

FCT Boleia

Relatório de Algoritmos e Estruturas de Dados 2019/20 (MIEI)

Alunos : José Murta (55226) e Diogo Rodrigues (56153)

Curso: Mestrado Integrado em Engenharia Informática

Turno: P7 e P1

Docente prático: Sofia Cavaco

Boleia

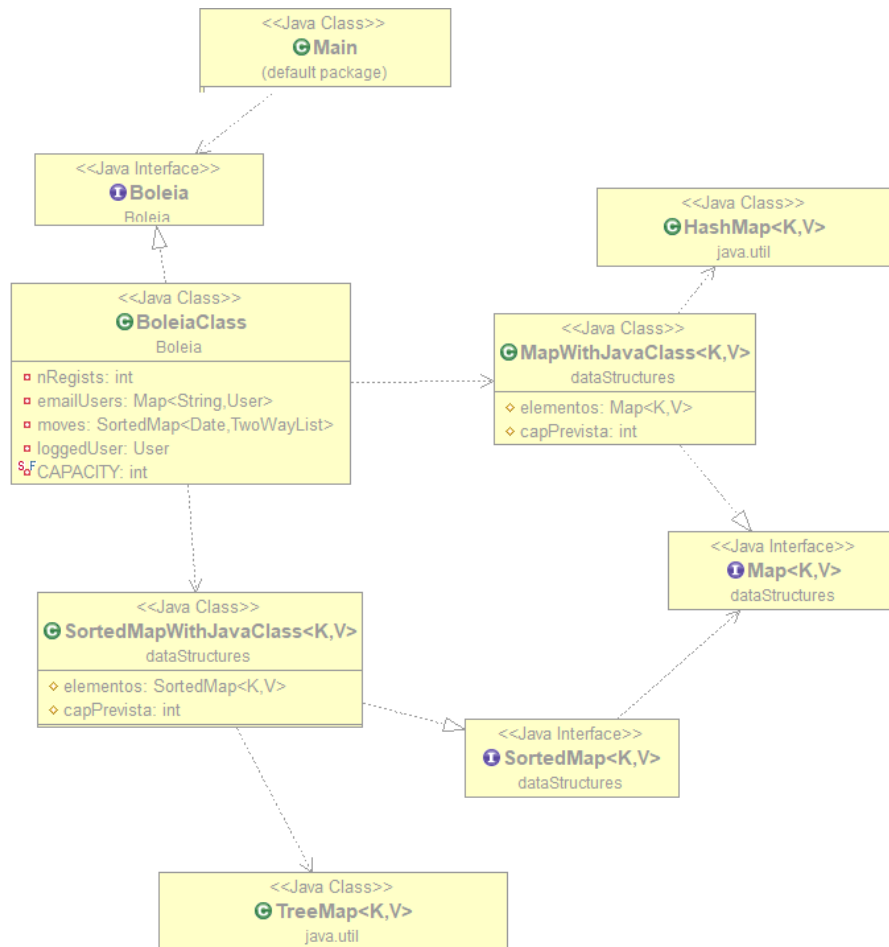


Figura 1 - Diagrama de Main e Classe topo

A interface **Boleia** representa a aplicação da **FCT Boleia**. A classe **BoleiaClass** necessita de guardar informação acerca do número de registos (nRegists), o utilizador que tem/não tem sessão iniciada na aplicação (loggedUser), de todos os utilizadores registados (emailUsers) e de todas as deslocações registadas(moves).

Como estruturas de dados utilizamos um SortedMap para organizar todas as deslocações por ordem cronológica, sendo a *Key* a interface Date e o *Value* uma lista duplamente ligada ordenada de deslocações e um Map para armazenar todos os utilizadores da aplicação, sendo a *Key* uma String, que é o email do utilizador, e o *Value* o utilizador registado com esse email. Optámos por utilizar estes TADs visto que precisamos de acesso constante ao utilizador, com o seu identificador, que é o email, e todas as nossas deslocações ordenadas cronologicamente(1) e por email do dono da deslocação(2).

User

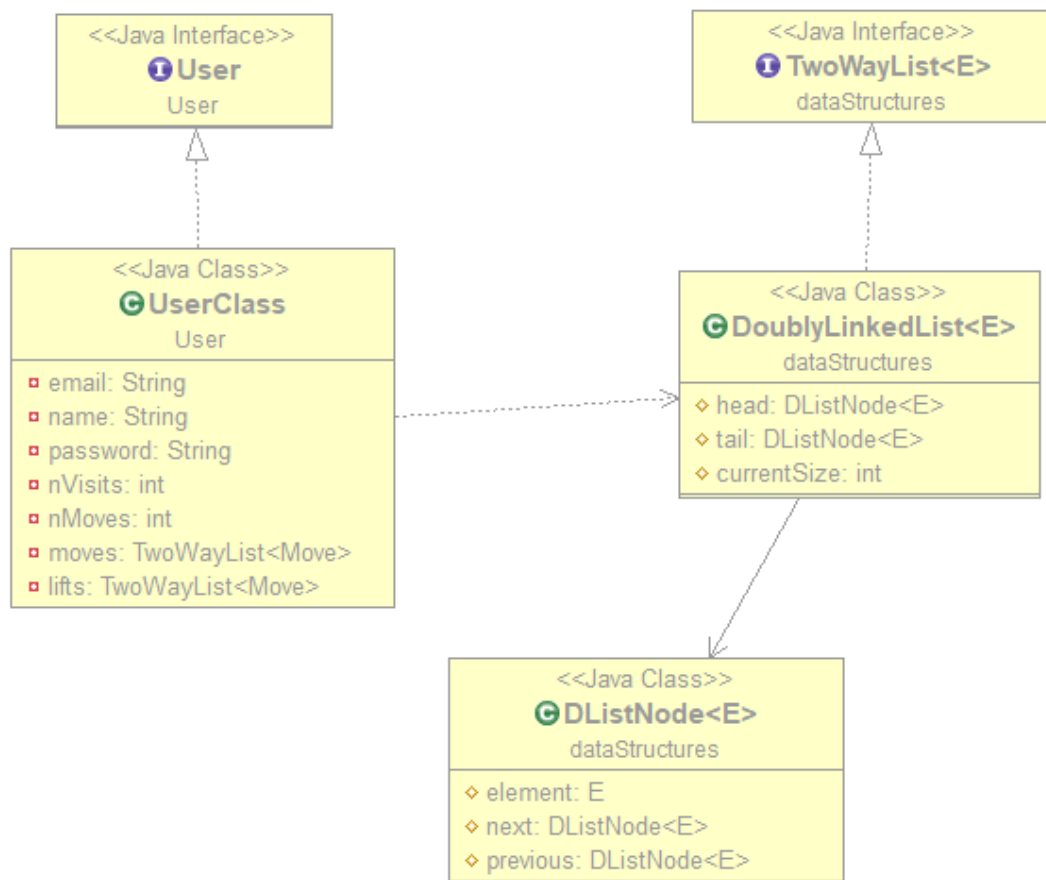


Figura 2- Diagrama de Classe User

A interface **User** representa um **utilizador** da aplicação. A classe **UserClass** necessita de armazenar informação acerca do email (email), o nome (name), e password (password) do utilizador. Além disso, deve também guardar o número de vezes que o utilizador iniciou sessão na aplicação (nVisits), número de deslocações registadas (nMoves), deslocações (moves) e as boleias em que o utilizador se encontra inserido (lifts).

Optámos por utilizar listas duplamente ligadas pois as deslocações e boleias do utilizador são constantemente adicionadas/removidas e este tipo de TADs facilitam e tornam eficiente a inserção e remoção de Objetos.

Move

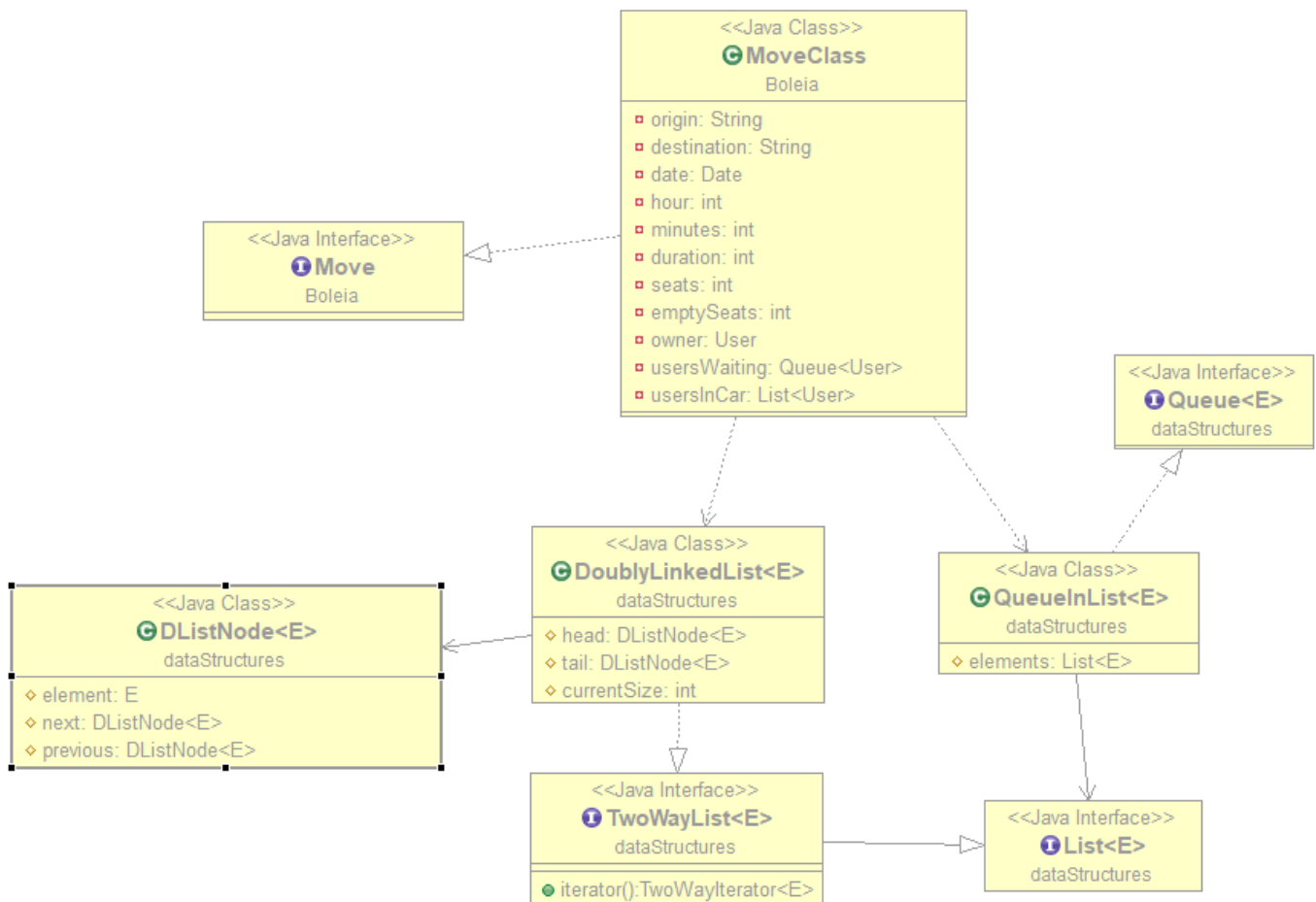


Figura 3 - Diagrama de Classe Move

A interface **Move** representa uma **deslocação**. A classe **MoveClass** necessita de guardar informação sobre a origem (`origin`), o destino (`destination`), a data com dia-mês-ano (`date`), o horário com horas e minutos (`hour` & `minutes`), duração (`duration`), o número de lugares total no veículo (`seats`), o número de lugares disponíveis (`emptySeats`), e o utilizador que criou a deslocação (`owner`). Além disso, deve também armazenar os utilizadores que estão inscritos na boleia (`usersInCar`) e os que estão em fila de espera para a ocasião de algum utilizador desistir da boleia (`usersWaiting`).

Como TADs utilizámos uma lista duplamente ligada para os utilizadores com boleia reservada e uma Queue para os que estão em fila de espera. Optámos por utilizar estas estruturas de dados pois a fila de espera tem disciplina FIFO com constantes inserções e remoções (implementação com Lista) e por essa mesma razão o uso de uma lista duplamente ligada para os utilizadores com boleia registada.

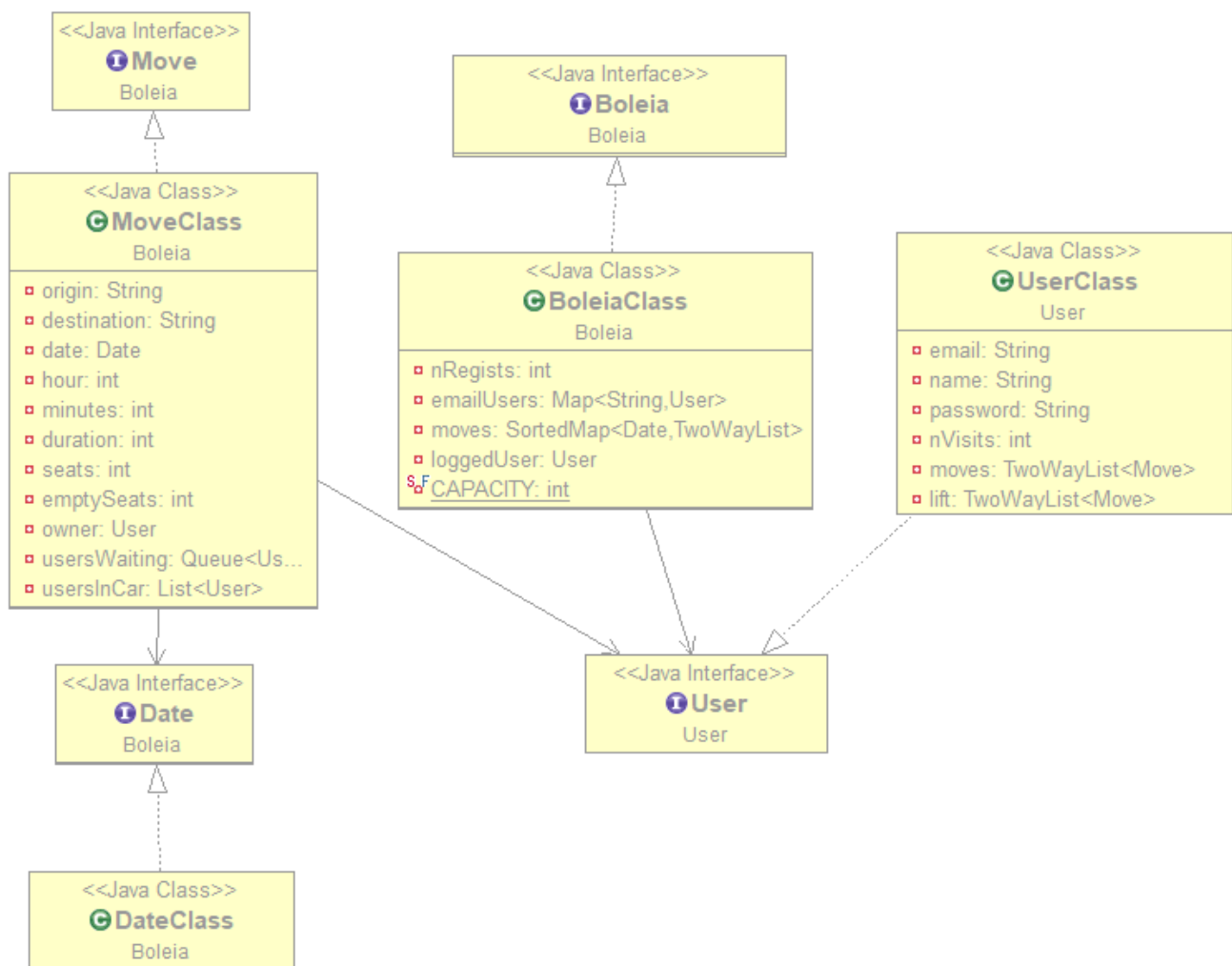


Figura 4 - Diagrama de Classes do Projeto