



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Programação Orientada a Objetos
Licenciatura em Engenharia Informática
2º ano – 1º semestre
Universidade de Coimbra

Relatório

Gestor de Viagens



Ricardo Martins – 2017246268

Tiago Ferreira Fernandes – 2017242428

PL 4 – 07/12/2018

Introdução:

A realização deste projeto teve como objetivo implementar uma aplicação para gerar viagens com base nas preferências dos utilizadores.

Para tal, foi desenvolvido um diagrama UML para apoiar na implementação da aplicação, que contém as classes do projeto e respetivas hierarquias. Este diagrama foi entregue na data pretendida (meta 1), porém iremos apresentar o diagrama final, após a conclusão da aplicação.

Após a entrega do diagrama inicial, começou-se a implementação da aplicação, composta do seguinte modo:

➔ Classes:

- Aluno
- Bar
- Handle_Ficheiros
- Horario
- Licenciatura
- Local
- Mestrado
- Parque
- Parque_Aquático
- Parque_Cultural
- Parque_Diversoes
- Parque_Tematico
- PontoInteresse
- Universidade
- Viagem

A explicação destas classes vai ser realizada mais à frente.

Por fim, teremos a nossa classe principal, onde trata a gestão das viagens:

➔ Viagens_POO

De seguida, iremos passar à explicação dos conteúdos descritos em cima.

Classes:

Para todas as classes, existe os respetivos *getters* e *setters* para cada argumento da classe.

→ Aluno:

- A classe Aluno contém o nome, o custo máximo que pretende gastar na viagem e o grau académico (licenciatura ou mestrado) do aluno em causa.

→ Licenciatura:

- A classe Licenciatura, além da informação do nome, custo máximo grau académico, também tem um ponto *hot*, isto é, o ponto de interesse que o aluno de licenciatura não quer perder

→ Mestrado:

- A classe Mestrado, além da informação do nome, custo máximo grau académico, também tem um local a evitar, isto é, o local que o aluno de mestrado não quer visitar

→ Handle_Ficheiros:

- A classe Handle_Ficheiros é onde estão todos os métodos relativos à leitura e escrita de ficheiros de objetos e de texto

→ Horario:

- A classe Horario contém as horas de início e as horas de fim de um dado ponto de interesse

→ Viagem:

- Para a classe Viagem, vai haver um conjunto de locais onde o aluno vai e um conjunto de pontos de interesse que vai visitar, sabendo também o preço total da viagem e a distância total a ser percorrida

→ Local:

- Na classe Local, existem vários argumentos como o nome do local, um conjunto de pontos de interesse (ArrayList), um gastoLocal (gasto total do local), um contador para a sua popularidade e um índice para conseguirmos saber a distância desse local aos outros todos, através de uma matriz com as distâncias todas

→ PontoInteresse:

Para cada ponto de interesse criado, existe um nome, um horário de funcionamento, um contador relativo à sua popularidade e um custo total gasto nesse ponto de interesse (preço de entrada, despesas dentro do ponto de interesse e preço de espetáculos). Todos estes argumentos estão associados às classes descendentes desta:

- **Museu:**
 - Na classe Museu, além da informação referente ao ponto de interesse, existe um tema do museu e o preço de entrada
- **Universidade:**
 - A cada universidade, além da informação referente ao ponto de interesse, existe um conjunto de cursos relacionados com Engenharia Informática
- **Bar:**
 - Cada bar tem a si associado uma classificação média dos clientes do bar, o preço de entrada, o valor médio das despesas e ainda mais as informações de um ponto de interesse
- **Parque:**

Os parques têm todos um preço de entrada e podem ser de vários tipos:

 - **Parque_Cultural:**
 - Os parques culturais apenas têm associado a ele as mesmas informações que o parque tem
 - **Parque_Diversoes:**

Os parques de diversões apenas têm a eles associados as mesmas informações que o parque tem. Estes podem ser de 2 tipos:

 - **Parque_Tematico:**
 - Estes parques, além das mesmas informações que os parques de diversões, têm um tema associado

- **Parque_Aquatico:**

- Os parques aquáticos têm informações sobre o número de piscinas e escorregas que têm, se existem espetáculos ou não, o preço dos espetáculos e ainda as mesmas informações que os parques de diversões

Ficheiros:

→ Ficheiros de texto:

- **Ficheiro dos locais:**

- O ficheiro de texto dos locais está organizado da seguinte maneira:
Local/Bar/Universidade/Museu/Parque

- **Ficheiro das distâncias:**

- O ficheiro das distâncias apenas contém uma matriz com as várias distâncias entre os vários locais

→ Ficheiros de objetos:

- **Ficheiro dos locais:**

- O ficheiro de objetos dos locais contém o ArrayList dos locais que fora previamente lido do ficheiro de texto dos locais

- **Ficheiro das distâncias:**

- O ficheiro de objetos das distâncias com a matriz com as distâncias entre os vários locais que fora previamente lido do ficheiro de texto das distâncias

- **Ficheiro dos alunos:**

- O ficheiro de objetos dos alunos é criado após ser feito o registo do primeiro aluno. Este contém o ArrayList dos alunos todos que estão registados no programa

Diagrama de classes inicial:

O diagrama inicial já foi explicado anteriormente, na primeira meta, não sendo necessária uma breve explicação sobre o mesmo.

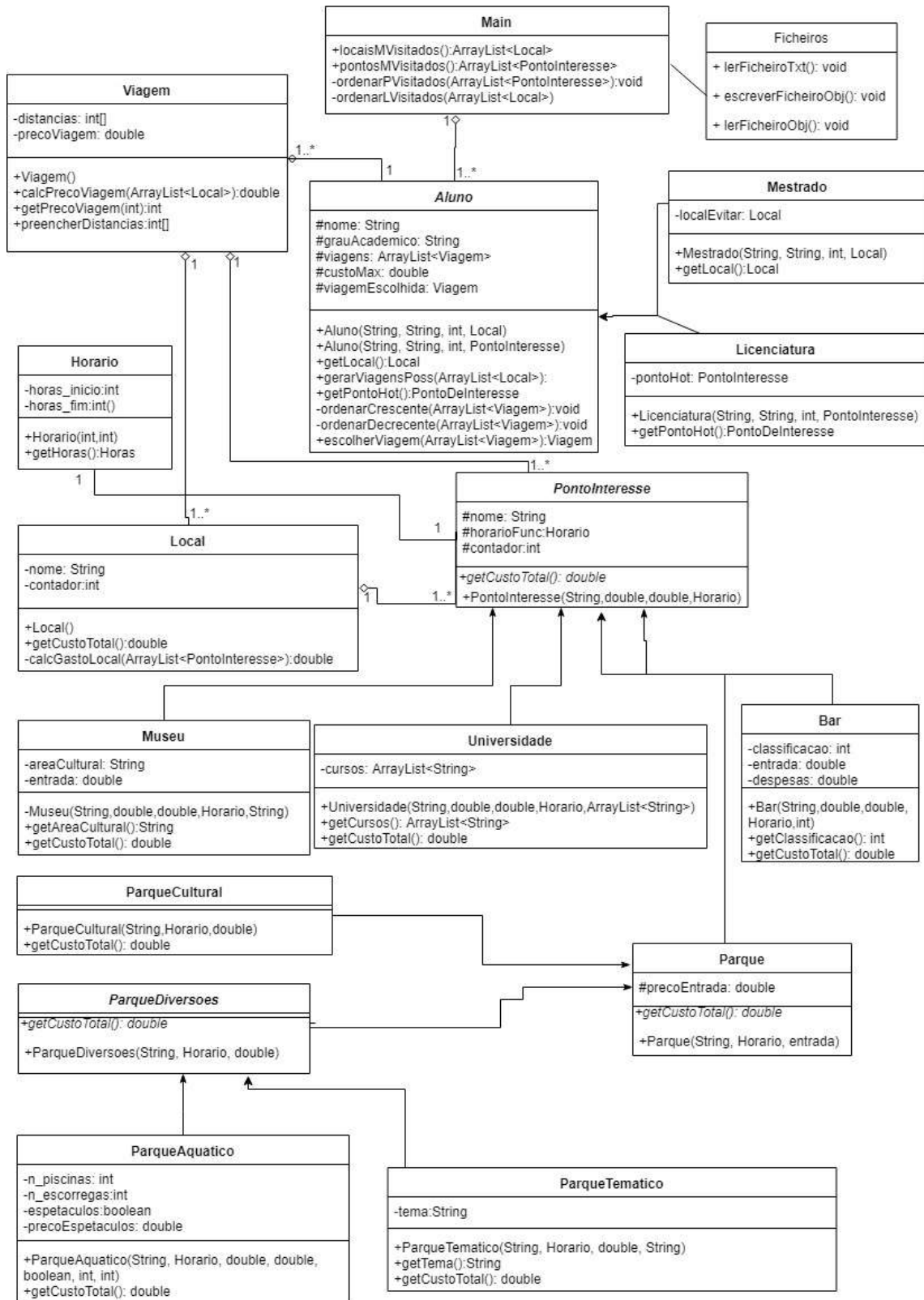
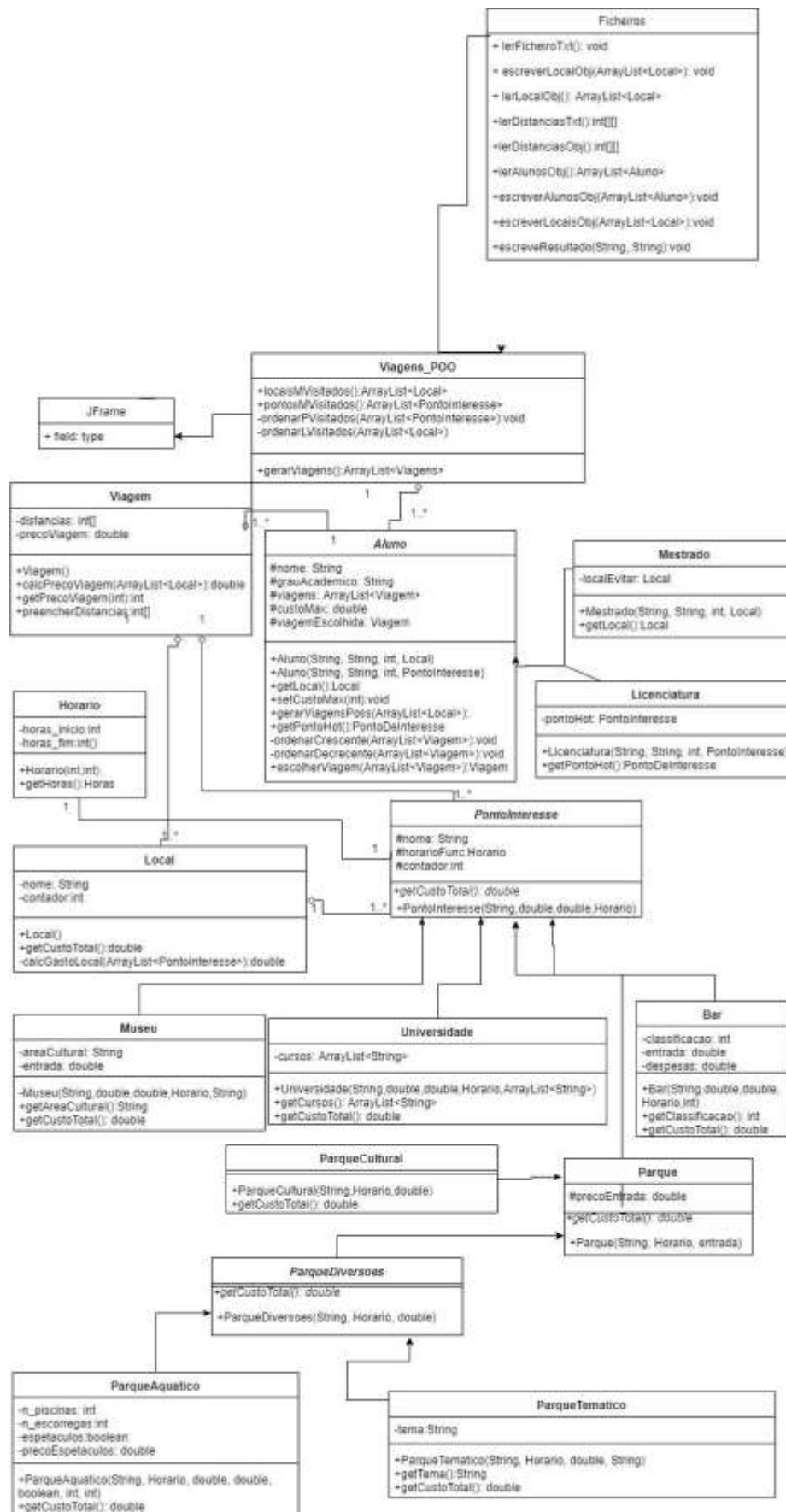
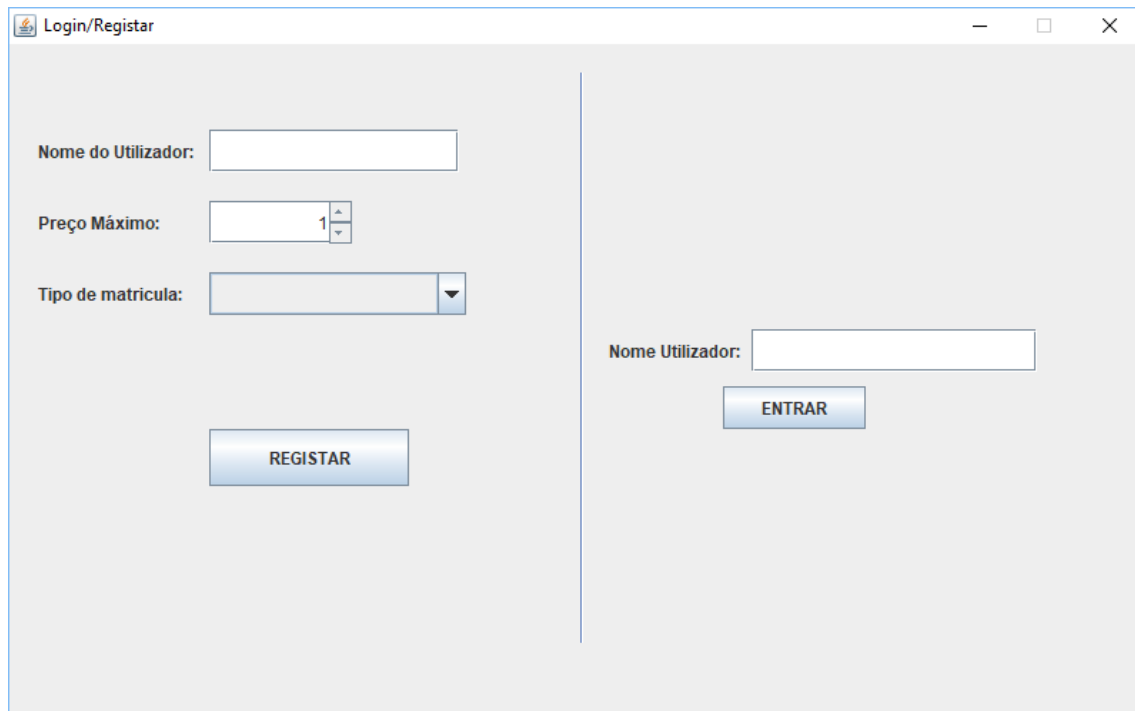


Diagrama de classes final:

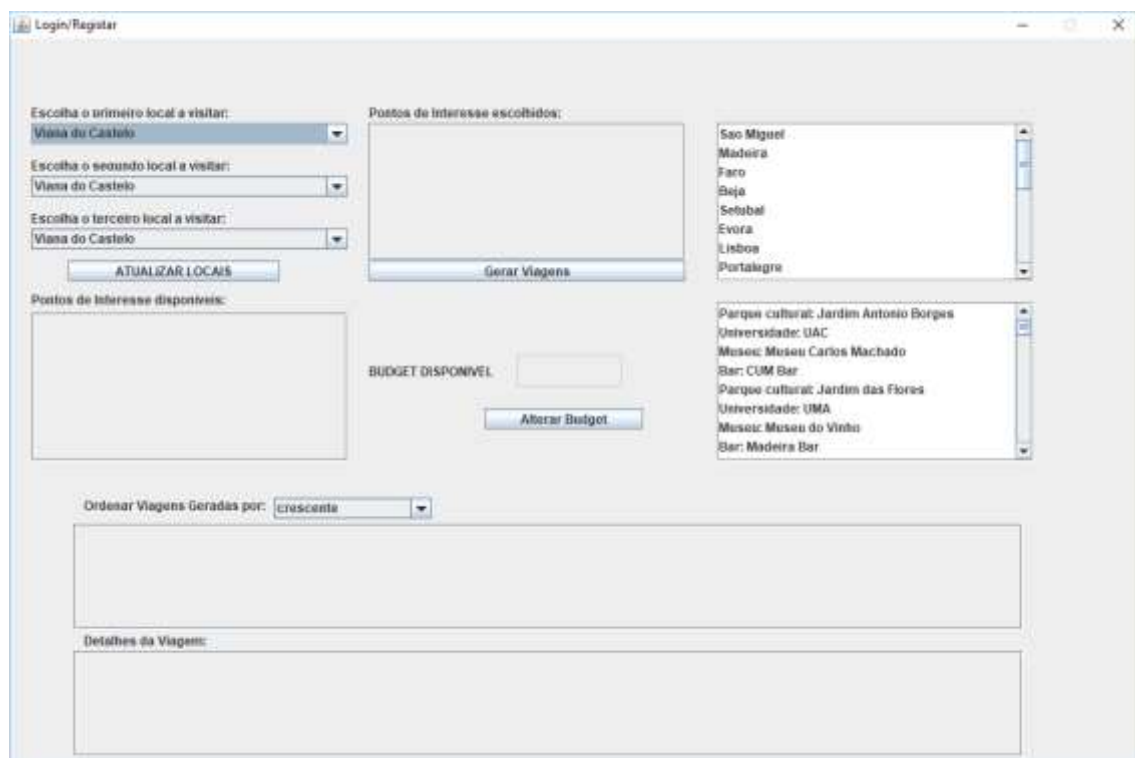


Manual de utilizador:

Para correr o projeto, apenas é necessário abrir o executável (viagens_poo.jar). De seguida, vai aparecer uma janela para efetuar o *login* ou o registo (caso não exista nenhum utilizador registado no sistema).



Após o registo de um utilizador e o devido *login*, o utilizador terá acesso ao menu principal onde poderá começar a realizar a sua viagem:



Primeiramente, o utilizador começa por escolher os locais que pretende visitar e depois clica em “ATUALIZAR LOCAIS”:

- ➔ Se for um aluno de Licenciatura, o primeiro local vai estar bloqueado no local de onde o ponto *hot* é;
- ➔ Se for um aluno de Mestrado, o local a evitar nem sequer vai aparecer nas opções das escolhas dos locais

De seguida, o utilizador vai escolher os locais que pretende visitar dentro dos possíveis que existem nos locais selecionados (são escolhidos um a um). Após ter escolhidos todos os locais, basta carregar em “Gerar viagens” e o programa vai gerar as viagens possíveis dentro das escolhas do utilizador e dentro do limite que definiu como sendo o seu montante máximo, sendo também obrigatório passar por um museu.

Por fim, o utilizador vai escolher a viagem que pretende realizar, sendo depois apresentado as informações relativas aos pontos de interesse que ele vai visitar na sua viagem. Esta informação vai ser num ficheiro de texto cujo o nome do ficheiro é o que o utilizador pretender.