

# UNIVERSIDADE DO MINHO

---

## Relatório de Projeto (Fase 2) - ConfiguraFacil

---

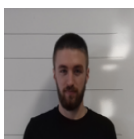
MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

DESENVOLVIMENTO DE SISTEMAS DE SOFTWARE

(1º SEMESTRE - 2018/2019)

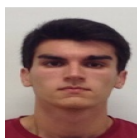
a79799

Gonçalo Nogueira Costeira



a73855

José Lopes Ramos



a73182

Pedro José Marques Terra Sousa



a74264

Rafael Machado da Silva



a79947

Rui Diogo da Silva Costa



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Desenvolvimento</b>	<b>4</b>
2.1	Modelo de Domínio	4
2.1.1	Carro	4
2.1.2	Vendedor	4
2.1.3	Funcionário de Fábrica	4
2.1.4	Administrador	5
2.2	Modelo de Use Case	5
2.2.1	Login	6
2.2.2	Visualizar Encomenda	6
2.2.3	Configuração Ótima	6
2.2.4	Submeter Encomenda	7
2.2.5	Cancelar Encomenda	8
2.2.6	Modificar Encomenda	8
2.2.7	Adicionar Vendedor	9
2.2.8	Remover Vendedor	9
2.2.9	Adicionar Funcionário	9
2.2.10	Atualizar Stock	10
2.2.11	Adicionar Pacote	10
2.2.12	Remover Pacote	11
2.2.13	Adicionar Componente	11
2.2.14	Remover Componente	11
2.3	Diagrama de Estado	12
2.3.1	Diagrama de Estado Inicial	12
2.3.2	Diagrama de Estado Administrador	12
2.3.3	Diagrama de Estado Vendedor	12
2.3.4	Diagrama Estado Configuracao Optima	14
2.4	Diagrama de Sequência	15
2.4.1	Login	16
2.4.2	Administrador	16
2.4.3	Vendedor	17
2.4.4	Funcionário Fabrica	17
2.4.5	Package	18
2.4.6	Componente	18
2.5	Diagrama Package	19
2.6	Diagrama de Classes (com métodos)	20
2.7	Diagrama de Classes com DAO's	20
2.8	Diagrama de Instalação	21
2.9	Diagrama de Atividades	22
2.9.1	Adiciona Componente	22
2.9.2	Adiciona Package	22
2.9.3	Adiciona Utilizador	23
2.9.4	Alterar Encomenda	23
2.9.5	Alterar Package	24
2.9.6	Configura Carro	24
2.9.7	Login	25
2.9.8	Remove Componente	25
2.9.9	Remove Encomenda	26
2.9.10	Remove Package	26
2.9.11	Remove Utilizador	27
2.9.12	Atualiza Stock	27
2.10	Base de Dados	28
<b>3</b>	<b>Conclusão</b>	<b>29</b>

## Lista de Figuras

1	Modelo de Domínio . . . . .	4
2	Modelo de Use Case . . . . .	5
3	Login . . . . .	6
4	Vizualizar Encomenda . . . . .	6
5	Configuração Ótima . . . . .	7
6	Submeter Encomenda . . . . .	7
7	Cancelar Encomenda . . . . .	8
8	MOdificar Encomenda . . . . .	8
9	Adiciona Vendedor . . . . .	9
10	Remove Vendedor . . . . .	9
11	Adiciona Funcionário . . . . .	10
12	Atualiza Stock . . . . .	10
13	Adiciona Package . . . . .	11
14	Remove Package . . . . .	11
15	Adiciona Componente . . . . .	12
16	Remove Componente . . . . .	12
17	Diagrama de Estado Inicial . . . . .	13
18	Diagrama de Estado Administrador . . . . .	13
19	Diagrama de Estado Vendedor . . . . .	13
20	Diagrama de Estado Funcionario Fabrica . . . . .	14
21	Diagrama de Estado Configuracao Optima . . . . .	14
22	Diagrama de Sequencia . . . . .	15
23	Diagrama de Sequencia Login . . . . .	16
24	Diagrama de Sequencia Administrador . . . . .	16
25	Diagrama de Sequencia Vendedor . . . . .	17
26	Diagrama de Sequencia Funcionário Fabrica . . . . .	17
27	Diagrama de Sequencia Package . . . . .	18
28	Diagrama de Sequencia Componente . . . . .	18
29	Diagrama de Package . . . . .	19
30	Diagrama de Classes . . . . .	20
31	Diagrama de Classes com DAO's . . . . .	20
32	Diagrama de Instalação . . . . .	21
33	Adiciona Componente . . . . .	22
34	Adiciona Package . . . . .	22
35	Adiciona Utilizador . . . . .	23
36	Alterar Encomenda . . . . .	23
37	Alterar Encomenda . . . . .	24
38	Configura Carro . . . . .	24
39	Login . . . . .	25
40	Remove Componente . . . . .	25
41	Remove Encomenda . . . . .	26
42	Remove Package . . . . .	26
43	Remove Utilizador . . . . .	27
44	Atualiza Stock . . . . .	27
45	Modelo Lógico da Base de Dados . . . . .	28

# 1 Introdução

Este projeto visa o desenvolvimento de um sistema que permita no stand e junto ao cliente gerar a encomenda de um novo carro. O objetivo deste sistema é que a configuração de um novo carro seja mais fácil e mais rápida de se fazer. A utilização deste sistema é útil pois para além de o cliente poder escolher as componentes que o seu carro vai possuir, mas também para o stand e fábrica que têm mas um controlo do que vão produzir e os componentes que iram ser mais utilizados.

De seguida iremos explicar resumidamente o funcionamento da aplicação referida, e quais os parâmetros mais importantes que a mesma deve obedecer.

Como foi referido acima, este sistema também vai ter um vertente para a fábrica, isto é útil, pois assim há uma maior gestão de que carro há a produzir, e das componentes que precisam de ser encomendadas com mais frequência. Caso um componente esteja sem stock o funcionario da fabrica pode adicionar os elementos em falta.

Por parte do stand, quando o vendedor começa a encomenda do carro, ele irá ter duas opções: se o cliente quiser ter um carro personalizado, ou seja, tendo por base uma configuração básica o próprio cliente escolha as componentes extra que deseja que o carro possua, ou o vendedor irá ter outra opção denotada ConfiguraçãoOptima, em que o cliente tem um Budget para gastar e o sistema escolhe automaticamente as componentes que o carro vai possuir, iremos falar desta opção mais a frente. Também o cliente poderá optar por escolher Packages, o sistema irá possuir três: Sport, Confort e por fim Economic, e o cliente se quiser poderá escolher componentes opcionais, e o sistema terá de verificar se certa componente opcional pode ser escolhida, iremos falar disso a frente.

O sistema deverá ser robusto, isto é, caso uma componente seja escolhida e essa mesma seja incompatível com uma componente já selecionada, o sistema deverá indicar ao utilizador, que existe uma incompatibilidade de componentes, e avisar das escolhas que puderam ser tomadas apartir dessa decisão.

Iremos explicar a decisões tomadas para a construção de cada diagrama.

## 2 Desenvolvimento

### 2.1 Modelo de Domínio

O Modelo de Domínio inclui todas as entidades que pensamos serem necessárias para o nosso projeto: Utilizador, Vendedor, Funcionário de Fabrica, Carro, Configuração, Otimizada, Personalizado, Pacote, Componente, Stock e Administrador.

Para um melhor planeamento, optamos por dividir o nosso modelo em três categorias: Vendedor, Funcionário de Fabrica e Carro.

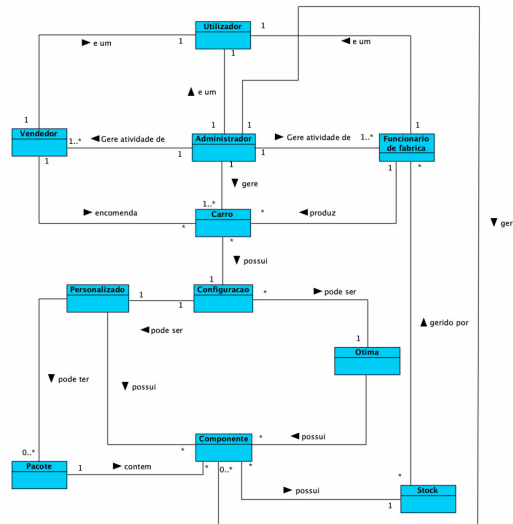


Figura 1: Modelo de Domínio

#### 2.1.1 Carro

As entidades que fazem parte desta categoria são: Carro, Configuração, Personalizado, Componente, Otimizada e Pacote.

As entidades Carro e Configuração, não iremos pronunciarmo-nos muito sobre isto, visto que os nomes das mesmas já se auto explicam. Sabemos que um Carro possui uma Configuração, e que por sua vez essa Configuração pode ser Personalizado ou Otimizada, caso a mesma seja Personalizado, iremos ter duas escolhas em que Personalizado pode ter um ou mais Pacote ou possui vários Componente. Caso seja Otimizada, esta mesma possui vários Componente.

Fizemos esta divisão de Configuração para o Utilizador ter as escolhas que o sistema permite fazer, e todas elas convergem para Componente visto que podemos ter duas opções de configurar um carro mas ambas precisam de componentes para fazer essa operação.

#### 2.1.2 Vendedor

As entidades que fazem parte desta categoria são: Vendedor, Utilizador e Carro.

O Vendedor neste sistema é uma entidade pois é este que faz a encomenda de vários carros, a partir daí vamos para a entidade Carro que é necessária para fazermos este processo, por fim decidimos criar uma entidade chamada Utilizador, pois iremos ter vários utilizadores a atuar no sistema e queríamos diferenciá-los.

#### 2.1.3 Funcionário de Fábrica

As entidades que fazem parte desta categoria são: Funcionário de fabrica, Stock, Utilizador, Carro e Componente.

Nesta categoria reparamos que como um Funcionário de fabrica vai produzir um carro necessitamos também que referir a entidade Carro.

Como na fabrica existe um Stock, este mesmo tem de ser gerido pelo Funcionário de fabrica, e por consequência, Componente possui um Stock.

Como foi referido na secção acima de Vendedor, nos decidimos também usar a entidade Utilizador para termos uma diferenciação.

#### 2.1.4 Administrador

A entidade Administrador vamos falar dela em separado das outras pois a mesma, vai gerir atividade de Vendedor, Funcionario de fabrica, vai também gerir a entidade Carro e de Componente.

Decidimos criar esta entidade visto que o nosso sistema precisava de uma entidade para monitorizar o sistema e as entidades que nele atuam, para deixar sempre o sistema coeso e sem que ocorram erros no mesmo.

Um Administrador é um Utilizador, pelas mesmas razões as entidades referidas em cima.

## 2.2 Modelo de Use Case

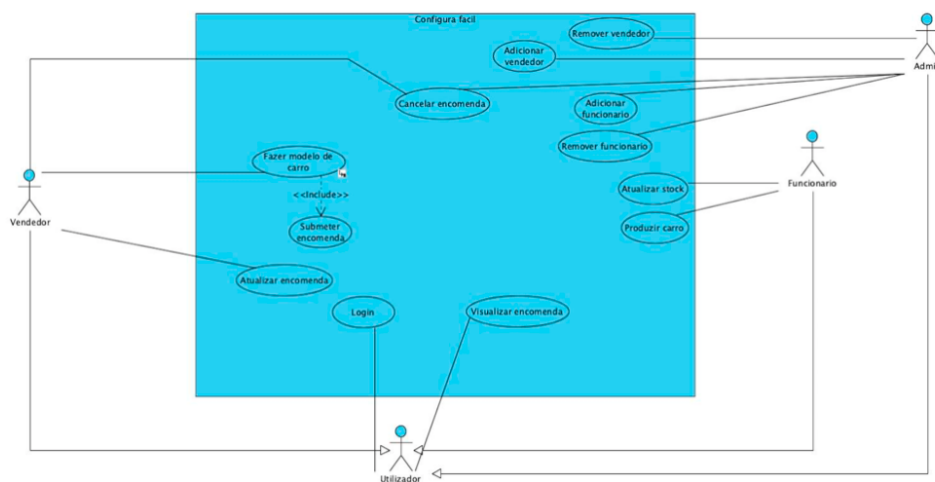


Figura 2: Modelo de Use Case

### 2.2.1 Login

Este Use Case permite ao utilizador autenticar-se na plataforma. Para que tal aconteça é necessário que o Utilizador esteja registado.

Os dados do Utilizador têm que ser inseridos para que estes possam ser validados pelo sistema. Se forem válidos, o Utilizador fica autenticado, e o sistema informa do sucesso da operação e encaminha para o Menu correspondente, caso contrário, o sistema avisa do insucesso da operação e volta para o Login.

Use Case:	Login	
Actor:	Utilizador	
Pré condição:	Utilizador não se encontra com a sessão iniciada	
Pós condição:	Mais um login registado	
Cenário Normal	Actor input	System response
	1. Fornece UserName e Password	2. Verifica se utilizador esta registado
		3. Verifica a Password
		4. Login Efetuado
		5. Regista Login
Exception 1 (Passo 2) [Utilizado nao registado]		2.1 Avisa que o utilizador não se encontra no sistema 2.2 Avisa o Administrador
Exception 2 (Passo 3) [Password invalida]		3.1 Avisa o utilizador que a password é invalida

Figura 3: Login

### 2.2.2 Visualizar Encomenda

Com este Use Case é permitido ao Utilizador do sistema Visualizar as encomendas respetivas, isto consiste em que os respetivos Utilizadores do sistema sejam capazes de visualizar sempre as suas respetivas encomendas.

No caso do Admin ele é capaz de visualizar todas as encomendas do sistema, enquanto que o Vendedor e Funcionário só conseguem visualizar as suas encomendas.

Use Case:	View Car	
Actor:	Utilizador	
Pré condição:	Utilizador autenticado	
Pós condição:	Imprime Encomenda	
Cenário Normal	Actor input	System response
	1. Pede a lista das encomendas	2. Verifica se lista de encomendas possui elementos
		3. Imprime lista de encomendas
	4. Escolhe encomenda	5.verifica se a encomenda selecionada é válida
		6.Imprime encomenda selecionada
Exception 1 (Passo 2) [Lista de enc vazia]		2.1 Avisa que lista não possui encomendas
Exception 2 (Passo 5) [Encomenda escolhida não valida]		2.1 Avisa que a encomenda escolhida não é válida

Figura 4: Vizualizar Encomenda

### 2.2.3 Configuração Ótima

Este Use Case permite ao Vendedor efetuar uma encomenda de um carro de forma mais simplificada, onde apenas é necessário selecionar a marca, o modelo e indicar um orçamento.

A configuração do carro será feita seguindo uma lista de prioridades relativamente as componentes que, serão selecionadas até atingir o limite estabelecido pelo orçamento.

Use Case:	Configure Otim	
Actor:	Vendedor	
Pré condição:	Vendedor autenticado	
Pós condição:	Encomenda com configuracao otima	
Cenário Normal	Actor input	System response
	1. Introduzir marca	
	2. Introduzir modelo	
	3. Introduzir budget	
		4. Apresentar configuracao ao vendedor
	5. Aceita/Não Aceita	6. Registo da encomenda
Exception 1 (Passo 4) [vendedor recusa configuracao]		4.1 Emite mensagem de cancelamento de encomenda
Com.Alt 1 (Passo 4) [vendedor altera configuracao]	2.2 Escolhe componentes	2.1 Menu de escolha de componentes
		2.3 Adiciona encomenda

Figura 5: Configuração Ótima

#### 2.2.4 Submeter Encomenda

Com este Use Case é nos permitido fazer uma submissão de uma encomenda já finalizada para a fábrica, começando então a produção da mesma encomenda na fábrica. Caso não exista componentes em stock para começar a produção do carro, o sistema vai avisar o vendedor do mesmo e adiva se pretende continuar ou não.

Use Case:	Submit Order	
Actor:	Vendedor	
Pré condição:	Vendedor autenticado	
Pós condição:	Encomenda adicionada a lista de encomendas	
Cenário Normal	Actor input	System response
	1. Indica que a encomenda esta concluida	
		2. Verifica stock de componentes da encomenda
		3. Verifica existencia de componentes obrigatorios na encomenda
		4. Submete encomenda
Comp Alt 1 (Passo 2) [Componentes sem stock escolhidos]	2.2 Aceita	2.1 Pergunta se pretende colocar a encomenda em fila de espera ate receber stock
Exception 1 (Passo 2) [User não aceita encomenda]		2.1 Apresenta mensagem de cancelamento de encomenda
Exception 2 (Passo 3) [encomenda sem componentes obrigatorios]		3.1 Apresenta mensagem de erro

Figura 6: Submeter Encomenda



### 2.2.5 Cancelar Encomenda

Este Use Case permite ao Administrador e ao Vendedor cancelar encomendas.

Em ambos menus destes utilizadores irá aparecer a lista de encomendas feitas, onde puderam seleccionar a que desejarem para cancelar.

Use Case:	Cancel Order	
Actor:	Vendedor ou Administrador	
Pré condição:	Vendedor autenticado	
Pós condição:	Encomenda cancelada	
Cenário	Actor input	System response
Normal	1. Pede a lista das encomendas	2. Verifica se lista de encomendas possui elementos
		3. Imprime lista de encomendas
	4. Seleciona encomenda a cancelar	5. Verifica status de encomenda
		6. Cancela a encomenda
		7. Regista o cancelamento
Exception 1 (Passo 2) [Lista de enc vazia]		2.1 Avisa que não possui encomendas
Exception 2 (Passo 5) [Encomenda em producao]		5.1 Avisa da impossibilidade de cancelar
Alternativa 3 (Passo 5) [Não cancelar]		5.3 Voltar para passo 4

Figura 7: Cancelar Encomenda

### 2.2.6 Modificar Encomenda

Este Use Case permite ao Administrador e ao Vendedor alterar encomendas.

Em ambos menus destes utilizadores irá aparecer a lista de encomendas feitas, onde puderam seleccionar a que desejarem alterar.

Use Case:	Change Order	
Actor:	Vendedor	
Pré condição:	Vendedor autenticado	
Pós condição:	Registo de Alteracao de dados de encomenda	
Cenário	Actor input	System response
Normal	1. Pede a lista das encomendas	2. Verifica se lista de encomendas possui elementos
		3. Imprime lista de encomendas
	4. Seleciona encomenda a atualizar	5. Disponibiliza encomenda
	6. Altera os dados	7. Confirma novos dados
		8. Atualiza a encomenda
Exception 1 (Passo 2) [Lista de enc vazia]		2.1 Avisa que não tem encomendas registadas em seu nome
Exception 2 (Passo 7) [Componente incompativel]		7.1 Avisa de componente(s) incompativel(is)
Alternativa 1 (Passo 7) [Componente incompativel]		7.1 Voltar ao passo 6

Figura 8: MODificar Encomenda

### 2.2.7 Adicionar Vendedor

Este Use Case permite ao Administrador adicionar um novo Vendedor para utilizar o programa. O Administrador deve encontrar-se autenticado.

Os dados do Vendedor têm que ser inseridos para que estes possam ser validados pelo sistema. Se forem válidos, o Vendedor é adicionado à lista de vendedores e o sistema informa do sucesso da operação e encaminha para o Menu correspondente, caso contrário, o sistema avisa do insucesso da operação.

Use Case:	Adiciona User	
Actor:	Administrador	
Pré condição:	Administrador autenticado	
Pós condição:	Vendedor adicionado	
Cenário Normal	Actor input	System response
	1. Pede para adicionar User	2. Mostra campos de User
	3. Preenche os campos	4. Verifica os dados
		5. Regista novo User
Exception 1 (Passo 4) [Utilizador existente]		4.1 Avisa que o User já se encontra registado
Exception 2 (Passo 4) [Dados incorretos]		4.2 Avisa que dados estão incorretos

Figura 9: Adiciona Vendedor

### 2.2.8 Remover Vendedor

Em contexto real é normal que haja alterações na equipa de vendedores, desta forma decidimos dar ao Administrador o poder de remover Vendedores que já não façam parte do sistema.

O Administrador pode então remover o Funcionario de Fabrica visualizando a lista de utilizadores, a seguir o Administrador seleciona o Funcionario de Fabrica a ser removido.

Use Case:	Remove User	
Actor:	Admin	
Pré condição:	Admin autenticado	
Pós condição:	Lista de Users	
Cenário Normal	Actor input	System response
	1. Pede para remover User	2. Mostra lista de Users
	3. Escolhe User a remover	4. Pede confirmação
	5. Confirma	5. Remove componente
Exception 1 (Passo 4) [Componente escolhido inválido]	5.1 não confirma	5.2 Volta a passo 2

Figura 10: Remove Vendedor

### 2.2.9 Adicionar Funcionário

Este Use Case permite ao Administrador adicionar um novo funcionario para utilizar o programa. O Administador deve encontrar-se autenticado.

Os dados do Funcionário têm que ser inseridos para que estes possam ser validados pelo sistema. Se forem válidos, o Funcionário é adicionado à lista de funcionários e o sistema informa do sucesso da operação e encaminha para o Menu correspondente, caso contrário, o sistema avisa do insucesso da operação.

Use Case:	Adicionar Funcionario	
Actor:	Administrador autenticado	
Pré condição:	Administrador autenticado	
Pós condição:	Funcionario Adicionado	
Cenário Normal	Actor input	System response
	1. Pede para adicionar Funcionario	2. Pede credenciais do funcionario
	3. Preenche dados de funcionario	4. Verifica os dados
		5. Adiciona funcionario
Exception 1 (Passo 4) [Utilizador existente]		4.1 Avisa que o funcionario já se encontra registado
Exception 2 (Passo 4) [Dados incorretos]		4.2 Avisa que dados estão incorretos

Figura 11: Adiciona Funcionário

#### 2.2.10 Atualizar Stock

O Funcionário de fábrica é responsável por atualizar o Stock dos componentes, seja no momento em que manda produzir um carro (onde subtrai na contagem o numero de componentes utilizadas nessa produção), como também quando recebe novas peças (onde adiciona na contagem o numero de componentes recebidas).

Use Case:	Atualizar Stock	
Actor:	Funcionario	
Pré condição:	Funcionário autenticado	
Pós condição:	Stock Atualizado	
Cenário Normal	Actor input	System response
	2. Selecciona Componente a Atualizar	1. Mostra lista de Componentes
	3. Introduz Valor de novos elementos	
		4. Verifica validade dos dados
		5. Atualiza Stock
Exception 1 (Passo 4) [Utilizador existente]		7. Notifica administrador
	4.2 Pesquisa nova peça	4.1. Avisa houve erro

Figura 12: Atualiza Stock

#### 2.2.11 Adicionar Pacote

Este Use Case permite ao Administrador adicionar um novo Pacote para a ser utilizado na composição de um novo carro. O Administrador deve encontrar-se autenticado.

O Utilizador pede para adicionar um novo pacote e é apresentada a lista de componentes para este seleccionar os que fazem parte do novo pacote. Se os componentes forem válidos e compatíveis, o pacote é adicionado à lista pacotes, o sistema informa do sucesso da operação e encaminha para o Menu correspondente. No entanto, se os componentes apresentarem incompatibilidades, ou já existir um pacote com os mesmos componentes, informa que não é possível adicionar o novo pacote.

Use Case:	Adicionar pacote	
Actor:	Admin	
Pré condição:	Admin autenticado	
Pós condição:	Lista de pacotes atualizada	
Cenário Normal	Actor input	System response
	1. Pede para adicionar novo pacote	2. Mostra lista de componentes
	3. Escolhe lista de componentes para o pacote	4. Verifica validade de componentes escolhidos
		5. Verifica existencia de pacote igual
Exception 1 (Passo 4) [Componentes escolhidos invalidos]		2.1 Informa que os dados introduzidos não são validos
Exception 2 (Passo 5) [Pacote já existente]		2.1 Informa que já existe um pacote igual no sistema

Figura 13: Adiciona Package

### 2.2.12 Remover Pacote

Existem pacotes base que não é possível alterar, porém também existem pacotes promocionais que podem ser removidos, desta forma decidimos que o Administrador decide quais os pacotes que devem ser removidos.

O Administrador pede então a lista de pacotes, o sistema apresenta-lhe a lista e permite que o Administrador remova um pacote selecionado, o sistema verifica a validade do pacote e remove-o. Se, de alguma forma, o Administrador pedir para remover um pacote não existente o sistema avisa-o.

Use Case:	Remove Package	
Actor:	Admin	
Pré condição:	Admin autenticado	
Pós condição:	Lista de pacotes atualizada	
Cenário Normal	Actor input	System response
	1. Pede lista de pacotes	2. Mostra lista de Pacotes
	3. Escolhe pacote a remover	4. Pede para confirmar
	5. Escolhe confirmação	6. Remove Package
Exception 1 (Passo 5) [Não Confirma]	5.1 não confirma	5.2 volta ao passo 3

Figura 14: Remove Package

### 2.2.13 Adicionar Componente

Este Use Case permite ao Administrador adicionar um novo Componente para a ser utilizado na composição de um novo carro. O Administrador deve encontrar-se autenticado.

Os dados do componente têm que ser inseridos para que estes possam ser validados pelo sistema. Se forem válidos, o Componente é adicionado à lista de componentes e o sistema informa do sucesso da operação e encaminha para o Menu correspondente. No entanto, se os dados do componente forem insuficientes ou o mesmo já existir, informa que não é possível adicionar o componente responsável.

### 2.2.14 Remover Componente

Em contexto real há certos componentes que são produzidos com defeito ou simplesmente são descontinuados, desta forma decidimos conceder ao Administrador o poder de remover esses componentes.

Use Case:	Adicionar componente	
Actor:	Admin	
Pré condição:	Admin autenticado	
Pós condição:	Lista de componentes atualizada	
Cenário Normal	Actor input	System response
	1. Pede para adicionar novo componente	
	2. Coloca dados de novo componente	
		3. Verifica dados de novo componente
		4. Verifica a existência do componente introduzido no sistema
Exception 1 (Passo 3) (Componente escolhido inválido)		2.1 Informa que os dados introduzidos não são válidos
Exception 2 (Passo 4) (Componente já existente)		2.1 Informa que já existe um componente igual no sistema

Figura 15: Adiciona Componente

O Administrador pede então a lista de componentes, o sistema apresenta-lhe a lista e permite que o Administrador remova um componente seleccionado, o sistema verifica a validade do componente e remove-o. Se, de alguma forma, o Administrador pedir para remover um componente não existente o sistema avisa-o.

Use Case:	Remove Component	
Actor:	Admin	
Pré condição:	Admin autenticado	
Pós condição:	Lista de componentes atualizada	
Cenário Normal	Actor input	System response
	1. Pede para remover componente	
	3. Escolhe componente a remover	2. Mostra lista de componentes
		4. Pede para confirmar
	5. Escolhe confirmação	6. Remove componente
Exception 1 (Passo 5) [Não Confirma]	5.1 não confirma	5.2 volta ao passo 3

Figura 16: Remove Componente

## 2.3 Diagrama de Estado

### 2.3.1 Diagrama de Estado Inicial

Na Figura em cima representada vemos, que o Utilizador para entrar no sistema Configura Facil terá de primeiro efectuar o Menu Login, só depois se estiver credenciado é que poderá entrar no sistema e ser reencaminhado para o seu Menu correspondente que será explicado á frente.

### 2.3.2 Diagrama de Estado Administrador

Se o Utilizador neste caso for um Admin ele terá vários estados: AddUser, RemoveFuncionario, RemoveCar, Add-Component e por fim Remove Component. Estes estados todos vão usar o estado ConfirmContinue e ConfirmRemove, que são usados como um mecanismo de segurança.

### 2.3.3 Diagrama de Estado Vendedor

Neste Estado o vendedor pode realizar várias ações. Pode iniciar a configuração de uma nova encomenda que o leva para um novo estado Configura Carro onde procede à escolha dos componentes do mesmo. Pode planear a encomenda

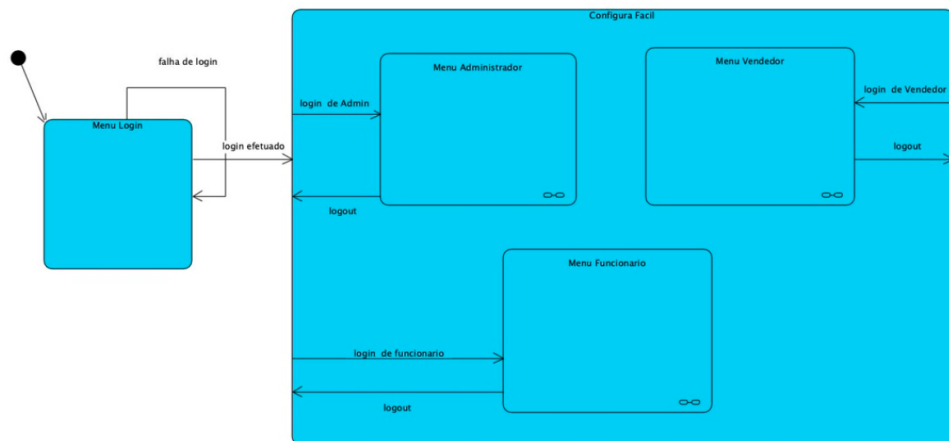


Figura 17: Diagrama de Estado Inicial

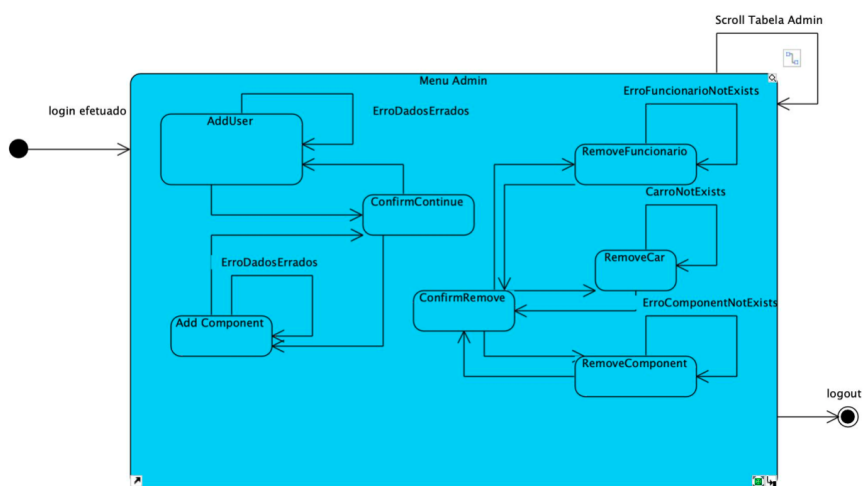


Figura 18: Diagrama de Estado Administrador

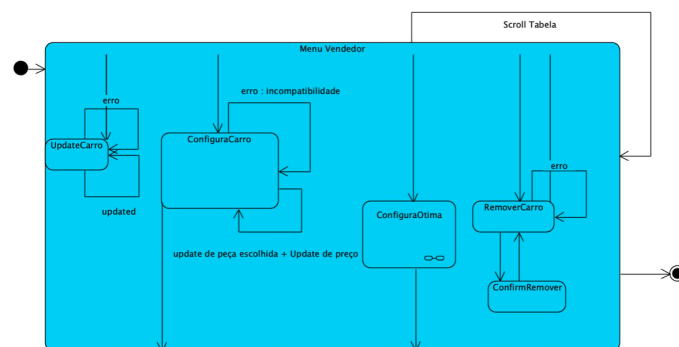


Figura 19: Diagrama de Estado Vendedor

através da opção ConfiguraOtima, pode atualizar uma encomenda já realizada alterando os seus componentes e pode também proceder a remoção de uma encomenda já gravada.

#### 2.3.4 Diagrama de Estado Funcionario de Fabrica

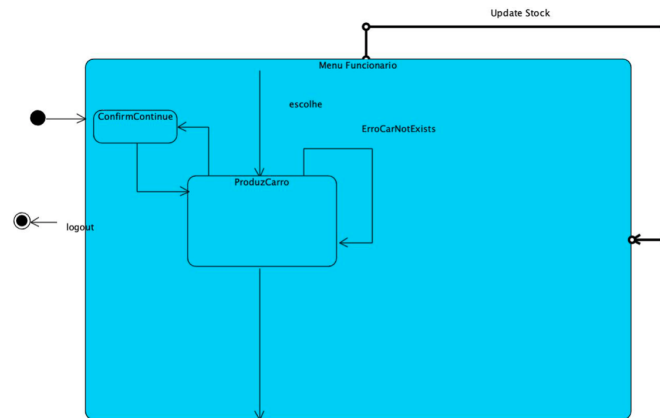


Figura 20: Diagrama de Estado Funcionario Fabrica

Neste Estado o funcionario pode atualizar o stock de um dado componente através de uma lista presente no estado ou escolher iniciar a produção de um carro em espera.

#### 2.3.4 Diagrama Estado Configuracao Optima

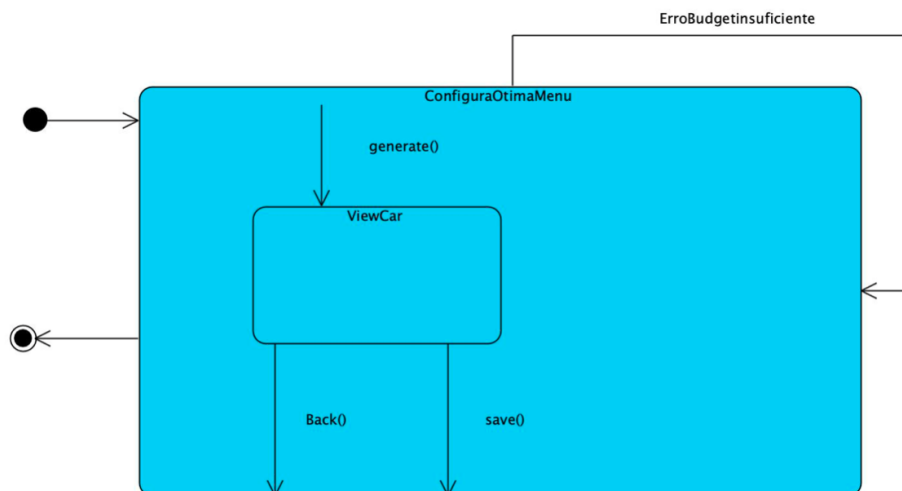


Figura 21: Diagrama de Estado Configuracao Optima

Neste Estado o utilizador pode apresentar os valores para a opção configura optima e gerar a melhor opção de configuração de carro para o budget escolhido que aparecerá numa nova janela com a configuração gerada. Após a visualização o utilizador pode gravar a encomenda ou caso não se encontre satisfeito pode voltar atrás e apresentar um novo valor para o algoritmo encontrar uma nova opção.

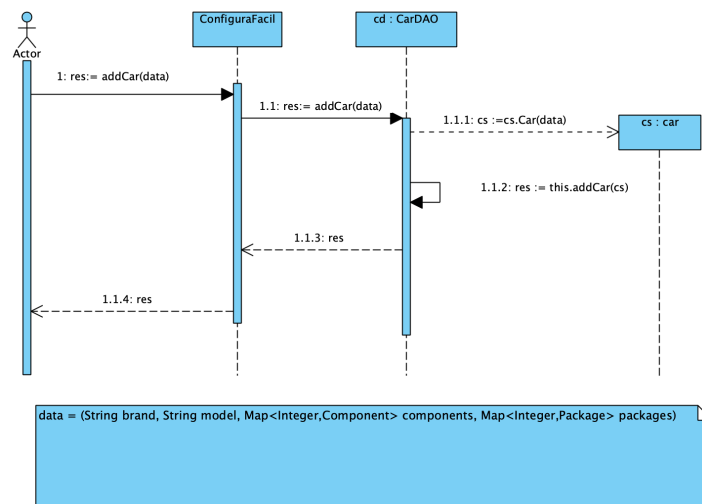


Figura 22: Diagrama de Sequencia

## 2.4 Diagrama de Sequência

O Diagramas de Sequência representam as interações entre objetos através das mensagens que são trocadas entre eles. Permite também analisar a distribuição de “responsabilidade” pelas diferentes entidades, ou seja, analisar onde está a ser efetuado o processamento.

De seguida, são apresentados os Diagramas de Sequência relativos aos Use Case descritos anteriormente.



2.4.1 Login

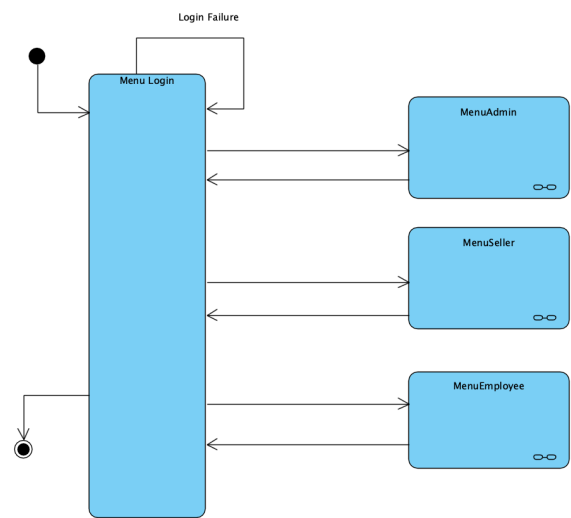


Figura 23: Diagrama de Sequencia Login

2.4.2 Administrador

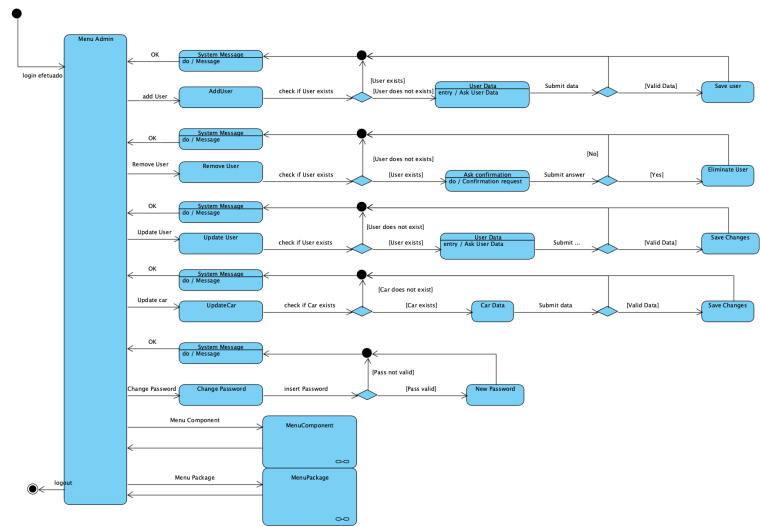


Figura 24: Diagrama de Sequencia Administrador

### 2.4.3 Vendedor

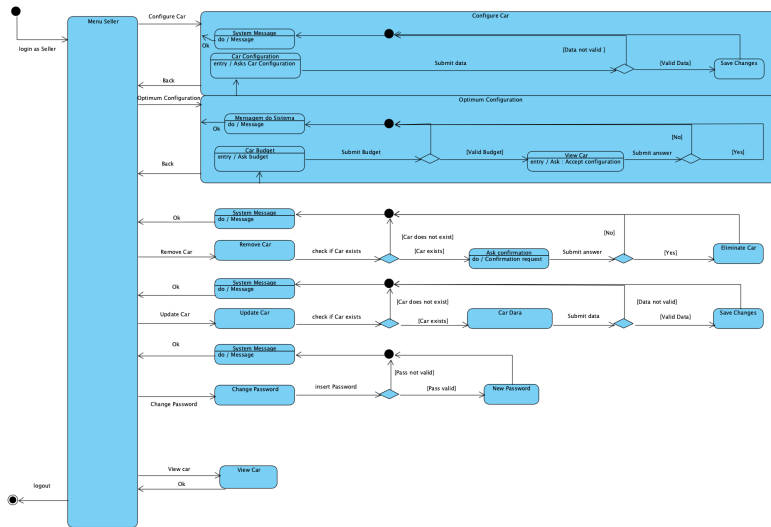


Figura 25: Diagrama de Sequencia Vendedor

### 2.4.4 Funcionário Fabrica

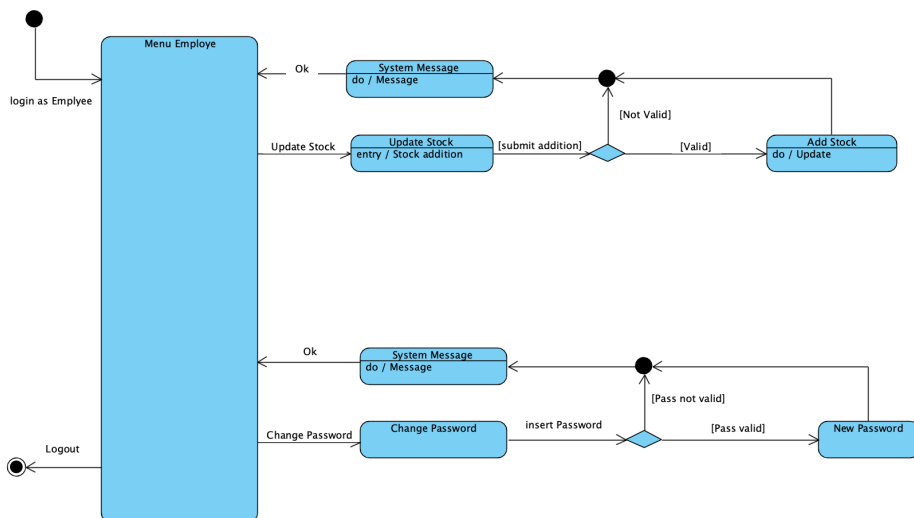


Figura 26: Diagrama de Sequencia Funcionário Fabrica

### 2.4.5 Package

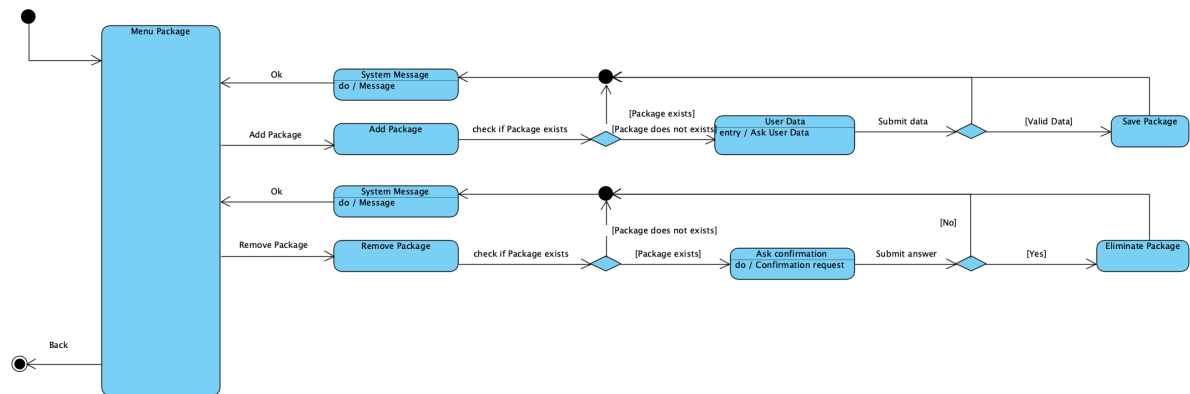


Figura 27: Diagrama de Sequencia Package

### 2.4.6 Componente

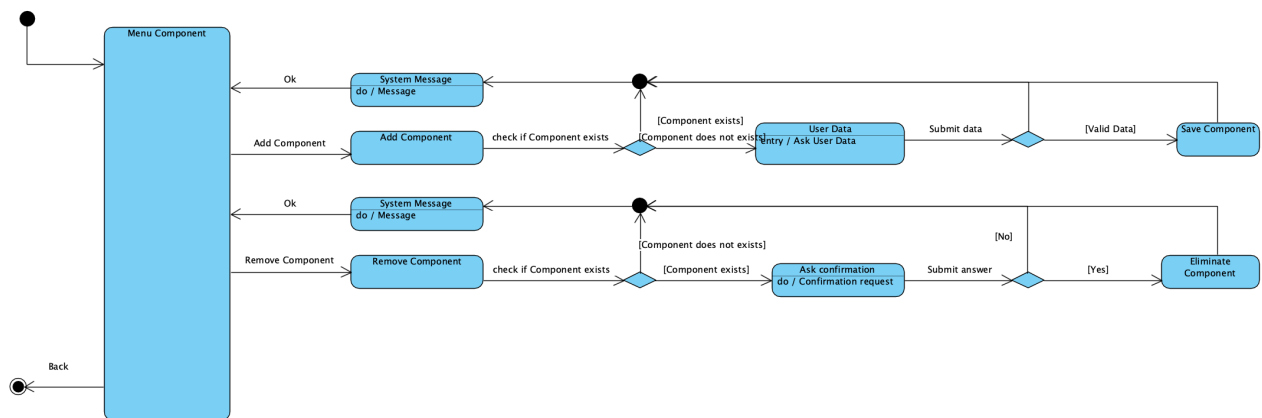


Figura 28: Diagrama de Sequencia Componente

## 2.5 Diagrama Package

No Diagrama de Package existem três camadas principais: a Camada de Negócio, a Camada de Interface e a Camada de Dados.

O Package da Camada de Negócio é o dono dos packages Componentes, Interface ConfiguraFacil, Utilizadores e Encomendas.

O Package Camada de Interface acede aos elementos exportados pelo package Interface ConfiguraFacil e este último acede aos elementos exportados pelo package Camada de Dados.

Qualquer alteração feita no package Interface ConfiguraFacil afeta os package destino Utilizadores, Encomendas e Componentes.

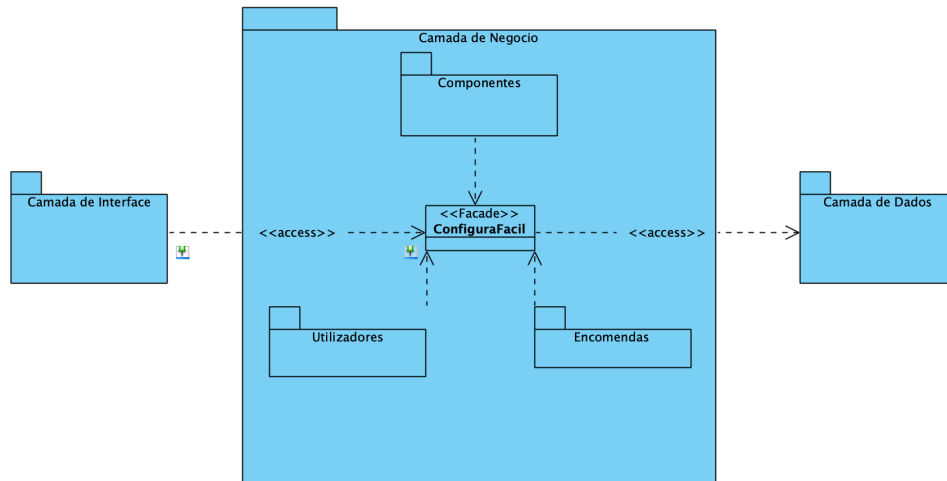


Figura 29: Diagrama de Package

## 2.6 Diagrama de Classes (com métodos)

Partindo dos relacionamentos do Modelo de Domínio e das suas multiplicidades, foi construído o Diagrama de Classes. Neste diagrama, encontram-se definidas as variáveis de cada classe. Neste diagrama, codU, codC, codE, e codIP funcionam como chaves das relações a que pertencem.

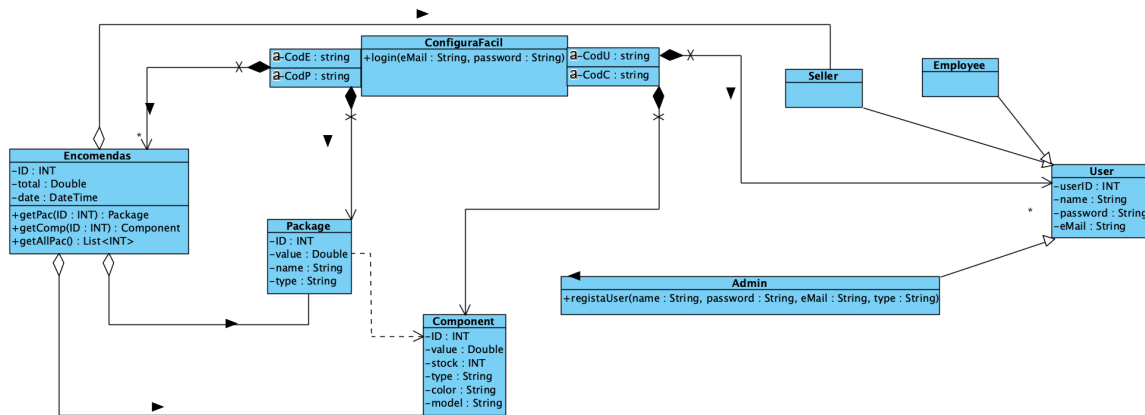


Figura 30: Diagrama de Classes

## 2.7 Diagrama de Classes com DAO's

A partir deste Diagrama de Classes começamos com a implementação propriamente dita, implementando a persistência, tendo sido escolhida a visão onde a lógica está na base de dados, ou seja, usando DAO's.

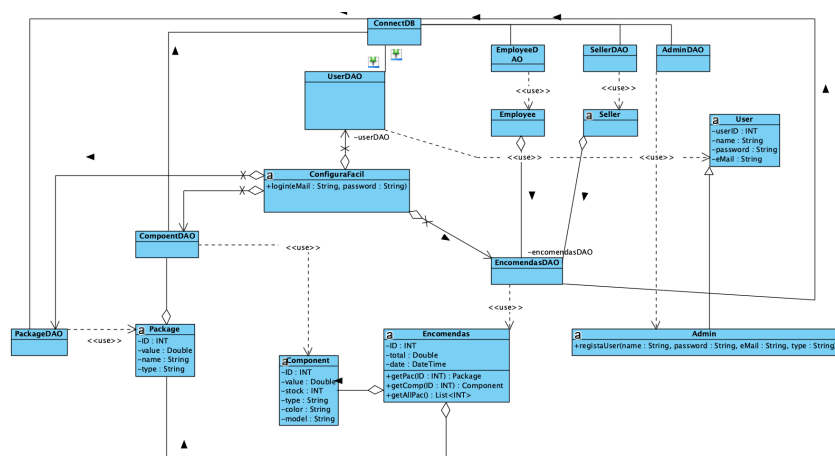


Figura 31: Diagrama de Classes com DAO's

## 2.8 Diagrama de Instalação

Neste diagrama de instalação é possível ver os requisitos necessários a nível de hardware e software para que o modelo do nosso sistema possa ser instalado e se torne funcional. O ambiente de execução do servidor terá ainda acesso a um componente, através de uma ligação JDBC, que será a base de dados em MySQL implementada neste servidor, que irá conter todos os dados necessários para o funcionamento correto da aplicação.

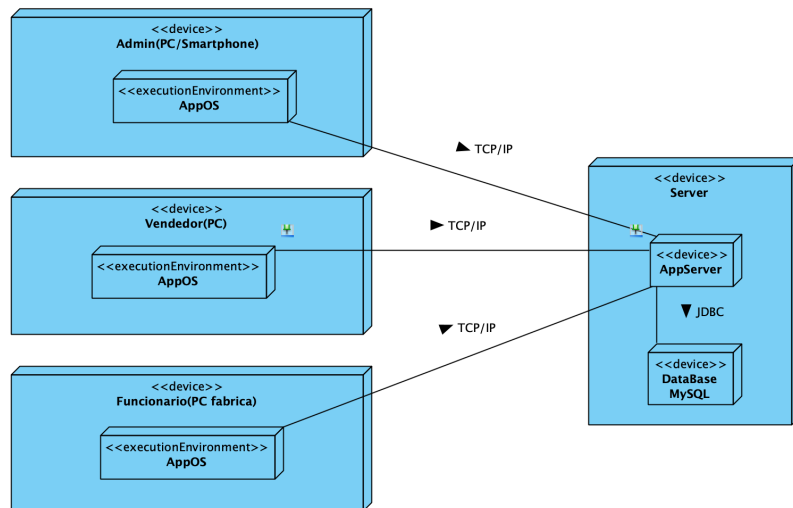


Figura 32: Diagrama de Instalação

## 2.9 Diagrama de Atividades

Nesta secção são apresentados alguns exemplos dos Diagramas de Atividade relativos a cada Use Case.

### 2.9.1 Adiciona Componente

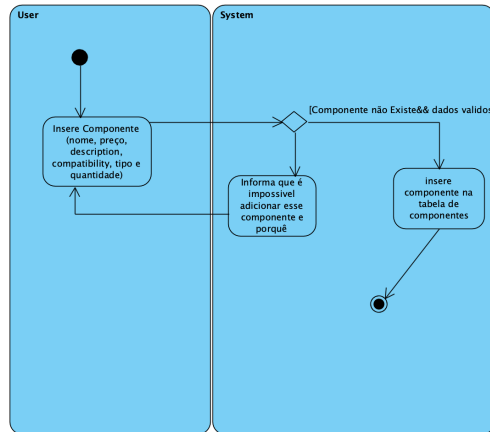


Figura 33: Adiciona Componente

### 2.9.2 Adiciona Package

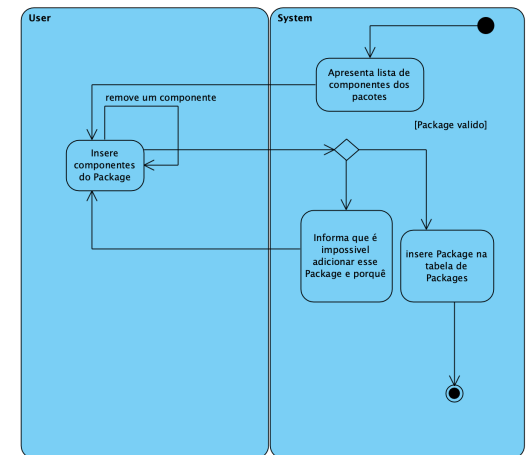


Figura 34: Adiciona Package

2.9.3 Adiciona Utilizador

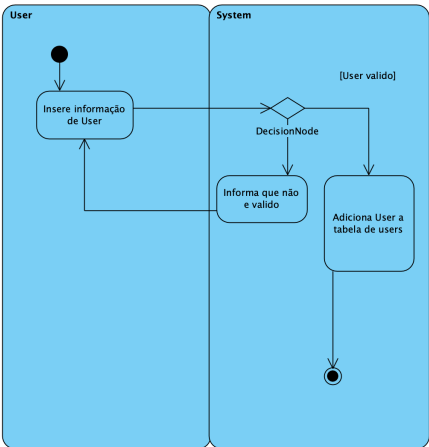


Figura 35: Adiciona Utilizador

2.9.4 Alterar Encomenda

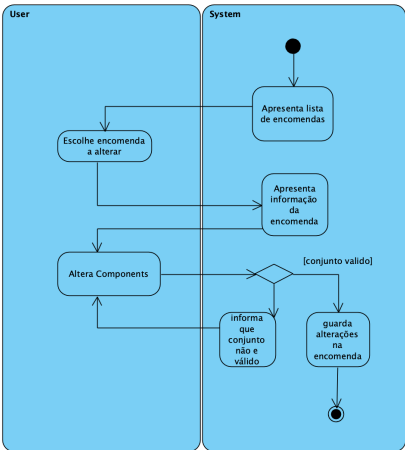


Figura 36: Alterar Encomenda





2.9.7 Login

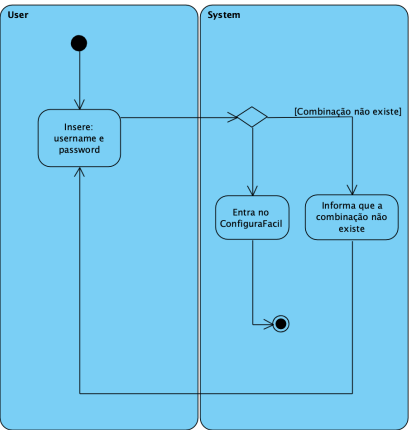


Figura 39: Login

2.9.8 Remove Componente

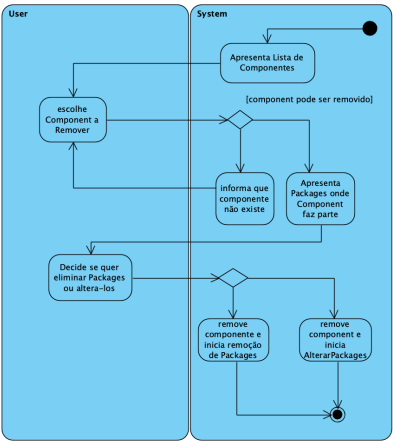


Figura 40: Remove Componente

2.9.9 Remove Encomenda

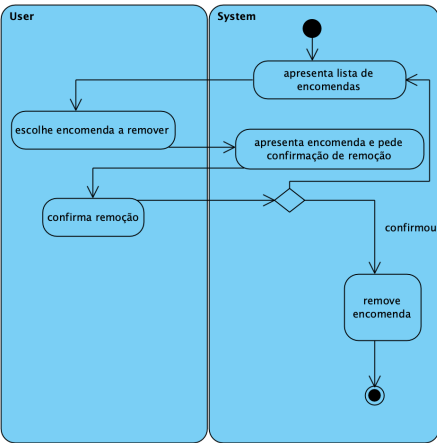


Figura 41: Remove Encomenda

2.9.10 Remove Package

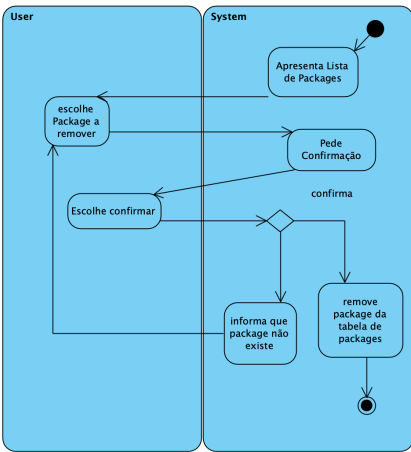


Figura 42: Remove Package

2.9.11 Remove Utilizador

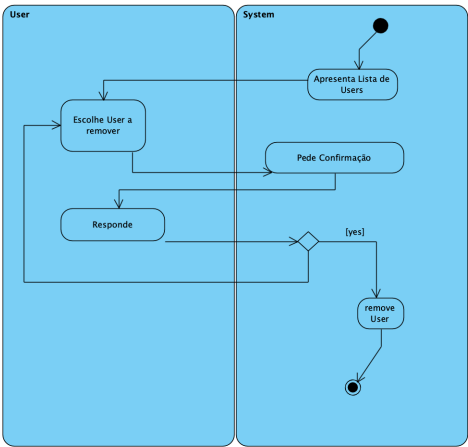


Figura 43: Remove Utilizador

2.9.12 Atualiza Stock

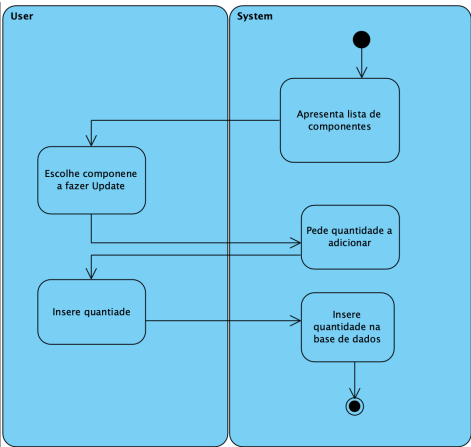


Figura 44: Atualiza Stock

## 2.10 Base de Dados

Para que o projeto possa ser considerado viável e os dados do sistema sejam guardados, é necessário criar uma base de dados robusta. Para esta etapa, começámos por construir o Modelo Lógico (usando o MySQL Workbench), que após varias alterações e tentativas conseguimos aprimorar tendo depois convertido para o Modelo Físico (usando a opção do SQL database forwarding). O Modelo Lógico encontra-se representado na figura 45, com todas as entidades e relacionamentos que achamos necessários. Depois de criar a base de dados e fazer o povoamento da mesma, foi necessário fazer a conexão entre a mesma e o nosso sistema do Netbeans, permitindo que o sistema do nosso projeto tenha acesso à base de dados construída.

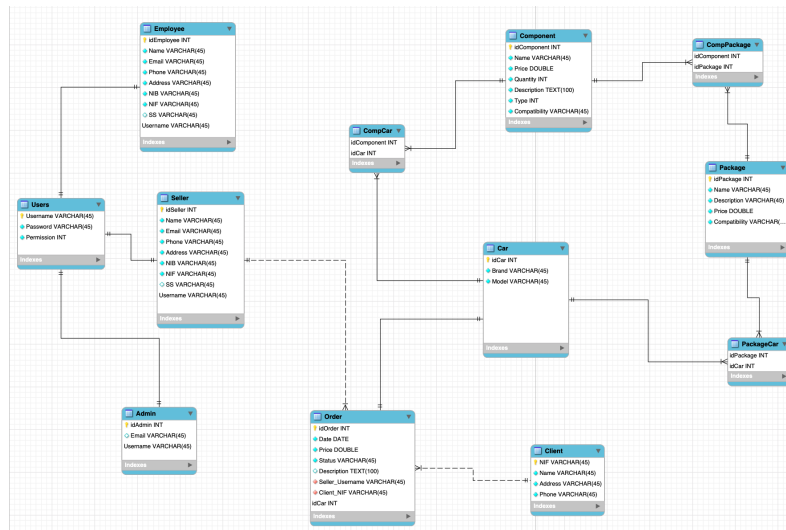


Figura 45: Modelo Lógico da Base de Dados

### **3 Conclusão**

No âmbito da segunda fase do projeto, foi-nos pedida a implementação do problema em causa. A modelação efetuada na primeira fase deste projeto facilitou em grande escala a implementação do problema devido ao planeamento ao detalhe do sistema de dados e de implementação das diversas funcionalidades.

Para isto, numa primeira instância desenvolvemos a base de dados em SQL na qual iríamos armazenar todos os dados necessários para o nosso projeto.

De seguida começamos a pensar na ligação da base de dados para que a conseguíssemos consultar. Aqui implementamos os DAO.

Finalmente fizemos a parte de implementação de funcionalidades e respetiva interface gráfica para qual o netbeans foi uma grande ajuda.

Assim acreditamos ter conseguido elaborar um sistema de software eficaz, simples e robusto que satisfaz todas as especificações necessárias ao bom funcionamento deste sistema.