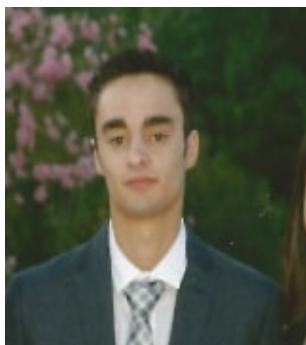


# Relatório do Projeto DSS

Bruno Veloso (a78352)      João Pimentel (a80874)  
Rodolfo Silva (a81716)      Pedro Gonçalves (a82313)

Janeiro 2019



Universidade do Minho  
Desenvolvimento de Sistemas de Software

# Conteúdo

<b>1</b>	<b>Descrição do Enunciado</b>	<b>4</b>
1.1	Configuração . . . . .	4
1.2	Utilização na Fábrica . . . . .	4
<b>2</b>	<b>Objetivos e Principais Decisões Tomadas</b>	<b>5</b>
<b>3</b>	<b>Modelação</b>	<b>6</b>
3.1	Modelo de Domínio . . . . .	6
3.2	Modelo de Use Cases . . . . .	8
3.2.1	Visualizar Histórico de Encomendas . . . . .	9
3.2.2	Adicionar User . . . . .	9
3.2.3	Remover User . . . . .	9
3.2.4	Autenticar-se no Sistema . . . . .	10
3.2.5	Escolher Componentes . . . . .	10
3.2.6	Escolher Pacote . . . . .	10
3.2.7	Escolher Carro Base . . . . .	10
3.2.8	Encomendar Configuração . . . . .	11
3.2.9	Encomendar Configuração Ótima . . . . .	11
3.2.10	Consultar o Stock de Peças . . . . .	11
3.2.11	Fazer Chegar Stock de Peças . . . . .	12
3.3	Máquina de Estado . . . . .	13
3.4	Diagramas de Sequência de Sistema . . . . .	14
3.4.1	Escolher Pacote . . . . .	14
3.4.2	Fazer chegar Stock de Peças . . . . .	14
3.4.3	Escolher Componentes . . . . .	15
3.4.4	Escolher Carro Base . . . . .	16
3.4.5	Encomendar Configuração . . . . .	17
3.4.6	Encomendar Configuração Ótima . . . . .	18
3.5	Diagrama de Sequência com Subsistemas . . . . .	20
3.5.1	Escolher Pacote . . . . .	20
3.5.2	Fazer chegar Stock de Peças . . . . .	20
3.5.3	Escolher Componentes . . . . .	21
3.5.4	Escolher Carro Base . . . . .	22
3.5.5	Encomendar Configuração . . . . .	23
3.6	Diagrama de Packages . . . . .	24
3.7	Diagrama de Classe ORM . . . . .	25
3.8	Diagrama de Sequência de Implementação . . . . .	26
3.8.1	Fazer chegar Stock de Peças . . . . .	26
3.8.2	Escolher Componentes . . . . .	27
3.8.3	Escolher Carro Base . . . . .	28
3.8.4	Encomendar Configuração . . . . .	29

<b>4</b>	<b>Interface Gráfica</b>	<b>31</b>
4.1	Funcionalidades do Programa . . . . .	32
4.1.1	Menu Administrador . . . . .	32
4.1.2	Menu Vendedor . . . . .	32
4.1.3	Operário . . . . .	32
<b>5</b>	<b>Conclusões</b>	<b>33</b>

# 1 Descrição do Enunciado

Nos dias de hoje, a configuração de um automóvel novo é feita utilizando aplicações informáticas disponíveis nos *sites* das marcas. Durante este processo, o utilizador é direcionado por várias etapas onde escolhe os componentes que pretende que o seu novo carro possua.

A aplicação desenvolvida ao longo do semestre permite que um vendedor ajude o respetivo comprador na compra e configuração do seu futuro automóvel. Além disso permite aos operários, responsáveis pela montagem, encomendar peças que necessitem para a construção dos carros, bem como o acesso aos componentes necessários para o fabrico de cada encomenda.

## 1.1 Configuração

Existem duas formas de configurar um novo automóvel, sendo essas a utilização de pacotes pré-definidos pela marca, isto é, configurações que incluem um certo tipo de equipamentos escolhidos pelo fabricante, ou a total personalização do veículo de acordo com as preferências do cliente.

Existe também um modo de configuração que apenas necessita que o cliente escolha as características básicas do automóvel, ou seja, a cor, motorização e o custo máximo do veículo. Feito isto, o sistema recomendará a melhor configuração, tendo em conta o valor máximo que o cliente pretende gastar no automóvel.

## 1.2 Utilização na Fábrica

Esta aplicação será também utilizada pelos funcionários das fábricas onde os automóveis são montados. Permite a estes funcionários o acesso às encomendas em lista de espera, fornecendo a lista de componentes necessários ao fabrico de cada uma e, também, permitirá a requisição de peças necessárias ao bom funcionamento da linha de montagem da fábrica.

## 2 Objetivos e Principais Decisões Tomadas

Os requisitos deste projeto consistiam em gerar uma encomenda de um carro, podendo este possuir um pacote de peças extra ao carro base, bem como um conjunto de extras à escolha. Além disso, deve ser possível gerar uma configuração ótima através de um certo preço máximo dado pelo utilizador. Esta configuração ótima é a configuração com mais peças extra que pode ser gerada dentro do limite monetário. Era, ainda, pedido que fosse possível atualizar o *stock* de peças na fábrica, ou seja, registar a chegada de uma determinada quantidade do componente  $X$  à fábrica.

Tendo os requisitos em conta, as decisões que mais pesaram na execução do projeto foram a identificação das entidades chave, bem como as relações existentes entre si, encontrar a forma mais eficiente para gerir as restrições de cada componente extra, encontrar um pacote numa determinada lista de extras, entre outros.

No que toca às entidades, a ideia passou por imaginar como seria a aplicação, consoante conversas com o cliente. Sendo assim, o problema resolveu-se sozinho, tendo sido claro que, por exemplo, o cliente não teria um papel crucial no processo de encomenda, pois seria o vendedor quem acedia à aplicação.

Relativamente à gestão das restrições, o truque baseou-se na geração das possíveis combinações com o material que estava disponível. Assim, comparando uma a uma, foi simples determinar a lista de componentes e/ou pacote a utilizar consoante os gostos do cliente.

Por fim, de modo a encontrar um pacote no meio de uma lista de peças, era necessário ter forma de aceder a cada um dos pacotes existentes. Sendo assim, bastaria uma procura nos componentes finais da encomenda por todos os elementos do pacote de modo a encontrá-lo.

## 3 Modelação

### 3.1 Modelo de Domínio

Tendo em conta que não é o cliente quem utiliza a aplicação, de modo a encomendar o novo veículo, mas sim o vendedor, tem de existir uma ligação entre estas duas entidades com a encomenda. O comprador faz a encomenda, na medida em que diz ao vendedor o que pretende adquirir. No final, a encomenda tem de ser validada pelo vendedor.

Esta encomenda é relativa a um determinado modelo de um carro. Assim, ao escolher o modelo e marca, os componentes base desse veículo ficam escolhidos, sendo possível adicionar componentes extra.

O utilizador poderá, ainda, optar por adicionar um pacote, ou seja, adicionar um conjunto fixo de extras com um preço total previamente determinado. Caso opte por alterar um ou mais dos componentes do pacote, o desconto passa a ser inexistente e o valor dos extras é a soma dos preços de cada item escolhido.

Além disso, a escolha de um componente está dependente das suas restrições, por exemplo, se o uso do extra  $X$  não permitir a existência de  $Y$  (já no veículo), é dada a opção de resolver o problema, na medida em que se gera a configuração com mais componentes extra dos pretendidos, incluindo os componentes cuja existência é necessária devido a restrições.

No que toca à oficina, um operário tem de conhecer o material referente a uma determinada encomenda, bem como ter a possibilidade de assinalar a chegada de componentes à fábrica.

Relativamente ao pagamento, o vendedor pode efetuar um desconto sobre o valor total da configuração. Por exemplo, seja o total igual a 37000€. Se o vendedor fizer um desconto de 5000€, o preço a pagar é 32000€. No caso de nenhum desconto ser efetuado, este campo toma o valor de 0€.

O modelo de domínio relativo a este projeto encontra-se representado na Figura 1.

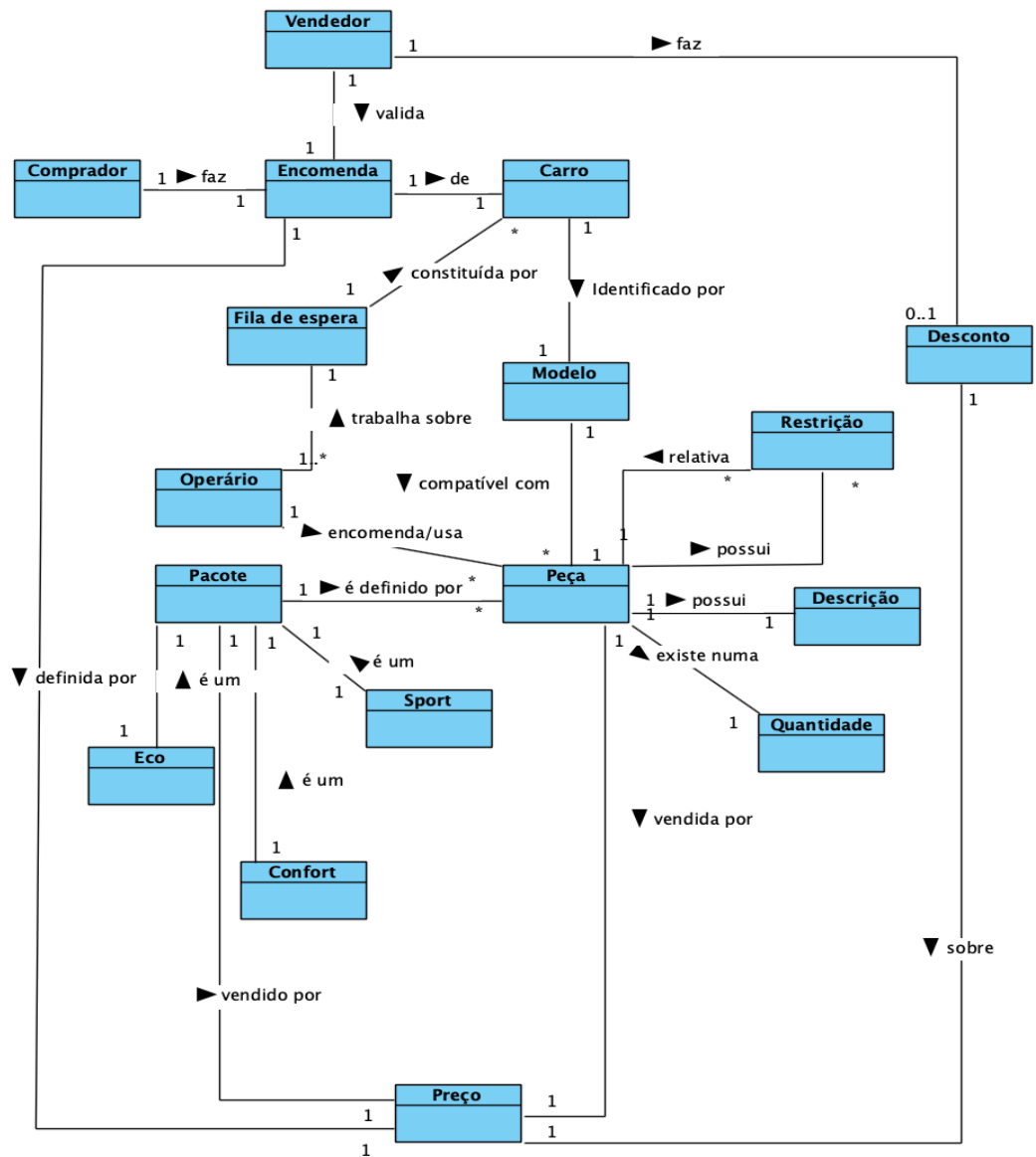


Fig. 1: Diagrama do Modelo de Domínio.

### 3.2 Modelo de Use Cases

Sendo um diagrama de *use case* o local onde se representam todas as funcionalidades proporcionadas pelo sistema, é o local onde a totalidade dos requisitos deve ser representada. Veja-se a Figura 2.

De destacar o facto de os *UCs* "Escolher componentes" e "Escolher pacote" extenderem o "Encomendar configuração". Isto deve-se ao facto de serem comportamentos que nem sempre existem, já que se pode optar por não incluir extras, nem pacotes. Já o *UC* "Escolher carro base" é incluído no tanto em "Encomendar configuração", como em "Encomendar configuração ótima", na medida em que é algo estritamente necessário de modo a atingir a pós-condição dos *UC* em análise.

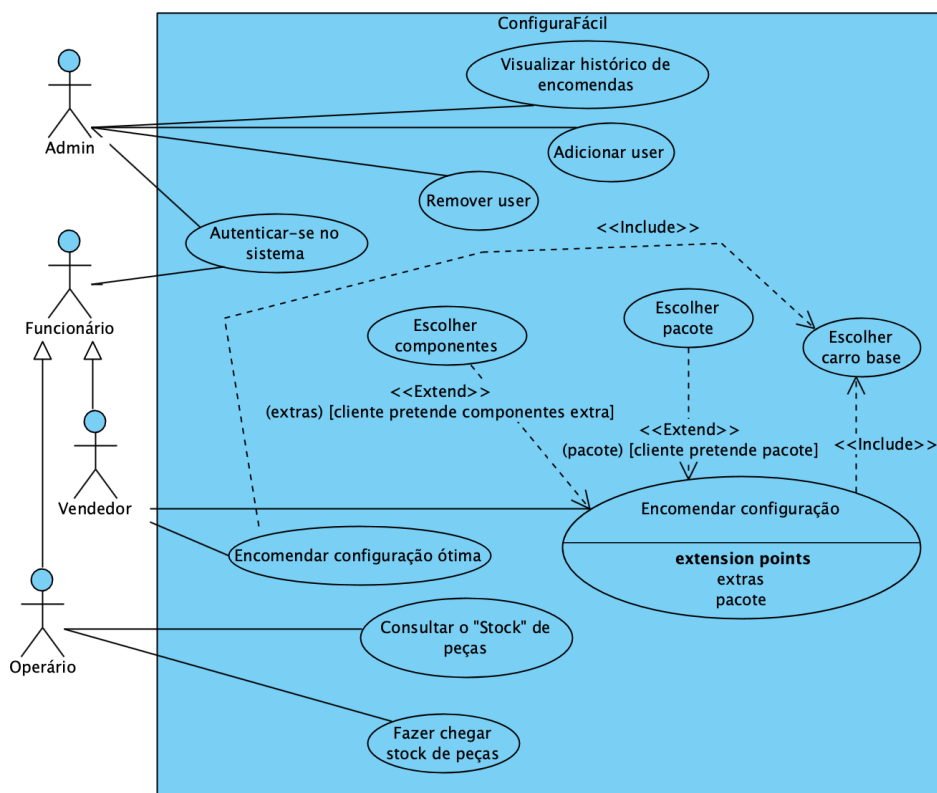


Fig. 2: Diagrama do Use Cases.



### 3.2.1 Visualizar Histórico de Encomendas

Use Case:	Visualizar histórico de encomendas	
Actor:	Admin	
Pré condição:	Estar autenticado no sistema	
Pós condição:	Histórico de encomendas visível	
Cenário Normal	Actor input	System response
	1. Seleciona a opção de visualizar todas encomendas do sistema	2. Mostra todas encomendas na aplicação

### 3.2.2 Adicionar User

Use Case:	Adicionar User	
Actor:	Admin	
Pré condição:	Estar autenticado no sistema	
Pós condição:	Novo utilizador adicionado	
Cenário Normal	Actor input	System response
	1.Introduzir nome do Utilizador	
	2.Introduzir password do Utilizador	
		3.Gerar ID de utilizador
		4.Verificar se ID está disponível
		5.Regista o novo Utilizador
Exceção 1 [ID não está disponível] (passo 4)		6.Encerra Processo
		4.1. Volta a 3

### 3.2.3 Remover User

Use Case:	Remover User	
Actor:	Admin	
Pré condição:	Estar autenticado no sistema	
Pós condição:	User removido	
Cenário Normal	Actor input	System response
	1.Introduzir ID do user	
		2.Verificar se ID existe
		3.Fornecer Informação sobre o user
	4.Confirmar se é o user correto	
Exceção 1 [ID não existe] (passo 2)		5.Apagar user
		6.Fecha processo
Exceção 2 [ID Incorreto] (passo 4)		2.1. Informar que o ID inserido não existe
		2.2. Volta a 1
Exceção 2 [ID Incorreto] (passo 4)	4.1.Indicação de user incorreto	
		4.2. Volta a 1

### 3.2.4 Autenticar-se no Sistema

Use Case:	Autenticar-se no Sistema	
Actor:	Funcionário    Admin	
Pré condição:	Abertura do Software	
Pós condição:	Funcionário tem acesso ao sistema	
Cenário Normal	Actor input	System response
	1.Introduzir identificador do utilizador e password	2.Valida credenciais
		3.Apresenta menu com opções
Exceção 1 [Credenciais Inválidas] (Passo 2)		2.1.Informa que credenciais são Inválidas
		2.2. Fecha aplicação

### 3.2.5 Escolher Componentes

Use Case:	Escolher componentes	
Actor:	Vendedor	
Pré condição:	Configuração base realizada	
Pós condição:	Configuração pronta a validar	
Cenário Normal	Actor input	System response
	2. Seleciona componentes	1. Apresenta componentes possíveis a acrescentar
		3. Verifica compatibilidade e regista 4.continua processo
Comp. Alternativo 1 [Restrições não cumpridas] (passo 3)	3.3 Confirma	3.1 Indica que restrições não estão a ser cumpridas 3.2 Pergunta se quer resolver problemas 3.4 Gera uma configuração com a maior quantidade de peças possível, consoante as escolhidas, e regista 3.5 Volta a 4
Comp. Alternativo 2 [Restrições não cumpridas] (passo 3)	3.3 Recusa	3.1 Indica que restrições não estão a ser cumpridas 3.2 Pergunta se quer resolver problemas 3.4 Não adiciona componentes extra à configuração 3.5 Volta a 4

### 3.2.6 Escolher Pacote

Use Case:	Escolher pacote	
Actor:	Vendedor	
Pré condição:	Carro e modelo escolhidos	
Pós condição:	Pacote adicionado à encomenda	
Cenário Normal	Actor input	System response
	2.Escolhe pacote	1. Apresenta pacotes disponíveis para o modelo
		3. Adiciona pacote à configuração

### 3.2.7 Escolher Carro Base

Use Case:	Escolher carro base	
Actor:	Vendedor	
Pré condição:	Autenticado no sistema	
Pós condição:	Modelo selecionado	
Cenário Normal	Actor input	System response
	2.Escolhe marca	1. Apresenta marcas disponíveis
	4. Escolhe modelo	3. Apresenta modelos disponíveis
	6. Escolhe motorização	5. Apresenta motorizações disponíveis
	8. Escolhe cor	7. Apresenta cores disponíveis
		9. Regista escolhas
		10. Continua configuração

### 3.2.8 Encomendar Configuração

Use Case:	Encomendar configuração	
Actor:	Vendedor	
Pré condição:	Estar autenticado no sistema	
Pós condição:	Processo de encomenda concluído	
Cenário Normal	Actor input	System response
	1. <<include>> Escolher carro base	2. Questiona sobre uso de pacote [ponto de extensão: pacote]
		3. Questiona sobre uso de extras [ponto de extensão: extras]
		4. Apresenta componentes de encomenda
		5. Calcula valor total da encomenda
	6. Indica valor a descontar	
		7. Apresenta valor a pagar
	8. Valida encomenda	
		9. Encomenda adicionada à fila da fábrica
		10. Encerra processo

### 3.2.9 Encomendar Configuração Ótima

Use Case:	Encomendar configuração ótima	
Actor:	Vendedor	
Pré condição:	Estar autenticado no sistema	
Pós condição:	Processo de encomenda concluído	
Cenário Normal	Actor input	System response
	1. <<include>> Escolher carro base	2. Questiona sobre o valor a gastar em extras
	3. Indica valor	4. Gera melhor combinação de pacote/extras soltos consoante o valor
		5. Apresenta componentes de encomenda
	6. Valida encomenda	
		7. Indica valor total de encomenda
	8. Indica valor a descontar	
		9. Apresenta valor a pagar
	10. Regista pagamento	
		11. Encomenda adicionada à fila da fábrica
		12. Encerra processo
Exceção 1 [Encomenda não é válida] (passo 6)	5.1 Indica que encomenda não é válida	5.2. Encerra processo

### 3.2.10 Consultar o Stock de Peças

Use Case:	Consultar o "Stock" de peças	
Actor:	Operário	
Pré condição:	Autenticado no sistema	
Pós condição:	Stock é visível	
Cenário Normal	Actor input	System response
	1. Selecionar opção para consultar stock	2. Sistema apresenta todas as peças, com respetivas quantidades

### 3.2.11 Fazer Chegar Stock de Peças

Use Case:	Fazer chegar stock de peças	
Actor:	Operário	
Pré condição:	Autenticação feita	
Pós condição:	Stock de peças atualizado	
Cenário Normal	Actor input	System response
		1. Pergunta qual a peça
		2. Mostra peças
	3. Escolhe peça	
		4. Pergunta qual a quantidade
	5. Indica qual a quantidade	
		6. Ve se quantidade é válida
Exceção1 [Nao escolhe peça] [Nao escolhe peça] (passo 1)		7. Atualiza novo valor do stock
	1.1. Pergunta qual a peça	
	1.2. Não quer escolher peça (Voltar)	
Comp Alternativo1 [Quantidade invalida] (passo 6)		6.1. Indica quantidade inválida
		6.2. Volta a 4

### 3.3 Máquina de Estado

Uma máquina de estado permite modelar o comportamento de um objeto de forma global, retratando os diferentes estados pelos quais o mesmo passa. Neste caso o foco foi retratar os diferentes estados durante o uso da aplicação em si. Além disso, foi representada a máquina de estado para uma encomenda, retratando os diversos momentos pelos quais o objeto passa. Tenha-se a Figura 3.

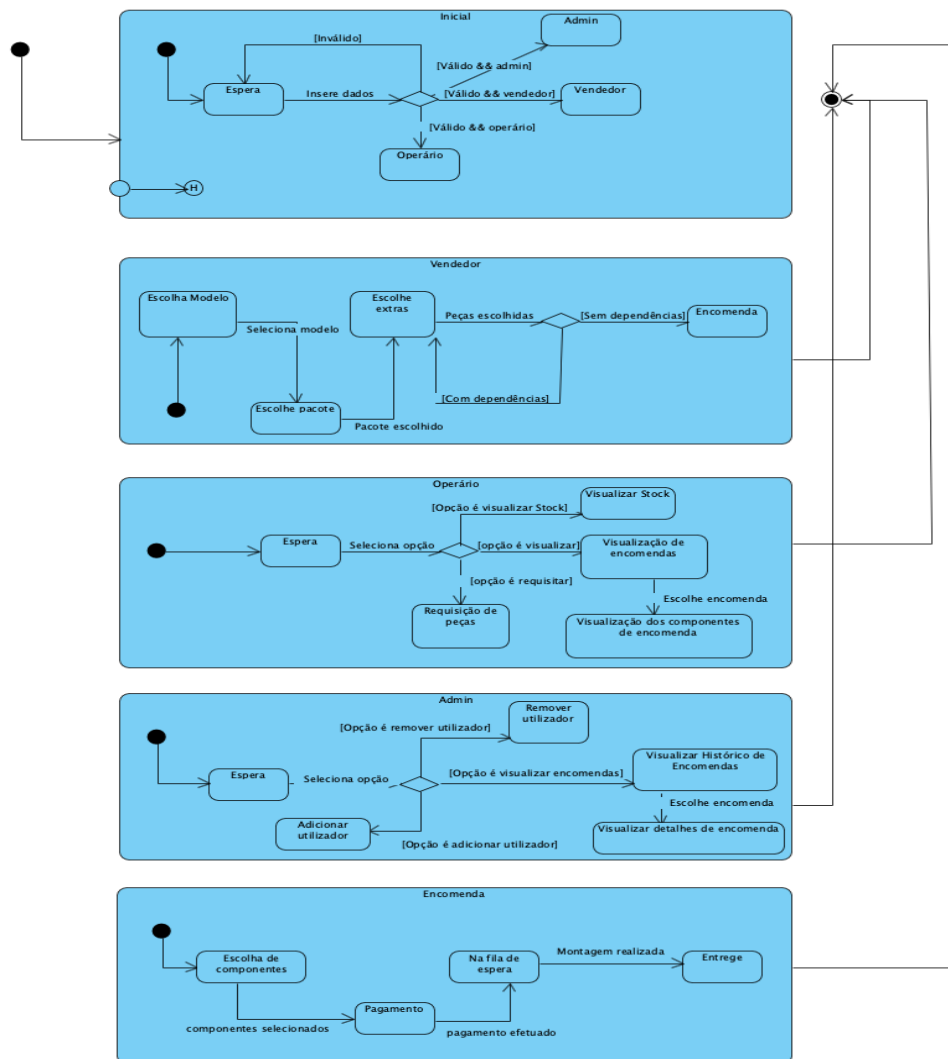


Fig. 3: Máquina de Estados.

### 3.4 Diagramas de Sequência de Sistema

Um Diagrama de Sequência de Sistema representa todas as interações entre objetos através da troca de mensagens entre eles, colocando ênfase na ordenação temporal das mesmas, permitindo compreender a distribuição de responsabilidades de cada objeto.

#### 3.4.1 Escolher Pacote

O DSS que se segue representa as interações entre o sistema, *ConfiguraFacil*, e o vendedor, quando este seleciona um pacote para equipar uma dada configuração de um automóvel.

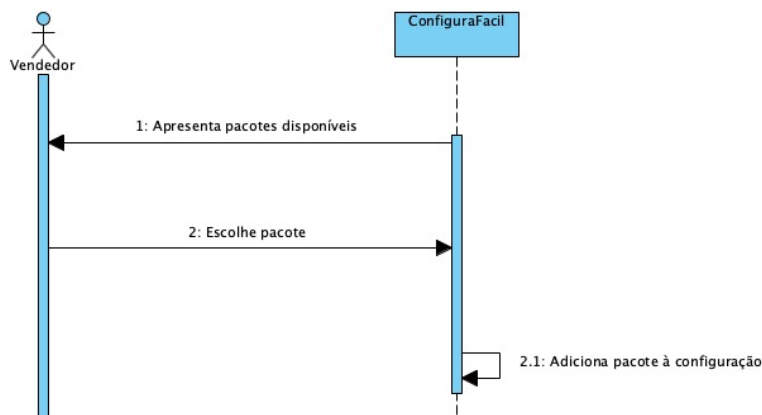


Fig. 4: DSS Escolher Pacote.

#### 3.4.2 Fazer chegar Stock de Peças

O DSS que se segue representa as interações entre o sistema, *ConfiguraFacil*, e o operário, quando este seleciona uma peça para adicionar ao stock.

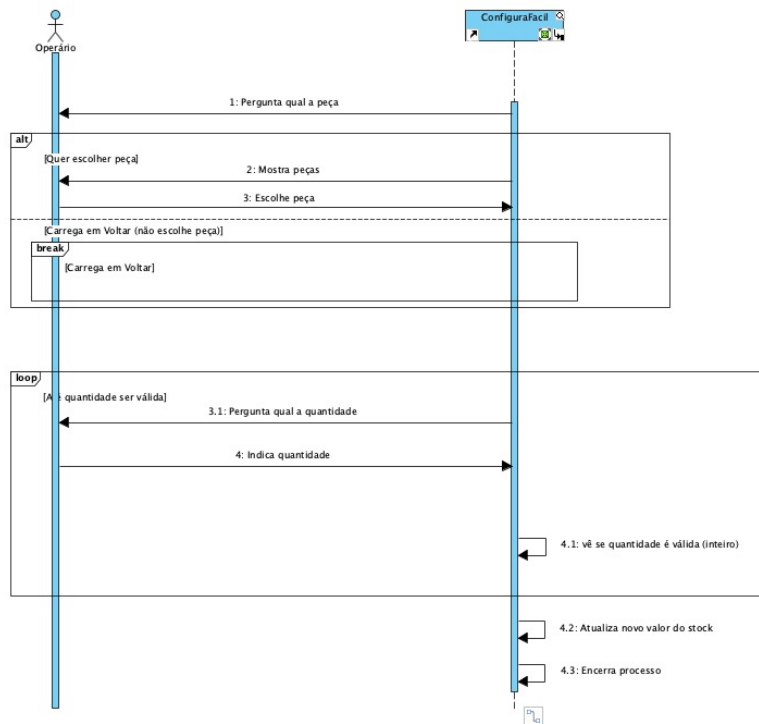


Fig. 5: DSS Fazer chegar Stock de Peças.

### 3.4.3 Escolher Componentes

O DSS que se segue representa as interações entre o sistema, *ConfiguraFacil*, e o vendedor, quando este selecciona os extras para adicionar à configuração do automóvel.

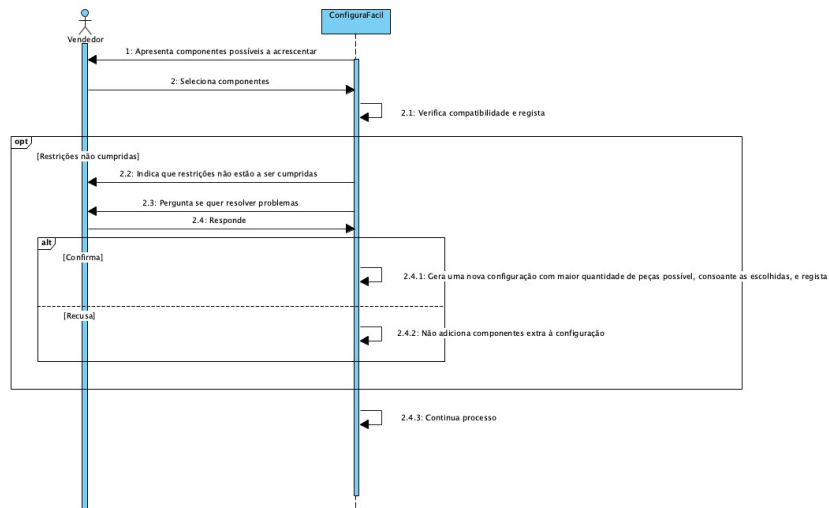


Fig. 6: DSS Escolher Componentes.

#### 3.4.4 Escolher Carro Base

O DSS que se segue representa as interações entre o sistema, *ConfiguraFacil*, e o vendedor, quando este seleciona o carro que será a base da configuração.



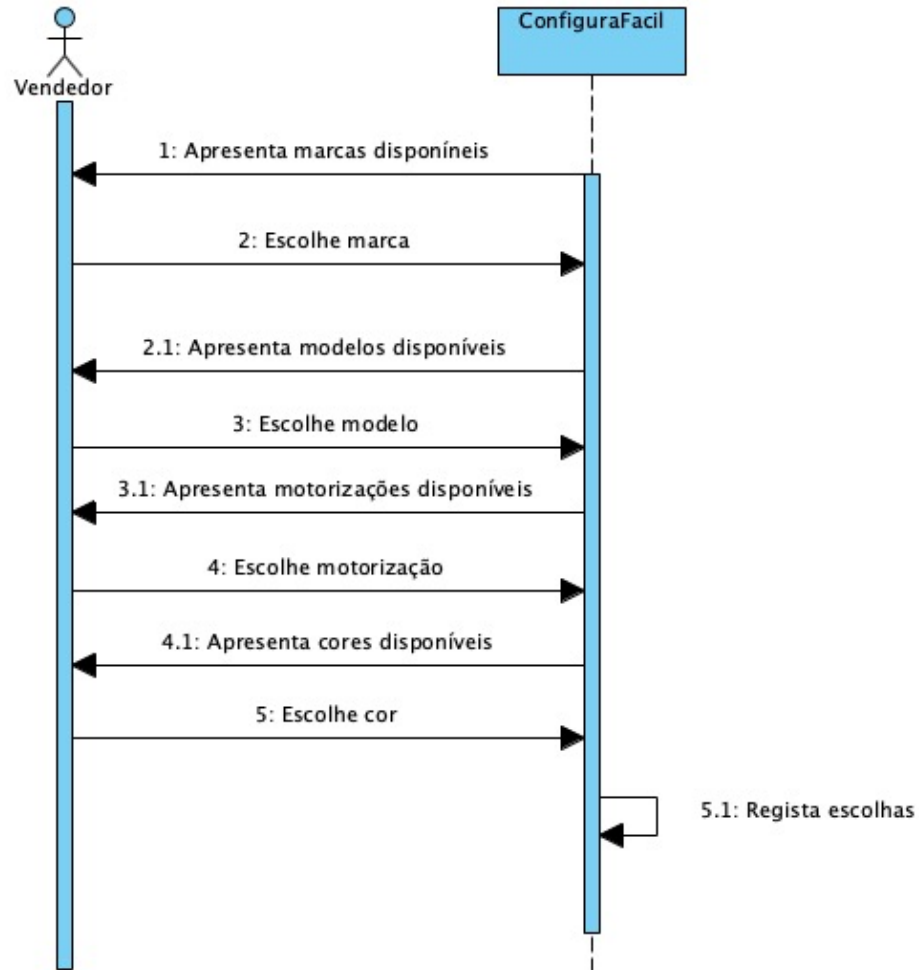


Fig. 7: DSS Escolher Carro Base.

### 3.4.5 Encomendar Configuração

O DSS que se segue representa as interações entre o sistema, *ConfiguraFacil*, e o vendedor, quando este seleciona um pacote para equipar uma dada configuração de um automóvel.

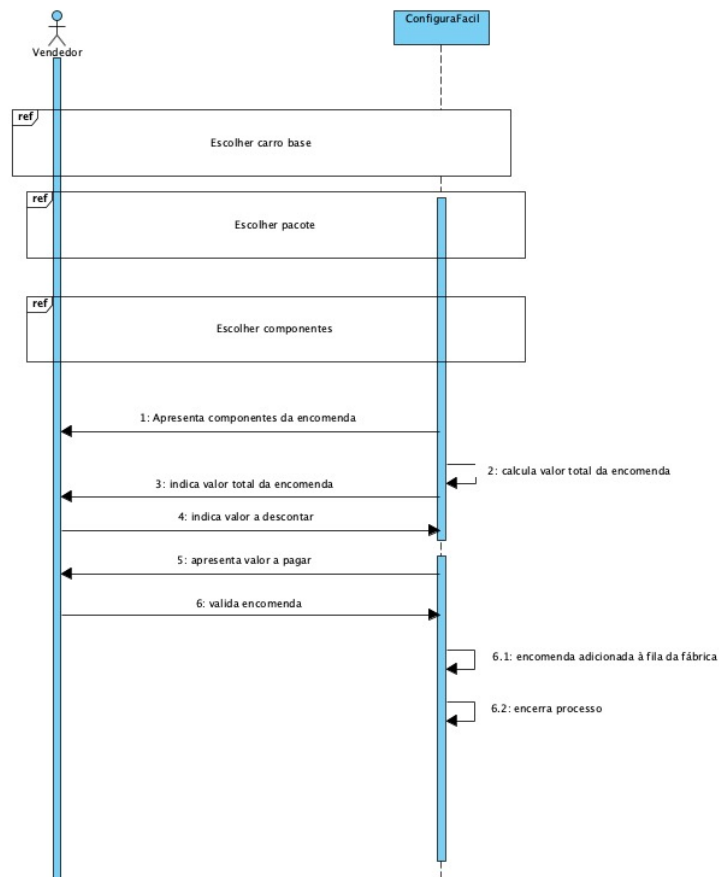


Fig. 8: DSS Encomendar Configuração.

### 3.4.6 Encomendar Configuração Óptima

O DSS que se segue representa as interações entre o sistema, *ConfiguraFacil*, e o vendedor, quando o sistema escolhe automaticamente a configuração óptima de extras para um dado valor.

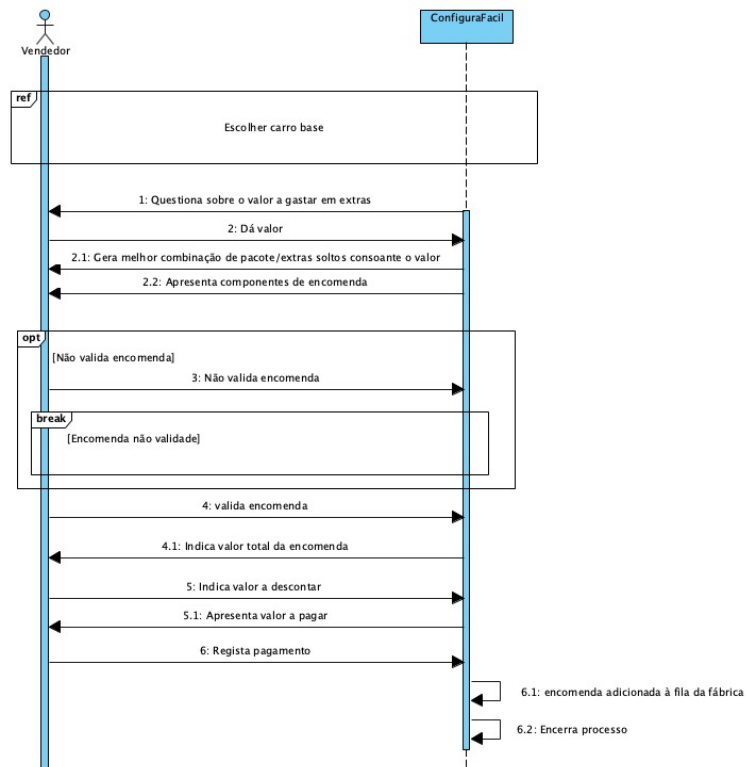


Fig. 9: DSS Encomendar Configuração.

## 3.5 Diagrama de Sequência com Subsistemas

### 3.5.1 Escolher Pacote

O Diagrama de Sequência com Subsistemas que se segue representa as interações entre os vários packages do programa e o vendedor, quando este completa todo o processo de criação de uma encomenda.

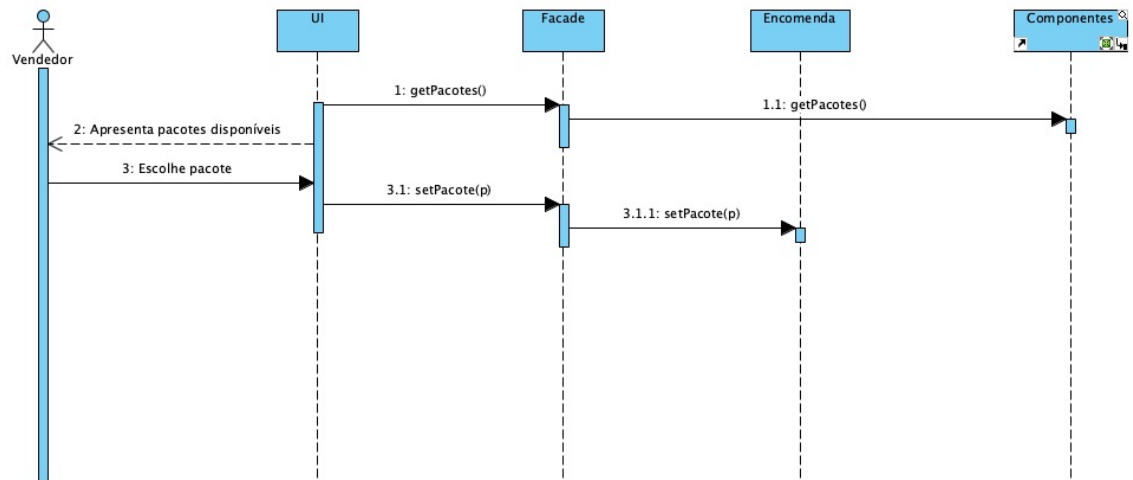


Fig. 10: Diagrama de Sequência com Subsistemas Escolher Pacote.

### 3.5.2 Fazer chegar Stock de Peças

O Diagrama de Sequência com Subsistemas que se segue representa as interações entre os vários packages do programa e o vendedor, quando este completa todo o processo de criação de uma encomenda.

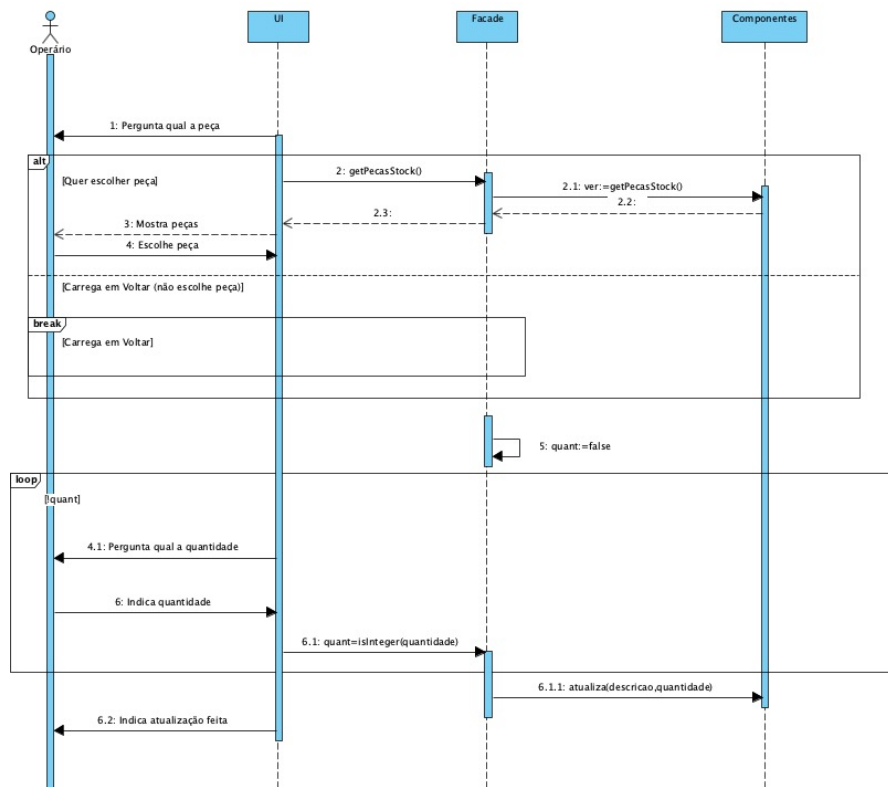


Fig. 11: Diagrama de Sequência com Subsistemas Fazer chegar Stock de Peças.

### 3.5.3 Escolher Componentes

O Diagrama de Sequência com Subsistemas que se segue representa as interações entre os vários packages do programa e o vendedor, quando este seleciona os extras para adicionar à configuração do automóvel.

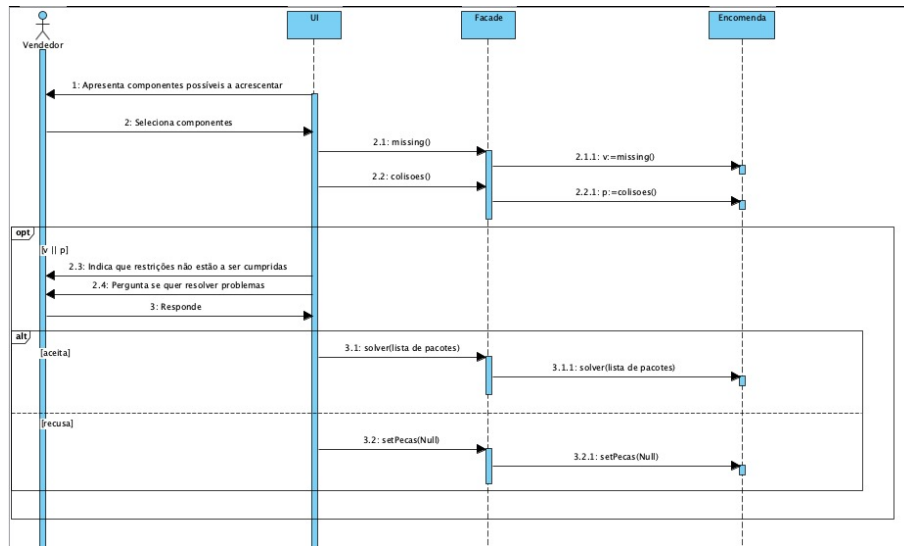


Fig. 12: Diagrama de Sequência com Subsistemas Escolher Componentes.

### 3.5.4 Escolher Carro Base

O Diagrama de Sequência com Subsistemas que se segue representa as interações entre os vários packages do programa e o vendedor, quando este seleciona o carro que será a base da configuração.

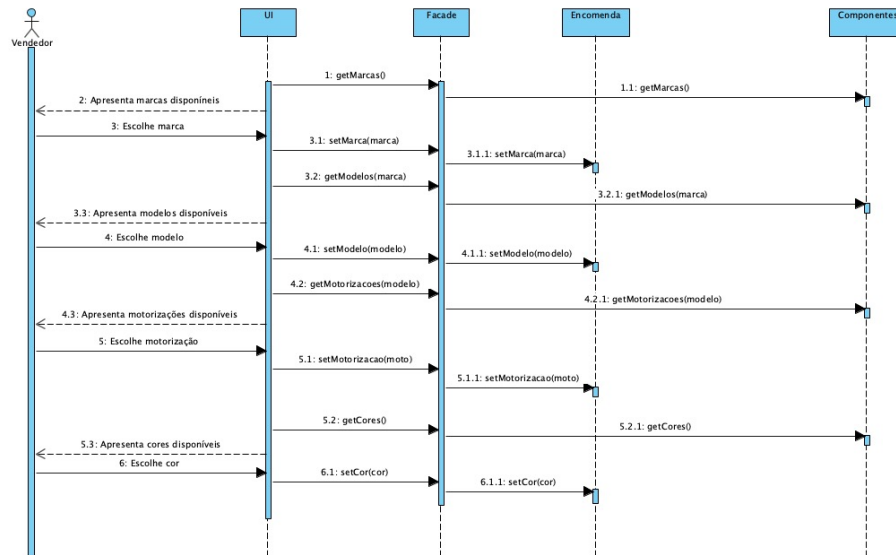


Fig. 13: Diagrama de Sequência com Subsistemas Escolher Carro Base.

### 3.5.5 Encomendar Configuração

O Diagrama de Sequência com Subsistemas que se segue representa as interações entre os vários packages do programa e o vendedor, quando este seleciona um pacote para equipar uma dada configuração de um automóvel.

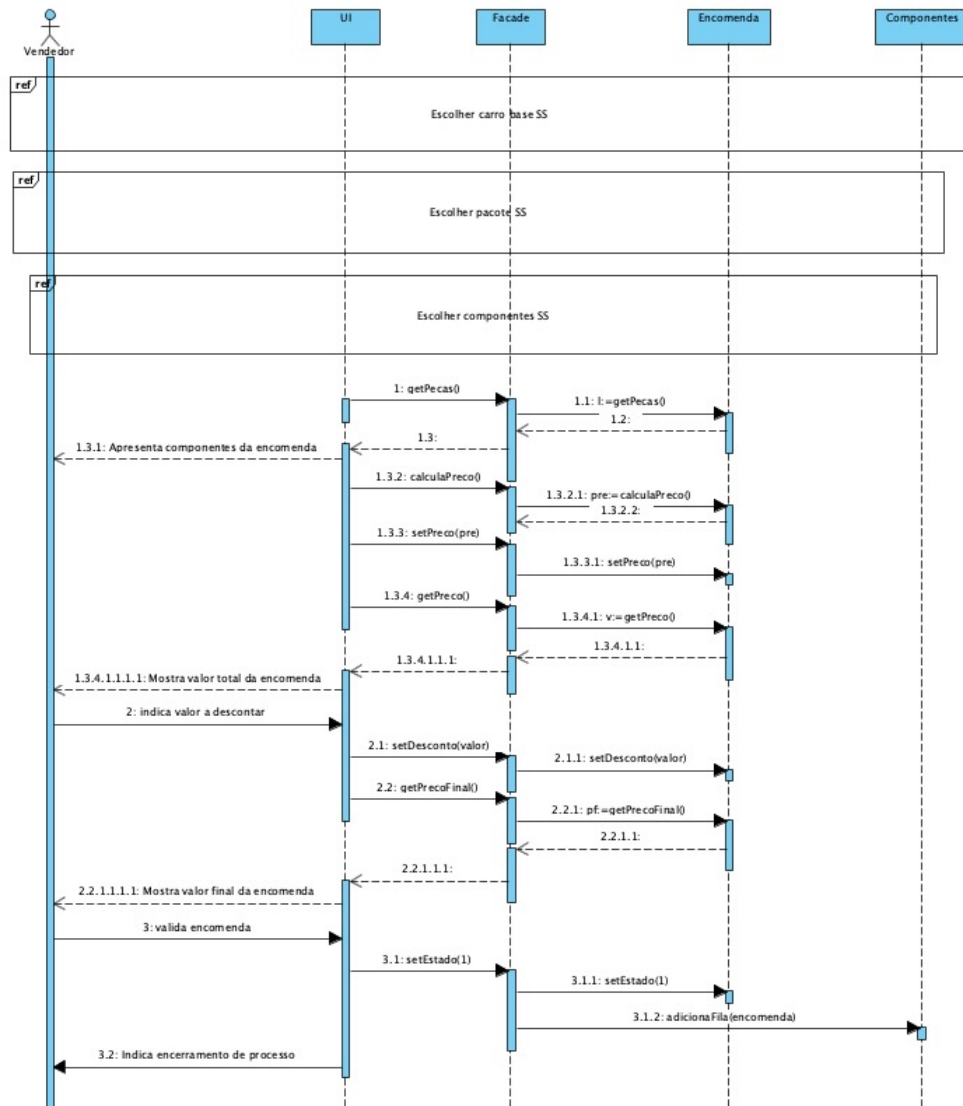


Fig. 14: Diagrama de Sequência com Subsistemas Encomendar Configuração.

### 3.6 Diagrama de Packages

O diagrama de packages representa os packages e as relações estabelecidas por estes. Como se pode observar na figura em baixo, o grupo decidiu dividir o projeto em seis pacotes diferentes.

No package UI, encontra-se todas as classes referentes à Interface Gráfica. No package Utilizadores encontra-se as classes sobre os utilizadores do programa. No package Dados encontra-se as classes referentes à Base de Dados do trabalho. Já no package Encomenda é possível encontrar a classe referente às Encomendas. O package Componentes contém as classes referentes a todas as peças dos carros e também contém a classe Fábrica, onde estes vão ser produzido.

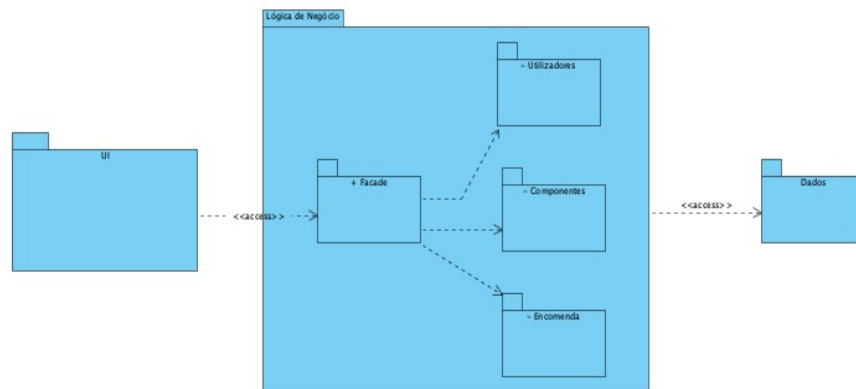


Fig. 15: Diagrama de Packages.



### 3.7 Diagrama de Classe ORM

Aqui são retratadas as classes que serão utilizadas durante a implementação do projeto, bem como os atributos e relações entre as mesmas.

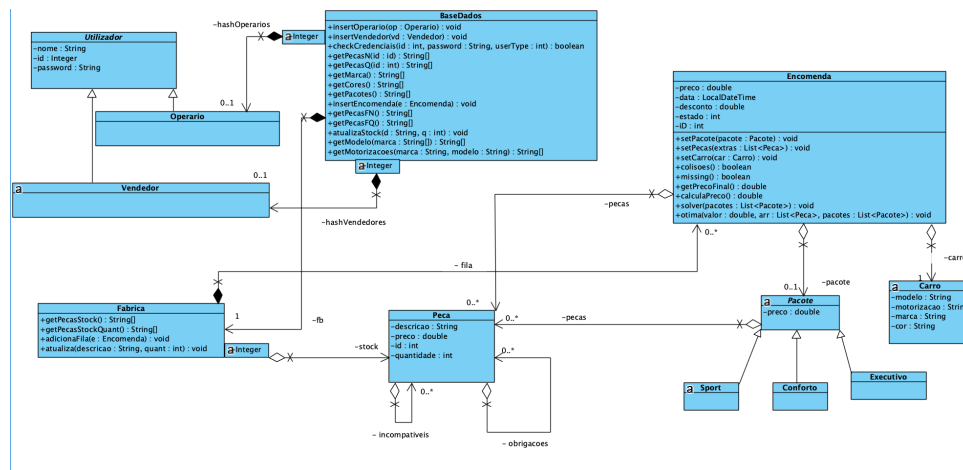


Fig. 15: Diagrama de Classes ORM.

### 3.8 Diagrama de Sequência de Implementação

O Diagrama de Sequência de Implementação que se segue representa as interações entre as várias classes do programa e o vendedor, quando este completa todo o processo de criação de uma encomenda.

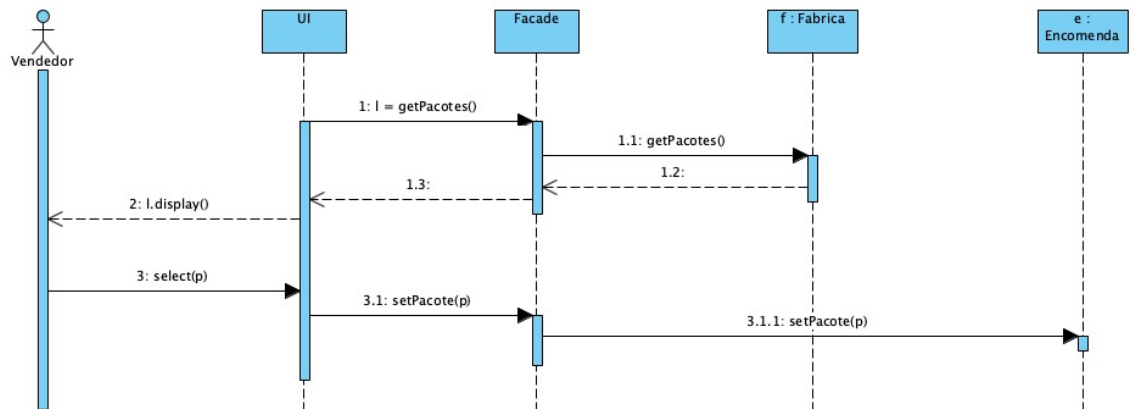


Fig. 16: Diagrama de Sequência de Implementação.

#### 3.8.1 Fazer chegar Stock de Peças

O Diagrama de Sequência de Implementação que se segue representa as interações entre as várias classes do programa e o vendedor, quando este completa todo o processo de criação de uma encomenda.

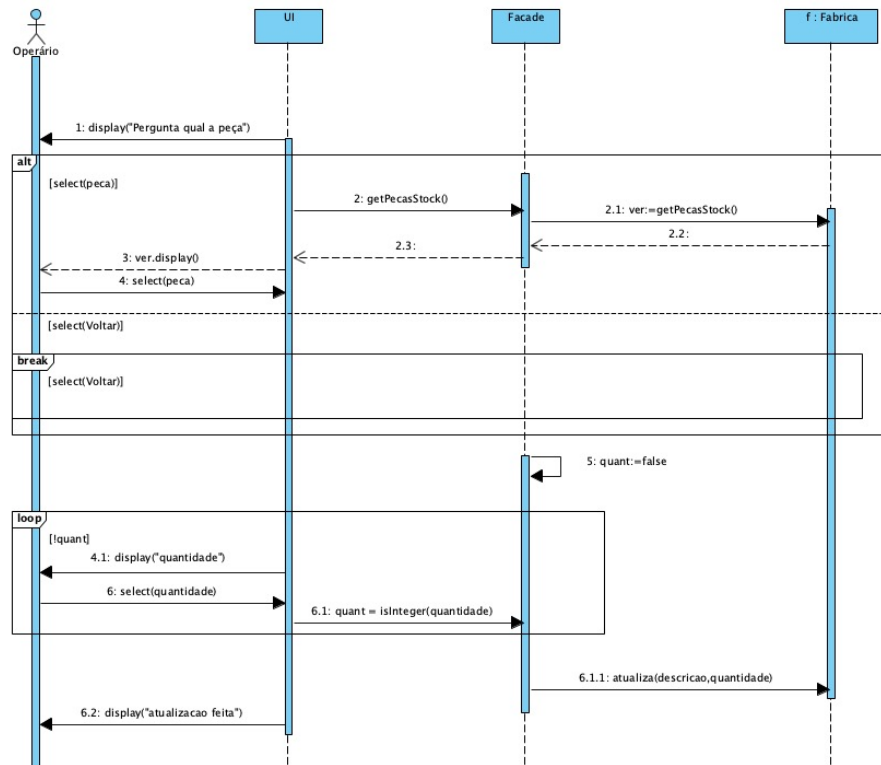


Fig. 17: Diagrama de Sequência de Implementação Fazer chegar Stock de Peças.

### 3.8.2 Escolher Componentes

O Diagrama de Sequência de Implementação que se segue representa as interações entre as várias clases do programa e o vendedor, quando este selecciona os extras para adicionar à configuração do automóvel.

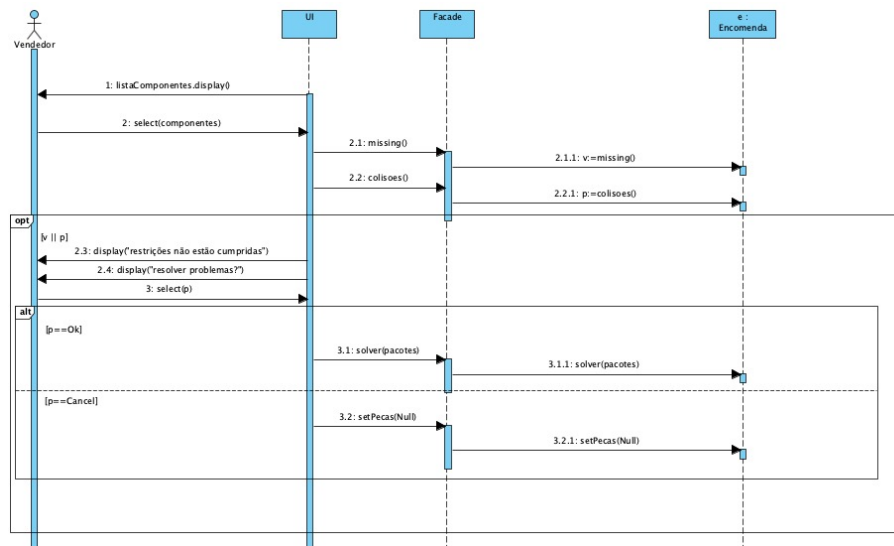


Fig. 18: Diagrama de Sequência de Implementação Escolher Componentes.

### 3.8.3 Escolher Carro Base

O Diagrama de Sequência de Implementação que se segue representa as interações entre as várias classes do programa e o vendedor, quando este seleciona o carro que será a base da configuração.

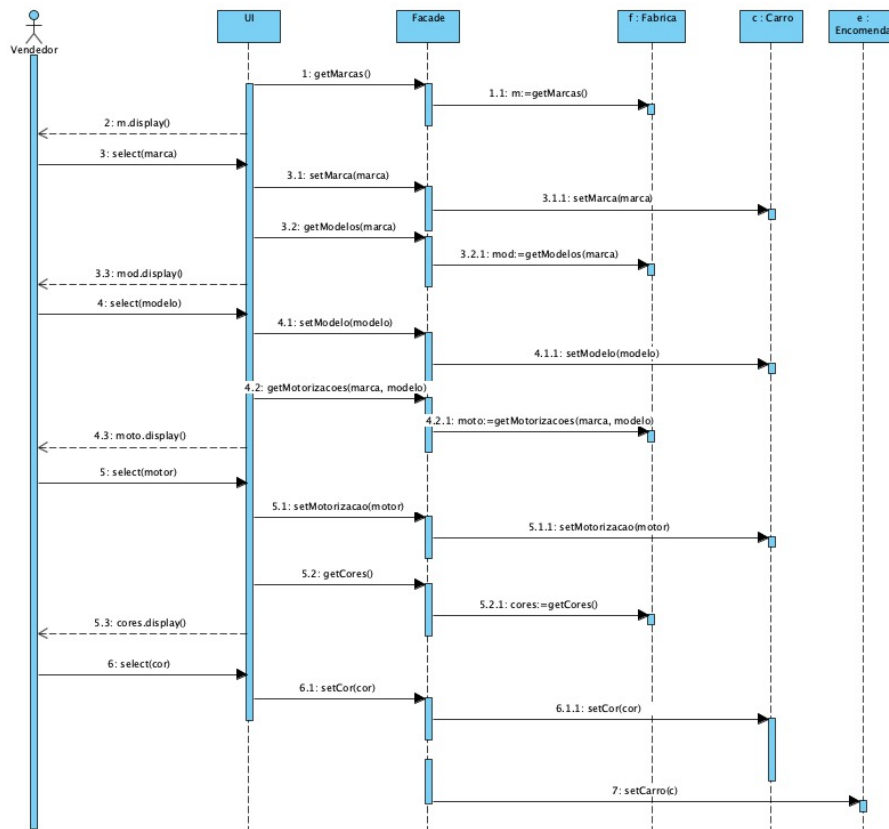


Fig. 19: Diagrama de Sequência de Implementação Escolher Carro Base.

### 3.8.4 Encomendar Configuração

O Diagrama de Sequência de Implementação que se segue representa as interações entre as várias classes do programa e o vendedor, quando este seleciona um pacote para equipar uma dada configuração de um automóvel.

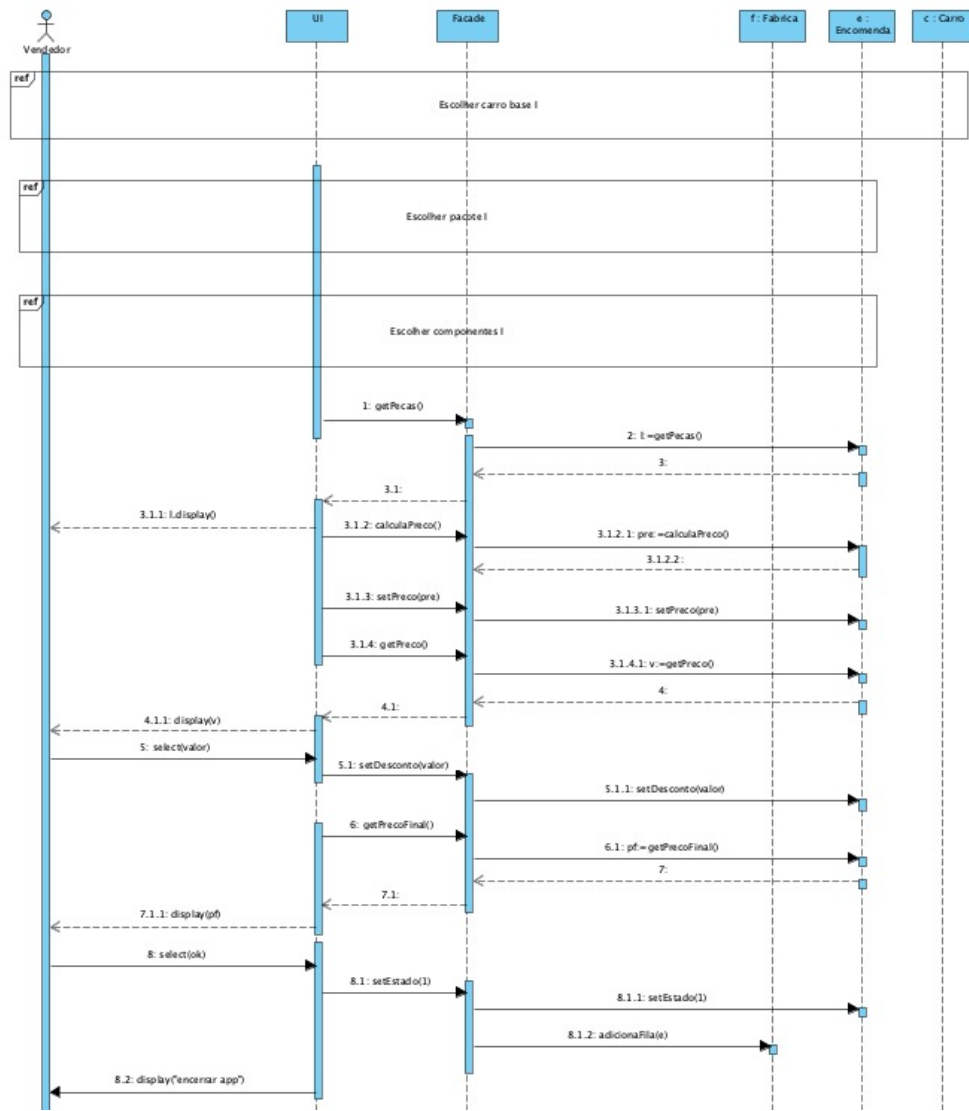
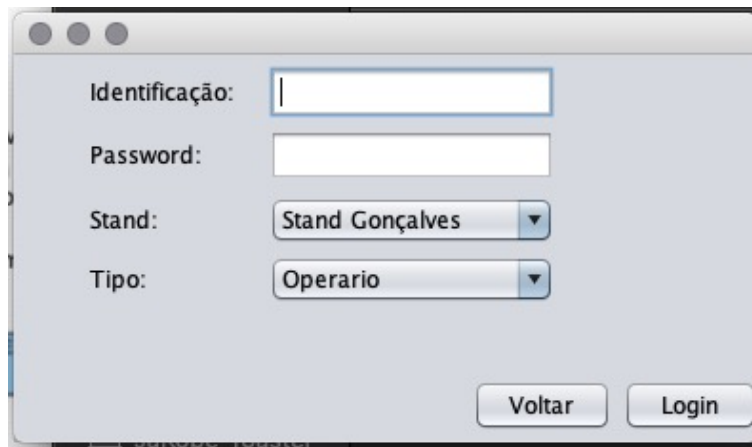


Fig. 20: Diagrama de Sequência de Implementação Encomendar Configuração.

## 4 Interface Gráfica

Este capítulo servirá para explicar o funcionamento do programa e quais as funcionalidades do mesmo.

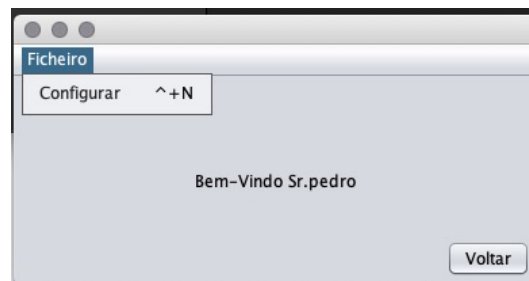
Ao iniciarmos o programa somos confrontados com o menu de *Login*.



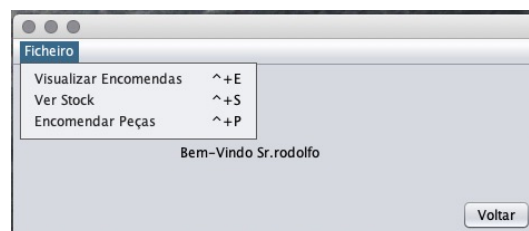
A screenshot of a login window with a light blue background. It contains four labels on the left: 'Identificação:', 'Password:', 'Stand:', and 'Tipo:'. To the right of each label is a corresponding input field: a text box for 'Identificação:', a password box for 'Password:', a dropdown menu for 'Stand:' showing 'Stand Gonçalves', and another dropdown menu for 'Tipo:' showing 'Operario'. At the bottom right, there are two buttons: 'Voltar' and 'Login'.

Depois de iniciarmos sessão podemos ser confrontados com três sub-menus diferentes.

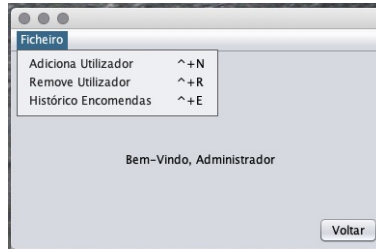
O menu do Vendedor:



O menu do Operário:



O menu do Administrador:



## 4.1 Funcionalidades do Programa

### 4.1.1 Menu Administrador

O menu do Administrador é acedido utilizando como *Identificação* 999 , e como *Password* admin. Neste Menu é possível criar novos Utilizadores, tanto Vendedores como Operários e também visualizar o histórico de encomendas.

### 4.1.2 Menu Vendedor

Neste Menu é configurar um novo carro.

### 4.1.3 Operário

Neste Menu é possível verificar as encomendas que se encontram na lista de espera para fabrico, ver o stock de peças da fábrica e também encomendar peças para repor esse mesmo stock.



## 5 Conclusões

Desenvolvidos todos os modelos necessários ao desenvolvimento do projeto, o grupo considera o trabalho como uma vitória, apesar de existirem aspetos a melhorar, nomeadamente a implementação de um modelo de dados.

Apesar desta falha, todos os diagramas realizados permitiram uma implementação bastante simples da aplicação, tendo sido fácil e rápido detetar possíveis erros que fossem ocorrendo. Além disso, tendo em conta que a modelação dos diagramas obriga a ter a noção de quais as necessidades do programa, não foi perdido tempo a tentar descobrir o que podia ou não ser implementado.

Em suma, o trabalho corresponde aos desafios propostos no enunciado do projeto, apesar de possuir falhas que, num futuro próximo, serão implementadas como forma de solidificação do conhecimento adquirido.