

Programação Orientada a Objetos

1º Bernardo Rodrigues

Eng. Informática

Número 41056

bernardo.i.rodrigues@ubi.pt

2º Miguel Pereira

Eng. Informática

Número 43392

miguelpereira90@sapo.pt

3º Filipe Catarino

Eng. Informática

Número 44280

catarino.fl@hotmail.com

4º Pedro Metelo

Eng. Informática

Número 44300

p.a.metelo10@outlook.pt

Resumo—Neste relatório estão contidas as devidas justificações de decisões e a explicação dos métodos usados no desenvolvimento do trabalho para a cadeira de Programação Orientada a Objetos.

Index Terms—Programação Orientada a Objetos(POO), Identificação(ID); Grupo G18

I. AGRADECIMENTOS

O grupo G18 gostaria de agradecer à professora Paula Prata pelo apoio constante que nos tem dado durante as aulas tanto teóricas como práticas, pela sua disponibilidade em ajudar os seus alunos e pela sua paciência com os mesmos. Agradecemos também pela sua disponibilidade em ajudar os alunos no projeto e também dar aulas mesmo com feriados durante a semana.

II. INTRODUÇÃO

Neste relatório podemos ver o complemento do trabalho de grupo realizado para a cadeira de POO. A linguagem Java é uma poderosa linguagem concebida para se parecer com "C++" porém com um modelo de programação orientada a objetos, esta é amplamente projetada para a sua utilização no ambiente distribuído da Internet. É a linguagem de programação mais popular em aplicações para *smartphones* Android e encontra-se disponível de forma gratuita para todos os programadores que na mesma pretendam programar. No mundo atual, as tecnologias estão a ficar cada vez mais evoluídas, o que faz com que haja uma grande necessidade de se obter uma maior diversidade de informação tanto a nível individual como de grupo num curto espaço de tempo para que, deste modo, possamos estar ao mesmo nível de qualidade de software em comparação a outras aplicações semelhantes à nossa, ou até mesmo num patamar superior às mesmas. Este projeto foi fundamental pois durante o desenvolvimento do mesmo, terão surgido inúmeros obstáculos que iriam "forçar" todos os elementos a por em prática não só os seus conhecimentos sobre o dado tema, como também desenvolver conhecimento sobre novos métodos de implementação de código ainda desconhecidos pelos mesmos.

III. DESENVOLVIMENTO DO PROJETO

A. Escolha da Ideia

A base do projeto consiste na criação de um programa que possua um modelo de programação orientado a objetos que seja capaz de realizar a gestão de uma loja de peças, sendo neste caso especificamente, de carros. Encontra-se orientado

para a simplicidade, para a introdução de dados e para a facilidade de manobra dos mesmos. Este foi desenvolvido no âmbito da unidade curricular de Programação Orientada a Objetos cujo trabalho era desenvolvimento de um projeto que permitisse a realização da gestão dos recursos de uma dada companhia. Após alguma argumentação sobre qual o tema de maior agrado a ser eleito entre todos os elementos, optou-se pelo tema "Peças de Automóvel".

B. Descrição da Aplicação

A aplicação visa a gestão de uma companhia de peças automóvel, onde conta com um menu para mostrar os produtos disponíveis, neste caso como é uma loja temos o catálogo, conseguimos também ver o estado dos pedidos e adicionamos também as estatísticas da empresa que podemos verificar o total produtos vendidos e o lucro total da empresa. A aplicação que terá sido desenvolvida possui 12 classes sendo uma dessas classes o Menu que vai dar o acesso total às funcionalidades da nossa empresa, tal como fazer pedidos, alterar pedidos, ver o catálogo, etc.

Na **class Carroçaria**, será possível consultar estes tipos de peças: Porta, capô, pára-choques, embaladeira, vidro e os frisos.

Na **class FuncMenu**, será possível listar os produtos disponíveis na loja, que neste caso é o nosso catálogo, também é possível adicionar produtos ao carrinho, verificar se existe ou não o produto pretendido pelo utilizador, mostrar o pedido que está a ser feito, adicionar esse pedido para que a empresa possa ver o que o comprador deseja da loja, mostrar todos os produtos já vendidos até ao momento e por fim mostrar o lucro total da empresa.

Na **class FuncProdutos**, será possível adicionar pneus, sendo possível escolher a marca e o modelo em que irá aparecer o preço e o stock do mesmo. Caso o produto pretendido pelo utilizador não exista no stock, irá aparecer uma mensagem a dizer que não existe stock e pede para o utilizador escolher outra marca/modelo. Será possível também adicionar a iluminação (tipos de iluminação), depois da iluminação podemos também adicionar produtos acerca da carroçaria, jantes, motor e interior, em que todos os produtos pretendidos são todos verificados com as definições já acima escritas.

Na **class Iluminacao**, será possível verificar os métodos da iluminação (getters, setters, toString, equals e clone).

Na **class Interiores**, será possível verificar os métodos do interior(getters, setters, toString, equals e clone).

Na **class Jantes**, será possível verificar os métodos das jantes(getters, setters, toString, equals e clone).

Na **class Ler**, conseguimos ler strings, numeros inteiros, numeros não inteiros(floats), char, short, byte, long, double e boolean.

Na **class Menu**, será disponibilizado um menu ao utilizador em que consegue fazer todas as funções disponíveis na empresa. Será mostrado um menu inicial com 4 funções: Produtos, pedidos, estatísticas e sair, em que a partir dessas funções será possível "navegar" na empresa para que seja possível satisfazer o nosso cliente.

Na **class Motor**, será possível verificar os métodos dos motores(getters, setters, toString, equals e clone).

Na **class Pedido**, será possível verificar os métodos dos pedidos(getters, setters e toString).

Na **class Pneus**, será possível verificar os métodos dos pneus(getters, setters, toString, equals e clone).

Na **class Produto**, será possível verificar os métodos do produto (getters, setters, toString, equals e clone)

C. Solução

Para a realização da nossa proposta o grupo achou que a utilização de doze classes, sendo que uma destas o **Menu**, iria tornar o trabalho mais fácil de interpretar e de se poder alterar. As classes existentes para além do **Menu** são a "Carroçaria", "FuncMenu", "FuncProdutos", "Iluminação", "Interiores", "Jantes", "Ler", "Motor", "Pedido", "Pneus", "Produto".

A **classe Carroçaria** possui os seguintes parâmetros:

- Marca;
- Tipo;

A **classe FuncMenu** Serve para guardar os vários métodos seguintes listados:

- delay;
- Método mostrar produtos;
- Método adicionar produto;
- Método remover produto;
- Método mostrar pedidos;
- Método adicionar pedido;
- Método mostrar produtos vendidos;
- Método mostrar lucro.

A **classe FuncProdutos** Serve para guardar os vários métodos seguintes listados:

- Método adicionar pneu;
- Método adicionar iluminação;
- Método adicionar carroçaria;
- Método adicionar jantes;
- Método adicionar motor;
- Método adicionar interior.

A **classe Iluminação** possui os seguintes parâmetros:

- Tipo;
- Cor;
- Intermitente.

A **classe Interiores** possui os seguintes parâmetros:

- Tipo;
- Cor;
- marcaInterior.

A **classe Jantes** possui os seguintes parâmetros:

- Tipo;
- marcaJantes;
- Cor.

A **classe Ler** possui os seguintes parâmetros:

- Ler uma string;
- Ler um int;
- Ler um char;
- Ler um short;
- Ler um byte;
- Ler um long;
- Ler um float;
- Ler um double;
- Ler um boolean.

A **classe Menu** mostra os seguintes menus:

- Menu Inicial;
- Menu dos Produtos;
- Menu dos Pedidos;
- Menu das estatísticas.

A **classe Motor** possui os seguintes parâmetros:

- Tipo;
- marcaMotor;
- referenciaMotor.

A **classe Pedido** possui os seguintes parâmetros:

- Nome;
- Data;
- Hora;
- Produtos;
- Quantidades;
- preçoFinal.

A **classe Pneus** possui os seguintes parâmetros:

- tipo;
- marcaPneu.

A **classe Produto** possui os seguintes parâmetros:

- Ultimo;
- Id;
- Marca;
- Modelo;
- Preço;
- Stock;

A classe *FuncMenu* é a que permite a criação da base de dados da aplicação. Sendo esta a responsável pela escrita de novos elementos num ficheiro *.dat* e leitura de elementos existentes no ficheiro *.dat* correspondente ao pedido.

D. Defesa

O trabalho desenvolvido pelo grupo tem, como base, o conteúdo lecionado ao longo das aulas teóricas e práticas da cadeira, tendo havido a necessidade de fazer uma escolha daqueles que melhor se adequam ao nosso projeto.

Devido a isto o grupo decidiu criar 12 (doze) classes, incluindo a *Menu*, pois esta revelou ser a forma mais fluida e prática do grupo trabalhar. Deste modo, no caso de ser necessário alterar algum atributo específico de uma classe, não teríamos de estar a procurar onde estaria essa classe, nem correríamos o risco de alterar a classe errada.

IV. CONCLUSÃO

A. Trabalho Futuro

Tendo concluído os requisitos essenciais para que o projeto pudesse funcionar sem problemas, o grupo concluiu que este trabalho final possui bastante potencial em termos de funcionamento de uma empresa de peças para automóveis, sendo mesmo possível implementar o mesmo numa empresa na realidade, tendo em conta que se teriam que fazer algumas alterações para que o mesmo possa funcionar sem qualquer problema.

B. Conclusões

O desenvolvimento deste projeto foi bastante importante para o grupo ganhar e solidificar conhecimentos na área de POO, em termos de programação e no desenvolvimento do tema. Este permitiu-nos também ter uma ideia de todo o esforço, trabalho e pormenor existente por detrás da criação de uma aplicação como esta.

Futuramente, numa empresa, o trabalho em equipa é algo indispensável para o sucesso, pois desta maneira é possível solucionar um determinado problema mais facilmente, demorando menos tempo do que aquele previsto para a realização de uma determinada tarefa, e tornando o trabalho desenvolvido mais eficiente, não esquecendo que o trabalho individual também é deveras importante pois algumas pessoas conseguem trabalhar melhor sozinhas do que em grupo devido ao nível de concentração das mesmas.

De maneira geral, e tendo em conta os conhecimentos dos elementos do grupo, o mesmo ficou deveras satisfeito com o resultado do trabalho final.