



Desenvolvimento de Sistemas de Software

Guilherme Viveiros (A80524) César Borges (A81644)
Luís Macedo (A80494) Nuno Valente (A81986)

Novembro 2018



Conteúdo

1	Descrição	3
2	Descrição da arquitetura de classes:	4
3	Descrição dos Use Case	5
4	Descrição da aplicação desenvolvida (ilustração das funcionalidades):	7
5	Descrição das máquinas de estado	7
6	Proposta de Interface	9
7	Análise Crítica	12

1 Descrição

No âmbito da disciplina Desenvolvimentos de Sistemas de Software foi nos proposto pelos docentes o desenvolvimento de uma plataforma para configuração de carros.

Esta plataforma consiste num conjunto de métodos disponíveis para que o utilizador (com a ajuda de um funcionário) consiga em tempo real, encomendar o(s) carro(s) mais adequado(s) de acordo com a sua visão.

Esta aplicação deverá ser capaz de criar uma configuração individualizada para um cliente ou criar uma configuração com pacotes pre-definidos.

Um pacote pré-definido é um conjunto de itens que constituem o carro já finalizado e que porventura irá usufruir de um desconto. Dito isto caso o Cliente deseje um Pacote X e queira retirar um item ao pacote, o pacote fica anulado pagando este a totalidade dos itens.

O cliente poderá ter a opção de adicionar mais itens ao carro após ter escolhido pacotes ou vice-versa, tendo a aplicação o dever de fazer chegar a mensagem ao utilizador sobre a possível incompatibilidade ou dependência de itens.

A chegada de stocks de componentes a fábrica também deverá ser controlada pela aplicação.

Para além disso, o cliente terá a oportunidade de pedir “auxílio” à plataforma acedendo ao método configuração ótima. Alguns dados terão que ser fornecidos, como o dinheiro máximo que pretende gastar entre outras preferências para que a aplicação consiga dar uma resposta válida e concreta. Este relatório irá especificar como esta plataforma foi analisada e modelada realçando os seguintes tópicos :

1. Diagrama de Classes;
2. Diagrama de Use Cases;
3. Diagrama de Máquinas de Estado;
4. Protótipo de baixa fidelidade de Interface.

2 Descrição da arquitetura de classes:

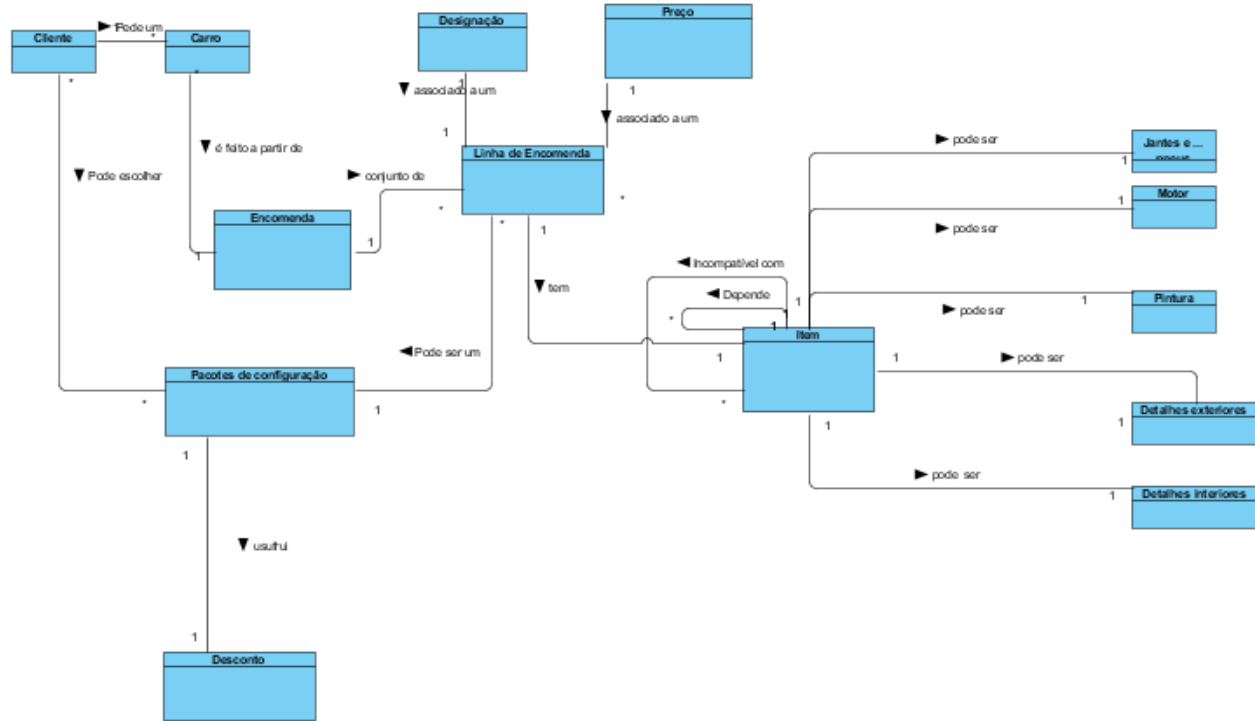


Figura 1: Diagrama de Classe

Decidimos que esta seria a melhor abordagem inicial para o diagramas de classes. Cada cliente pode pedir vários carros que serão compostos numa encomenda.

Cada encomenda é constituída por linhas de encomenda com uma respetiva designação e preço.

Uma linha de encomenda é composta por um certo item e está associada a um determinado preço e designação, dito isto, um pacote de configuração é composto por várias linhas de encomenda. Os pacotes de configuração usufruem de desconto.

3 Descrição dos Use Case

Foram feitos, até ao momento, 3 máquinas de estado de forma a modelar comportamentos do nosso sistema e consequentemente, garantir um bom funcionamento da aplicação a ser desenvolvida. Estas 3 máquinas são as seguintes:



Figura 2: Use Case

Sobre os Use Case, de acordo o enunciado proposto ,para o Cliente realizar uma encomenda terá que se dirigir a um Funcionário logo decidimos implementar 3 atores, Funcionário , Gestor e Admin.

Funcionário tem como função ajudar o cliente a realizar a sua compra , este consegue Encomendar um Veículo que porventura poderá adicionar/remover peça(s)/pacote(s) ou realizar o processo de configuração ótima, também é este que trata da emissão das faturas.

O Gestor tem como objetivo gerir as encomendas realizadas e o stock, caso o stock esteja a ficar vazio , é o dever do mesmo adicionar as peças escassas.

O Admin pode remover/adicionar funcionários e fazer tudo o que o Gestor e o Funcionário podem fazer, digamos que tem o poder total.

Após termos elaborado o Diagrama de Use Cases , começamos a especificar detalhadamente cada Use Case. Algumas das imagens que estão a ser apresentadas de seguida ilustram algumas destas especificações.

Use Case: Validar Encomenda		
Pré-condição: Estar autenticado		
Pós-condição: Encomenda validada		
Comportamento Normal	Actor	Sistema
		1. Mostra todos as encomendas
	2. Escolhe a encomenda	
		3. Verifica Encomenda
Exceção 1 [existem peças incompatíveis na encomenda] (passo 3)		4. Valida encomenda
		2.1. Mostra mensagem de erro
		2.2. Termina o serviço

Use Case: Rejeitar Encomenda		
Pré-condição: Estar autenticado		
Pós-condição: Encomenda rejeitada		
Comportamento Normal	Actor	Sistema
		1. Mostra todos as encomendas
	2. Escolhe a encomenda	
		3. Rejeita Encomenda

Use Case: Remover Pacote		
Pré-condição: Pacote na Encomenda		
Pós-condição: Pacote removido		
Comportamento Normal	Actor	Sistema
	1. Escolhe pacote	
		2. Verifica a Encomenda
		3. Remove pacote na encomenda
Cenário Alternativo 2 [Itens do pacote com dependências] (passo 2)		2.1. Avisa que se remover o pacote , irão ocorrer dependências, dando a oportunidade de remover os itens dependentes.
	2.2 Confirma	
		2.3. Remove pacote e itens dependentes.
		2.4. Volta para o 2
Caso Alternativo 2 [Pacote removida origina incompatibilidade] (passo 2)		2.1. Avisa que se remover o pacote, irão ocorrer incompatibilidades , dando a oportunidade de remover essas incompatibilidades
	2.2 Confirma	
		2.3. Remove pacote e peças incompatíveis
		2.4. Volta para o 2
Cenário Alternativo 2 [Não confirma a remoção das dependências] (passo 2.1)		2.1.1. Remove peça
		2.1.1. Mostra mensagem de incompatibilidade
Cenário Alternativo 2 [Não confirma a remoção das incompatibilidades] (passo 2.1)		2.2.2. Volta para o 2

Figura 3: Descrição de alguns Use Cases

4 Descrição da aplicação desenvolvida (ilustração das funcionalidades):

A classe Aplicação é o que possibilita o uso dos mecanismos desenvolvidos de forma simples. Esta classe permite a utilizadores e administradores criar e gerir as informações necessária

5 Descrição das máquinas de estado

1. Linha de encomenda: Esta máquina de estado modela o comportamento de encomendar várias peças iguais. Existem 3 estados principais e a transição destes estados é feita ao adicionar/remover peças e verificando se existem peças suficientes em stock para satisfazer a encomenda. Estes estados são “sem peças”, “com peças” e “encomendada” em que o estado “com peças” possui 2 sub-estados, “válida” e “em espera”.
2. Encomenda: Esta máquina de estado tem 4 estados, “vazia”, “válida” (podendo ser “dependente” também), “incompatível” e “encomendada”, as transações de estado resultam principalmente pela inserção ou remoção de componentes, a única forma de sair desta máquina de estado é através do estado “válida” ou “dependente”, isto é, só estes é que podem chegar ao estado “encomendada”.
3. Interface: Existem 4 estados principais, o “Not authenticated”, “Funcionário”, “Administrator” e “Stock Manager”:
 - (a) “Not authenticated” é o estado inicial e final do sistema, qualquer estado transita deste ou é transitado para este, sendo a única forma de sair deste estado ao fazer login;

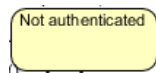


Figura 4: Not Authenticated

- (b) O estado “Funcionário” tem como sub-estados “Escolher Itens” e “Escolher Pacotes” em que, respectivamente, tem três transições interna que lhes permitem adicionar ou remover itens e pacotes à encomenda do seu super-estado, como temos o invariante de ter itens incompatíveis no final de cada transição a encomenda pode ser encomendada desde que não seja vazia, e por último, “Configuração Ótima” que gera uma configuração ótima e que à saída deste estado faz com que a encomenda seja substituída por essa configuração. Ainda é possível fazer reset ao estado “Funcionário” para gerar uma encomenda nova;

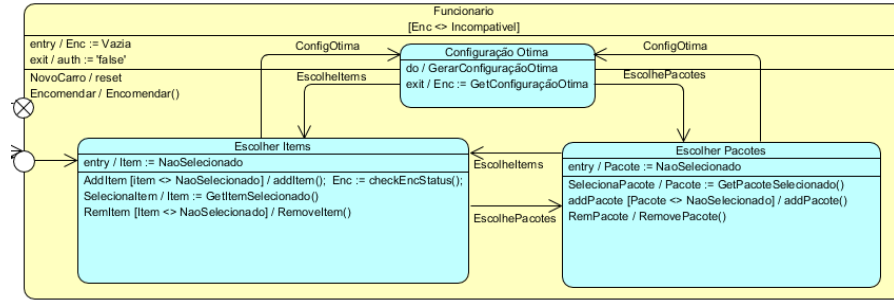


Figura 5: Funcionario

- (c) O estado “Stock Manager” possui 3 sub-estados, “Validar Encomendas”, “Ver Stock” e “Itens a encomendar”, o primeiro permite selecionar encomendas e validá-las, o segundo permite adicionar itens na lista de encomendas, que se encontra no super-estado, para o stocks que, posteriormente no “Itens a encomendar” poderá ser feita a sua consulta e adicionar itens ao stock.

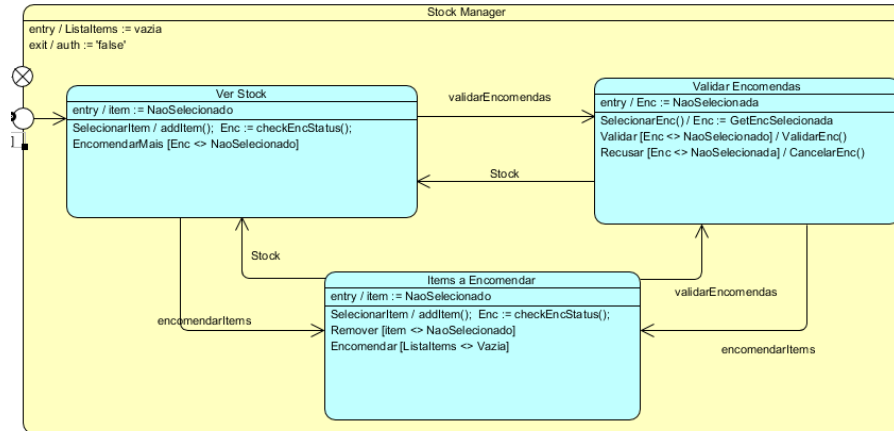


Figura 6: Stock Manager

- (d) O estado “Administrador” tem transições internas que permite remover/adicionar funcionários ao sistema, e ainda tem a opção de ir para qualquer um dos estados referidos anteriormente.

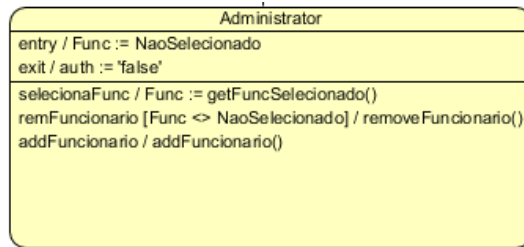


Figura 7: Not Authenticated

6 Proposta de Interface

Ao iniciar a aplicação, é pedido ao utilizador que introduza as suas credenciais. Existem três tipos de autenticações: administrador, gestor e funcionário.

Após a autenticação (Figura 1), o utilizador será reencaminhado para a sua respetiva interface:

The screenshot shows a window titled 'Autenticação' with standard Windows window controls (minimize, maximize, close). Inside the window, there are two text input fields labeled 'Usuário' and 'Senha'. Below these fields is a button labeled 'Autenticar'.

Figura 8: Autenticação

- Um gestor tem acesso à lista de encomendas (por validar) e à lista do stock atual através de uma janela onde pode escolher qual a lista que quer observar.
 - Na lista de encomendas, o gestor consegue visualizar cada elemento de cada encomenda, as suas dependências e incompatibilidades e validar ou recusar essa encomenda.
 - A lista do stock permite ao gestor ver os itens que se encontram no stock como as suas informações e encomendar mais, introduzindo a quantidade que quer encomendar.

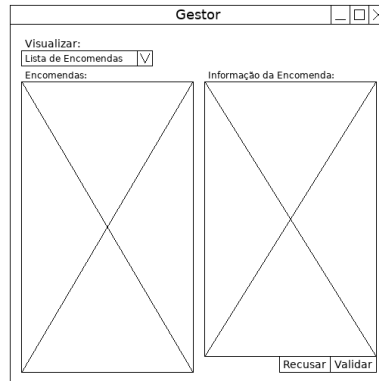


Figura 9: Gestor - Lista de Encomendas

- Um funcionário tem acesso a todos os itens (peças ou pacotes) no sistema, mesmo que estejam fora do stock, dando a possibilidade de escolhê-la ao cliente. Após a escolha estar concluída pode ser observado todas as dependências e incompatibilidades, através de um painel de logs. A aplicação dá a oportunidade de começar uma nova encomenda perdendo tudo que estava feito, guardar a encomenda num ficheiro, abrir um ficheiro com uma encomenda guardada, escolher uma configuração ótima de acordo com um orçamento que é pedido, encomendar o carro, abrindo a janela para introduzir as informações do cliente para a fatura, ver instruções e algum texto sobre a aplicação através de uma menu horizontal no topo da janela.

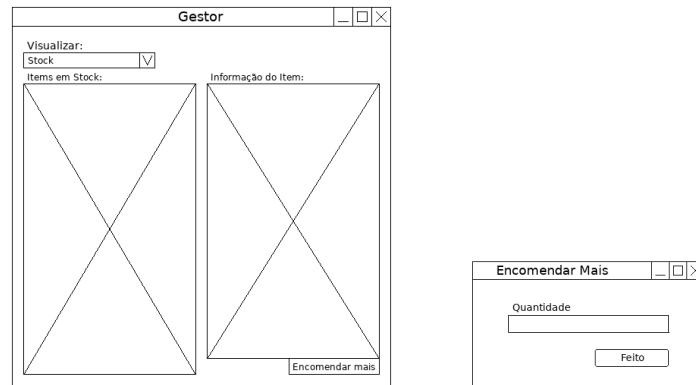


Figura 10: Gestor - Stock

- Um administrador tem acesso a todas a funcionalidades referidas anteriormente (gestor e funcionário) e ainda a possibilidade de adicionar novo funcionário tal como remover um (Figura 6).

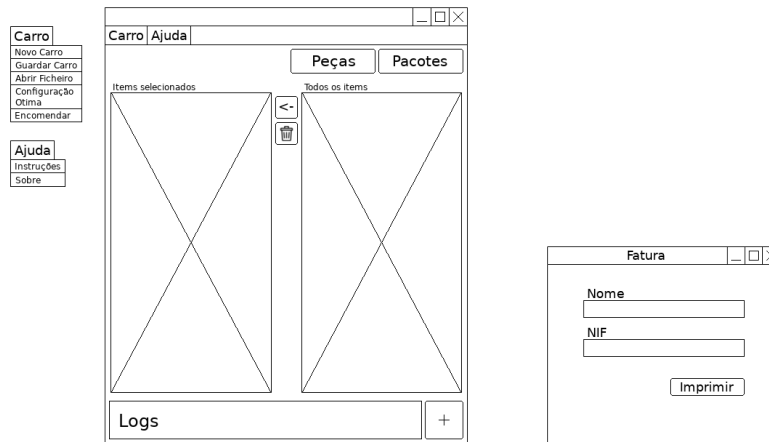


Figura 11: Funcionário

7 Análise Crítica

Conforme elaborávamos o diagrama de Classes íamos ficando com uma melhor ideia de como proceder no trabalho. Por exemplo , quando começamos a modelar o trabalho , tínhamos apenas uma associação entre um funcionário e encomendar veículo em que neste use case tínhamos praticamente meio trabalho feito.

Decidimos então separar os Uses Cases , dividimos Encomendar Veículo em Encomendar Veículo, Remover Peça, Adicionar Peça, Adicionar Pacote entre outros e fomos nos apercebendo que esta divisão iria ser mais oportuna para esta modelação, visto que não só iríamos ter uses cases mais pequenos como iríamos tirar os benefícios dos include/extend.. Através desta separação fomos encontrando cada vez mais comportamentos alternativos , tais como exceções o que implicou numa remodelação mais robusta da interface.

Também concluímos que através da modelação resolvemos conflitos de ideias dentro do grupo que se revelariam durante o desenvolvimento do código da aplicação, o que nos poupa tempo de estar a refazer código e uma de discussão crítica.

The image shows a graphical user interface window titled "Admin". It features three tabs: "Admin", "Gestor", and "Funcionário", with "Admin" currently selected. The main content area is divided into two columns. The left column contains a label "Usuário" above a text input field, and a "Remover" button positioned below the input field. The right column contains a "Usuário" label above a text input field, a "Senha" label above another text input field, and a "Registrar" button positioned below the password input field. The window includes standard OS window controls (minimize, maximize, close) in the top right corner.

Figura 12: Admin