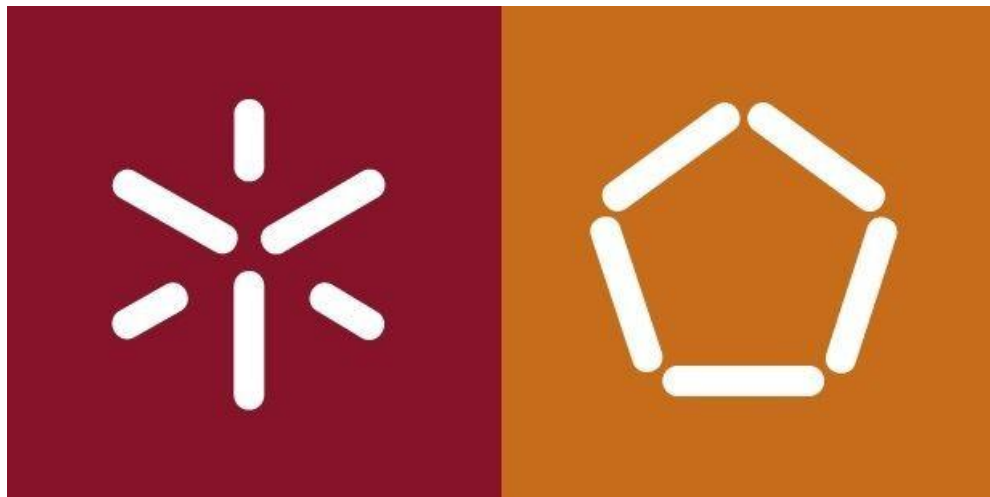


# Trabalho Prático de Programação Orientada aos Objetos

26/05/2019



## Grupo 21

António Santos - A83700

Pedro Oliveira - A83762

Bernardo Graça -

## Visão geral

Neste trabalho construímos um protótipo de um serviço de aluguer de veículos particulares com recurso á linguagem de programação Java.

Na aplicação desenvolvida um proprietário de um automóvel poderá registar o seu veículo na aplicação “UMCarroJá”, e este pode ser alugado por um cliente registado nesta mesma aplicação. A aplicação permite também a gravação e reposição do seu estado.

## Arquitetura do Projeto

Classes abstratas:

- Carro
- Actor

Classes regulares:

- Cliente
- Proprietário
- Carro Elétrico
- Carro Híbrido
- Carro Gasolina
- Menu
- Estado
- LogWriter
- UMCarroJá (main)

Interfaces:

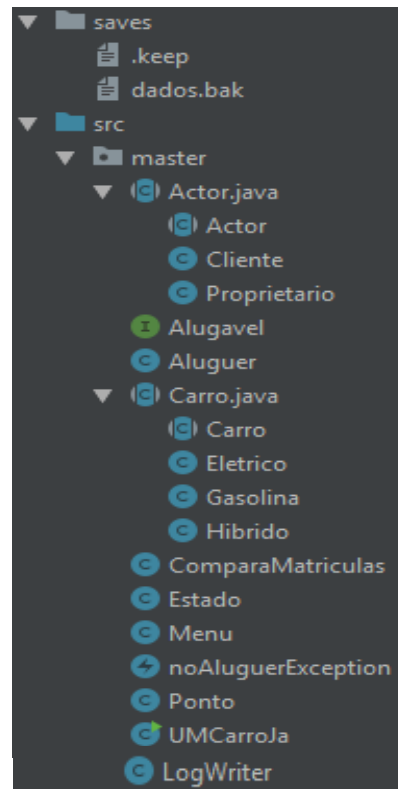
- Alugavel

Compares:

- ComparaMatriculas

Exceções:

- NoAluguerException



Ao desenvolver este projeto vimo-nos deparados com várias decisões sobre a forma como iríamos implementar diversas funcionalidades.

O cérebro da aplicação, ou seja onde temos a main é a classe “UMCarroJá” onde temos dois carregamentos de estado, um automático com várias exceções, e um em formato de texto, e, no fim da execução do programa, o gravar do estado, mais uma vez, um automático, e um em formato de texto.

Uma das primeiras decisões foi de como implementar os menus de texto para o utilizador poder fazer as suas escolhas e poder navegar pelo programa com facilidade. Decidimos criar uma Classe que cria um menu, e que executa o mesmo menu e só para de o executar quando for escrito o ‘0’ para sair do menu, ou quando for escolhida uma das opções.

De seguida decidimos como implementar o estado para “guardar” todos os clientes e proprietários, e acabamos uma Classe estado com três Maps, um para guardar os proprietários, um para guardar os clientes, e um para guardar os carros. O mapa de carros foi criado por conveniência para depois facilitar nas queries de carros com determinadas características e com determinadas especificações a escolher pelo utilizador.

Temos uma interface denominada de Alugavel para facilitar a possível implementação de futuros veículos e para mais facilmente serem adicionados ao estado e ao sistema.

Como comparador temos um comparador para organizar os carros que o proprietário tem, para assim facilitar a sua visualização pelo proprietário.

Por fim temos uma exceção para conseguir fazer a gestão de classificações do cliente, ou seja, temos no cliente um aluguer que pode ou não existir, dependendo se já classificou ou se ainda não foi confirmado o aluguer por parte do proprietário, e para sabermos utilizamos uma exceção e tratamos dela para saber se o cliente pode ou não classificar um determinado aluguer.

## Descrição da aplicação desenvolvida

A aplicação ao ser executada apresenta-nos com um menu com quatro opções:

- Continuar
  - Continuar para o programa, sem gravar ou carregar estado, neste caso será utilizado o estado com que o programa saiu da ultima vez, ou, se é a primeira execução o programa irá começar com um estado a limpo, este também é utilizado depois de carregar ou gravar para utilizar os dados carregados.
- Carregar Estado
  - Permite carregar um estado a partir de um ficheiro de logs, pode ser utilizado um nome pré-definido ou ser utilizado um nome introduzido pelo utilizador, utiliza a classe LogWriter que simplifica a escrita no ficheiro pretendido.
- Gravar Estado
  - Permite gravar um estado num formato de ficheiro de logs, pode ser utilizado um nome pré-definido ou ser utilizado um nome introduzido pelo utilizador, utiliza a classe LogWriter que facilita na gravação do ficheiro pretendido.
- Sair
  - Opção comum a todos os menus para sair do menu atual, neste caso, como este é o ultimo menu, irá fazer com que saia do programa (gravando o estado automaticamente).

```
*** Menu ***
1 - Continuar ->
2 - Carregar Estado
3 - Gravar Estado
0 - Sair
Opção:
```

Ao selecionar Continuar somos apresentados com outro menu em que temos 3 opções principais:

- Login
  - Vai para o menu de Login.
- Registo
  - Vai para o menu de registo onde é possível registar tanto um cliente como um proprietário, através de informações introduzidas pelo utilizador, o registo não permite utilizações de emails duplicados, e permite que um utilizador seja simultaneamente um cliente e um proprietário (mas terá de fazer login separadamente).
- 10 Cliente que mais utilizam o sistema
  - Procura na lista de clientes os 10 clientes que mais utilizam o sistema

```
*** Menu ***
1 - Login
2 - Registo
3 - 10 Clientes que mais utilizam o sistema
0 - Sair
Opção:
```

Depois de feito o login, se feito como Cliente somos apresentados com o seguinte menu:

```
Login efetuado com sucesso!
*** Menu ***
1 - Solicitar o aluguer do carro mais próximo das suas coordenadas.
2 - Solicitar o aluguer do carro mais barato.
3 - Solicitar o aluguer do carro mais barato dentro de uma distância que esteja disposto a percorrer a pé.
4 - Solicitar o aluguer de um carro específico.
5 - Solicitar o aluguer de um carro com uma autonomia desejada.
6 - Verificar alugueres recentes.
7 - Classificar aluguer recente.
0 - Sair
Opção:
```

Onde cada opção corresponde aos requisitos pedidos pelos docentes da disciplina e, nos casos de solicitar o aluguer, se houver carros disponíveis, será solicitado ao proprietário do veículo para aceitar ou registar o pedido de aluguer.

Neste menu também é possível visualizar os alugueres recentes, bem como classificar o aluguer feito mais recentemente, se o aluguer solicitado for aceite pelo proprietário.

Se feito como Proprietário somos apresentados com o seguinte menu:

```
Login efetuado com sucesso!
*** Menu ***
1 - Registrar novo Carro
2 - Sinalizar que um dos seus carros está disponível ou não para alugar
3 - Abastecer um veiculo
4 - Alterar o preço por km de um veiculo
5 - Aceitar/rejeitar o aluguer de um determinado cliente
6 - Registrar quanto custou a viagem
7 - Verificar alugueres recentes
8 - Total faturado por uma viatura
0 - Sair
Opção:
```

Neste menu temos a opção de registar um novo carro, a Gasolina, Elétrico ou Híbrido.

Temos também a opção de sinalizar que um dos nossos carros está disponível ou não para alugar, ou seja, se pretendermos que um carro fique indisponível por diversas razões esta opção permite fazê-lo.

Podemos através deste menu abastecer os nossos veículos ou alterar o seu preço de por quilómetro.

Case tenhamos pedidos de aluguer pendentes, podemos tanto aceitá-los, como rejeitá-los, depois de o aluguer ter sido concluído podemos classificar o cliente.

Podemos registar o quanto custou o aluguer feito mais recentemente, podemos também verificar os alugueres mais recentes e, por fim, o total faturado por uma determinada viatura, devolve 0.0 caso a viatura não tenha sido utilizada nenhuma vez (ou seja, não tenha alugueres).

## Como seria possível incluir novos tipos de viatura

Para facilitar a introdução de novos veículos no nosso programa nós criamos uma interface “Alugavel”, (para além da Serializable para permitir a gravação automática do objeto) que a classe abstrata carro implementa, ou seja, todas as classes que estenderem a classe carro, terão, por sua vez, implementar todos os métodos definidos na interface “Alugavel”.

Nesta Interface temos os seguintes métodos para serem implementados:

```
/** abastece o veículo com uma determinada quantidade */  
public void abastecer(double quantidade);  
/** calcular o preço de uma viagem */  
public double calcularPreco(double distancia);  
/** dar update ao carro depois de um aluguer */  
public void updateCarro(Aluguer a);  
/** classificar um carro */  
public void classificarCarro(int classificacao);
```

Abastecer para permitir o veículo ser abastecido (visto que uma das condições de funcionamento deste programa é o proprietário poder abastecer os seus carros.

Calcular o preço a partir de uma distancia, pois se um carro vai ser alugado, tem que custar um valor e este valor varia de acordo com a distancia, para, por exemplo, um carro com um determinado preço por quilómetro, e um determinado consumo varia de carro para carro.

Atualizar o carro depois da viagem, cada carro de acordo com o seu consumo e posição deve ser atualizado depois de o aluguer ser terminado.

Classificar o carro, o cliente poder classificar cada veículo como pretender.

De seguida apenas seriam necessárias pequenas alterações no código para permitir a criação de veículos do novo tipo, e pequenas mudanças nas funções “hubCliente” e “hubProprietario” para permitir o correto funcionamento do novo veículo no sistema.

Seria também necessária a alteração do sistema de logs e carregamento de dados, mas seriam limitadas a pequenas alterações.