



Universidade do Minho
Mestrado Integrado em Engenharia Informática

Desenvolvimento de Sistemas de Software

Ano Lectivo de 2018/2019

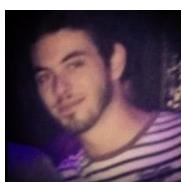
ConfiguraFácil



A76089 - Etienne Costa



A76270 - Joana Cruz



A71407 - Maurício Salgado



A72629 - Rafael Alves

DSS

Índice

1.	Introdução	3
2.	Modelação	5
2.1.	Modelo de domínio	5
2.2.	Modelo de Use Cases	7
2.3.	Protótipo da interface e máquina de estado da navegação	17
2.3.1.	Vista geral	18
2.3.2.	Acesso ao sistema - Login	19
2.3.3.	Listagem de componentes e atualização de stock	20
2.3.4.	Listagem e controlo de funcionários	21
2.3.5.	Escolha do modelo de veículo	23
2.3.6.	Configuração base	24
2.3.7.	Configuração Extra	25
2.3.8.	Registo da encomenda	26
2.3.9.	Listagem de encomendas	27
2.4.	Diagramas de sequência de sistema e subsistema	28
2.4.1.	Efetuar Autenticação	28
2.4.2.	Efetuar Configuração Base	31
2.4.3.	Adicionar Componente	33
2.4.4.	Remover componente	36
2.4.5.	Adicionar Pacote	39
2.4.6.	Remover Pacote	42
2.4.7.	Consultar funcionário	44
2.4.8.	Atualizar funcionário	46
2.4.9.	Registar funcionário	48
2.4.10.	Remover funcionário	50
2.4.11.	Consultar Encomenda	52
2.4.12.	Registar encomenda	54
2.4.13.	Gerir stock de componentes	56
2.4.14.	Efetuar Configuração Ótima	58
2.5.	Diagrama de classes com estrutura de dados e com ORM	59
2.5.1.	Diagrama de sequência de implementação	61
2.5.2.	Diagramas de sequência de implementação com DAO	70
2.6.	Diagrama de packages	73
2.7.	Diagrama de Seq. de nível de implementação (métodos mais relevantes)	
	73	
2.8.	Conclusão	78

1. Introdução

O projeto tem como objetivo a criação de uma plataforma que visa facilitar todo o processo de configuração de um veículo no ambiente de um stand de automóveis duma determinada marca bem como a sua parte “BackOffice”. Tais características implicam a necessidade de existir diferentes tipos de utilizadores para interagir com a aplicação, atendendo a tal requisito definiu-se que os tipos de utilizadores seriam:

- Funcionário do Stand;
- Funcionário da Fábrica;
- Administrador.

O processo de configuração do veículo só é permitido caso o utilizador que se encontra registado no sistema seja do tipo “Funcionário do Stand”. Este mesmo processo passa por o cliente em primeiro lugar, escolher o modelo do veículo, seguido da escolha de um conjunto de componentes base, após a configuração base segue-se uma nova etapa que consiste em o cliente escolher componentes opcionais e/ou pacotes de componentes que a marca oferece para incorporar no veículo.

Cada modelo disponibilizado pelo sistema tem os seus componentes e pacotes disponíveis (alguns partilhados pelos diversos modelos). Os componentes base não tem qualquer tipo de incompatibilidades e dependências, enquanto os componentes opcionais podem ser incompatíveis com outros, assim como podem depender de outros.

Outro aspeto tido em conta nesta plataforma é que quando os componentes escolhidos pelo cliente perfazem um pacote é disponibilizado um desconto, desconto esse que tem um valor monetário menor do que se o cliente escolhesse os mesmos componentes de forma individualizada

Por último e não menos importante, é de realçar que a plataforma dispõe da capacidade de caso o cliente opte por definir um orçamento para equipar o seu veículo. O cliente pode então dar indicação de quanto é que está disposto a gastar em equipamento opcional e sobre este mesmo plafond será gerada uma configuração extra para o veículo até que o preço total do conjunto de componentes atinja o plafond previamente indicado, esta funcionalidade é denominada por “Configuração Ótima”.

De referir, que o cliente poderá visualizar durante todo processo de configuração o valor corrente da sua configuração.

Já no final da configuração do veículo é pedido ao utilizador que insira os dados do cliente e assim registar a encomenda da configuração efetuada.

A plataforma também dispõe de uma área específica, que é apresentada aos utilizadores que são do tipo “Funcionário da fábrica”, que possibilita a visualização de todas as encomendas feitas até ao momento, bem como o estado em que as mesmas se encontram. Este mesmo tipo de utilizador também tem a função de gerir o stock dos componentes.

Por fim, existe um outro tipo de utilizador que é o “Administrador de sistema”. Para este tipo de utilizador a plataforma permite que o mesmo tenha acesso à lista de todos os utilizadores bem como registar, remover ou alterar informação dos funcionários.

Neste projeto tentou-se corresponder o máximo possível a um ambiente real do negócio em questão.

2. Modelação

2.1. Modelo de domínio

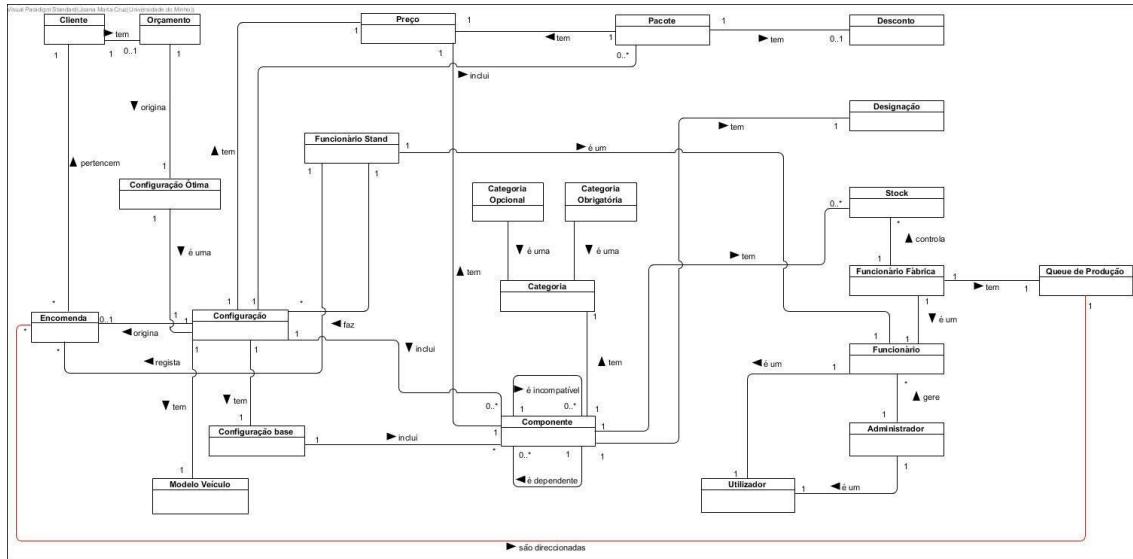


Figura 1 - Modelo de Domínio

O Modelo de domínio exibe as entidades do sistema bem como os relacionamentos estabelecidos entre as mesmas, expondo desta forma a *framework* conceptual usada para refletir sobre o problema, sendo esta a base de análise de requisitos do projeto.

O modelo de domínio inclui todas as entidades que foram necessárias para a plataforma desenvolvida, sendo as mais relevantes:

➤ Utilizador

Esta entidade encontra-se subdividida em outras 3 entidades. Entidades essas que representam diferentes tipos de utilizadores com funções e privilégios distintos, nomeadamente Funcionário do Stand, Funcionário da Fábrica e Administrador.

- Funcionário do Stand
É responsável por efetuar as configurações do veículo que o cliente desejar e submeter a encomenda para uma lista de encomendas.
 - Funcionário da Fábrica
É responsável pela verificação de todas as encomendas e gestão do stock de todos os componentes existentes na fábrica.
 - Administrador
Responsável pela gestão dos funcionários – registo de um novo funcionário, remoção de um funcionário do sistema assim como atualização de informações.

➤ **Componente**

Representa os componentes que incorporam a configuração de um veículo. É importante realçar que os componentes opcionais são os únicos que podem ter incompatibilidades e dependências, situação que não se verifica para os componentes que podem fazer parte da configuração base.

➤ **Pacote**

Trata-se de uma entidade que representa os Pacotes que podem ser incorporados na configuração dos veículos. O Pacote nada mais é do que um aglomerado de componentes com um preço mais atrativo do que quando comparado a uma escolha individual dos mesmos componentes que constituem o Pacote, o que torna possível a aplicação de desconto na configuração do cliente.

➤ **Configuração**

Esta entidade representa toda a configuração feita para o modelo escolhido pelo utilizador. Está-se a falar do registo das seguintes informações:

- Modelo do veículo;
- Escolha dos seus componentes base;
- Todos os pacotes escolhidos que o cliente quer que o seu veículo inclua;
- Componentes opcionais e por ventura todos os componentes dependentes dos demais anteriormente escolhidos.

➤ **Configuração Ótima**

É uma entidade que de certa forma é semelhante à entidade '*Configuração*', mas com a característica que visa maximizar o número de componentes de forma autónoma, isto é, após o cliente escolher o modelo e a configuração base do veículo é dada ao sistema a responsabilidade de atribuição de componentes opcionais no veículo tendo em conta um plafond máximo previamente definido pelo utilizador.

➤ **Encomenda**

A entidade em questão é responsável por guardar toda a configuração do veículo feita consoante as escolhas do cliente, informação do cliente, informação dos dados do funcionário que fez a configuração, a data em que foi submetida a encomenda, o estado em que se encontra a encomenda e o valor total a pagar pelo veículo com as devidas configurações.

➤ **Queue de Produção**

Esta entidade visa representar a listagem de todas as encomendas submetidas para o sistema bem como o estado em que as mesmas se encontram.

2.2. Modelo de Use Cases

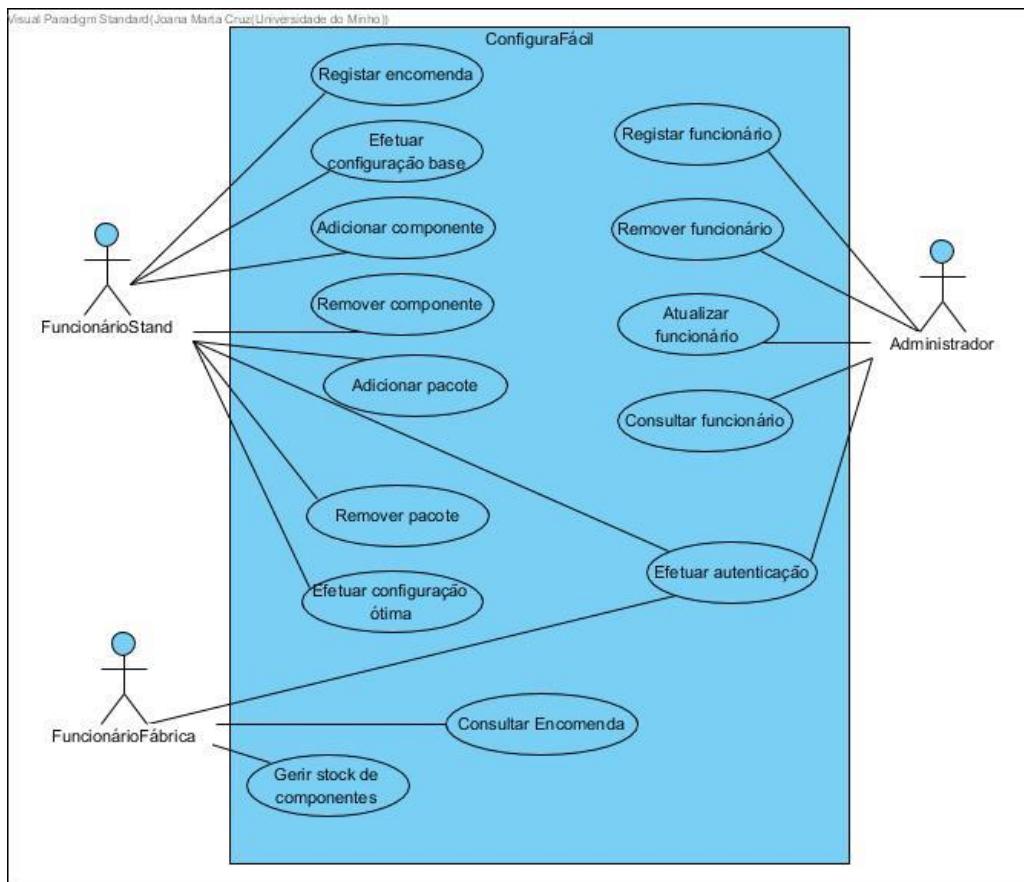


Figura 2 - Diagrama de Use Cases

Na imagem acima encontra-se o Diagrama de Use Cases que mostra de uma forma explícita os tipos de utilizadores existentes no sistema, nomeadamente, FuncionarioStand, FuncionarioFábrica e Administrador, bem como as funcionalidades de cada um na aplicação.

Também se pode observar a única operação comum entre todos os utilizadores é o 'Efetuar autenticação', pois é através da autenticação que são atribuídas as distintas funcionalidades a cada utilizador.

Quanto ao use case pedido efetuar configuração individualizada decidimos segmentar em pequenos problemas derivando daí os use case Adicionar Componente e Remover Componente. Assim, após ter sido efetuada a configuração base, o utilizador seleciona componente a componente podendo assim verificar as incompatibilidades e dependências. A mesma abordagem foi seguida para o use case pedido efetuar configuração com pacotes.

Funcionalidades do sistema

- FuncionárioStand:
 - Efetuar autenticação
 - Efetuar configuração base
 - Adicionar componente
 - Remover componente
 - Adicionar pacote
 - Remover pacote
 - Efetuar configuração ótima
 - Registrar encomenda

- ↳ FuncionárioFábrica:
 - Efetuar autenticação
 - Consultar encomenda
 - Gerir stock de componentes

- ↳ Administrador:
 - Consultar funcionário
 - Registar funcionário
 - Remover funcionário
 - Atualizar funcionário

Especificação textual para cada Use Case

Use Case:	Efectuar autenticação	
Actor:	Funcionário do Stand, Funcionário da Fábrica ou Administrador	
Pré condição:	Actor não está autenticado	
Pós condição:	Actor encontra-se autenticado	
Descrição	Realizar autenticação	
	Actor input	System response
Cenário Normal	2. Fornece credenciais	1. Pede credenciais de acesso 3. Valida credenciais 4. Informa que o utilizador está autenticado
Excepção 1 [username inexistente](passo 3)		3.1 Informa que username inserido não é válido
Excepção 2 [password incorreta](passo 3)		3.1 Informa que a password inserida está incorreta

Figura 3 - Especificação textual para o Use Case Efetuar Autenticação

Use Case:	Consultar Funcionário	
Actor:	Administrador	
Pré condição:	Funcionário existe no sistema	
Pós condição:	Funcionário consultado no sistema	
Descrição	Consultar informação de um funcionário do sistema	
	Actor input	System response
Cenário Normal	1. Pede para consultar listagem de funcionários	2. Apresenta listagem de funcionários 3. Insere username do funcionário 4. Apresenta informação do funcionário
Excepção 1[username não existe](passo 3)		2.1 Informa que o username não existe

Figura 4 - Especificação textual para o Use Case Consultar Funcionário

Use Case:	Atualizar funcionário	
Actor:	Administrador	
Pré condição:	Funcionário existe no sistema	
Pós condição:	Informação do funcionário atualizada	
Descrição	Atualizar ficha de informação de um funcionário	
	Actor input	System response
Cenário Normal	1. Insere username do funcionário 3. Altera os campos necessários	2. Apresenta informações do funcionário 4. Efetua registo 5. Informa que o funcionário não existe
Excepção 1[Funcionário inexistente](passo 1)		2.1 Informa que o ID/nome são inválidos

Figura 6 - Especificação textual para o Use Case Atualizar Funcionário

Use Case:	Remover Funcionário	
Actor:	Administrador	
Pré condição:	Funcionário existe no sistema	
Pós condição:	Funcionário removido do sistema	
Descrição	Remover um funcionário do sistema	
	Actor input	System response
Cenário Normal	1. Insere username do funcionário 4. Confirma remoção	2. Procura funcionário com esse username 3. Apresenta informações do funcionário 5. Remove funcionário do sistema 6. Informa que o funcionário foi removido
Excepção 1[ID ou nome não existem](passo 2)		2.1 Informa que o username é inválido

Figura 5 - Especificação textual para o Use Case Remover Funcionário

Use Case:	Registrar Funcionário	
Actor:	Administrador	
Pré condição:	Administrador está autenticado	
Pós condição:	Funcionario está registado	
Descrição	Registrar um funcionário	
	Actor input	System response
Cenário Normal	1. Fornece os dados do funcionário	2. Valida os dados do funcionário 3. Regista o novo funcionário 4. Informa que o funcionário foi inserido com sucesso
Excepção 1 [username já existe](passo 2)		2.1 Informa que o username inserido já está a ser utilizado

Figura 7 - Especificação textual para o Use Case Registrar Funcionário

Use Case:	Efetuar configuração base	
Actor:	Funcionário do Stand	
Pré condição:	Estar autenticado no sistema	
Pós condição:	Configuração base efetuada	
Descrição	Escolha do modelo e componentes base do veículo	
	Actor input	System response
Cenário Normal	1. Solicita os modelos disponíveis	2. Apresenta os modelos disponíveis
	3. Selecciona um modelo	4. Disponibiliza os componentes disponíveis para as categorias obrigatórias
	5. Escolhe componentes base da sua configuração	5. Regista configuração base
		3.1 Informa que é necessário escolher o modelo pretendido
		5.2 Informa que é necessário escolher todos os componentes base
Excepção 1[Utilizador não escolhe modelo](passo 3)		
Excepção 2[Utilizador não seleccionou componente de todas as categorias](passo 5)		

Figura 8 - Especificação textual para o Use Case Efetuar Configuração Base

Use Case:	Adicionar componente	
Actor:	Funcionário do Stand	
Pré condição:	Ter sido efetuada configuração base	
Pós condição:	Configuração atualizada - componente adicionada na configuração	
Descrição	Adicionar um componente durante o processo de configuração do veículo	
Cenário Normal	Actor input	System response
		1. Apresenta componentes disponíveis e ista componentes e pacotes que integram a configuração atual
	2. Seleciona componente que quer adicionar	3. Verifica se componente já integra configuração atual tanto como componente opcional ou como componente de um pacote
		4. Verifica incompatibilidades entre componente que se pretende adicionar e as que integram a configuração atual
		5. Verifica dependências entre componente que se pretende adicionar e as disponibilizadas pelo sistema
		6. Adiciona componente à configuração
		7. Informa que a componente foi adicionada
	Excepção 1 [Componente já foi adicionado] (passo 3)	
Cenário Alternativo 1 [componente incompatível] (passo 4)		4.1. Informa que a componente é incompatível com outra/s componente/s previamente escolhida/s
	4.3 Diz que sim	4.2 Pergunta se pretende remover a outra/s componente/s para puder adicionar esta
		4.4 Remove componente/s necessárias/s
		Regressa a 6
Excepção 2 [Utilizador recusa] (passo 4.2)	4.2.1 Diz que não	4.2.2 Informa que a componente não foi adicionada
Cenário Alternativo 2 [componente dependente] (passo 5)		5.1. Informa que a componente é dependente de outra/s componente/s disponibilizadas pelo sistema
	5.3 Diz que sim	5.2 Pergunta se pretende adicionar esta/s componente/s para puder adicionar esta
		5.4 Adiciona componente/s necessárias/s
		Regressa a 6
Excepção 3 [Utilizador recusa] (passo 5.2)	5.2.1 Diz que não	5.2.2 Informa que a componente não foi adicionada

Figura 9 - Especificação textual para o Use Case Adicionar Componente

Use Case:	Remover componente	
Actor:	Funcionário do Stand	
Pré condição:	Componente integra configuração atual	
Pós condição:	Configuração atualizada - Componente removido da configuração	
Descrição	Remover componente	
Cenário Normal	Actor input	System response
		1. Lista os componentes e pacotes que integram a configuração atual
	2. Seleciona componente que quer remover	3.Verifica se componente faz parte da configuração base
		4.Verifica se componente faz parte de um pacote
		5. Verifica dependências entre componente que se pretende remover e as que integram a configuração atual
		6. Remove componente da configuração
		7.Informa que componente foi removida da configuração
Excepção 1[componente é um componente base](passo 3)		3.1 Informa que não é possível remover o componente e caso o queira alterar tem que reiniciar a configuração base
Excepção 2[componente integra configuração atual na forma de pacote](passo 4)		4.1 Informa que não é possível remover o componente e caso o queira remover tem que remover o pacote por completo
Cenário Alternativo[Componente exibe dependências] (passo 5)		5.1 Informa que a componente exibe dependências com outra/s componente/s previamente escolhida/s
		5.2 Pergunta se pretende remover esta e a/s que depende/m dela
	5.3 Diz que sim	5.4 Remove componentes necessárias para depois poder remover a pedida Regressa a 6
Excepção 3 [Utilizador recusa](passo 5.2)	5.2.1 Diz que não	5.2.2 Informa que a componente não foi removida

Figura 10 - Especificação textual para o Use Case Remover Componente

Use Case:	Adicionar Pacote	
Actor:	Funcionário do Stand	
Pré condição:	Ter sido efetuada configuração base	
Pós condição:	Configuração atualizada - pacote adicionado	
Descrição	Adiciona um pacote (conjunto de componentes) á configuração	
Cenário Normal	Actor input	System response
		1. Apresenta os pacotes disponíveis e lista componentes e pacotes que integram a configuração atual
	2. Seleciona pacote que quer adicionar	3. Verifica se pacote já foi adicionado
		4. Verifica incompatibilidades entre componentes do pacote e componentes previamente escolhidos
		5. Verifica dependências entre componentes do pacote e componentes disponíveis
		6. Adiciona pacote e atualiza configuração
	Excepção 1 [pacote já foi adicionado] (passo 3.2)	3.1 Informa que o pacote já foi adicionado
Cenário Alternativo[alguma componente do pacote incompatível](passo 4)		4.2 Pergunta se pretende remover a outra/s componente/s da configuração atual para puder adicionar o pacote
	4.3 Diz que sim	4.4 Remove componente/s necessárias/s Regressa a 5
	4.2.1 Diz que não	4.2.2 Informa que o pacote não foi adicionado
Cenário Alternativo[alguma componente do pacote dependente](passo 5)		5.1. Informa que alguma/s componente/s do pacote é dependente com outra/s componente/s previamente escolhida/s
		5.2 Pergunta se pretende adicionar a outra/s componente/s para puder adicionar o pacote
	5.3 Diz que sim	5.4 Adiciona componente/s necessárias/s Regressa a 6
Excepção 3 [Utilizador recusa] (passo 5.2)	5.2.1 Diz que não	5.2.2 Informa que o pacote não foi adicionado

Figura 11 - Especificação textual para o Use Case Adicionar Pacote

Use Case:	Remover Pacote	
Actor:	Funcionário do Stand	
Pré condição:	Pacote integra configuração atual	
Pós condição:	Configuração atualizada - pacote removido	
Descrição	Remove um pacote (conjunto de componentes) da configuração	
Cenário Normal	Actor input	System response
		1. Lista os componentes e pacotes que integram a configuração atual
	2. Seleciona pacote que quer remover	3. Verifica dependências entre componentes da configuração atual e os componentes do pacote
		4. Remove pacote e atualiza configuração
		5. Informa que o pacote foi removido da configuração
Cenário Alternativo 1[alguma componente do pacote exibe dependências](passo 3)		4.1. Informa que alguma/s componente/s da configuração atual depende/m de alguma/s do pacote
		4.2 Pergunta se pretende remover a outra/s componente/s para puder remover o pacote
	4.3 Diz que sim	4.4 Remove componente/s necessárias/s Regressa a 5
Excepção 1 [Utilizador recusa] (passo 4.2)	4.2.1 Diz que não	4.2.2 Informa que o pacote não foi removido

Figura 12 - Especificação textual para o Use Case Remover Pacote

Use Case:	Registrar encomenda	
Actor:	Funcionário do Stand	
Pré condição:	Ter sido efetuada configuração base	
Pós condição:	Encomenda registada e enviada para a fila de produção	
Descrição	Funcionário confirma dados da configuração com cliente, insere dados do cliente e regista encomenda	
Cenário Normal	Actor input	System response
		1. Apresenta configuração efetuada e o respetivo preço
	2. Verifica se tem direito a desconto	
		2. Verifica se alguma combinação de componentes opcionais dá origem a um pacote
		3. Pede dados do cliente
	4. Insere dados do cliente	5. Pergunta se deseja confirmar encomenda
	6. Diz que sim	7. Efetua registo da encomenda 8. Informa que a encomenda foi registada com sucesso
Cenário alternativo[Componentes originam pacote](passo 2)		2.1 Aplica desconto sobre o preço desconto 2.2 Informa o utilizador que teve direito a desconto Regressa a 3
Excepção [Rejeita encomenda] (passo 5)	5.1 Diz que não	5.2 Cancela encomenda

Figura 13 - Especificação textual para o Use Case Registrar Encomenda

Use Case:	Consultar Encomenda	
Actor:	Funcionário da Fábrica	
Pré condição:	Estar autenticado	
Pós condição:	Encomenda consultada	
Descrição	Consulta a fila de encomendas, e os detalhes de uma encomenda	
Cenário Normal	Actor input	System response
	1. Pede para consultar fila de encomendas	2. Apresenta listagem das encomendas
	3. Insere ID da encomenda	4. Apresenta informações da encomenda
	5. Seleciona detalhes da encomenda	5. Apresenta detalhes da encomenda
		3.1 Informa que o ID da encomenda não existe
Excepção [ID não existe](passo 3)		

Figura 14 - Especificação textual para o Use Case Consultar Encomenda

Use Case:	Gerir Stock de componentes	
Actor:	Funcionário da fábrica	
Pré condição:	Estar autenticado no sistema	
Pós condição:	Stock atualizado	
Descrição	Atualiza o stock de componentes	
Cenário Normal	Actor input	System response
	1. Pede para consultar componentes disponíveis	2. Apresenta tabela com a listagem dos componentes
	3. Insere ID do componente	4. Apresenta detalhes do componente
	5. Altera o stock	6. Efetua registo 4. Notifica que o stock foi alterado
		2.1 Informa que o componente não existe
Excepção 1[ID não existe](passo 1)		

Figura 15 - Especificação textual para o Use Case Gerir Stock de Componentes

Use Case:	Efetuar Configuração Ótima	
Actor:	Funcionário do Stand	
Pré condição:	Ter sido efetuada a configuração base	
Pós condição:	Configuração ótima efetuada	
Descrição	Maximização do número de componentes opcionais na configuração conforme o seu preço e dos seus dependentes	
Cenário Normal	Actor input	System response
	1. Pede para efetuar configuração ótima	2. Solicita orçamento que quer disponibilizar
	3. Insere orçamento	4. Maximiza número de componentes conforme o seu preço e dos seus dependentes dentro do orçamento disponibilizado
		5. Apresenta configuração otimizada

Figura 16 - Especificação textual para o Use Case Efetuar Configuração Ótima

2.3. Protótipo da interface e máquina de estado da navegação

As Máquinas de estado têm o objetivo de expor o comportamento do Sistema. Através das mesmas é possível exibir todos os estados que um Sistema atravessa face aos ocorridos.

Neste projeto optou-se pela implementação destes diagramas na interface com o utilizador, independentemente do seu estatuto/privilégios para com o Sistema.

Quanto à prototipagem optamos por uma de baixo nível.

Deste modo, especifica-se os vários estados da interface para os seguintes cenários:

- Vista geral de todas as capacidades e fluxo entre as mesmas suportam o sistema
- Acesso ao sistema através da inserção das devidas credenciais
- Procura e atualização do stock de um componente
- Listagem, procura, criação, edição e eliminação de um funcionário
- Apresentação dos modelos do veículo e sua seleção (1º passo da configuração Base)
- Apresentação e escolha dos diversos componentes base que integram o modelo previamente selecionado (2º passo da configuração base)
- Escolha ou não de Componentes Complementares ou Pacotes e todas as implicações que os mesmos impõem (Opcional para o utilizador – 3º passo da configuração do veículo)
- Registo e submissão da encomenda para a fila de espera de produção bem como a aplicação ou não de descontos
- Listagem de encomendas

Com a realização destes diagramas verificou-se uma maior propensão a uma correta e mais fundamentada estruturação da forma como toda a interface interage com o utilizador conforme as decisões tomadas pelo mesmo (utilizador).

2.3.1. Vista geral

É importante referir que o programa iniciasse sempre num estado não autenticado, o qual, obriga todos os utilizadores que queiram interagir com o sistema tenham que previamente autenticar-se. Depois do utilizador estar devidamente autenticado, este terá acesso a determinados estados dependendo dos privilégios atribuídos ao tipo de utilizador identificado.

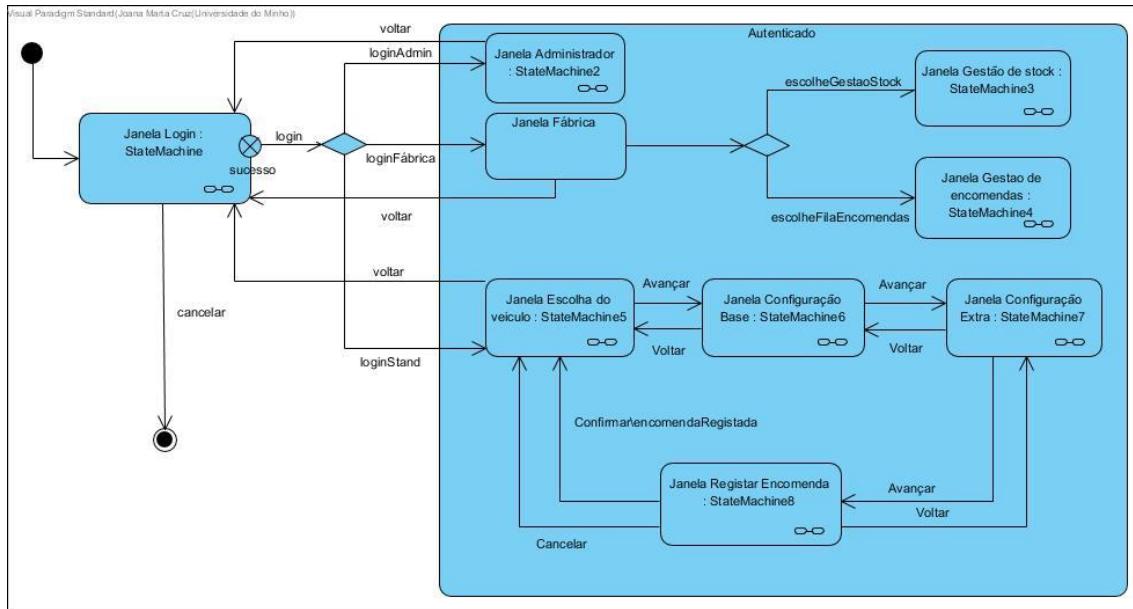


Figura 17 - Máquina de estado geral

2.3.2. Acesso ao sistema - Login

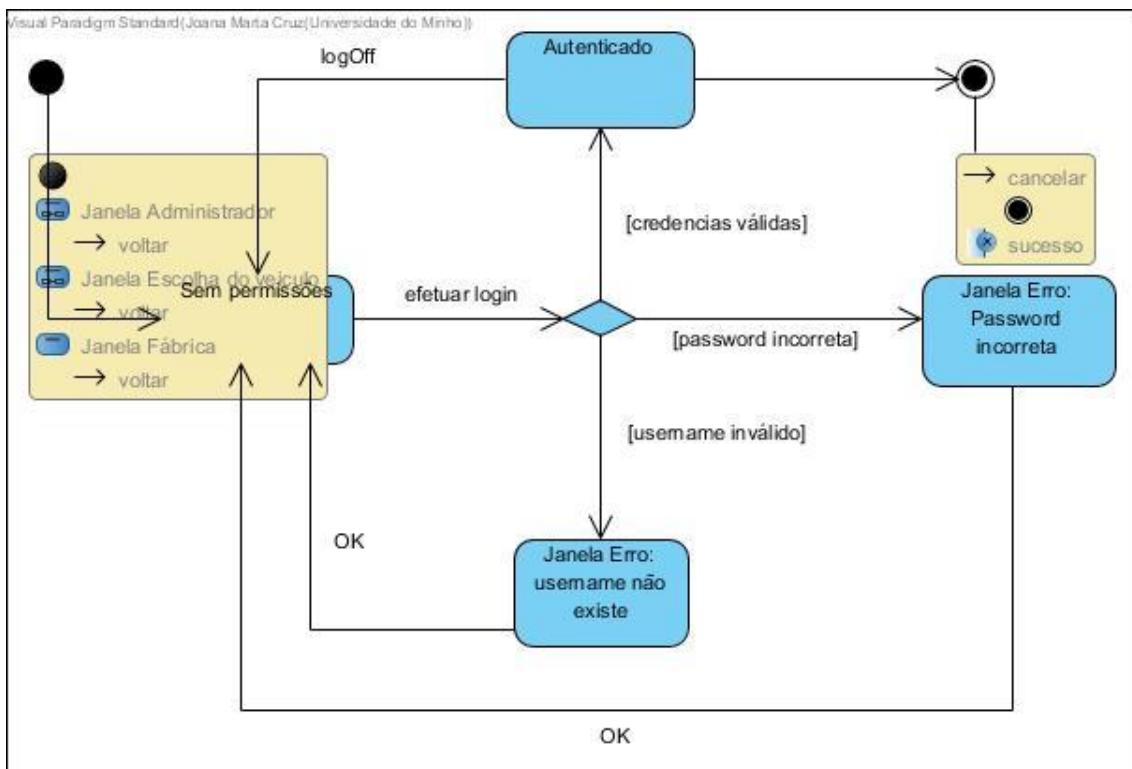
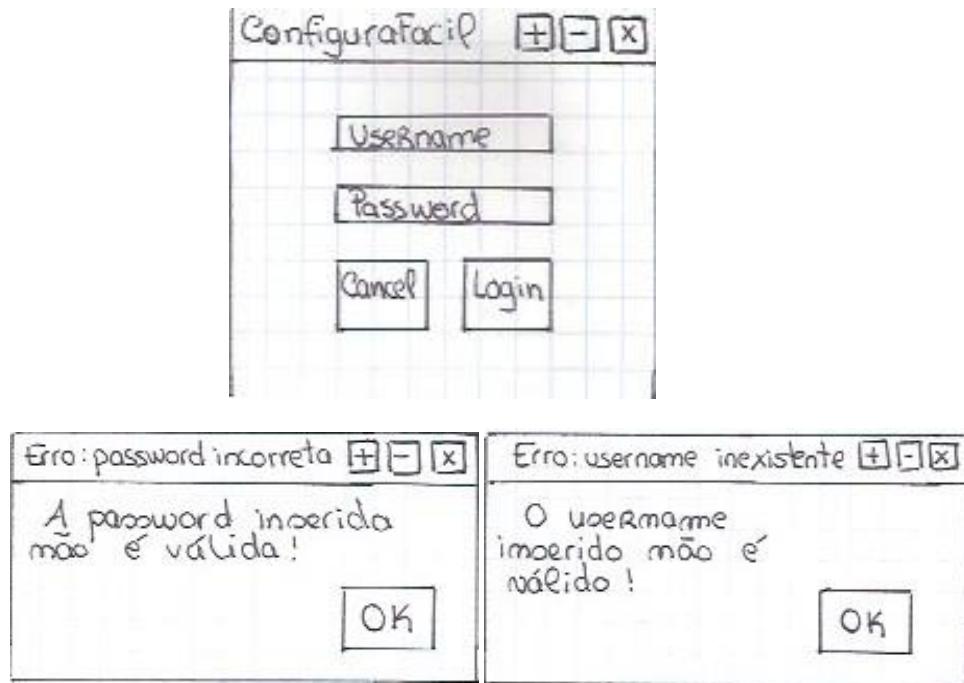


Figura 18 - Máquina de estado janela login

2.3.3. Listagem de componentes e atualização de stock

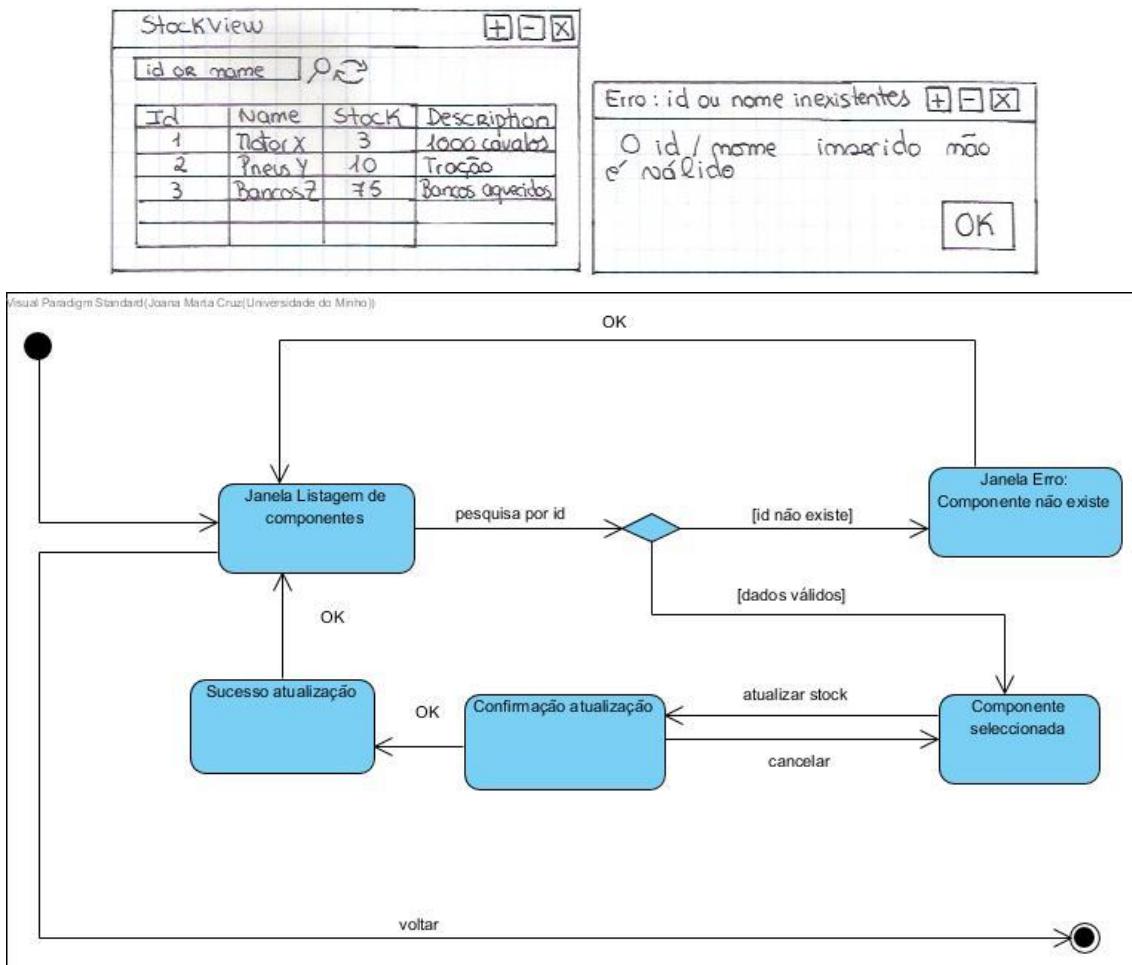


Figura 19 - Máquina de estado gestão de stock de componentes

2.3.4. Listagem e controlo de funcionários

AdminView

id or name				
<input type="text"/> P - R +				
ID	Name	Department	Address	Contact
1	Joano	Stand	Praceta	916235555
2	Etienne	Fábrica	Rua Flores	914546789

Voltar

Erro : id ou nome inexistentes	Aviso : Confirmação Remoção	Sucesso
O id / nome inserido não é válido	Pretende realmente eliminar o funcionário?	O funcionário foi removido com sucesso!
<input type="button" value="OK"/>	<input type="button" value="Cancel"/> <input type="button" value="OK"/>	<input type="button" value="OK"/>

Aviso : Confirmação atualização	Sucesso
Prende realmente alterar a informação do funcionário?	A informação do funcionário foi atualizada com sucesso!
<input type="button" value="Cancel"/> <input type="button" value="OK"/>	<input type="button" value="OK"/>

InsertView	Erro : Dados Incompletos	Erro : Id existente
<input type="text" value="Número"/> <input type="text" value="Name"/> <input type="text" value="Department"/> <input type="text" value="Address"/> <input type="text" value="Contact"/> <input type="button" value="Cancel"/> <input type="button" value="OK"/>	É necessário inserir todos os dados!	O funcionário com esse ID já existe!
	<input type="button" value="OK"/>	<input type="button" value="OK"/>

Sucesso
O funcionário foi inserido com sucesso!
<input type="button" value="OK"/>

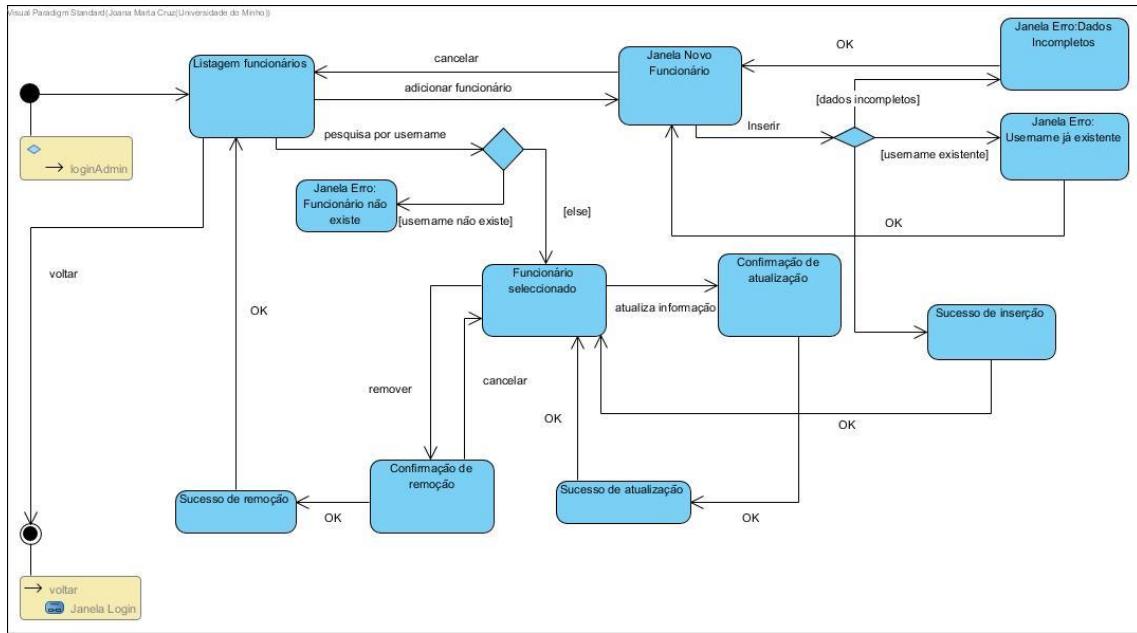


Figura 20 - Máquina de estado janela administrador

2.3.5. Escolha do modelo de veículo

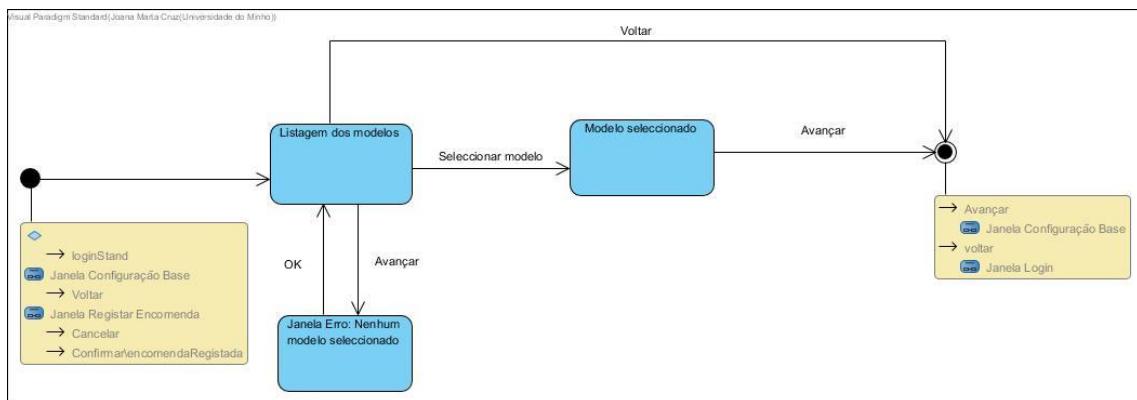
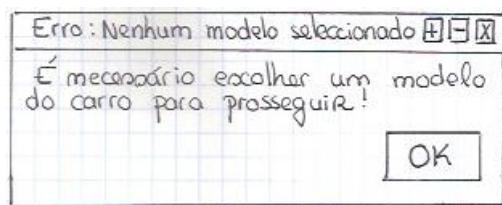
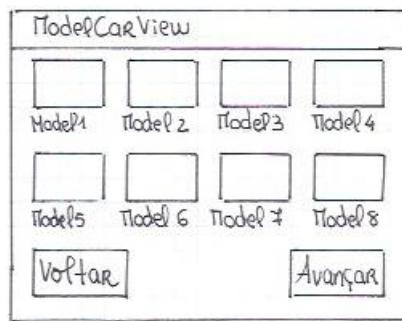


Figura 21 - Máquina de estado janela escolha do veículo

2.3.6. Configuração base

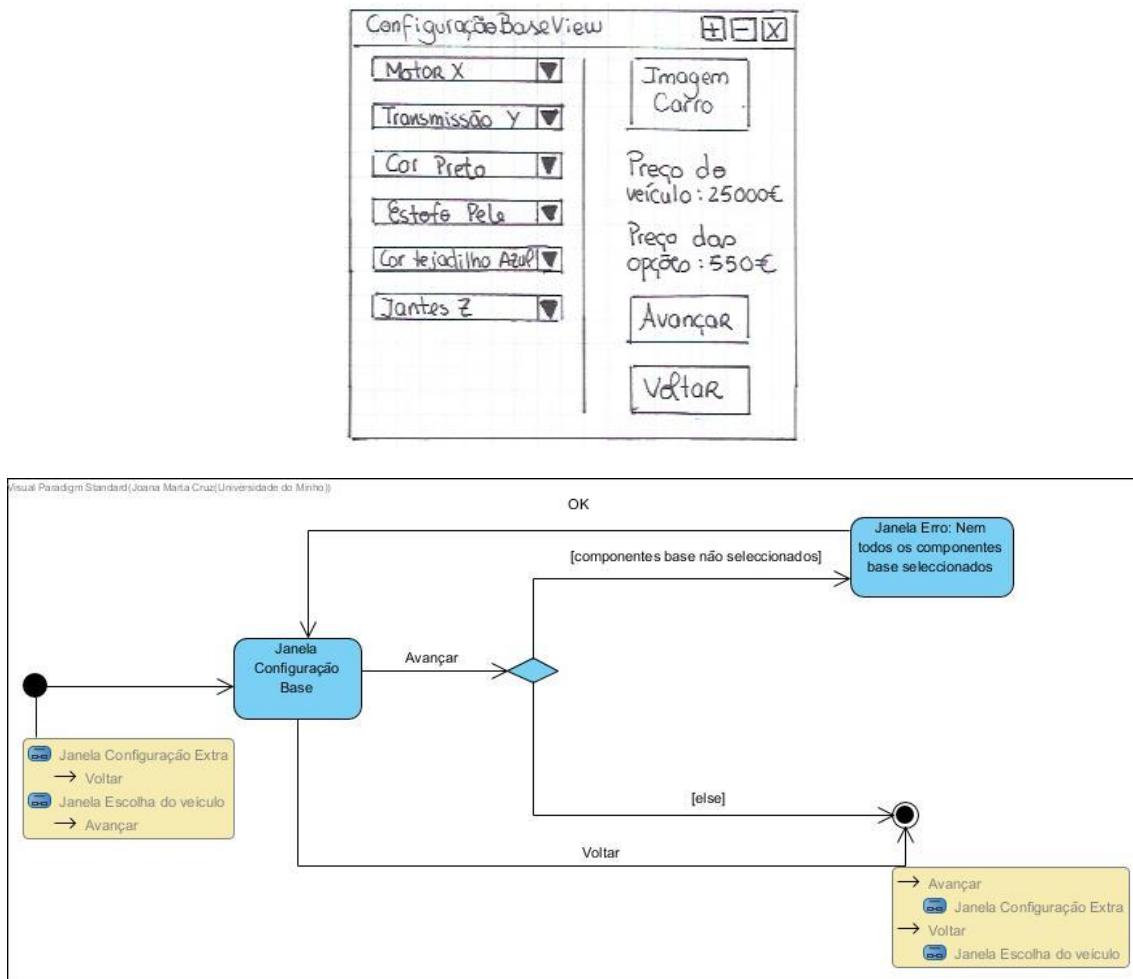


Figura 22 - Máquina de estado janela configuração base

2.3.7. Configuração Extra

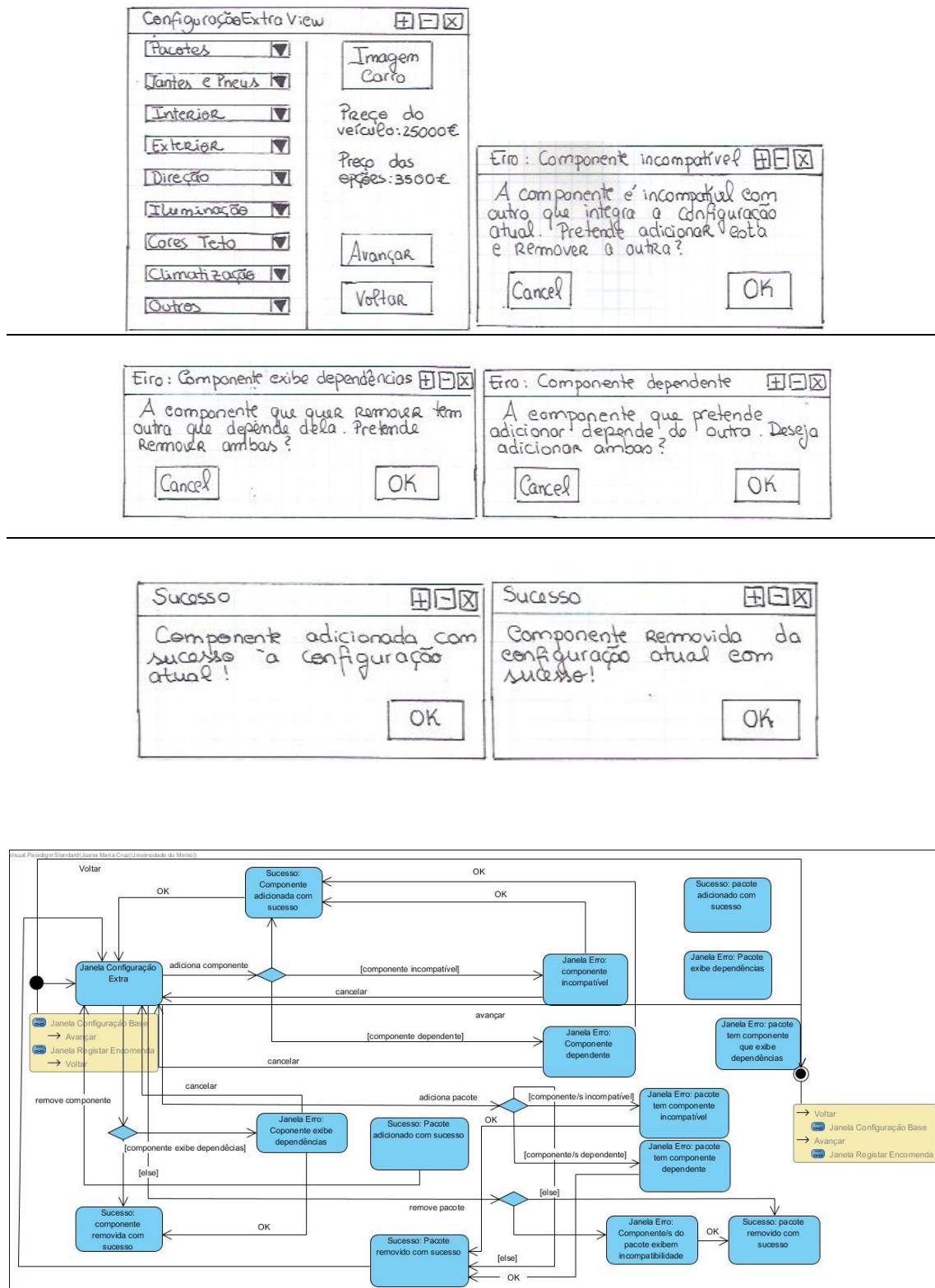
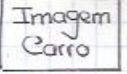


Figura 23 - Máquina de estado efetuar configuração extra

2.3.8. Registo da encomenda

RegistaEncomendaView

 Imagem Carro Preço do veículo: 25 000€ Preço das opções: 3500€ Opções selecionadas: <ul style="list-style-type: none"> - Opção 1 - Opção 2 - Opção 3 - Opção 4 - Opção 5 - Opção 6 <input type="button" value="Verifica Desconto"/>	Dados do cliente: <input type="text" value="Name"/> <input type="text" value="Morada"/> <input type="text" value="Contacto"/> <input type="button" value="Confirmar"/> <input type="button" value="Rejeitar"/>
---	---

<p>Aviso: Direito a desconto</p> <p>Aplicou-se um desconto sobre o preço devido a algumas opções escolhidas!</p> <p><input type="button" value="OK"/></p>	<p>Erro : Dados Incompletos</p> <p>É necessário inserir todos os dados !</p> <p><input type="button" value="OK"/></p>
--	--

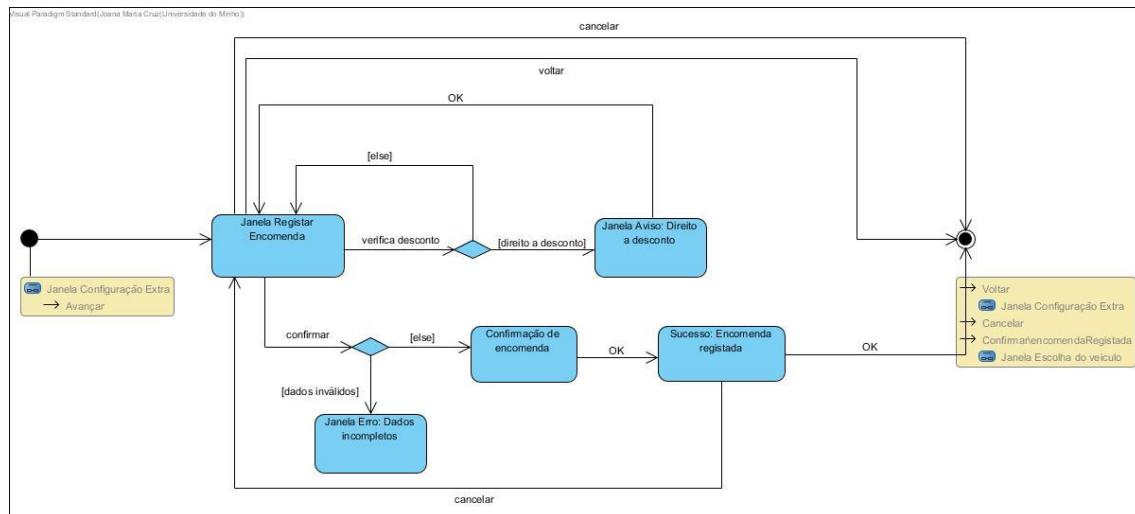


Figura 24 - Máquina de estado janela registrar encomenda

2.3.9. Listagem de encomendas

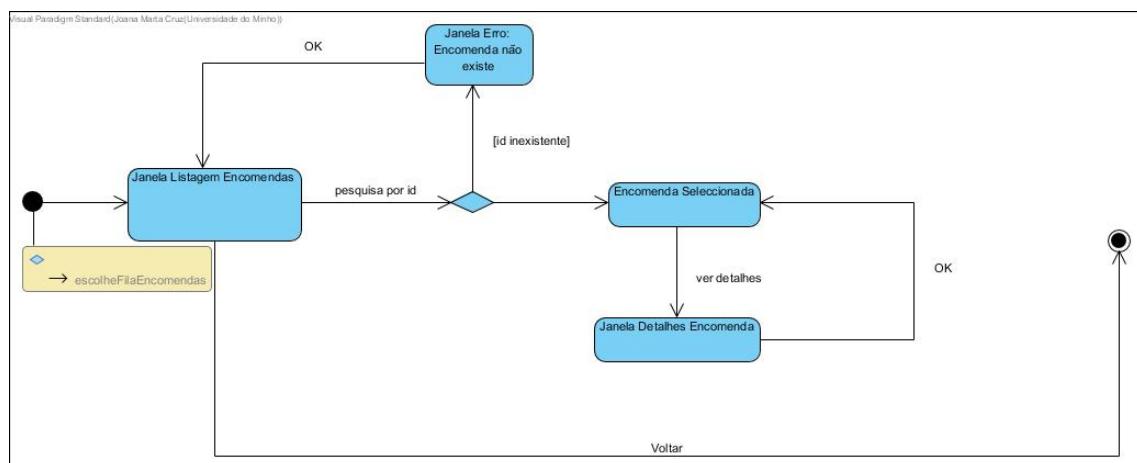
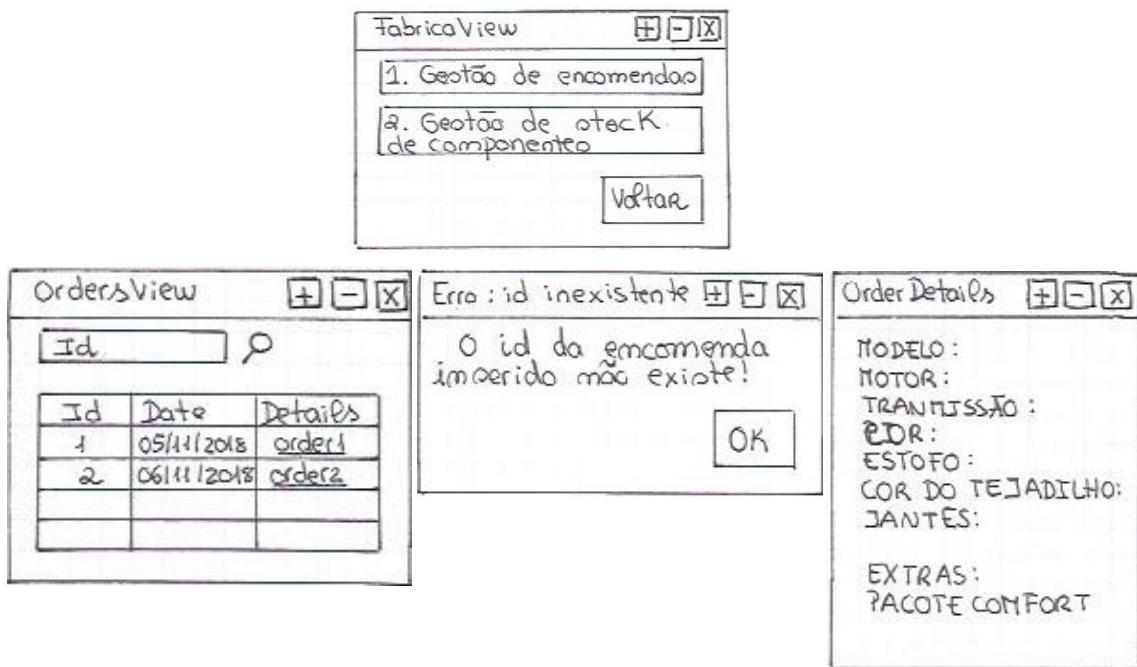


Figura 25 - Máquina de estado janela listagem encomendas

2.4. Diagramas de sequência de sistema e subsistema

Com os diagramas de sequência de sistema consegue-se expor, de uma forma mais atómica, como funciona a interação sistema-utilizador.

Já nos diagramas de subsistema, depois de estruturadas as classes que o programa iria necessitar, dá-se início á etapa em que se pensa nos métodos que vão ser incluídos em cada classe, pegando nos diagramas de sequência de Sistema já feitos e estendendo cada um ao respetivo Subsistema.

Assim, abaixo, para cada Diagrama de Sequência de Sistema, imediatamente a seguir, tem-se a respetiva extensão ao Diagrama de Sequência de Subsistema.

Tornando-se assim mais fácil visualizar como tem de ser feito o programa e como este irá conectar-se à interface do Utilizador.

2.4.1. Efetuar Autenticação

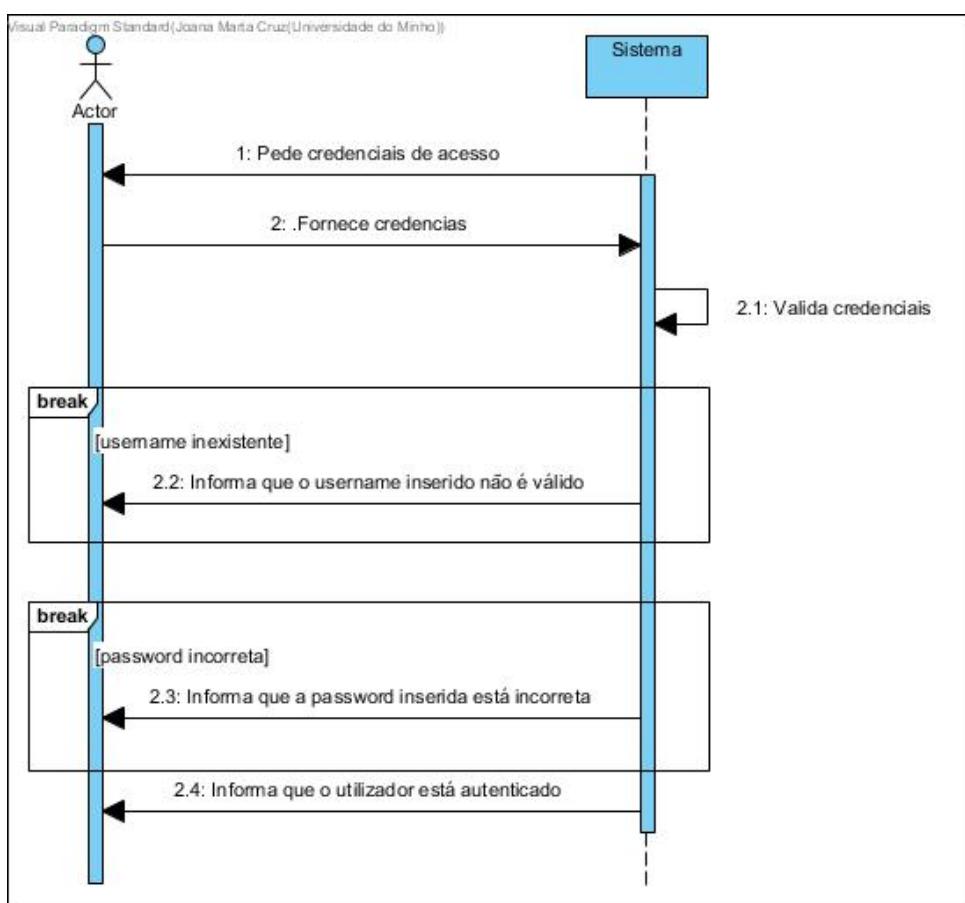


Figura 26 - Diagrama de sequência de sistema

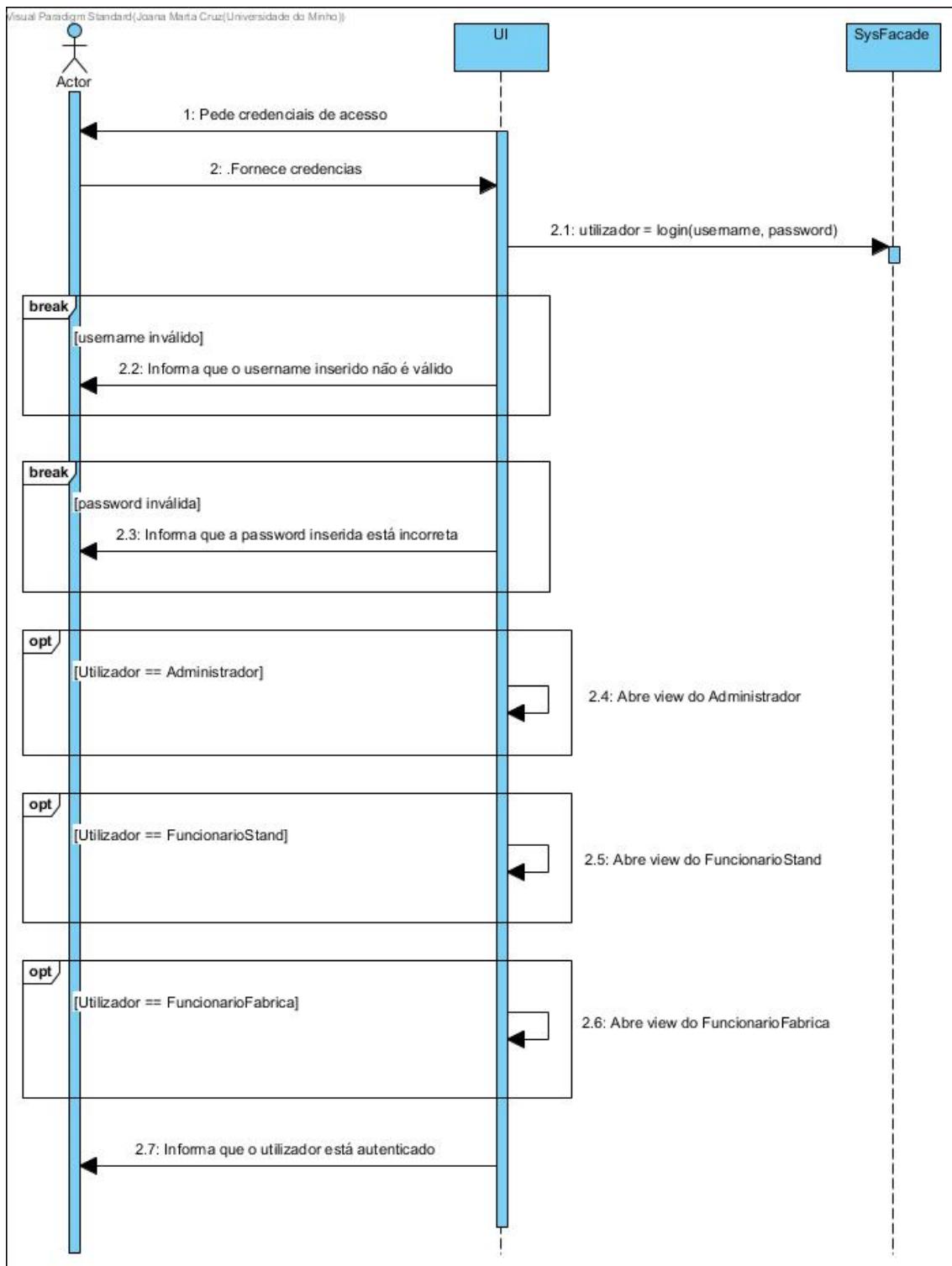


Figura 27 - Diagrama de sequência com UI e Facade

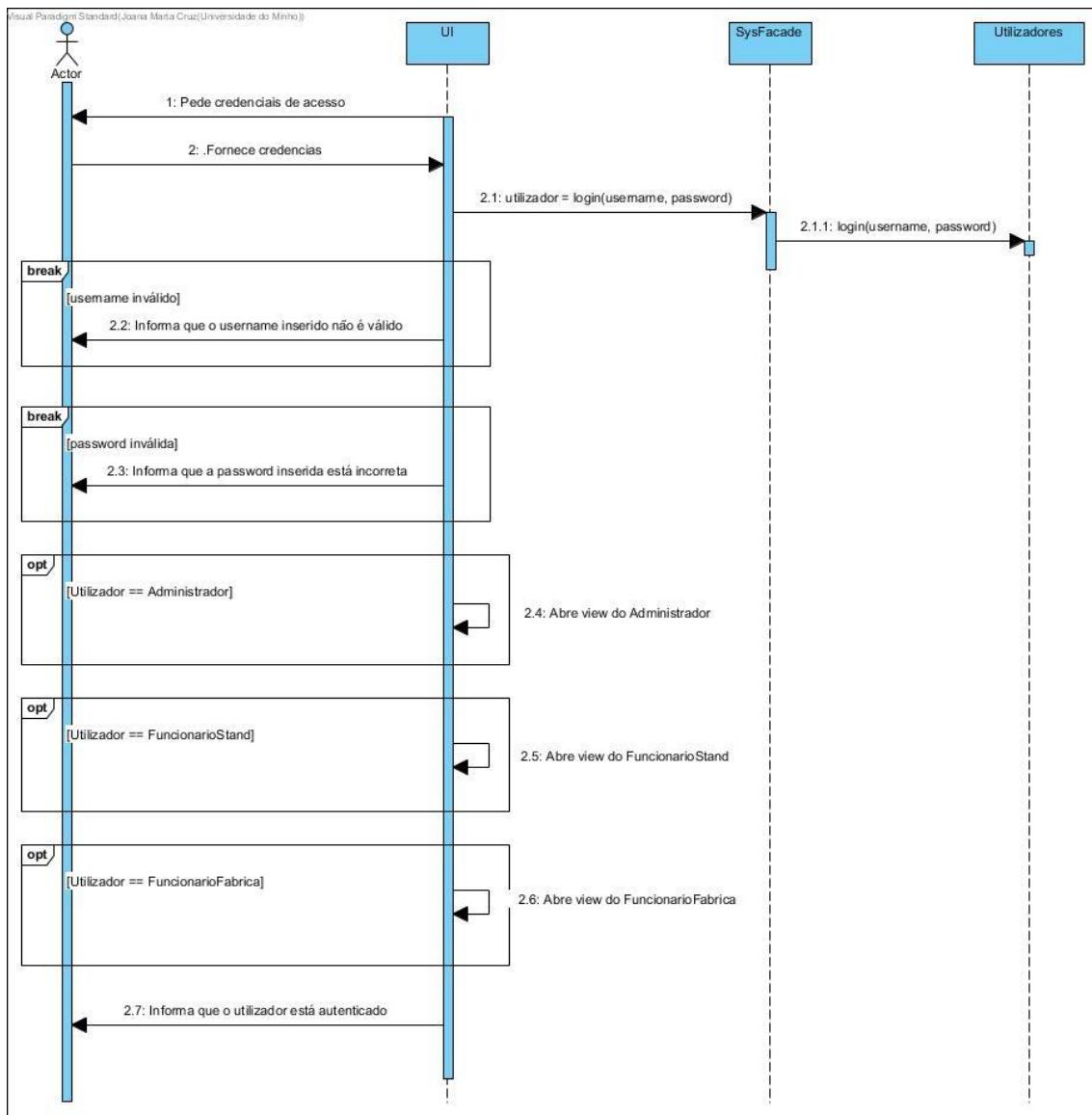


Figura 28 - Diagrama de sequência com subsistemas

2.4.2. Efetuar Configuração Base

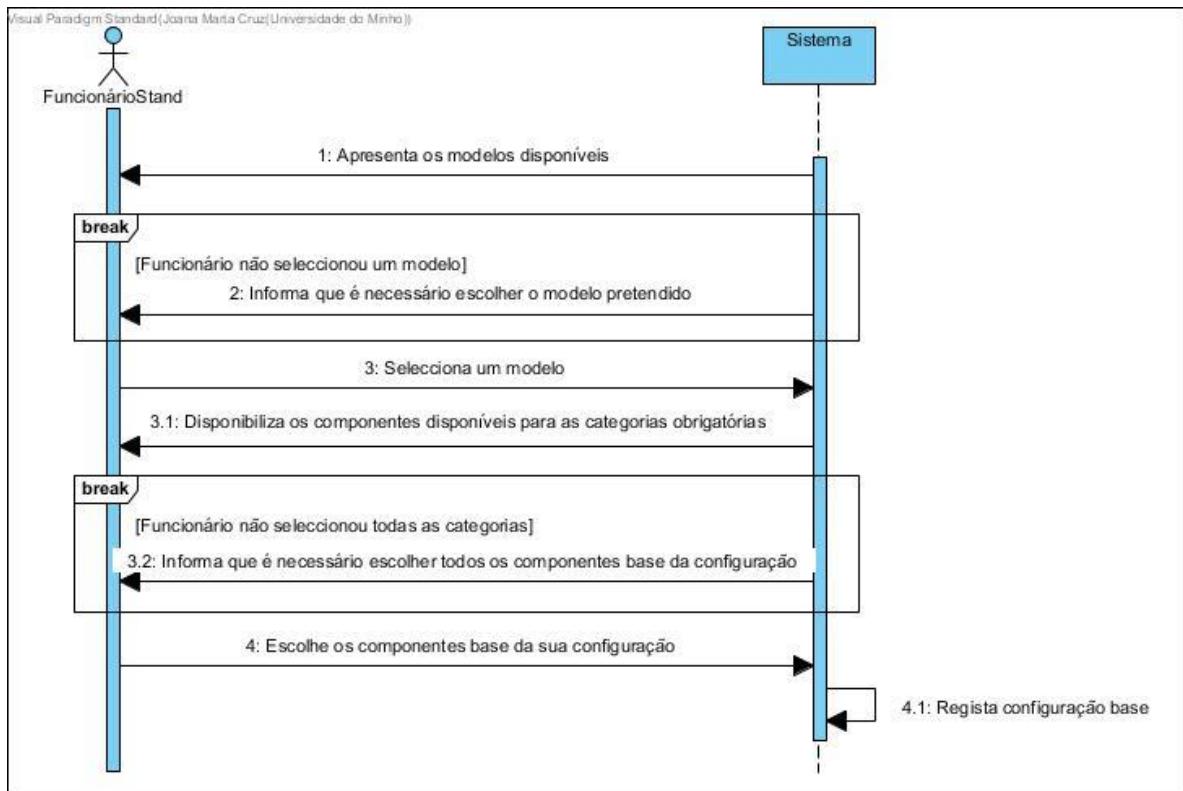


Figura 29 - Diagrama de sequência de sistema

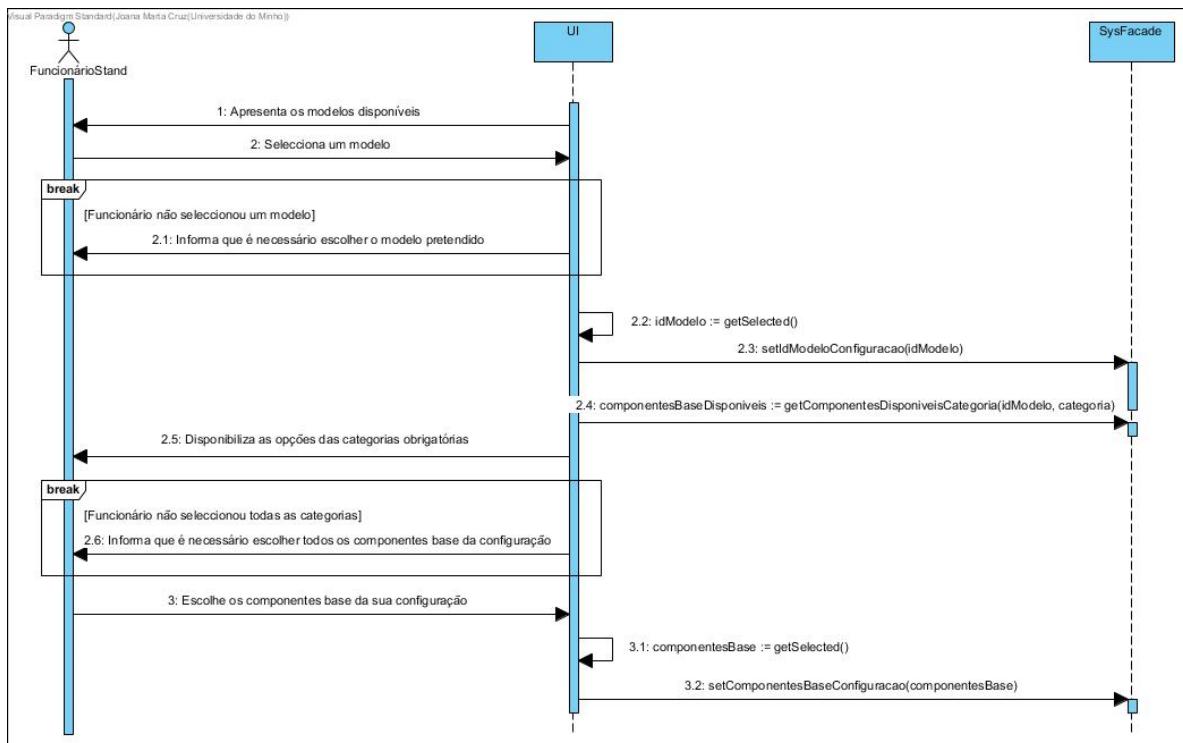


Figura 30 - Diagrama de sequência com UI e Facade

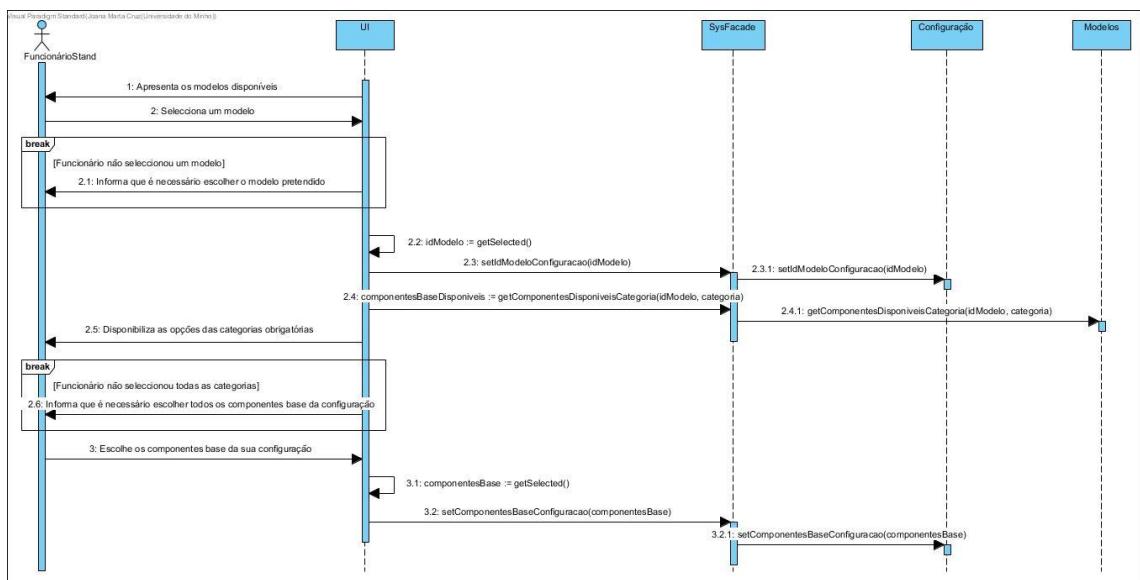


Figura 31 - Diagrama de sequência com subsistemas

2.4.3. Adicionar Componente

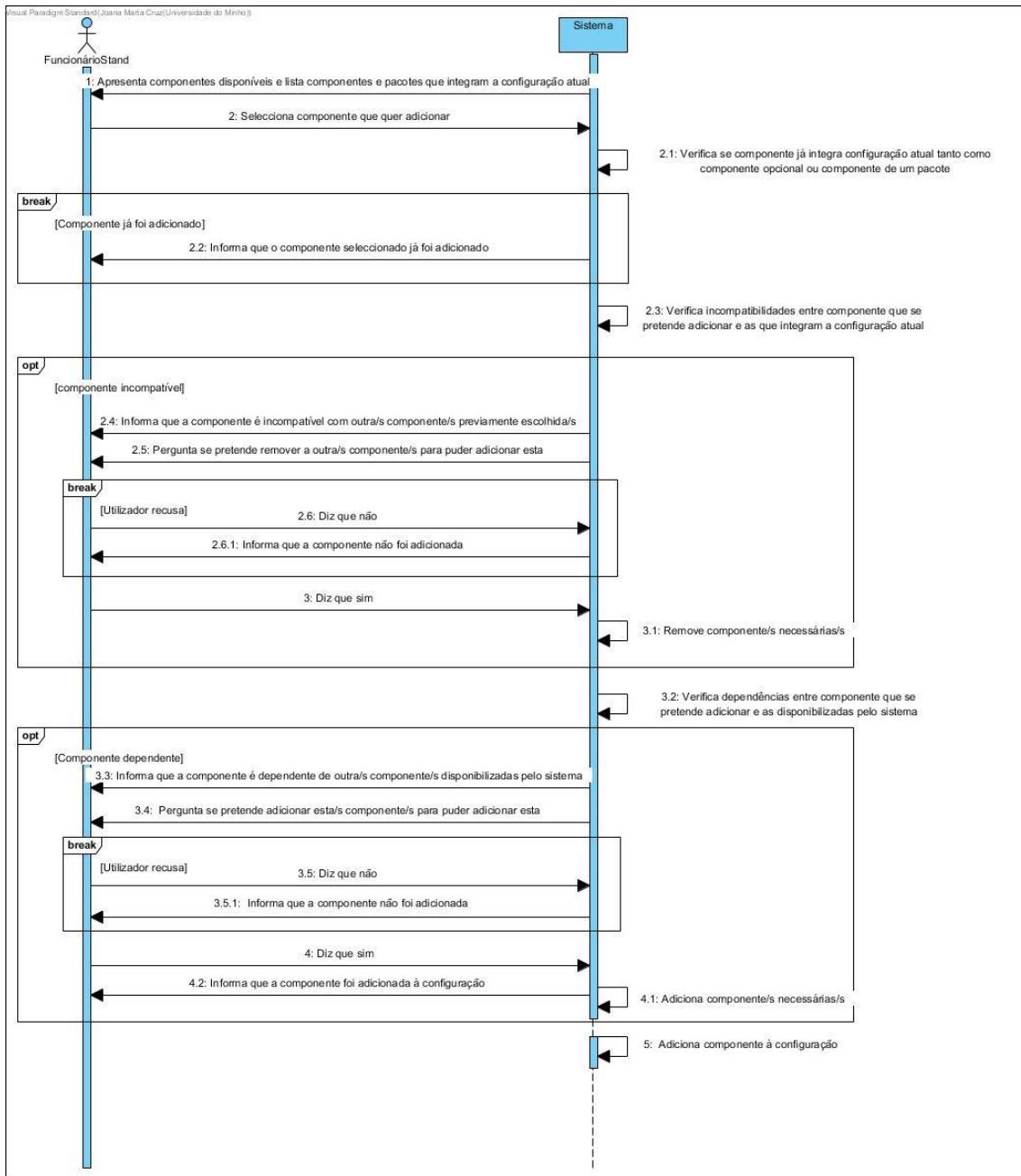


Figura 32 – Diagrama de sequência de sistema

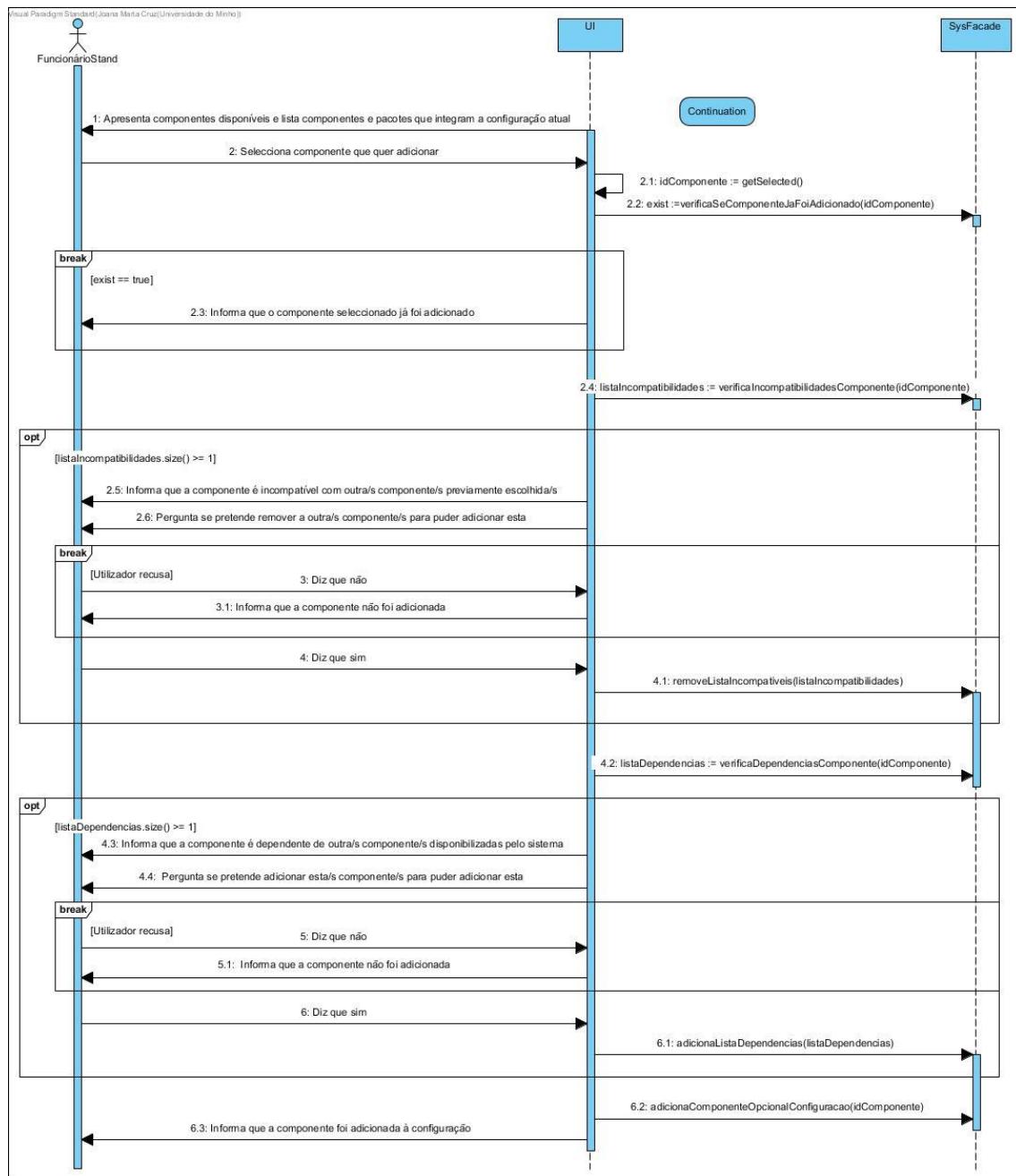


Figura 33 – Diagrama de sequência com UI e Facade

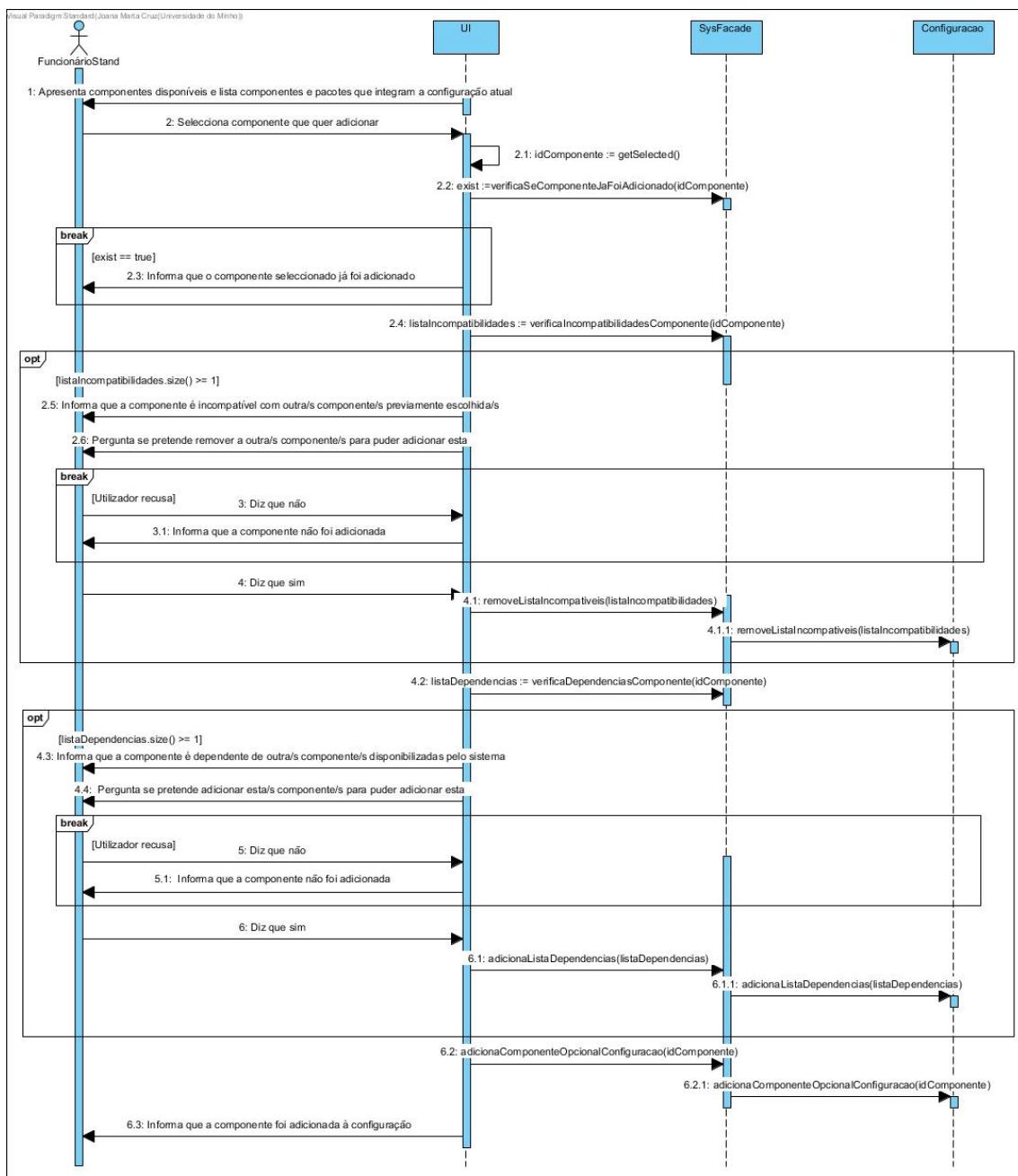


Figura 34 – Diagrama de sequência com subsistemas

2.4.4. Remover componente

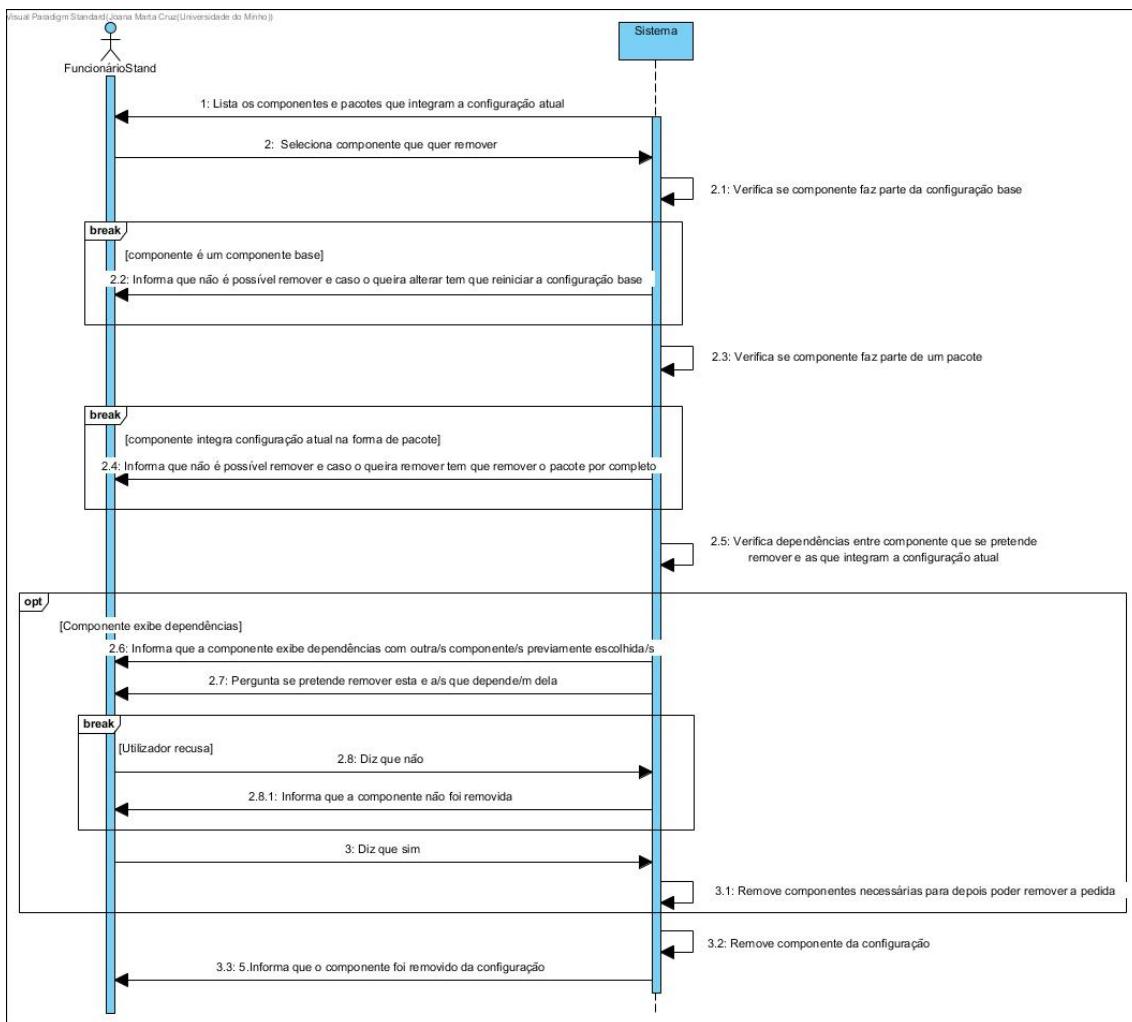


Figura 35 - Diagrama de sequência de sistema

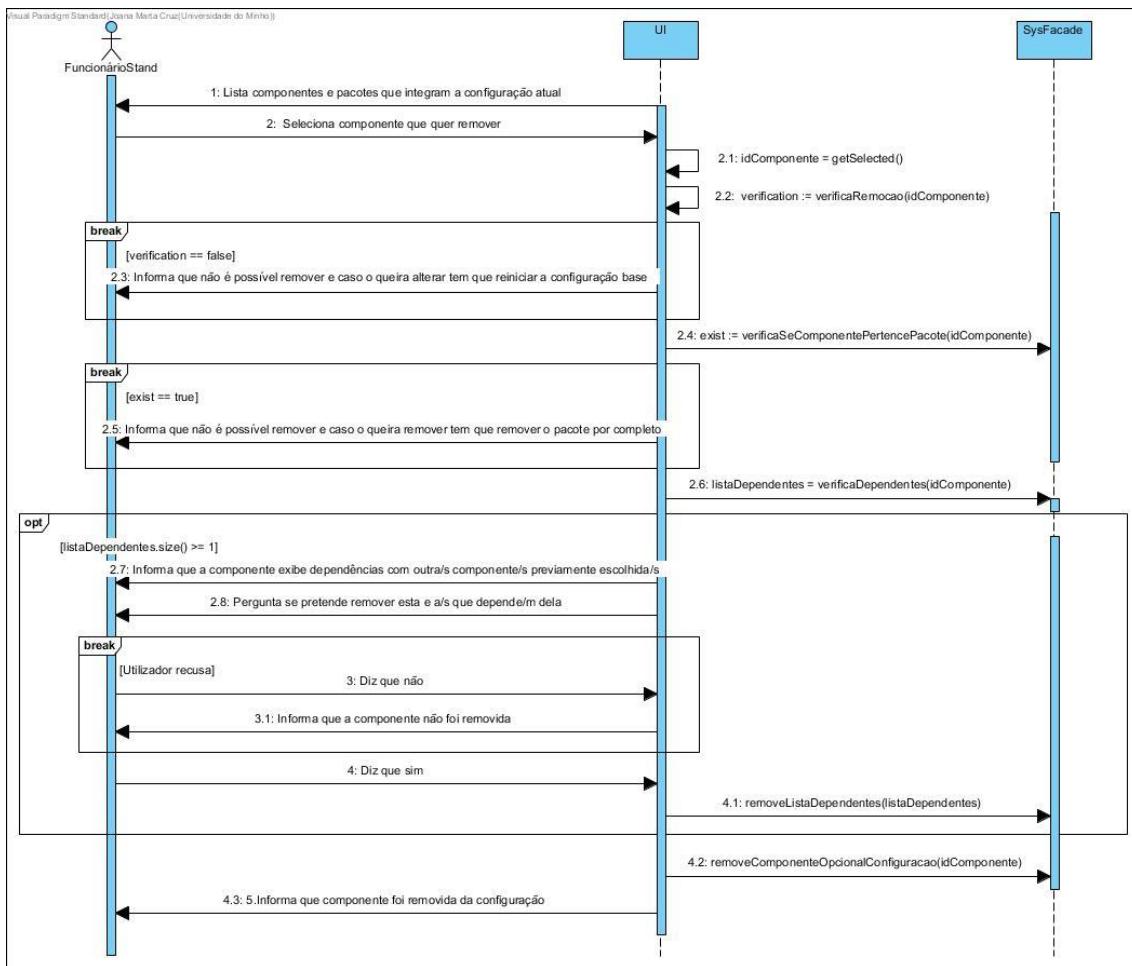


Figura 36 - Diagrama de sequência com UI e Facade

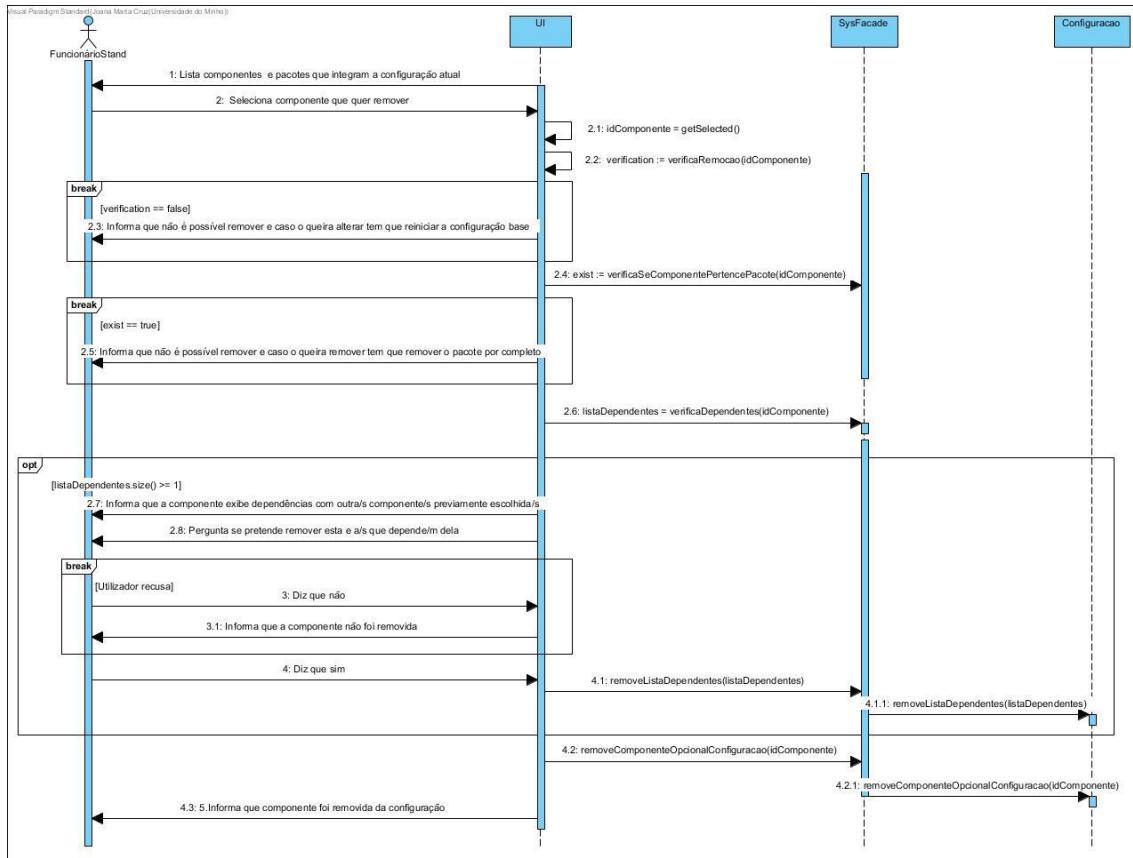


Figura 37 - Diagrama de sequência com subsistemas

2.4.5. Adicionar Pacote

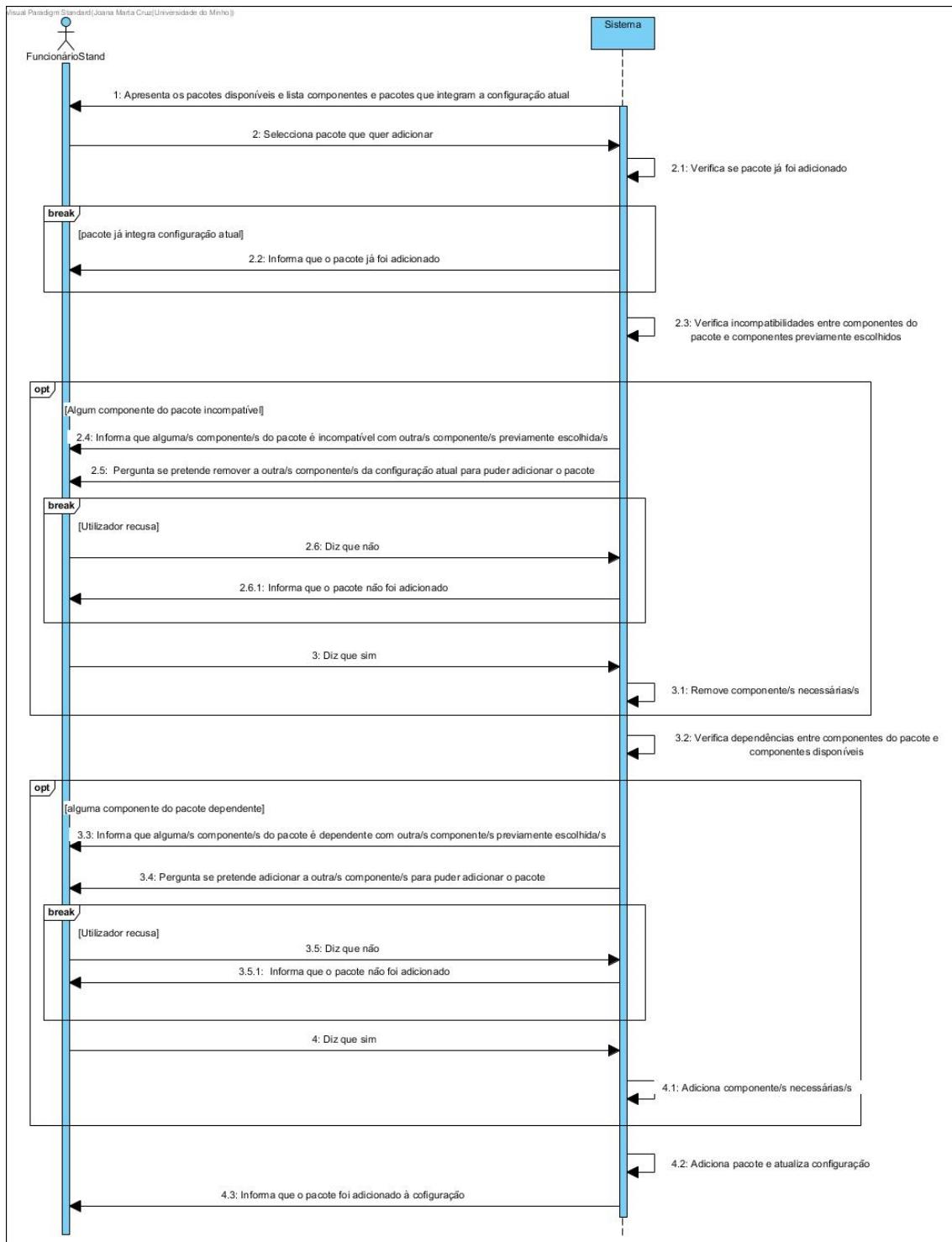


Figura 38 – Diagrama de sequência de sistema

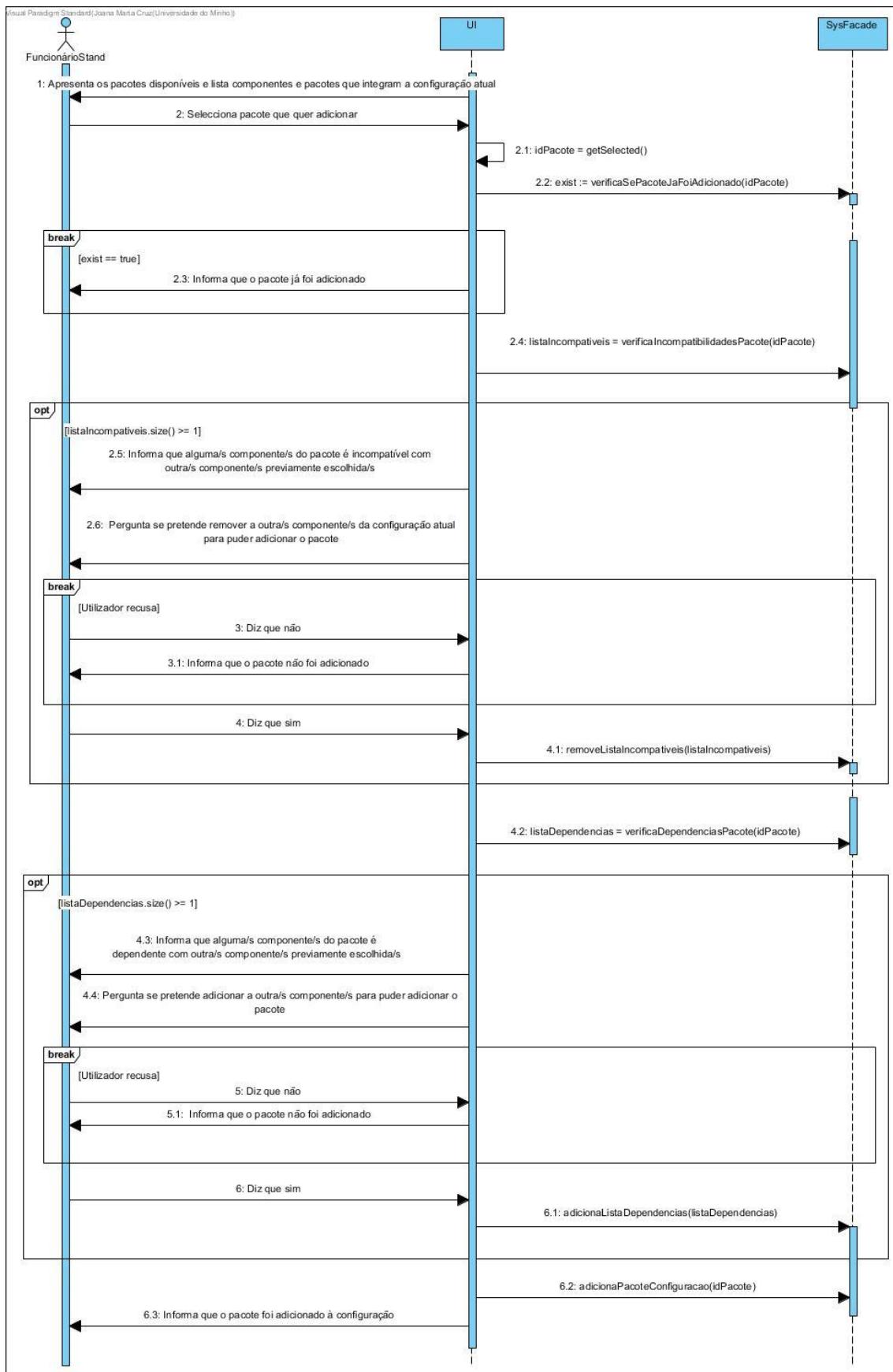


Figura 39 – Diagrama de sequência com UI e Facade

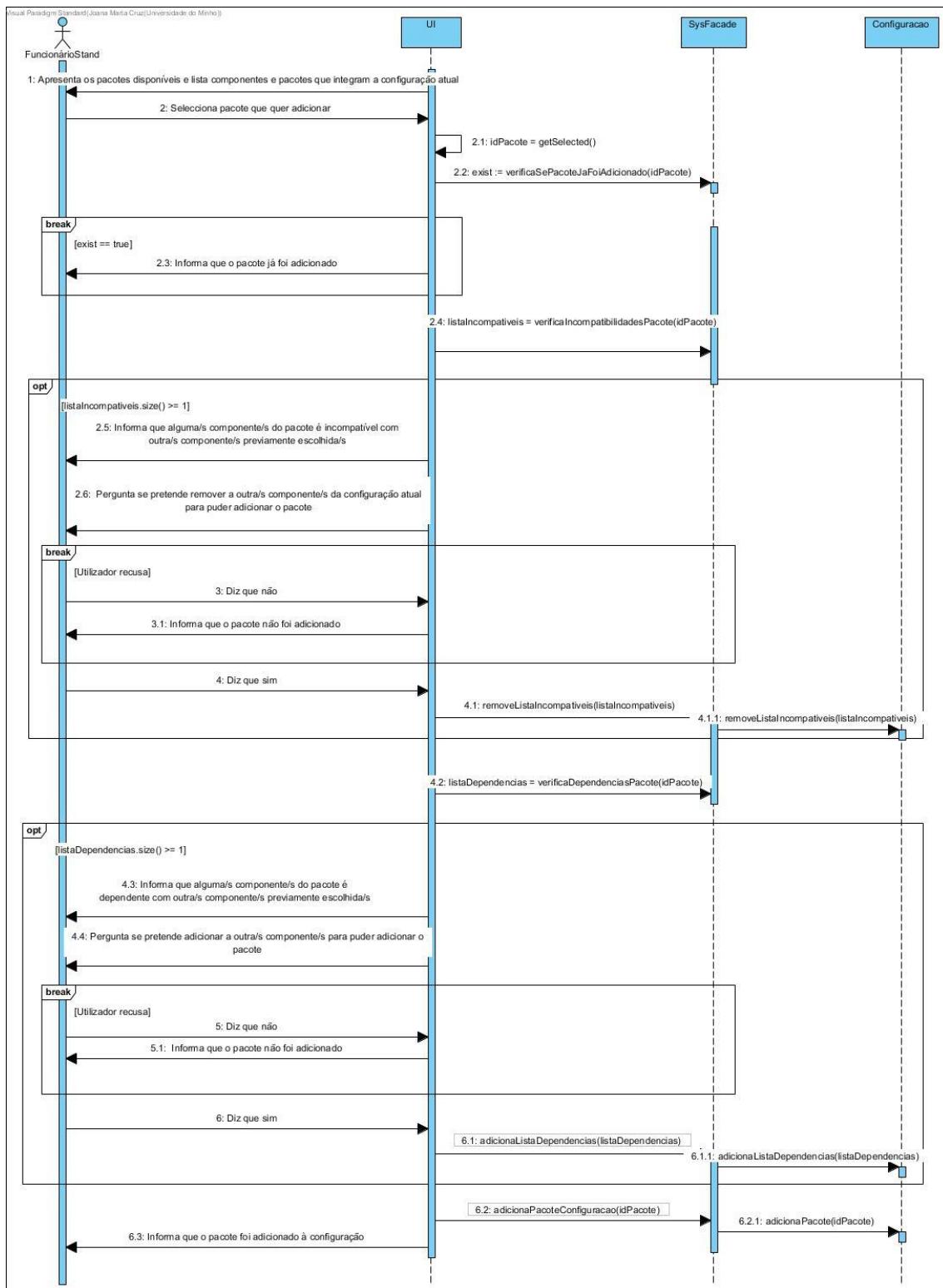


Figura 40 – Diagrama de sequência com subsistemas

2.4.6. Remover Pacote

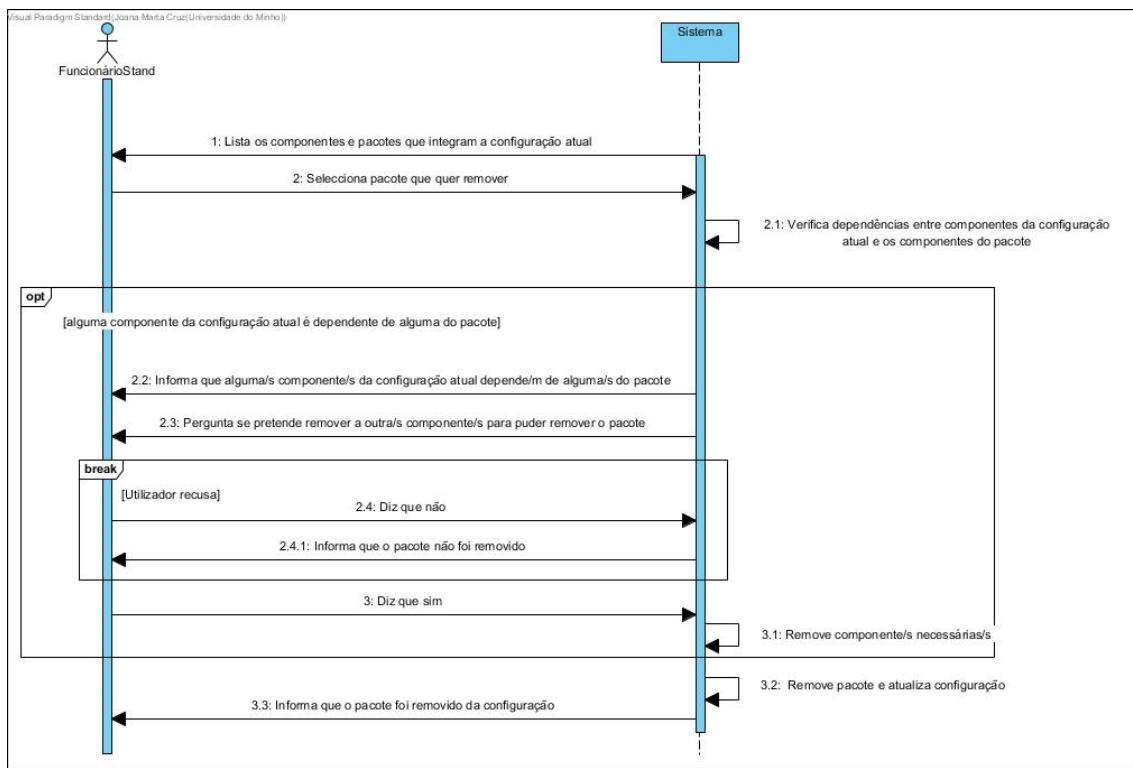


Figura 41 - Diagrama de sequência de sistema

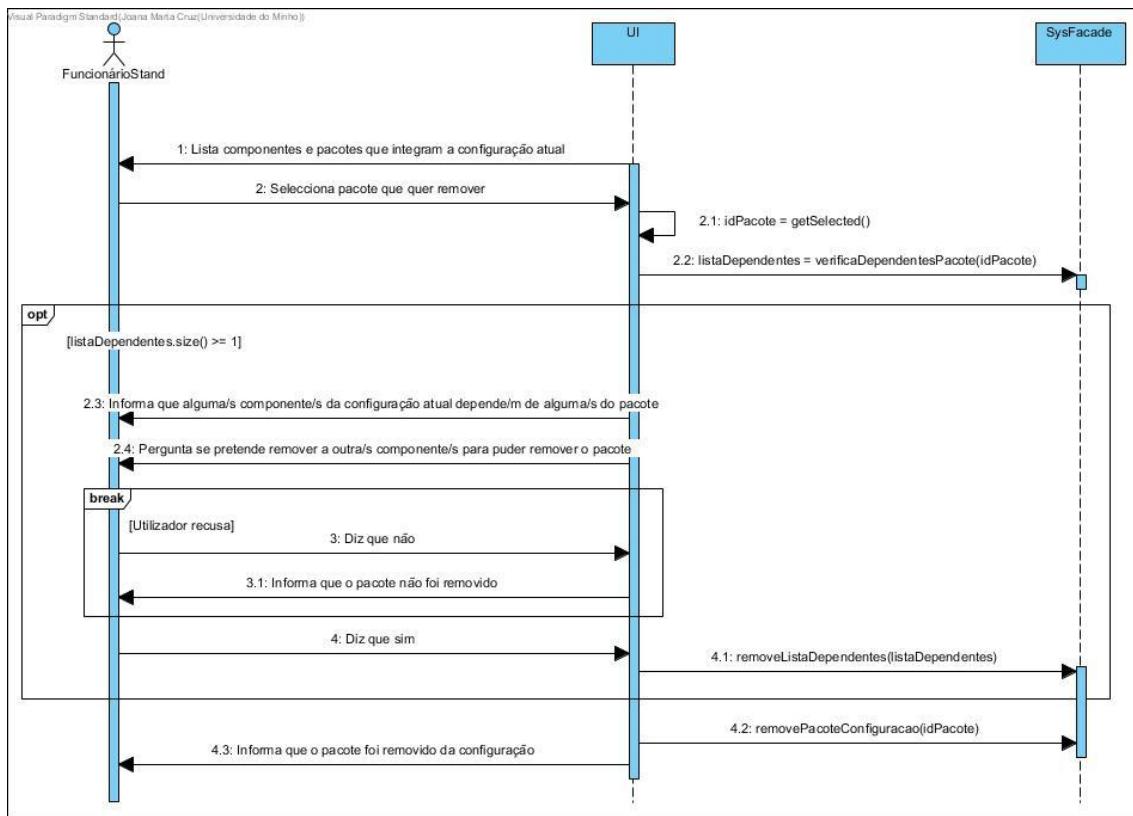


Figura 42 - Diagrama de sequência com UI e Facade

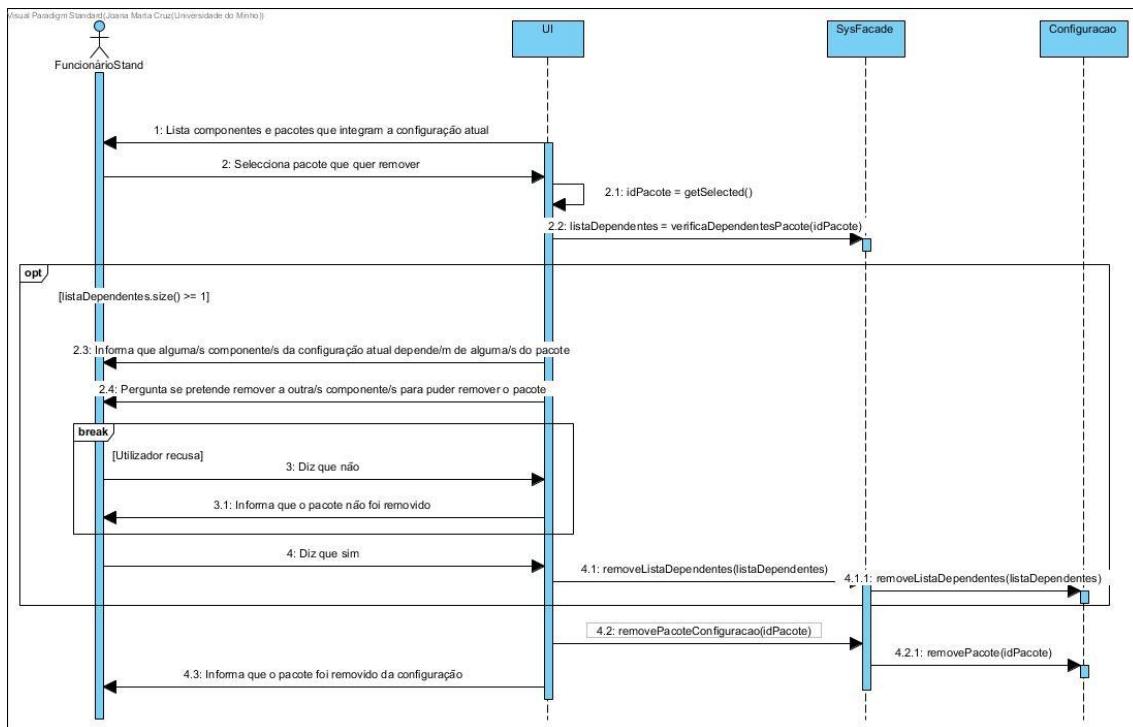


Figura 43 - Diagrama de sequência com subsistemas

2.4.7. Consultar funcionário

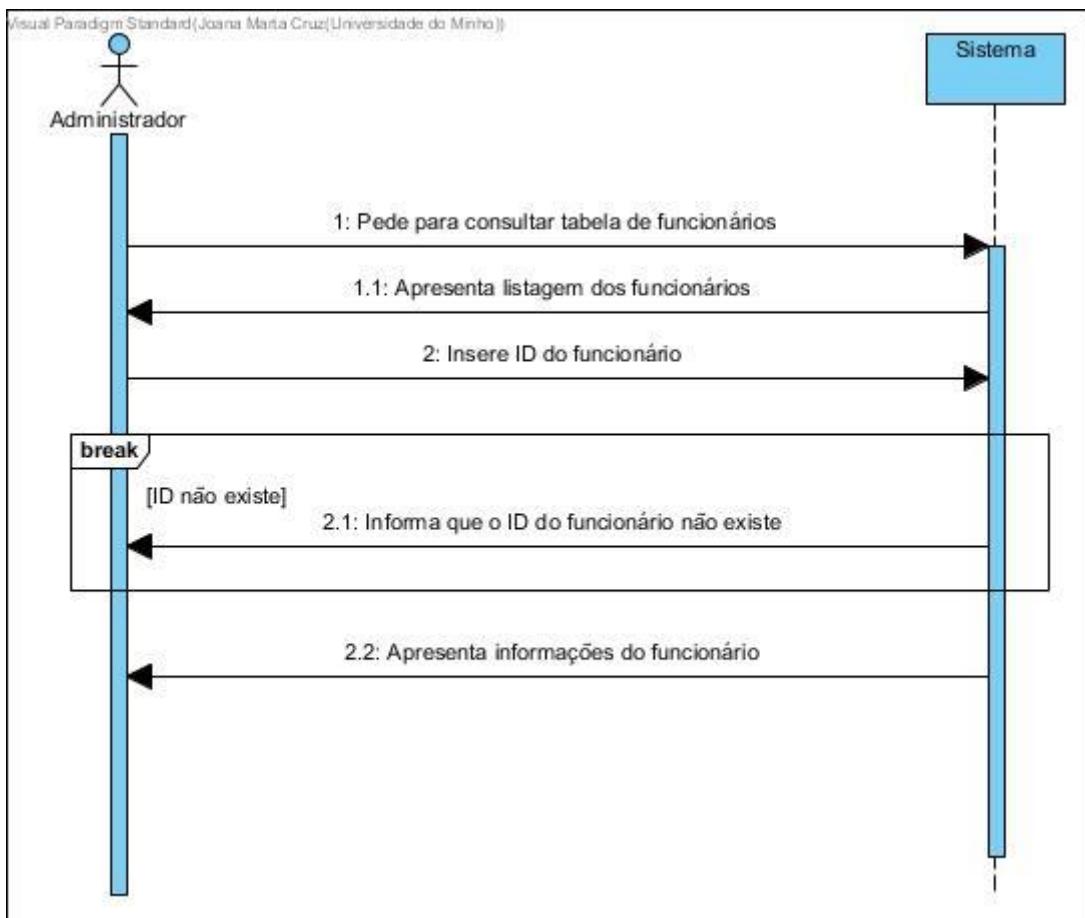


Figura 44 - Diagrama de sequência de sistema

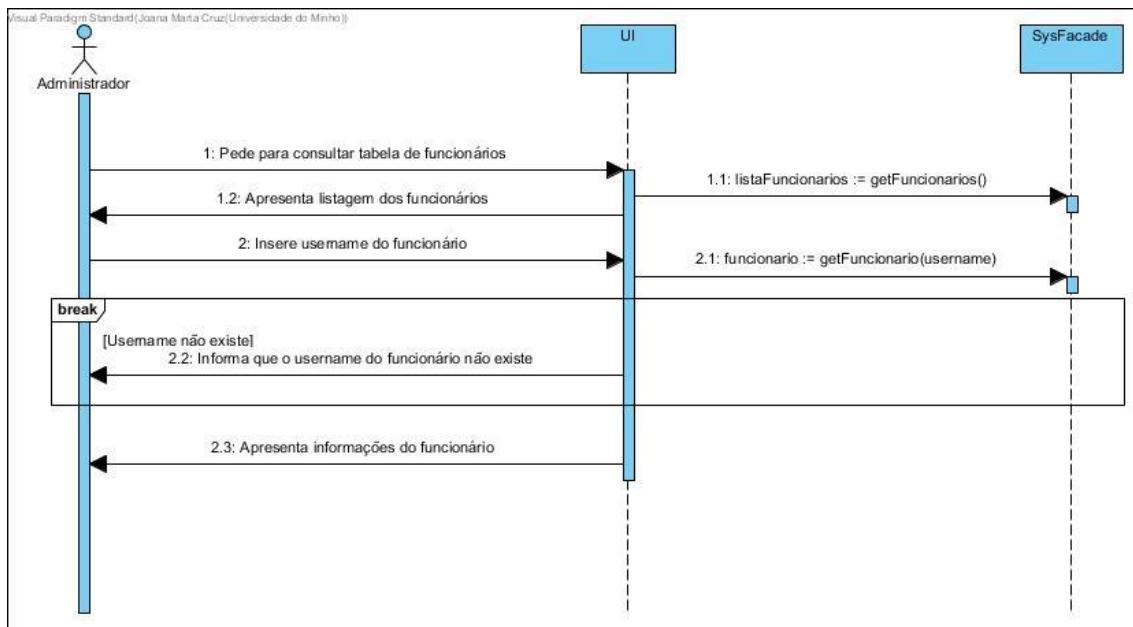


Figura 45 - Diagrama de sequência com UI e Facade

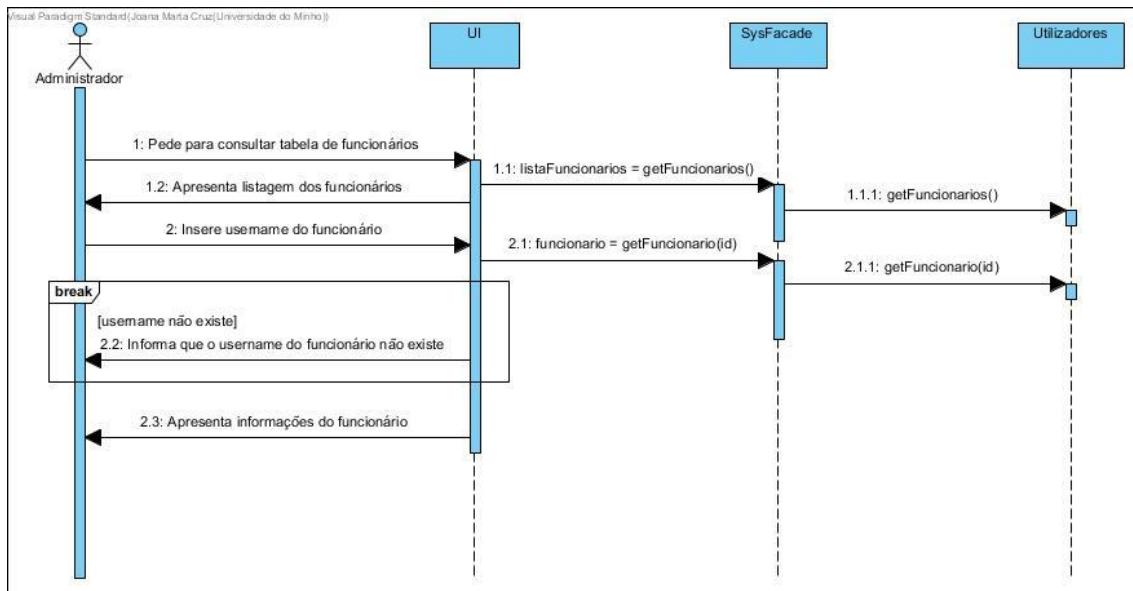


Figura 46 - Diagrama de sequência com subsistemas

2.4.8. Atualizar funcionário

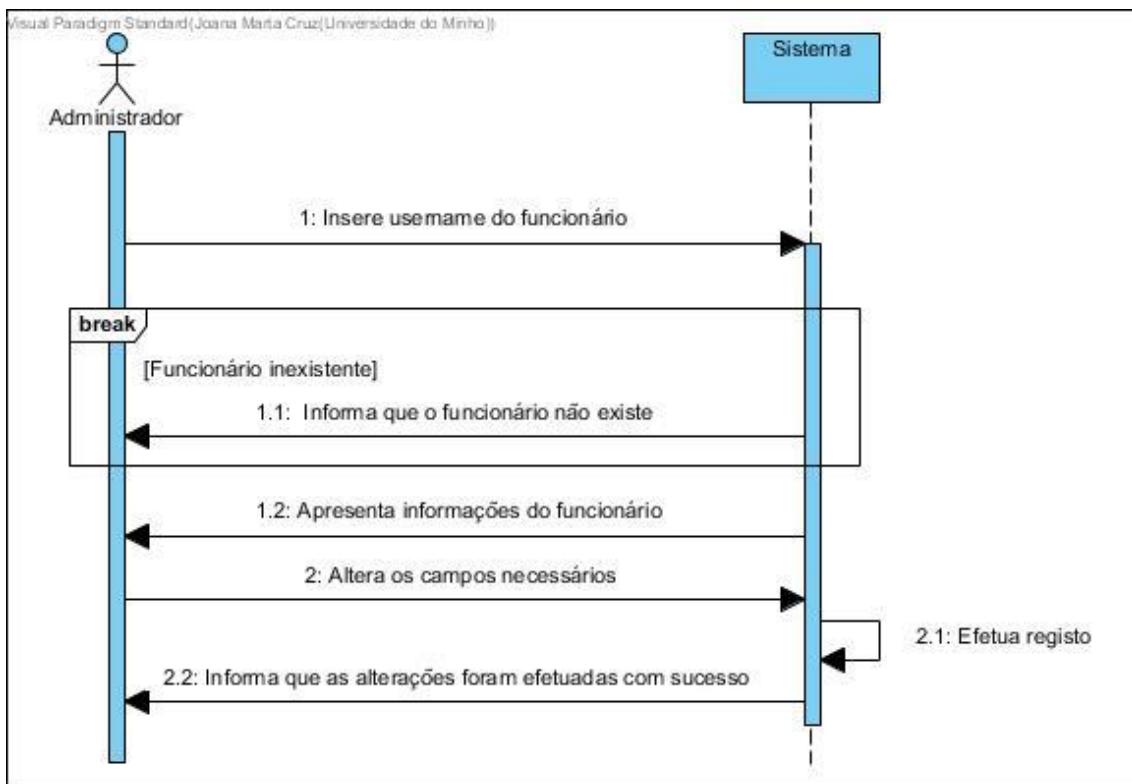


Figura 47 - Diagrama de sequência de sistema

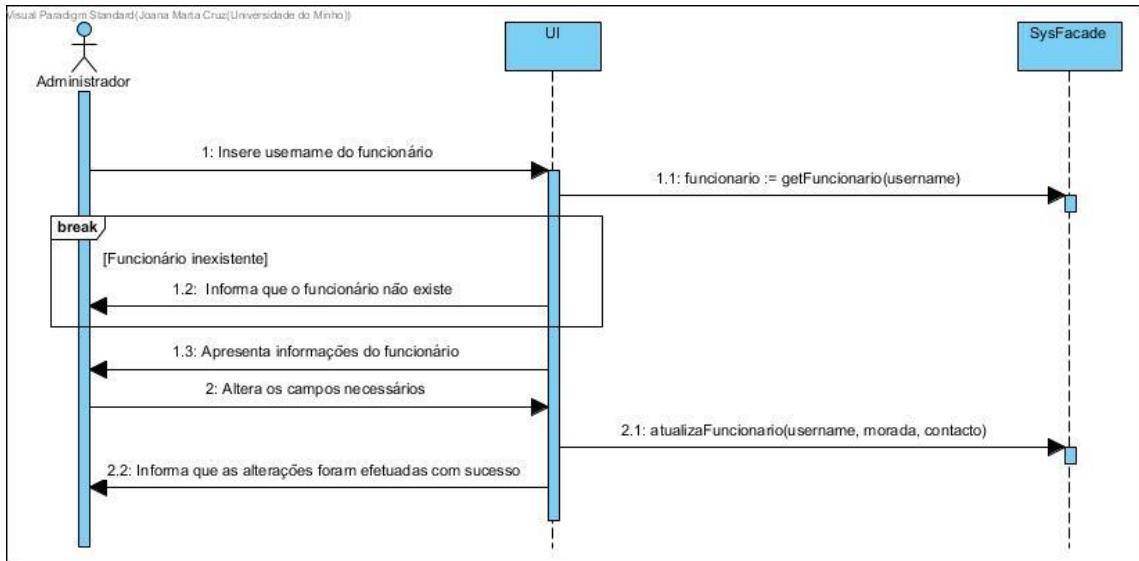


Figura 48 - Diagrama de sequência com UI e Facade

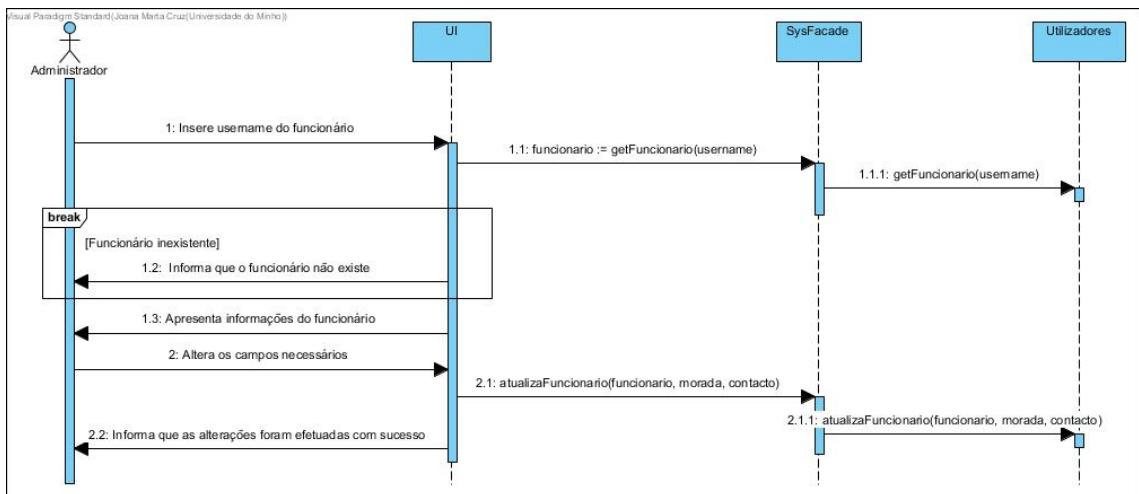


Figura 49 - Diagrama de sequência com subsistemas

2.4.9. Registrar funcionário

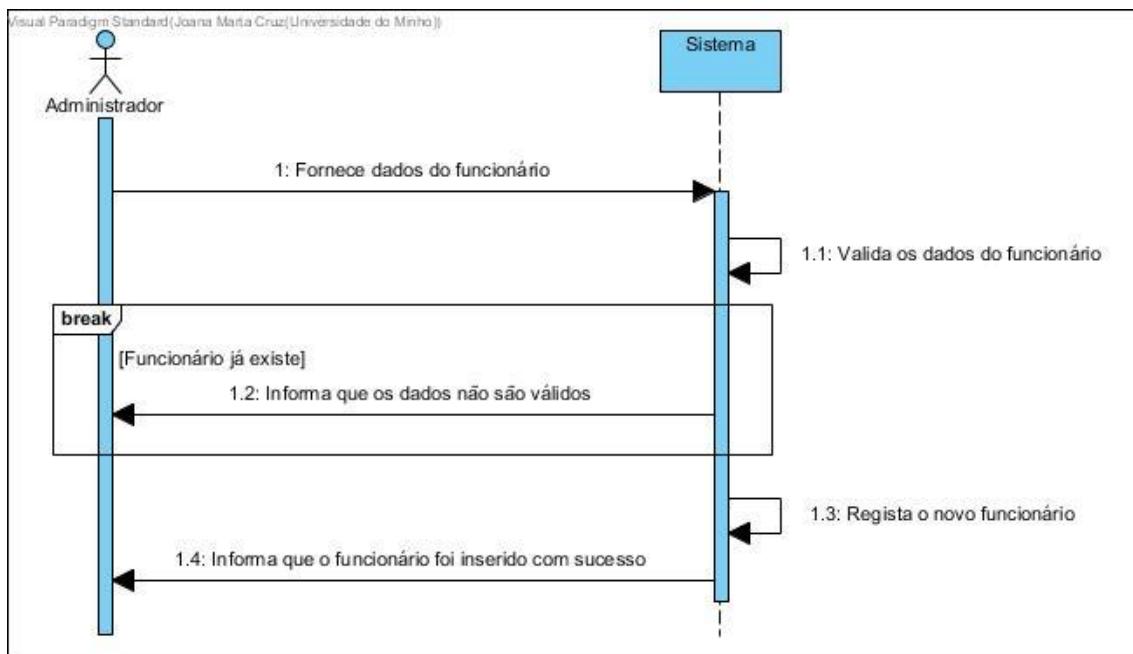


Figura 50 - Diagrama de sequência de sistema

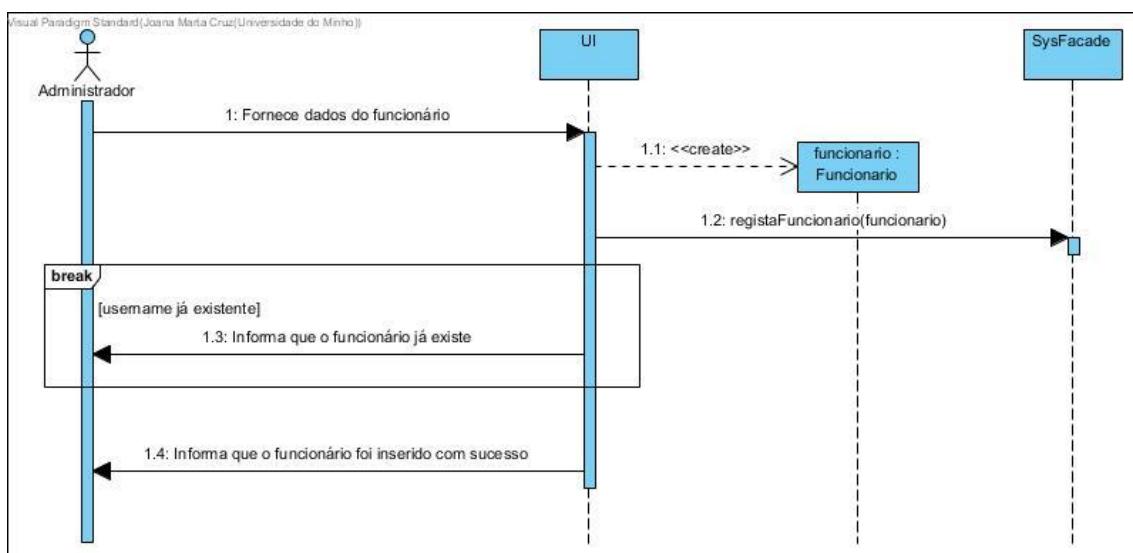


Figura 51 - Diagrama de sequência com UI e Facade

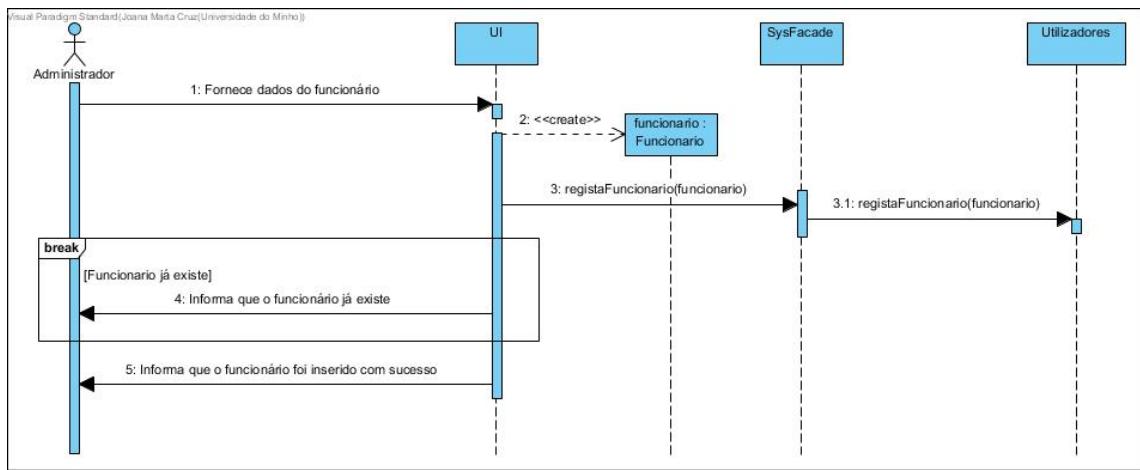


Figura 52 - Diagrama de sequência com subsistemas

2.4.10. Remover funcionário

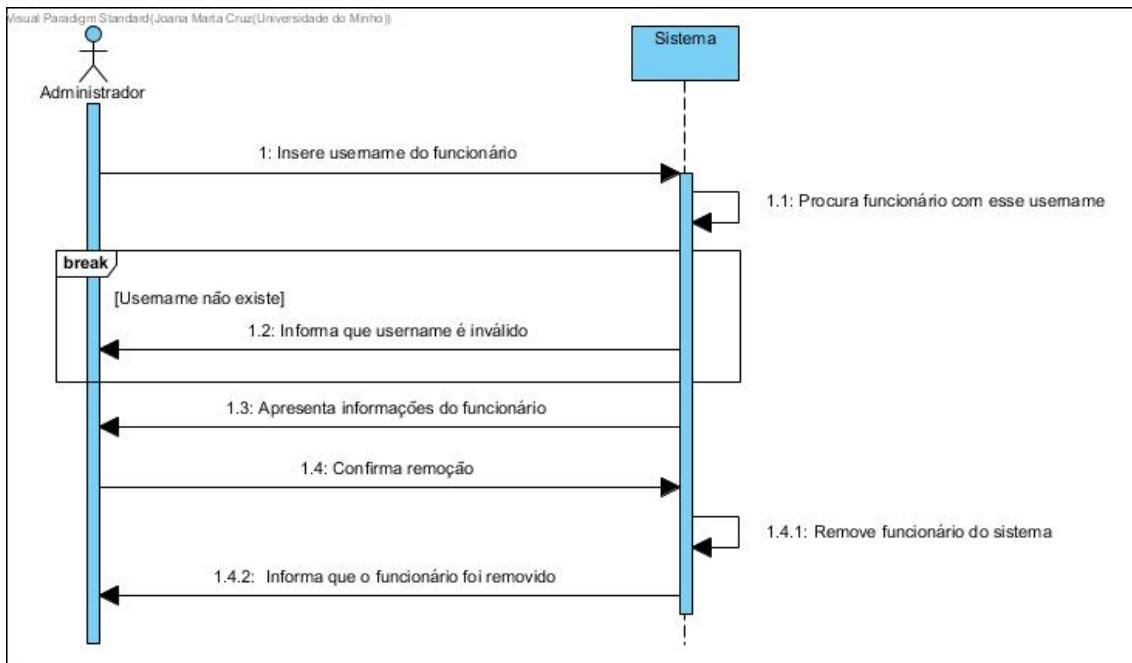


Figura 53 - Diagrama de sequência de sistema

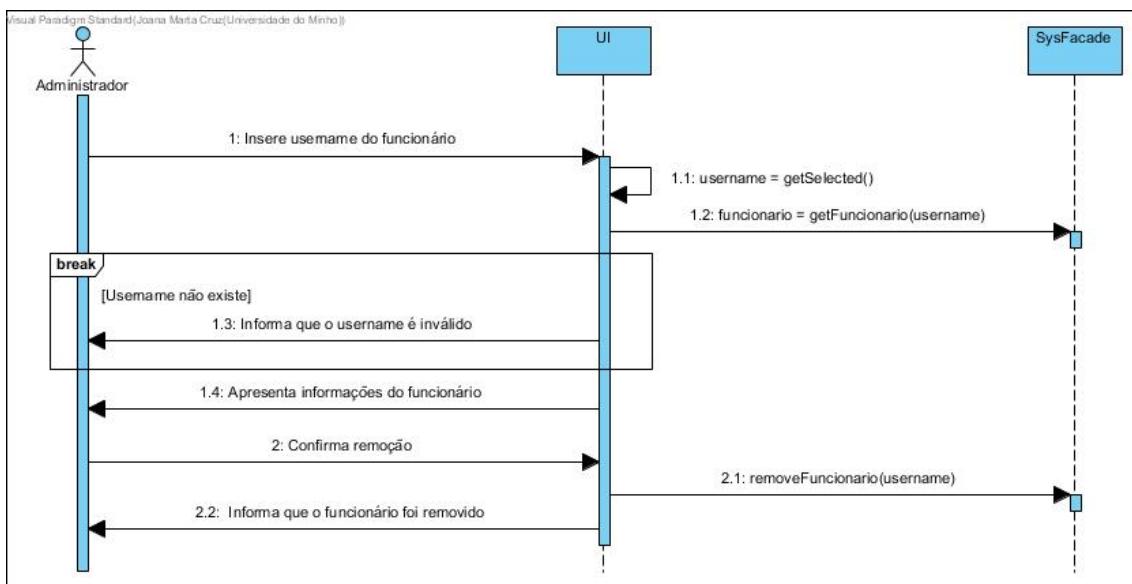


Figura 54 - Diagrama de sequência com UI e Facade

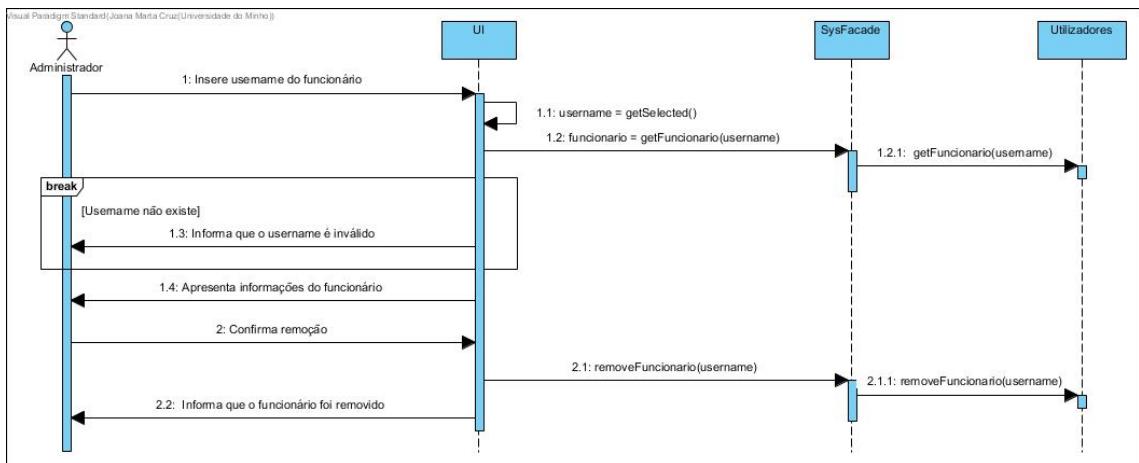


Figura 55 - Diagrama de sequência com subsistemas

2.4.11. Consultar Encomenda

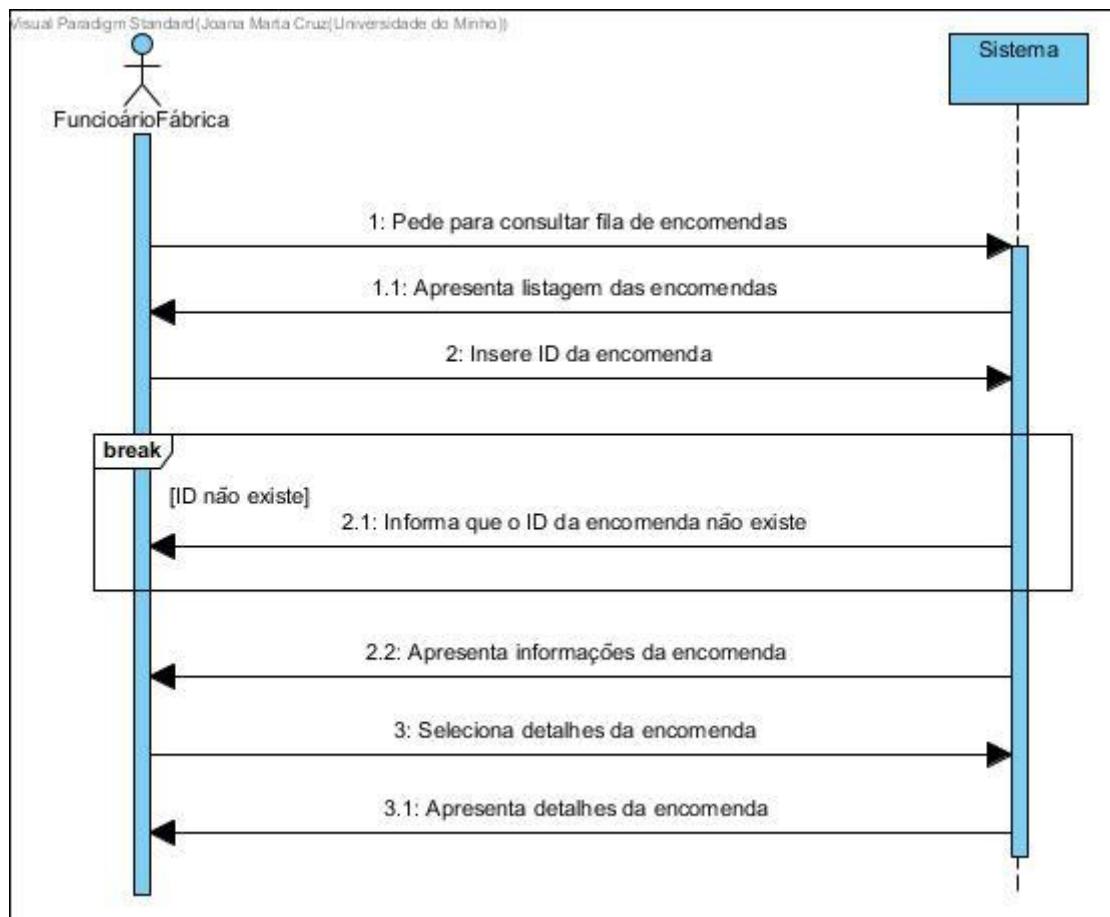


Figura 56 - Diagrama de sequência de sistema

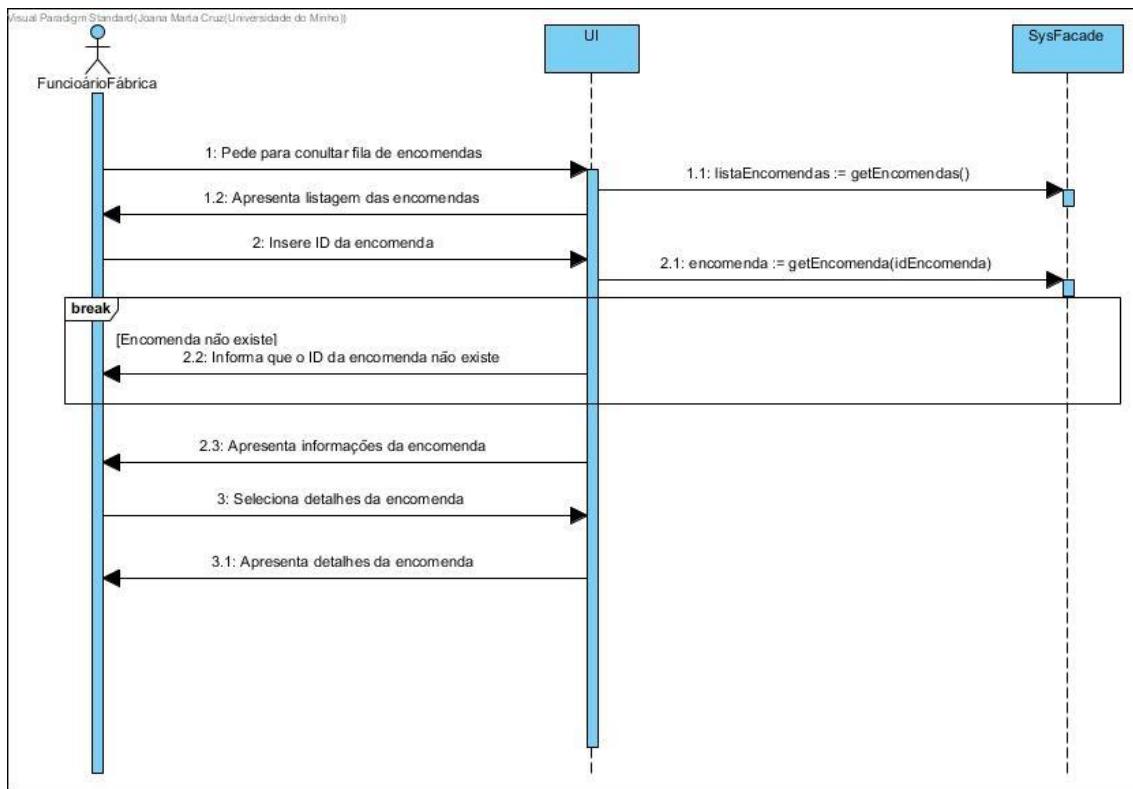


Figura 57 - Diagrama de sequência com UI e Facade

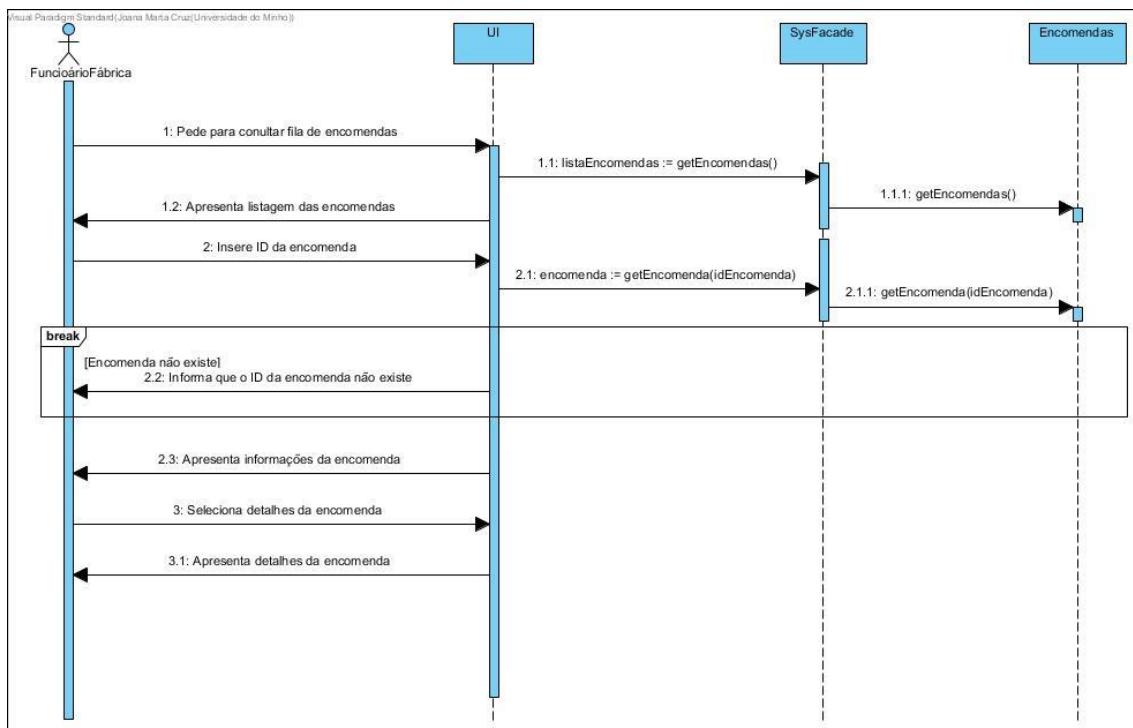


Figura 58- Diagrama de sequência com subsistemas

2.4.12. Registar encomenda

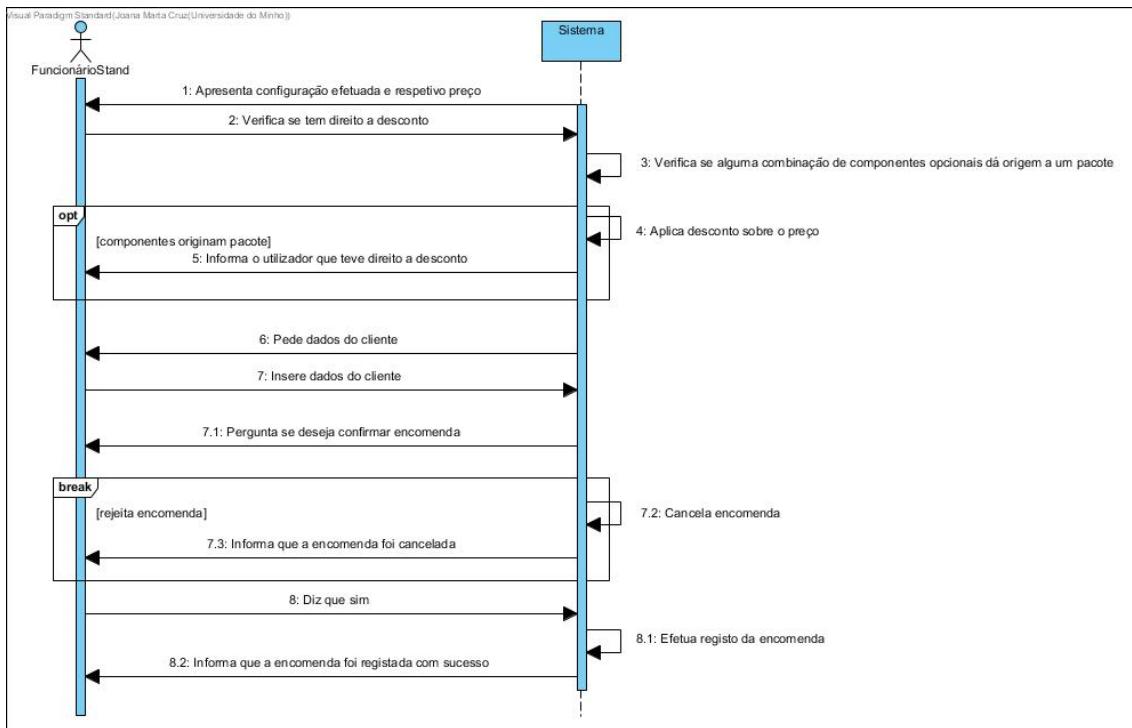


Figura 59 - Diagrama de sequência de sistema

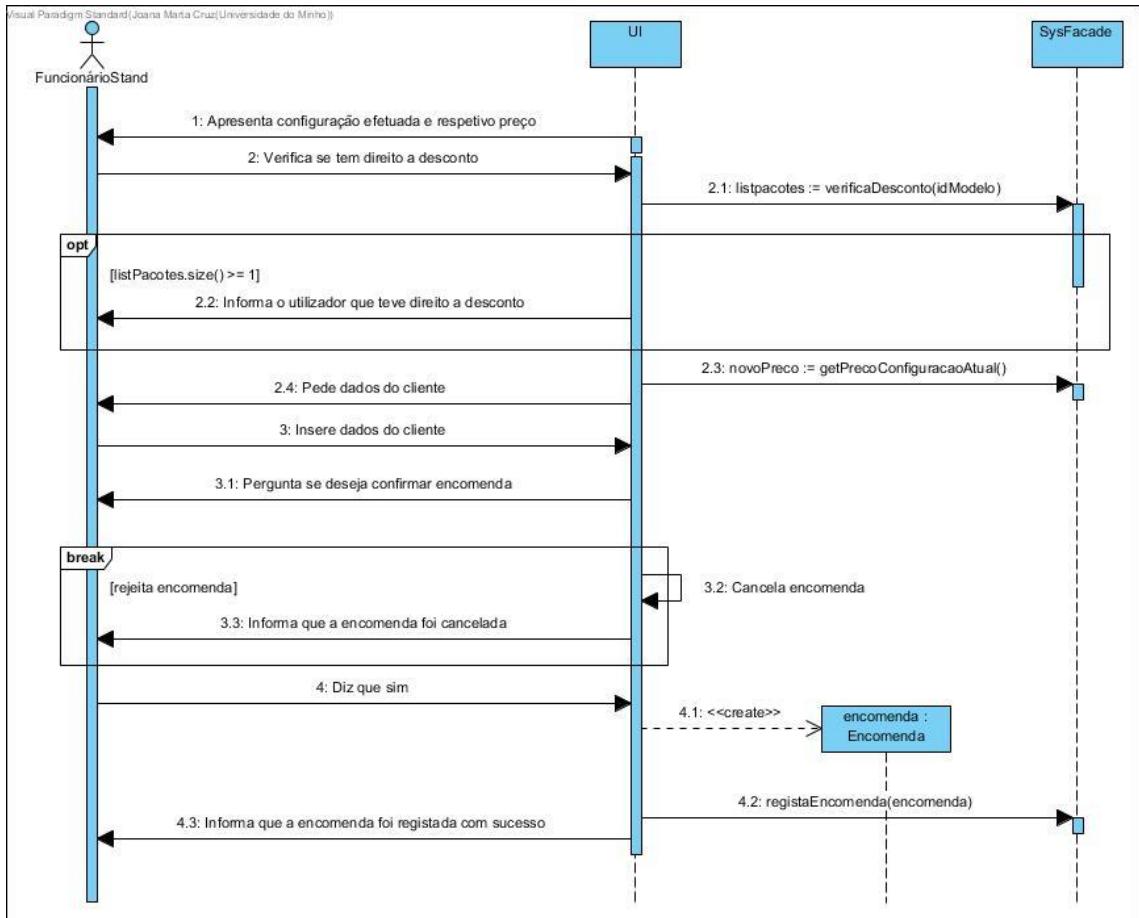


Figura 60 - Diagrama de sequência com UI e Facade

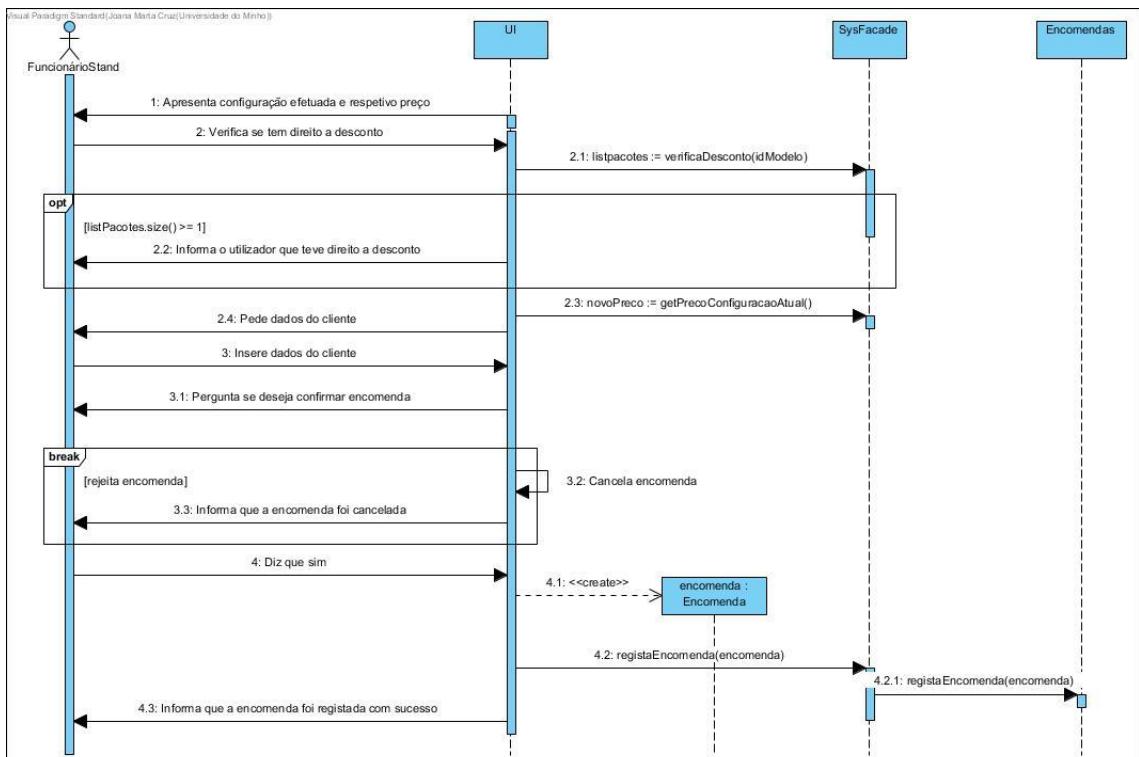


Figura 61 - Diagrama de sequência com subsistemas

2.4.13. Gerir stock de componentes

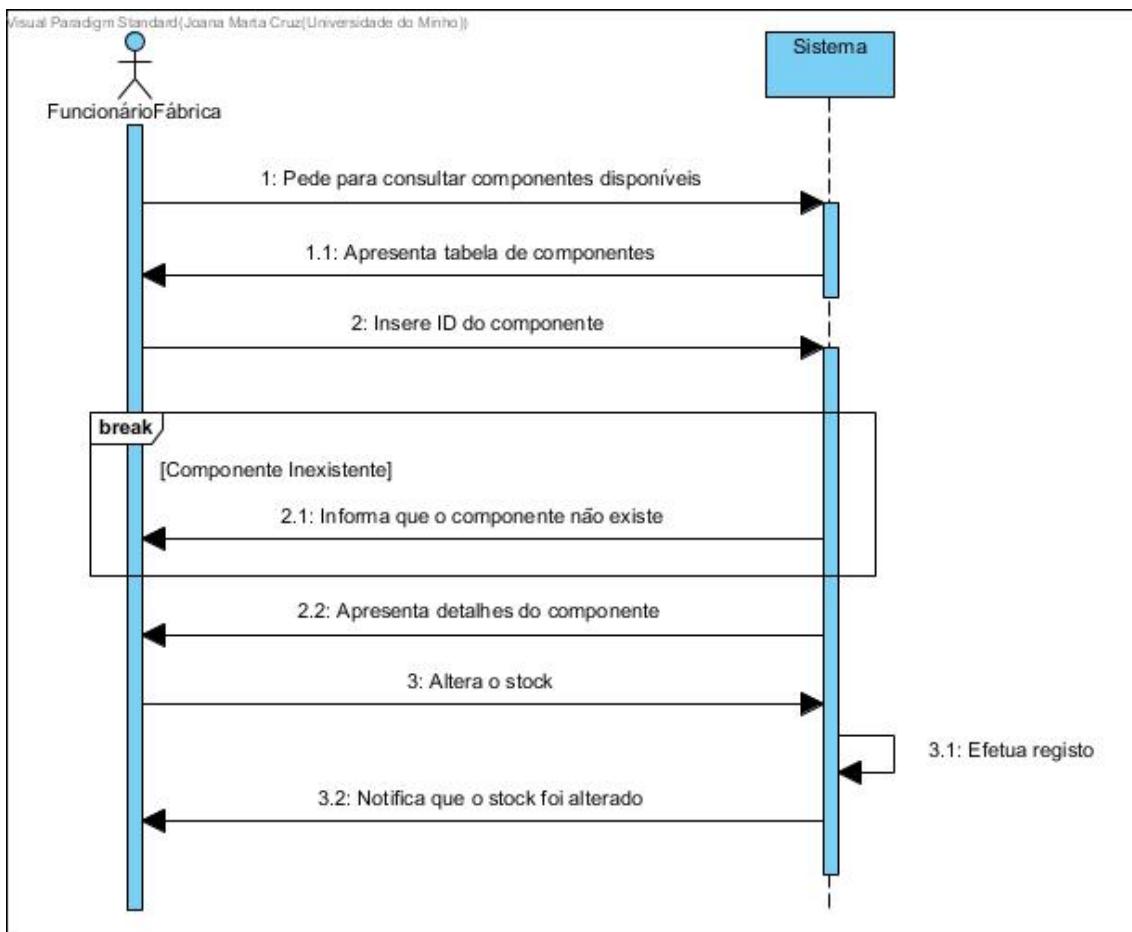


Figura 62 - Diagrama de sequência de sistema

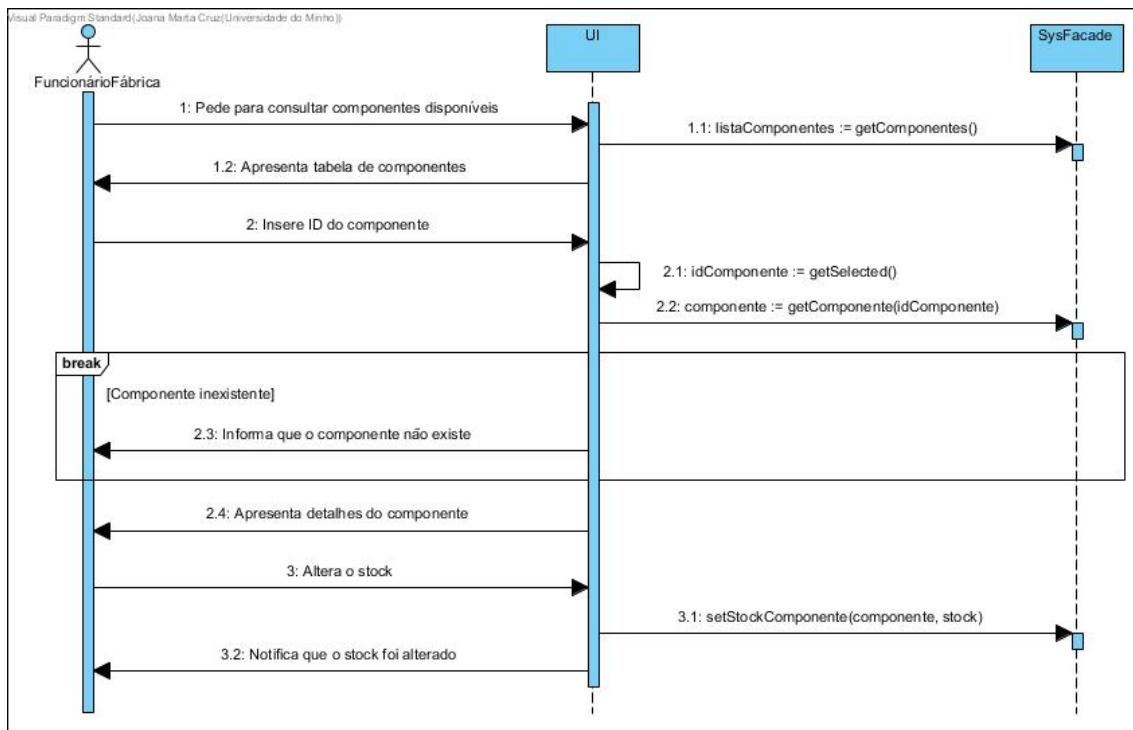


Figura 63 - Diagrama de sequência com UI e Facade

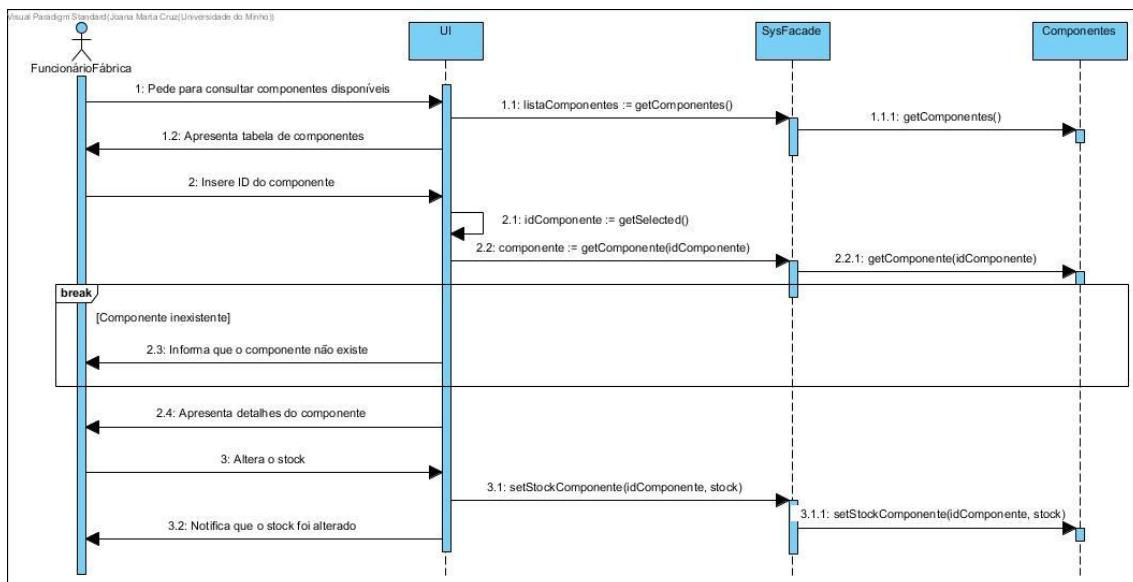


Figura 64 - Diagrama de sequência com subsistemas

2.4.14. Efetuar Configuração Ótima

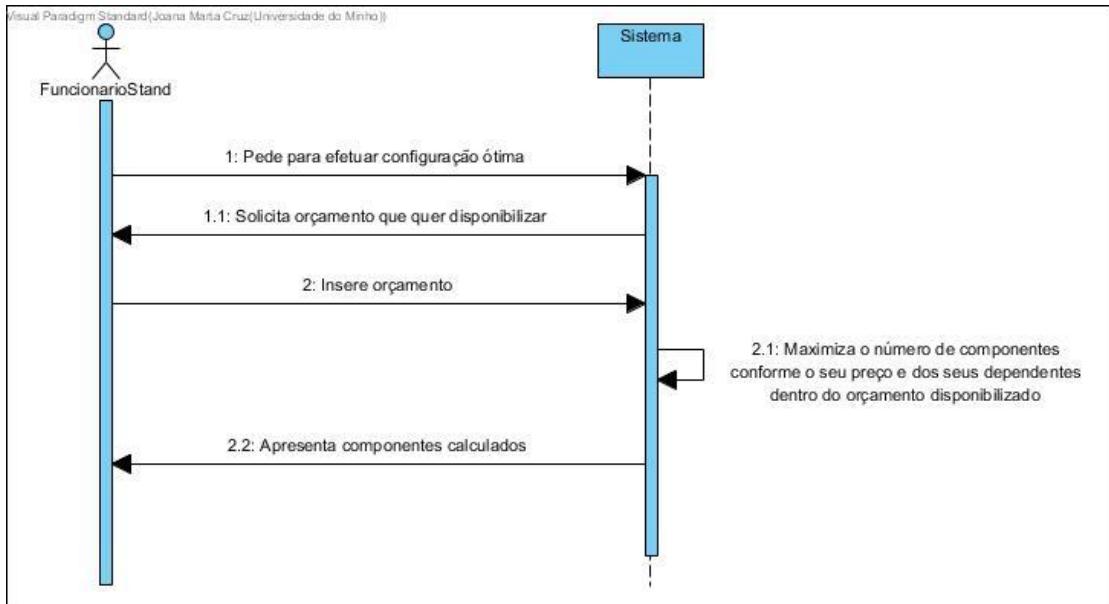


Figura 65 - Diagrama de sequência de sistema

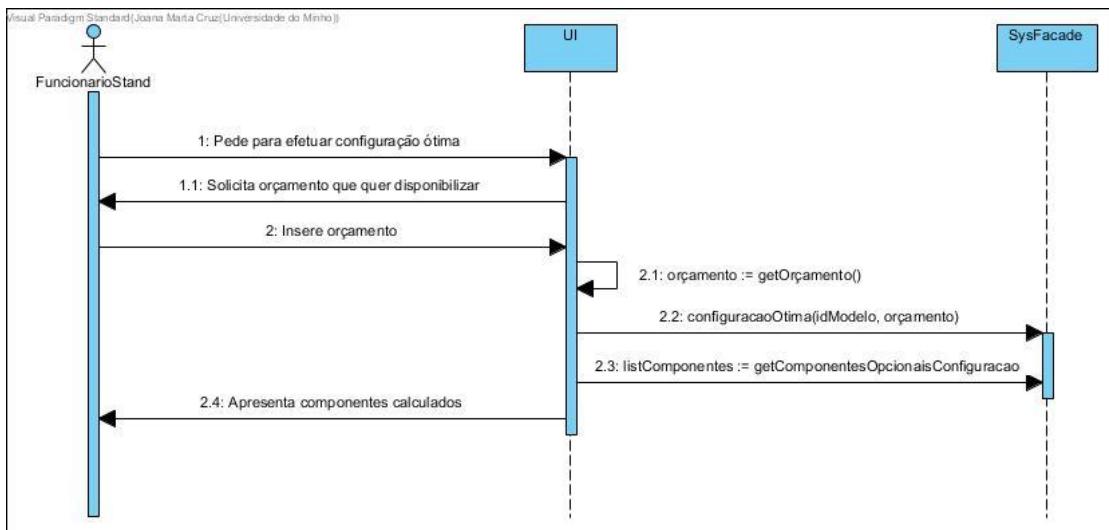


Figura 66 - Diagrama de sequência de sistema com UI e Facade

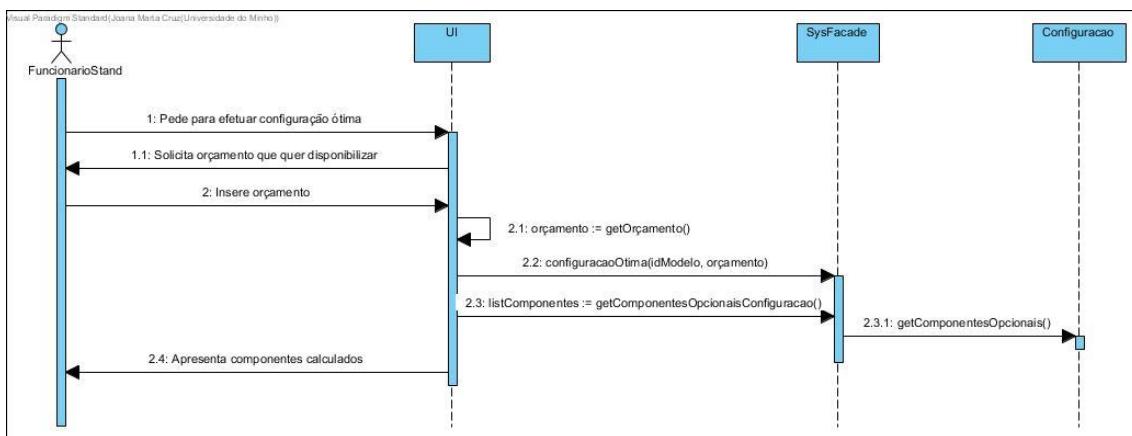


Figura 67 - Diagrama de sequência de sistema com subsistemas

2.5. Diagrama de classes com estrutura de dados e com ORM

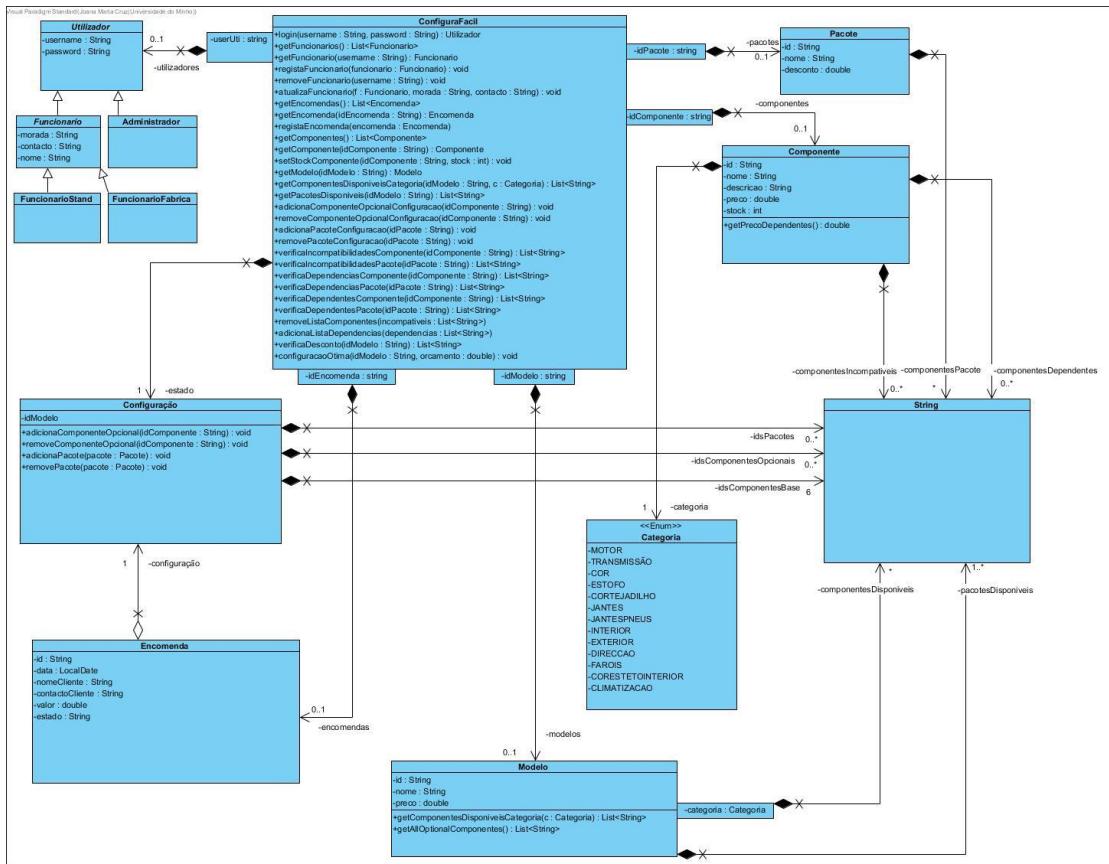


Figura 68 - Diagrama de classes com estrutura de dados

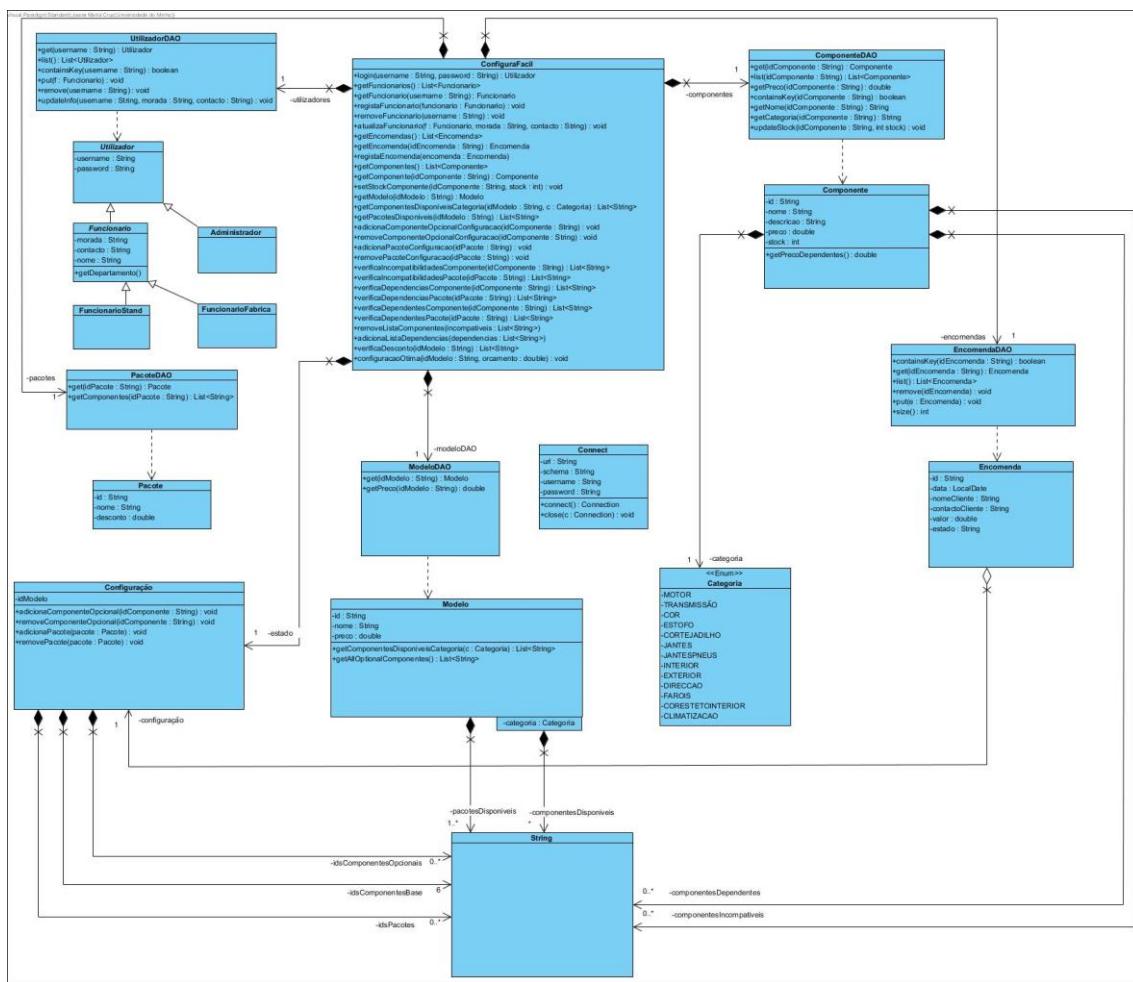


Figura 69 - Diagrama de classes com ORM

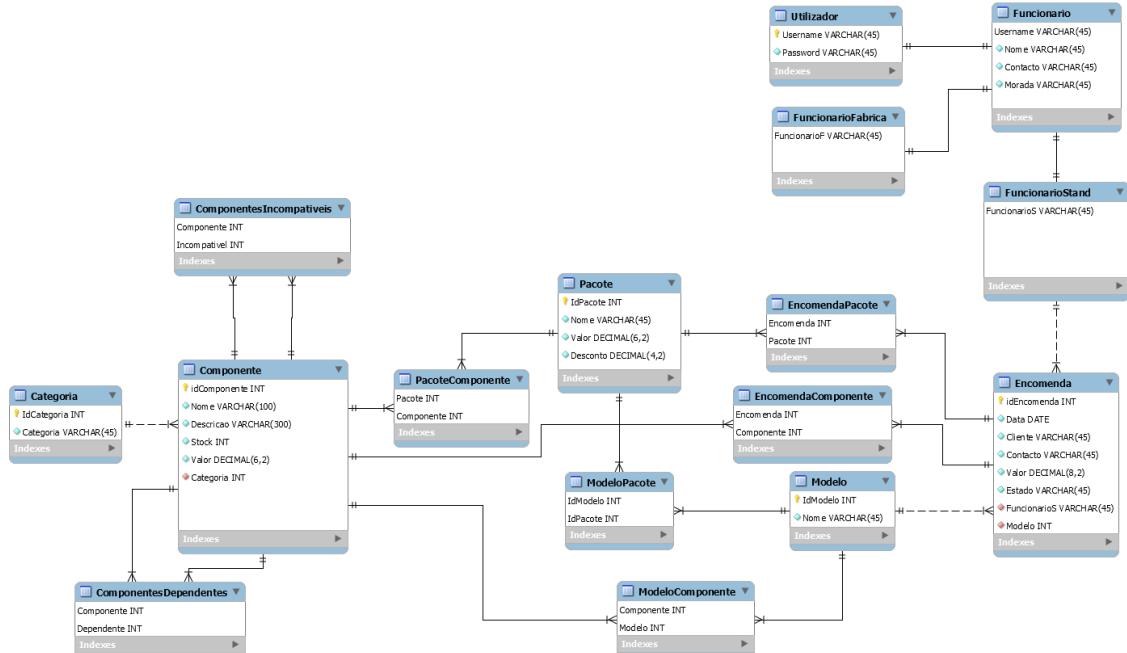


Figura 70 - Modelo de dados

2.5.1. Diagrama de sequência de implementação

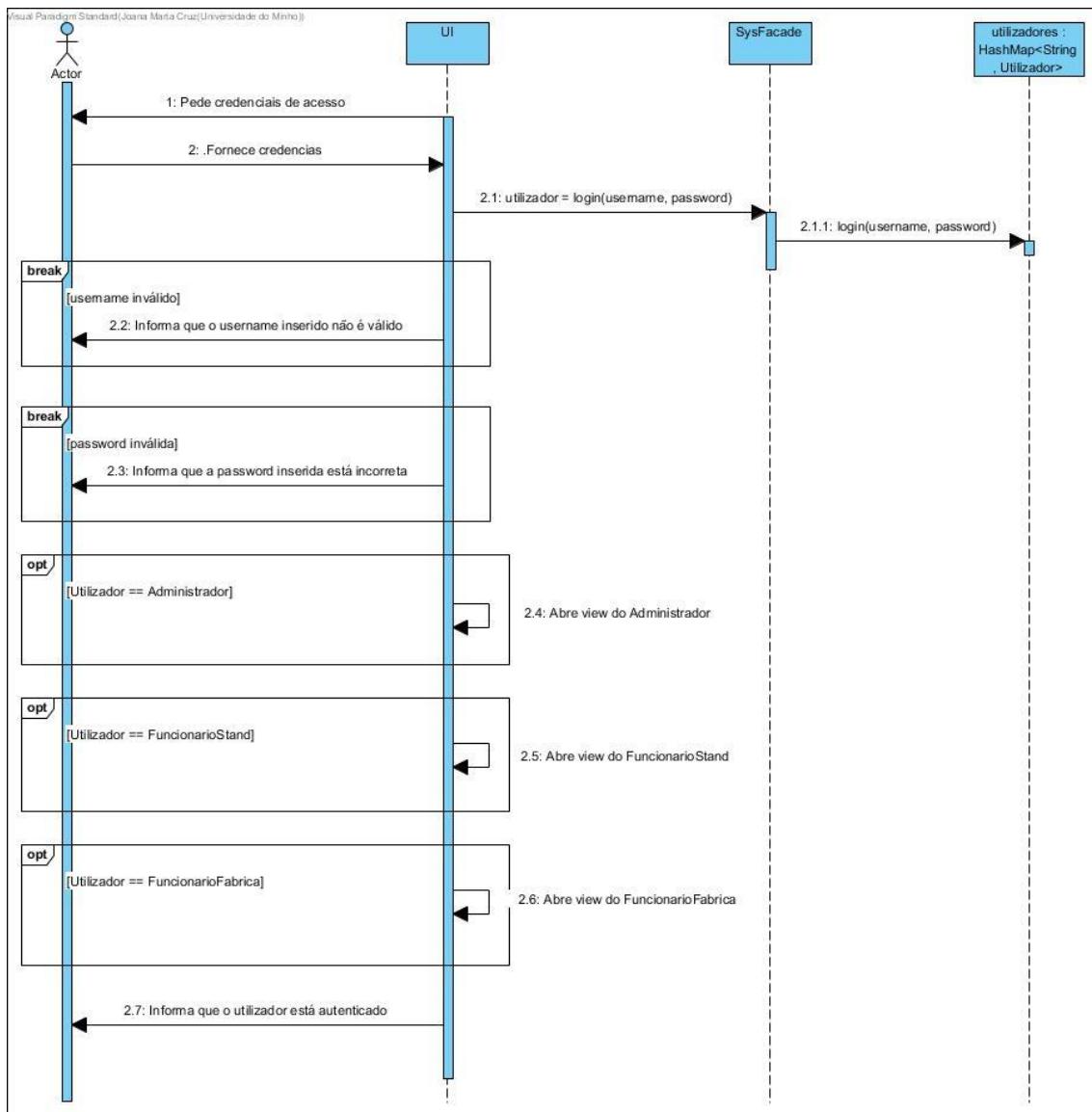


Figura 71 - Diagrama de sequência de implementação para o use case Efetuar Autenticação

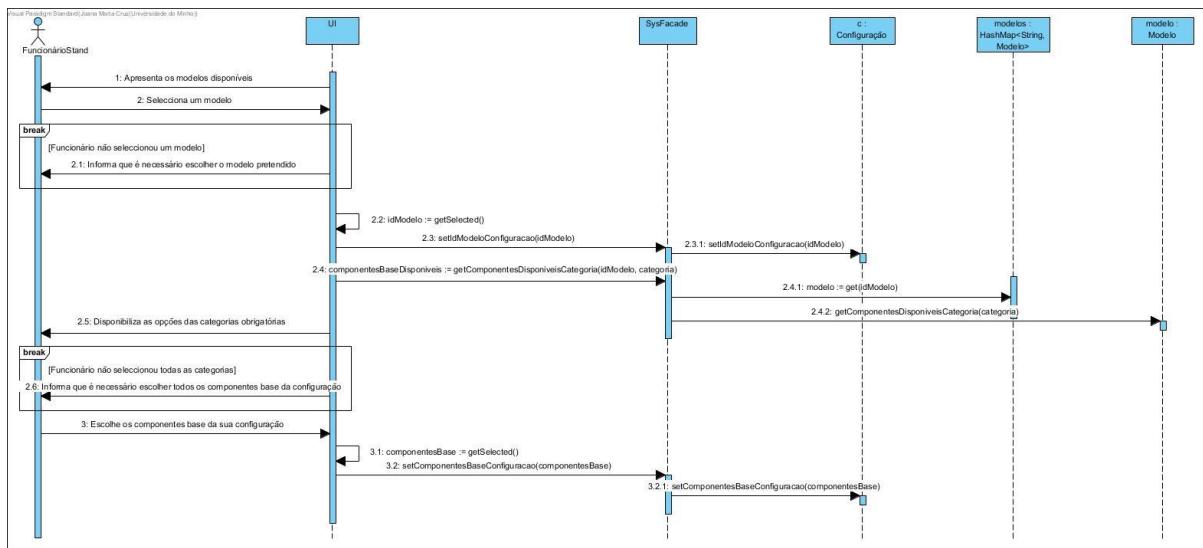


Figura 72 - Diagrama de sequência de implementação para o use case Efetuar Configuração Base

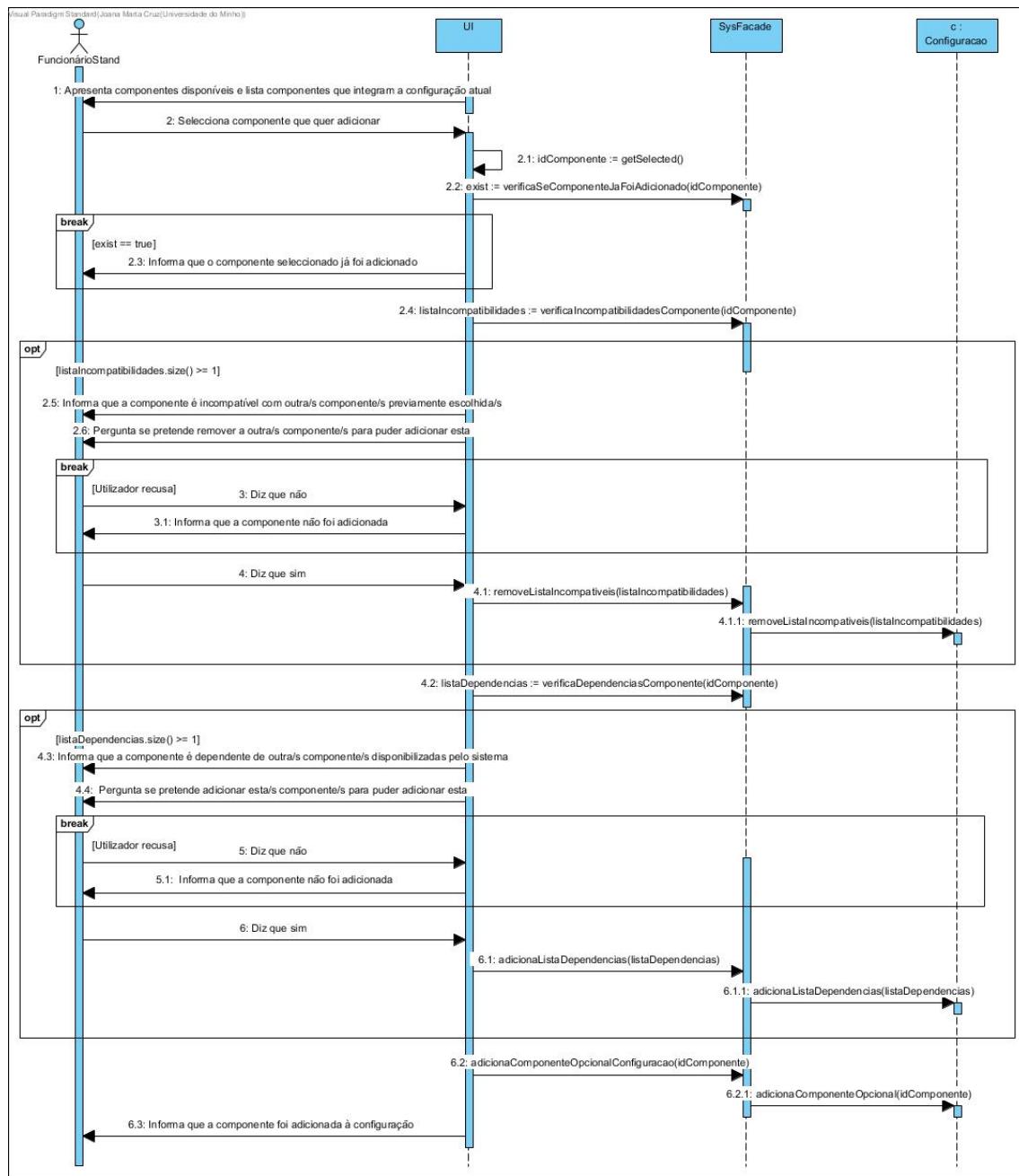


Figura 73 - Diagrama de sequência de implementação para o use case Adicionar Componente

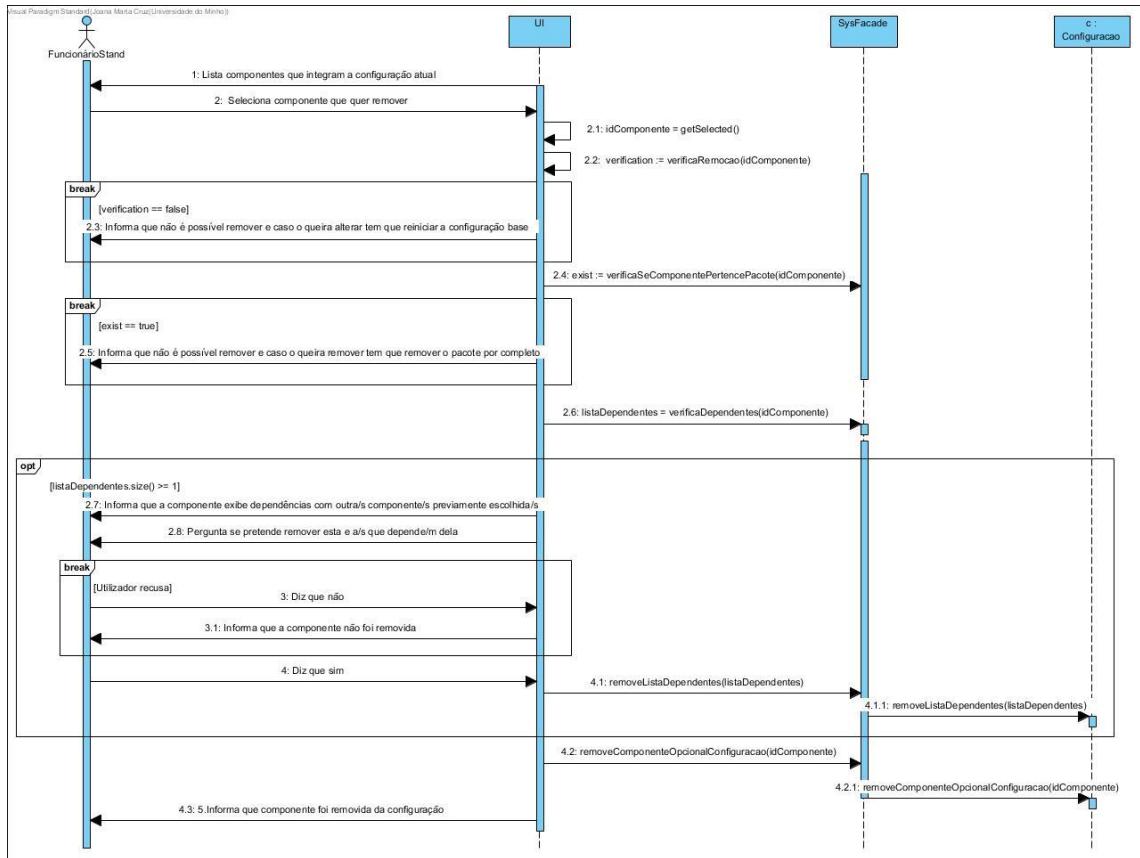


Figura 74 - Diagrama de sequência de implementação para o use case Remover Componente

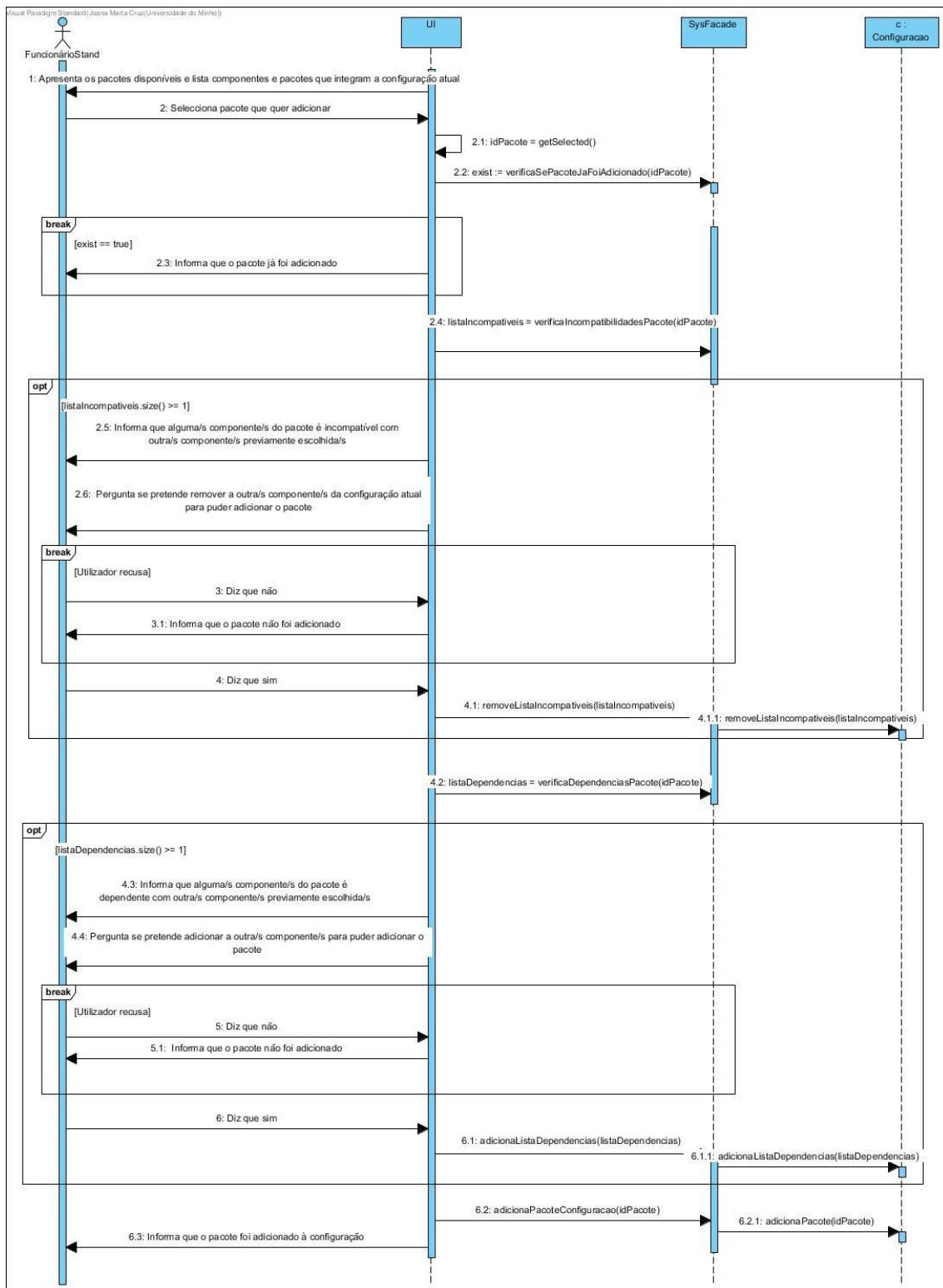


Figura 75 - Diagrama de sequência de implementação para o use case Adicionar Pacote

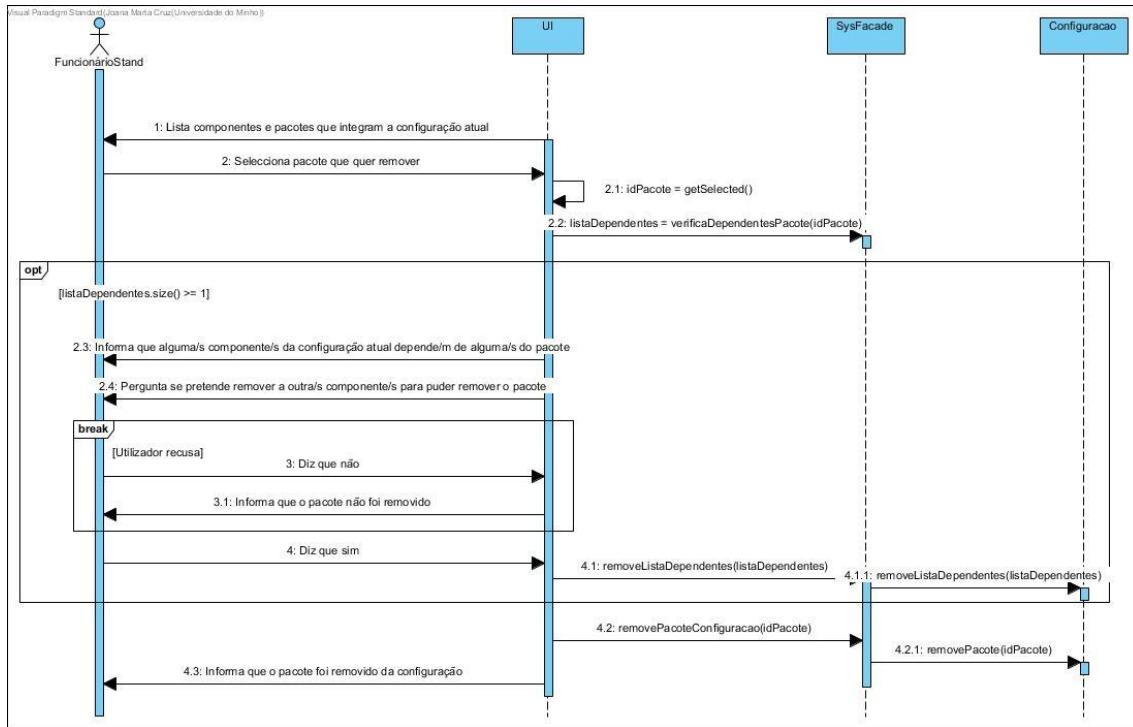


Figura 76 - Diagrama de sequência de implementação para o use case Remover Pacote

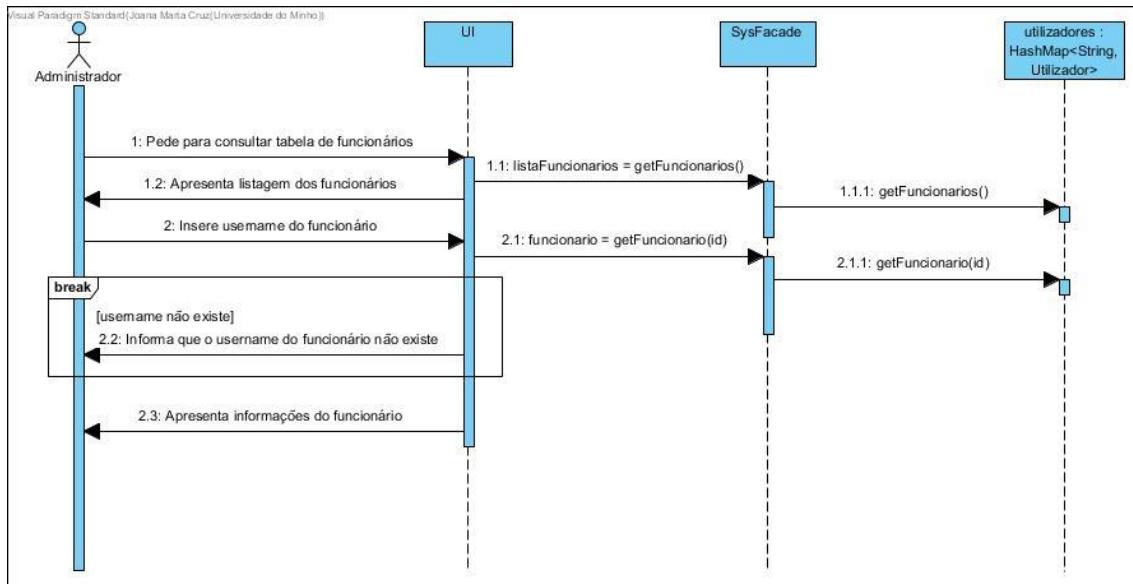


Figura 77 - Diagrama de sequência de implementação para o use case Consultar Funcionário

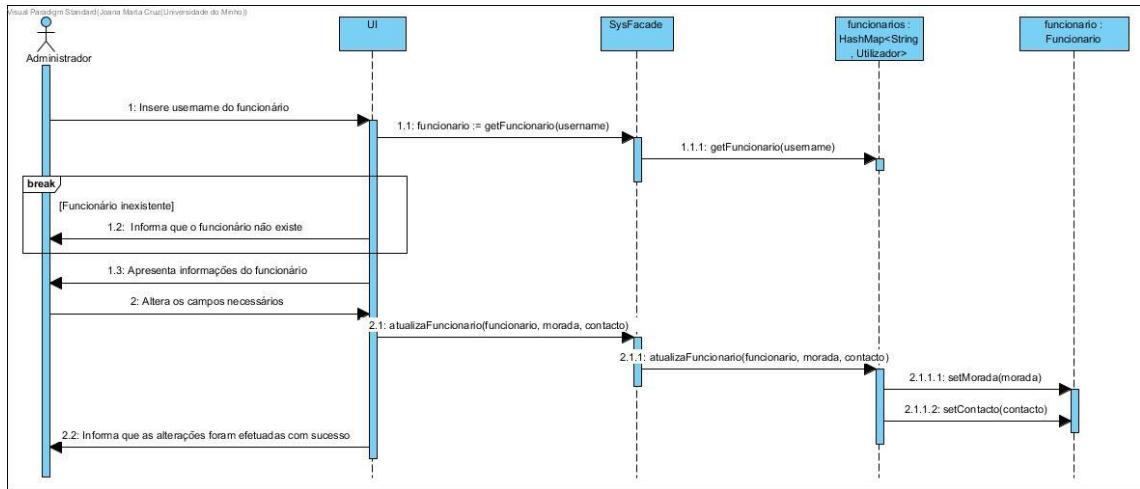


Figura 78 - Diagrama de sequência de implementação para o use case Atualizar Funcionário

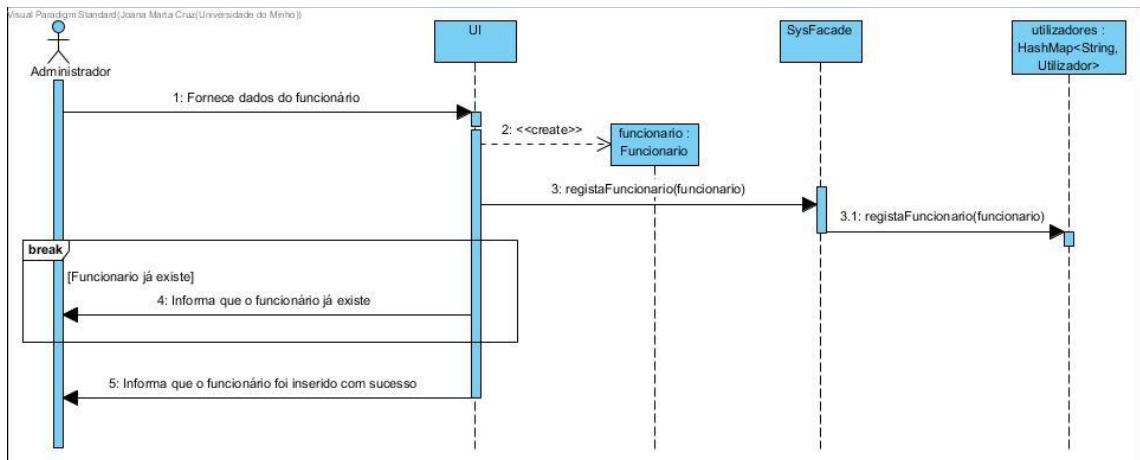


Figura 79 - Diagrama de sequência de implementação para o use case Registar Funcionário

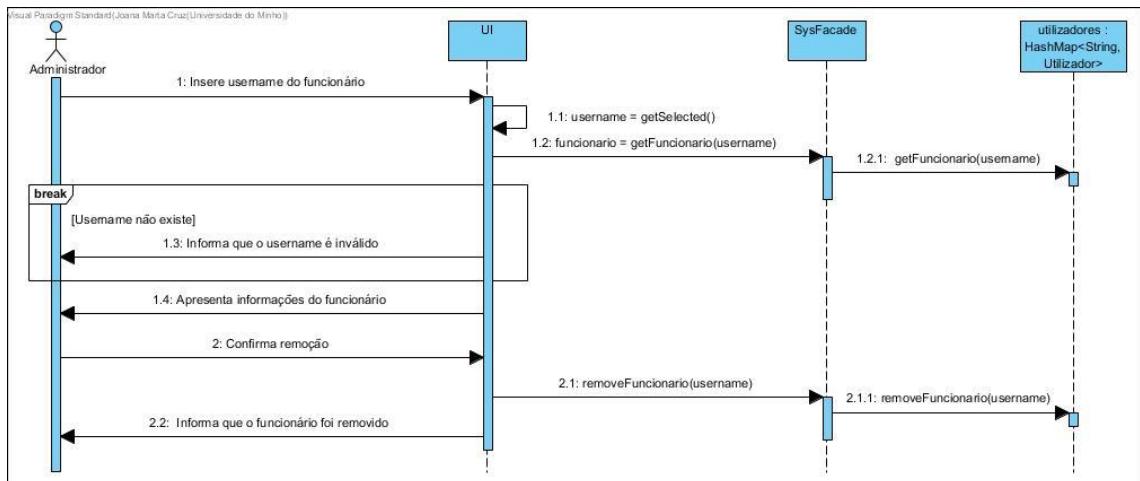


Figura 80 - Diagrama de sequência de implementação para o use case Remover Funcionário

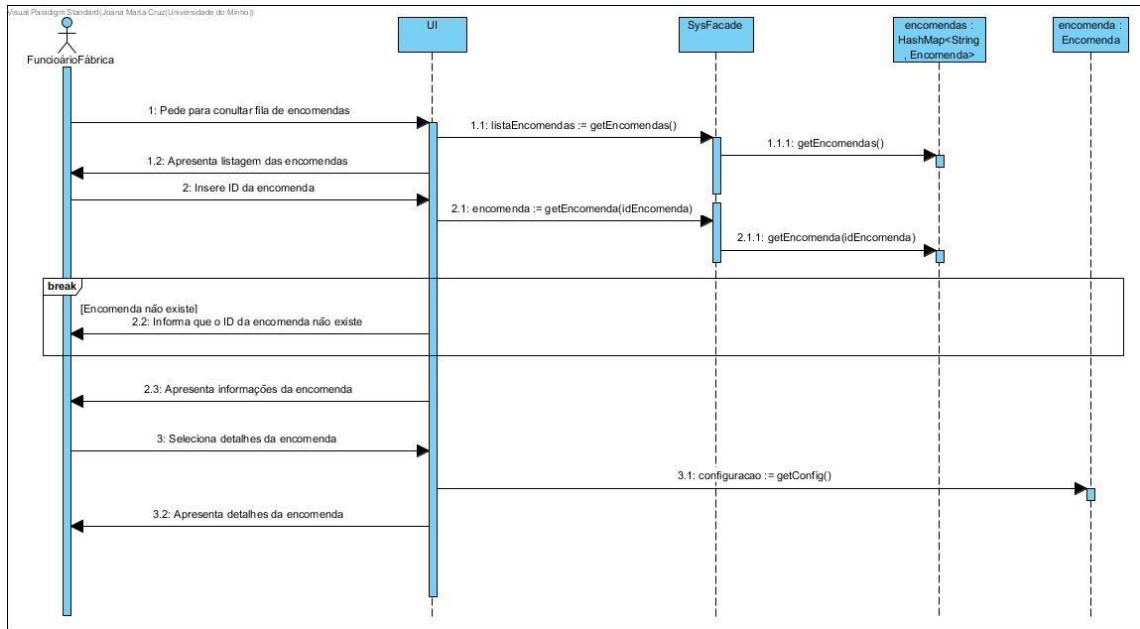


Figura 81 - Diagrama de sequência de implementação para o use case Consultar Encomenda

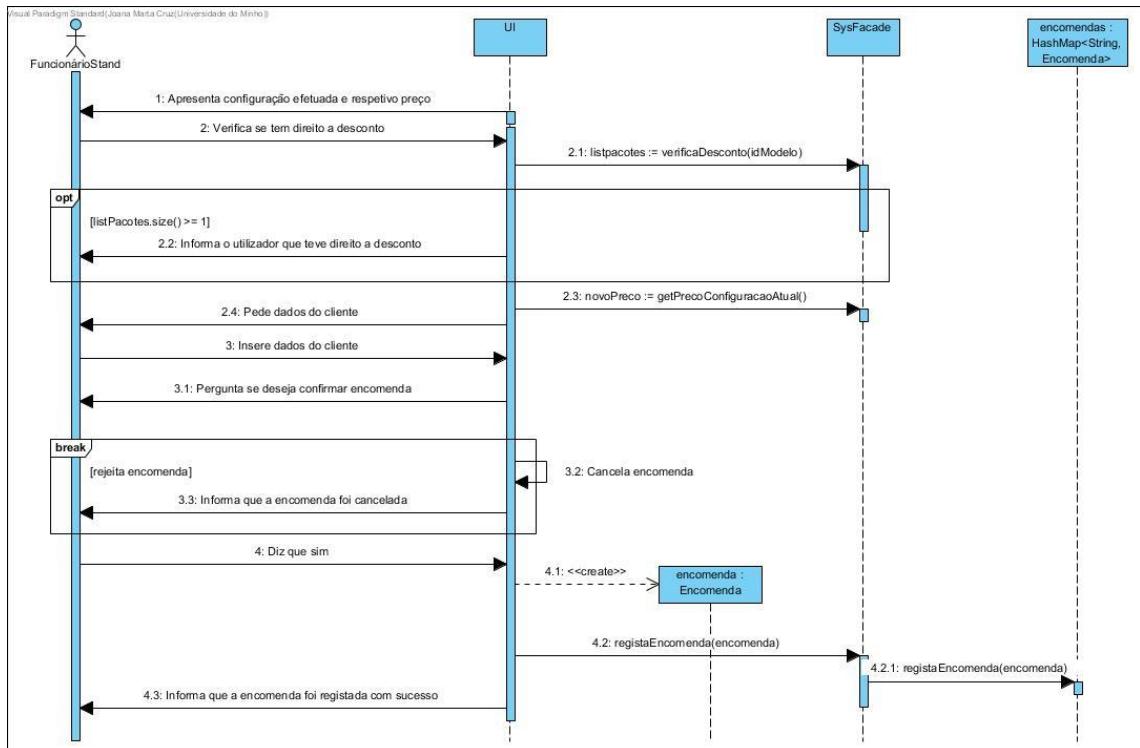


Figura 82 - Diagrama de sequência de implementação para o use case Registar Encomenda

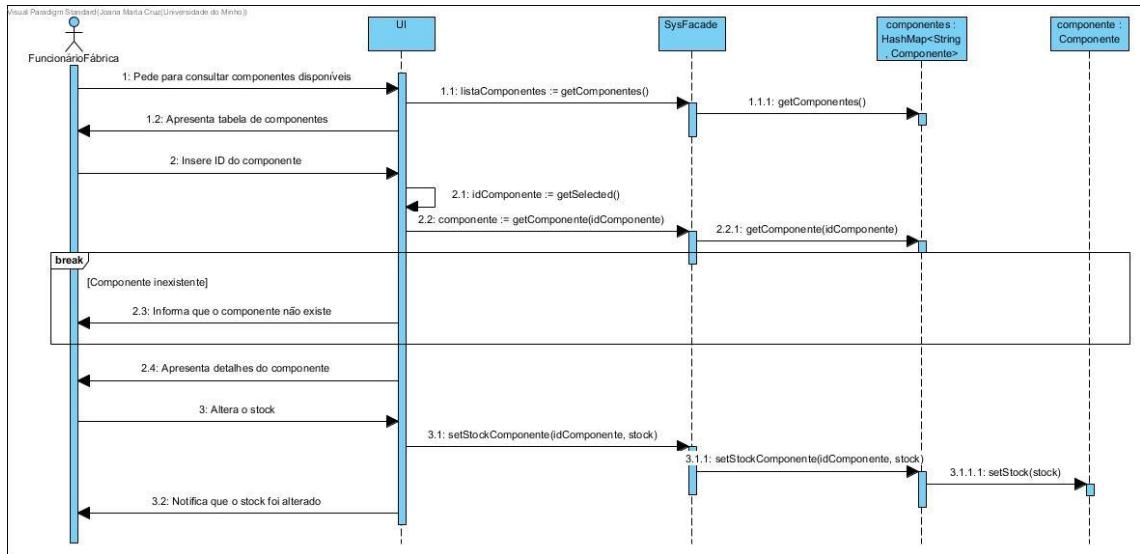


Figura 83 - Diagrama de sequência de implementação para o use case Gerir stock de componentes

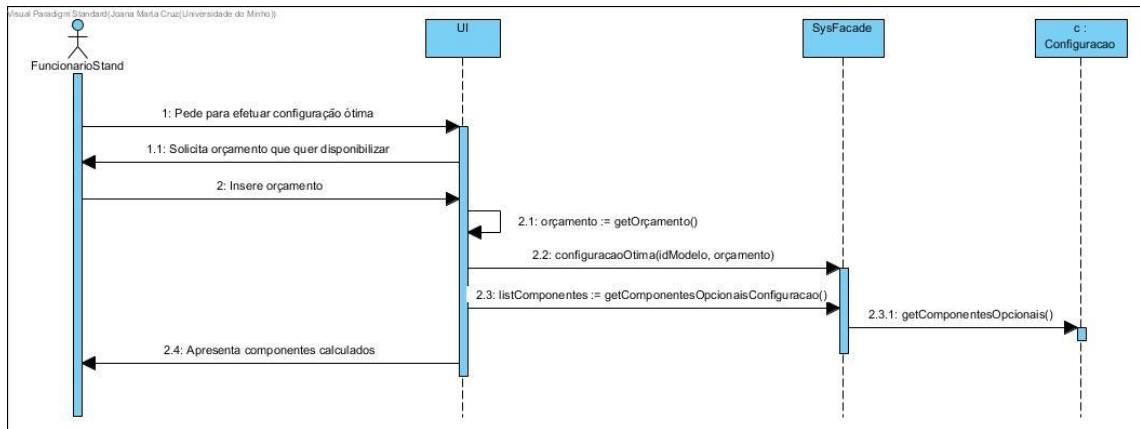


Figura 84 - Diagrama de sequência de implementação para o use case Efetuar Configuração ótima

2.5.2. Diagramas de sequência de implementação com DAO

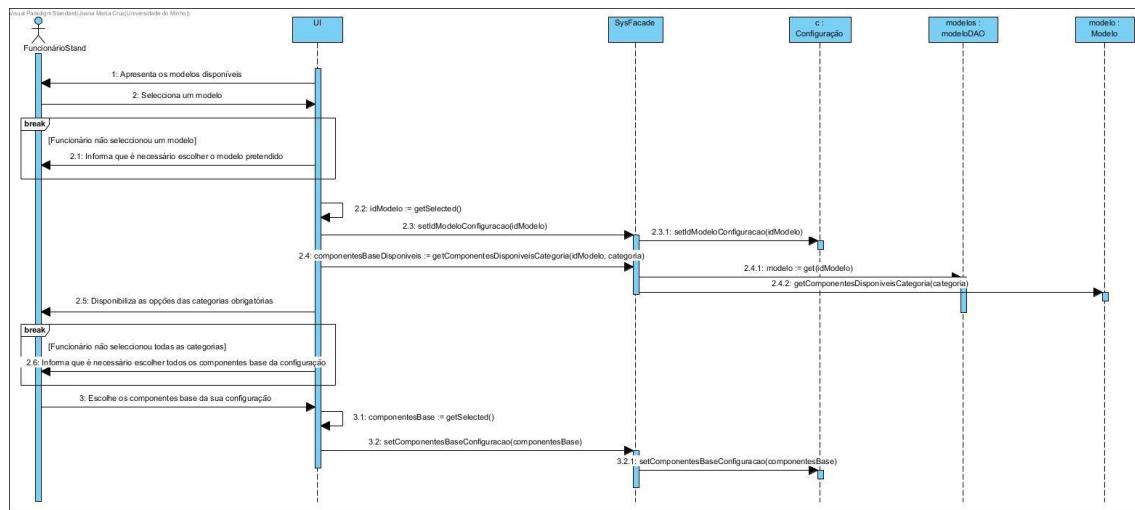


Figura 85 - Diagrama de sequência de implementação para o use case Efetuar Configuração Base

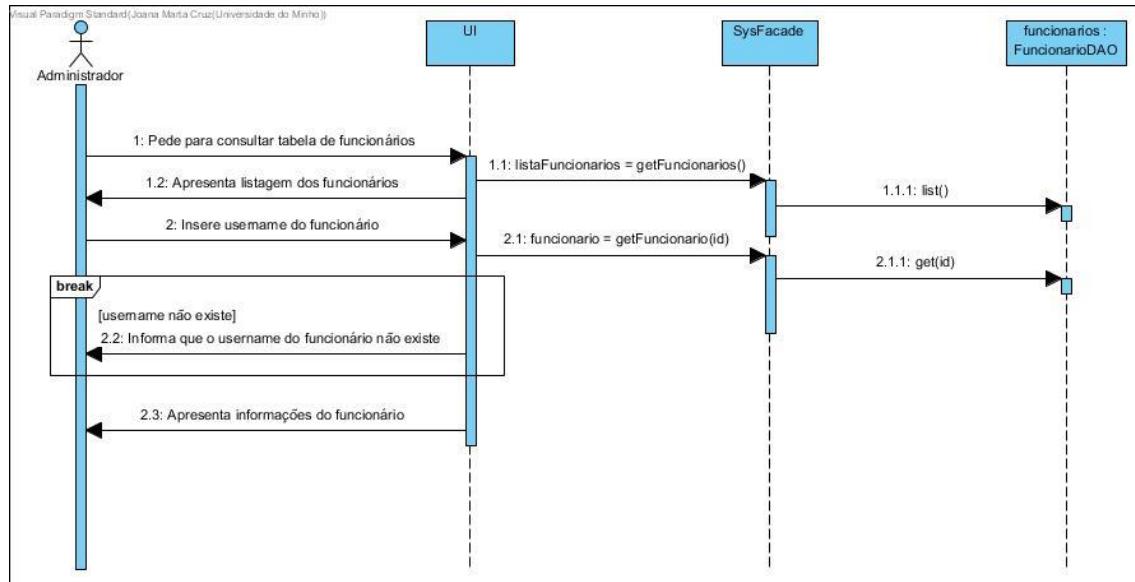


Figura 86 - Diagrama de sequência de implementação para o use case Consultar Funcionário

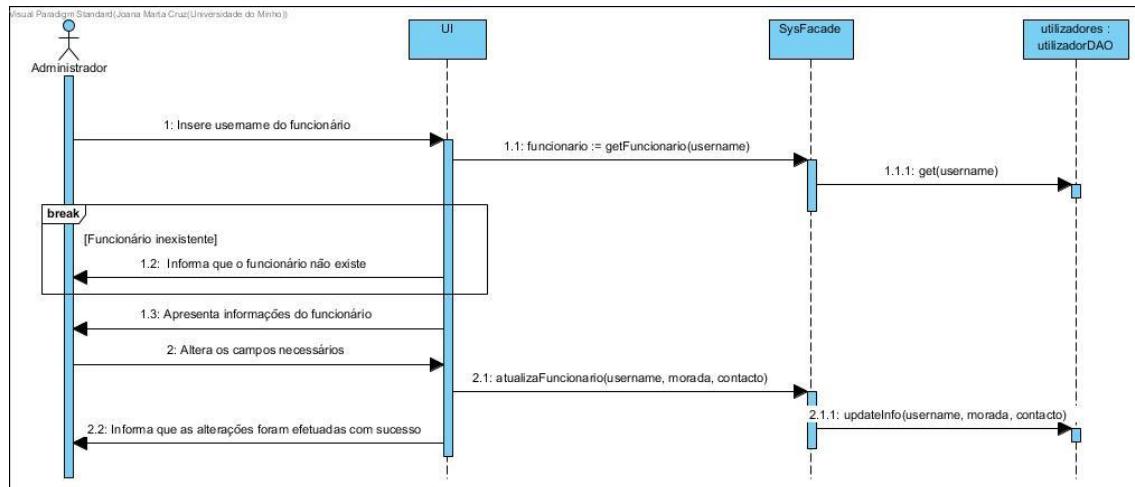


Figura 87 - Diagrama de sequência de implementação para o use case Atualizar Funcionário

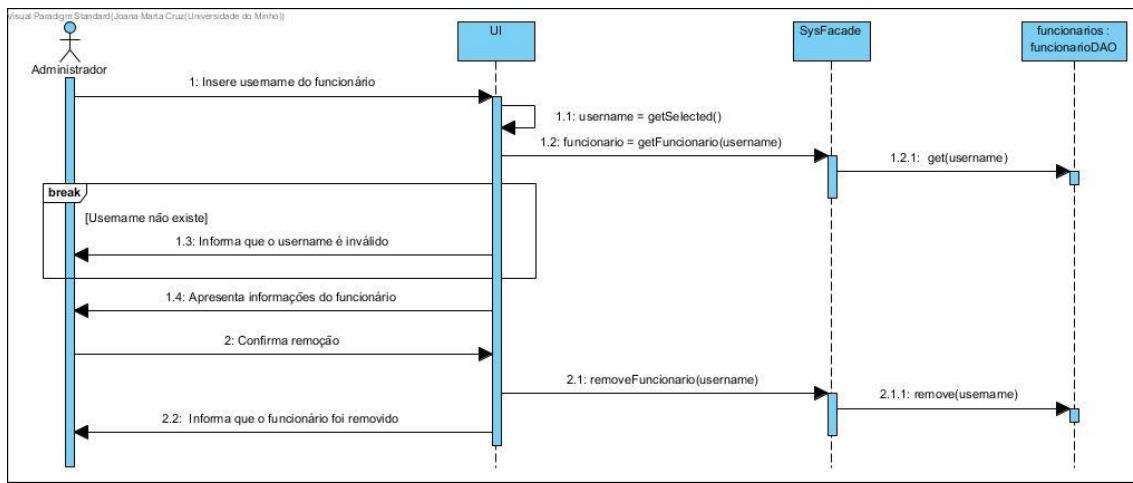


Figura 88 - Diagrama de sequência de implementação para o use case Remover Funcionário

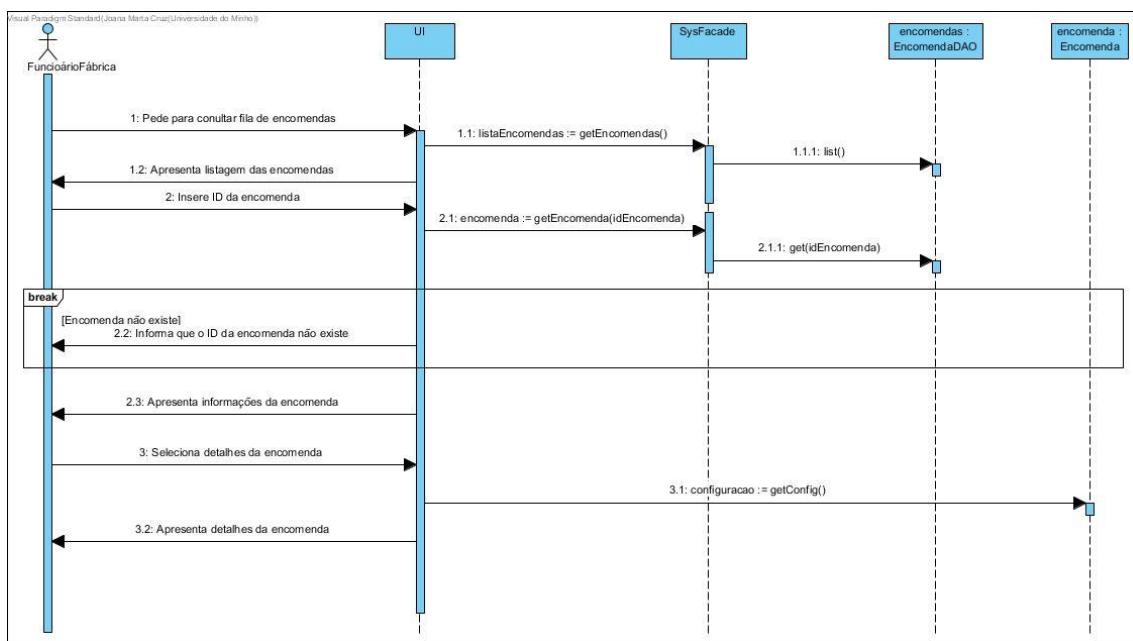


Figura 89 - Diagrama de sequência de implementação para o use case Consultar Encomenda

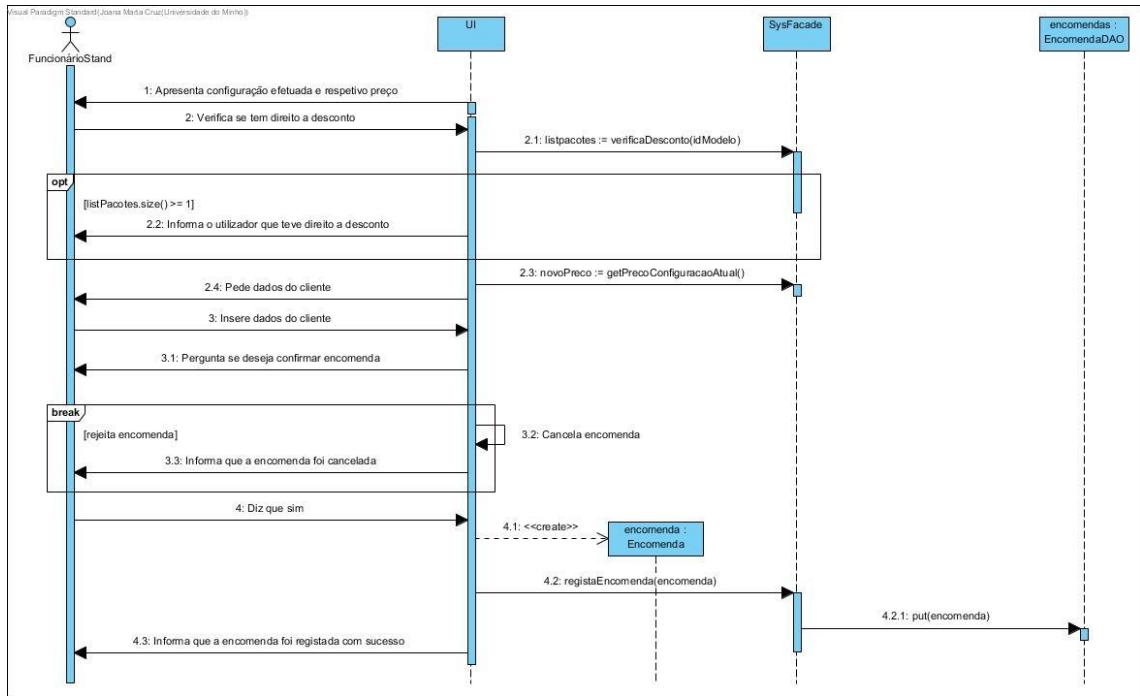


Figura 90 - Diagrama de sequência de implementação para o use case Registar Encomenda

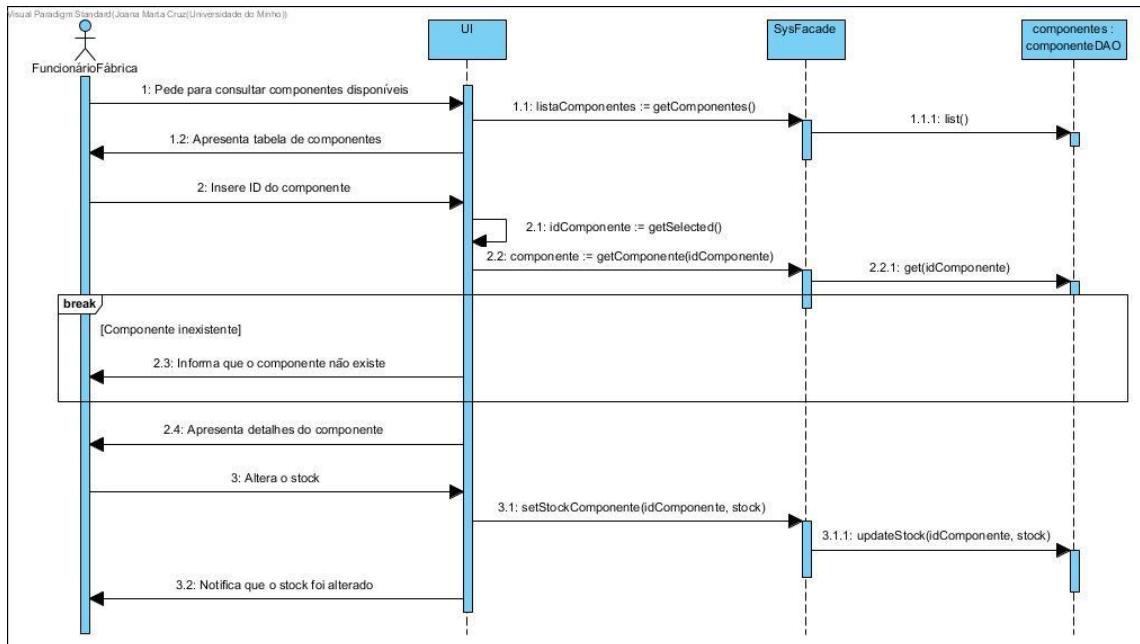
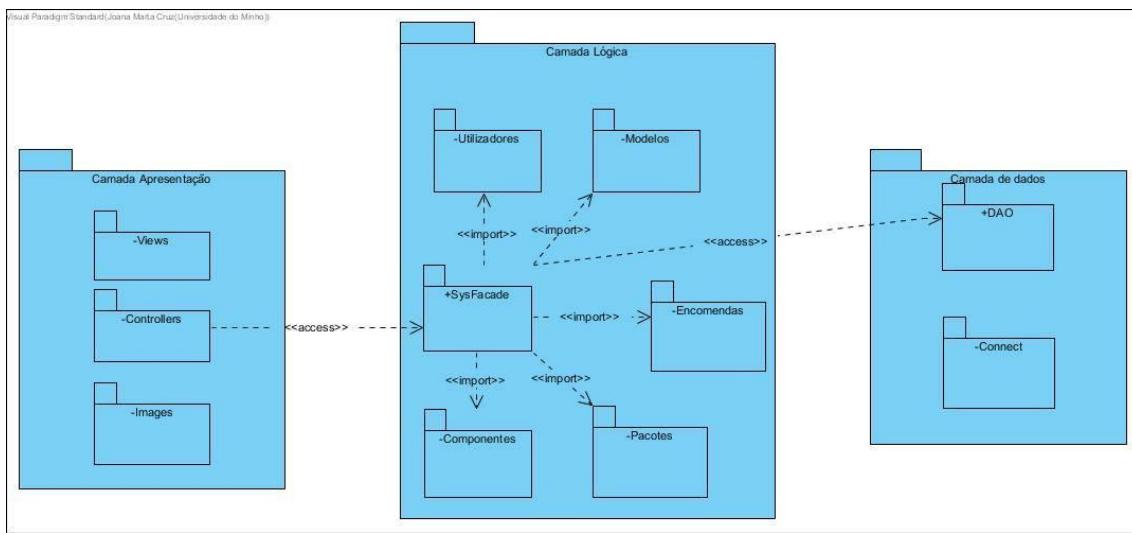


Figura 91 - Diagrama de sequência de implementação para o use case Gerir stock de componentes

2.6. Diagrama de packages



2.7. Diagrama de Sequência de nível de implementação (métodos mais relevantes)

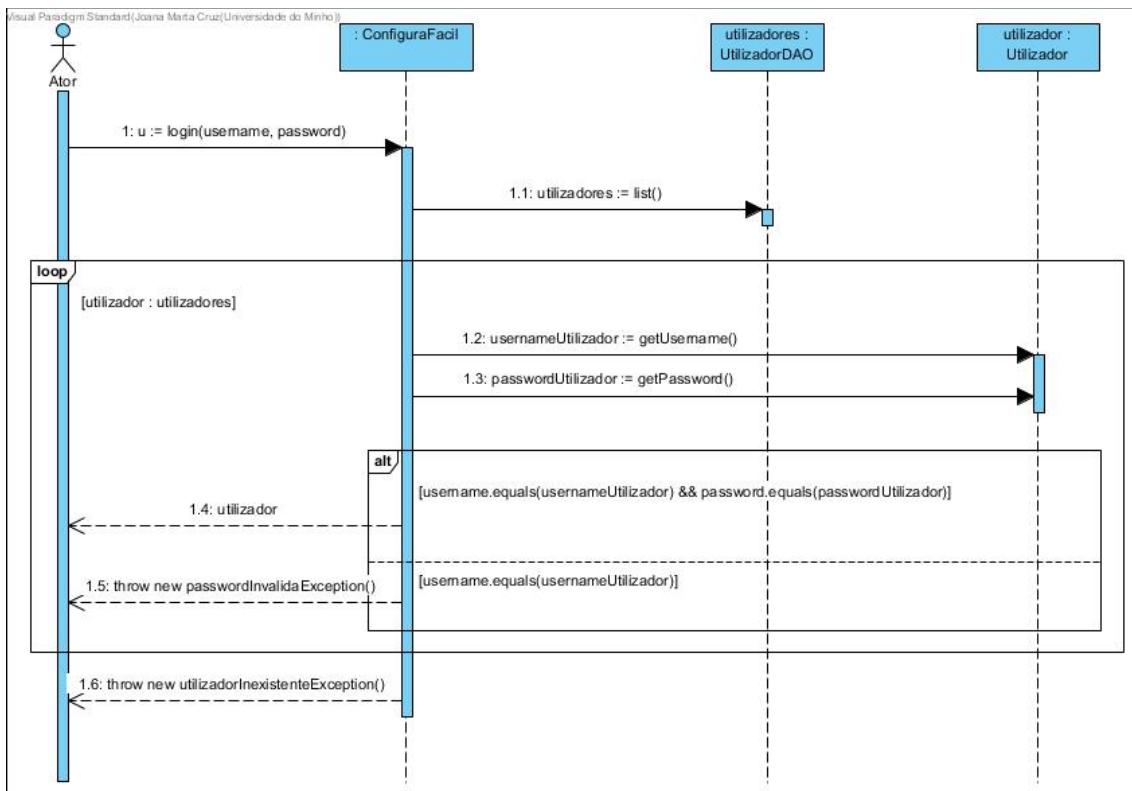


Figura 92 - Diagrama de sequência de implementação para o método login

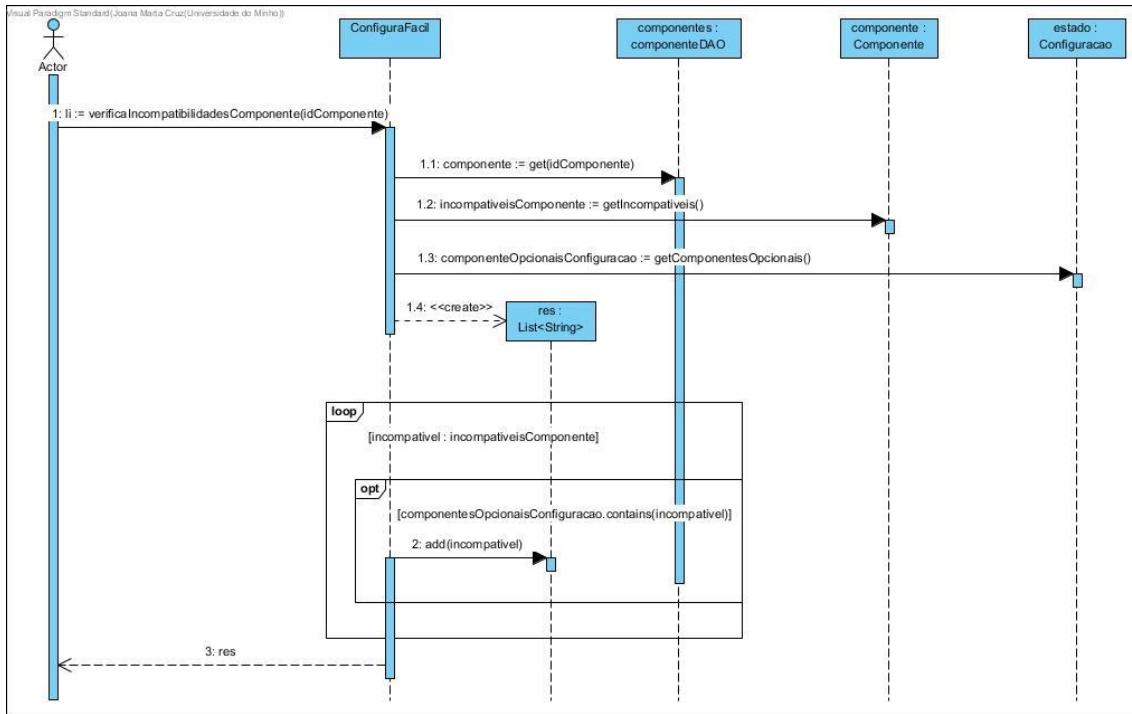


Figura 93 - Diagrama de sequência de implementação para o método verificaIncompatibilidadesComponente

Método `verificaDependenciasComponente` é usado no Adicionar Componente, para calcular os dependentes de um componente que se pretende adicionar excluindo os dependentes que possam já integrar a configuração atual. Ou seja, caso adicione um componente tem-se obrigatoriamente de adicionar os seus componentes.

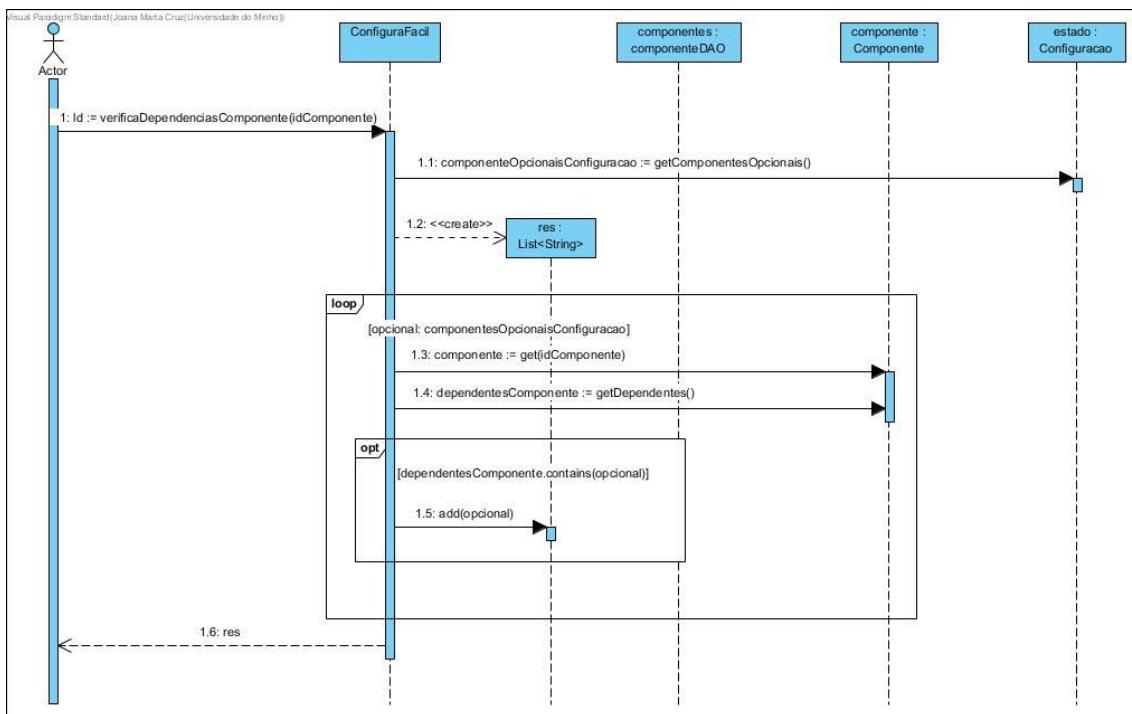


Figura 94 - Diagrama de sequência de implementação para o método verificaDependenciasComponente

Método `verificaDependentesComponente` é usado no Remover Componente, para verificar se existe algum ou mais componentes que dependam dele na configuração atual, este método verifica a lista dos dependentes de todos os componentes opcionais da configuração atual, e caso pertença a alguma dessas listas remove esses componentes. Ou seja, caso remova um componente, tenho que remover os que dependem dele na configuração atual.

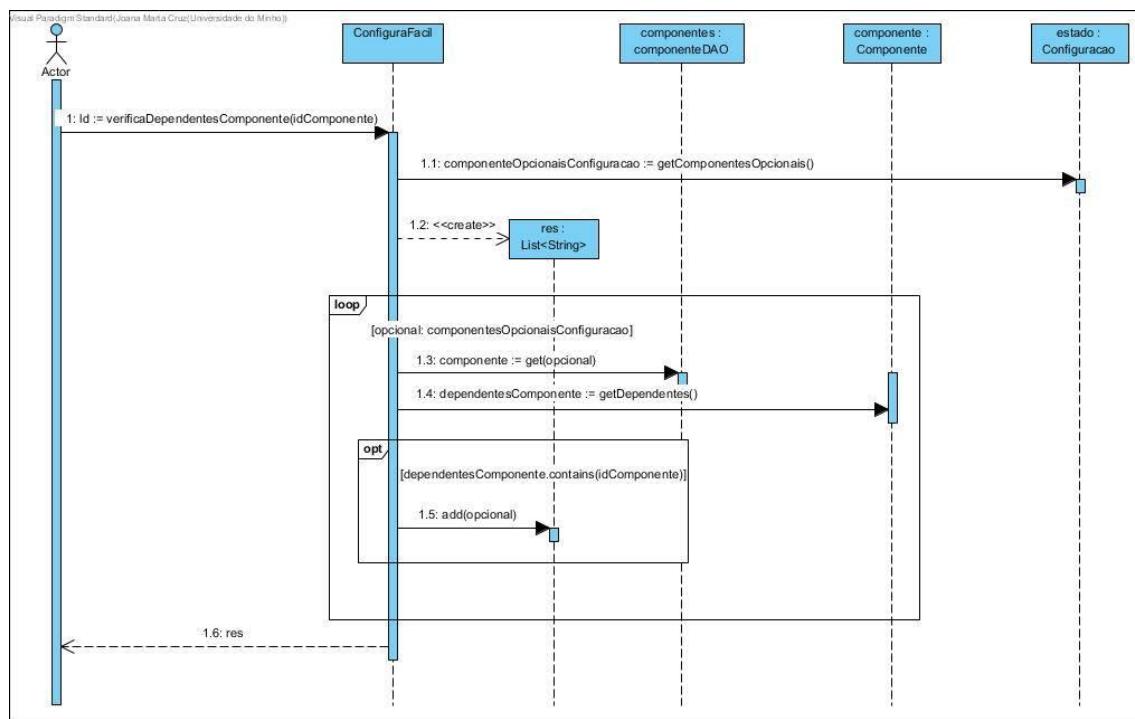


Figura 95 - Diagrama de sequência de implementação para o método `verificaDependentesComponente`

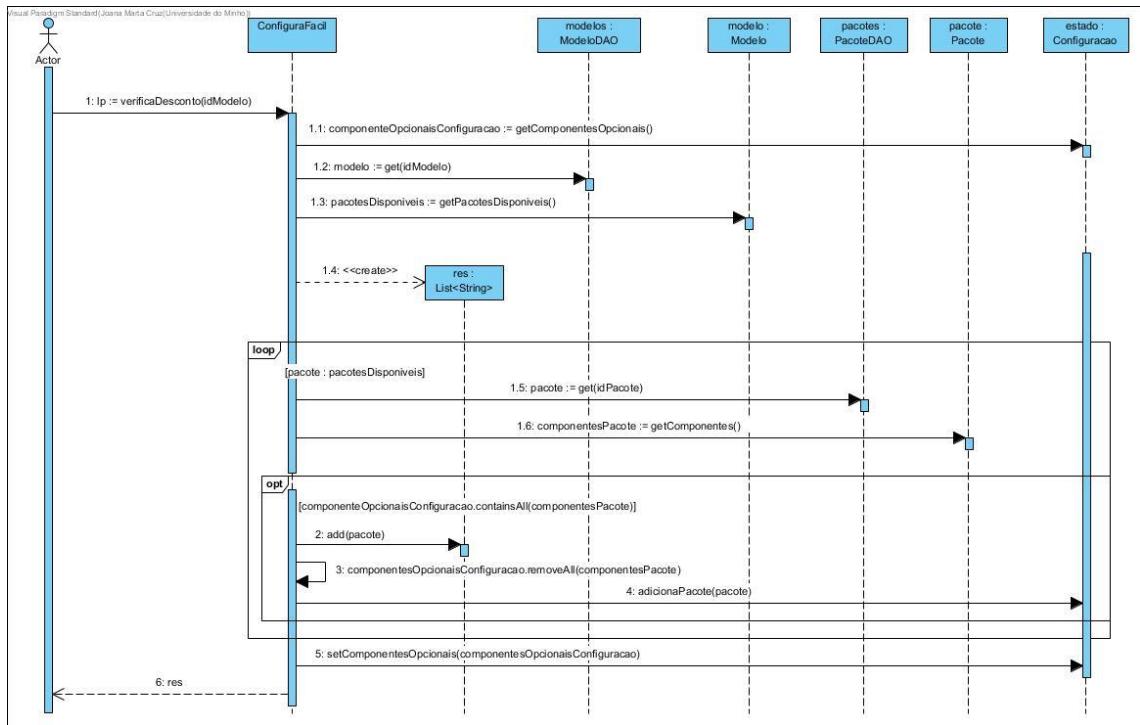


Figura 96 - Diagrama de sequência de implementação para o método verificaDesconto

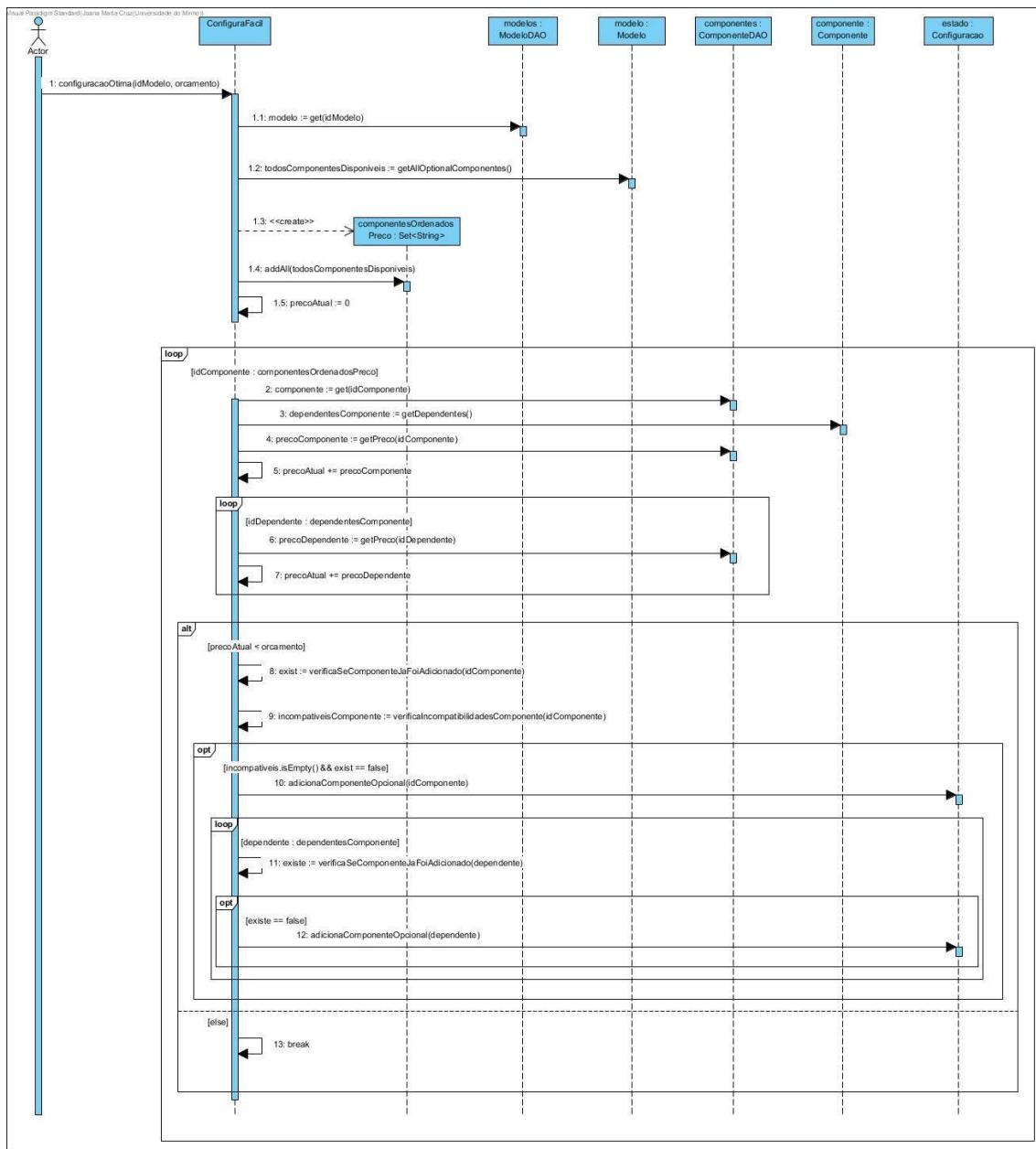


Figura 97 - Diagrama de sequência de implementação para o método Configuração ótima

2.8. Conclusão

Graças a este projeto e todo o conteúdo obtido na cadeira de Desenvolvimento de Sistemas de Software, foi possível identificar os vários capítulos de estudo, sendo eles:

- O modelo de domínio, que permitiu esclarecer todos os relacionamentos e entidades necessárias para suportar todo o sistema de auxílio ao modelo de negócio em questão, neste caso uma aplicação para configuração de veículos a ser usada num stand.
- O Diagrama de Use Cases, que por sua vez traduz toda funcionalidade do sistema.
- Especificação dos Use Cases, que traduz todo o comportamento que um 'Actor' pode tomar em cada funcionalidade do sistema.
- Prototipagem da interface, que permitiu, através de esboços, ter uma noção de como iria ficar a aplicação.
- Máquinas de estado, que possibilitaram entender ainda melhor, qual a interação entre os diferentes estados do programa e uma melhor orientação no desenvolvimento do código Java a nível de métodos.
- Diagramas de Classe, que nos deu uma ideia clara da estruturação de todo o código, acompanhado pelos Diagramas de Sequência dos subsistemas e implementação, pois estes ajudaram bastante das classes e métodos precisos para cada uma delas.

Num espírito mais crítico para com este projeto, na análise de requisitos não houve grande dificuldade na sua definição. Pode-se dizer que foi relativamente acessível a definição da interface do programa, pois os esboços previamente feitos do aspeto da aplicação foram praticamente respeitados, havendo apenas algumas alterações porque no decorrer do projeto tentou-se cada vez mais corresponder às necessidades de um ambiente real do negócio.

A respeito dos Use Cases e das suas especificações estes sofreram alguns refinamentos desde a primeira fase de entrega do projeto pois com o decorrer do tempo apercebemo-nos de algumas situações que tinha de ser abordadas e tida em conta o seu tratamento, assim como o aumento da eficiência de funcionalidades já anteriormente proposta.

De um modo geral, concretizamos toda a modelação do projeto, assim como a sua implementação.