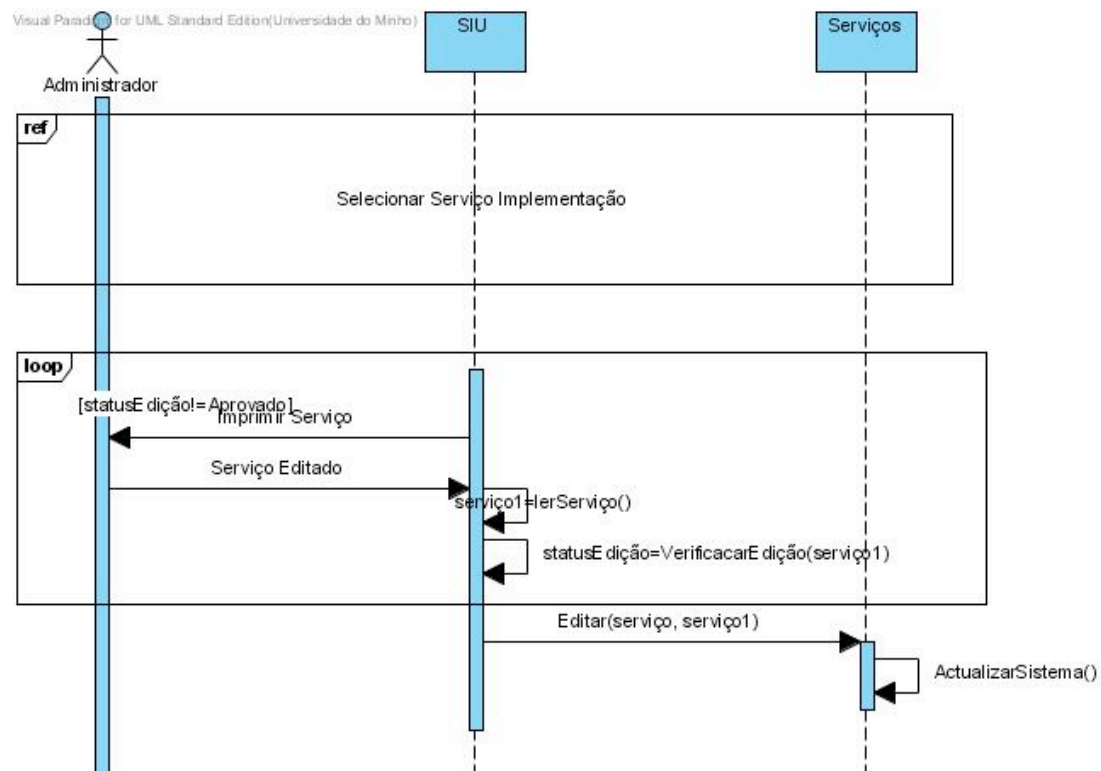
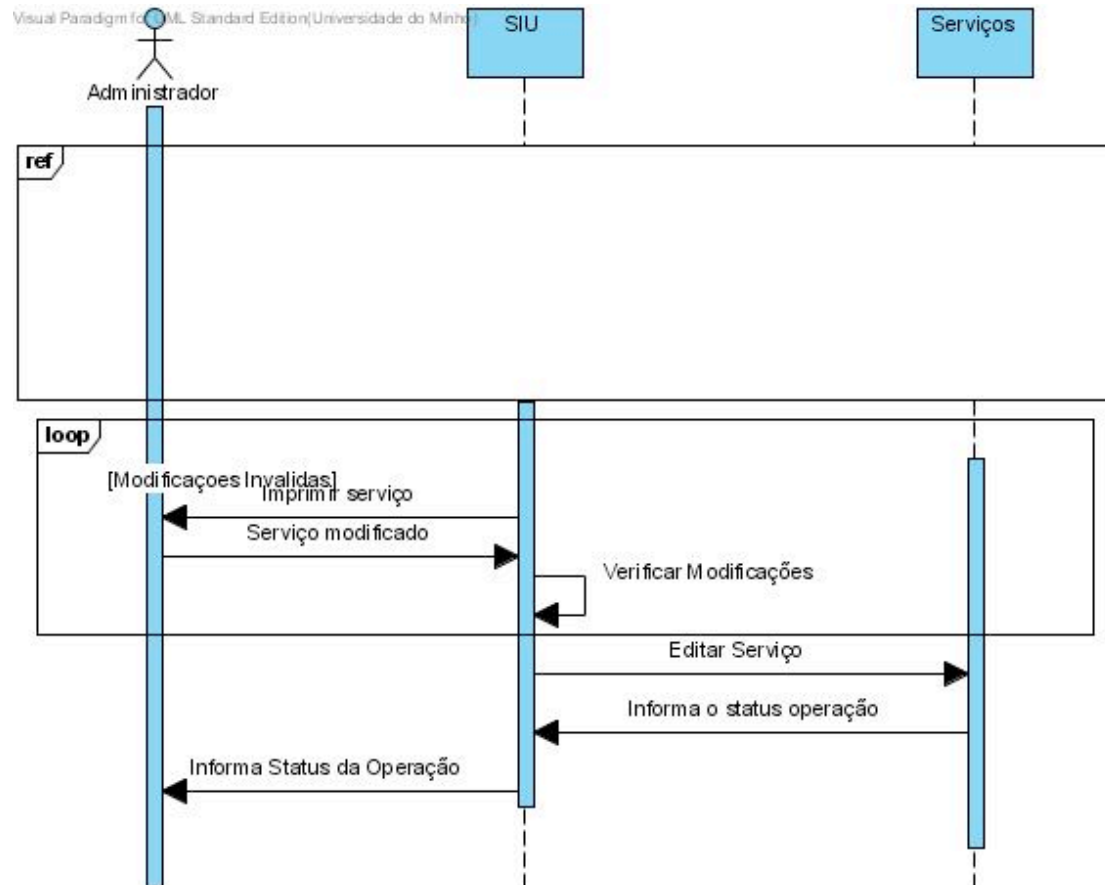
Diagramas Consultar Serviço



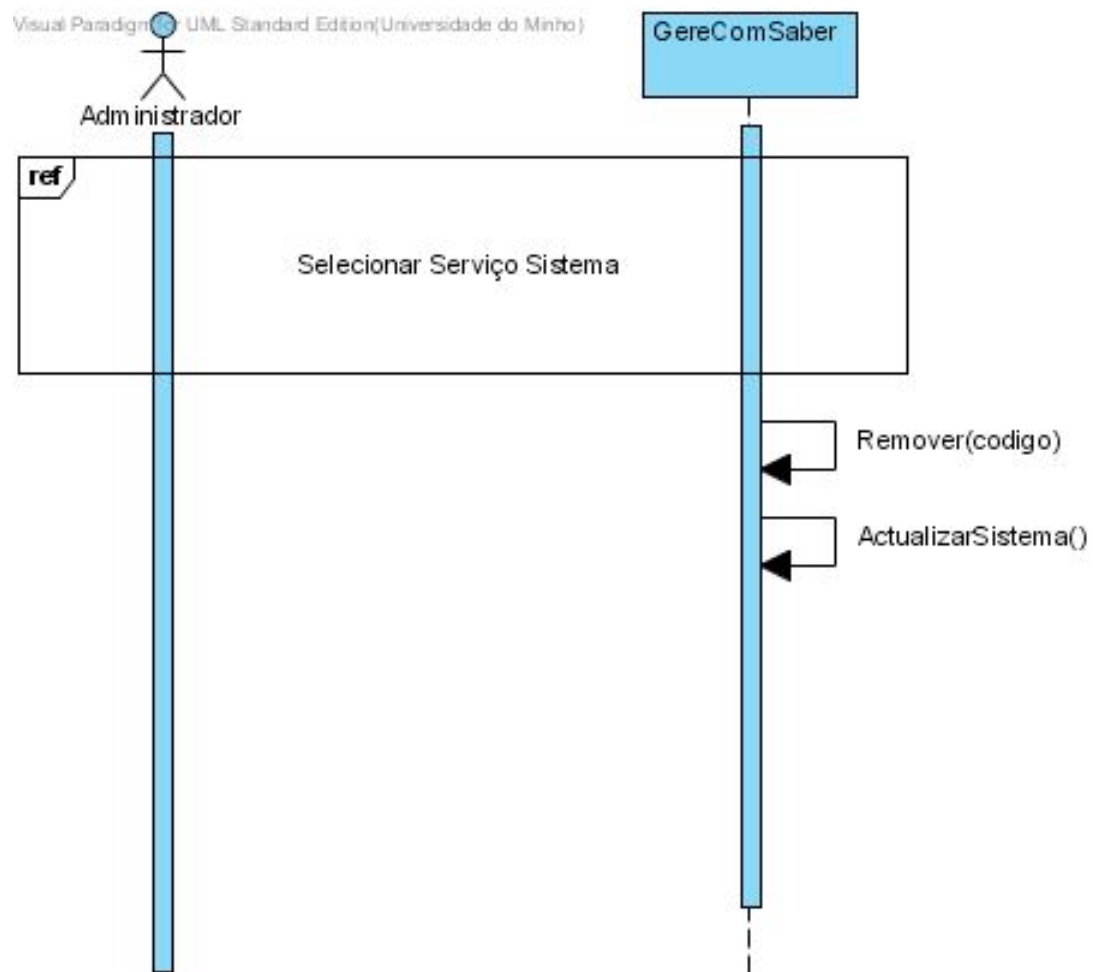


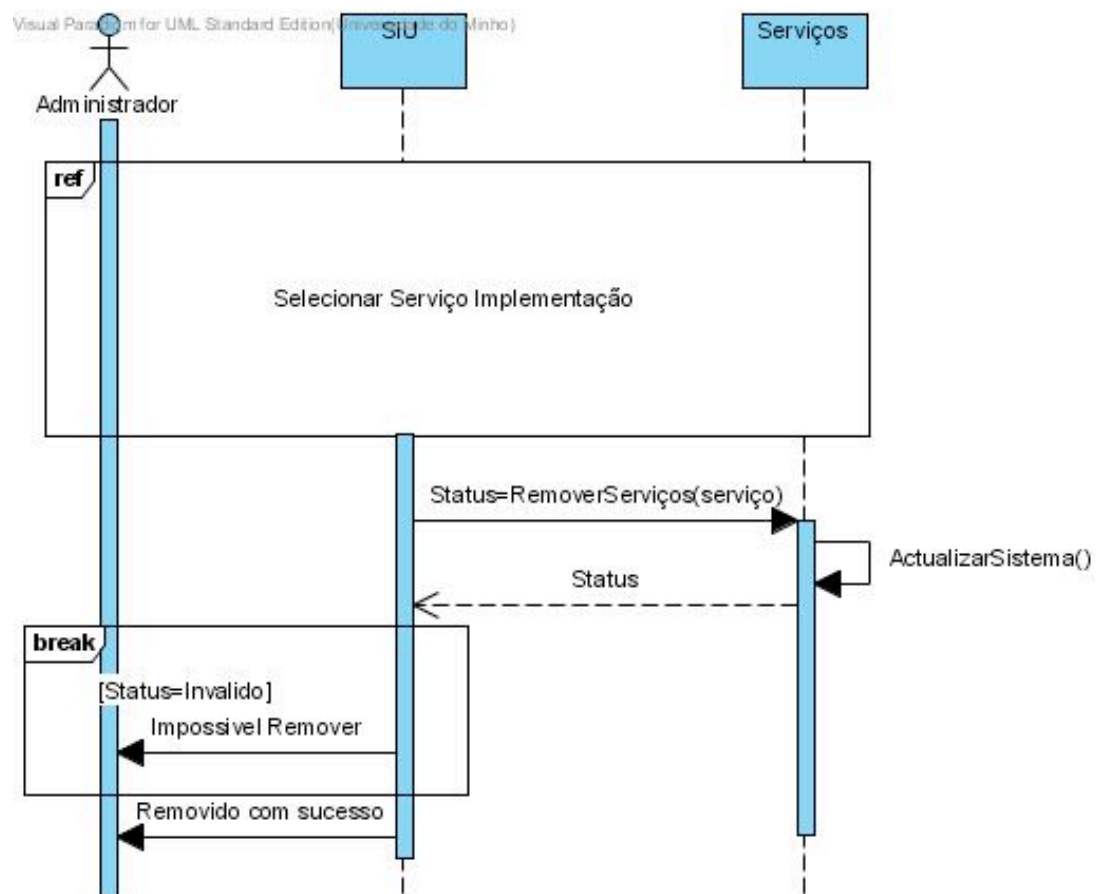
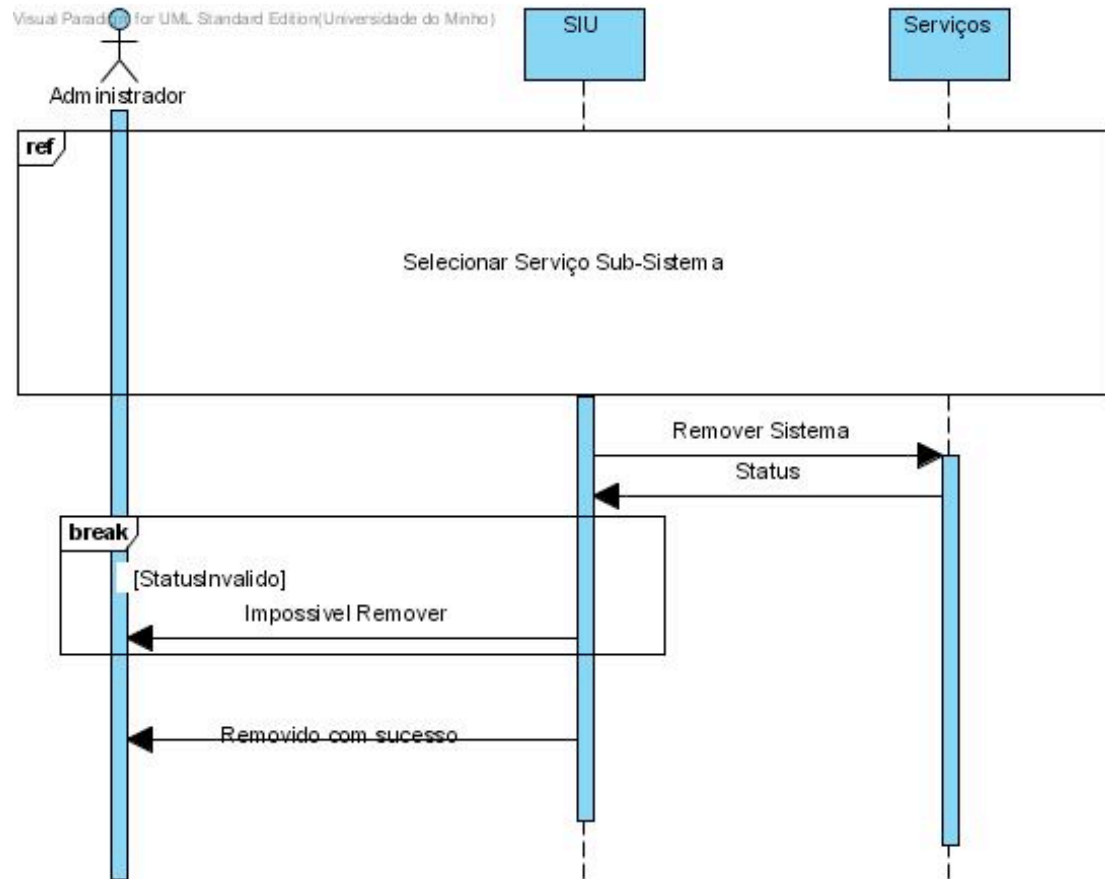
Diagramas Editar Serviço



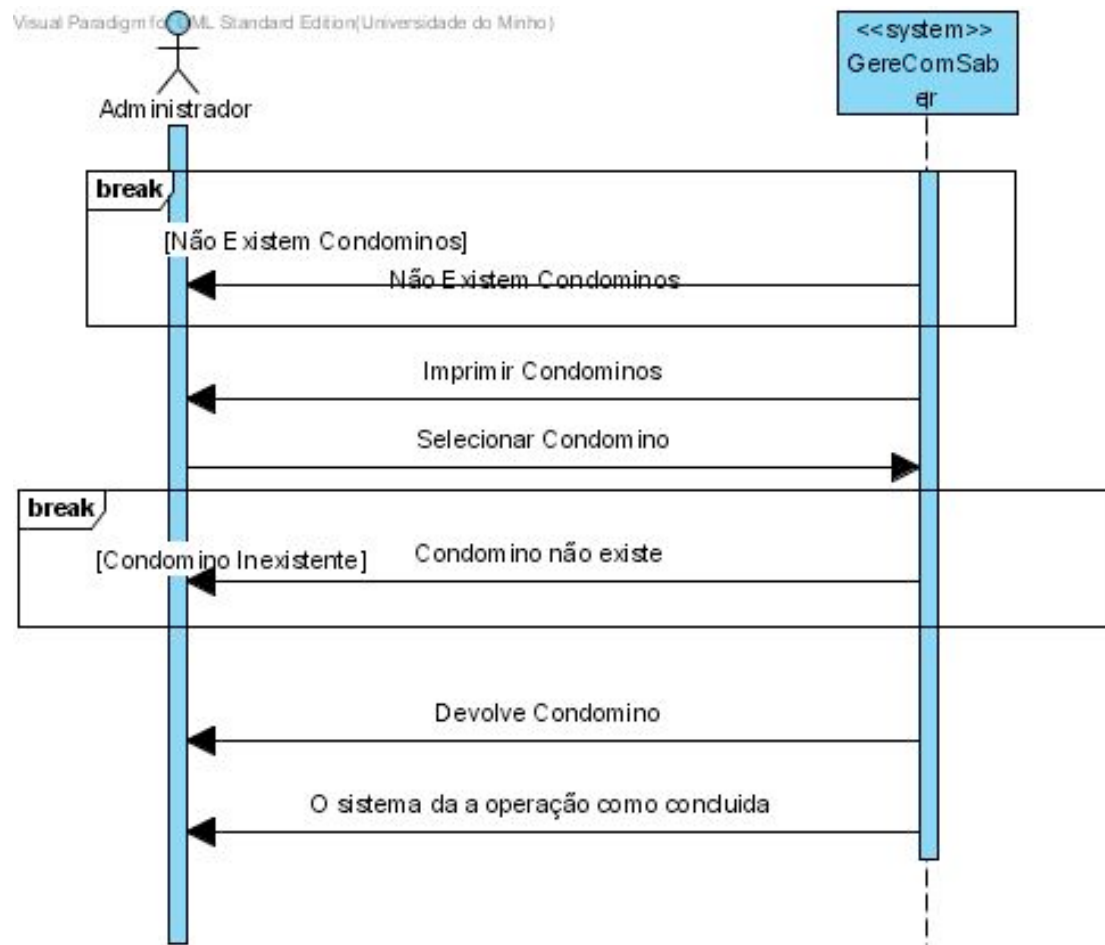


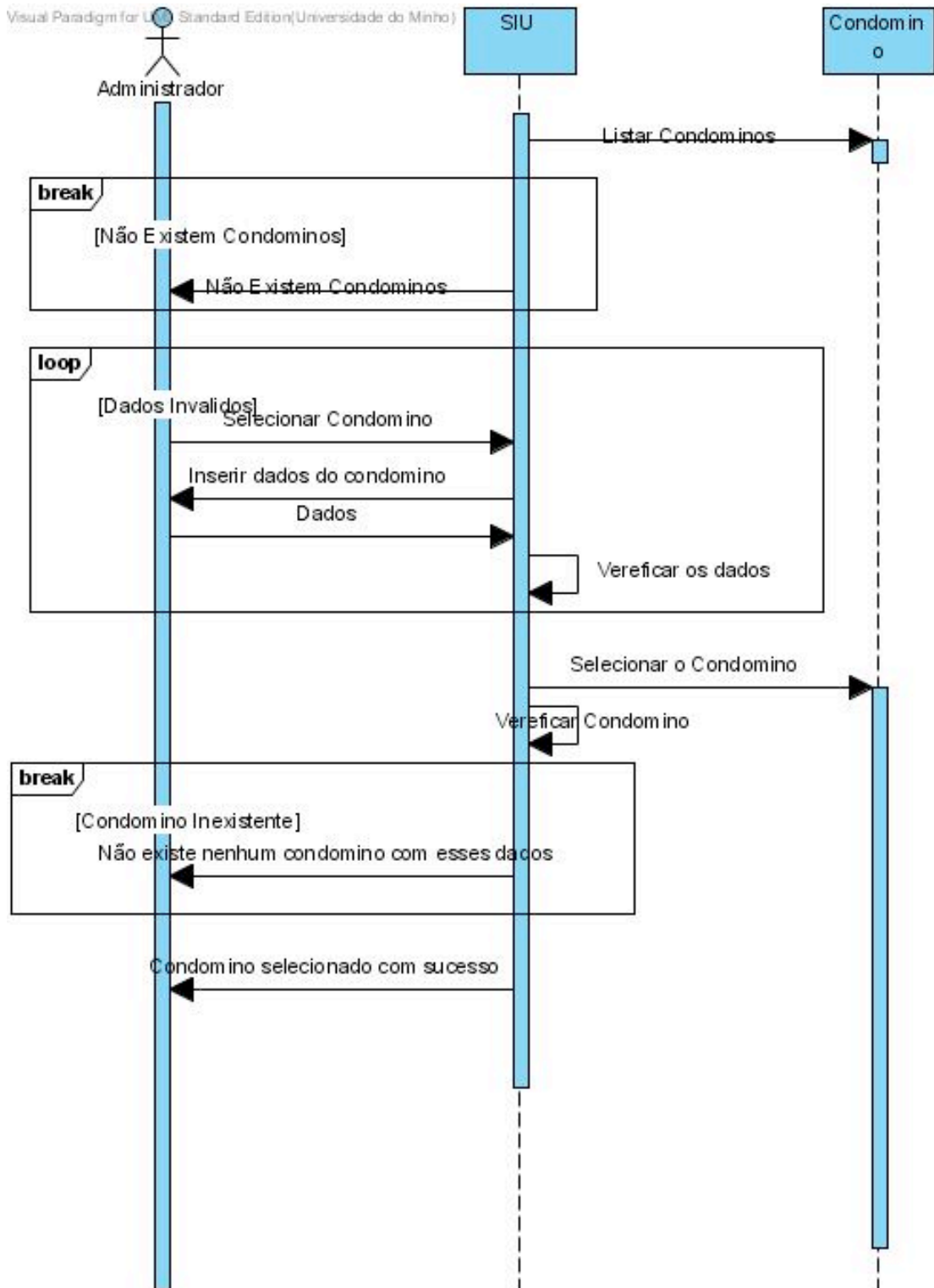
Diagramas Remover Serviço

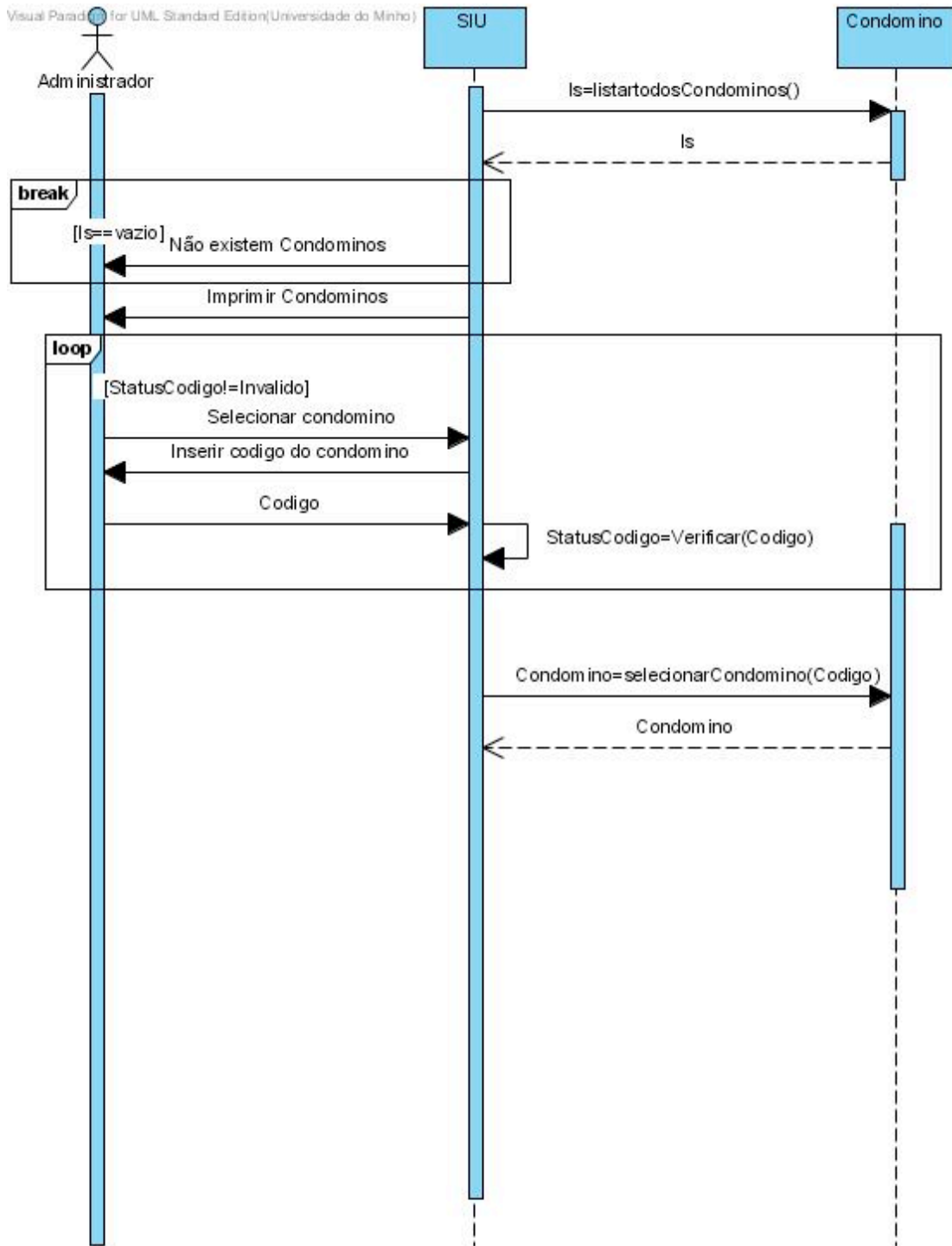




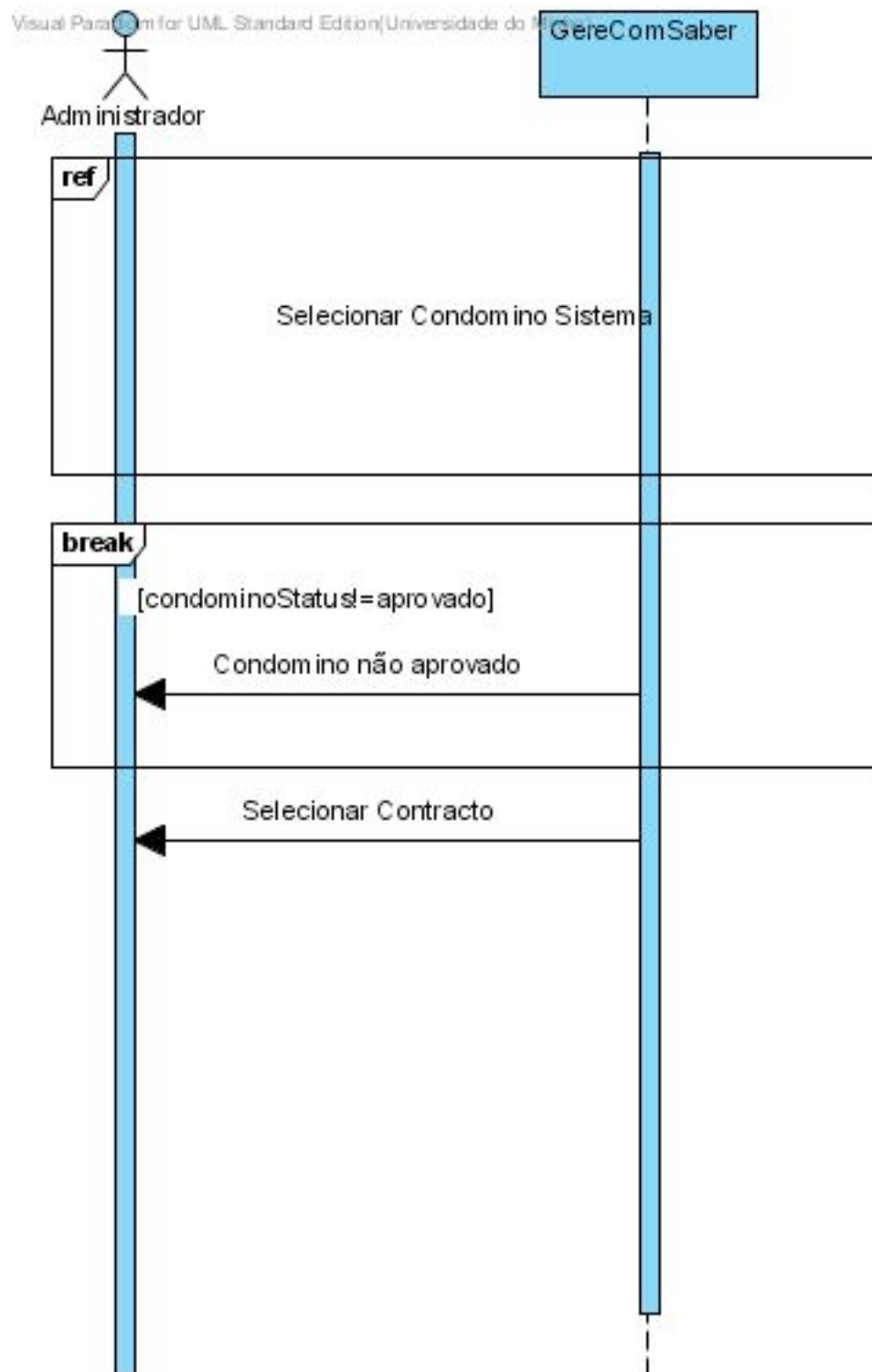
Diagramas Seleccionar Condómino

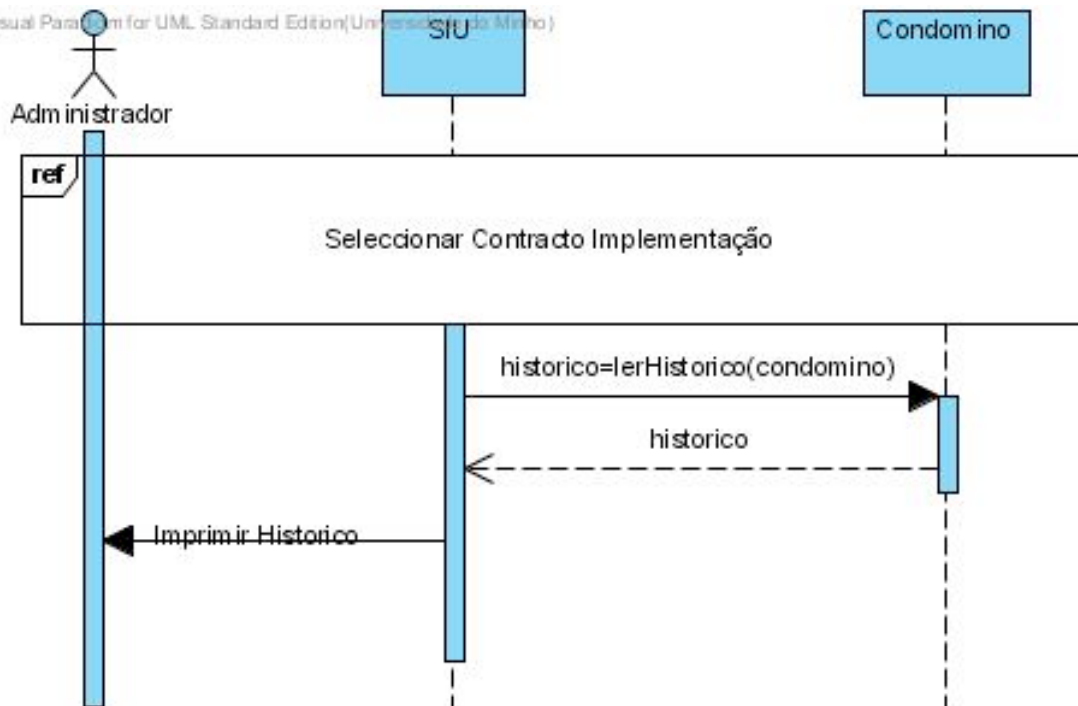
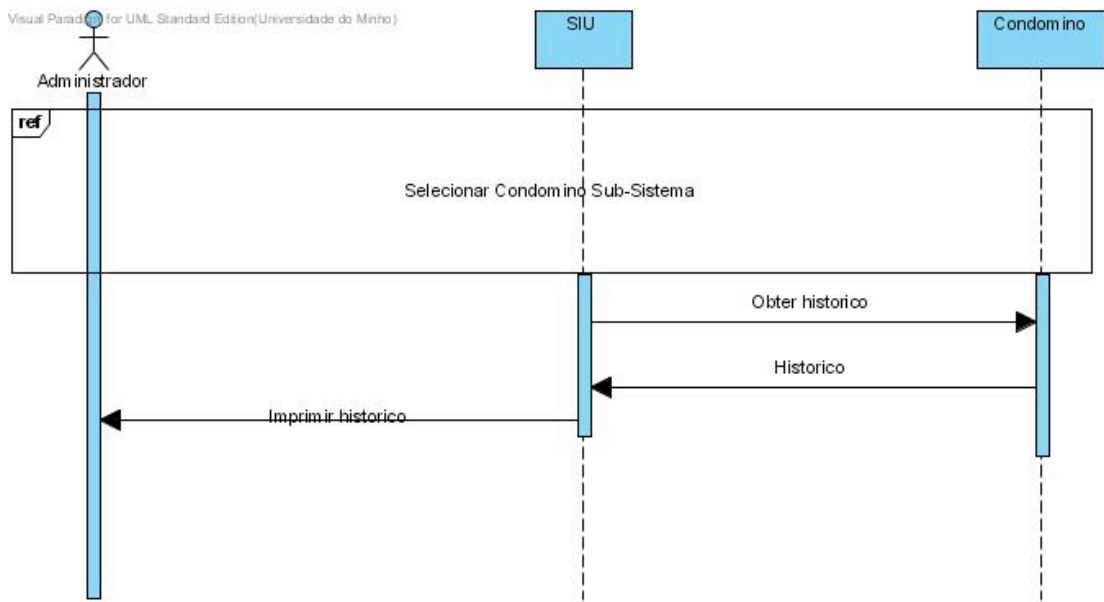




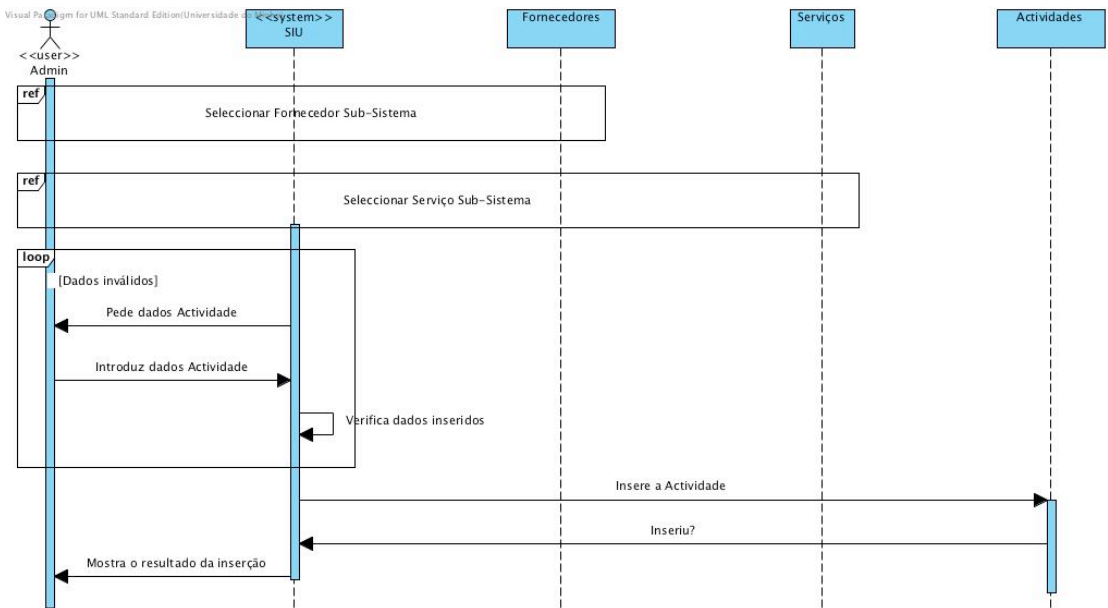
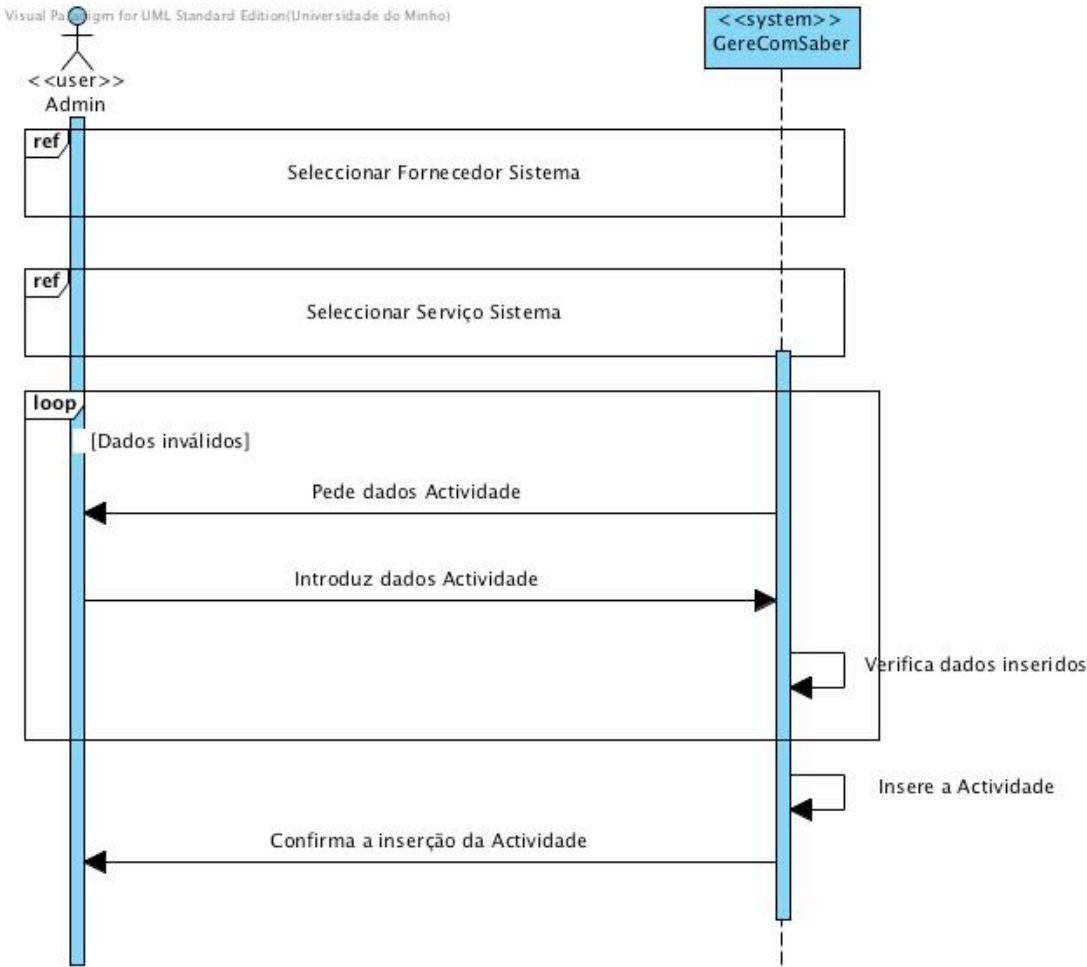


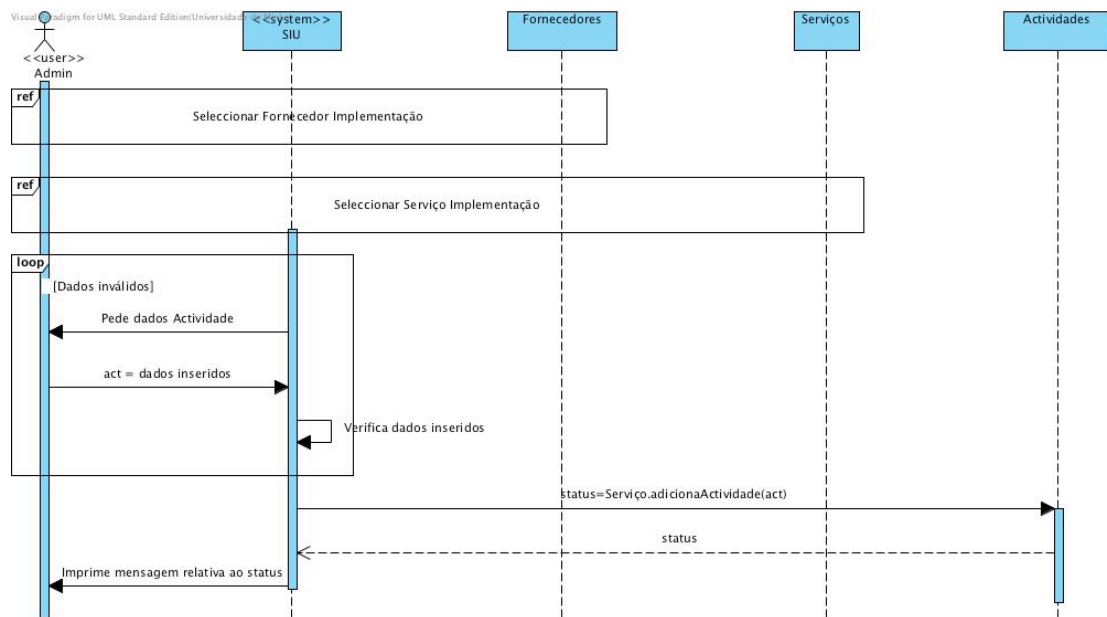
Diagramas Ver Histórico



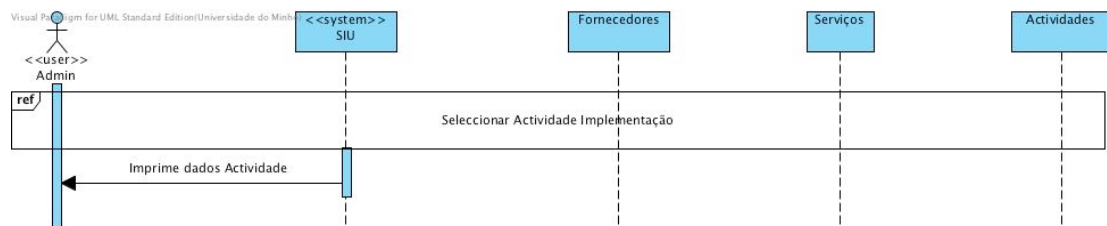
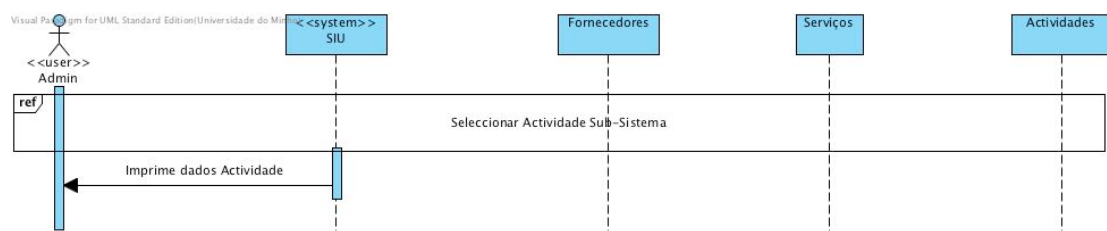
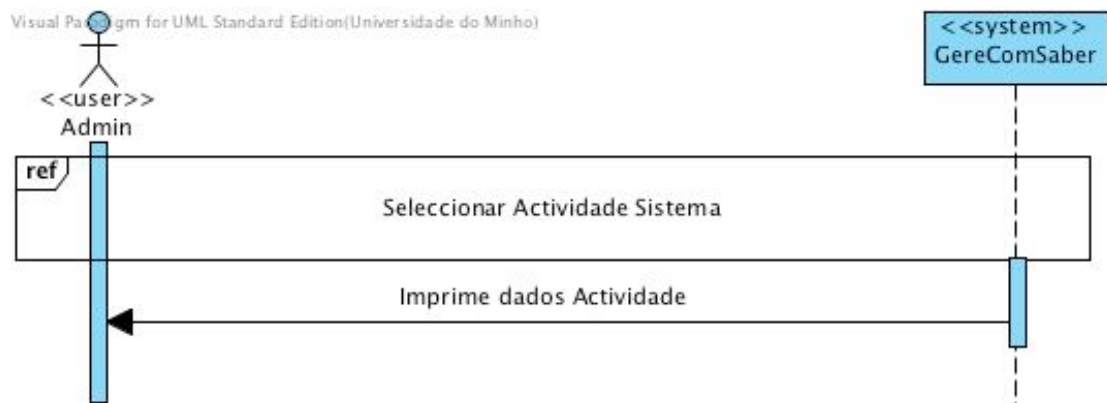


Diagramas Adicionar Actividade

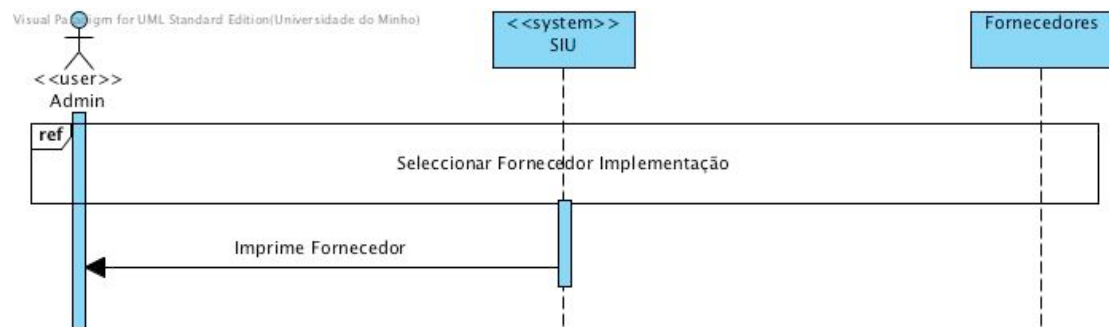
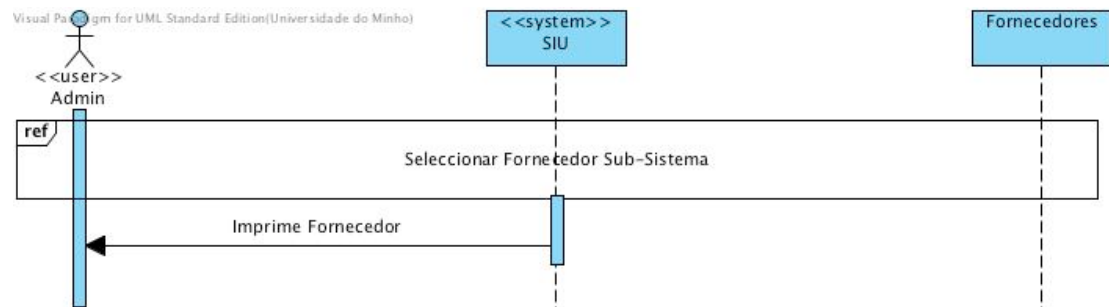




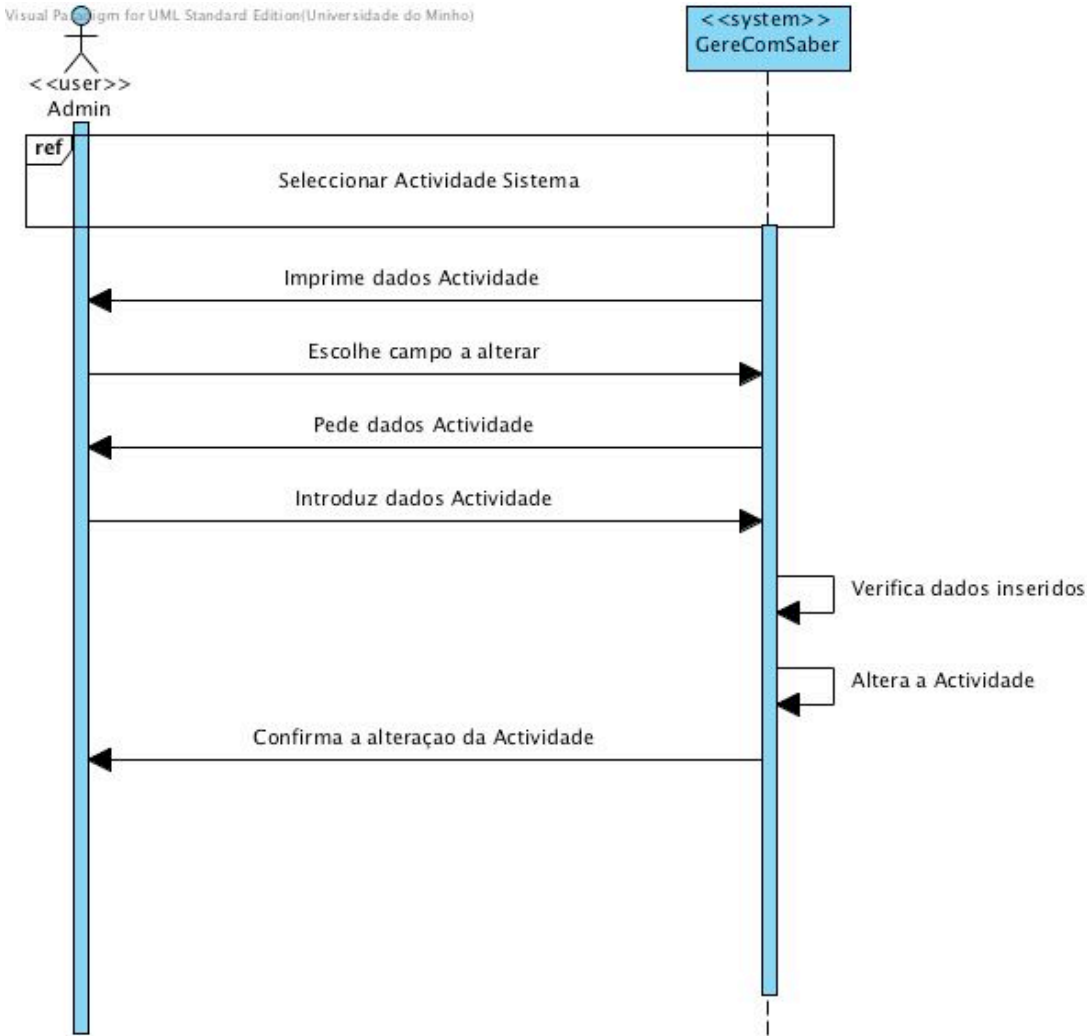
Diagramas Consultar Actividade

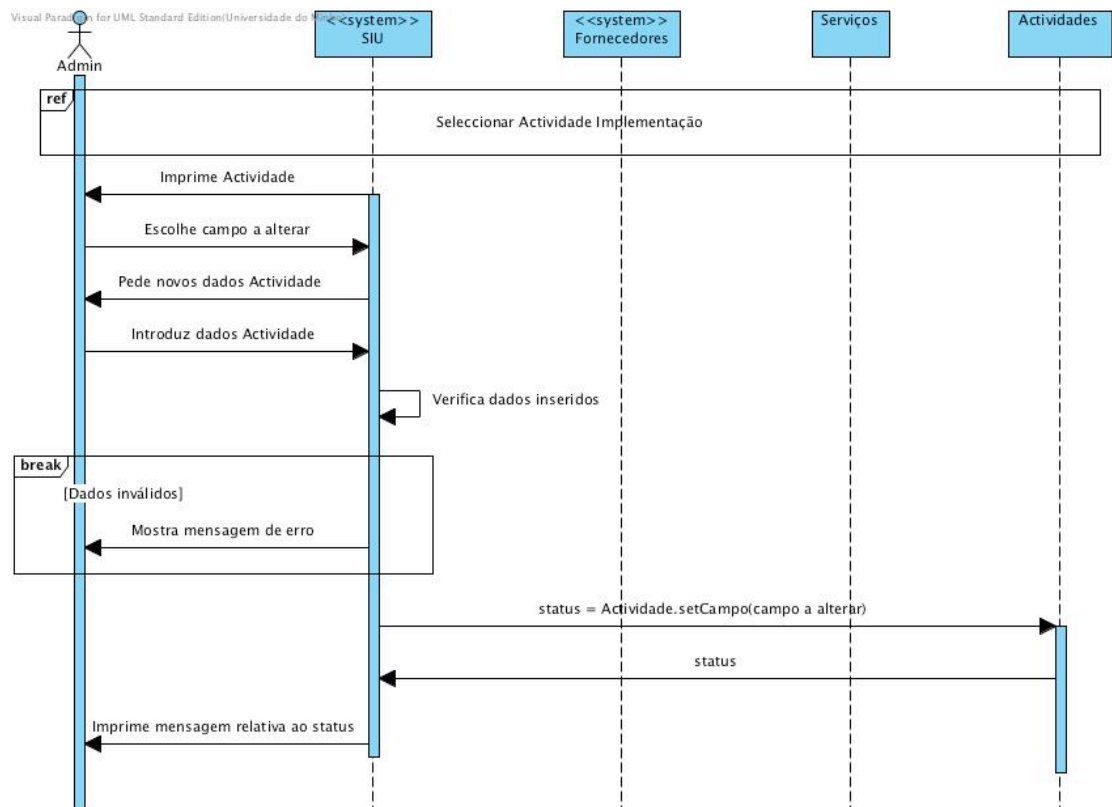
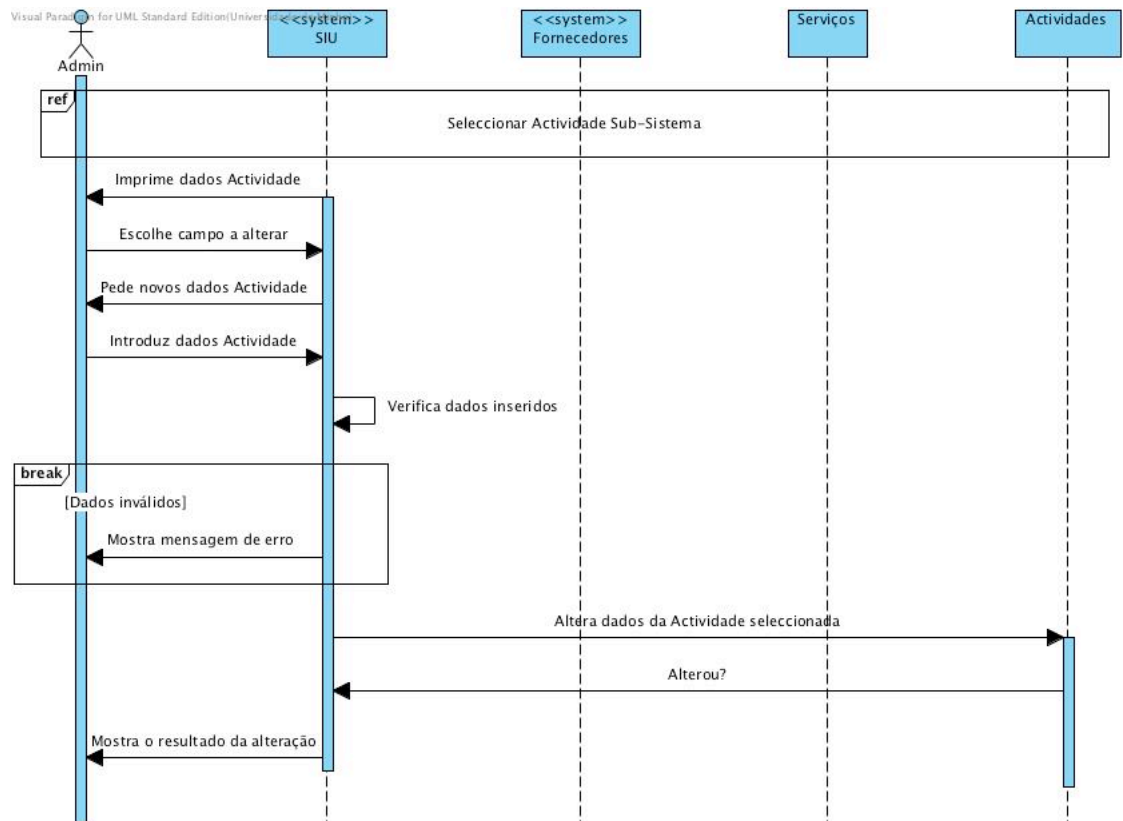


Diagramas Consultar Fornecedor

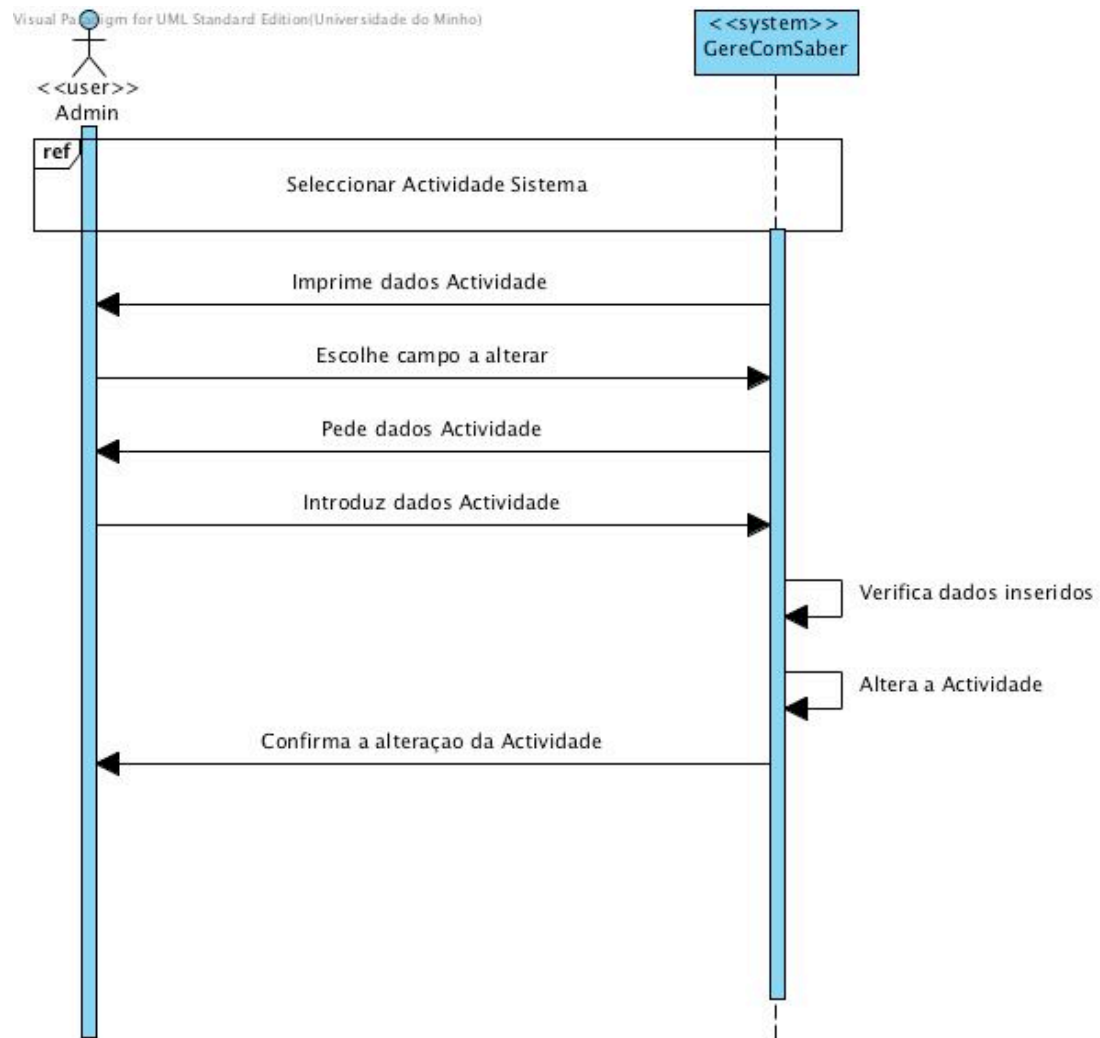


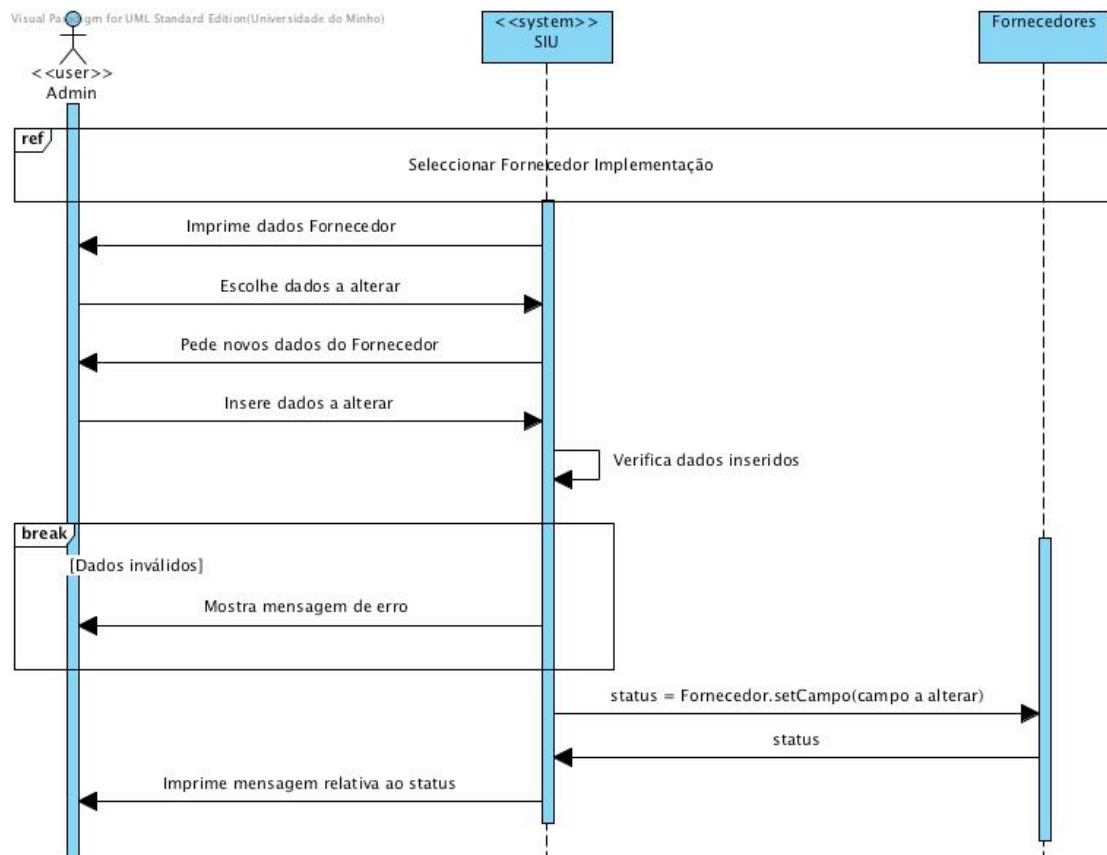
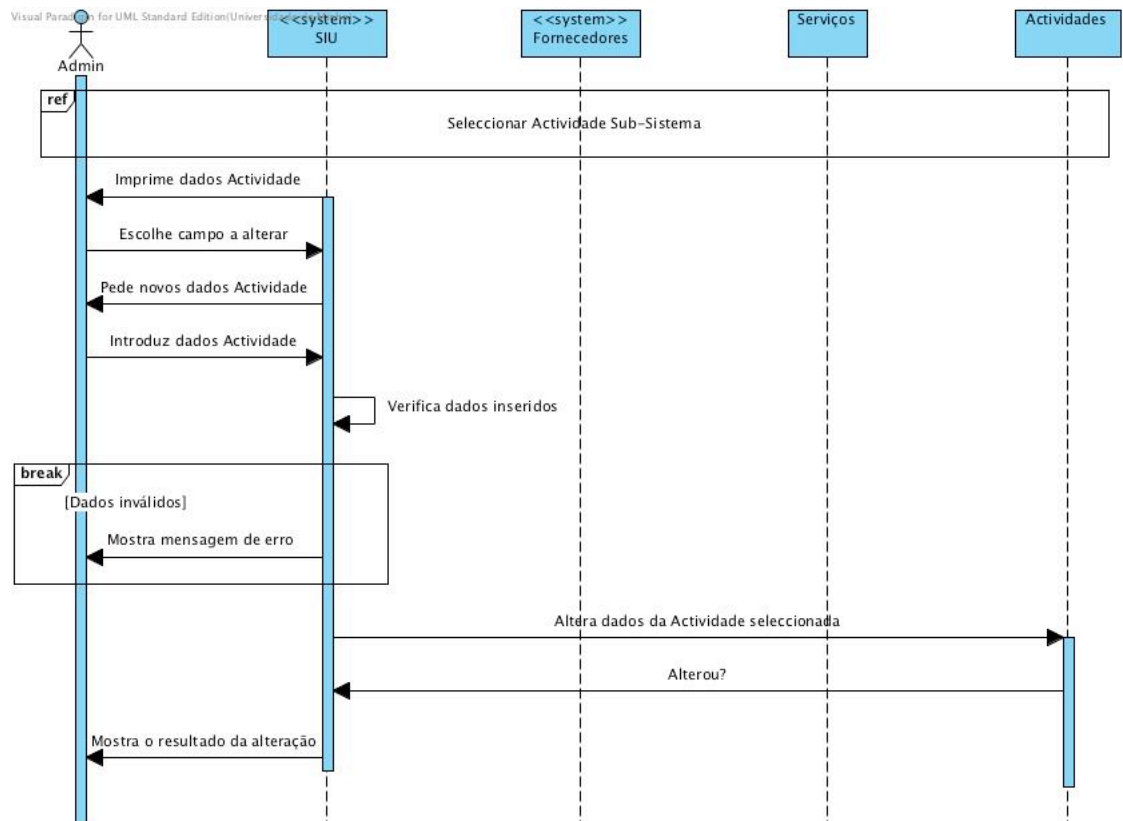
Diagramas Editar Actividade



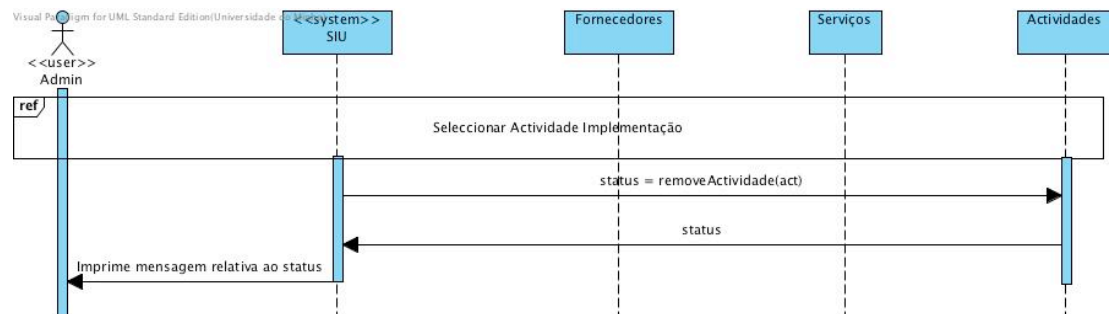
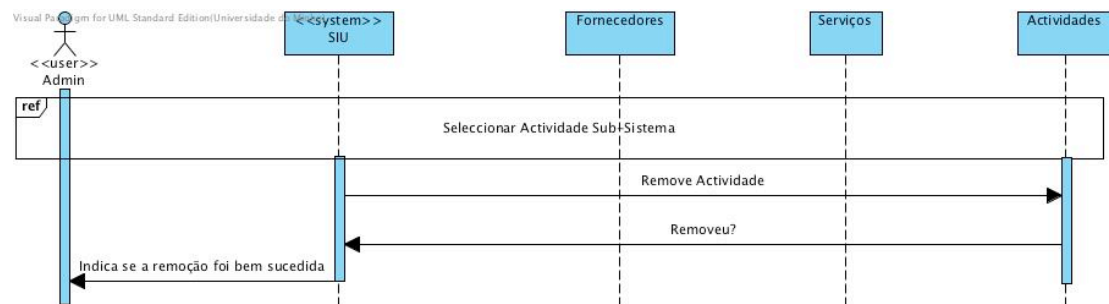
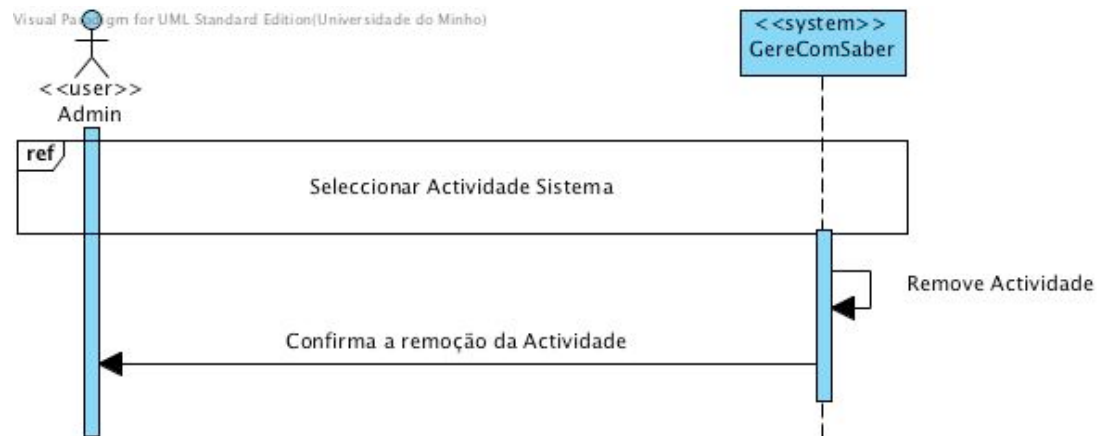


Diagramas Editar Fornecedor

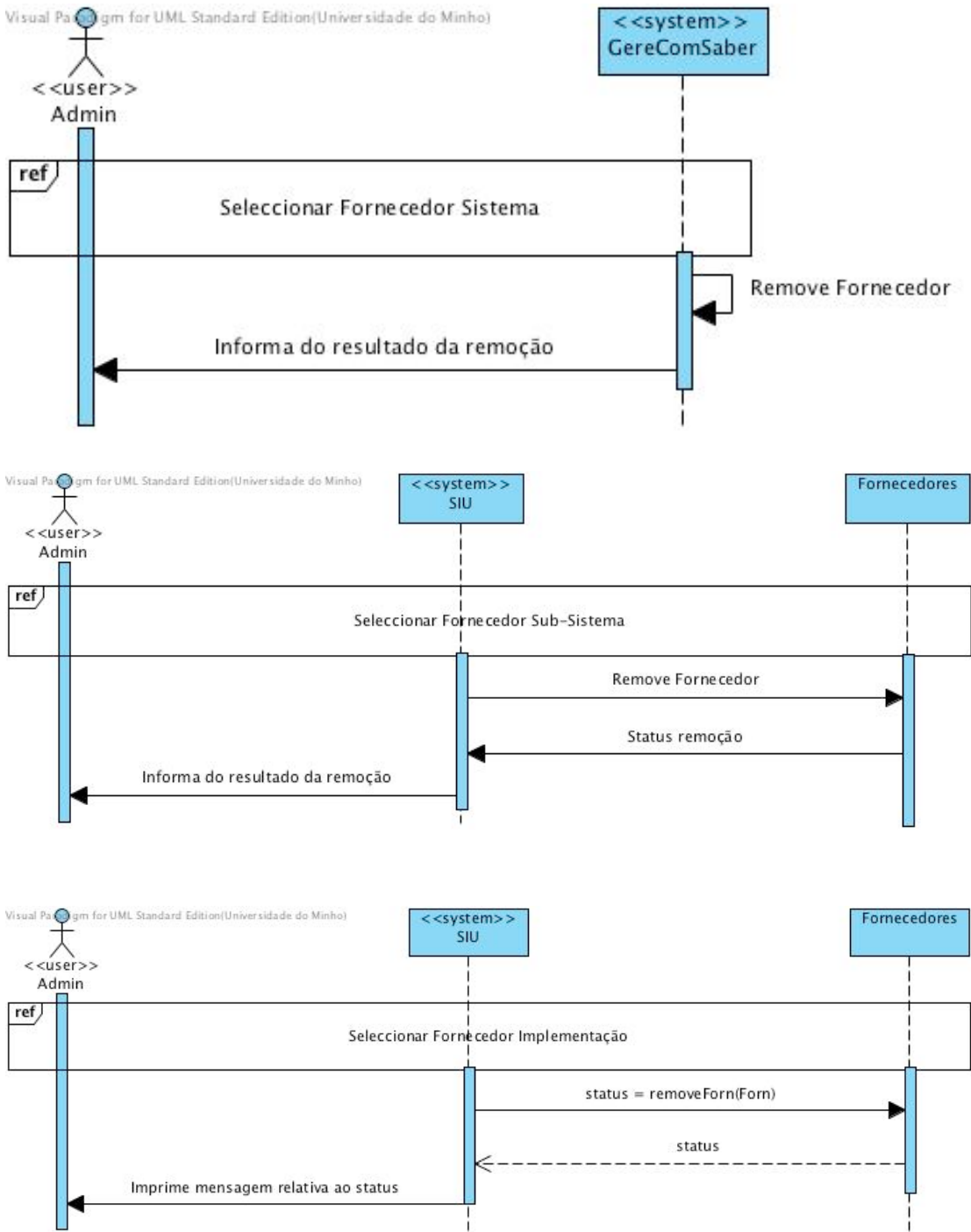




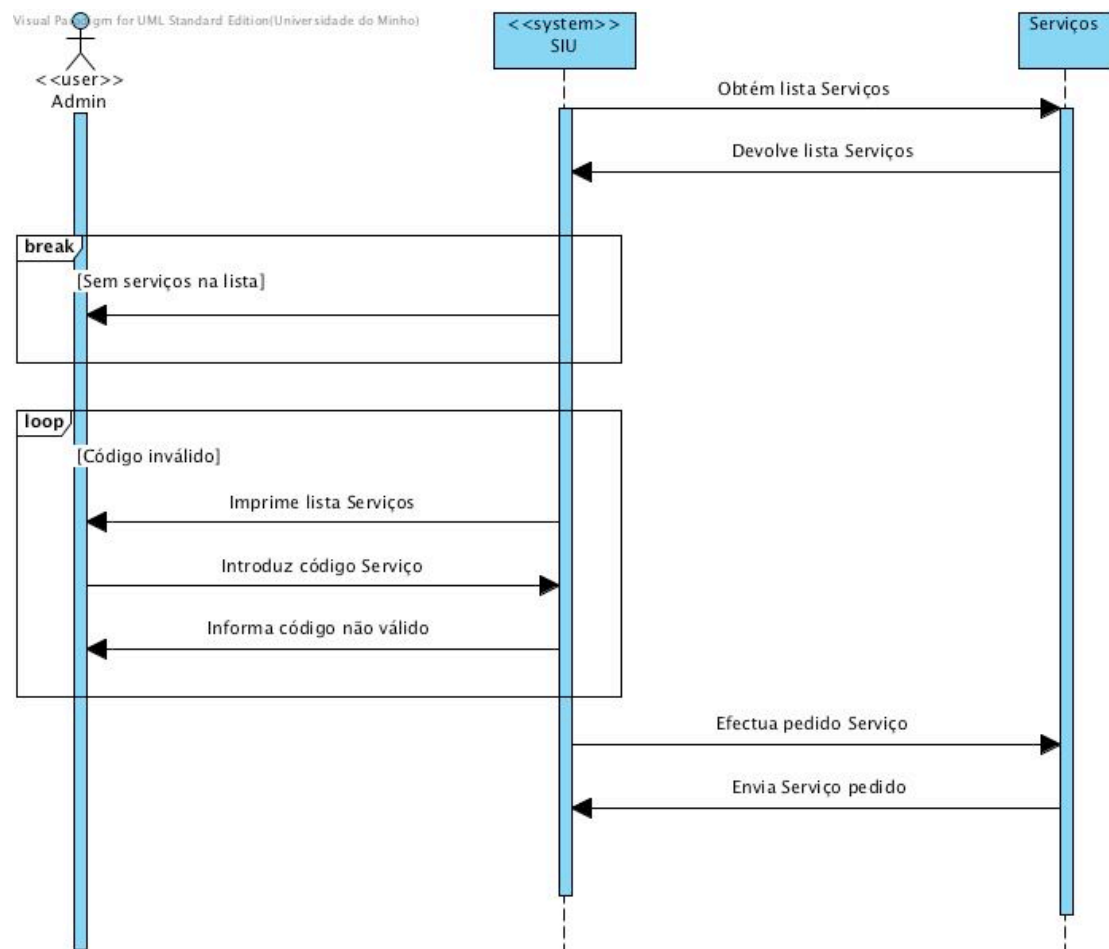
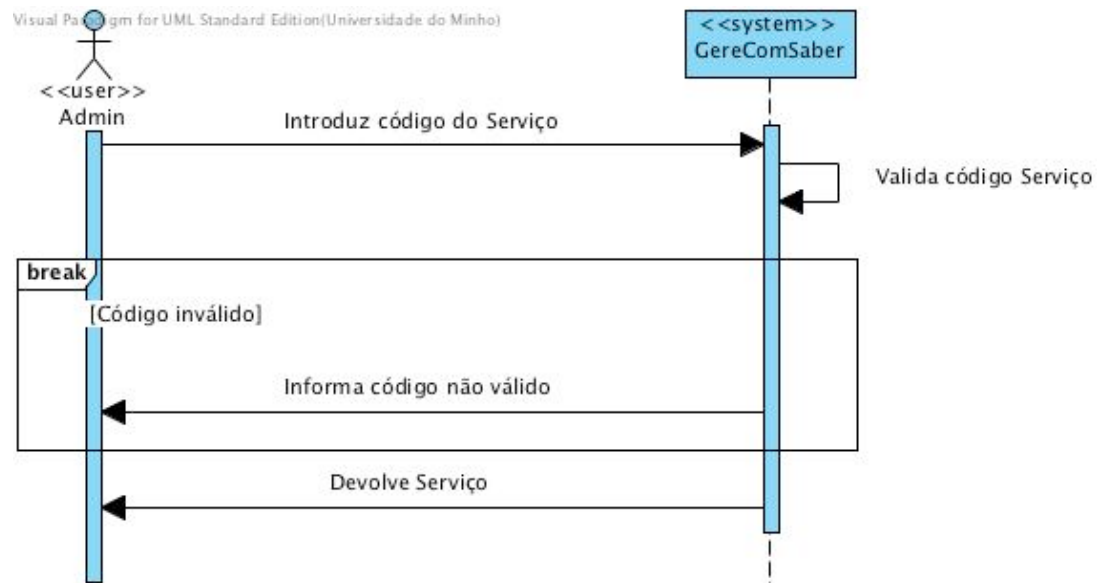
Diagramas Remover Actividade

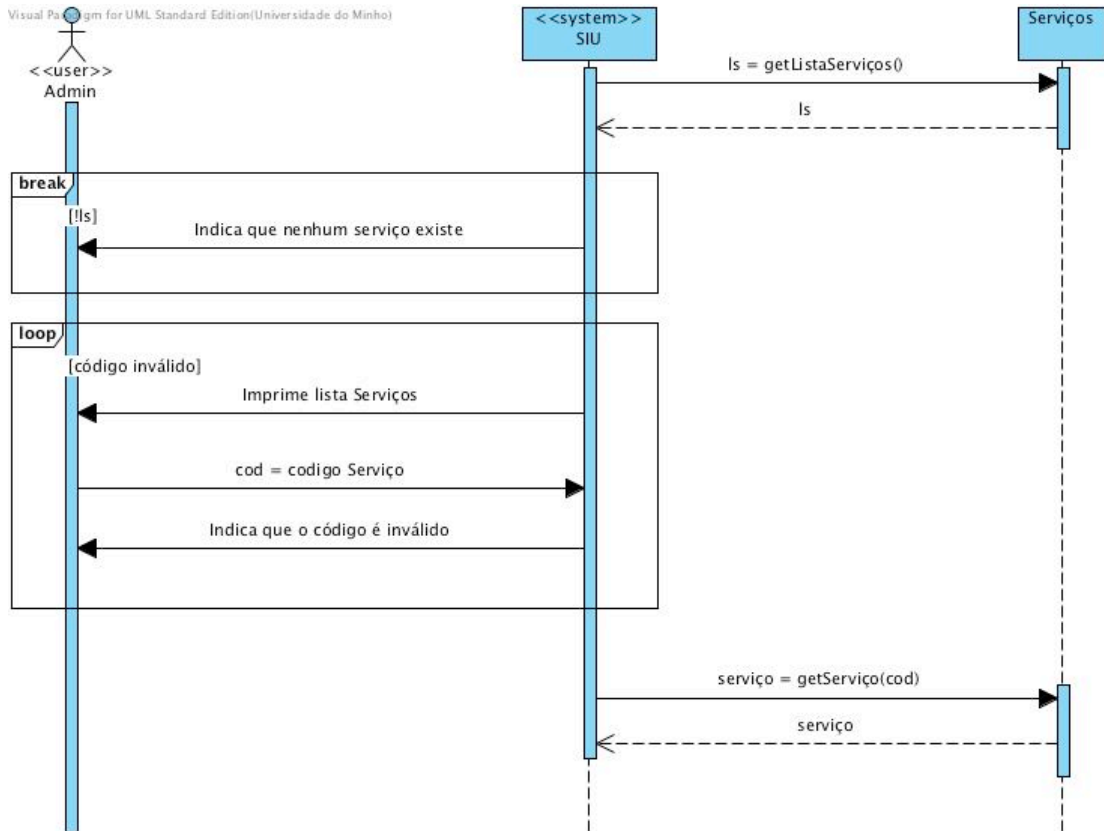


Diagramas Remover Fornecedor

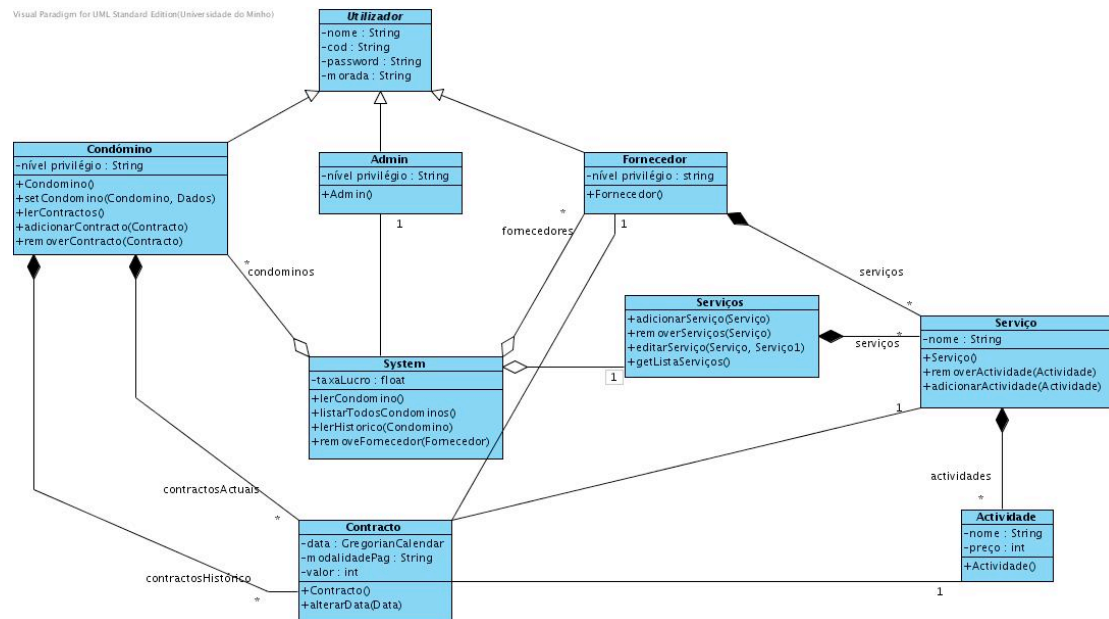


Diagramas Seleccionar Serviço



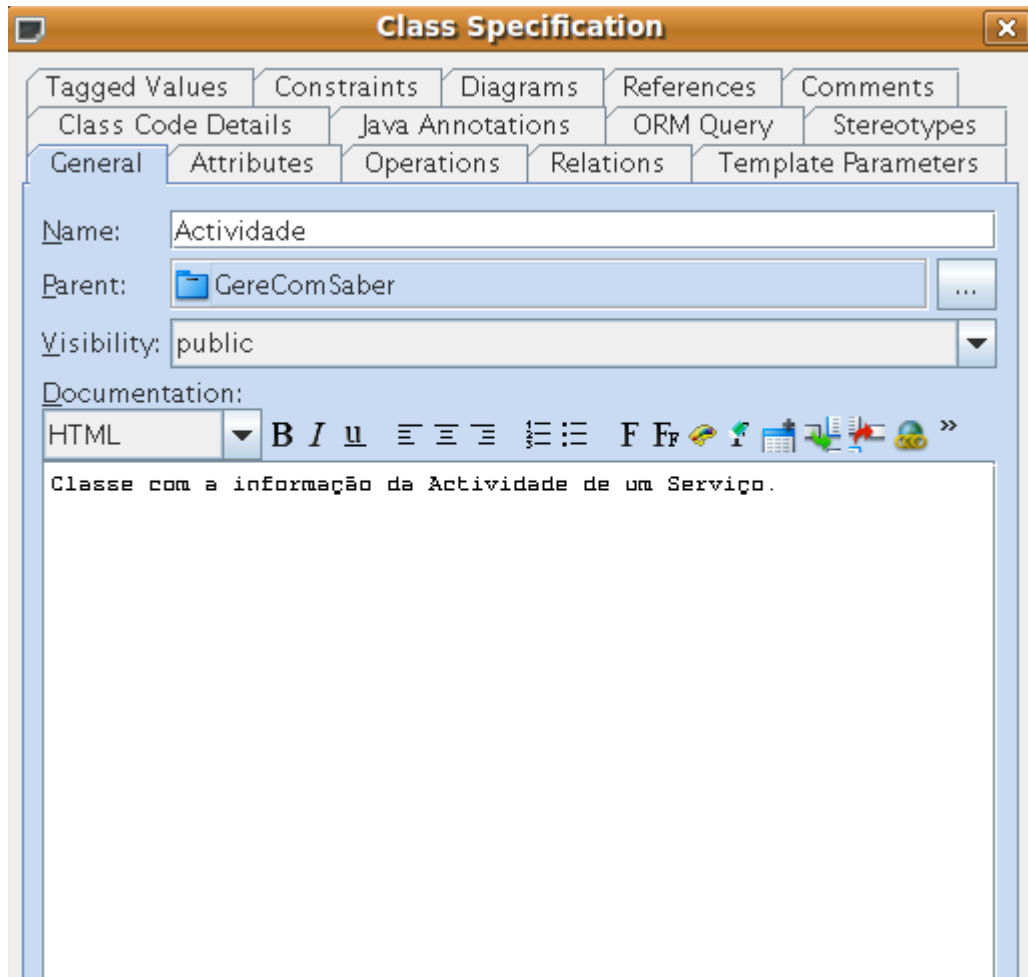


8. Diagrama de Classes



Nota: Para manter o diagrama mais compreensível, omitiu-se os métodos considerados standard: get, set, clone, equals e toString. Alguns métodos de gestão das variáveis de instância também foram omitidos como adicionar, remover, editar e consultar.

8.1. Descrição das Classes



The screenshot shows the 'Class Specification' dialog box with the following details:

- Title Bar:** Class Specification
- Tabs:** Tagged Values, Constraints, Diagrams, References, Comments, Class Code Details, Java Annotations, ORM Query, Stereotypes, General (selected), Attributes, Operations, Relations, Template Parameters.
- Name:** Admin
- Parent:** GereComSaber (with a folder icon)
- Visibility:** public
- Documentation:**
 - Format dropdown: HTML
 - Rich text toolbar: Bold (B), Italic (I), Underline (U), Bulleted List, Numbered List, Indent Left, Indent Right, Undo, Redo, Find, Print, Copy, Paste, Insert Image, Insert Link, Insert Comment, etc.
 - Text: Classe referente ao Utilizador Admin. Utilizador este que será único.

Class Specification

Tagged Values

Constraints

Diagrams

References

Comments

Class Code Details

Java Annotations

ORM Query

Stereotypes

General

Attributes

Operations

Relations

Template Parameters

Name:

Condómino

Parent:

GereComSaber

...

Visibility:

public

▼

Documentation:

HTML ▼ **B** *I* u **F** **Ff** >>

Classe referente aos Utilizadores Condómino. Cada Condómino pode ter vários contractos actuais e um histórico de contractos.

Class Specification

Tagged Values

Constraints

Diagrams

References

Comments

Class Code Details

Java Annotations

ORM Query

Stereotypes

General

Attributes

Operations

Relations

Template Parameters

Name:

Contracto

Parent:

GereComSaber

...

Visibility:

public

Documentation:

HTML

B

I

u

F

Ff

>>

O contracto é referente a um Condómino. Este tem o Fornecedor, o Serviço e a Actividade contractada.

Class Specification

Tagged Values

Constraints

Diagrams

References

Comments

Class Code Details

Java Annotations

ORM Query

Stereotypes

General

Attributes

Operations

Relations

Template Parameters

Name:

Fornecedor

Parent:

GereComSaber

...

Visibility:

public

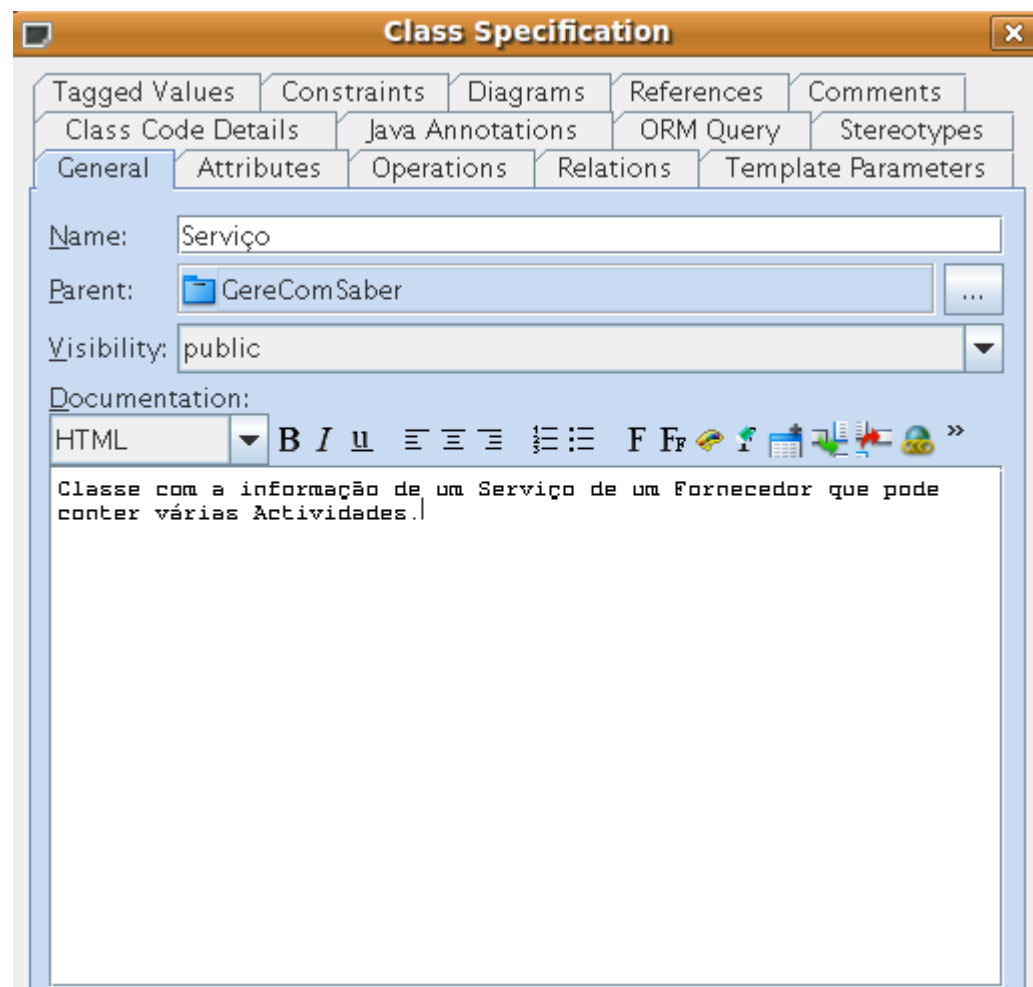
▼

Documentation:

HTML

▼

B *I* u      **F** **Ff**                           



Class Specification

Tagged Values

Constraints

Diagrams

References

Comments

Class Code Details

Java Annotations

ORM Query

Stereotypes

General

Attributes

Operations

Relations

Template Parameters

Name:

Serviços

Parent:

GereComSaber

...

Visibility:

public

▼

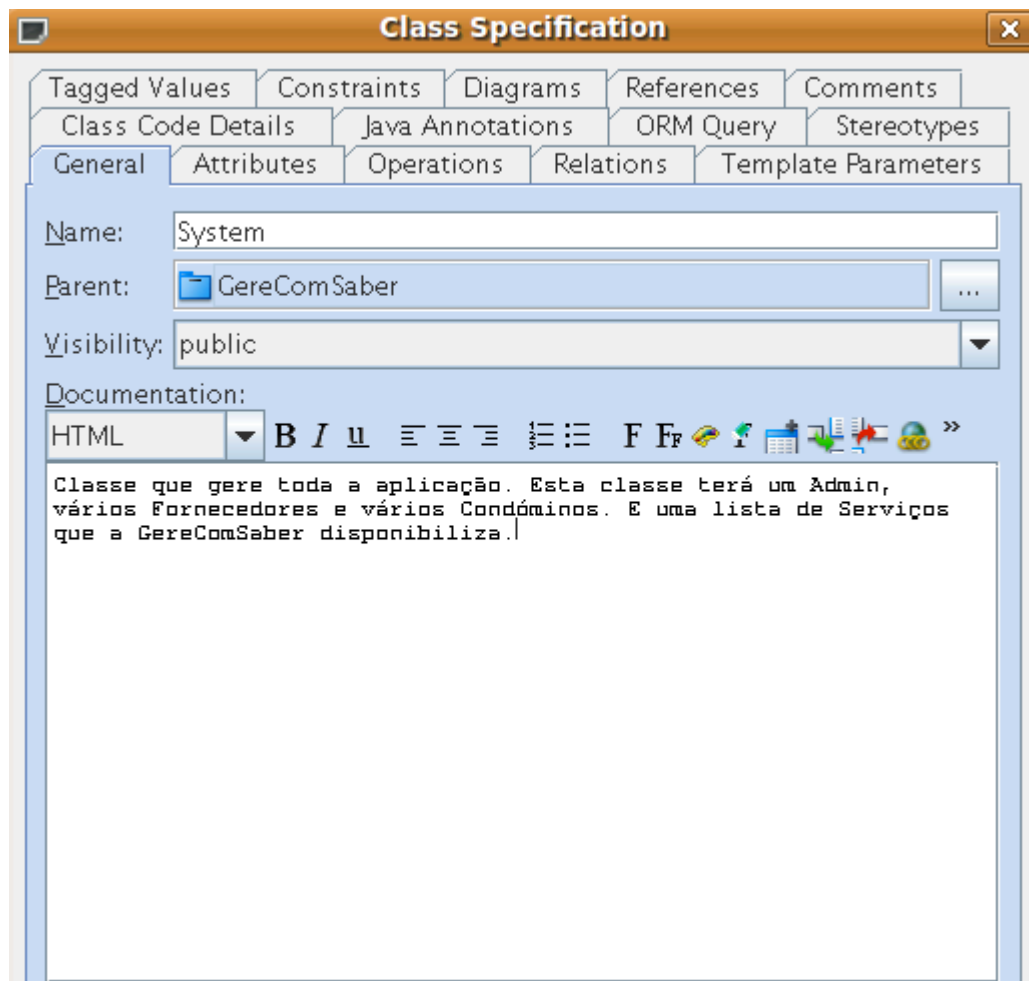
Documentation:

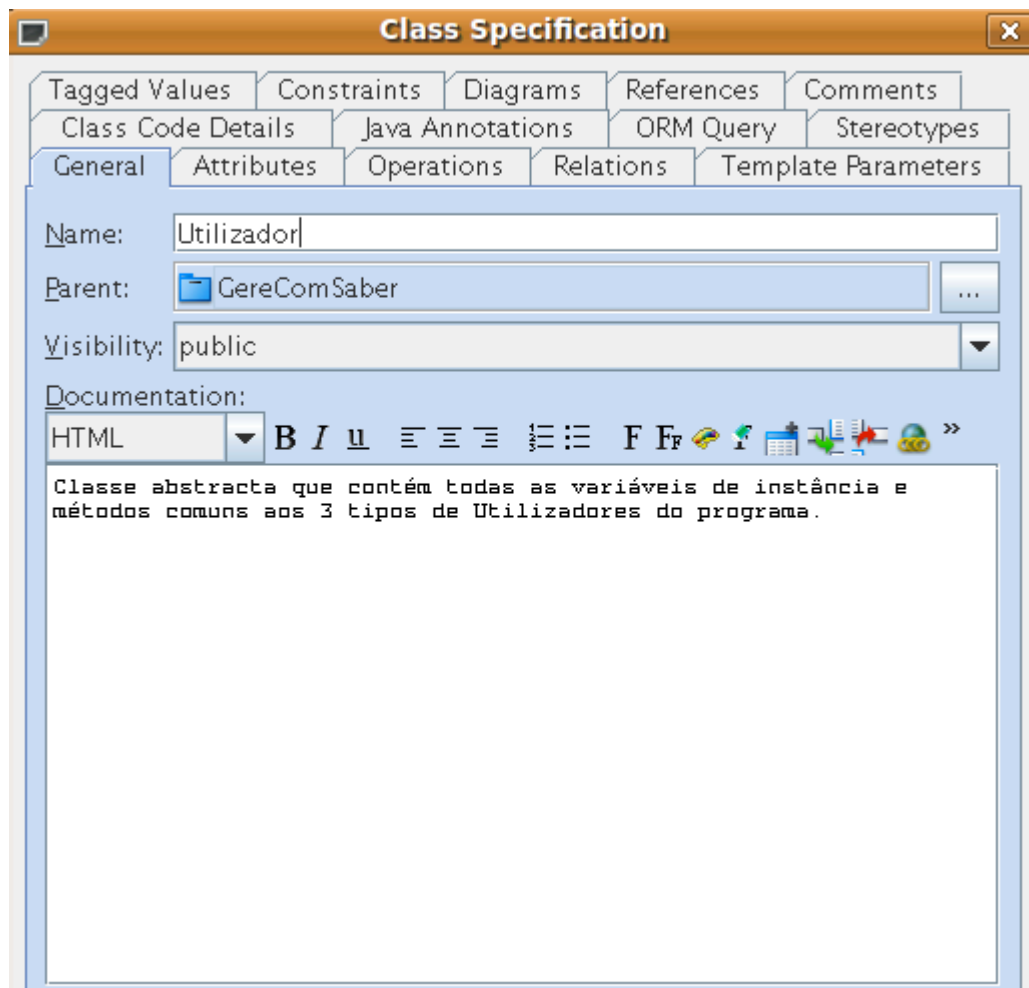
HTML

▼

B *I* u       **F** **Ff**        >>

Classe com todos os Serviços disponibilizados pela GereComSaber.





9. Conclusão

Na primeira fase de desenvolvimento nada se pôde concluir sobre o comportamento da aplicação no futuro, mas através da linguagem UML no *Visual Paradigm*, conseguiu-se construir uma estrutura base que serviu de guia para a implementação dos diagramas de sequência e do diagrama de classes. A caracterização dos *Use Cases* era um pouco vaga pois ainda não permitia testar o sistema quanto à sua robustez mas, no entanto, já se conseguia estimar as suas funcionalidades, prevendo-se um sistema com corretude e estabilidade.

Com o avançar do projecto concluiu-se que a modelação da aplicação é muito iterativa, o que possibilitou o seu refinamento durante este desenvolvimento. A partir dos diagramas de sequência foi fácil retirar os métodos necessários para a implementação da GereComSaber, e espera-se que a criação de algoritmos da aplicação seja facilitada devido a estes diagramas. Também se definiram as classes necessárias, assim como as variáveis de instância de cada classe.

Com a criação destes dois tipos de diagramas foram perceptíveis algumas incoerências na primeira fase, que tiveram de ser corrigidas. Tal desfecho era previsível, pois a este nível é muito complicado pensar em todos os cenários possíveis sem ter forma de testar a aplicação. Provavelmente, mais alterações terão de ser efectuadas com o desenrolar do desenvolvimento, tal como um modelo de desenvolvimento em cascata sugere.