

— Exame —
Desenvolvimento de Sistemas Software

MiEI 2017/18

24/01/2018

Duração máxima: 1h45

Leia o exame com atenção e responda a cada grupo em folhas separadas!

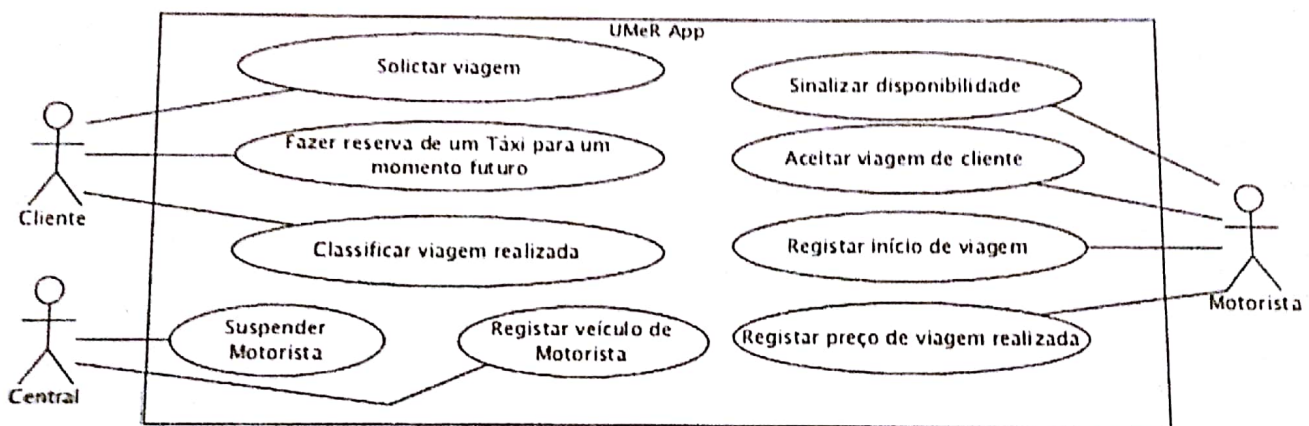
Considere que se pretende criar uma aplicação para um operador de Táxis. Pretende-se que a aplicação a ser desenvolvida dê suporte a toda a funcionalidade que permita que um utilizador realize uma viagem num dos Táxis do operador. Deve ainda abranger todos os mecanismos de criação de utilizadores, motoristas, táxis e posteriormente a marcação das viagens, a realização das mesmas e respectiva imputação do preço. Pretende-se também que o sistema guarde registo de todas as operações efectuadas e que depois tenha mecanismos para as disponibilizar (exemplo: viagens de um utilizador, extracto de viagens de um taxi num determinado período, valor facturado por um taxi num determinado período, etc.).

O processo de fazer uma viagem, utilizando a aplicação a desenvolver, deverá obedecer às seguintes regras:

1. o cliente indica as coordenadas ou a morada do local em que se encontra e do destino pretendido;
2. indica ainda se pretende chamar um Táxi específico ou então solicitar o que esteja mais próximo;
3. no primeiro caso, a aplicação calculada a distância a percorrer entre a posição do cliente e a do Táxi pretendido, calculando ainda quanto tempo este demora a chegar ao cliente e depois ao destino que o cliente solicitou;
4. no segundo caso, o sistema realiza os cálculos para todos os Táxis disponíveis na área geográfica do cliente e selecciona o mais próximo;
5. o serviço é comunicado ao motorista que o poderá aceitar, ou não; em caso negativo, se o cliente pediu o Táxi mais próximo, o próximo Táxi é contactado até não existirem mais Táxis disponíveis, se o cliente pediu um Táxi específico ou se esgotaram os Táxis disponíveis, o cliente é informado da impossibilidade de realizar o serviço;

6. em caso de resposta positiva de um Motorista, o sistema apresenta então ao cliente qual o custo estimado e o tempo total de viagem;
7. o cliente decide se aceita a proposta; em caso positivo, o serviço é comunicado ao motorista, que informa o sistema do início do serviço quando inicia a deslocação para o cliente, do momento em que recolhe o cliente e do momento em que termina o serviço;
8. nesse momento a aplicação calcula o valor final do serviço, que é cobrado directamente ao cliente;
9. após a viagem o cliente pode dar uma nota ao motorista e fica com o documento relativo à viagem guardado na sua área pessoal.

Foi já desenvolvida uma primeira versão do Diagrama de Use Case, apresentada de seguida.



Grupo I

1. Utilizando um diagrama à sua escolha, **descreva o processo de realização de uma viagem**. Indique claramente as responsabilidades dos vários intervenientes. (3 valores)
2. **Indique e justifique a veracidade/falsidade das seguintes afirmações:** (3 valores)
 - (a) A UML é um processo de desenvolvimento orientado a objetos.
 - (b) Um Diagrama de Sequência de Sistema (DSS) é uma versão restrita de um Diagrama de Sequência.
 - (c) Um *Interaction Overview Diagram* combina Diagramas de Actividade com Diagramas de Estado.

Responda a cada grupo em folhas separadas!

Grupo II

3. Sabendo que Clientes e Motoristas só deverão poder utilizar a aplicação depois de terem iniciado sessão e que, para isso ser possível, deverão primeiro registar-se na mesma; sabendo ainda que o acesso da central deverá também ser controlado por um mecanismo de autenticação, existindo uma palavra passe por omissão que deverá depois ser alterada; finalmente, sabendo que deverão ser suportadas duas alternativas para a autenticação (via nome de utilizador e palavra passe e via leitura da impressão digital) e três alternativas para o pagamento (uma por cartão de crédito, outra por mbway e ainda por dinheiro) e que estas deverão estar representadas no diagrama: **complete o Diagrama de Use Case** utilizando herança para simplificar o modelo. (3.5 valores)

4. Escreva a versão tabular do Use Case Solicitar Viagem. (3.5 valores)

————— Responda a cada grupo em folhas separadas! —————

Grupo III

Considere o seguinte excerto de código Java:

```
public class TaxisApp extends JFrame implements Observer {
    public TaxiLN emp;
    ...
}

public class TaxiLN extends Observable {
    private Map<String,Taxi> frota;

    public TaxiLN() {
        this.frota = new TaxiDAO();
    }

    public void removeTaxi(String mat) throws TaxiException {
        if (!this.frota.containsKey(num)) {
            StringBuilder sb = new StringBuilder("Taxi ");
            sb.append(mat);
            sb.append(" não encontrado!");
            throw new TaxiException(sb);
        }
        this.frota.remove(mat);
        this.setChanged();
        this.notifyObservers();
    }

    public void addTaxi(Taxi t) { ... }

    public Taxi getTaxi(String mat) throws TaxiException { ... }
```

```

        private int contaAtivos() {
            Collection<Taxi> taxis = this.frota.values();
            int ativos = 0;

            for (Taxi t: taxis) {
                ativos += t.ativo()? 1:0;
            }
            return ativos;
        }
        ...
    }

    public class TaxiDAO implements Map<String,Taxi> {
        private Connection conn;

        public TaxiDAO () { ... }

        public boolean containsKey(Object key) {
            Statement stm = conn.createStatement();
            String sql = "SELECT * FROM TabTaxis WHERE matricula='"+(String)key+"'";
            ResultSet rs = stm.executeQuery(sql);
            return rs.next();
        }

        public Taxi get(Object key) { ... }

        public Taxi put(String key, Taxi value) { ... }

        public Taxi remove(Object key) {
            Taxi t = this.get(key);
            Statement stm = conn.createStatement();
            String sql = "DELETE '"+key+"' FROM TabTaxis";
            stm.executeUpdate(sql);
            return t;
        }
        ...
    }

    public class Taxi {
        private String matricula;
        private int cilindrada, num_passageiros;
        ...
    }

```

5. Construa um **Diagrama de Classes** para o código apresentado (inclua no modelo o máximo de informação possível). (3.5 valores)
6. Desenhe um **Diagrama de Sequência** para o método void removeTaxi(String mat) da classe TaxiLN. Utilize breaks para representar throws. (3.5 valores)

_____ Respondeu a cada grupo em folhas separadas? _____