

Relatório do projeto de POO

UMCarroJá!

Grupo 38



Mafalda Colaco Parente
Morais da Costa

Número: a83919



Pedro Miguel Araújo
Costa

Número: a85700



Ulisses Tiago Simões Araújo

Número: a84010

Conteúdo

Introdução	3
Classes	4
Diagrama de classes.....	6
Principais estruturas de dados	7
Manual de utilização	8
Conclusões.....	10

Introdução

Foi-nos proposto, no âmbito da cadeira de Programação Orientada aos Objetos, que desenvolvêssemos uma aplicação de aluguer de veículos particulares (como a Uber, por exemplo) cujo nome é UMCarroJá! Para tal efeito, construímos várias classes particulares (as quais serão abordadas, mais profundamente, a seguir) capazes de representar o comportamento de cada parte necessária ao programa (veículos, clientes, utilizadores, etc...). Através de módulos superiores, que armazenam toda a informação relativa a cada entidade do programa, fomos capazes de trabalhar com a mesma, produzindo o efeito desejado. Nos pontos seguintes deste relatório iremos abordar, essencialmente, a composição de cada classe, o diagrama que correlaciona as mesmas, um pequeno manual de utilização do nosso programa e, por fim, o que poderia ter sido melhorado.

Classes

Aluguer: Contém na sua estrutura a matrícula do veículo utilizado, o email do cliente, a data da realização do aluguer, o custo da viagem, a distância percorrida pelo veículo, o tempo de viagem, a localização destino, o tipo do carro utilizado e a preferência (Mais barato, mais perto, etc). Para esta simulação como não eram fornecidas datas no ficheiro de logs achamos melhor utilizar um gerador aleatório de datas que produz uma data entre 2018 e 2019.

Localização: Uma simples classe que contém apenas as coordenadas 2D e um método para calcular a distância entre duas localizações.

Main: Classe simples que apenas dá ao utilizador a opção de escolher entre carregar o programa de um ficheiro de estado ou de um ficheiro de logs e invoca posteriormente o menu.

Menu: Classe que contém o método estático menu a ser invocado pela main.

Utilizador: Contém o número de criação do utilizador, o seu email, NIF, password, morada, nome e data de nascimento. É a super classe das classes Proprietário e Cliente.

Proprietário: É subclasse da classe Utilizador e, adicionalmente, contém uma classificação, um número de classificações, uma lista de alugueres realizados, assim como uma lista de alugueres em espera de aprovação e um mapa com os seus veículos indexados por matrícula.

Cliente: Subclasse da classe Utilizador que adicionalmente contém a sua localização, uma lista de alugueres, uma classificação, um número de classificações, uma lista com as matriculas dos veículos ainda por classificar (após o aluguer ser aprovado pelo proprietário automaticamente é inserida a matrícula do respetivo veículo nesta lista para posteriormente o proprietário e o veículo terem a avaliação dada pelo cliente registada).

Veículo: Contém o email do proprietário, a matrícula do veículo, a velocidade em km/hora, o preço por km, a classificação do veículo, o número de classificações e a lista de alugueres realizados. É também a superclasse da classe Carro.

Carro: Subclasse da classe Veículo, contém o tipo de carro (Gasolina, Elétrico ou Híbrido), o consumo por km, a sua marca, o modelo, a quantidade de combustível atual/máxima em litros, e a sua autonomia atual/máxima em quilómetros e, por fim, a sua localização.

A partir da autonomia atual/máxima e o consumo por km calculamos as quantidades de combustível atual/máxima.

gestaoUtilizadores: Classe que contém um mapa de proprietários e um mapa de clientes ambos indexados por email.

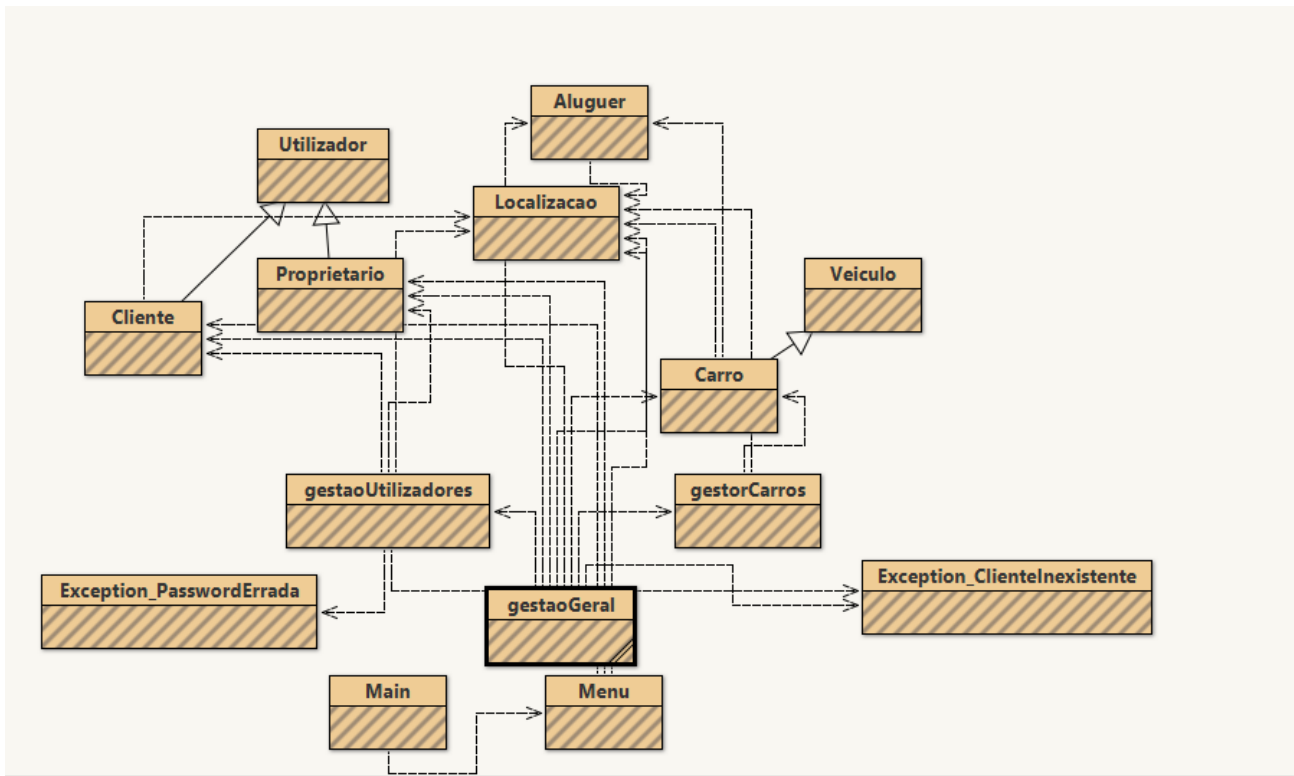
gestorCarros: Classe que contém um mapa de carros indexados por matrícula.

gestaoGeral: Classe que junta as classes gestaoUtilizadores e gestorCarros facilitando e possibilitando a manipulação dos dados.

Exception_PasswordErrada: Simples exceção usada para falhar o login quando a password está errada.

Exception_ClienteNaoExiste: Exceção usada quando o utilizador que se procura não existe.

Diagrama de classes



Principais estruturas de dados

As principais estruturas de dados neste projeto encontram-se em uso nos módulos de gestão, nomeadamente, na gestão de utilizadores, de carros e, como seria de esperar, na gestão geral.

Maioritariamente, tomamos partido de mapas de chave valor pois permitem manter uma correspondência entre clientes/proprietários e o seu email, assim como veículos e a sua matrícula. Além disso, acreditamos veemente que, com o uso destas mesmas estruturas, torna-se possível alcançar a maior eficiência nas diversas operações que realizamos ao longo do nosso projeto.

Manual de utilização

Ao correr o programa pela primeira vez é dada ao utilizador a opção de escolher entre carregar a partir de um ficheiro de logs ou ler de um ficheiro de estado previamente guardado, e, caso a opção dada seja inválida é carregado um ficheiro de logs por default.

Posteriormente são dadas 3 opções, login de um utilizador já existente, criar um novo utilizador (seja proprietário ou cliente) ou verificar o top 10 dos clientes (que mais gastaram ou que mais vezes utilizaram o programa).

A criação dos utilizadores pede todos os dados necessários e, posteriormente, coloca-os no programa voltando então a este segundo menu (é importante lembrar que é impossível a criação de um utilizador do mesmo tipo com o mesmo email).

Sendo escolhida a opção de login é dada a escolha de dar login de um cliente ou de um proprietário. Conforme o tipo de utilizador, as opções fornecidas pelo menu variam.

Para os clientes existem as seguintes possibilidades:

- Imprimir a lista de todos os carros de um certo tipo.
- Um aluguer default sem preferência específica de carro, pede todas as informações relevantes quanto ao tipo do carro desejado, se é carro mais barato ou carro mais perto, etc.
- Aluguer com preferência de carro. (Dada a matrícula realiza o aluguer se o carro existir e estiver disponível para aluguer).
- Aluguer do carro mais barato dentro de uma distância dada pelo utilizador.
- Imprimir a lista de todos os alugueres realizados por este cliente.
- Dadas duas datas, imprimir a lista dos alugueres realizados entre as mesmas.
- Apresentar a primeira classificação em espera, após concluir uma viagem.
- Guardar o estado do programa para um ficheiro binário.
- Dá logout do utilizador.

Para os proprietários existem as possibilidades de:

- Imprimir a lista de carros pertencentes a esse proprietário.
- Abastecer um carro dando a matrícula e quantidade de km.
- Adicionar uma nova viatura à sua lista de veículos.
- Guardar o estado do programa para ficheiro.
- Imprimir os alugueres realizados pelos carros deste proprietário.
- Aceitar ou rejeitar o primeiro aluguer na lista de espera.
- Imprimir todos os alugueres realizados pelos carros deste proprietário entre duas datas.
- Trocar a disponibilidade de aluguer de um carro deste proprietário.
- Dada a matrícula de um veículo e duas datas obter o total faturado por esse veículo entre essas as mesmas.
- Alterar o preço por km de um veículo.
- Dar logout do utilizador.

Conclusões

No futuro deste projeto seria possível adicionar a existência de outros tipos de veículos, como sugerido no enunciado do projeto. Para esse efeito, seria apenas necessário definir outras subclasses da classe veículo e alterar o gestor de carros para um gestor de veículos. Era ainda prático fazer algumas alterações no código de modo a tirar o máximo proveito da flexibilidade que a herança de classes nos traz. A classe veículo seria, ainda, implementada como uma classe abstrata, obrigando cada veículo a disponibilizar as operações necessárias.