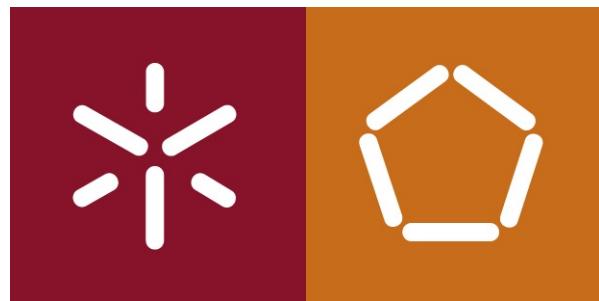


Desenvolvimento de Sistemas de Software



ConfiguraFácil

Project Report
Grupo 25



Davide Matos
A80970



Francisco Freitas
A81580



Maria Dias
A81611



Pedro Freitas
A80975

Universidade do Minho
January 5, 2019

Contents

1	Introdução	1
2	Modelação	2
2.1	Modelo de Domínio	2
2.1.1	Alterações realizadas no modelo	2
2.1.2	Entidades relevantes	2
2.2	Modelo de Use Cases	3
2.2.1	Atores	4
2.3	Use Cases	5
2.3.1	Efetuar Login	5
2.3.2	Logout	5
2.3.3	Registar Funcionário	6
2.3.4	Escolher Carro	6
2.3.5	Inserir Componente	7
2.3.6	Remover Componente	8
2.3.7	Escolher Pacote	9
2.3.8	Configuração Ótima	10
2.3.9	Gerar Configuração	10
2.3.10	Encomendar Configuração	11
2.3.11	Validar Encomenda	11
2.3.12	Consultar Encomendas	12
2.3.13	Verificar Stock	12
2.3.14	Adicionar Stock	12
2.4	Interface	13
2.4.1	Iniciar sessão	13
2.4.2	Menu inicial do funcionário	14
2.4.3	Menu inicial do gestor	14
2.4.4	Menu Principal	15
2.4.5	Escolher modelo de carro	16
2.4.6	Menu dos componentes obrigatórios	16
2.4.7	Escolher Detalhes	19

2.4.8	Escolher componentes opcionais	20
2.4.9	Escolher pacote de configuração	20
2.4.10	Efetuar configuração ótima	20
2.4.11	Remover componente	21
2.4.12	Enviar encomenda	21
2.4.13	Menu Fábrica	22
2.4.14	Verificar e adicionar stock	23
2.4.15	Ver e registar funcionários	23
2.4.16	Ver encomendas	24
2.4.17	Máquina de Estado da navegação	25
2.5	Diagrama de Sequência de Sistema	27
2.5.1	Login	28
2.5.2	Escolher carro	28
2.5.3	Inserir Componente	29
2.5.4	Remover Componente	30
2.5.5	Configuração Ótima	31
2.5.6	Encomendar Configuração	31
2.5.7	Consultar Encomendas	32
2.5.8	Verificar Stock	32
2.5.9	Adicionar Stock	32
2.6	Diagrama de Sequência de Subsistemas	33
2.6.1	Login	33
2.6.2	Escolher carro	33
2.6.3	Inserir Componente	34
2.6.4	Remover Componente	35
2.6.5	Configuração Ótima	36
2.6.6	Encomendar Configuração	37
2.6.7	Consultar Encomendas	38
2.6.8	Verificar Stock	38
2.6.9	Adicionar Stock	38
2.7	Diagrama de Packages	39
2.8	Diagrama de Classes	41
2.8.1	Diagrama de Classes	41
2.9	Diagrama de ORM	42
2.10	Diagrama de Sequência de Implementação	43
2.10.1	Login	43
2.10.2	Registrar Funcionário	44
2.10.3	Escolher Carro	45
2.10.4	Inserir Componente	46
2.10.5	Escolher Pacote	50
2.10.6	Configuração Ótima	51

Contents	iii
2.10.7 Encomendar Configuração	55
2.10.8 Consultar Encomendas	56
2.10.9 Verificar Stock	57
2.10.10 Adicionar Stock	58
2.11 Diagrama de Instalação	59
2.12 Base de Dados implementada	60
3 Conclusão	61

Chapter 1

Introdução

Neste trabalho é pedido o desenvolvimento de um sistema de suporte para uma aplicação de configuração de carros. O sistema desenvolvido deve permitir, num stand e junto de um cliente, gerar uma configuração para a encomenda de um carro novo. O objetivo deste relatório consiste em esclarecer o método usado na resolução do problema proposto pelo enunciado. Iremos, ao longo deste relatório, explicitar a nossa interpretação do enunciado, apresentar as escolhas feitas durante a análise e modelação do projeto e as justificações para as mesmas.

Ao longo da realização do Modelo de Domínio e do Diagrama dos Use Case, tivemos de tomar várias decisões que, consequentemente, tiveram bastante impacto no resultado final da primeira fase do trabalho. Primeiramente, decidimos que a nossa aplicação iria possibilitar a escolha do modelo de carro a configurar, e que apenas um gestor estaria elegível para criar conta relativa a um funcionário, por questões de segurança. À medida que fomos desenvolvendo o projeto, foram surgindo algumas melhorias na modelação da nossa aplicação, de forma a tornar possível a implementação das funcionalidades do sistema. Consideremos, por exemplo, os modelos de Use Case, onde após alguma reflexão sobre o modelo que foi esquematizado na primeira parte do projeto, alteramos alguns aspectos. Também o próprio modelo de Domínio sofreu alterações. Ao longo deste relatório iremos apresentar as alterações feitas e a razão das mesmas.

Em relação a estratégia usada para a funcionalidade de o sistema criar uma configuração ótima quando lhe é dado um orçamento para tal, foi decidido que a configuração criada pelo sistema será apenas para componentes opcionais e detalhes e que tenta ser o mais economicamente eficiente possível. Para tal inserimos, se o orçamento o permitir, o Pacote com o melhor preço médio por componente dando assim acesso ao desconto correspondente. De seguida adicionamos os componentes mais baratos enquanto o orçamento o permitir. Na inserção de Componentes evitamos todos os componentes que dependem de outros porque iria sempre adicionar custos extras.

Chapter 2

Modelação

2.1 Modelo de Domínio

O Modelo de Domínio captura as entidades da nossa aplicação e os relacionamentos entre elas.

2.1.1 Alterações realizadas no modelo

Em relação ao modelo apresentado na primeira fase deste projeto, foi feita apenas uma alteração, que consiste na adição da entidade "carro", que é associado a uma configuração, uma vez que na nossa aplicação, para além de componentes, é também possível escolher o modelo do carro a configurar.

2.1.2 Entidades relevantes

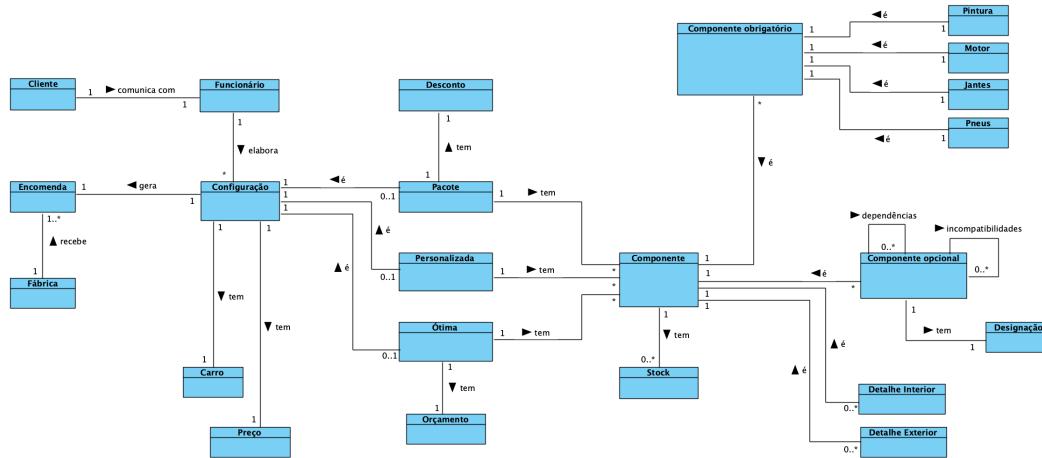


Figure 2.1: Modelo de Domínio

2.2 Modelo de Use Cases

Nesta segunda fase do projeto, tivemos a necessidade de alterar alguns Use Cases de forma a cumprirem os requisitos propostos da melhor forma. Para além de terem sido alterados alguns detalhes de use cases já existentes, foi adicionado o use case "Validar Encomenda", que consiste em passar uma encomenda do estado "Em espera" para "Em execução", sendo que nesta mudança de estado são decrementados os stocks dos vários componentes incluídos nas configurações encomendadas. Isto não era possível anteriormente, em que apenas tínhamos o use case "Enviar Encomenda", que transforma uma configuração numa encomenda associada a um cliente e adiciona-a à fila de encomendas em espera, mas não a envia para produção nem altera os stocks dos componentes.

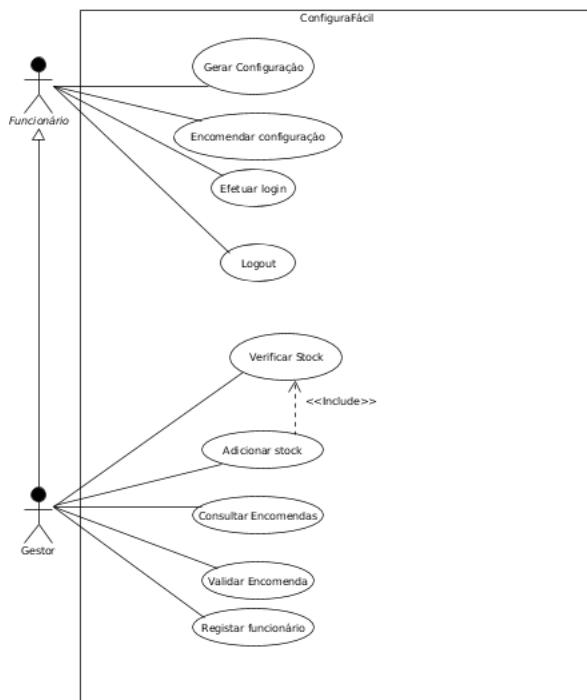


Figure 2.2: Modelo de Use Cases Geral

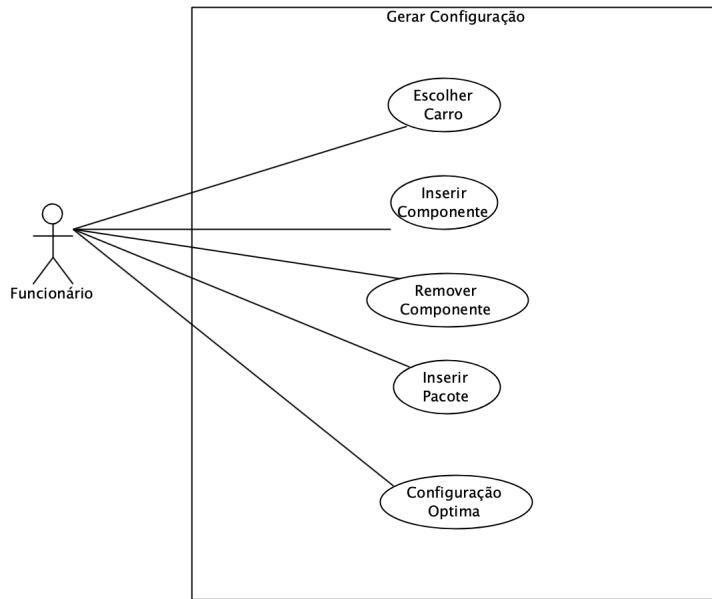


Figure 2.3: Modelo de Use Cases para gerar a configuração

2.2.1 Atores

Funcionário

Representa um funcionário da aplicação, estando associadas a si as diferentes funcionalidades que apenas funcionários registados e o gestor podem utilizar.

Gestor

Representa um gestor da aplicação, podendo desempenhar as mesmas funcionalidades de um funcionário e ainda fazer a gestão da fábrica (que inclui gestão de encomendas, funcionários e stock).

2.3 Use Cases

2.3.1 Efetuar Login

Efetuar Login		
Use Case:		
Actor:	Funcionário/Gestor	
Pré condição:	Não estar autenticado na aplicação	
Pós condição:	Ficar autenticado no sistema	
Cenário Normal	Actor input	System response
	1. Indica que quer iniciar sessão	2. Pede as credenciais
	3. Fornece as credenciais	4. Autentica as credenciais
		5. Utilizador com sessão iniciada
		4.1 O sistema informa que as credenciais estão incorretas
Exceção 1 [Credenciais inválidas] Passo 4		

Figure 2.4: Use case efetuar login

Após um funcionário estar registado na aplicação, este pode iniciar sessão na aplicação, fornecendo o respetivo username e password. Devido ao processo de estruturação da implementação, achamos por bem mudar este use case, para em vez de fornecer apenas as credenciais, pedir primeiro ao sistema para iniciar sessão.

2.3.2 Logout

Logout		
Actor:	Funcionario	
Pré condição:	Funcionario autenticado	
Pós condição:	Funcionario deixa de estar autenticado	
Cenário Normal	Actor input	System Response
	1. Indica que pretende terminar sessão	2.Processa pedido

Figure 2.5: Use case verificar stock

Os utilizadores podem terminar sessão sempre que o desejarem, deixando assim de poder usufruir da aplicação.

2.3.3 Registar Funcionário

Use Case:	Registrar Funcionário	
Actor:	Gestor	
Pré condição:	O Gestor deve ter feito "log-in" e obtido autorização do sistema	
Pós condição:	O Funcionário é registado e é lhe fornecido as credenciais	
Cenário Normal	Actor input	System response
	1. Indica que pertende registrar um funcionario	2. Pede o nome e a password do funcionário
	3. Fornece o dados do funcionário	4. Verifica se nenhum outro funcionário tem o mesmo nome
		5. Processa o registo do funcionário
	3.1 Cancela registo	
Exceção 1 [cancela inserção] (Passo 3)		
Exceção 2 [o nome já existe] (Passo 4)		4.1 Informa que o nome já existe

Figure 2.6: Use case registrar funcionário

Para um funcionário fazer parte do registo de funcionários da aplicação, este deve ser adicionado pelo gestor, tendo este de fornecer um username e uma palavra-passe para o novo utilizador. Considere-se que o gestor define o nome de utilizador e a password de cada funcionário. Numa fase onde já estariamos a pensar na implementação da aplicação, chegamos à conclusão que este usecase também precisaria de algumas alterações em relação à 1ª fase. Nisto tivemos de colocar exceções para caso o gestor cancela a operação ou caso ele tente registrar um funcionário com um username já utilizado.

2.3.4 Escolher Carro

Para escolher o modelo do carro do qual se pretende efetuar uma configuração, o funcionário tem de escolher, de entre as opções apresentadas pelo sistema, aquela que lhe é indicada pelo cliente, e a informação relativa ao modelo do carro é adicionada pelo sistema à configuração.

Use Case:	Escolher Carro	
Actor:	Funcionário	
Pré condição:	O Funcionário deve ter feito "log-in" e obtido autorização do sistema	
Pós condição:	O Carro é escolhido e guardado na Configuração	
Cenário Normal	Actor input	System response
	1. Indica que pretende escolher carro	2. Apresenta as opções de escolha do carro
	3. Seleciona carro	4. Processa a seleção do carro
	3.1 Cancela escolha	
Exceção 1 [cancela escolha] (Passo 3)		

Figure 2.7: Use case escolher carro

2.3.5 Inserir Componente

Use Case:	Inserir Componente	
Actor:	Funcionário	
Pré condição:	O funcionário deve ter feito "log-in" e obtido autorização do sistema	
Pós condição:	O componente deve ser inserido e gravado na configuração, se possível	
Cenário Normal	Actor input	System response
	1. Indica que pertence inserir componente	2. Apresenta as opções das componentes
	3. Seleciona a componente	4. Verifica se o componente escolhido não implica componentes incompatíveis ou componentes obrigatórios 5. O sistema adiciona componente à configuração.
	4.3. O funcionário responde	4.1.O sistema informa da incompatibilidade 4.2.O sistema pergunta ao funcionário se quer cancelar a adição do componente ou alterar a configuração atual 4.4. O sistema guarda a alteração
Comportamento Alternativo 1 [O componente é incompatível com a configuração atual] Passo 4		
	4.3. O funcionário responde	4.1.O sistema informa da dependência de componentes 4.2.O sistema pergunta ao funcionário se pretende adicionar os componentes que geram a dependência ou se quer cancelar a adição do componente 4.4. O sistema guarda a alteração
Comportamento Alternativo 2 [O componente implica a adição de outros componentes] Passo 4		
Exceção 3 [cancela inserção] (Passo 3/4.3/4.3)	3.1 Cancela inserção	

Figure 2.8: Use case inserir componente

Quando um utilizador pretende inserir um componente na configuração, o sistema começa por apresentar as opções de entre as quais o utilizador pode escolher. Este, por sua vez, faz a sua escolha, e o sistema trata de avaliar se o componente selecionado obriga ou impede outros componentes a fazerem parte da configuração. Em caso afirmativo, o sistema apresenta uma mensagem ao utilizador, perguntando se o mesmo pretende adicionar os componentes obrigatórios ou eliminar os componentes incompatíveis, respetivamente. Cabe ao utilizador escolher o que pretende fazer. Houve pequenas alterações em relação à 1ª fase devido à funcionalidade desejada para a aplicação a implementar, criando assim mais uma interação entre o funcionário e o sistema. Antes de este seleccionar o componente em si ele informa o sistema que quer inserir um componente, e o sistema responde com todas as opções disponíveis.

2.3.6 Remover Componente

Remover Componente		
Use Case:	Funcionário	
Actor:	O Funcionário deve ter feito "log-in" e obtido autorização do sistema	
Pré condição:	Remove as Componentes que pretendia da configuração	
Cenário Normal	Actor input	System response
	1. Indica que pretende remover Componente	2. O sistema apresenta as Opções
	3. Escolhe componentes a remover	4. Remove as Componentes da configuração
Exceção 1 [A configuração não tem componentes] Passo 2		2.1 Informa que a configuração não tem componentes
Alternativo 1 [Componente apresenta dependências] Passo 4	4.2 Confirma	4.1 Pergunta se quer eliminar componentes dependentes 4.3 Processa remoção dos dependentes
Exceção 2 [O Ator decide cancelar a remoção] Passo 3	3.1 O funcionário indica que pretende cancelar	3.2 O Sistema cancela a remoção.
Exceção 3 [O Ator decide cancelar a remoção do componente dependente] Passo 4.2	4.2.1 Cancela remoção	4.2.2 O sistema cancela a remoção

Figure 2.9: Use case remover componente

Quando um utilizador pretende remover um componente da configuração, deve escolher, de entre a lista de componentes incluídos na configuração atual, qual o componente a remover. De forma análoga ao use case anterior, o sistema verifica se existem dependências ou incompatibilidades entre o componente que o utilizador pretende remover e os restantes componentes da configuração.

2.3.7 Escolher Pacote

Use Case:	Escolher Pacote de Configuração	
Actor:	Funcionário	
Pré condição:	O funcionário deve ter feito "log-in" e obtido autorização do sistema	
Pós condição:	O pacote é guardado como configuração e segue para o último passo da mesma.	
Cenário Normal	Actor input	System response
	1. Indica que pertence escolher um pacote	2. Processa o pedido 3. Apresenta a lista de pacote
	4. Seleciona pacote	5. Processa a escolha do pacote
Exceção 1 [Cancela escolha] (Passo 4)	4.1. Cancela seleção	

Figure 2.10: Use case configuração pacote

Um funcionário que queria selecionar um pacote de configuração deve selecionar um dos pacotes disponíveis na aplicação. O sistema adiciona-o então à configuração. Em relação há 1^a fase, houve pequenas alterações tendo em vista a funcionalidade da futura aplicação. Antes do funcionário selecionar o pacote desejado, ele informa o sistema que quer uma configuração por pacote e só depois de lhe serem apresentados os pacotes disponíveis é que este o pode selecionar.

2.3.8 Configuração Ótima

Use Case:	Efetuar Configuração Ótima	
Actor:	Funcionário	
Pré condição:	O funcionário deve ter feito "log-in" e obtido autorização do sistema	
Pós condição:	A configuração ótima é efetuada e gravada, se possível	
Cenário Normal	Actor input	System response
	1. Indica que pertence usar a funcionalidade de configuração ótima	2. Processa o pedido 3. Pede o orçamento
	4. Fornece orçamento	6. Processa a configuração
	5. Funcionário confirma	5.1 Não confirma/cancelar 5.2 Cancela pedido
Exceção 1 [funcionário não concorda com configuração e quer cancelar] Passo 5		

Figure 2.11: Use case configuração ótima

Um funcionário pode, a pedido do cliente, selecionar a opção Configuração Ótima fornecendo o orçamento que está disposto a gastar.

2.3.9 Gerar Configuração

Use Case:	Gerar Configuração	
Actor:	Funcionário	
Pré condição:	O Funcionário deve ter feito "log-in" e obtido autorização do sistema	
Pós condição:	Cria uma configuração	
Cenário Normal	Actor input	System response
	1. Indica que pertence criar configuração	2. O sistema apresenta as opções
	3. Escolhe opções que pertence	
Exceção 1 [O Ator decide cancelar a remoção] Passo 3	3.1 O funcionário indica que pertence cancelar	3.2 O Sistema cancela a criação da configuração

Figure 2.12: Use case gerar configuração

Depois dos componentes obrigatórios e opcionais estarem todos selecionados é possível ao funcionário gerar uma configuração com estes componentes.

2.3.10 Encomendar Configuração

Encomendar Configuração		
Use Case:		
Actor:	Funcionário	
Pré condição:	O funcionário deve ter efetuado e registado todas as configurações escolhidas pelo cliente	
Pós condição:	A encomenda da configuração do carro é enviada à fábrica	
Cenário Normal	Actor input	System response
	1. Indica que pertence enviar encomenda	2. Sistema pede a informação relativa ao cliente
	3. Fornece a informação	
	4. Confirma	
		5. Processa o envio
Exceção 1 [cancela encomenda] (Passo 4)	4.1 Não confirma	4.2 Cancela Encomenda

Figure 2.13: Use case encomendar configuração

Um funcionário pode enviar uma configuração de um carro para a fábrica, de forma a que esta entre na fila de espera para ser produzida. Para isso, o funcionário tem de introduzir os dados do cliente e o sistema processa a encomenda. A principal mudança deste usecase em relação à 1ª fase, foi a necessidade da encomenda ter os dados do cliente que pretende fazer a mesma.

2.3.11 Validar Encomenda

Validar Encomenda		
Use Case:		
Actor:	Gestor	
Pré condição:	O Gestor deve ter feito "log-in", existir encomendas e respetivo Stock	
Pós condição:	Encomenda Validada e stock atualizado	
Cenário Normal	Actor input	System response
	1. Indica que quer validar encomenda	2. Processa pedido 3. Apresenta Encomendas
	4. Escolhe a encomenda a validar	5. Valida encomenda 6. Atualiza stocks
Exceção 1 [cancela validação] (Passo 4)	4.1 Cancela Validação	

Figure 2.14: Use case configuração pacote

Quando o gestor quer validar uma encomenda, deve escolher, de entre a lista de encomendas que podem ser feitas naquele momento com base no stock disponível. O sistema valida a encomenda e o stock dos componentes utilizados na encomenda é decrementado.

2.3.12 Consultar Encomendas

Consultar Encomendas		
Use Case:		
Actor:	Gestor	
Pré condição:	O Gestor deve ter feito "log-in" e obtido autorização do sistema	
Pós condição:	O sistema apresenta todas as encomendas por fazer	
Cenário Normal	Actor input	System response
	1. Indica que quer consultar encomendas	2. Processa o pedido 3. Apresenta uma lista de todas as encomendas e estado delas

Figure 2.15: Use case consultar encomendas

O gestor pode consultar as encomendas efetuadas a qualquer momento.

2.3.13 Verificar Stock

Verificar Stock		
Use Case:		
Actor:	Gestor	
Pré condição:	O Gestor deve ter feito "log-in" e obtido autorização do sistema	
Pós condição:	O sistema apresenta todos os componentes e respetivo Stock	
Cenário Normal	Actor input	System response
	1. Indica que quer ver o stock	2. Processa o pedido 3. O sistema apresenta uma lista de todos os componentes e respetivo Stock

Figure 2.16: Use case verificar stock

O gestor pode verificar o stock de todos os componentes existentes.

2.3.14 Adicionar Stock

Adicionar Stock		
Use Case:		
Actor:	Gestor	
Pré condição:	O Gestor deve ter feito "log-in" e obtido autorização do sistema	
Pós condição:	O sistema apresenta todos os componentes e respetivo Stock	
Cenário Normal	Actor input	System response
	1. Indica que pertence adicionar componente 4. Selecciona o componente que quer adicionar 6. Indica a quantidade 7. Confirma a adição	2. Processa pedido 3. Apresenta a lista de todos os componentes 5. Pergunta a quantidade que pretende adicionar 8. Processa a adição de stock
Exceção 3 [cancela inserção] (Passo 7)	7.1 Cancela a adição	

Figure 2.17: Use case adicionar stock

O gestor pode adicionar stock de um dado componente sempre que desejar, indicando qual o componente e a quantidade que pretende adicionar.

2.4 Interface

Na primeira fase deste projeto foi feito um esboço inicial daquele que seria o aspeto da nossa aplicação. A versão final da interface, que se apresenta de seguida, foi implementada com base no protótipo inicial, tendo sido sujeita a alterações que foram feitas de forma a que a aplicação cumprisse os requisitos esperados.

2.4.1 Iniciar sessão

A imagem que se segue é um exemplo da interface gráfica apresentada que permite ao utilizador iniciar sessão na aplicação. Caso um utilizador não registado tente efetuar login, o mesmo é informado que as credenciais inseridas estão incorretas. Na nossa aplicação, apenas o gestor está autorizado a adicionar funcionários à aplicação, uma vez que não pretendemos que seja possível que qualquer pessoa consiga registar-se. Ao limitar a autorização de registar utilizadores ao gestor, é garantido que apenas tem acesso à aplicação pessoal autorizado.

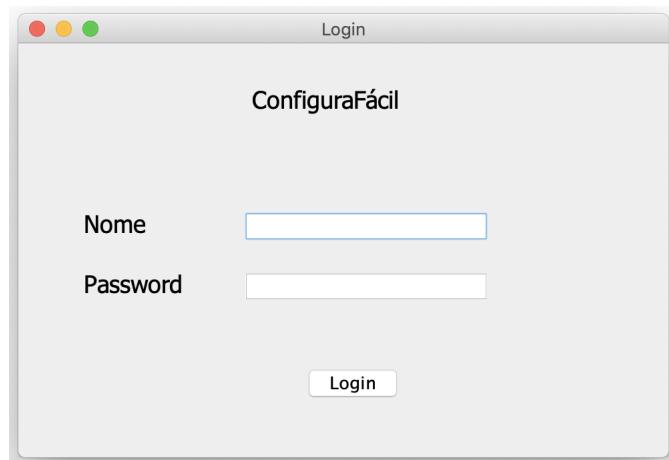


Figure 2.18: Janela de login

2.4.2 Menu inicial do funcionário

Este menu é apresentado ao funcionário quando este inicia sessão. Enquanto que o gestor pode, para além de criar configurações, fazer a gestão da fábrica (stocks, encomendas, etc.), o funcionário pode apenas executar a primeira destas ações.



Figure 2.19: Menu inicial do funcionário

2.4.3 Menu inicial do gestor

Menu apresentado ao gestor quando este inicia sessão.



Figure 2.20: Menu inicial do gestor

2.4.4 Menu Principal

A imagem que se segue representa a interface gráfica que é apresentada a um funcionário ou gestor da aplicação ConfiguraFacil que tenha iniciado sessão e escolhido a opção "Cria Configuração". Neste menu, o utilizador tem acesso a todas as funcionalidades relativas à criação de uma configuração, desde a escolha do modelo do carro à escolha de uma configuração ótima. Após escolher um modelo de carro e todos os componentes obrigatórios, o utilizador pode enviar a encomenda para a fábrica.



Figure 2.21: Menu principal

2.4.5 Escolher modelo de carro

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Carro". Neste menu, o funcionário pode escolher o modelo do carro a ser configurado.

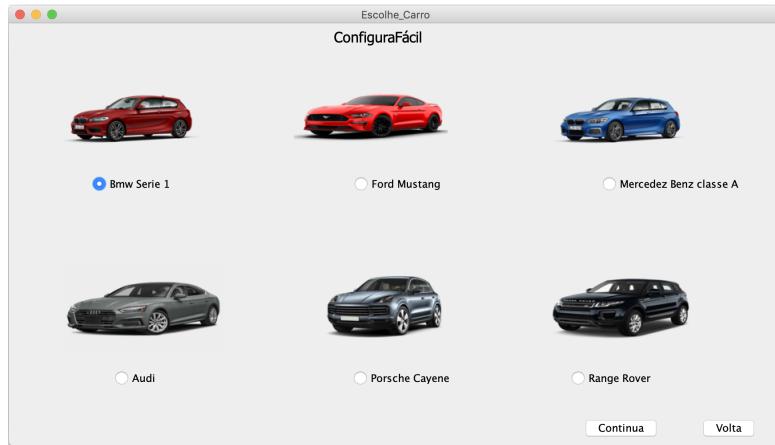


Figure 2.22: Janela de escolha de carro

2.4.6 Menu dos componentes obrigatórios

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Componentes Obrigatórios". Neste menu, o funcionário pode escolher a pintura, o motor, os pneus e as jantes do carro a ser configurado.

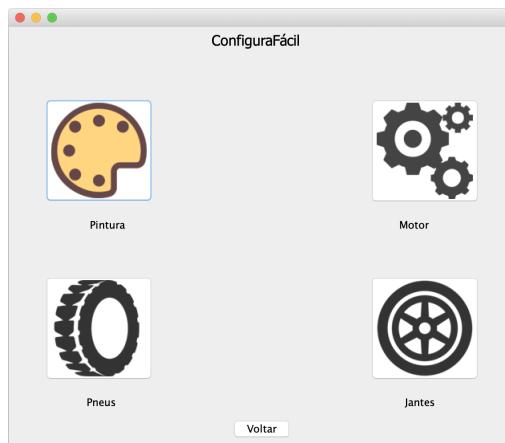


Figure 2.23: Menu de componentes obrigatórios

Escolher Pintura

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Pintura" no menu "Componentes Obrigatórios". Neste menu, o funcionário pode escolher a cor da pintura do carro.



Figure 2.24: Janela de escolha de cor de pintura

Escolher Motor

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Motor" no menu "Componentes Obrigatórios". Neste menu, o funcionário pode escolher o motor do carro.

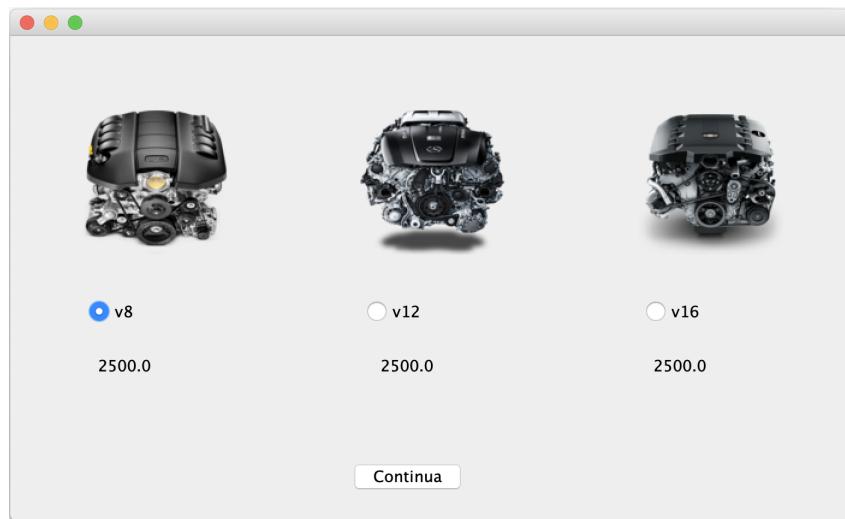


Figure 2.25: Janela de escolha de motor

Escolher Pneus

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Pneus" no menu "Componentes Obrigatórios". Neste menu, o funcionário pode escolher o tipo de pneus do carro.



Figure 2.26: Janela de escolha de pneus

Escolher Jantes

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Jantes" no menu "Componentes Obrigatórios". Neste menu, o funcionário pode

escolher o modelo de jantes do carro.



Figure 2.27: Janela de escolha de jantes

2.4.7 Escolher Detalhes

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Detalhes" no menu principal da configuração. Neste menu, o funcionário pode adicionar detalhes à sua configuração. Caso seja adicionado um componente que obrigue ou impeça a existência de outros componentes na configuração, é apresentado um aviso e é dada a possibilidade de escolha ao utilizador.

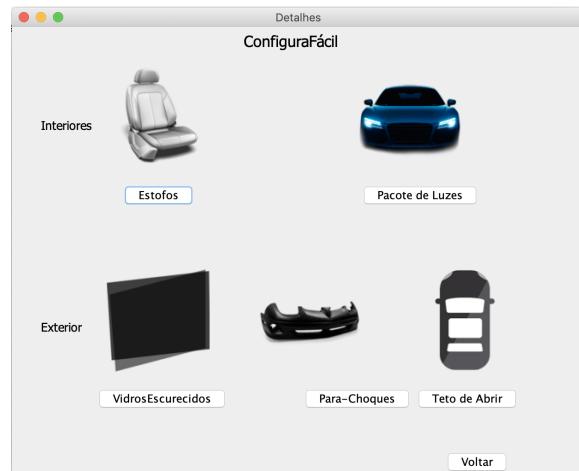


Figure 2.28: Janela de escolha de detalhes

2.4.8 Escolher componentes opcionais

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Componentes opcionais" no menu principal da configuração. Neste menu, o funcionário pode adicionar componentes opcionais à sua configuração. O funcionamento deste menu é análogo ao anterior.

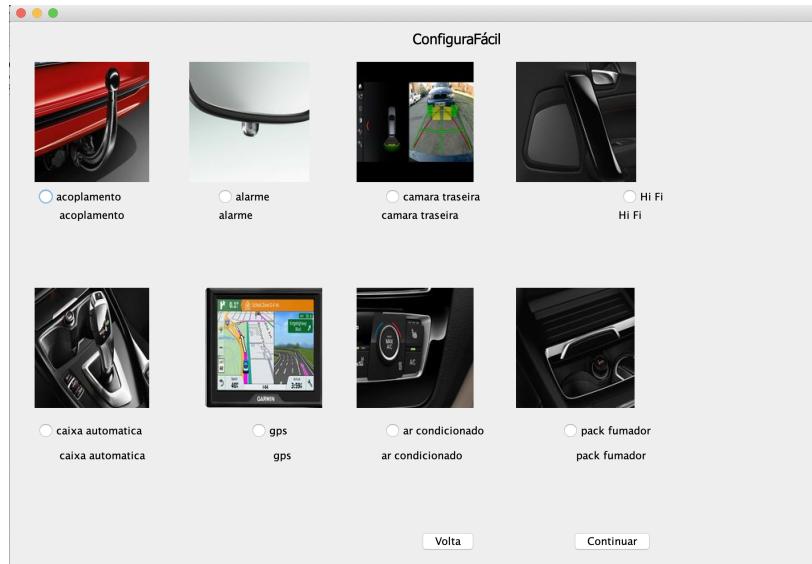


Figure 2.29: Janela de escolha de componentes opcionais

2.4.9 Escolher pacote de configuração

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Pacotes" no menu principal da configuração. Neste menu, o funcionário pode escolher um pacote de configuração para adicionar ao seu carro. Neste caso é aplicado o mesmo método de avaliação e resolução de incompatibilidades e dependências entre os componentes do que no menu apresentado acima.

2.4.10 Efetuar configuração ótima

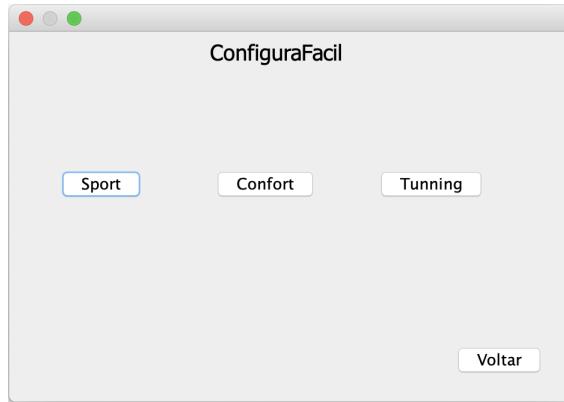


Figure 2.30: Janela de escolha de pacote de configuração

2.4.11 Remover componente

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Remover Componente" no menu principal da configuração. Neste menu, é apresentada a lista dos componentes que fazem parte da configuração do carro, e o utilizador pode remover os componentes que pretender.

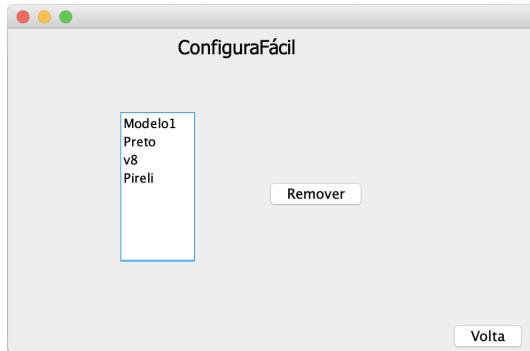


Figure 2.31: Janela de remoção de componentes

2.4.12 Enviar encomenda

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Enviar" no menu principal da configuração. Neste menu, é apresentada a lista dos componentes que fazem parte da configuração do carro, bem como campos em branco que devem ser preenchidos com as informações do cliente que pretende efetuar a configuração.

The screenshot shows a window titled "gestor" with the identifier "bmw45". Inside, there is a table with the following data:

Componente	Preco
Modelo1	2500.0
Preto	150.0
v8	2500.0
Pireli	2500.0

Below the table, the total price is displayed as "Preço 7650.0". There are three text input fields labeled "Nome", "NIF", and "Contacto" with empty boxes. At the bottom are two buttons: "Cancelar" and "Enviar".

Figure 2.32: Janela de envio de encomenda

2.4.13 Menu Fábrica

Exemplo da interface gráfica apresentada quando o gestor seleciona a opção "Fábrica" no seu menu principal inicial. Neste menu, são apresentadas as opções de verificar e gerir encomendas, funcionários e stock.

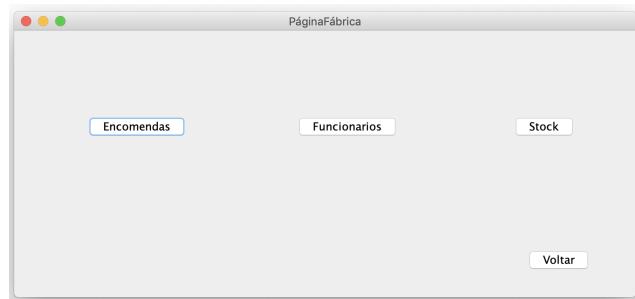


Figure 2.33: Janela de escolha de stock

2.4.14 Verificar e adicionar stock

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Stock" no menu principal da fábrica. Neste menu, é apresentada a lista dos componentes disponíveis em fábrica e o respetivo stock. Para além disso, é apresentada a lista de encomendas que podem avançar para a produção tendo em conta o stock existente e é disponibilizada a opção de validar uma encomenda, significando que a mesma é enviada para produção e deixa de estar em estado de espera.

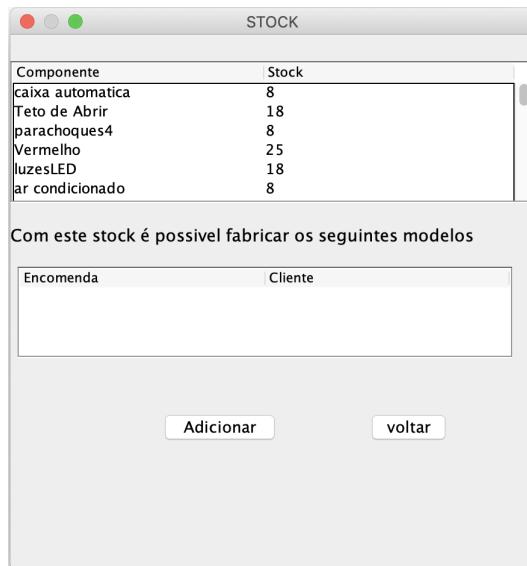


Figure 2.34: Janela de escolha de stock

2.4.15 Ver e registrar funcionários

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Funcionários" no menu principal da fábrica. Neste menu, é apresentada a lista dos funcionários registados na aplicação. Para além disso, apresenta a opção de registrar um novo funcionário na aplicação.

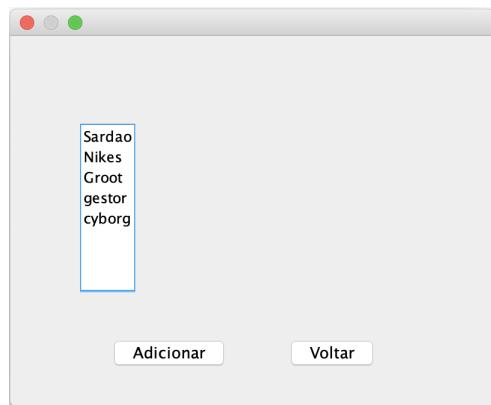


Figure 2.35: Janela de gestão de funcionários

2.4.16 Ver encomendas

Exemplo da interface gráfica apresentada quando o utilizador seleciona a opção "Encomendas" no menu principal da fábrica. Neste menu, é apresentada a lista das configurações enviadas em forma de encomenda para a fábrica.

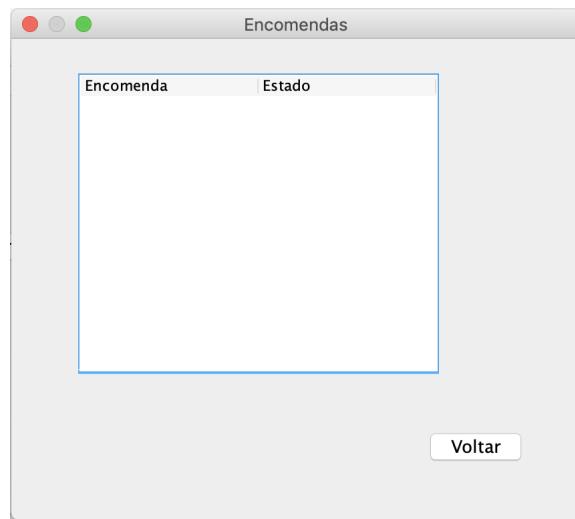


Figure 2.36: Janela gestão de encomendas

2.4.17 Máquina de Estado da navegação

A máquina de estado modela os estados possíveis que o objeto/sistema atravessa em resposta aos eventos que podem ocorrer, dando uma ideia do comportamento do sistema no geral, mais especificamente da interface. De seguida são apresentadas as máquinas de estado de alguns estados referentes à interface do nosso projeto.

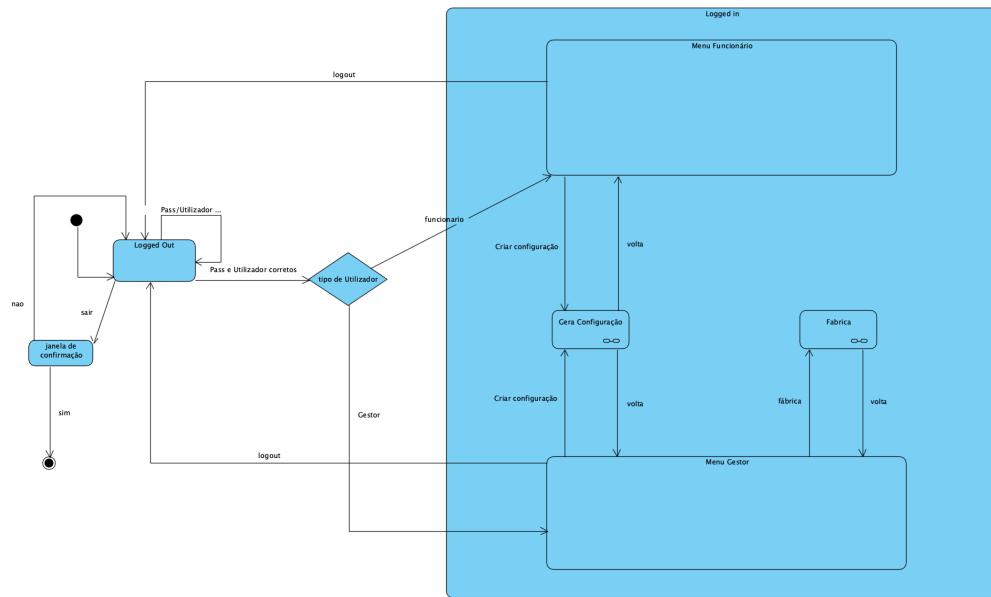


Figure 2.37: Máquina de Estados do Menu Principal

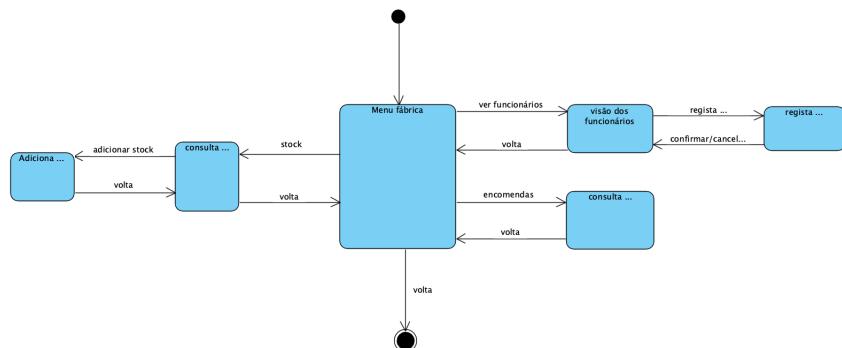


Figure 2.38: Máquina de Estados do Menu da Fábrica

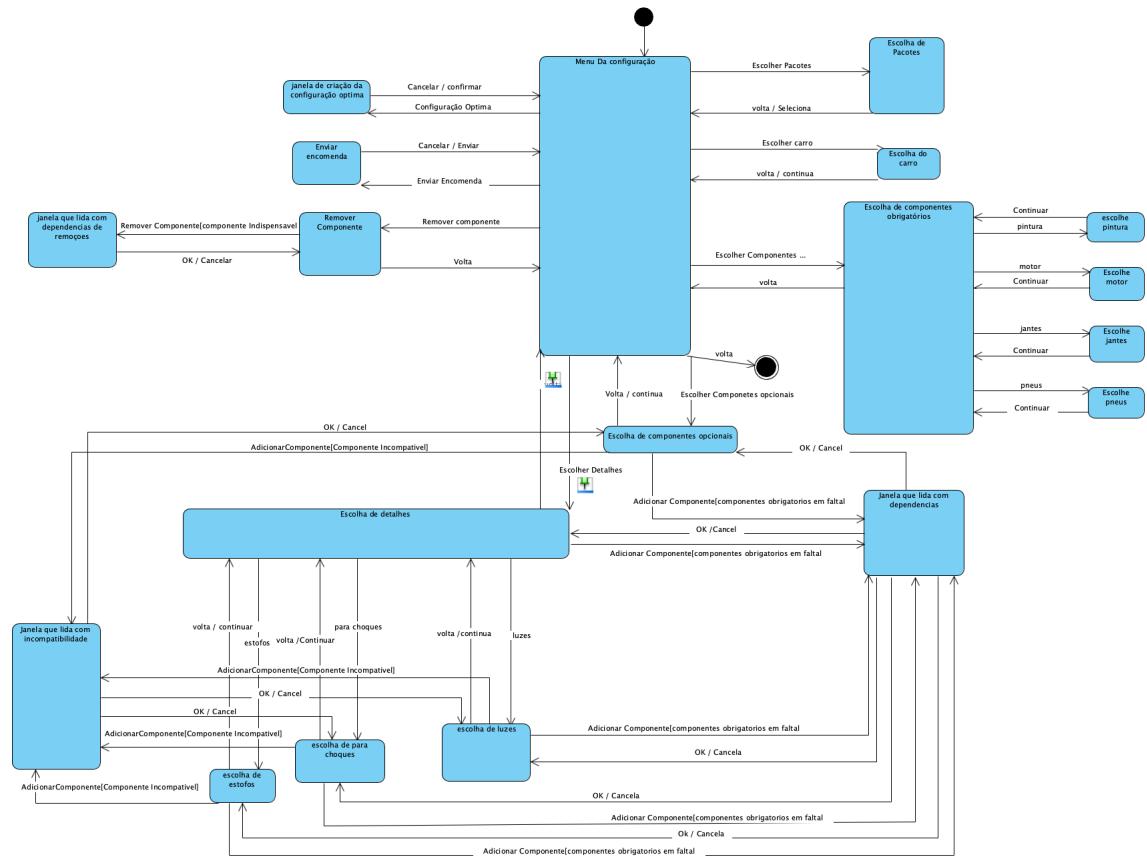
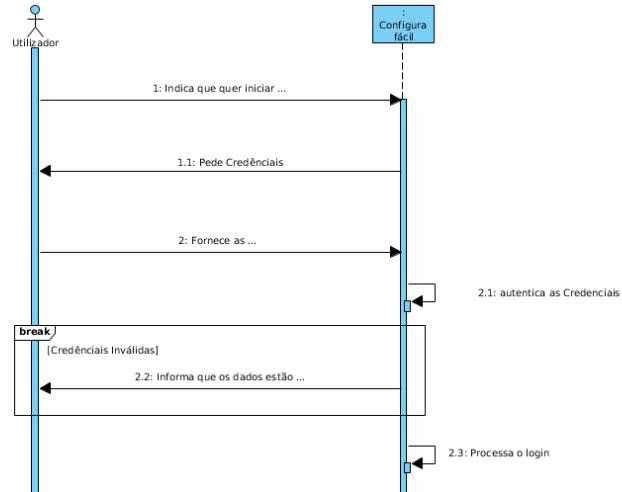


Figure 2.39: Máquina de Estados do Menu do Funcionário

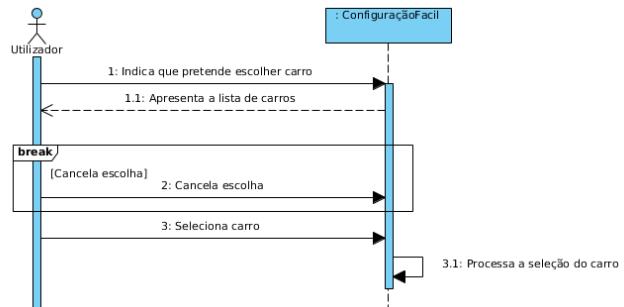
2.5 Diagrama de Sequência de Sistema

De seguida são apresentados todos os diagramas de sequência relativos a cada um dos use cases previamente estabelecidos. Estes diagramas representam os atores que interagem com o sistema, os eventos que os mesmos geram e as respostas dadas pelo sistema. De uma forma geral, estes diagramas dão uma noção geral do conjunto de operações que são mais relevantes para o bom funcionamento da nossa aplicação.

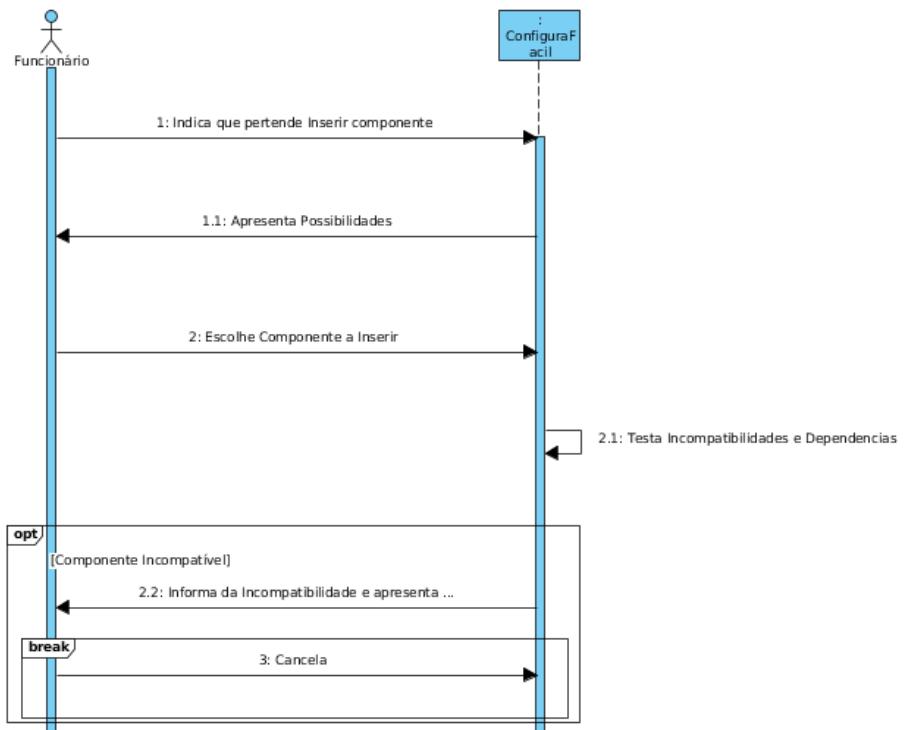
2.5.1 Login



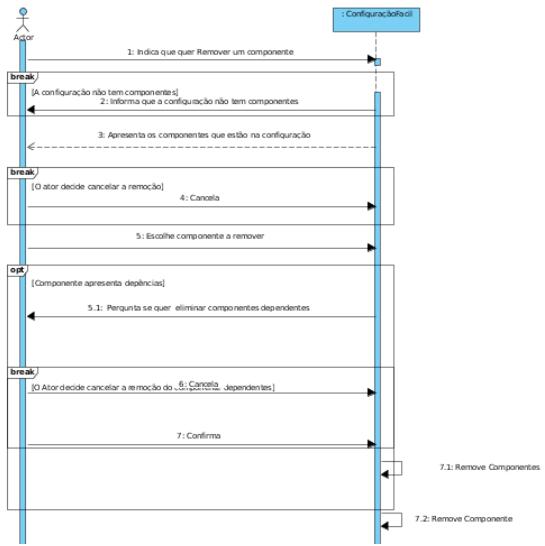
2.5.2 Escolher carro



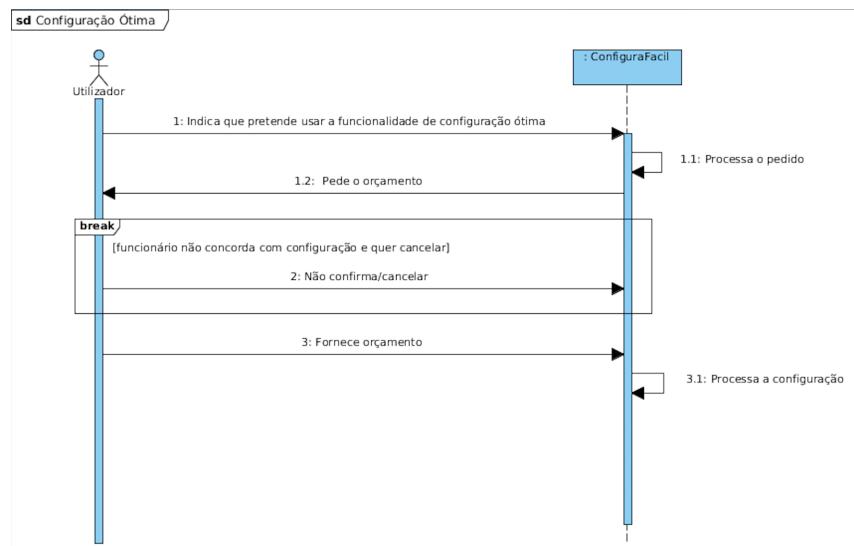
2.5.3 Inserir Componente



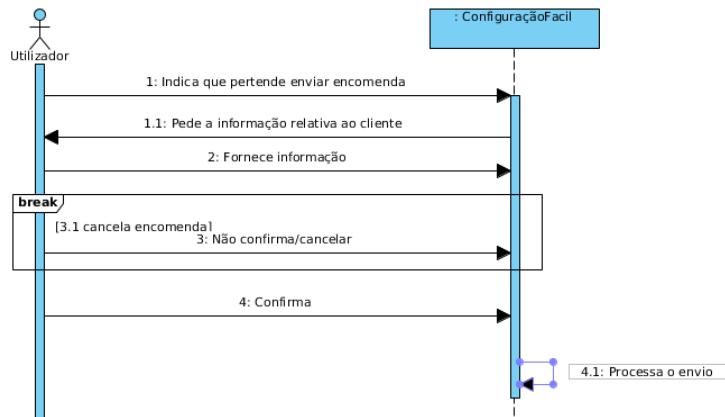
2.5.4 Remover Componente



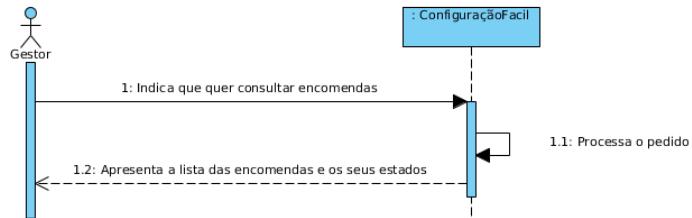
2.5.5 Configuração Ótima



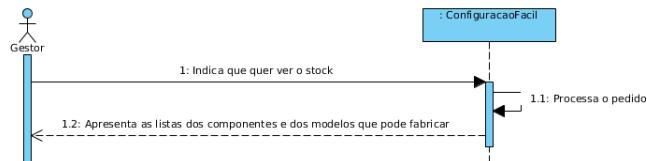
2.5.6 Encomendar Configuração



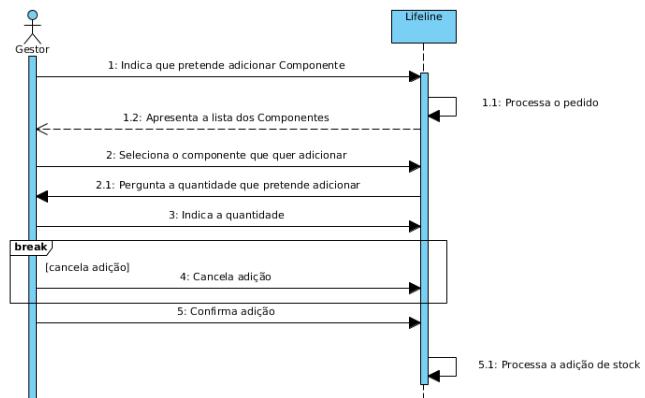
2.5.7 Consultar Encomendas



2.5.8 Verificar Stock



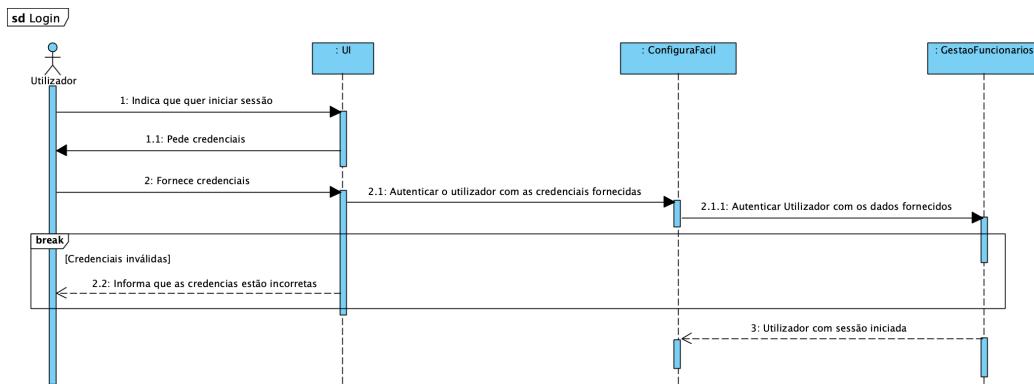
2.5.9 Adicionar Stock



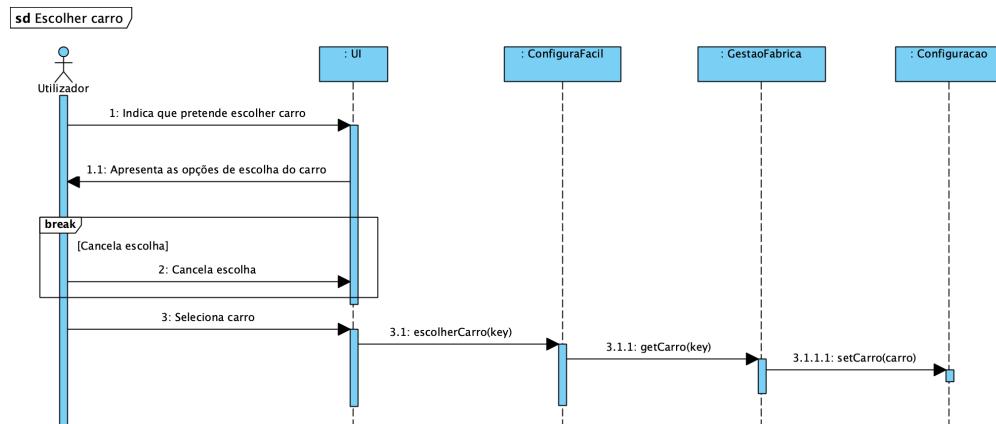
2.6 Diagrama de Sequência de Subsistemas

Nesta fase do trabalho podemos começar a identificar os subsistemas que existem na nossa aplicação, definindo para cada use case o respetivo diagrama de subsistemas. Para isto, desenvolvemos os diagramas apresentados no capítulo anterior e especificamos mais detalhadamente as interações que ocorrerão na futura implementação do código e que subsistema efetua cada ação, de forma a cumprir todos os requisitos previamente estabelecidos.

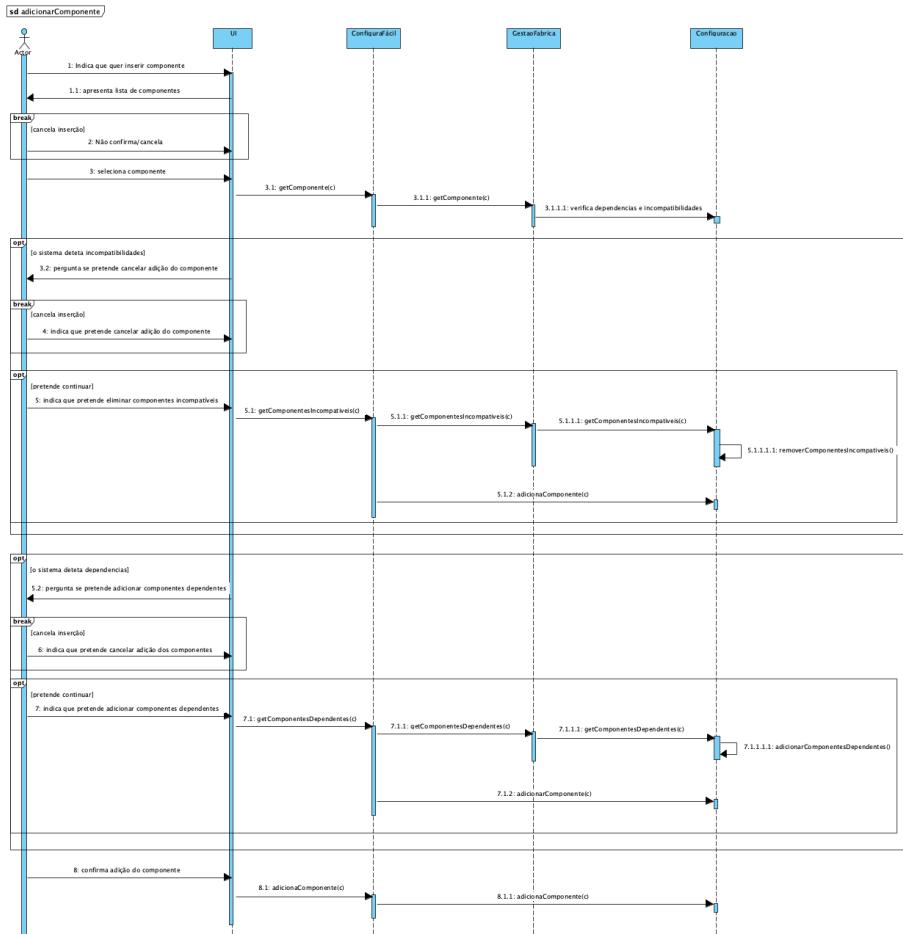
2.6.1 Login



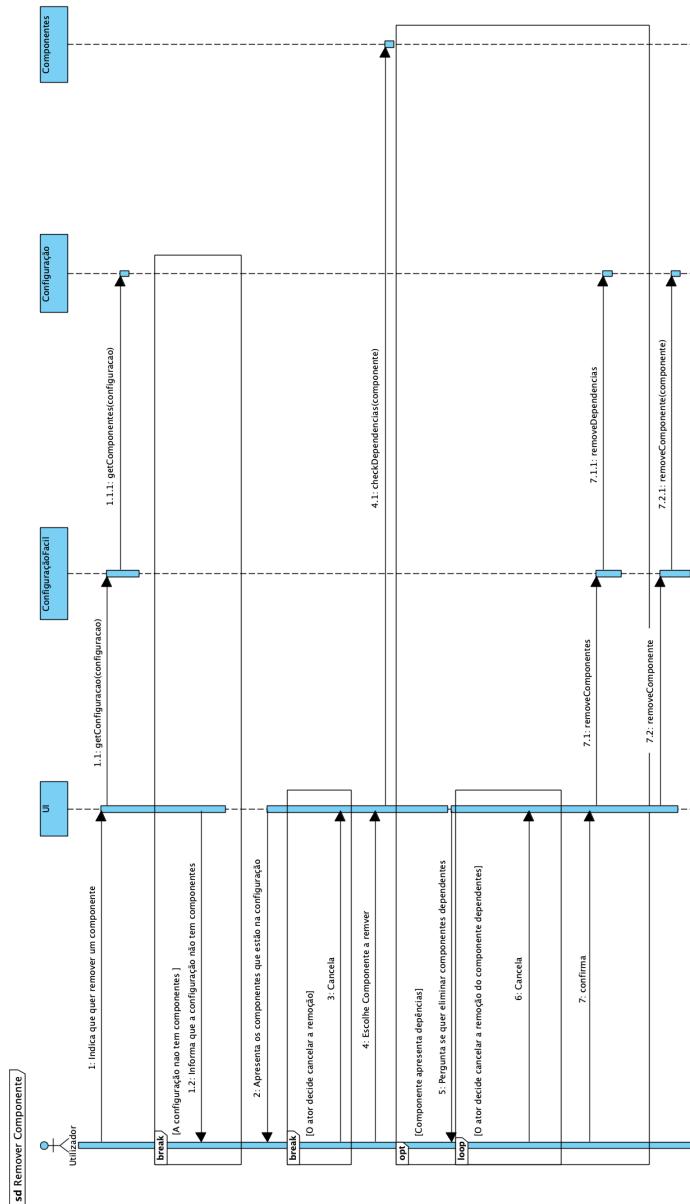
2.6.2 Escolher carro



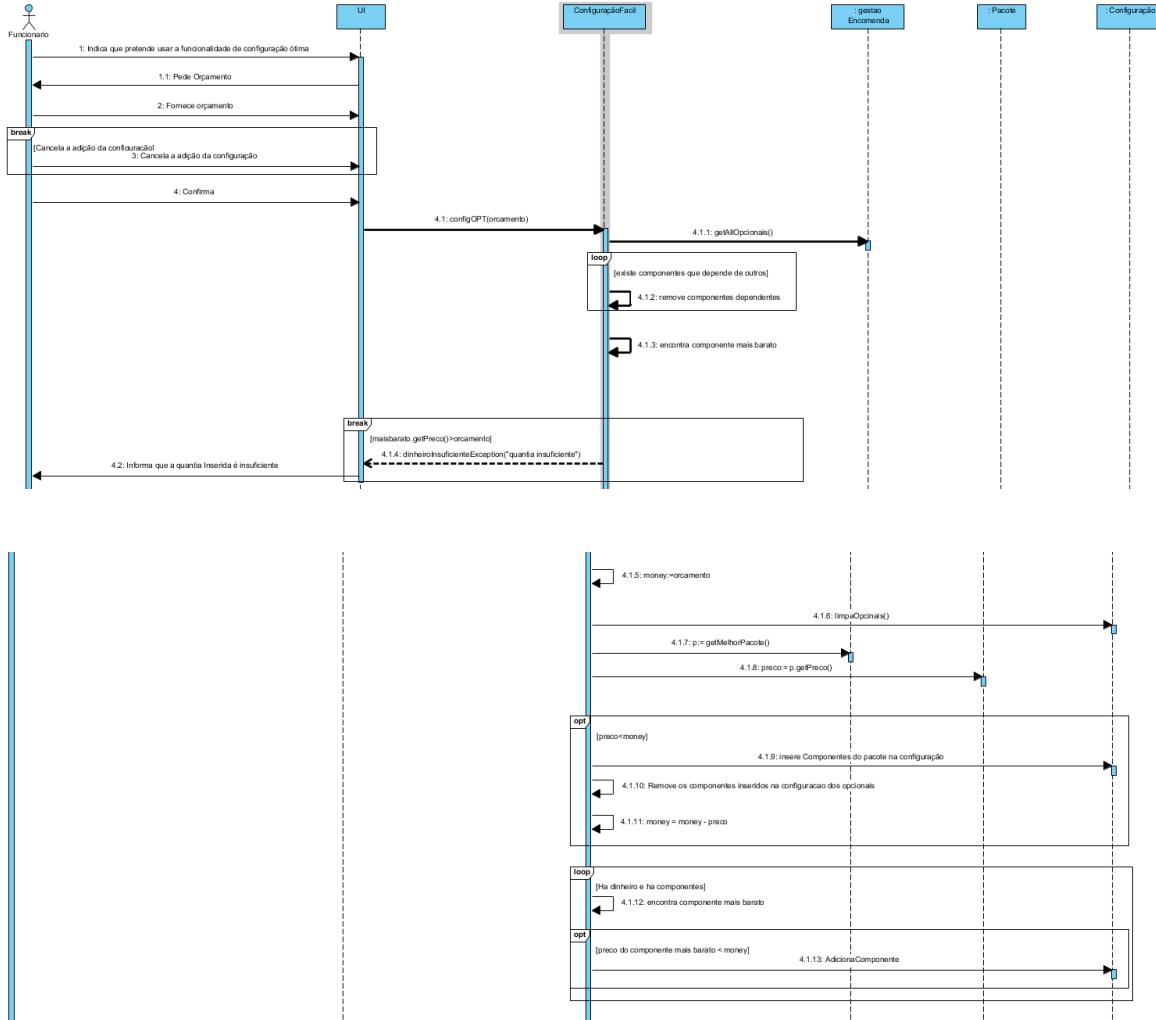
2.6.3 Inserir Componente



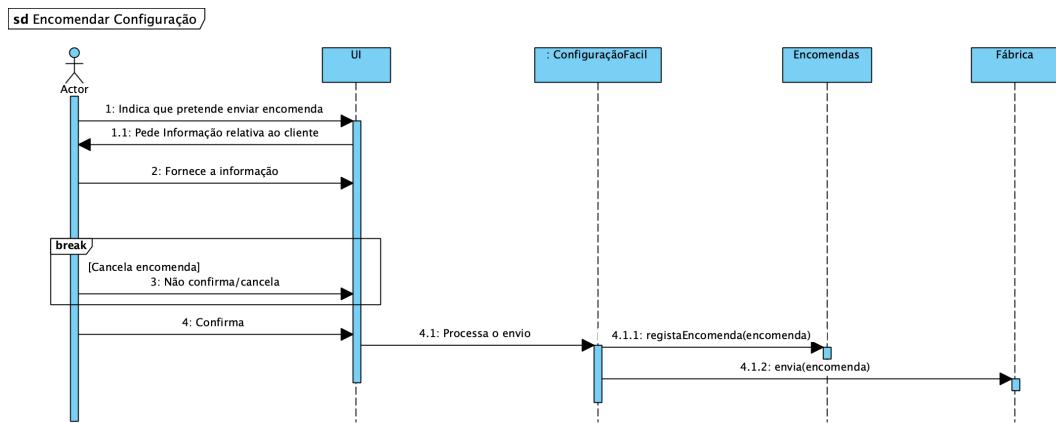
2.6.4 Remover Componente



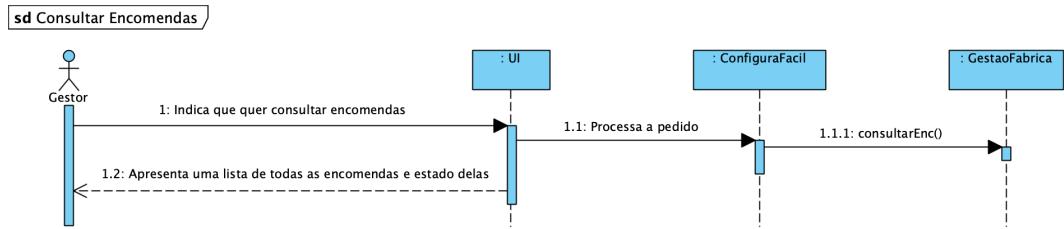
2.6.5 Configuração Ótima



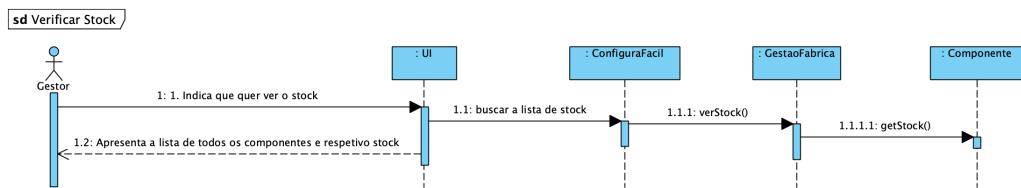
2.6.6 Encomendar Configuração



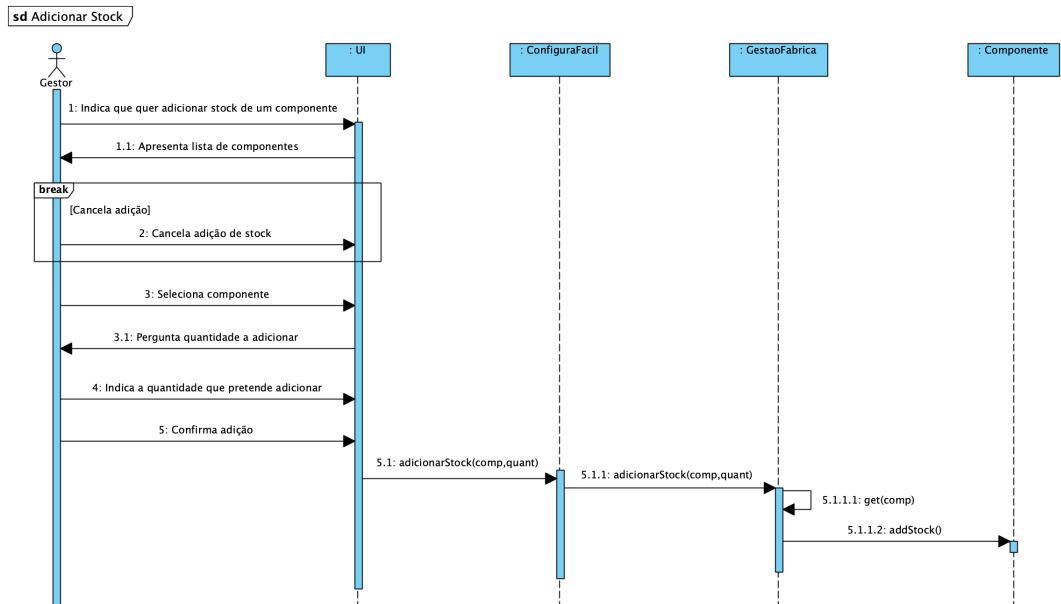
2.6.7 Consultar Encomendas



2.6.8 Verificar Stock



2.6.9 Adicionar Stock



2.7 Diagrama de Packages

Após definidas as classes do projeto, segue-se a modelação do diagrama de packages, através do qual conseguimos efetuar uma melhor organização das classes e estruturar o modelo do nosso sistema.

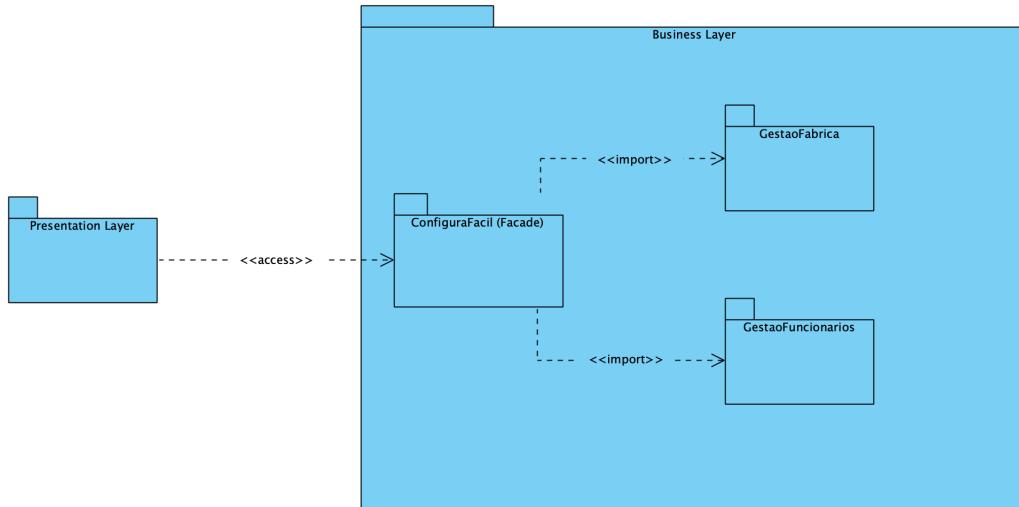


Figure 2.40

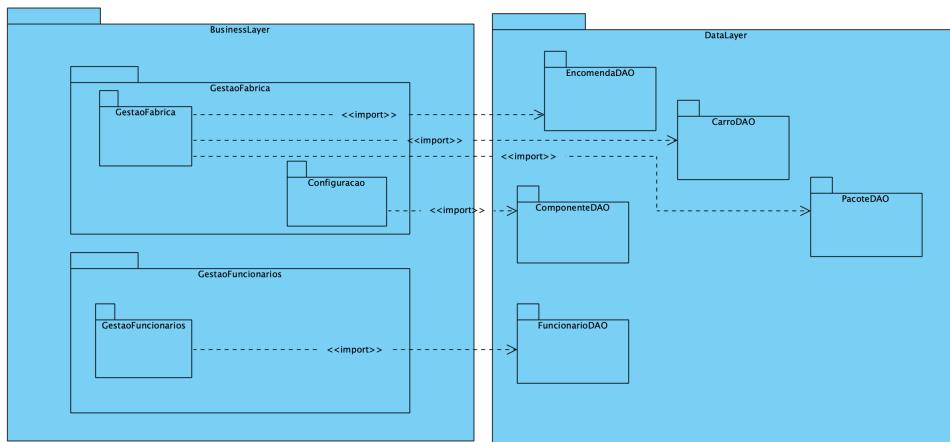


Figure 2.41

Package Presentation Layer

A camada da interface contém todas as classes que fazem a comunicação entre o utilizador e o programa (é aqui que a GUI está implementada).

Package Business Layer

Na lógica de negócio estão incluídos três packages mais pequenos, por uma questão de organização do código, facilitando a identificação das operações que são efetuadas no package maior.

O package ConfiguraFacil efetua a ligação entre a interface e a camada lógica.

Na GestaoFabrica é feito o tratamento e armazenamento dos dados referentes às configurações/encomendas feitas na aplicação.

Na GestaoFuncionarios são tratados e armazenados os dados relativos aos funcionários que utilizam a aplicação.

Package Data Layer

Neste package são implementadas todas as operações que estabelecem a conexão entre a lógica de negócios e a base de dados que suporta a aplicação. Ou seja, neste package temos implementados todos os DAO's.

2.8 Diagrama de Classes

A organização do código em classes tem dois objetivos fundamentais: primeiro, facilita a reutilização, através da reutilização de classes previamente desenvolvidas em novos sistemas; em segundo e último lugar, facilita a manutenção — o sistema deverá ser desenvolvido de forma a que a alteração de uma classe tenha o menor impacto possível no resto do sistema.

2.8.1 Diagrama de Classes

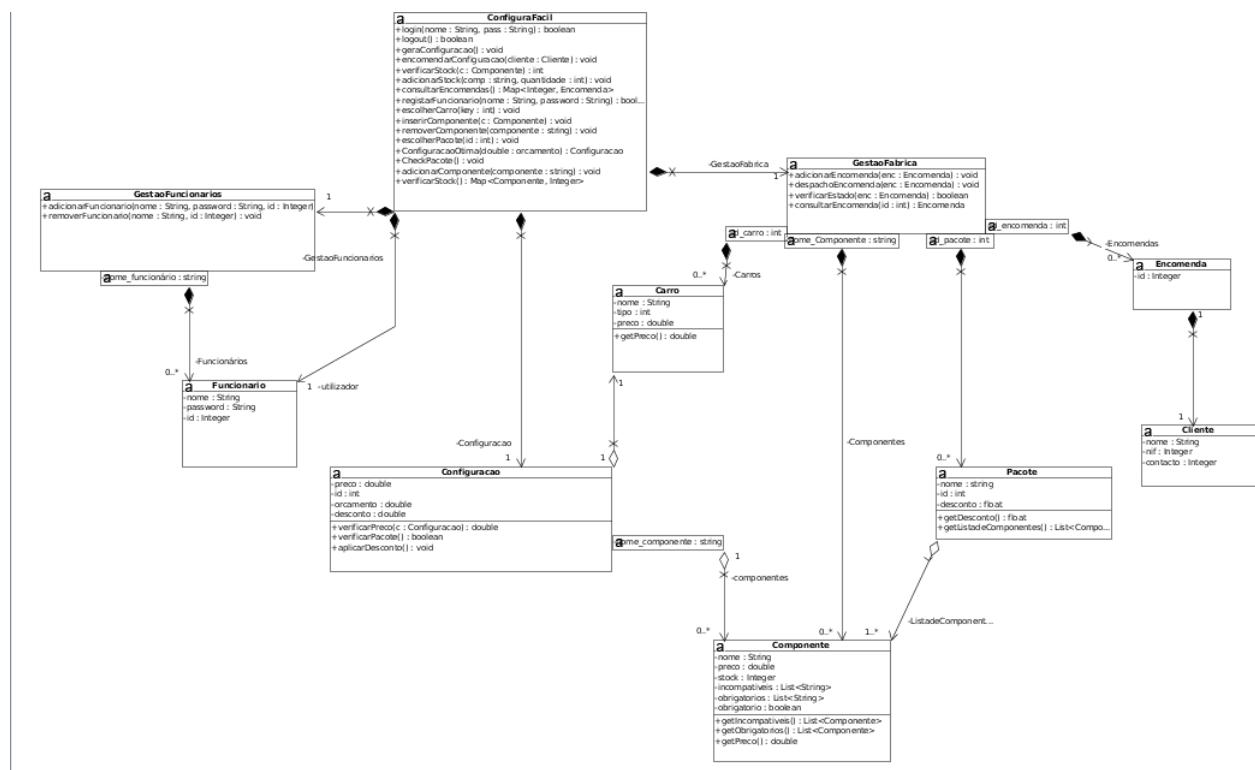


Figure 2.42: Diagrama de Classes

2.9 Diagrama de ORM

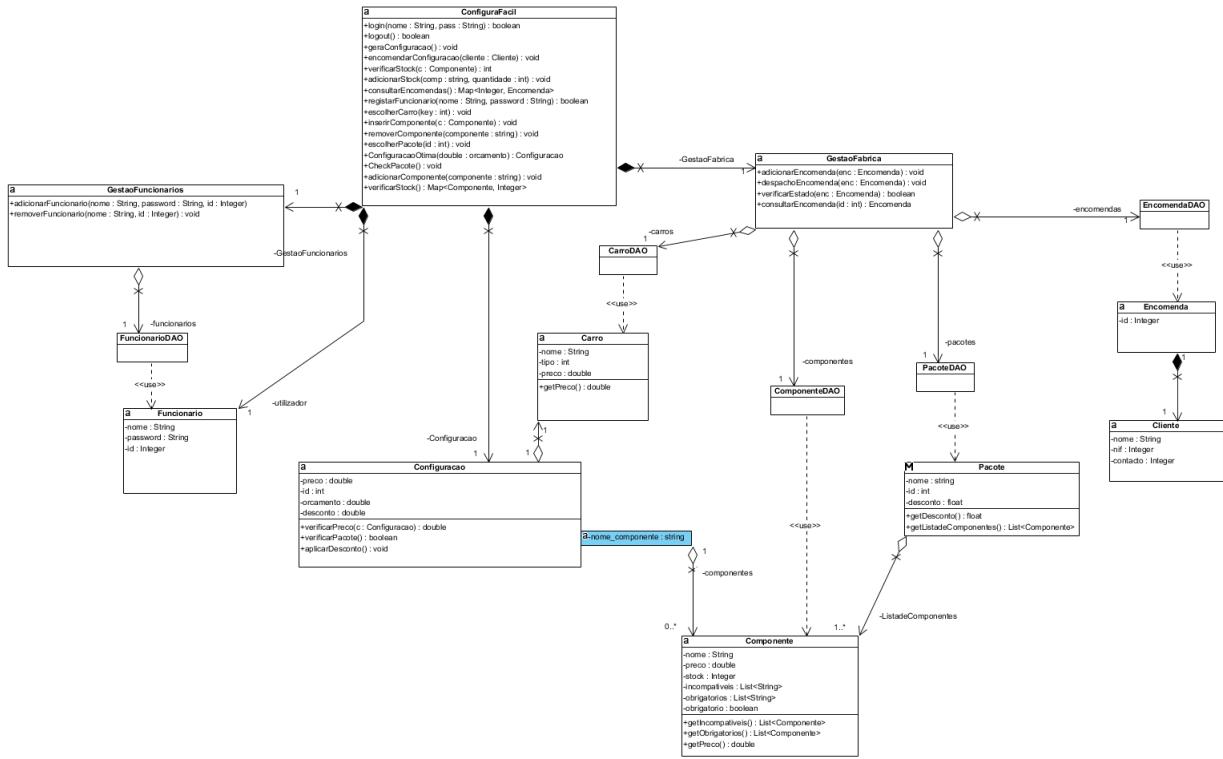


Figure 2.43: Diagrama de ORM

2.10 Diagrama de Sequência de Implementação

Nesta fase do projeto já nos encontramos muito próximos da implementação da aplicação. Assim vamos pegar em alguns use cases que achamos mais relevantes para a funcionalidade da mesma e mostrar mais detalhadamente o nível de implementação do código.

2.10.1 Login

Para a nossa aplicação poder ser usada o primeiro passo de todos será o funcionário fazer o login para poder ter acesso às respetivas funcionalidades na aplicação. Esse login está implementado no método;

```
public void login(String username, String password)
```

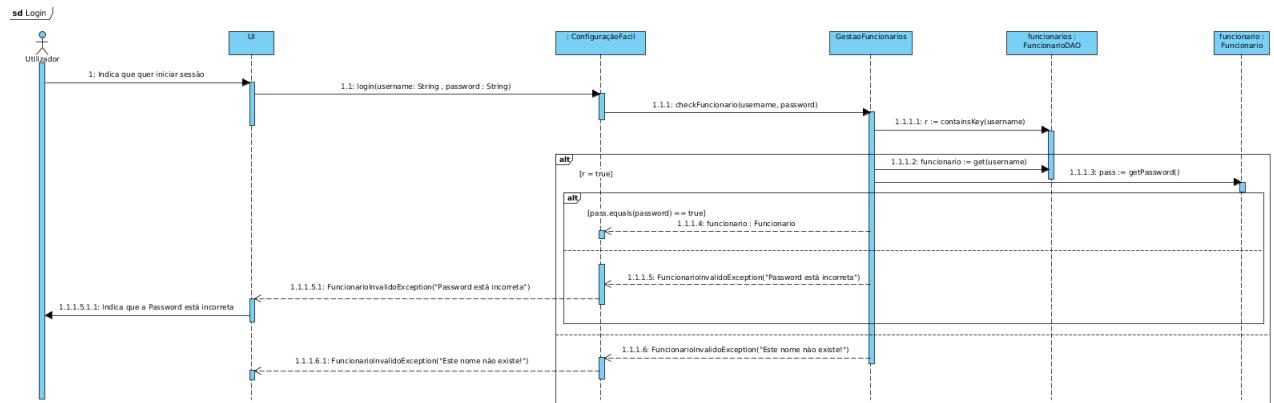


Figure 2.44: Diagrama de Implementação do Login

2.10.2 Registar Funcionário

Estando o gestor logado é possível ele registar funcionários na aplicação. Para este registo ele apenas tem de fornecer o nome e a password para o método

```
public void registrarFuncionario(String username, String password)
```

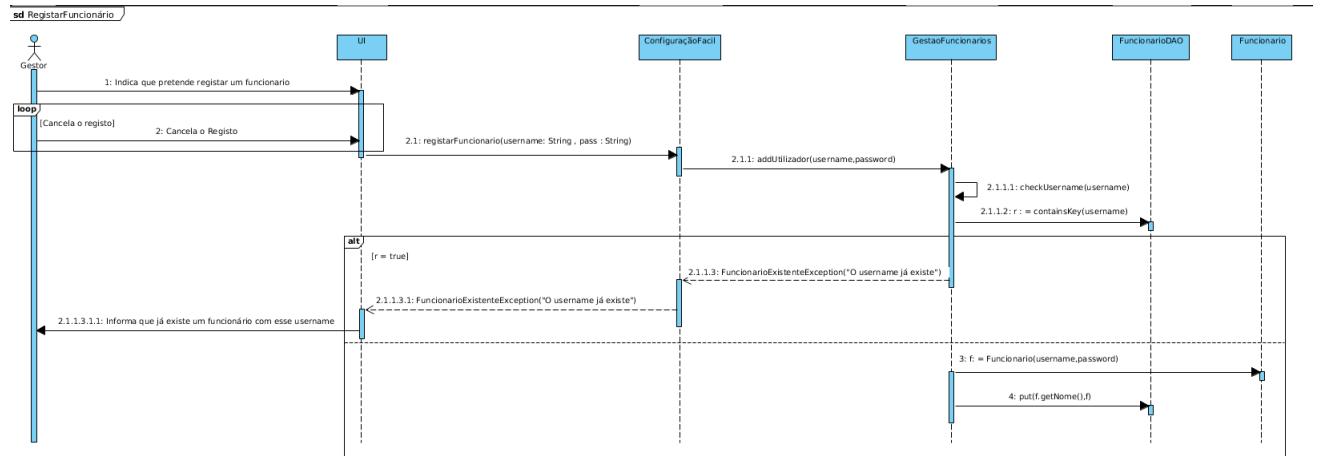


Figure 2.45: Diagrama de Implementação do Registar Funcionário

2.10.3 Escolher Carro

Sendo a nossa aplicação usada para fazer uma configuração de um carro ao gosto do cliente, torna-se lógico que a primeira seleção do cliente seja de facto o carro. Assim podemos escolher o carro através do método

```
public void EscolherCarro(int key)
```

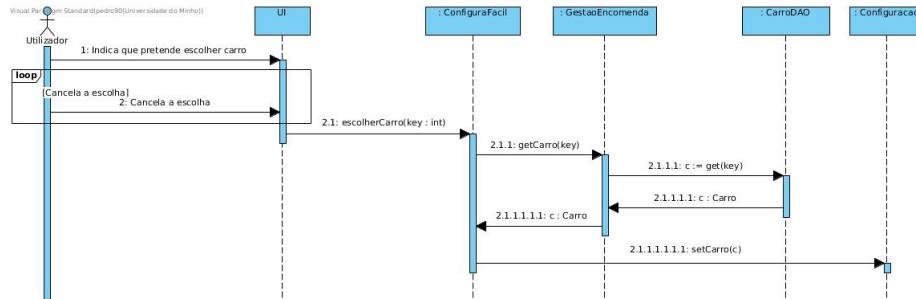


Figure 2.46: Diagrama de Implementação do Escolher Carro

2.10.4 Inserir Componente

Inserir Componente é um dos momentos cruciais e uma das principais funcionalidades da nossa aplicação. Com o adicionar componente, permite ao funcionário inserir uma componente na configuração desejada pelo cliente e assim proceder à encomenda. Este ação encontra-se implementada pela junção de vários métodos, sendo alguns destes:

public List<String> getComponenteIncompativeis(String comp)

public List<String> CompObrigatorios(String comp)

adicionarComponente(String comp)

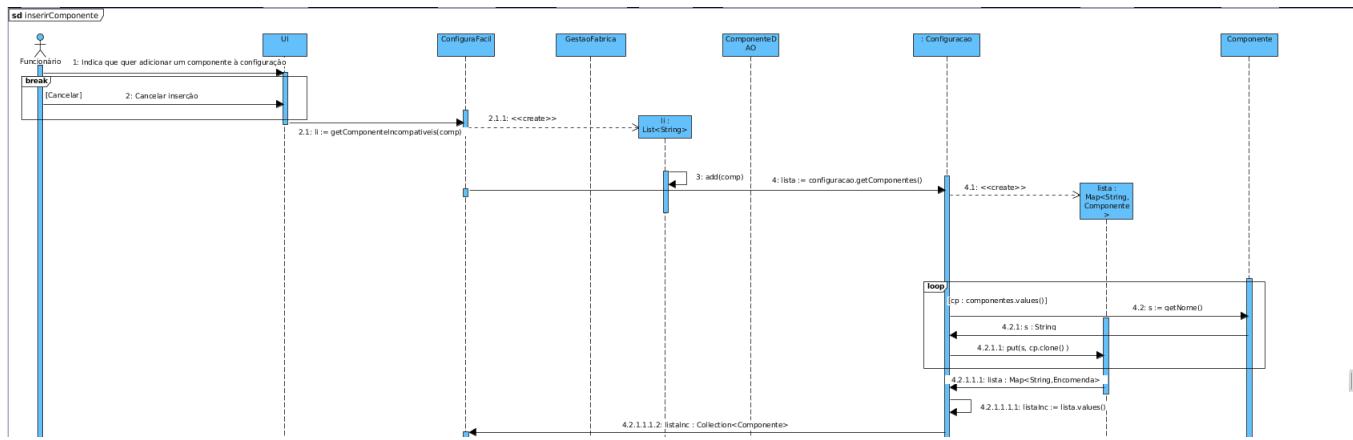


Figure 2.47: Diagrama de Implementação de InserirComponente-1

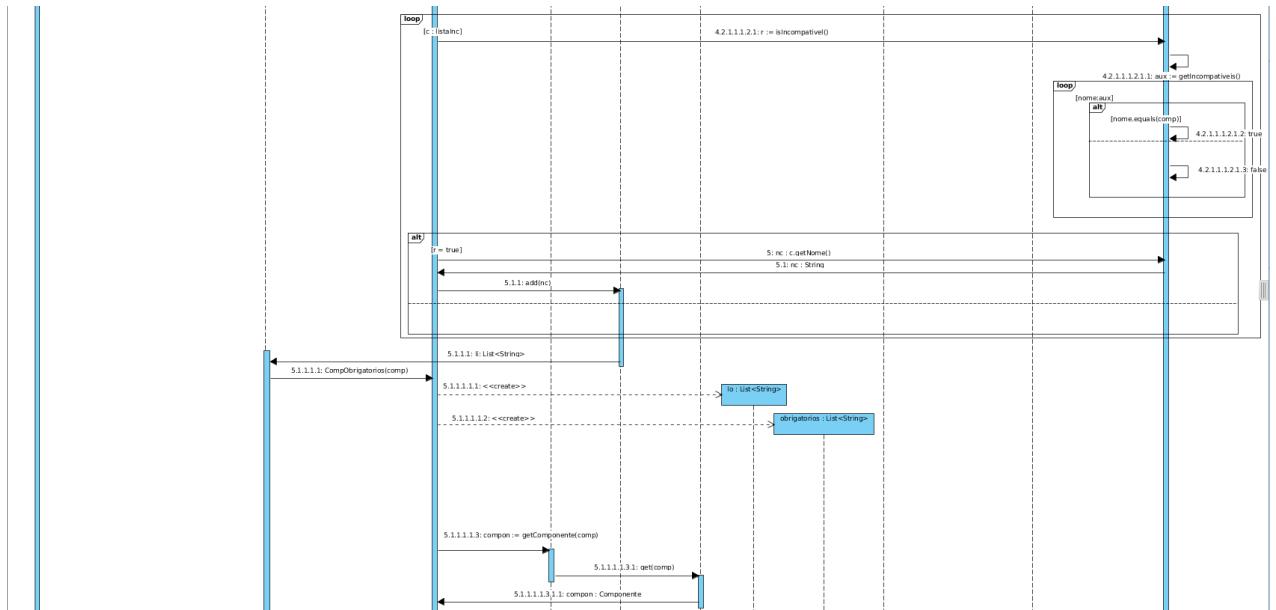


Figure 2.48: Diagrama de Implementação de InserirComponente-2

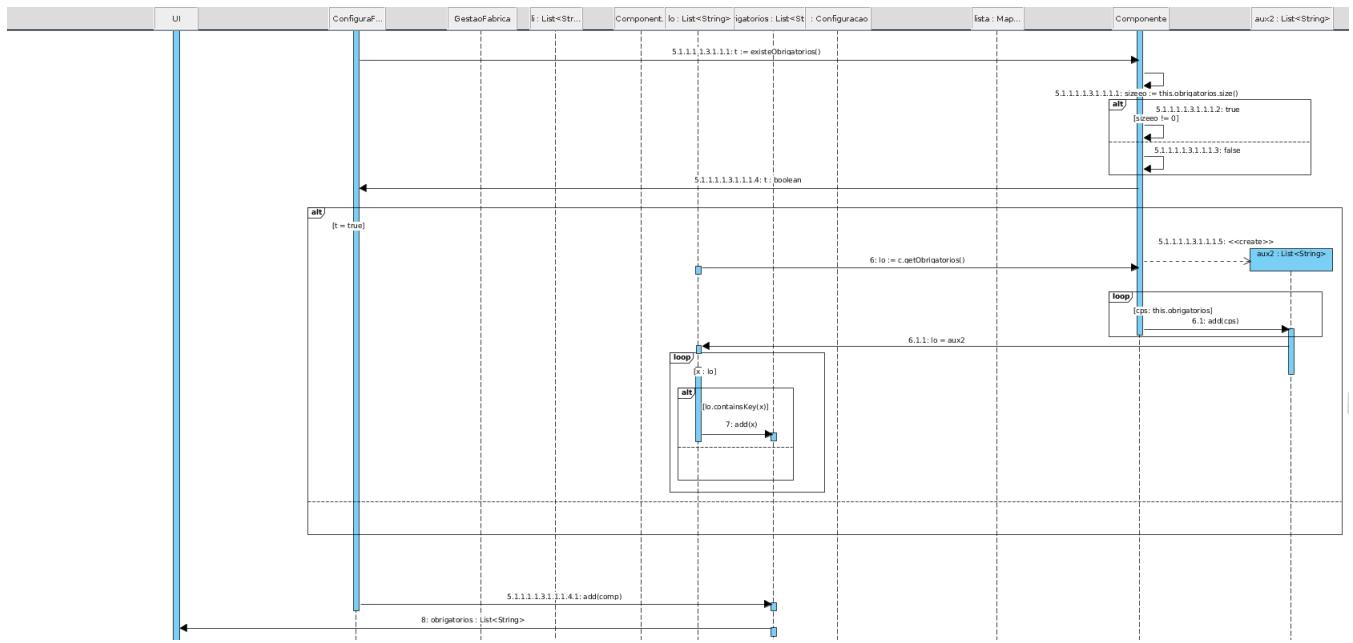


Figure 2.49: Diagrama de Implementação de InserirComponente-3

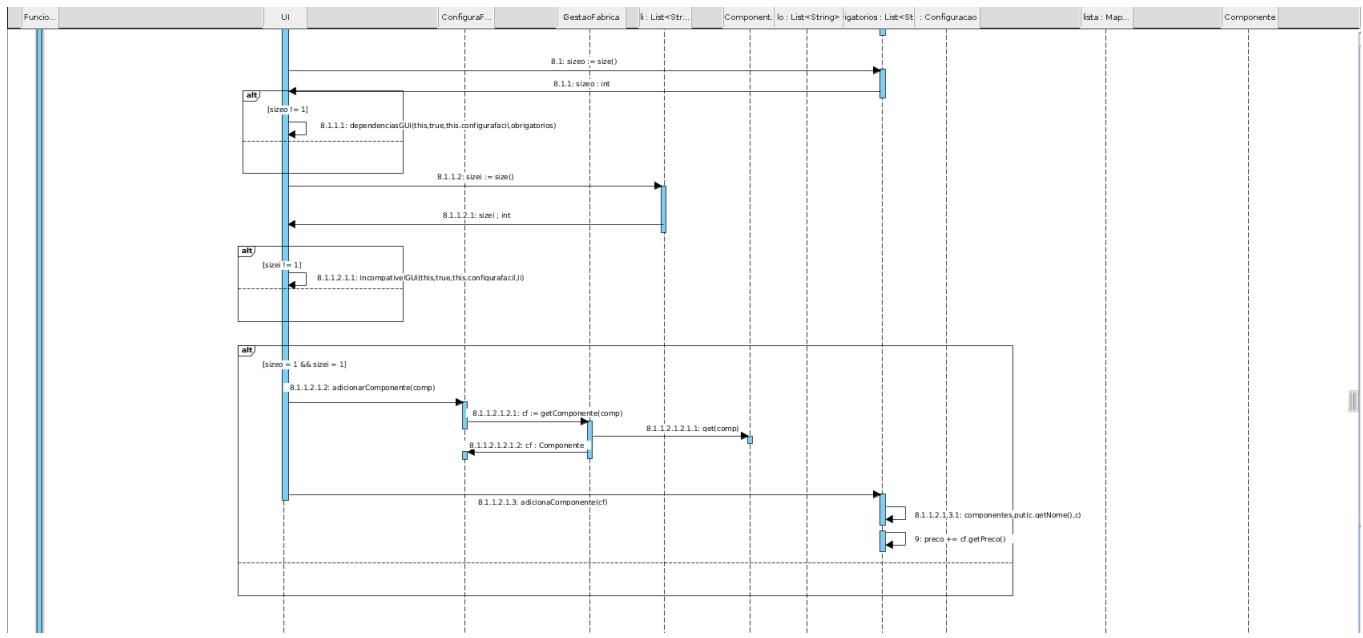


Figure 2.50: Diagrama de Implementação de InserirComponente-4

2.10.5 Escolher Pacote

Ao cliente também é possível escolher uma configuração por Pacote. Para isto o funcionário informa o mesmo à aplicação e é atribuído uma configuração pelo pacote escolhido através do método

```
public void EscolherPacote(int key)
```

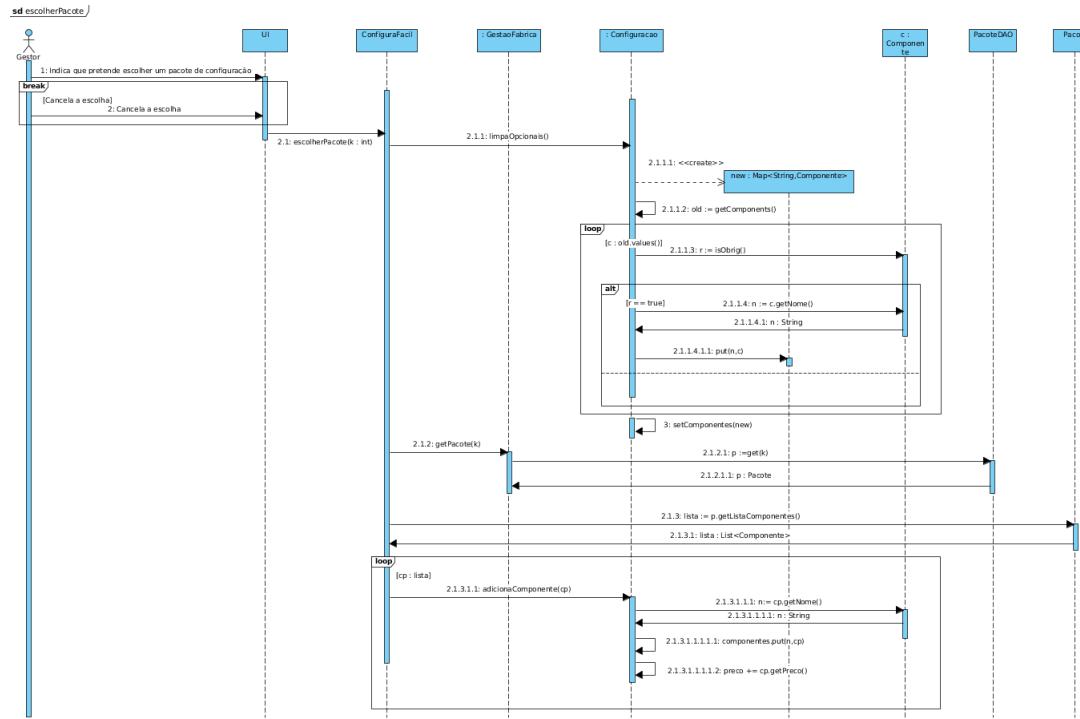


Figure 2.51: Diagrama de Implementação do Escolher Pacote

2.10.6 Configuração Ótima

É possível ao cliente optar por uma configuração ótima, consoante o orçamento que ele esteja disposto a gastar. Para isto o funcionário informa o mesmo à aplicação e é atribuído uma configuração ótima através do método

```
public void configOPT(double orçamento) throws dinheiroInsuficienteException
```

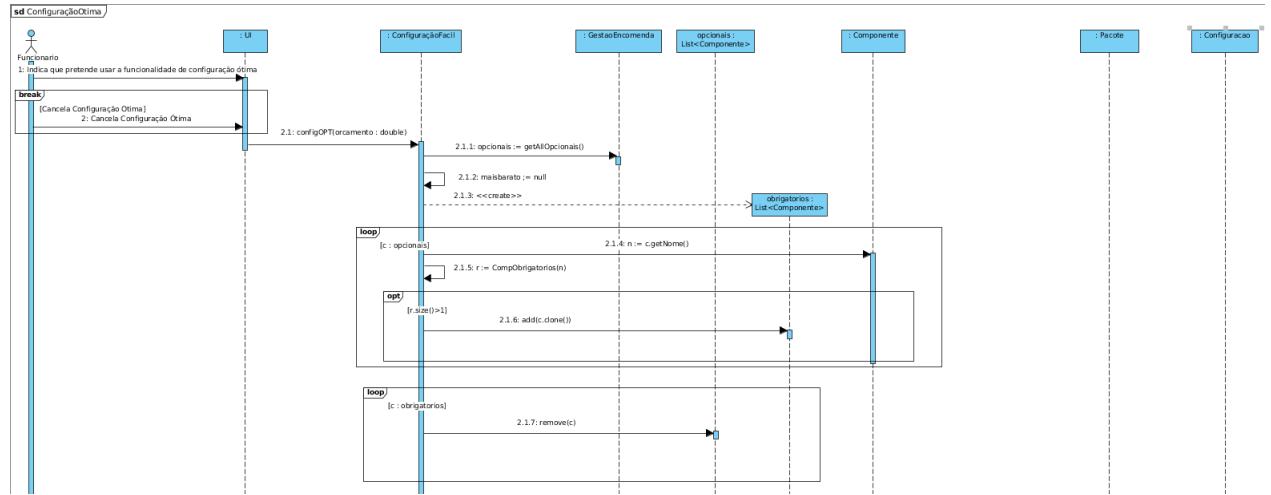


Figure 2.52: Diagrama de Implementação do Configuração Ótima 1

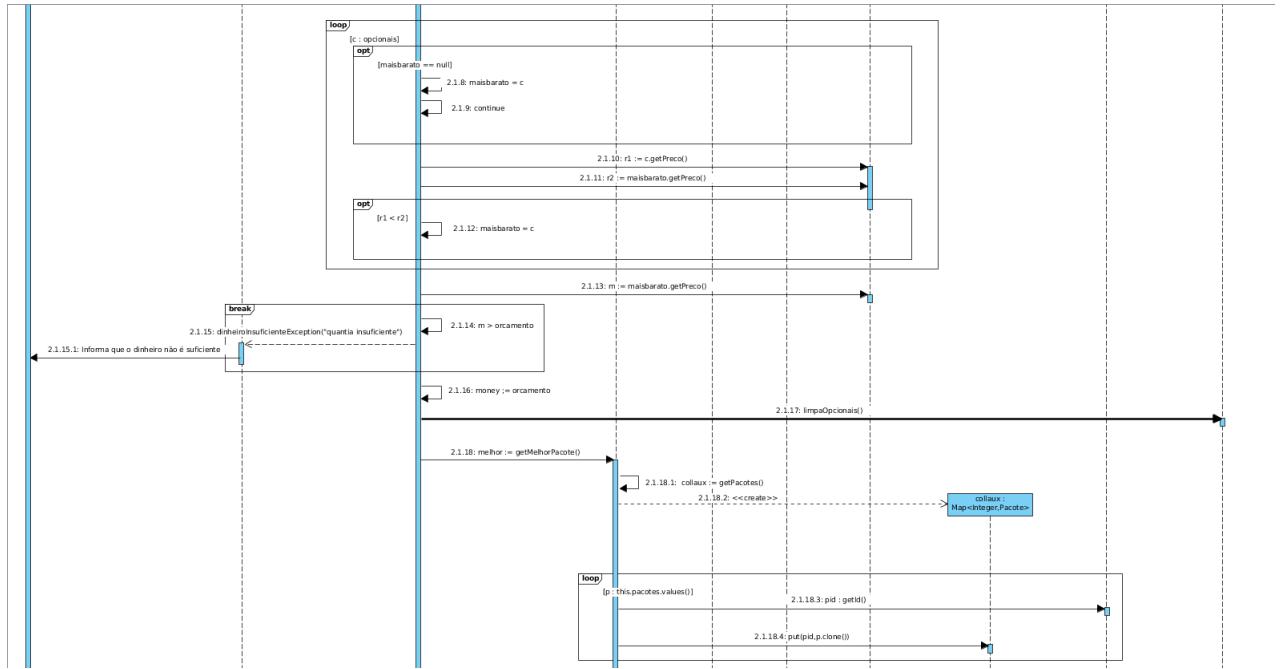


Figure 2.53: Diagrama de Implementação do Configuração Ótima 2

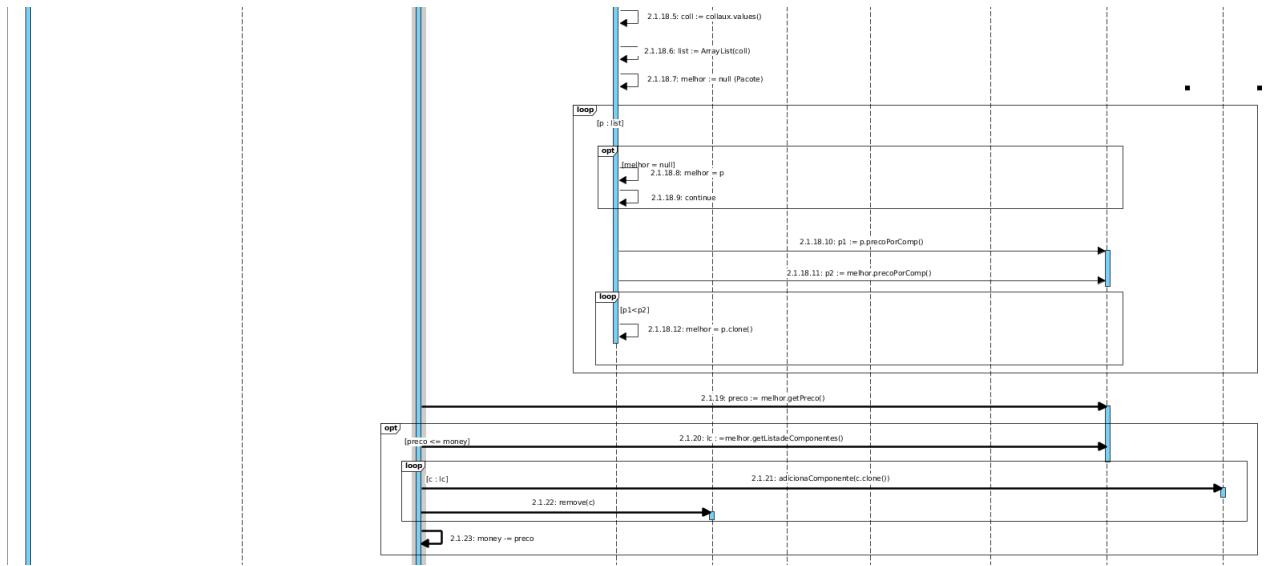


Figure 2.54: Diagrama de Implementação do Configuração Ótima 3

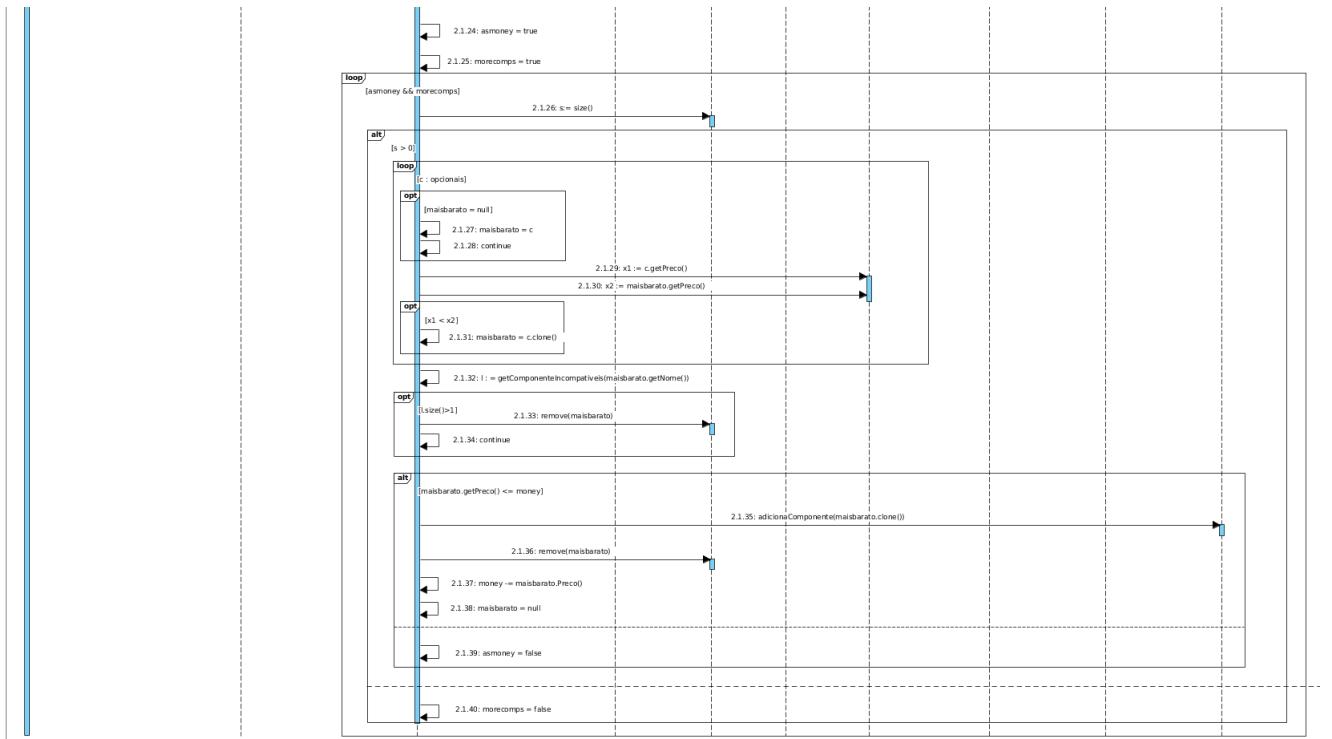


Figure 2.55: Diagrama de Implementação do Configuração Ótima 4

2.10.7 Encomendar Configuração

Depois de ser feita toda a configuração, ao gosto do cliente, é necessário enviar para a fábrica os dados da encomenda a fazer. Para isso implementamos o método

```
public void EncomendaConfiguracao(String nome, int nif, int contacto)
```

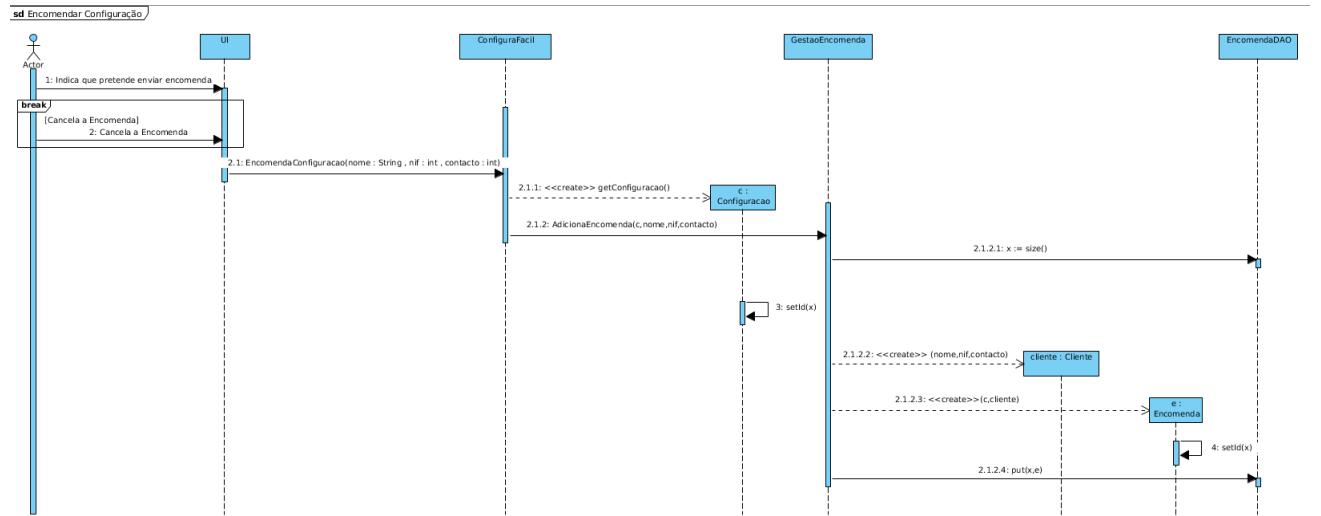


Figure 2.56: Diagrama de Implementação do Encomendar Configuração

2.10.8 Consultar Encomendas

Sendo a nossa aplicação uma aplicação que interliga a empresa em questão à fábrica, também é possível ao gestor fazer a consulta das encomendas e o estado das mesmas. Esta funcionalidade foi implementada através do método

```
public Map<Integer, String> consultarEncomendas()
```

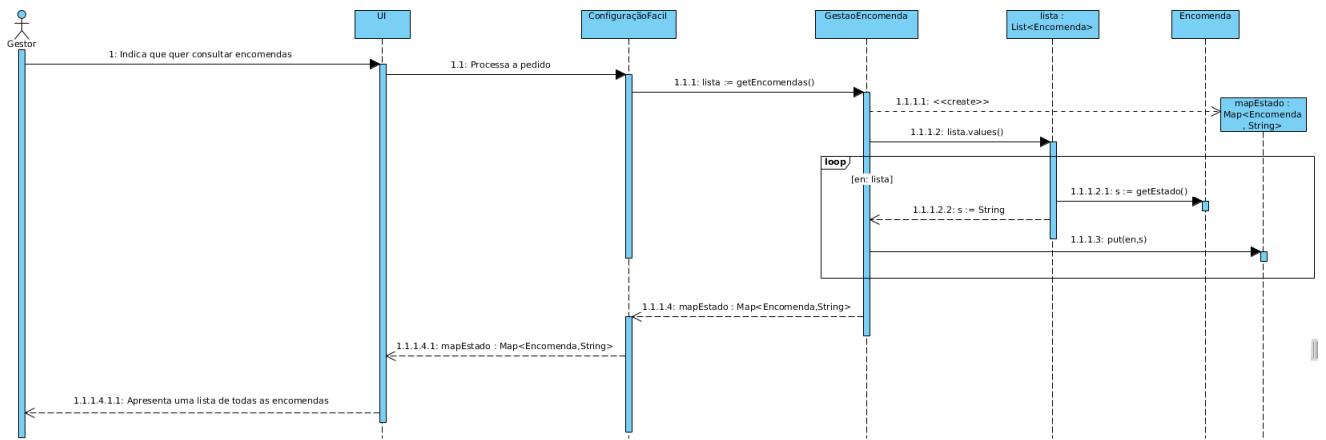


Figure 2.57: Diagrama de Implementação do Consultar Encomendas

2.10.9 Verificar Stock

Outra funcionalidade da nossa aplicação é a possibilidade de verificar stock. Esta verificação foi implementada no método:

```
public Map<String, Integer> verificarStock()
```

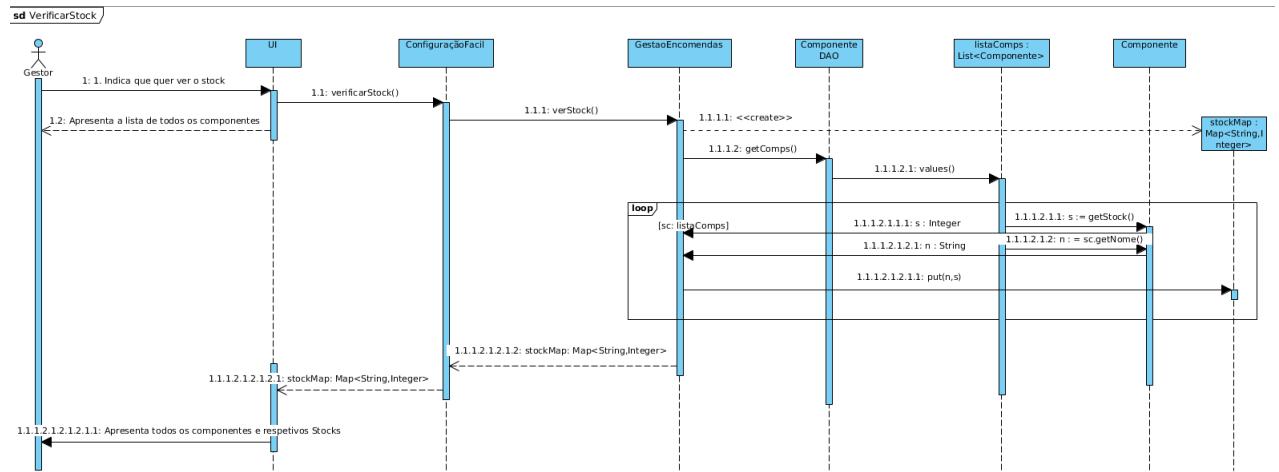


Figure 2.58: Diagrama de Implementação do Verificar Stock

2.10.10 Adicionar Stock

Outra funcionalidade da nossa aplicação é poder adicionar stock de um determinado componente. Para isso foi implementado o método

```
public void adicionarStock(String comp, int inc)
```

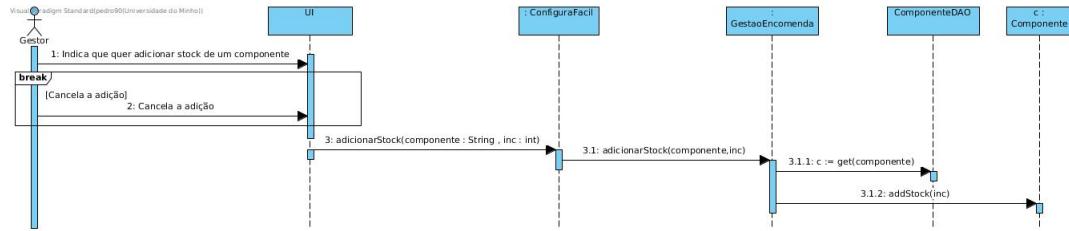


Figure 2.59: Diagrama de Implementação do Adicionar Stock

2.11 Diagrama de Instalação

O UML tem como principal objetivo a estruturação e comportamento do software de um sistema. No entanto, o diagrama de instalação, que descreve o sistema do ponto de vista dos componentes físicos que este abrange, modelando a sua topologia de hardware, aborda a parte física da aplicação a ser desenvolvida, demonstrando os componentes de hardware nos quais o software se encontra implementado.

O diagrama de instalação apresentado explicita que a aplicação ConfiguraFácil é executada no computador de um utilizador e, dentro desta, o software da aplicação (implementado em código java) procede ao envio de instruções SQL para uma base de dados relacional (implementada em MySQL), onde se encontram informações relativas aos vários componentes, modelos de carros, encomendas, funcionários, etc.

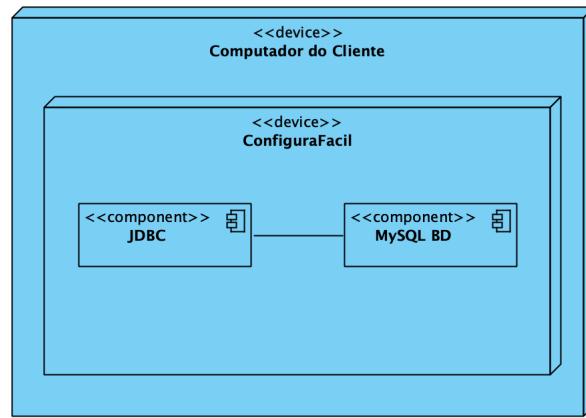


Figure 2.60: Diagrama de Instalação

2.12 Base de Dados implementada

Para guardar o estado da aplicação mesmo depois de a fechar, implementamos uma base de dados que armazena todos os processos realizados na aplicação. O modelo lógico da base de dados apresenta-se de seguida.

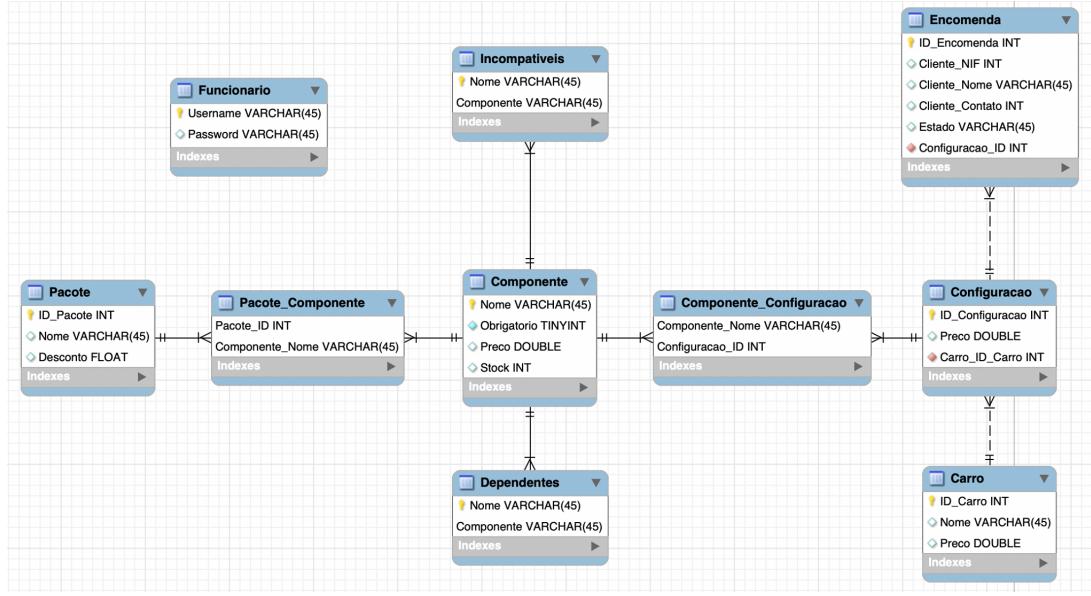


Figure 2.61: Modelo lógico da base de dados

Chapter 3

Conclusão

Na primeira fase do trabalho foi efetuada uma análise de requisitos, onde numa primeira fase analisámos tudo sobre o problema proposto para termos uma melhor noção de como iria funcionar o sistema/aplicação em si, e de seguida construímos o Modelo de Domínio. Após termos este modelo definido e consolidado, abordamos os requisitos funcionais do problema, tendo sido desenvolvido o Diagrama de Use Case, no qual identificámos os atores que vão interagir com o nosso sistema, e também aquilo que pretendemos que cada um destes faça. Depois de termos todos os Use Case, especificámos cada um deles de forma a percebermos os diferentes tipos de fluxos de eventos que existirão com a interação do nosso sistema com o utilizador.

Já no final da primeira fase, tendo por base o Diagrama de Use Case fizemos as mockups da interface da nossa aplicação e o respetivo mapa de navegação entre os diferentes ecrã. Utilizamos uma Máquina de Estados para representar o mapa de navegação.

Nesta segunda fase do trabalho e dando seguimento ao que foi realizado na primeira fase, continuamos com as etapas pela ordem que foram apresentadas na UC, da linguagem UML que nos permitiram chegar ao objetivo pretendido, ou seja, permitiram implementar um sistema/aplicação capaz de fazer configurações de carros. Esta fase começou com o desenvolvimento dos diagramas de sequência de sistemas para os Use Case criados na primeira parte, o que nos permitiu representar as interações entre o ator e o sistema através das mensagens trocadas entre eles. Depois de analisar os Diagrama de Sequência desenvolvidos no passo anterior, identificamos os nossos Subsistemas a partir dos Diagramas de Sequência de Subsistemas onde mostrámos as possíveis interações na aplicação o que nos deu uma visão concreta sobre o que "estava acontecer no sistema quando ocorria uma interação com ele".

Após termos todos os subsistemas identificados prosseguimos com o diagrama de Packages , no qual distribuímos os subsistemas em 3 partes lógicas: camada da

interface(Presentation layer), camada da lógica do negócio (Business layer) e a camada das operações de acesso a dados (data Layer). De seguida foi desenvolvido o Diagrama de Classes visto que já sabíamos as classes que nosso projeto teria que ter, nele(diagrama de classes) apresentamos os atributos e as operações que cada classe deve ter. Findadas estas representações e, aumentando ainda mais o nível de detalhe, partimos para o desenvolvimento dos Diagramas de sequência de Implementação. Como tal, especificamos as interações com o sistema que consideramos mais relevantes. Para isso, utilizamos os diagramas de use case criados, o diagrama de classes e os Diagramas de Sequencia , o que nos levou a modelar alguns aspectos dinâmicos do nosso sistema. Também realizamos um Diagrama de Instalação para o ConfiguraFácil que mostra a topologia de hardware que servirá de suporte á nossa aplicação.

Este projeto não foi "linear" como pode parecer, pois à medida que fazíamos a próxima etapa por vezes surgia a necessidade de retroceder alguns passos e editar os Use Case previamente esquematizados e a sua respetiva especificação, de forma a que eles permitissem ao nosso sistema ter todas as funcionalidades que suportam os requisitos pedidos. As limitações do nosso trabalho prendem-se com o facto de os nossos pacotes só terem componentes que não são obrigatórios, o que obriga o utilizador a escolher os componentes obrigatórios á mesma.

Com um olhar crítico sobre o panorama geral do projeto percebemos que a nossa implementação consegue implementar os requisitos do problema em causa. Por outras palavras, o ConfiguraFácil é uma aplicação que se dedica a um stand, sobre o qual consegue fazer configurações de carros e gerir a fábrica processando encomendas e stock.