

Universidade Do Minho

Mestrado Integrado em Engenharia Informática

Desenvolvimento de Sistemas de Software

Adriana Meireles

A82582



Nuno Silva

A78156



Ricardo Pereira

A73577



Shahzod Yusupov

A82617



8 de Novembro de 2018

1 Introdução

Numa sociedade cada vez mais monopolizada pelo tempo, foi nos proposto um trabalho prático pela unidade curricular Desenvolvimento de Sistemas de Software, que consiste no desenvolvimento de um sistema que permitisse a um consumidor de forma mais prática e cómoda poder escolher e customizar um veículo.

Com esta aplicação podemos criar um veículo totalmente customizado de forma a satisfazer as necessidades/desejos de cada cliente, tendo em conta todas as possíveis circunstâncias incompatíveis.

O cliente poderá ainda assim delinear a forma de pagamento pretendida e, caso seja necessário, um plano de pagamento acomodado a seu desejo.

2 Análise de Requisitos

Após uma análise ao enunciado, tivemos de identificar o que seria mais relevante para o programa poder cumprir com todos os requisitos propostos.

Primeiro, a aplicação a ser desenvolvida deverá suportar a autenticação do Funcionário de modo a ser possível iniciar o desenvolvimento da Encomenda. O mesmo começa pela escolha do modelo do veículo. Após isso, inicia-se o processo de Configuração que passa por escolha de pacotes ou desenvolver uma configuração personalizada.

Na escolha de pacotes, a aplicação mostra os vários pacotes pré-definidos que são a agregação de diversos componentes relacionados que têm de ser adquiridos como pacote fechado, isto é, não será possível comprar o pacote e

só querer alguns dos componentes incluídos. Nesta situação, se o cliente não quiser escolher o componente então não se aplica a totalidade do pacote.

Caso o cliente decida fazer uma configuração personalizada tem de escolher componentes obrigatórias que são a escolha da pintura, jantes e pneus e motor. Depois, irá escolher os detalhes que podem ser interiores como, por exemplo, estofos e pacotes de luzes; e podem ser exteriores como, por exemplo, vidros escurecidos, para-choques da mesma cor, tecto de abrir. Além disso, terá também de decidir quais as componentes opcionais que deseja e para cada uma é necessário verificar quais são as incompatibilidades que existem e quais as necessidades de outros componentes.

No fim, é realizado o pagamento do veículo por paypal, transferência bancária ou outra forma escolhida pelo cliente.

3 Modelo de Domínio

O modelo de domínio ilustra as classes conceituais significativas no domínio do problema, sem representar as componentes do software.

Assim, este modelo contém todas as entidades envolvidas no sistema como também os relacionamentos existentes entre estas.

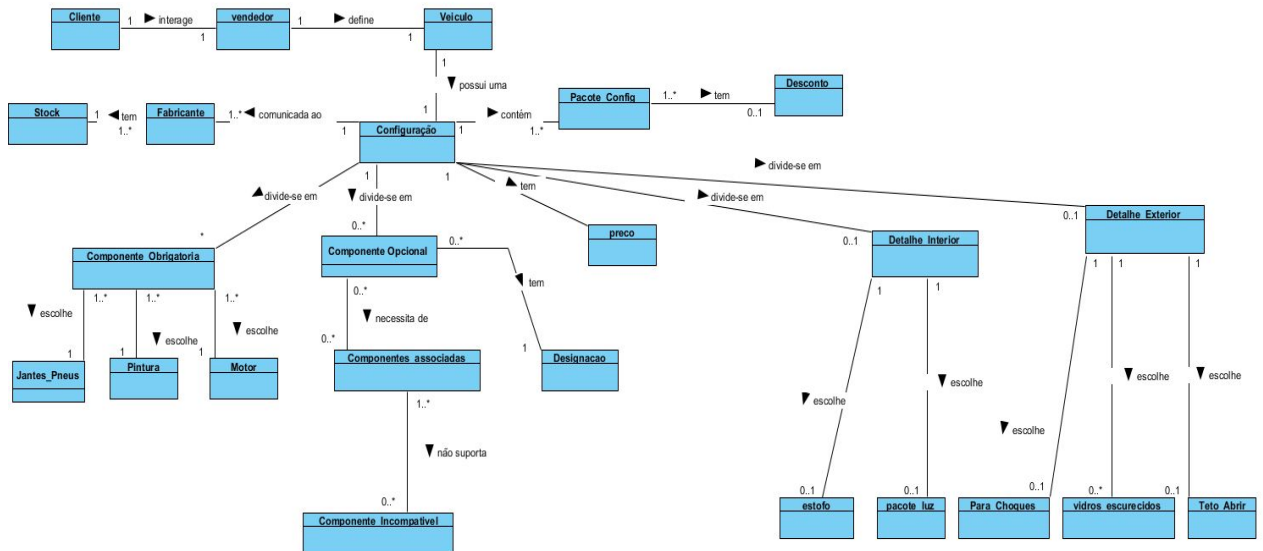
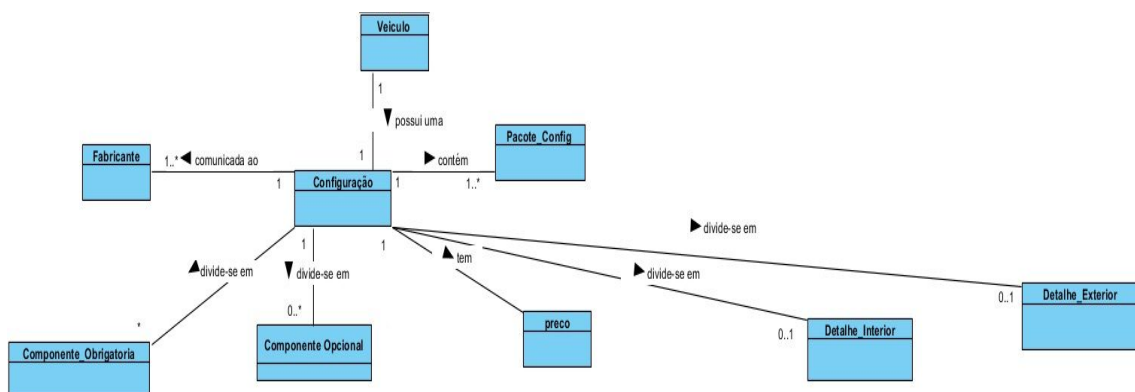


Figura 1: Modelo de Domínio

Posteriormente à análise do modelo de domínio, retiramos que existem entidades principais, tais como:

- Configuração
- Componente Obrigatória
- Componente Opcional
- Detalhe Interior
- Detalhe Exterior



3.1 Configuração

Figura 2: Porção do modelo de domínio correspondente a Configuração

3.2 Componente Obrigatória

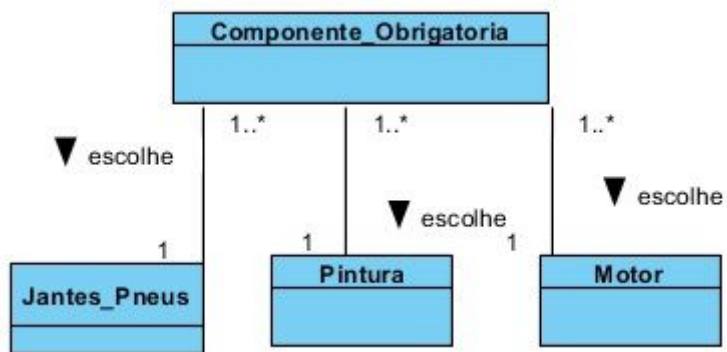


Figura 3: Porção do modelo de domínio correspondente a Componente Obrigatória

3.3 Componente Opcional

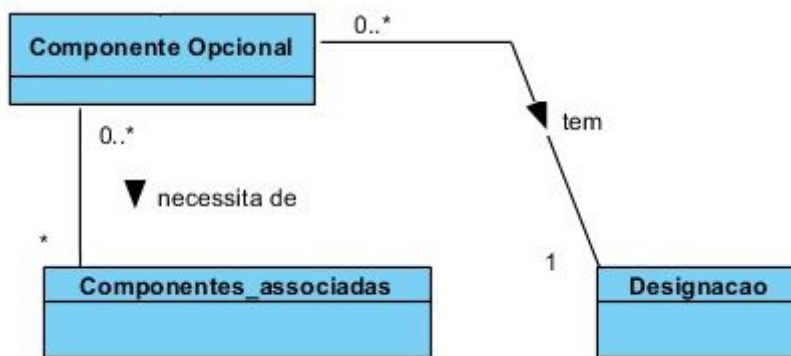


Figura 4: Porção do modelo de domínio correspondente a Componente Opcional

3.4 Detalhe Interior

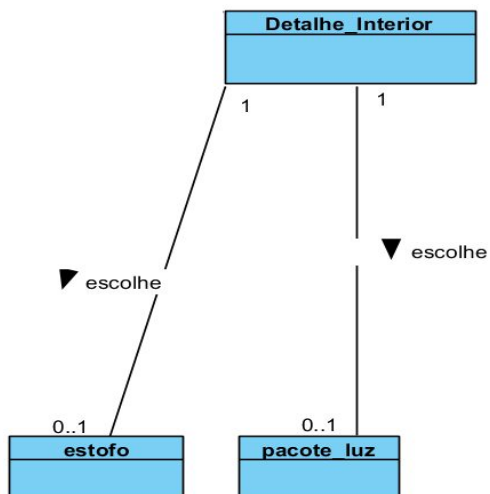


Figura 5: Porção do modelo de domínio correspondente a Detalhe Interior

3.5 Detalhe Exterior

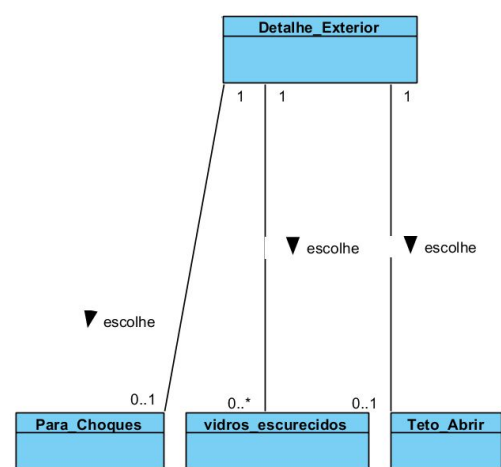
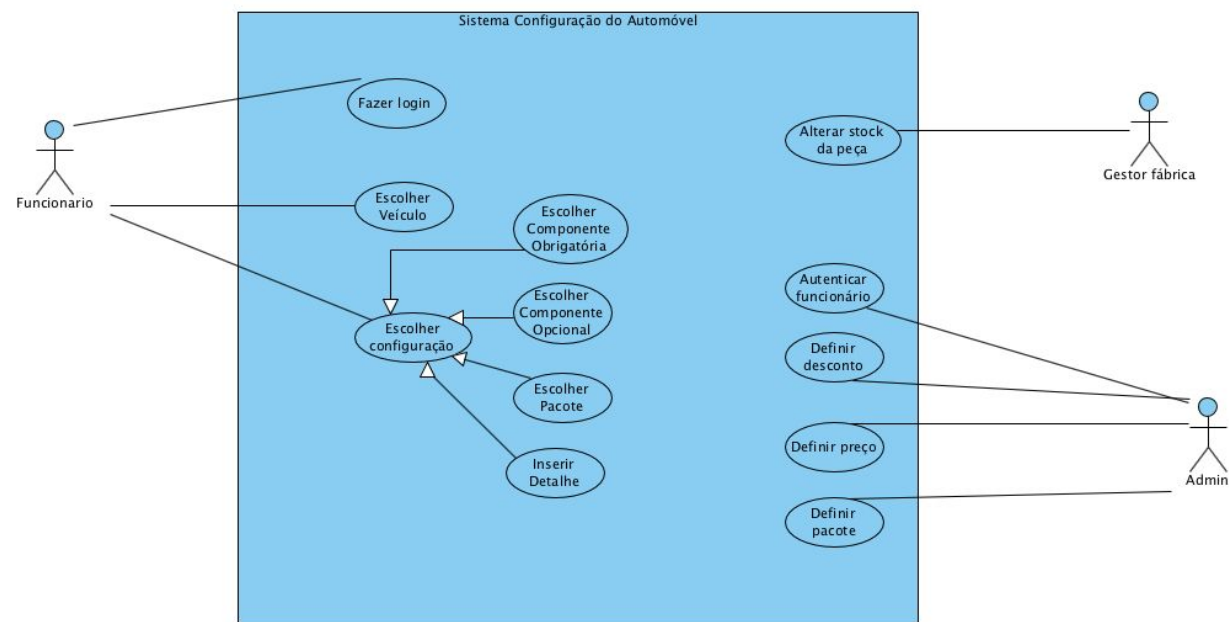


Figura 6: Porção do modelo de domínio correspondente a Detalhe Exterior

5 Diagrama de Use Case

5.1 Diagrama



5.2 Atores

5.2.1 Funcionário: Representa a pessoa responsável por colocar na aplicação a escolha do cliente para a configuração

5.2.2 Gestor Fábrica: Gere o stock da aplicação

5.2.3 Admin:Controla tudo da aplicação

5.3 Use Cases

A maneira ideal de explicar o Modelo de Use Case elaborado, será explicando de forma sequencial a utilização da aplicação.

5.1 Fazer Login

Use Case:	Fazer Login	
Actor:	Funcionário	
Pré condição:	Inicia a aplicação	
Pós condição:	Acesso a aplicação	
Cenário Normal	Actor input	System response
	1.Insere os dados da conta	
		2.Validação dos dados inseridos
		3.Acesso a aplicação e as suas configurações
Exceção 1 (passo2)		1. Informa que os dados inseridos são inválidos

5.2 Escolher Modelo

Use Case:	Escolher veículo	
Actor:	Funcionário	
Pré condição:	Funcionário estar autenticado	
Pós condição:	Modelo do veículo fica decidido	
Cenário Normal	Actor input	System response
	1. Faz login	
	2. Indica modelo do veículo que quer escolher	
		3. Modelo é adicionado à encomenda.
Exceção1 (passo1) [Login inválido]		1. Informa que credenciais do login são inválidas.

5.3 Escolher Componente Obrigatória

Use Case:	Escolher componente obrigatória	
Actor:	Funcionario	
Pré condição:	Componente ainda por escolher	
Pós condição:	Componente adicionada com sucesso	
Cenário Normal	Actor input	System response
	1. Escolhe a componente	
		2. Valida componente
		3. Adiciona as componentes ao veículo
		4. Calcula o preço respetivo
Exceção 1 [Modelo não disponível]		1.1. Informa que componente pretendida não está disponível no stock

5.4 Escolher Componente Opcional

Use Case:	Componente Opcional	
Actor:	Funcionário	
Pré condição:	Não houve escolha de componente opcional	
Pós condição:	Possível acrescimo de componentes opcionais no veiculo	
Cenário Normal	Actor input	System response
	1.Escolhe Componentes	
		2.Valida componentes
		3.Calcula e informa o preço do veiculo
		4.Adiciona novas componentes ao veiculo
Exceção1 (passo1)	1.Rejeita componentes	
		2.Informa que não serão aplicados novos componentes
Exceção2 (passo2)		1.Informa que as componentes são inválidas

5.5 Inserir Detalhe

Use Case:	Inserir Detalhe	
Actor:	Funcionário	
Pré condição:	Configuração tem de ser selecionada	
Pós condição:	Detalhes ficam decididos	
Cenário Normal	Actor input	System response
	1.Escolhe os detalhes interiores	
		2.Verifica se a componente pretendida
		3.Detalhes são adicionados à encomenda.
	4.Escolhe os detalhes exteriores	
		5.Verifica se a componente pretendida
		6.Adiciona detalhes à encomenda.
		7. Averigua se os detalhes escolhidos
		8.Atualiza o preço da viatura após a
Cenário Alternativo1 (passo2) [Detalhe interior indisponível]		1. Informa que determinada peça está indisponível, oferecendo outras
	2.Cliente aceita e decide que alternativo pretende.	
		3.Regressa a 3.
Cenário Alternativo2 (passo5) [Detalhe exterior indisponível]		1. Informa que determinada peça está indisponível, oferecendo outras
	2.Cliente aceita e decide que alternativo pretende.	
		3.Regressa a 6.
Exceção1 (passo5) [Componentes incompatíveis]		1.Indica que as componentes são incompatíveis.

5.6 Escolher Pacote

Cenário Normal	Actor input	System response
	1. Indica pacote que quer Escolher	
		2.Verifica se pacote tem as peças todas disponíveis em stock.
		3.Verifica quais componentes incompatíveis e necessidade de outros.
		4.Calcula preço já com desconto associado.
		6.Pacote é adicionado à encomenda
Alternativo1 (passo2)		1.Informa que determinada peça está indisponível.
break1 (passo3) [Nao quer escolher componentes]	2.Decide escolher outro pacote de configuração.	
		3.Regressa a 3
		1.Informa que não é aplicada a Totalidade do pacote

5.7 Alterar stock da peça

Use Case:	Alterar stock da peça	
Actor:	Gestor da fabrica	
Pré condição:	Escolheu o veiculo em questão	
Pós condição:	Stock reabastecido com sucesso	
Cenário Normal	Actor input	System response
	1.Pede a lista de peças	
		2.Fornece a lista de peças
	3.Escolhe a peça em questão	
	4.Adiciona mais peças ao stock	
		5.Informa que adição feita com sucesso
Excecao 1 [Sto		5.1.Informa que nao pode ser adicionado mais peças

5.8 Autenticar Funcionário

Use Case:	Autenticar Funcionário	
Actor:	Admin	
Pré condição:	Candidatura de um funcionário a uma conta	
Pós condição:	Adição de um funcionário ao sistema	
Cenário Normal	Actor input	System response
	1.Inserção de dados	2.Validação dos dados inseridos
Alternativa1 (passo 2)		1. Informa que um determinado dado é inválido
	2.Admin retifica os dados inválidos	
		3.Regressa passo 2
Excepção1 (passo 2)		1. Informa que um determinado funcionário já existe no sistema

5.9 Definir Desconto

Use Case:	Definir desconto	
Actor:	Administrador	
Pré condição:	Existência de um ou mais pacotes	
Pós condição:	Pacote escolhido fica com um valor de desconto	
Cenário Normal	Actor input	System response
		1. <<include>> Autentica utilizador
		2. Apresenta pacotes
	3. Indica pacote	
		4. Calcula o preço total do pacote
		5. Apresenta o preço total do pacote
	6. Introduce percentagem de desconto no pacote	
		7. Calcula o preço total do pacote com desconto
		8. Apresenta o preço total do pacote com desconto
	9. Confirma a percentagem de desconto	
		10. Guarda as alterações
		11. Confirma o sucesso das alterações
Alternativa 1 (passo 3) [já possui desconto]		1. Informa que já existe desconto para aquele pacote
		2. Possibilidade de alterar desconto
	3. Confirma alteração do desconto	
Exceção 1 (passo 1) [Autenticação falha]		4. Regressa a 4
		1. Informa sobre falha na autenticação
		2. Cancela atribuição do desconto
Exceção 2 (passo 10) [Guardar]		
		1. Informa sobre falha a guardar desconto
		2. Cancela atribuição do desconto

5.10 Definir Preço Peça

Use Case:	Definir preço de componente	
Actor:	Administrador	
Pré condição:	Existir peça	
Pós condição:	Preço atribuído à peça	
Cenário Normal	Actor input	System response
		1. <<include>> Autenticar utilizador
		2. Apresenta os componentes
	3. Seleciona o componente ao qual quer atribuir preço	
		4. Solicita um valor
	5. Introduce valor	
		6. Apresenta o preço introduzido
	7. Confirma o preço da peça	
		7. Guarda o preço
Alternativa 1 (passo 3) [alterar preço]		8. Confirma a alteração do preço
		1. Informa que o componente já tem preço
		2. Questiona a alteração de preço
	3. Confirma alteração de preço	
Exceção 1 (passo 1) [autenticar falha]		4. Regressa a 4
		1. Informa sobre falha na autenticação
		2. Cancela atribuição do preço
Exceção 2 (passo 7) [guardar falha]		1. Informa sobre falha a guardar preço
		2. Cancela atribuição do preço

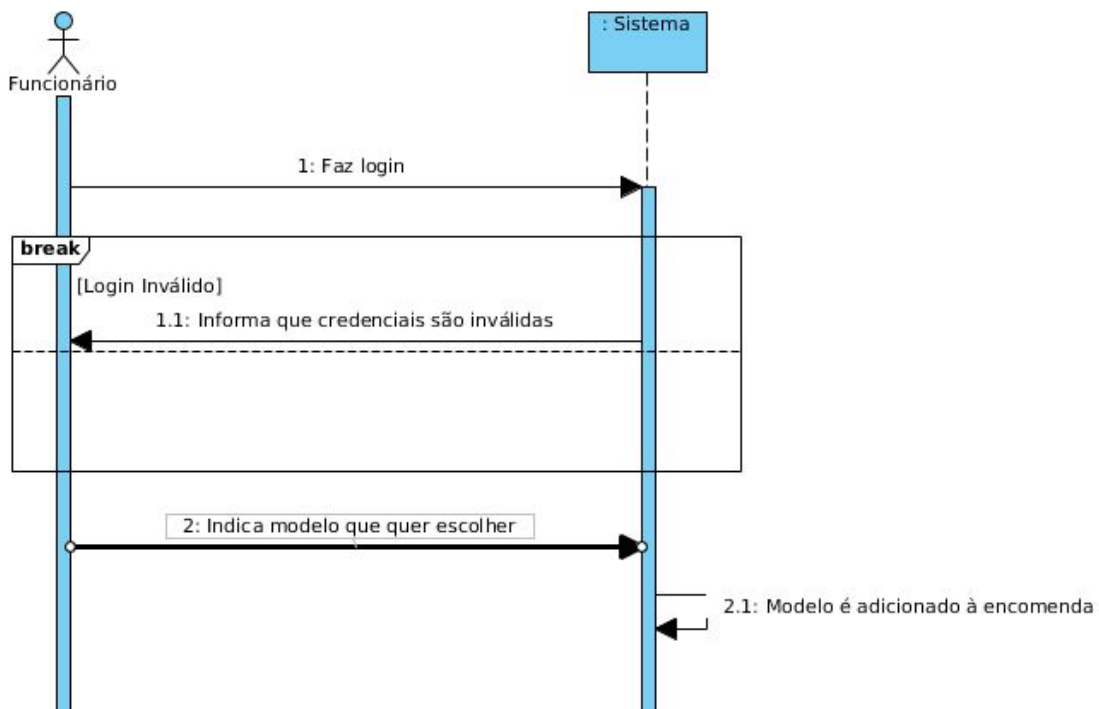
5.11 Definir Pacote

Use Case:	Definir pacote	
Actor:	Administrador	
Pré condição:	Existem vários componentes	
Pós condição:	Existem um ou mais pacotes	
Cenário Normal	Actor input	System response
		1. <<include>> Autentica utilizador
		2. Apresenta componentes
	3. Seleciona os componentes que o pacote irá ter	
		4. Verifica que não existem componentes em conflito
	5. Confirma os componentes do pacote	
Alternativa (passo 4) [Existe conflito]		6. Guarda o pacote
		7. Confirma criação de um novo pacote
Alternativa (passo 4) [Existe conflito]		1. Informa que existem componentes em conflito
		2. Apresenta opções para corrigir o conflito
	3. Escolhe componentes que ficam e os que são removidos	
Exceção 1 (passo 1) [autenticar falha]		4. Regressa a 6
		1. Informa sobre falha a autenticar utilizador
		2. Cancela definição do pacote
Exceção 2 (passo 6) [guardar]		
		1. Informa sobre falha a guardar pacote
		2. Cancela definição do pacote

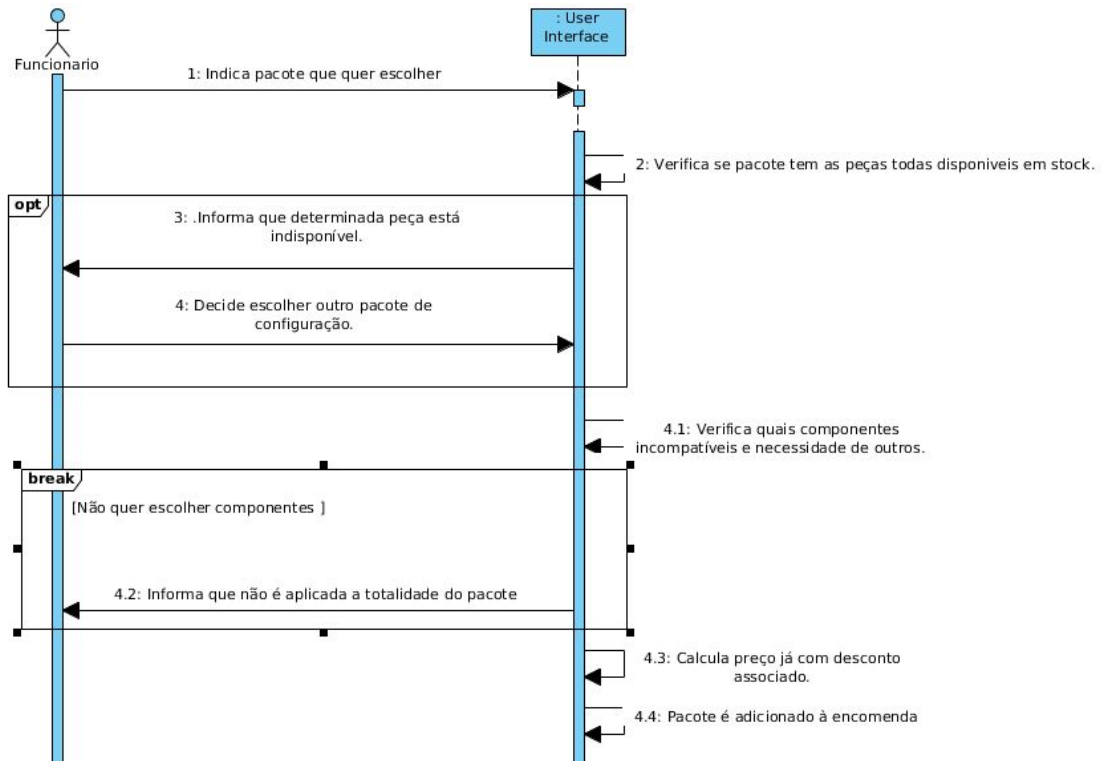
6 Diagrama de Sequência de Sistemas

Nesta secção vamos mostrar todos os diagramas de sequência que foram realizados com base nas especificações dos Use Cases, estes diagramas são denominados por diagrama de Sequência de Sistemas. De uma forma mais generalista, vamos poder ver todo o conjunto de operações que são relevantes para o bom funcionamento da nossa aplicação.

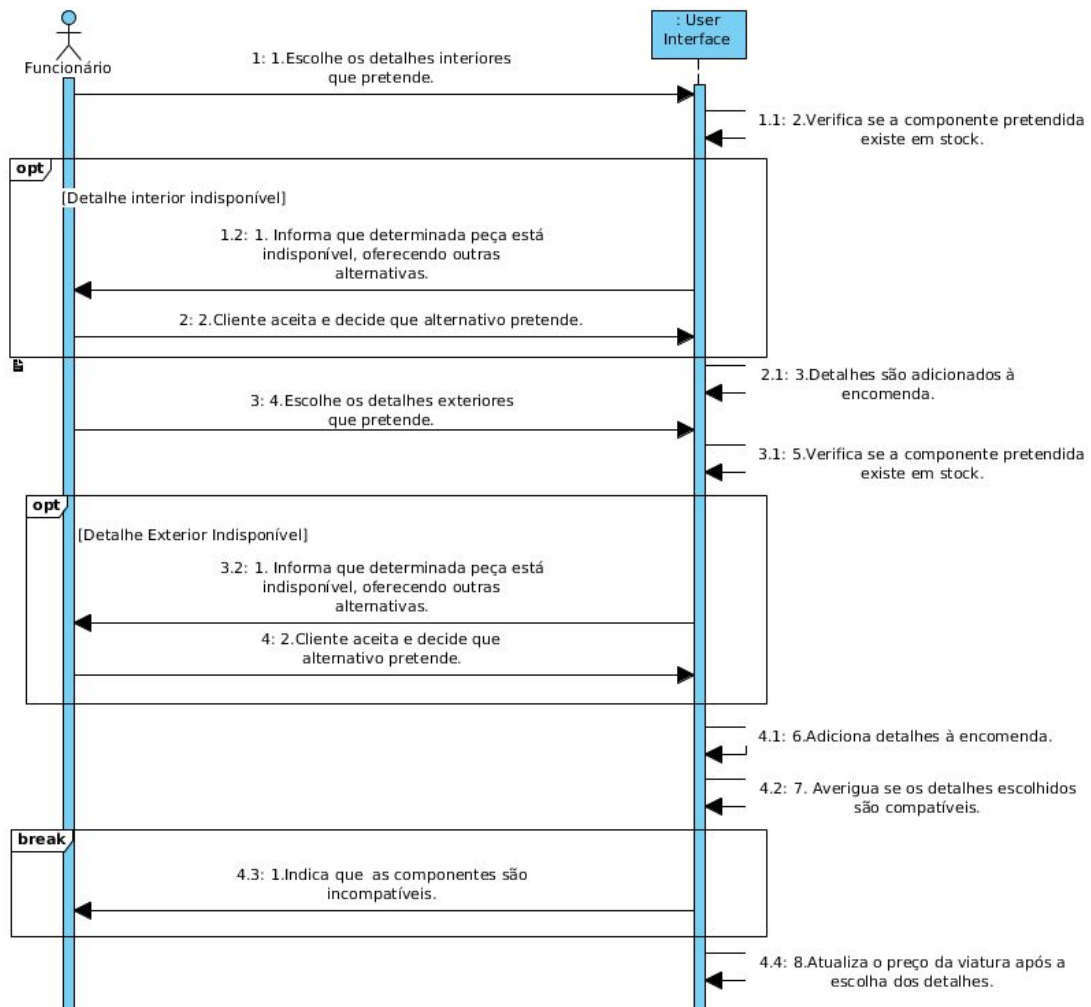
6.1 Escolher modelo



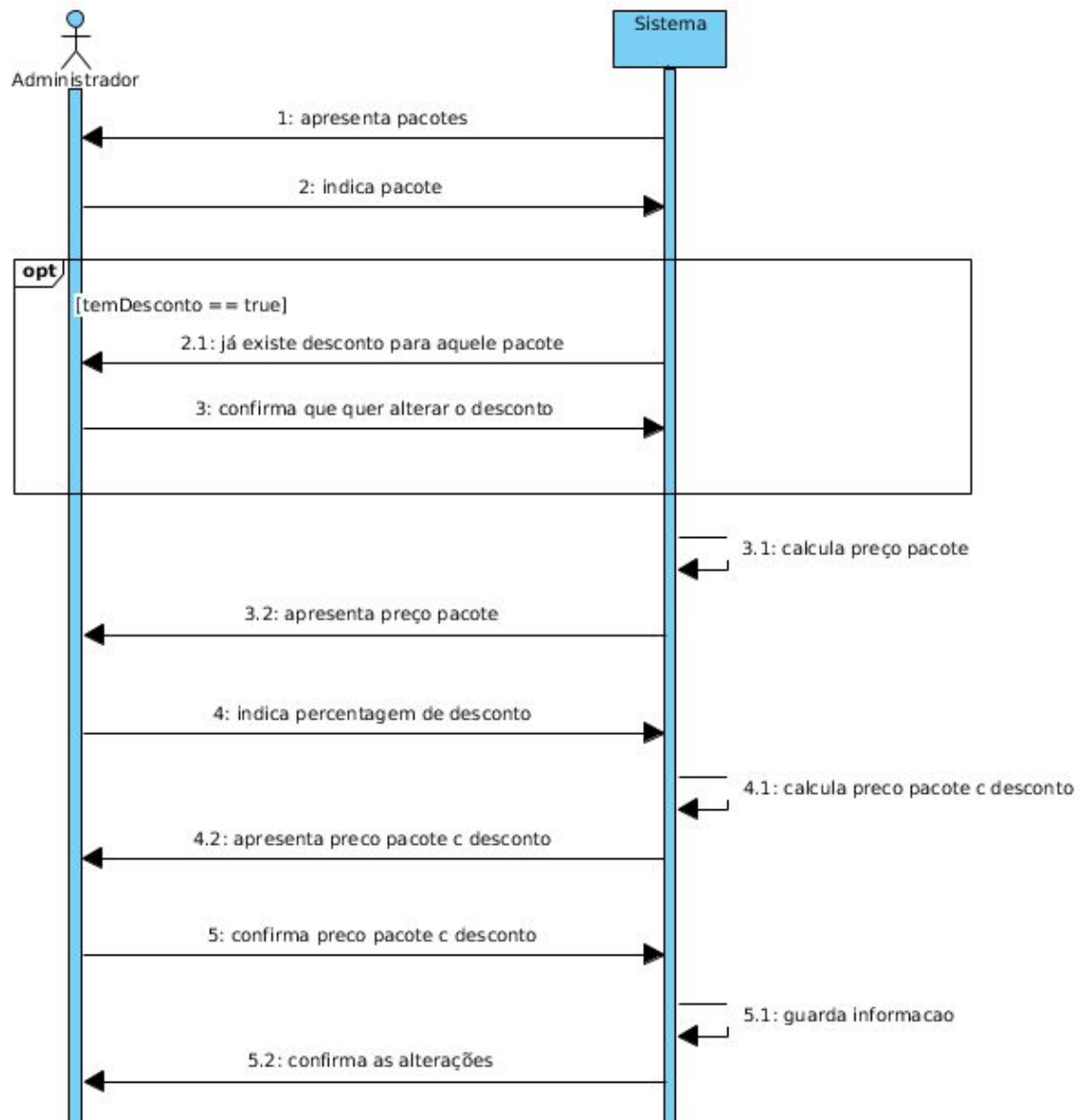
6.2 Escolher Pacote



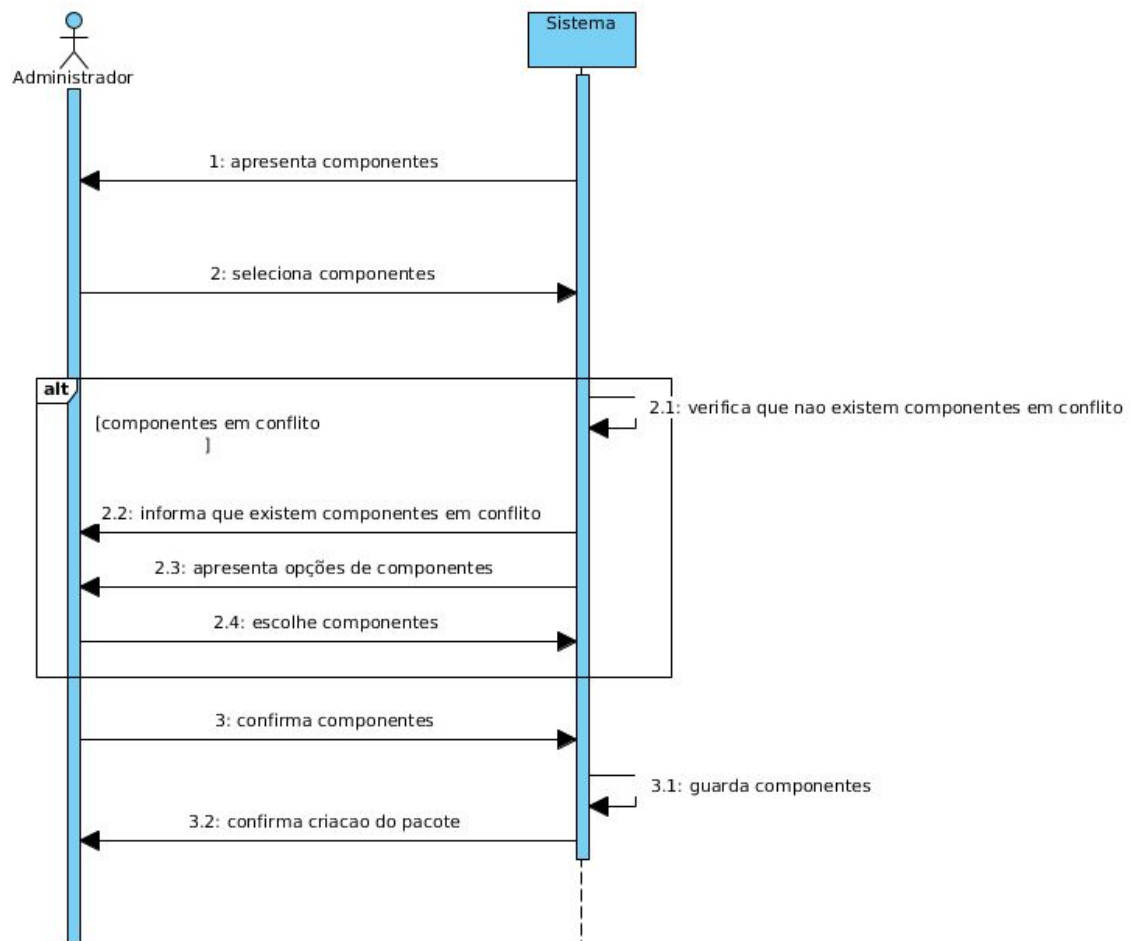
6.3 Inserir Detalhe



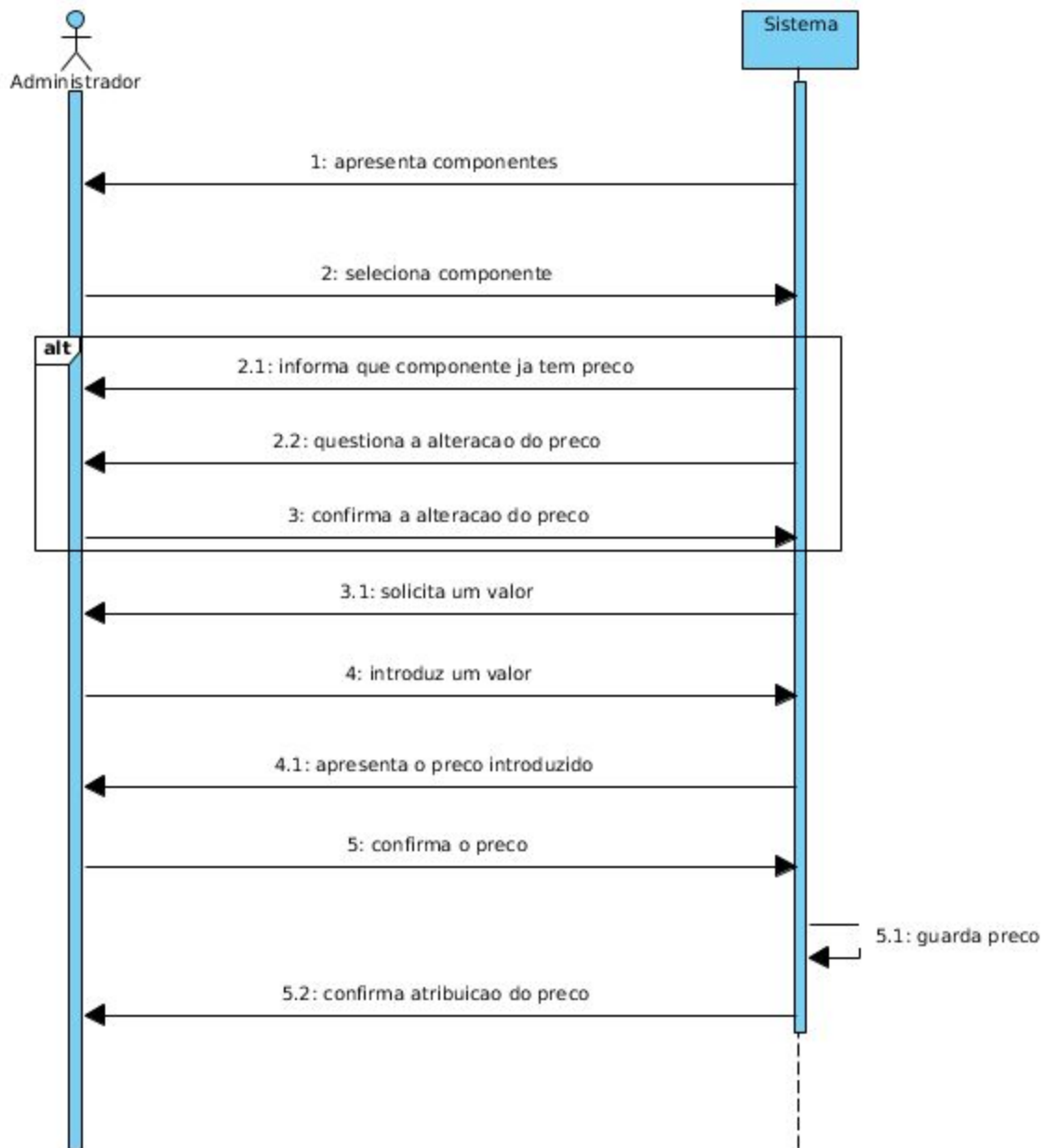
6.4 Definir desconto



6.5 Definir pacote

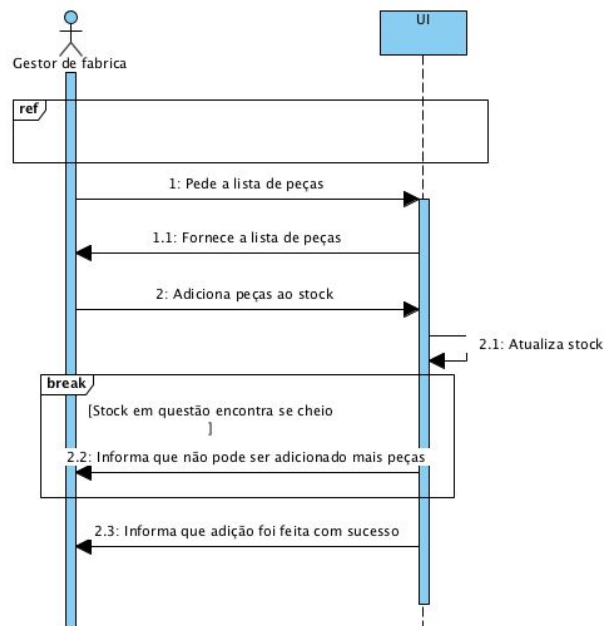


6.6 Definir preço peça



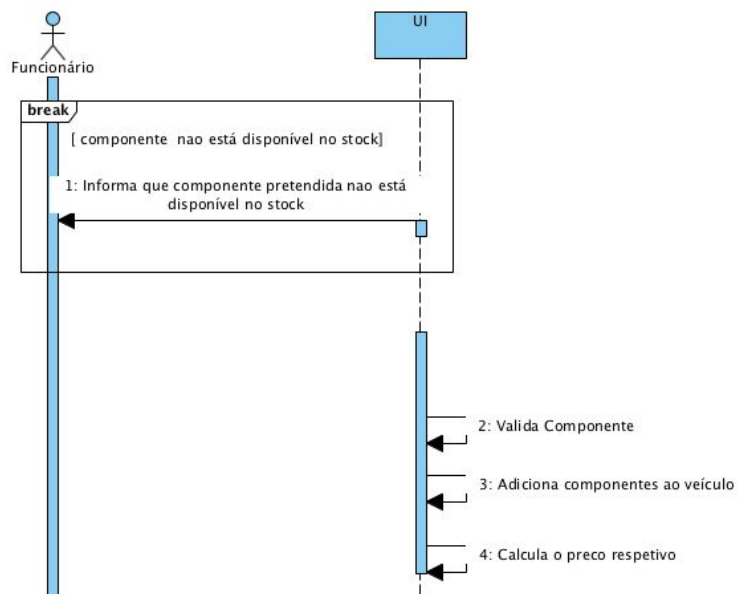
6.7 Alterar stock

sd SistemaAlterarStock



6.8 Escolher Componente Obrigatória

sd SistemaCompObg



7 Diagrama de Classes

A partir do modelo de domínio e do diagrama/especificação dos use case já esquematizados, conseguimos identificar as principais entidades e, conseqüentemente, potenciais candidatas a serem classes no desenvolvimento da nossa aplicação.

Como tal, surge agora o momento de dar um passo em frente na modelação do sistema de software da Ford, implementando por isso os designados diagramas de classe.

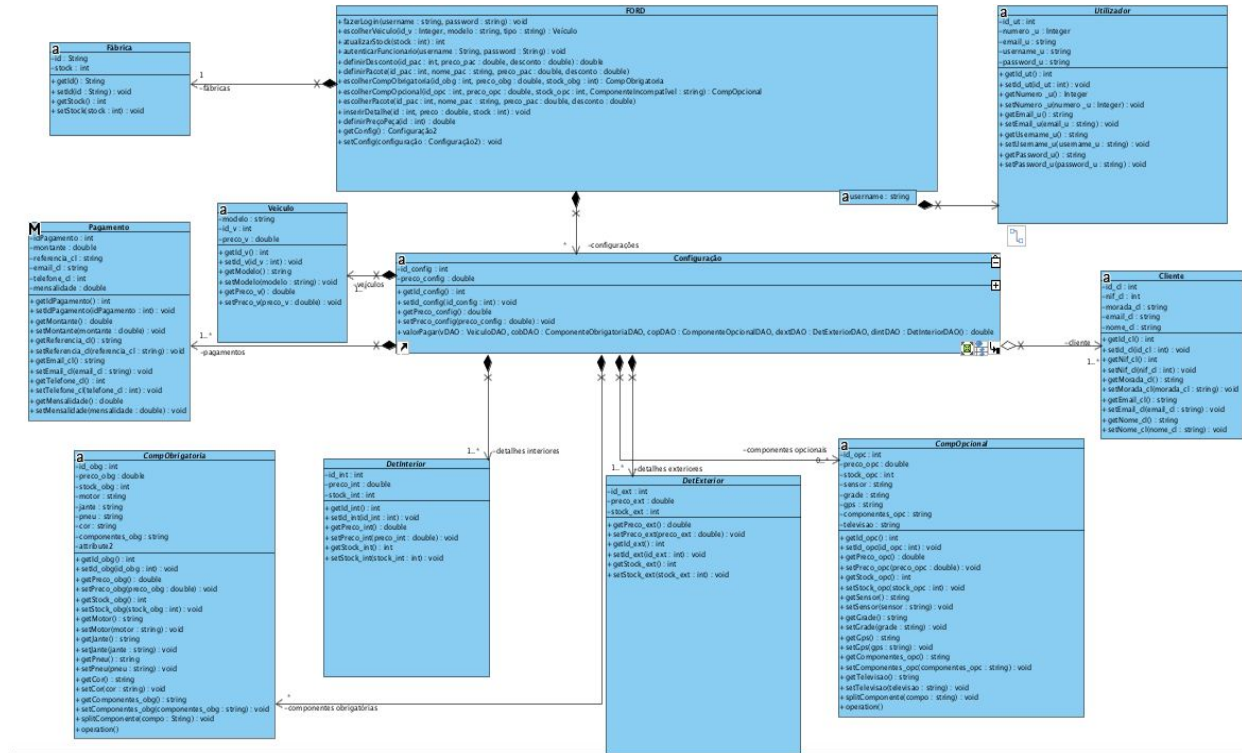
7.1 Breve Explicação do Diagrama de Classes

Um diagrama de classes consiste na representação estrutural das classes que servem de modelo para os objetos do nosso sistema, explicitando também as interações entre elas e o papel que cada uma tem na realização das operações solicitadas pelos utilizadores.

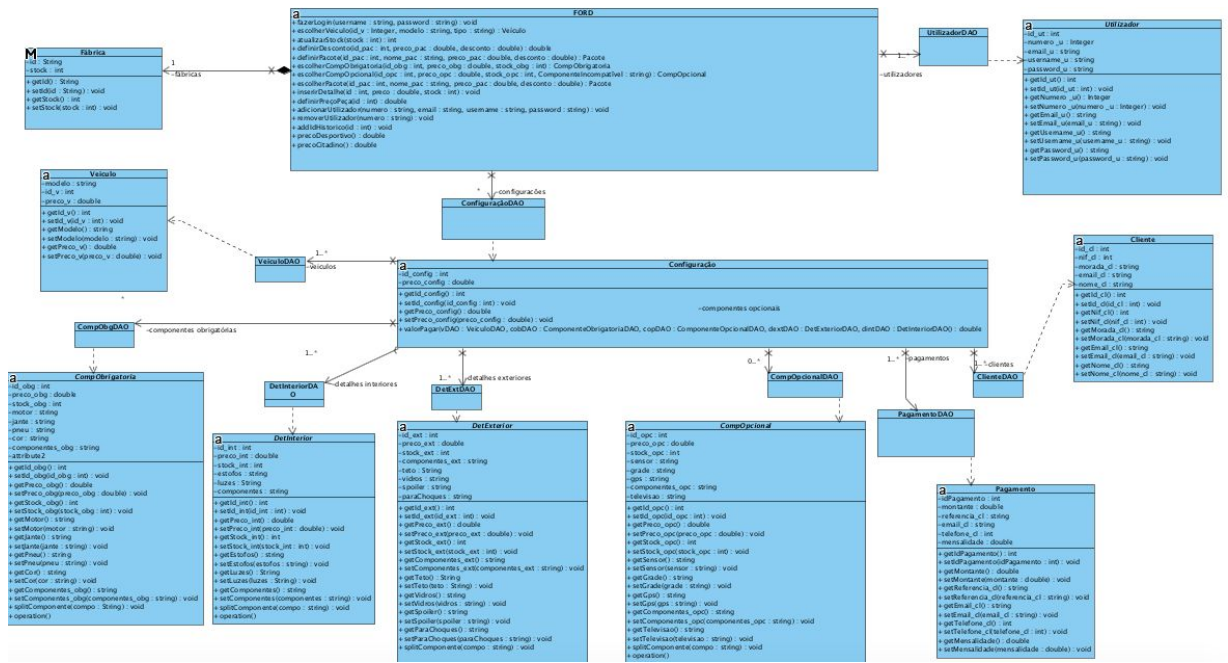
Cada classe representa um grupo de objetos que partilham o mesmo tipo de estrutura (atributos e relacionamentos) e as mesmas operações, podendo estar relacionadas umas com as outras através de associações (mais concretamente agregações ou composições).

Nos seguintes diagramas, a primeira imagem vai corresponder à primeira versão do modelo do diagramas de classes da nossa aplicação onde os dados são guardados em Maps, enquanto que a segunda imagem é a segunda versão onde já implementamos a persistência através dos classes DAO's, ou seja, guardamos os dados numa base de dados. É de referir que todas as classes possuem os respectivos Getters e Setters das variáveis de instância, apenas não são indicados no diagrama, para uma melhor perceção do mesmo.

7.1.1 Diagrama de Classes

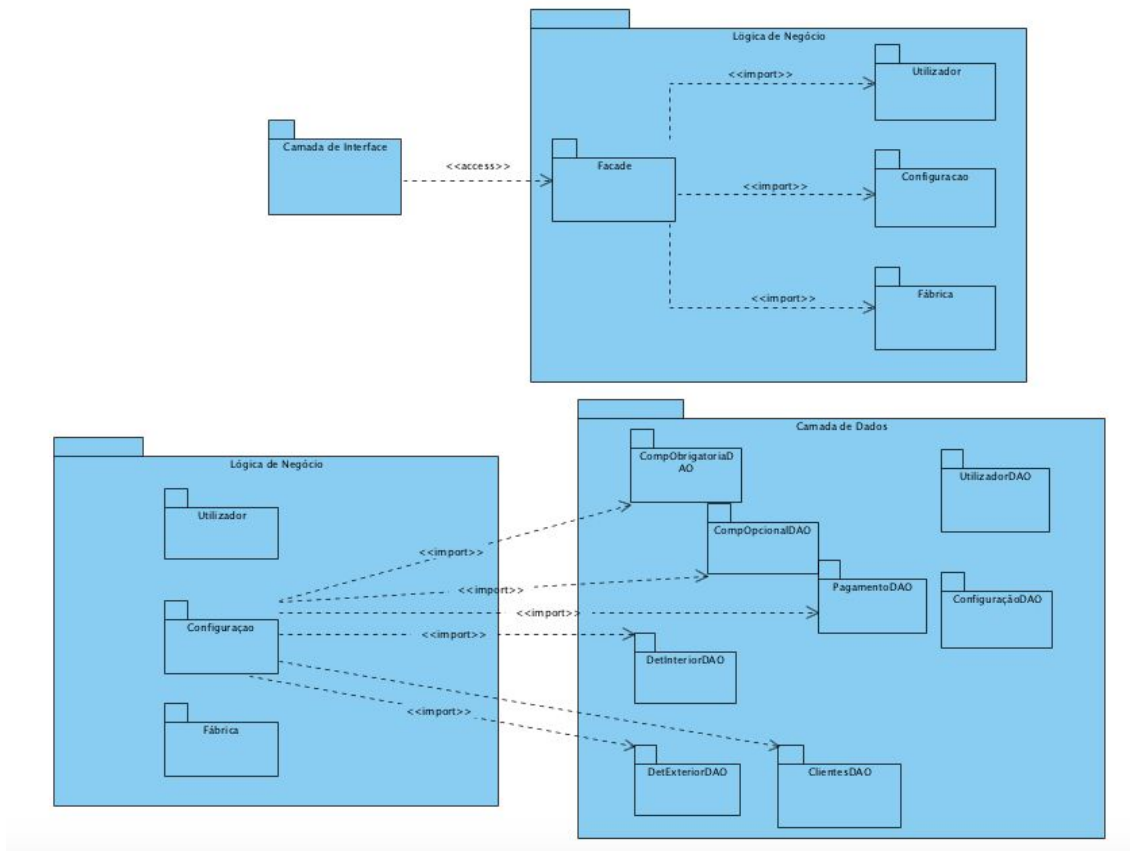


7.1.2 Diagrama de ORM



8 Diagrama de Packages

Neste ponto do projeto já se encontram todas as classes definidas e, como tal, através do diagrama de packages vamos conseguir organizá-las e melhorar a estrutura do modelo do nosso sistema.



8.1 Separação Lógica

O diagrama de packages esquematizado visa distribuir todas as classes que possuímos por packages (grupos) de uma forma lógica e estruturada, sendo que cada package corresponde a um mecanismo de agrupamento genérico com elementos relacionados, elementos esses que podem ser classes, interfaces, use cases, etc.

Note-se também que neste tipo de diagramas os elementos de um package podem ser ou não visíveis para outros fora dele, e que os packages podem apresentar dependência entre si caso a alteração de um afete o outro.

8.2 Package - Camada da Interface

A Camada da Interface vai possuir todas as classes responsáveis por fazerem a comunicação entre o utilizador e o programa

8.3 Package - Lógica de Negócio

A Lógica de Negócio está dividida em quatro packages mais pequenos, uma vez que se trata de um package muito extenso. Assim, a divisão ajuda a identificar todas as operações que ocorrem nesse package.

- Ford (Facade)

O Ford será responsável por fazer a ligação entre a interface e a camada lógica.

- Subsistema Utilizador

No Subsistema utilizador vai ser responsável por conter os 3 tipos de utilizadores: Funcionário, Fabricante e Administrador

- Subsistema Encomenda

No subsistema encomenda vamos efetuar a configuração do veículo dos nossos clientes.

- Subsistema Fábrica

No subsistema Fábrica vai ser responsável por repor o stock da nossa aplicação sempre que necessário.

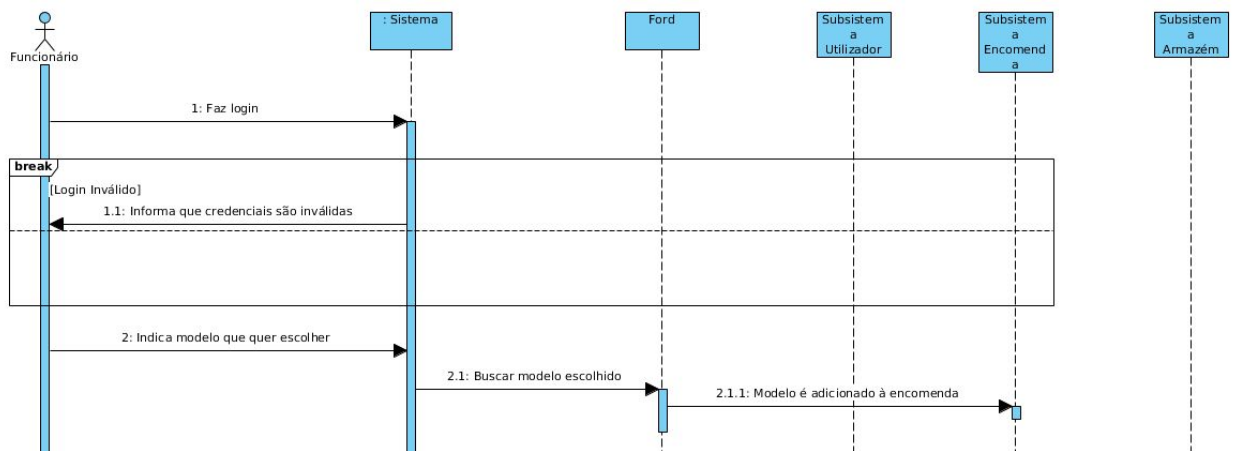
8.4 Package - Data Access

O Data Access vai ser onde vão ser implementados os DAO's, ou seja, todas as operações que estabelecem a conexão da Lógica de Negócio com a base de dados que suporta a nossa aplicação.

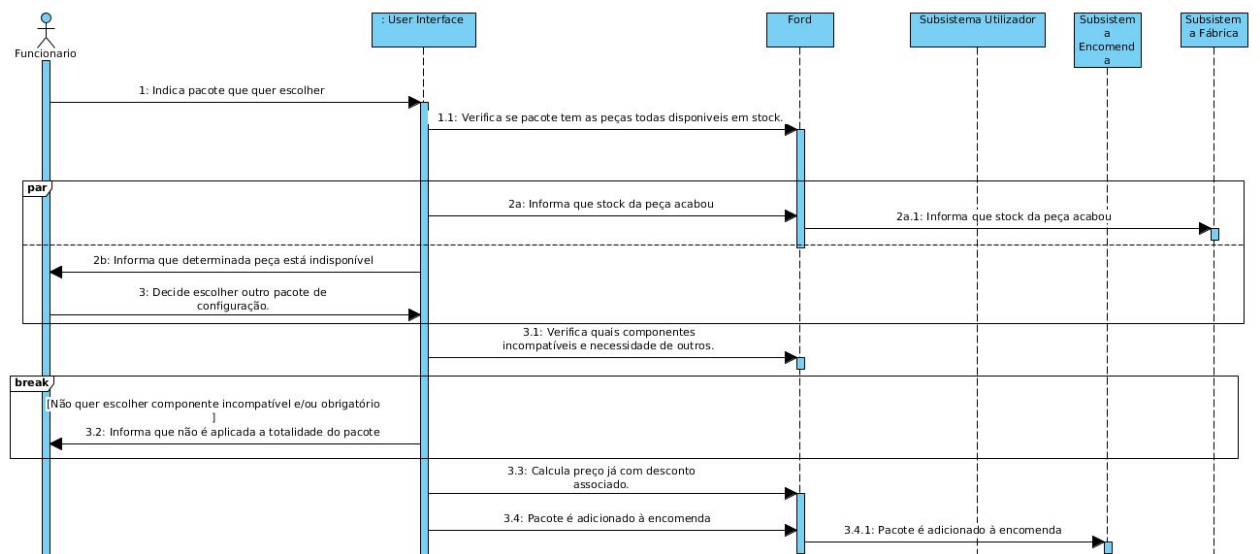
9 Diagrama de Sequência de Subsistemas

Após acabarmos de identificar os subsistemas através do Diagrama de Packages vamos para cada Use Case definir o seu respetivo diagrama de subsistemas, sendo que para isso efetuamos o desdobramento dos diagramas de sistemas nos vários subsistemas, onde iremos especificar mais detalhadamente as interações na nossa futura implementação do código para a aplicação funcionar e cumprir os requisitos corretamente.

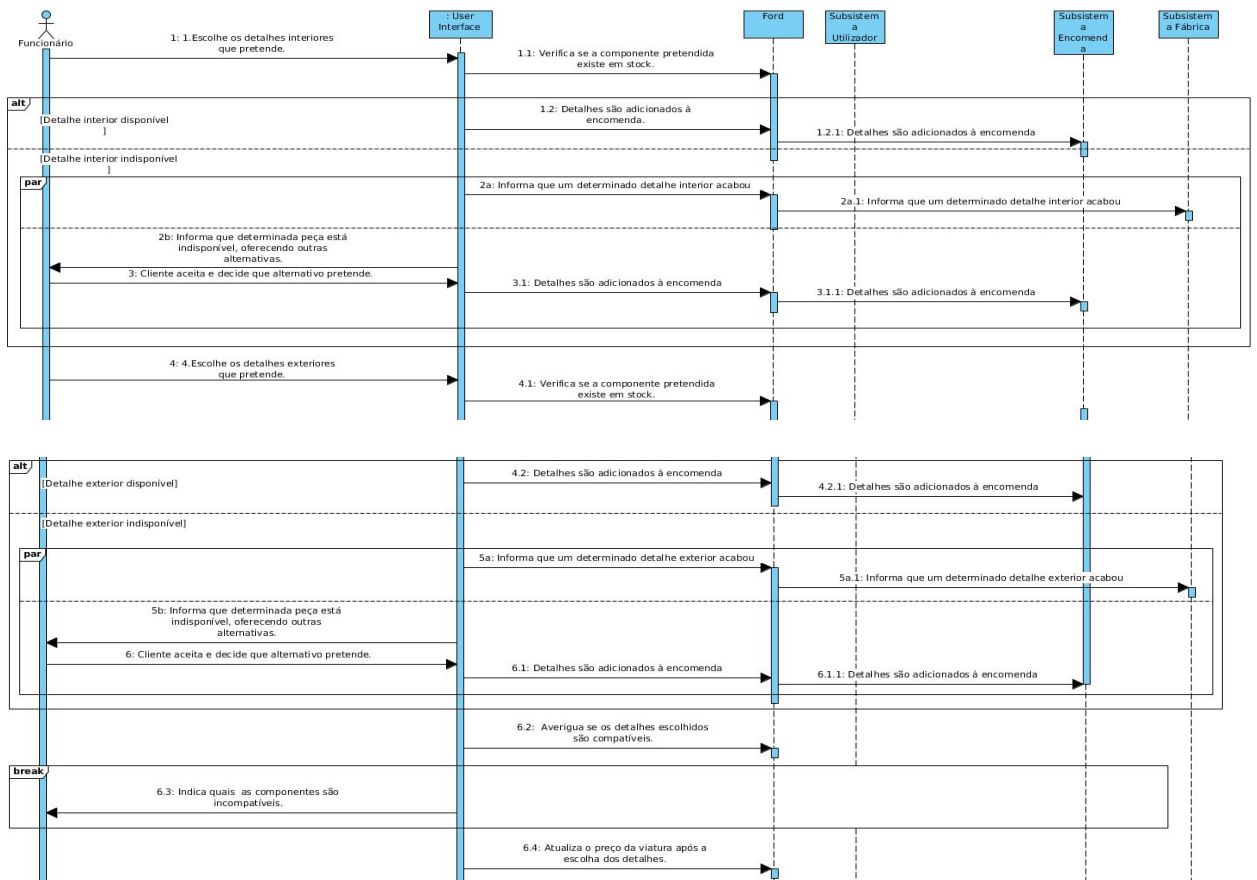
9.1 Escolher modelo



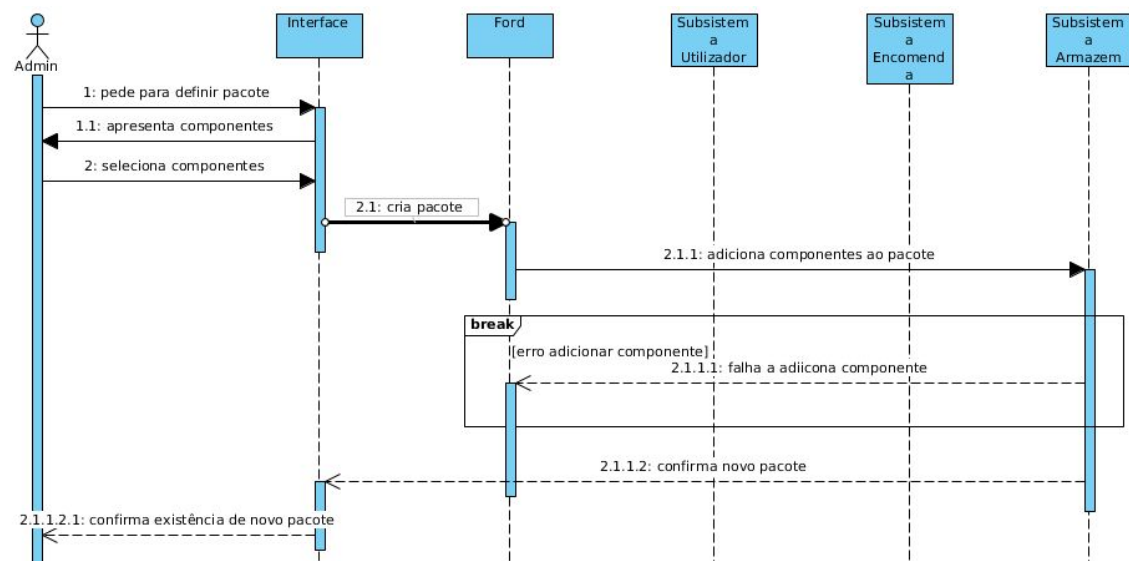
9.2 Escolher Pacote



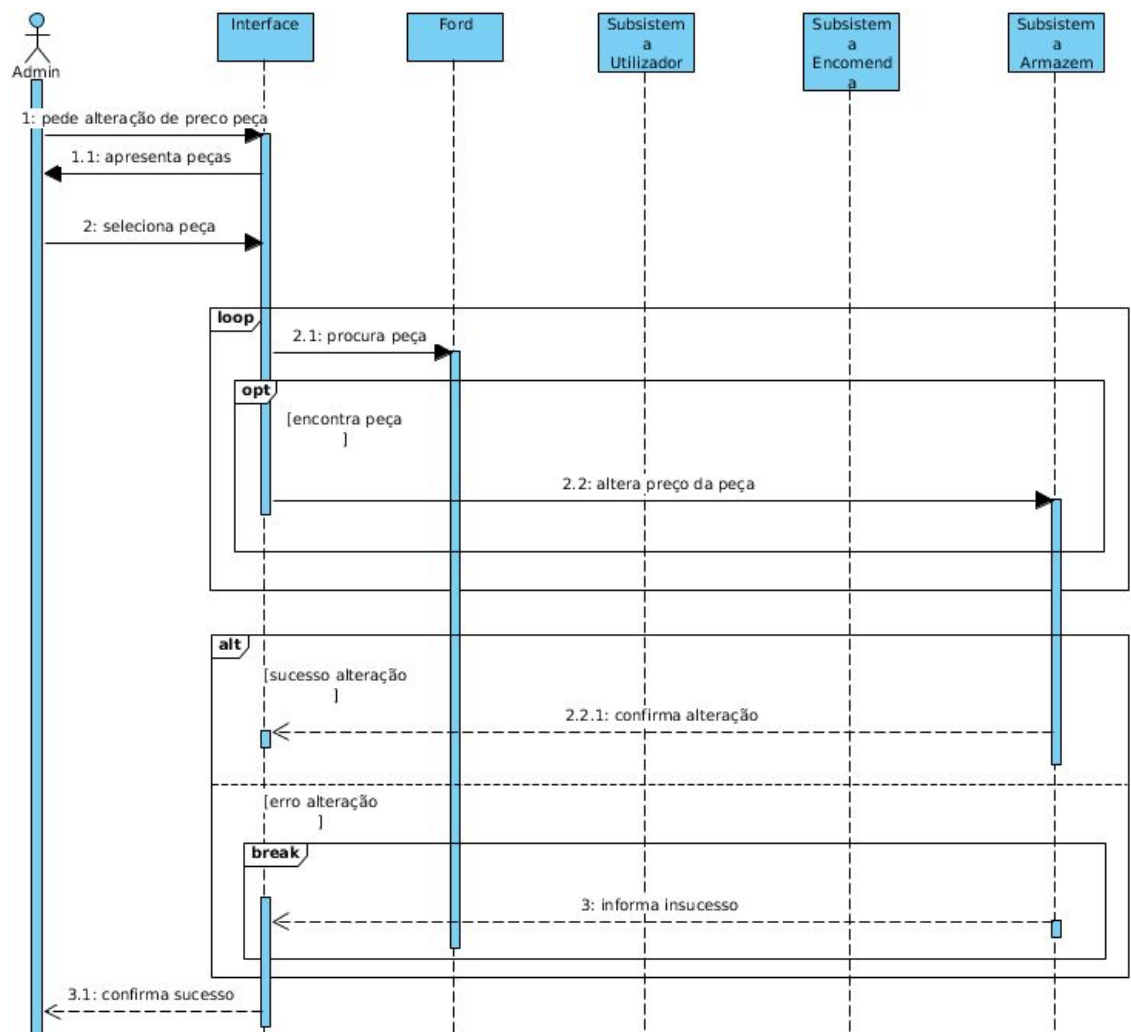
9.3 Inserir Detalhe(2 metades de um)



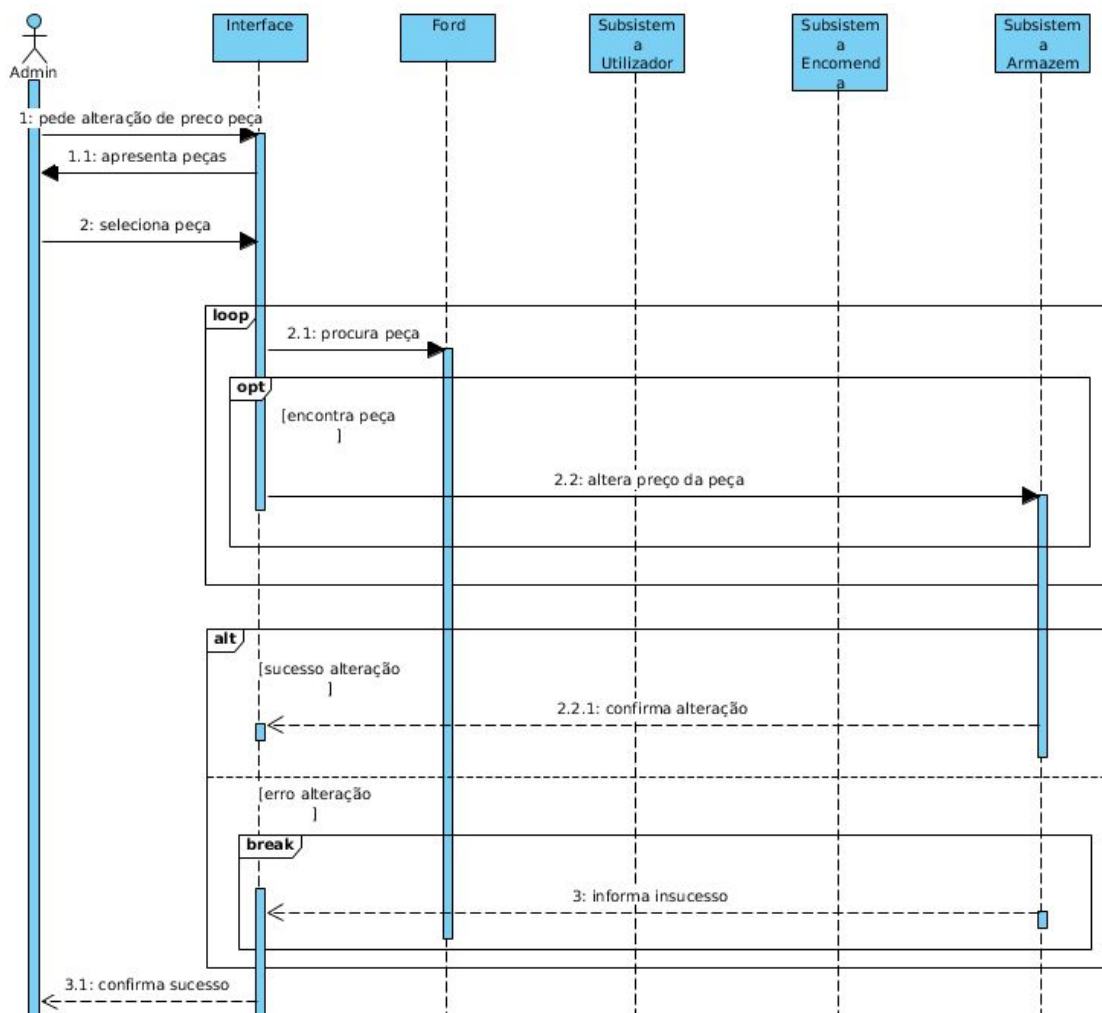
9.4 Definir pacote



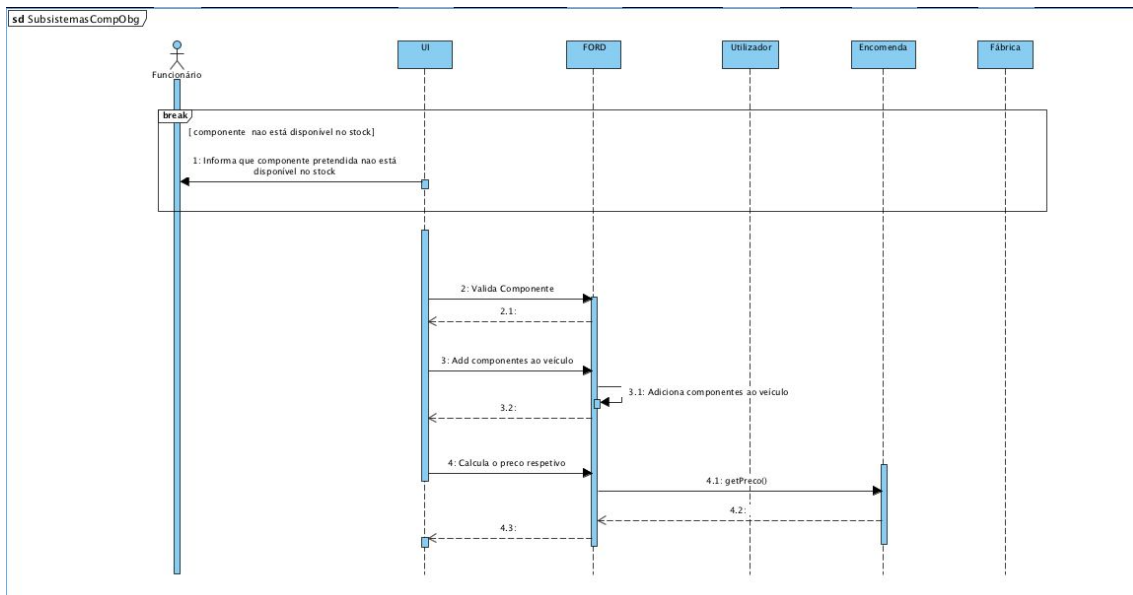
9.5 Definir preço da peça



9.6 Definir desconto do pacote







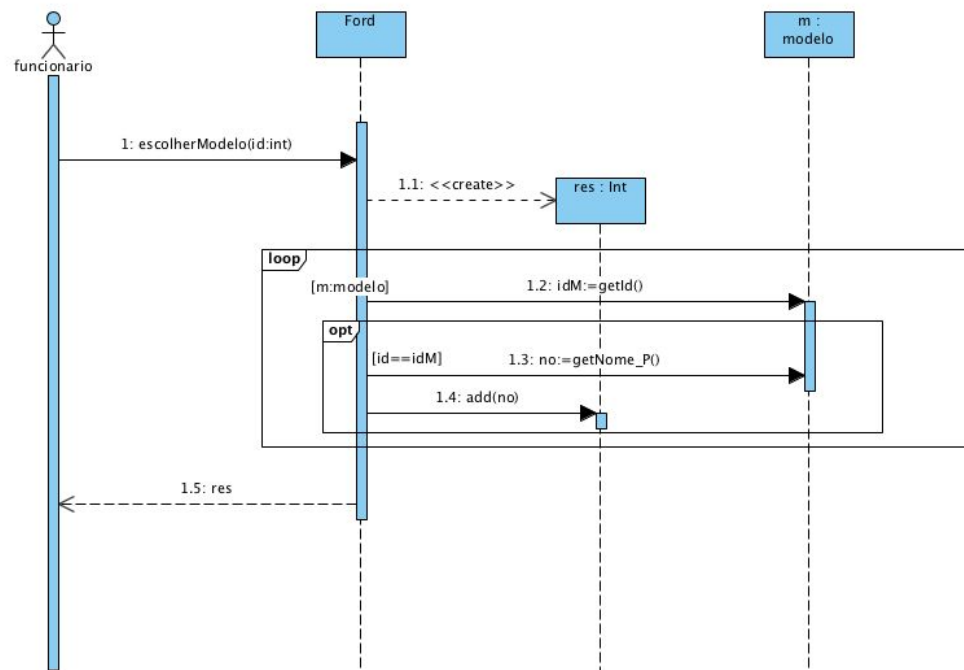
10 Diagrama de Sequência de Implementação

Nesta secção encontramos-nos numa fase próxima da implementação. Vamos desdobrar os use case em diagramas de sequência de Implementação que mostram mais em detalhe o nível da implementação do código das operações realizadas.

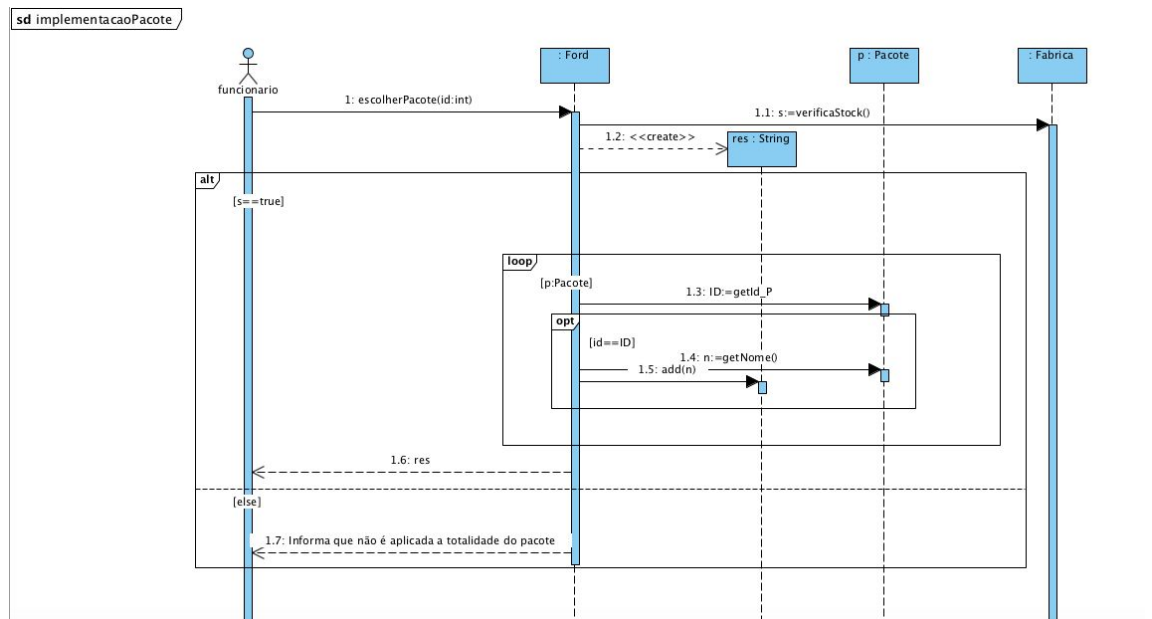
10.1 Escolher Veículo

\

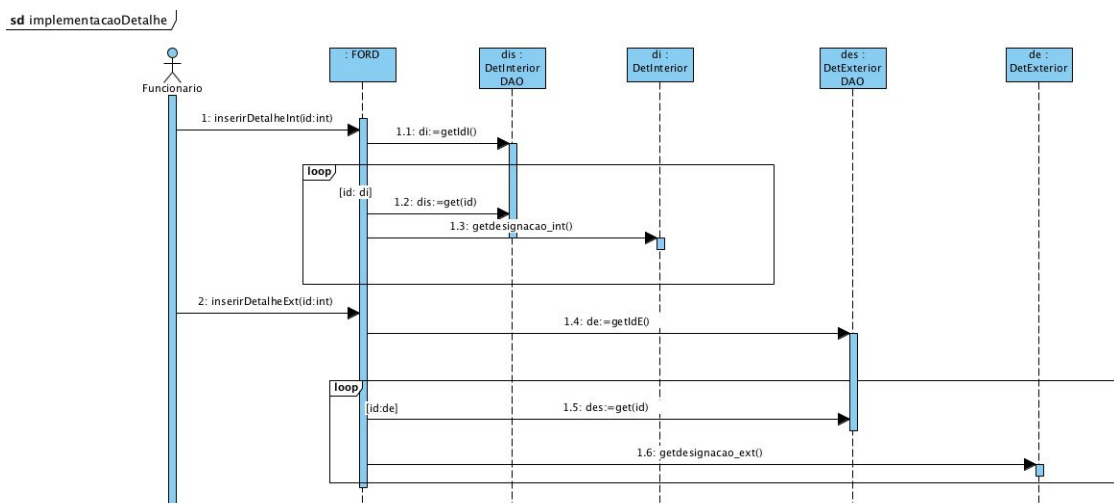
sd implementacaoVeiculo /



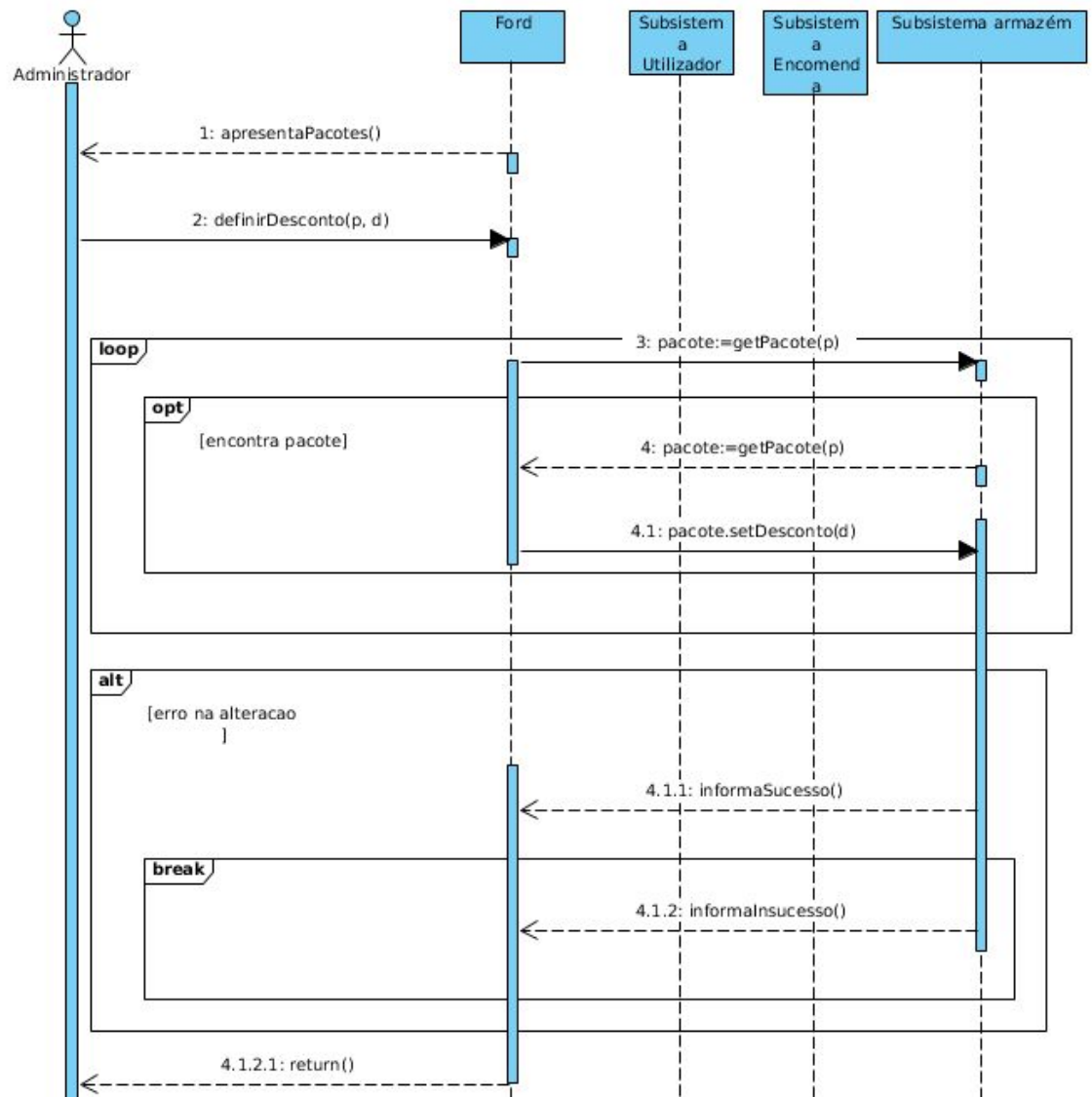
10.2 Escolher Pacote



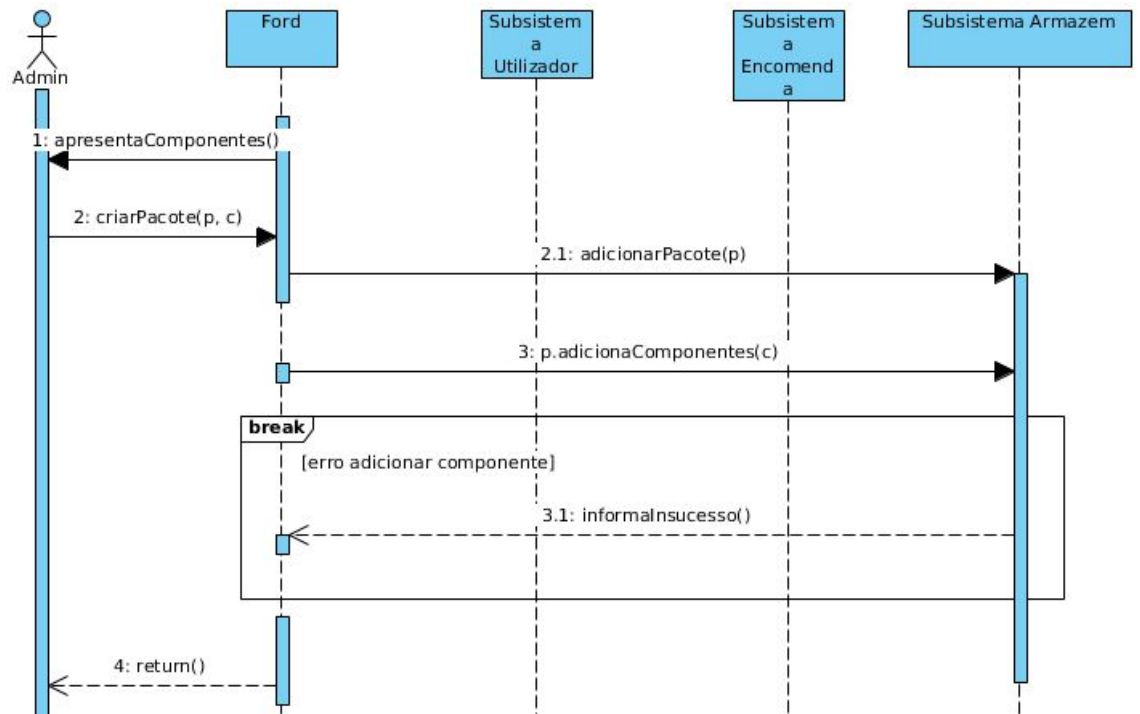
10.3 Inserir Detalhe



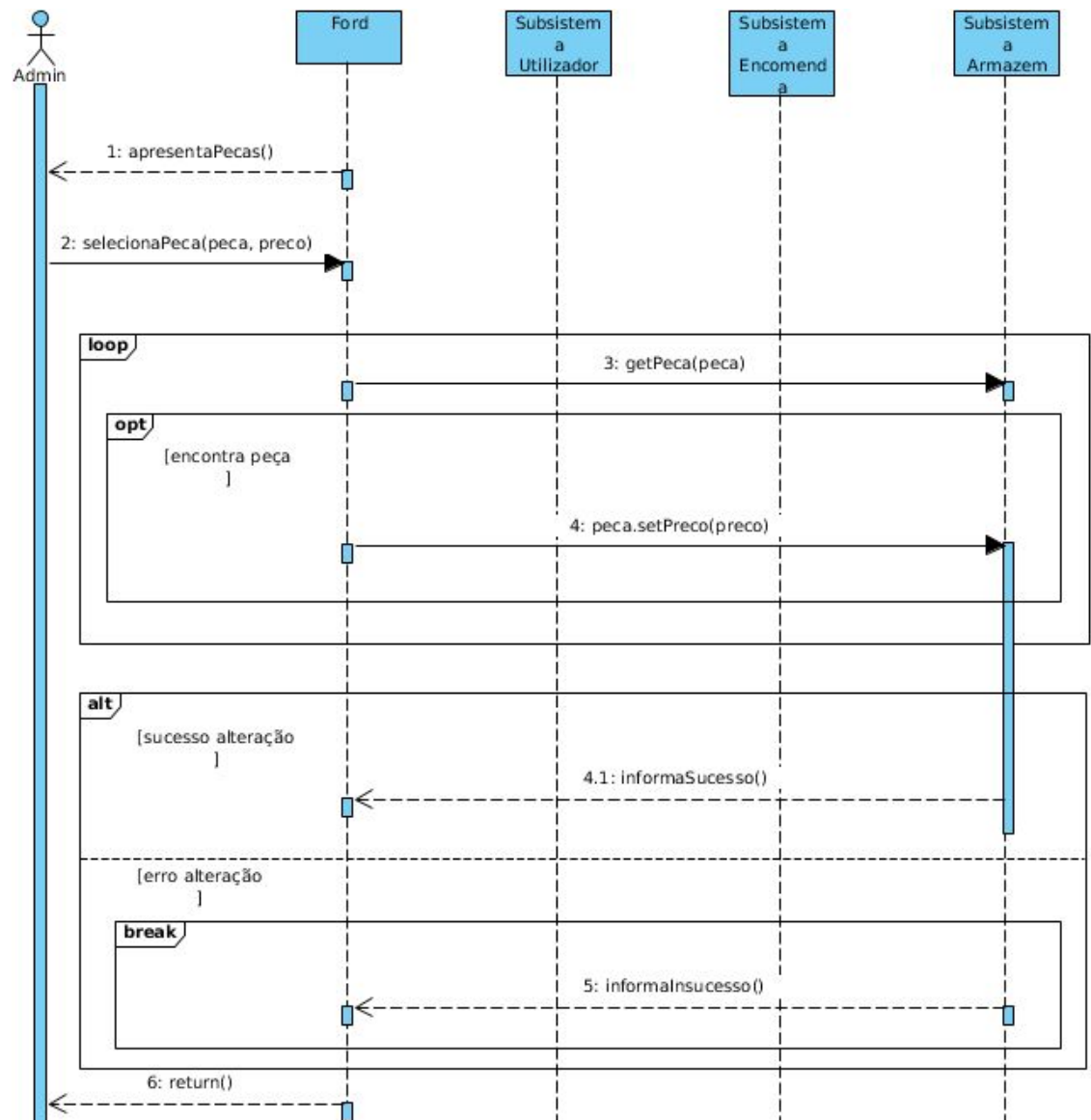
10.3 Definir desconto pacote



10.4 Definir pacote



10.5 Definir preço de uma peça



10.6

10.7

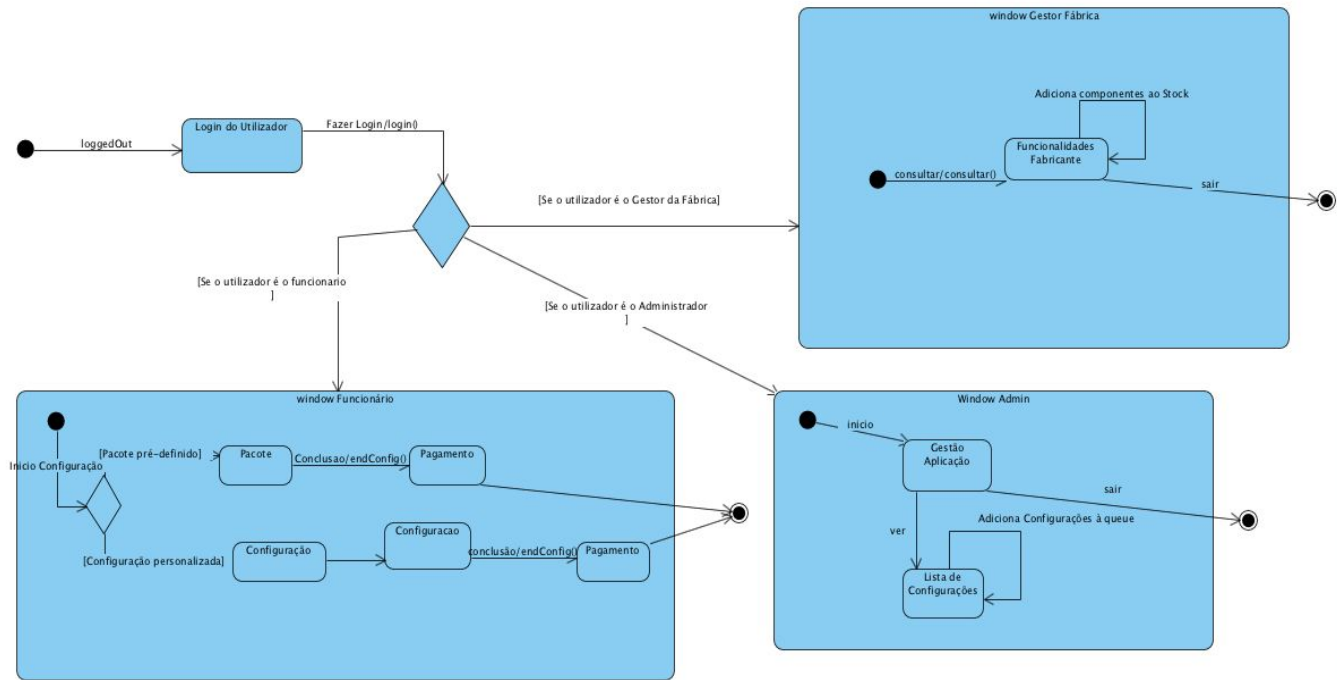
10.8

10.9

10.10

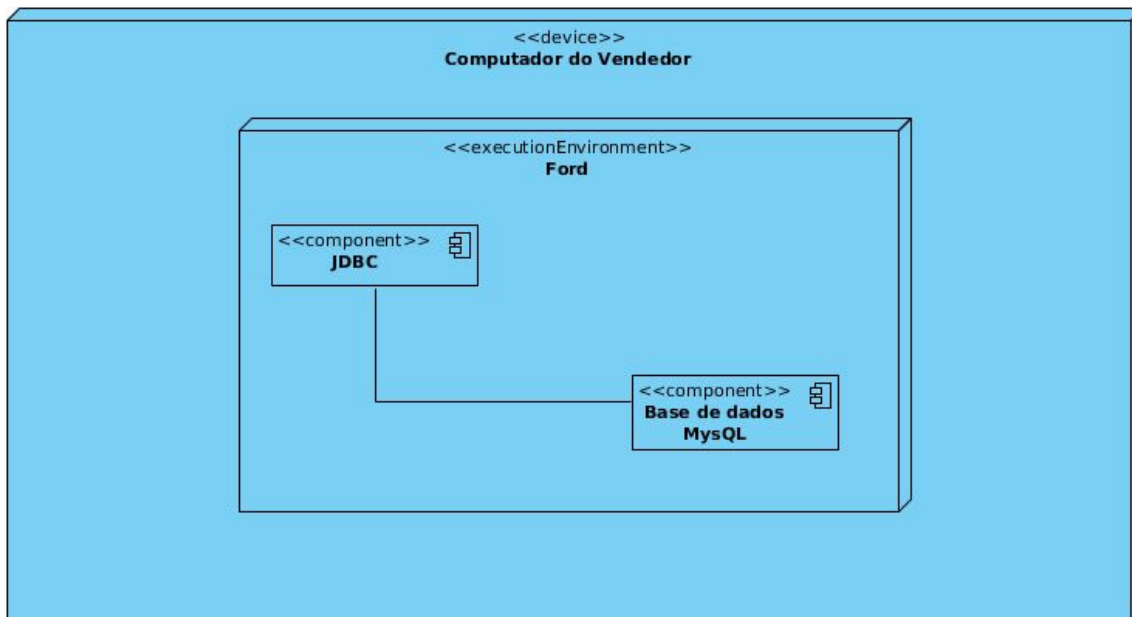
10.11

11 Máquinas de Estado



Uma máquina de estado permite modelar o comportamento de um dado sistema de forma global.

12 Diagrama de Instalação



Nesta fase do projeto, procedemos à esquematização/descrição do nosso sistema de software do ponto dos componentes físicos que ele abrange, modelando por isso a sua topologia de hardware. Apesar do ponto fulcral do UML ser a estruturação e o comportamento do software de um sistema, este tipo de diagrama aborda a parte física da aplicação a ser desenvolvida, demonstrando os componentes hardware nos quais o software desenvolvido se encontra implementado.

De uma forma simplista, o diagrama de instalação da Ford apresentado anteriormente mostra-nos que a aplicação executa no computador de um utilizador, e dentro dela a parte de software da aplicação implementado em código java (`<<component>>`JDBC) procede ao envio de instruções SQL para a uma base de dados relacional em MySQL (`<<component>>`Base Dados MySQL), onde se encontram informações relativas aos veículos,as suas configurações,etc.

13. Idealização da Interface Gráfica na Primeira Fase do Projeto

Na primeira fase realizamos um esboço de como iria ser a nossa aplicação no futuro. Trata-se de um esboço pois pode estar sujeito a alterações, e assim podemos afirmar que se trata de um protótipo da versão final.

13.1 Menu iniciar sessão

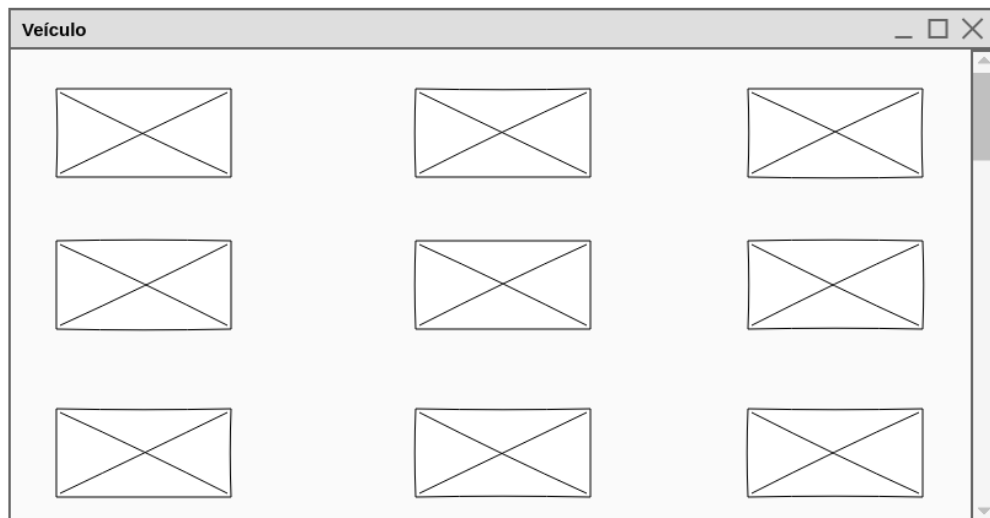


A mockup of a login window titled "Login". The window has a light gray background and a dark gray title bar with standard window controls (minimize, maximize, close). The main content area is white and contains the following elements:

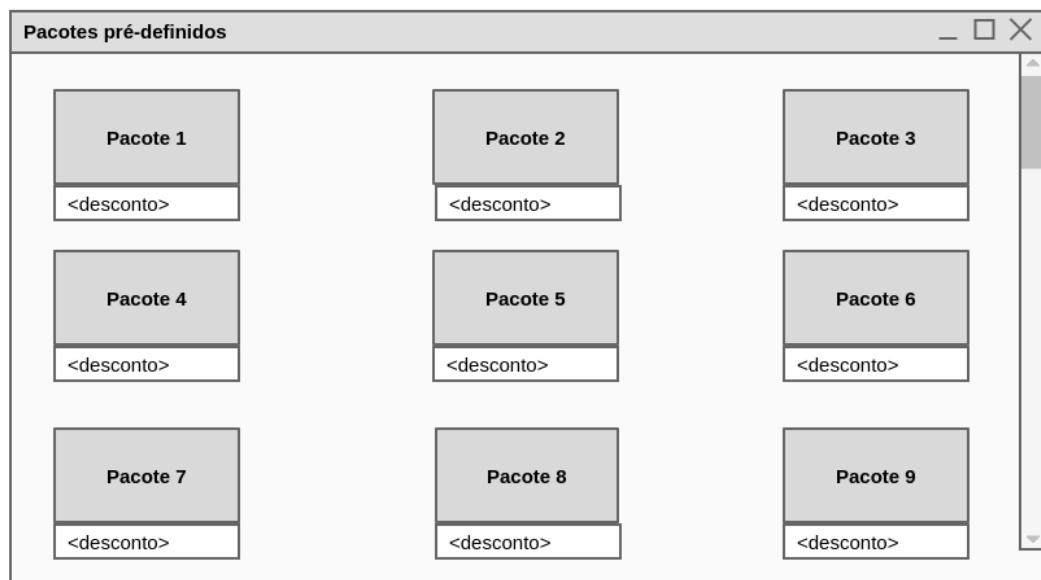
- The text "Bem vindo!" centered at the top.
- A text input field labeled "Utilizador:".
- A text input field labeled "Password:".
- Three radio buttons for user roles: "Vendedor", "Fabricante", and "Administrador".
- A "Login" button at the bottom.

13.2 Início da configuração

13.2.1 Escolha do modelo



13.2.2 Escolha do pacote



13.2.3 Escolha dos componentes obrigatórios

Configuração - Componentes obrigatórios

☐ Pintura 

☐ Jantes 
























































☐ Pneus 

☐ Motor 

Back Next

13.2.4 Escolha da cor da pintura

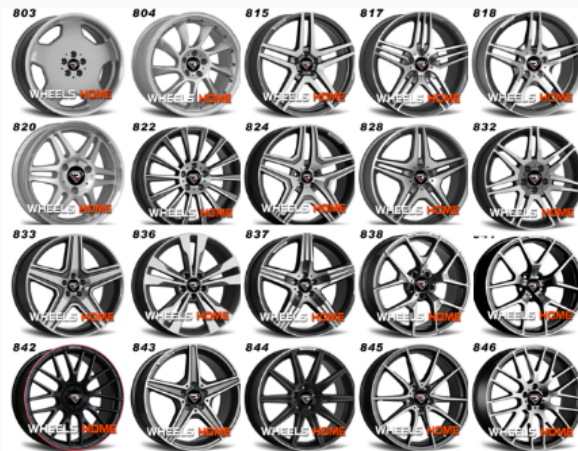
Configuração - Pintura

Back Next

13.2.5 Escolha das jantes

Configuração - Jantes



Back

Next

13.2.6 Escolha de pneus

Configuração - Pneumáticos

- ☐ usados
- ☐ de inverno
- ☐ de verão
- ☐ recauchutados



Back

Next

13.2.7 Escolha do motor

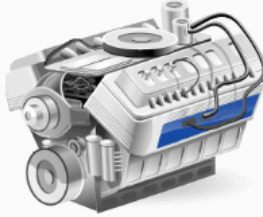
Configuração - Motor

☐ gasóleo

☐ gasolina

☐ híbrido

☐ elétrico



Back

Next

13.2.8 Escolha dos componentes opcionais

Configuração - Componentes opcionais

☐ Componente opcionais

Back

Next

13.2.9 Escolha dos detalhes interiores

Configuração-Detalhes interiores

☐ Estofos

Selecione ▼



☐ Pacotes de luz

Selecione ▼



Back

Next

13.2.10 Escolha dos detalhes exteriores

Configuração-Detalhes exteriores

☐ Vidros escurecidos

Selecione ▼

☐ Teto de abrir

Selecione ▼

☐ Pára-choques

Selecione ▼


Back

Next


13.2.11 Pagamento

Pagamento

☐ Paypal



☐ Referência bancária



☐ Prestações

Done

13.3 Funcionalidades do administrador

Funcionalidades administrador

Autenticar funcionário

Eliminar funcionário

Definir preço da peça

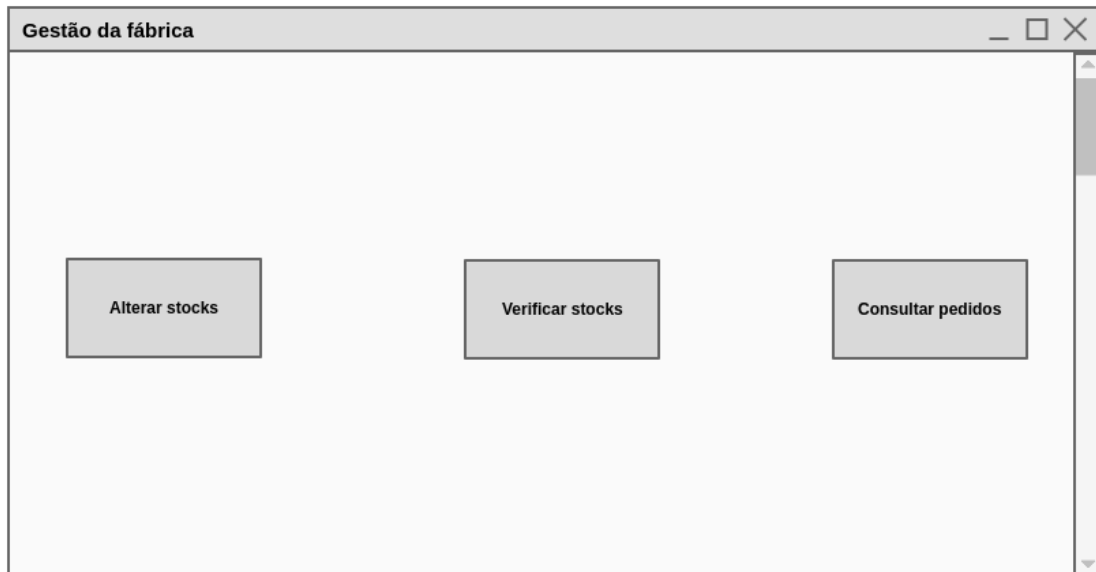
Definir desconto

Consultar funcionários

Alterar configuração

Definir pacote

13.4 Funcionalidades do fabricante



14 Interface final

14.1 Menu Principal



Username:

Password:

14.2 Início da Configuração

14.2.1 Escolha do modelo



Preço:

Veículo:		
Componente Obrigatoria:		
Detalhe Interior:		
Detalhe Exterior:		
Componente Opcional:		

Preço Total:

14.2.Funcionalidades do Administrador



14.2.1 Definir pacote

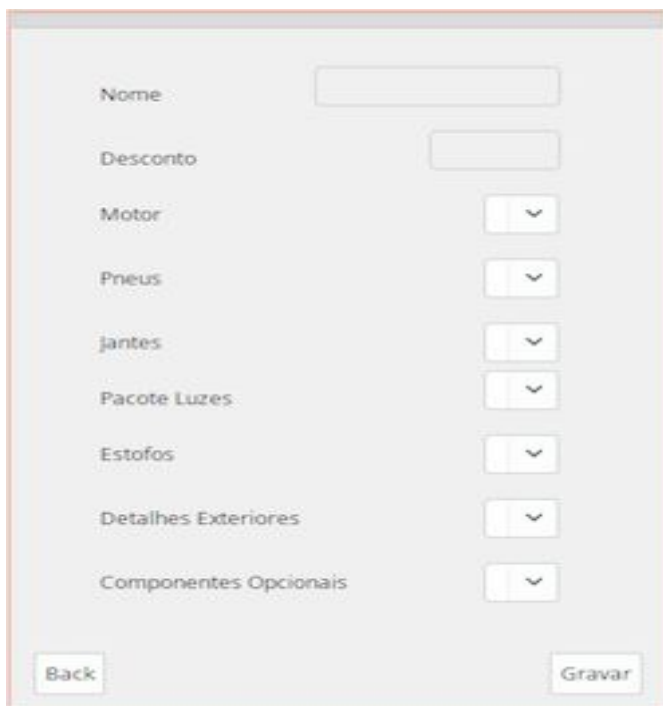


Imagem de uma interface de usuário para o formulário "Definir pacote". O formulário contém os seguintes campos e controles:

- Nome: Campo de texto.
- Desconto: Campo de texto.
- Motor: Menu suspenso.
- Pneus: Menu suspenso.
- Jantes: Menu suspenso.
- Pacote Luzes: Menu suspenso.
- Estofos: Menu suspenso.
- Detalhes Exteriores: Menu suspenso.
- Componentes Opcionais: Menu suspenso.
- Botões: "Back" e "Gravar".

14.2.2 Definir Desconto

A web form titled 'Definir Desconto'. It features a 'Username' label followed by a text input field. To the right of the input field is a 'Pesquisa' button. Below the input field is a large, empty rectangular box. At the bottom left is a 'Back' button, and at the bottom right is an 'Eleminar' button.

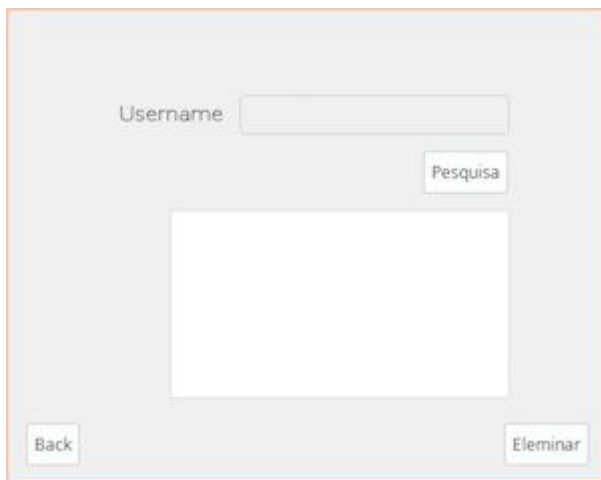
14.2.3 Adicionar Funcionário

A web form titled 'Adicionar Funcionário'. It contains four input fields stacked vertically, each with a label to its left: 'Username', 'Password', 'Numero', and 'Email'. At the bottom left is a 'Back' button, and at the bottom right is a 'Gravar' button.

14.2.4 Consultar Funcionário

A web form titled 'Consultar Funcionário'. It features four input fields at the top, each with a label: 'Numero', 'Email', 'Username', and 'Password'. Below these fields is a large, empty rectangular box. At the bottom left is a 'Back' button.

14.2.5 Eliminar Funcionário



A web form for deleting a user. It features a text input field labeled "Username" at the top left. To its right is a button labeled "Pesquisa". Below the "Username" field is a large, empty rectangular box. At the bottom left is a button labeled "Back", and at the bottom right is a button labeled "Eliminar".

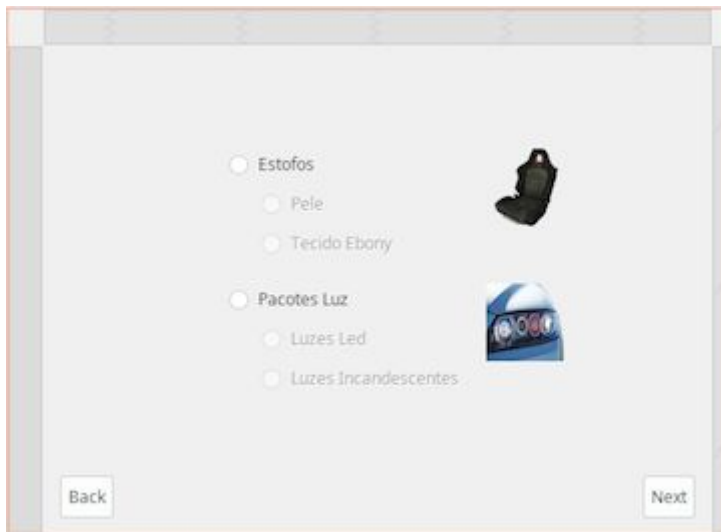
14.2.6 Adicionar Cliente



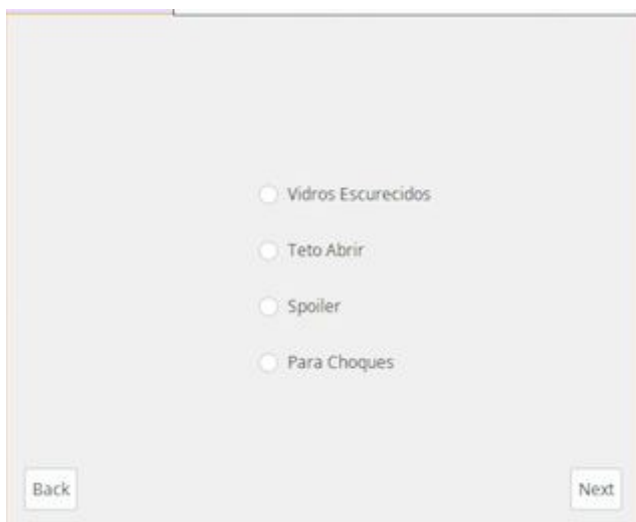
A web form for adding a client. It contains several input fields: "Nome" (Name), "Email", "Nif", and "Morada" (Address). Each field is preceded by a small vertical bar. Below these fields are two buttons: "Back" on the left and "Gravar" (Save) on the right. The form has a light gray background with a subtle border.

14.3 Funcionalidades Funcionário

14.3.1 Detalhe Interior




14.3.2 Detalhe Exterior




14.3.3 Jantes

☐




Foose Classics F226 10-Spoke
Disponível: 50.8cm
2000€

☐




Weds Sport Carbon Fiber Wheel
Disponível: 48.26cm
5000€

☐



BBS Forged Monobloc Ferrari F1
Disponível: 48.26cm e 50.8cm
1500€

☐



Ground Force GF9
Disponível: 48.26cm e 50.8cm
1000€

Back

14.3.4 Motor



☐ Elétrico

Consumo(L/100km): 0

Preço(€): 20000

Emissões(g/km): 0

☐ Híbrido

Consumo(L/100km): 10

Preço(€): 10000

Emissões(g/km): 190

☐ Gasoleo

Consumo(L/100km): 4.8

Preço(€): 7500

Emissões(g/km): 126

☐ Gasolina

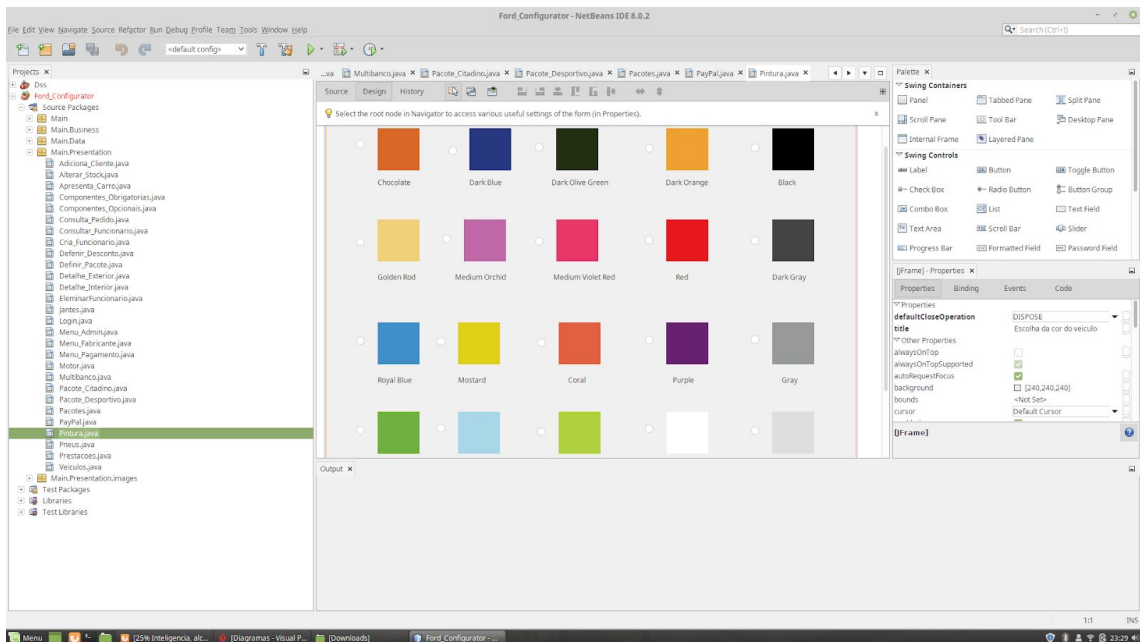
Consumo(L/100km):

Preço(€):

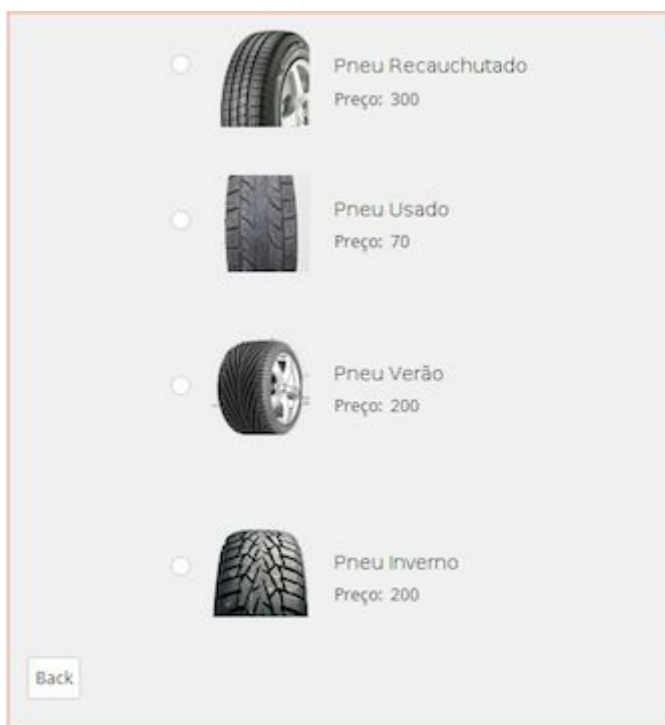
Emissões(g/km):

Back

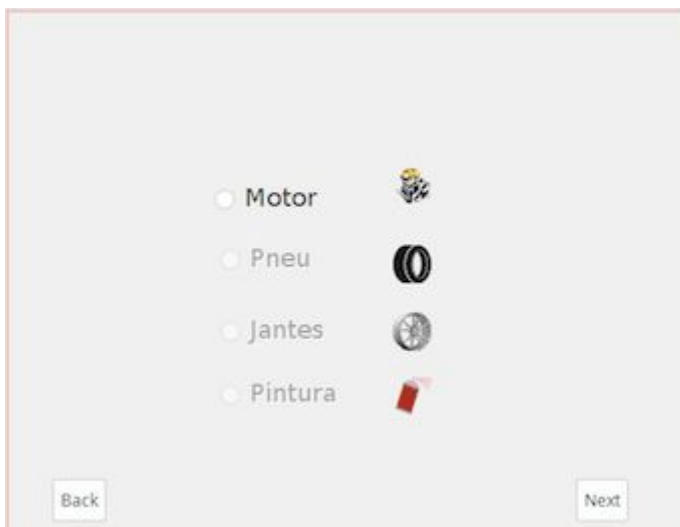
14.3.5 Pintura



14.3.6 Pneus



14.3.7 Componentes Obrigatórias



☐ Motor

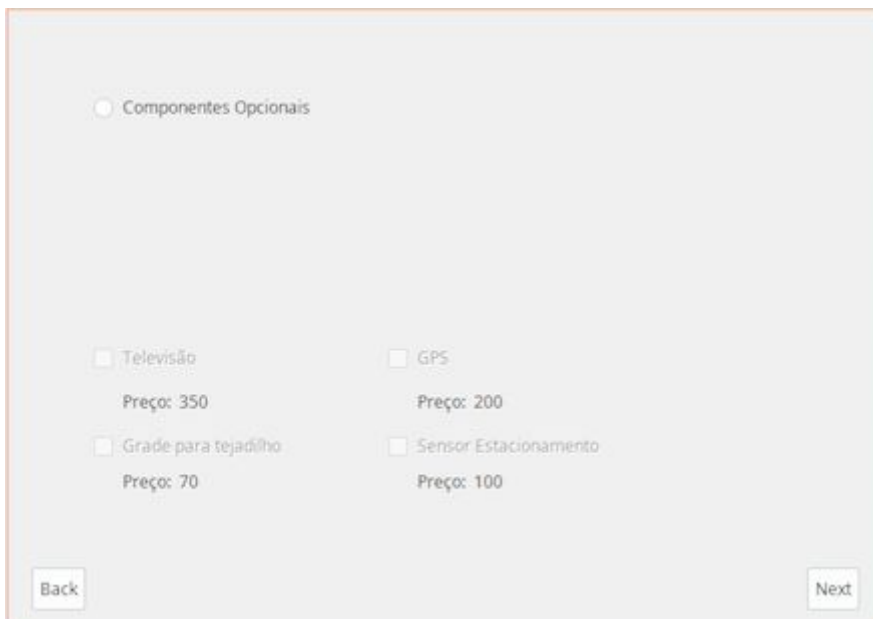
☐ Pneu

☐ Jantes

☐ Pintura

Back Next

14.3.8 Componentes Opcionais



☐ Componentes Opcionais

☐ Televisão
Preço: 350

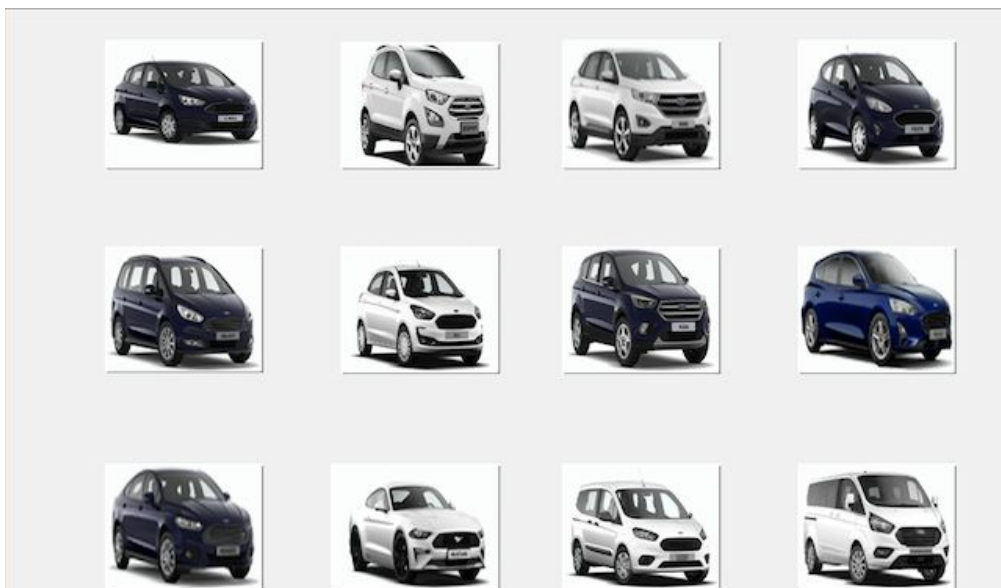
☐ Grade para tejadilho
Preço: 70

☐ GPS
Preço: 200

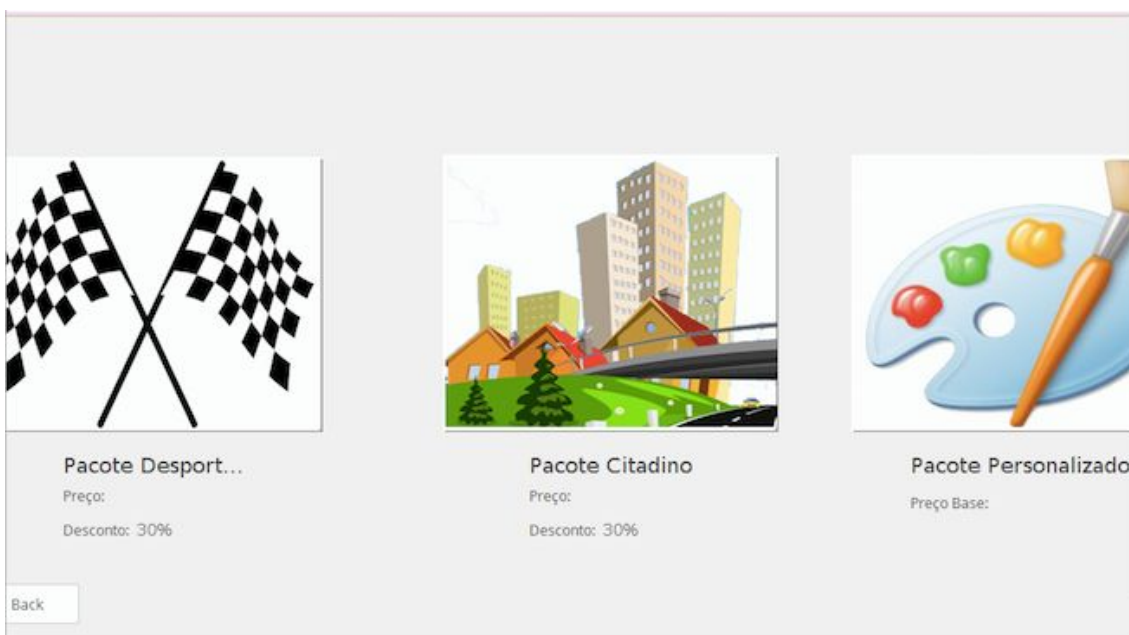
☐ Sensor Estacionamento
Preço: 100

Back Next

14.3.9 Veiculos



14.4 Menu Pacotes



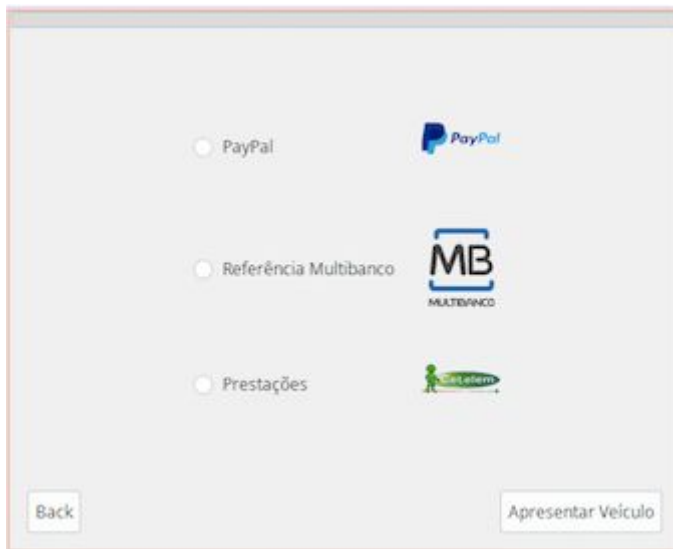
14.4.1 Pacote Citadino

Motor:	Motor a electrico 80 Cavalos de potência Autonomia para 120km		
Jantes:	Ground Force GF9		
Pneus:	Pneus Michelin recauchutados		
Estofos:	Tecido Ebony		
Pacote Luzes:	Incandescente		
spoiler:	Não	Teto de Abrir:	Não
Vidros Escurecidos:	Não	Para Choques:	Sim
Cor	Preço do Pacote: 25550		

14.4.2 Pacote Desportivo

Motor:	Motor a electrico 80 Cavalos de potência Autonomia para 120km		
Jantes:	Ground Force GF9		
Pneus:	Pneus Michelin recauchutados		
Estofos:	Tecido Ebony		
Pacote Luzes:	Incandescente		
spoiler:	Não	Teto de Abrir:	Não
Vidros Escurecidos:	Não	Para Choques:	Sim
Cor	Preço do Pacote: 25550		

14.5 Menu Pagamento



☐ PayPal

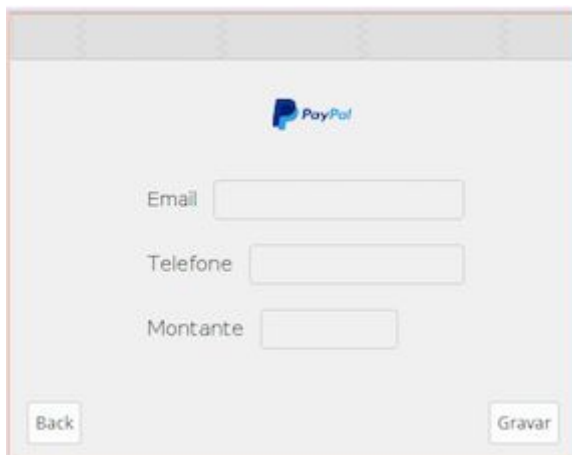
☐ Referência Multibanco

☐ Prestações

Back

Apresentar Veículo

14.5.1 Paypal



PayPal

Email

Telefone

Montante

Back

Gravar

14.5.2 Multibanco



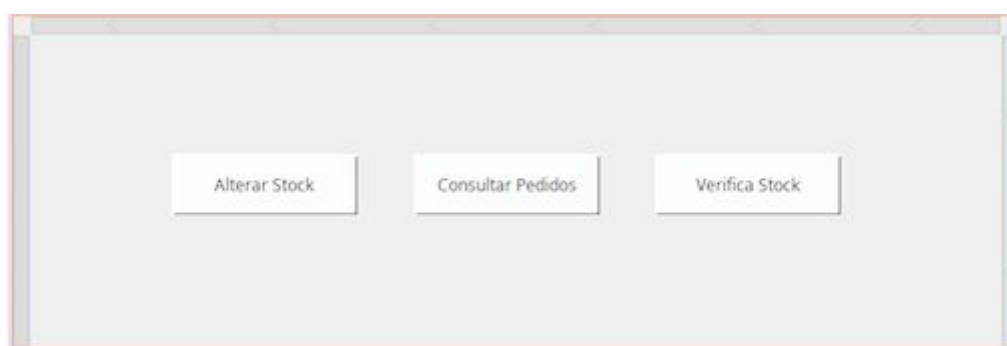
The interface for the Multibanco system. It features the MB logo at the top center. Below the logo, there are two input fields: "Referência" and "Montante". At the bottom left is a "Back" button, and at the bottom right is a "Gravar" button.

14.5.3 Prestações



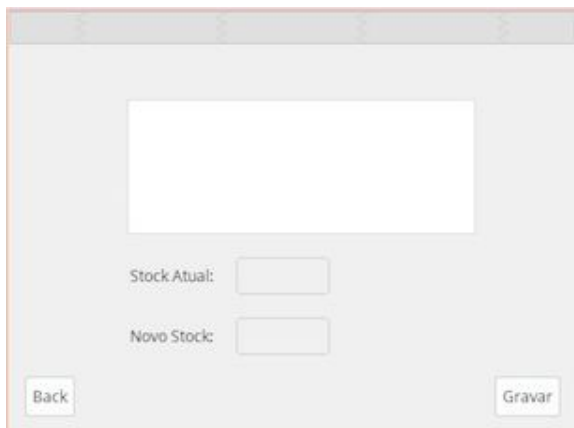
The interface for the Prestações system. It features a logo at the top center. Below the logo, there are two input fields: "Montante" and "Mensalidade". Below these fields is a horizontal timeline with tick marks and numbers: 24, 36, 48, 60, 72, 84, 96, 108, 120. The number 72 is highlighted with a yellow box. Below the timeline, there are labels: "TAEG 5.116%", "TAN 5%", "MTIC", and "valor". At the bottom left is a "Back" button, in the center is a "calcular" button, and at the bottom right is a "Gravar" button.

14.6 Menu Fabricante



The interface for the Menu Fabricante. It features three buttons arranged horizontally: "Alterar Stock", "Consultar Pedidos", and "Verifica Stock".

14.6.1 Alterar Stock



Stock Atual:

Novo Stock:

Back Gravar

14.6.2 Consultar Pedido

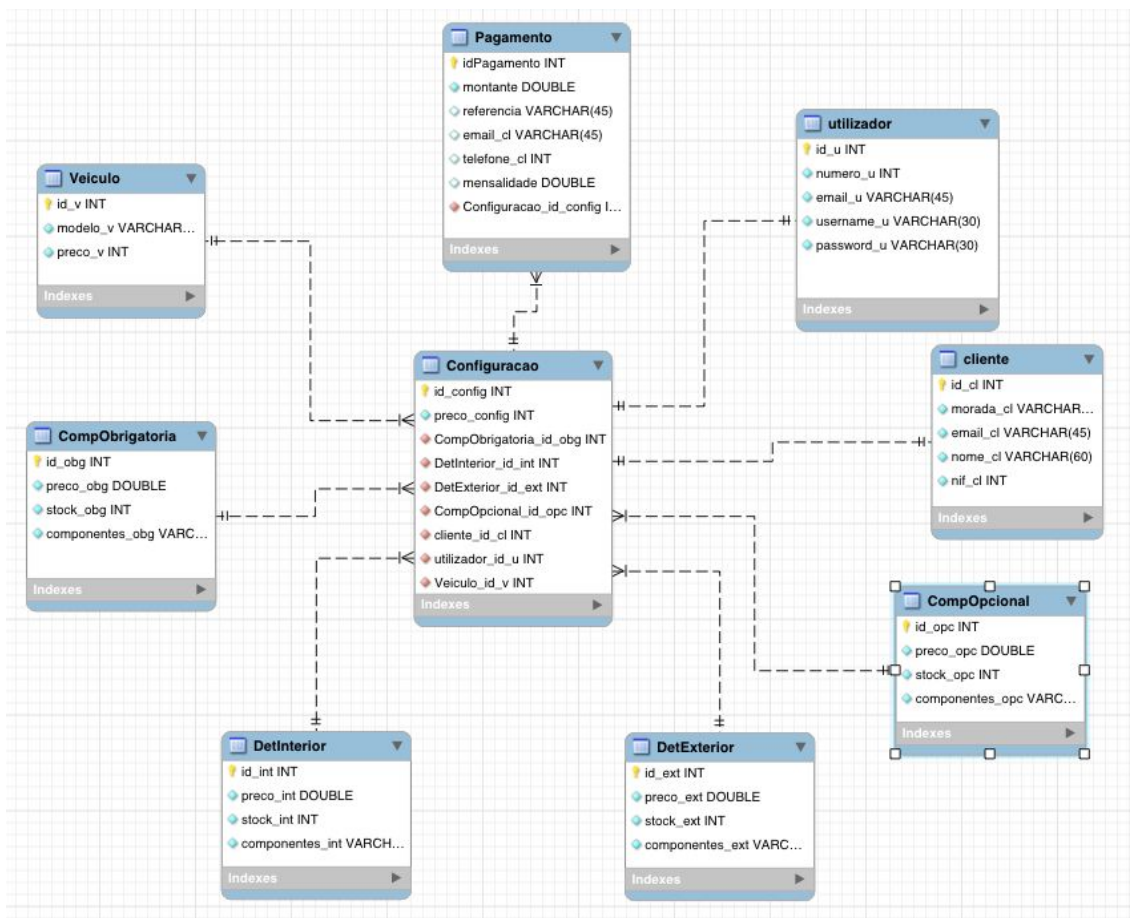


ID	Veiculo	Motor	Jantes	Pneus	Cor	Pacote Luz	Estofos	Extras

Back

15 Base de dados Implementada

Ao realizarmos a nossa aplicação tivemos a necessidade de implementar uma Base de Dados de forma a conseguirmos guardar todos os processos realizados na nossa aplicação de modo a não os perdermos quando desligamos a aplicação. De seguida é apresentado o Modelo Lógico da Base de Dados implementada.



16 Conceitos consolidados com a realização do Trabalho Prático

Com a realização deste trabalho prático adquirimos e consolidamos os conhecimentos básicos sobre a Linguagem de Modelagem de Unificada (UML), que nos mostrou a maneira standard para o design de um sistema de software e, através dos vários passos de modelação que abrange, guiou-nos no processo de implementação da aplicação Ford. À medida que fomos desenvolvendo o projeto percebemos também a importância de cada passo da UML na idealização da aplicação, ou seja, cada um desses passos permitiu-nos implementar/melhorar aspetos que pretendemos ter na aplicação final. Passos esses que se centram nos seguintes aspetos:

- Estudo do domínio do nosso problema.
- Modelação do nosso problema.
- Uso de Use Cases para termos uma noção mais clara sobre todas as interações

com os diversos utilizadores que a nossa aplicação irá ter.

- Uso de diagramas de sequência serviu para aproximarmos os Use Cases da implementação final
- Ao usarmos os diagramas de sequência de subsistemas estamos a adquirir uma melhor percepção de como irá ser feita a implementação do código nas diferentes camadas.
- Implementação do código por camadas.
- Uso de diagramas de package serviu para conseguirmos efetuar uma melhor gestão das diversas classes.
- Implementação de DAOs, objetos de conexão a uma base de dados com o suporte de JDBC.
- Aquisição de conhecimentos de java swing que foi utilizada para programarmos a nossa interface gráfica.

17 Conclusão

Numa primeira fase do trabalho foi efetuada uma análise de requisitos, onde foi analisado tudo sobre o enunciado proposto para termos uma noção de como iria funcionar a aplicação que foi essencial para a construção do Modelo de Domínio. Após a definição e consolidação do modelo foram abordados os requisitos funcionais tendo depois sido feito o Diagrama de Use Case, no qual foram identificados os vários atores que vão interagir com o nosso sistema e também aquilo que pretendemos que cada um destes faça. Posteriormente, foram especificados cada um dos Use Case de forma a serem entendidos os vários fluxos de controlo que existirão aquando da interação do nosso sistema com um utilizador.

Como final da primeira fase fizemos ainda alguns mockups da interface da nossa aplicação para termos uma ideia.

Na segunda fase do trabalho continuamos com as etapas sequenciais de forma a que a aplicação possa gerir as encomendas de veículos. Deste modo, desenvolvemos os diagramas de sequência para os Use Case criados na primeira parte, o que nos permitiu observar as interações entre o ator (Funcionário) e o sistema relativamente a cada operação que ocorre no sistema. Após isto, identificámos as possíveis entidades de se tornarem classes e com as mesmas foi desenvolvido o diagrama de classes relativo à Ford.

Depois das classes todas definidas foi possível definir subsistemas (Utilizador, Encomenda e Fábrica) através da sua representação em diagramas de packages, nos quais distribuámos as classes criadas em 3 partes lógicas: camada de interface (presentation layer), camada lógica do negócio (business layer) e a camada das operações de acesso a dados (data access). A identificação dos vários subsistemas permitiu-nos o desenvolvimento dos diagramas de sequência de subsistemas que nos deu uma visão do que estava a ocorrer no sistema quando acontecia uma interação com ele.

De modo a aumentar o nível de detalhe, foram construídos os diagramas de sequência de implementação onde especificamos as interações com o sistema, tendo por isso mostrado as operações de tais interações quase ao nível da implementação do código.

A topologia de hardware que servirá de apoio à nossa aplicação permitiu-nos a realização do respetivo diagrama de instalação para a Ford.

Relativamente à interface gráfica, o foco foi sempre em torná-la funcional e apelativa para o utilizador, tornando-a muito mais perceptível e fácil de manipular.

A realização do projeto nem sempre foi linear, foi necessário reescrever a especificação de alguns Use Case para facilitar ou permitir as diversas funcionalidades.

Em suma, a Ford é uma aplicação que efetua um pedido de um cliente, por parte do funcionário, com os respectivos passos para a configuração de um veículo.

18 Anexos

18.1 Scripts MySQL

Serão enviados scripts de criação da base de dados da aplicação de configuração.

18.2 Documentação javadoc

Será enviado na pasta de entrega a documentação do projeto.

