Universidade do Minho
Mestrado Integrado em Engenharia Informática

Projeto de Programação Orientada Aos Objetos

GRUPO:
Ana Teresa Gião Gomes - A89536
Maria Quintas Barros - A89325
Maria Beatriz Araújo Lacerda - A89535

11 de Junho 2020

Figure 1: A89536

Figure 2: A89525

Figure 3: A89535

1 Introdução

No âmbito da unidade curricular Programação Orientada aos Objetos, foi-nos proposta a elaboração de um projeto com o objetivo de pôrmos em pratica este paradigma de programação e a linguagem de programação Java. Este trabalho consiste em desenvolvermos uma aplicação que satisfaz as necessidades de pessoas que precisam de serviços e estão confianadas nas suas casas.

2 Classes

2.1 Encomenda

Dentro da classe "Encomenda" encontram-se 7 variáveis de instância:
- String codEncomenda; (Contem o código da encomenda)
- String codUtilizador; (Contem o código do Utilizador)
- String codLoja; (Contem o código da Loja)
- double peso; (Contem o peso total da encomenda, calculado com o método peso, também presente nesta classe)
- boolean entregue; (Boleano que indica o estado da encomenda, se está entregue corresponde a True)
- double tempo;
- boolean medica; (Boleano que indica se a encomenda de que se trata é médica ou não)
- Lista de produtos; (Lista dos produtos da encomenda e a sua informação)

Esta classe tem como função o armazenamento das encomendas que pretendemos colocar no nosso sistema.

2.2 Utilizador

Dentro da classe "Utilizador" encontram-se 4 variáveis de instância:
- String codUti; (Contem o código do utilizador)
- String nome; (Contem o nome do utilizador)
- Coordenadas gps; (Contem as coordenadas do utilizador)
- List de Encomendas pendentes; (Contem uma lista de encomendas pendentes)

- List de Strings transportadoraEntregue; (Registo de encomendas entregues)

Esta classe tem como função de apresentar as informações relativas ao utilizador.

2.3 Transporte

A nossa classe "Transporte" é uma classe abstrata que contem as seguintes variáveis de instância:
-String codEmpr; (Contem o código da transportadora)
-String nomeEmpr;(Contem o nome do transportadora)
-int nif; (Contem o nif do transportadora) -boolean multi;(Boleano que indica o tipo de transportadora, multi se True ou normal se False)
-boolean meds; (Boleano que indica o tipo de transportadora, multi se True ou normal se False)
-Coordenadas gps; (Variável da classe Coordenadas, que contem latitude e longitude)
-double raio; (Contem o raio no qual a transportadora opera)
-double precokm; (Contem o preço por kilómetro)
-double precopeso;(Contem o preço por kg)
-double velocidade; (Contem a velocidade média)
-List de Integer classificacao; (Contem a lista das classificações dadas à transportadora)
-Map (key:Encomenda e value:LocalDateTime) registoT; (Map que contem as encomendas entregues pela transportadora)

Esta classe tem como função de apresentar as informações relativas ao transporte.

Métodos da classe:
Métodos abstratos:
custoEncomenda e tempoDeEncomenda que vão determinar, respetivamente, qual o custo da encomenda e o tempo desta dependendo do tipo de empresa que a transporta.

Subclasses:
Dentro da nossa transportadora temos dois tipos diferentes:

-TransportadoraMulti, que transporta várias encomendas. Tem uma

variável adicional ("numero") que indica quantas encomendas transporta de uma só vez;
-TransportadoraNormal, qua é apenas capaz de transportar uma encomenda;

2.4 Voluntário

Dentro da classe "Voluntario" encontram-se 7 variáveis de instância:
- String codVolu; (Contem o código do voluntário)
- String nome; (Contem o nome do voluntário)
- Coordenadas gps; (Contem as coordenadas do voluntário)
- double raio; (Contem a área em que o voluntário aceita fazer a entrega)
- boolean livre; (Contem um booleano que indica se o voluntário está livre, ou seja, True, para fazer a entrega)
- boolean meds (Indica se o voluntário transporta ou não encomendas médicas)
- Lista de classificações; (Contem uma lista de inteiros com as classificações dadas ao voluntário após a entrega)
- Map¡Encomenda, LocalDateTime¿ registoV; (Contem um map com os registos das encomendas e das datas em que estas foram entregues)

Esta classe tem como função apresentar as informações relativas ao voluntário.

2.5 Loja

Dentro da classe "Loja" encontram-se 9 variáveis de instância:
- String codLoja; (Contem o código da loja)
- String nomeLoja; (Contem o nome da loja)
- double tempoAtend; (Contem o tempo de atendimento da loja)
- Coordenadas gps; (Contem as coordenadas da loja)
- Lista de registos; (Contem uma lista com os registos das encomendas que a loja processou)
- boolean temTempo; (Verifica se a loja dá a informação relativa ao tempo de fila ao utilizador)
- double tempoFila; (Se a variável anterior for verdadeira esta variável fornece o tempo de fila da loja)
- Lista de encomendas pendentes; (Contem a lista de encomendas que a loja ainda não entregou)

- Map (com key:String e value:LinhaEncomenda) catalogo; (Contem a lista de produtos disponíveis na loja para encomenda)

2.6 TrazAquiModel

Nesta classe estao contidos 6 maps, um que contem os utilizadores do nosso sistema, outro para os voluntarios, encomendas, transportadoras, lojas e encomendas aceites. Para isto, implementamos nesta classe metodos para a leitura dos dados dos logs que nos foram dados bem como para a inserção destes dados na map, introduzindo-os no nosso sitema.

3 Funcionalidades do Projeto

3.1 Carregamento de Ficheiros

O nosso programa começa por dar a escolha ao utilizador de carregar os ficheiros "default" isto é, os logs iniciais fornecidos para o projeto ou carregar um ficheiro que foi guardado na ultima sessão.

Em seguida o nosso sistema é iniciado com o ficheiro escolhido.

3.2 Menu Inicial

De seguida é apresentado ao utilizador um Menu Inicial, que lhe dá as opçoes de Iniciar sessão, criar um novo registo, registos de encomendas ou sair do sistema. Ao escolher a opção de saída, todas as alterações feitas ao ficheiro log são guardadas.

3.3 Iniciar Sessão

Se o utilizador escolher a primeira opção, é apresentado um segundo menu que o permite escolher iniciar sessão numa conta de como utilizador, voluntário, transportadora,loja ou sair.

Utilizador
Após a introdução de um código válido é iniciada a sessão, feito pelo método "aplicação" na classe "TrazAquiController". Começa por ser

imprimido o número de encomendas entregues e o número de encomendas pendentes. Em seguida é apresentado o menu com as opções: gerar encomenda, finalizar encomendas pendentes, lista de produtos, área de cliente ou sair.
Ao gerar uma encomenda, o utilizador começa por escolher se pretende fazer uma encomenda médico ou normal. Em seguida é adicionado o código do utilizador à encomenda e é gerado um código para esta. Depois é apresentado o catálogo da loja escolhida e o utilizador escolhe os produtos que pretende encomendar. A encomenda é adicionada à lista das encomendas pendentes da loja.
Caso o utilizador escolha finalizar encomendas pendentes é imprimida uma lista de encomendas que ainda não terminaram o processo. Em seguida é perguntado ao utilizador se este pretende levantar a encomenda ou não. Se a opção for não é verificado se existe algum voluntário livre para entregar a encomenda. Se não existir é apresentada uma lista de transportadoras disponíveis para entregar aquela encomenda ao cliente. Se a transportadora escolhida for uma transportadora multi (isto é, transporta várias encomendas) a encomenda é adicionada à sua lista de pendentes e será entrega quando tiver várias encomendas prontoas a isso (cada transportadora tem o seu limite).
Se for escolhida a 3ª opção será disponibilizado ao utilizador o catálogo de produtos da loja que escolher.
Se a opção escolhida for área de cliente, o utilizador porderá ver o seu perfil atual, editar este ou visualizar as encomendas registadas feitas pelas transportadoras ou voluntários.

Voluntário
Se a opção escolhida for Voluntário, aparece mais um menu, que oferece as opções de consultar a área de cliente, a sua classificação, o registo de encomendas ou sair.
A área de cliente permite nao só consultar o atual perfil do voluntário, como ainda alterar esse mesmo perfil.
Também poderá visualizar a sua atual classificação, que é calculada através do método "mediaClassificação" da classe "Voluntário".
Se escolher a 3ª opção, é apresentado a lista de encomendas que já doram entregues pelo voluntário em questão.

Transportadora
Após a inserção de um código válido, o sistema deteta de é uma transportadora multi(capz de transportadar várias encomendas) ou uma trans-

portadora normal (que transporta apenas uma única encomenda).
Se a transportadora for multi, terá acesso a um menu que terá como opções: area de cliente, classificação, registo de encomendas, faturação total, encomendas por entregar ou sair. Tal como anteriormente, a área de cliente permite visualizar e alterar as definições do perfil. Também a classificação é calculada de maneira semelhante, mas num método dentro da classe abstrata das transportadoras. O registo apresenta uma lista das encomendas que já foram entregues pela transportadora em questão. A 4ª opção apresenta a faturação. Isto é alcançável atravês do método "faturacaoDadoTempo", que cria lista das encomendas entregues no intervalo de tempo pretendido e a percorre, somando na variável fat o custo de cada encomenda, custo este que é calculado de diferente maneira, dependendo do tipo de transportadora.
Por último, é possível para uma transportadora ter acesso à lista de encomendas por entregar. Esta informação é dada pelo método "encomendasPorEntregar" da classe "View", através da lista de encomendas Pendentes.
Se se tratar de uma transportadora normal as opções são: area de cliente, classificação, registo de encomendas, faturação total e sair e são calculadas de forma semelhante à transportadora multi.

Loja
Se a opção escolhida for Loja, vai ser pedido que se insira o código da loja para iniciar sessão. Depois, programa alerta esta para as encomendas que tem pendentes e aparece mais um menu que oferece a opção de inserir produto, ver o catálogo, finalizar uma encomenda, ver o registo de encomendas ou sair.
Na opção de Inserir produto, pede-se que insira a referência, a descricao e o peso e o preço por unidade do produto que a loja quer inserir para o seu catálogo.
Em Catálogo de produtos, a loja pode visulaizar aquele que é o seu catálogo até ao momento.
Em finalizar encomenda, são mostradas todas as encomendas que estão pendentes e caso tenha alguma nesse estado, pede-se que insira o código da encomenda que pretende finalizar.
Em Registos de encomendas, é mostrado o registo de todas as encomendas que foram feitas até ao momento.
Por fim, temos a opção de sair deste menu.

3.4 Criar Registo

Se o utilizador escolher a segunda opção, é apresentado um segundo menu que o permite escolher criar registo de uma conta nova de utilizador, voluntário, transportadora,loja ou sair.

Utilizador
Se a opção escolhida for criar um novo utilizador, é pedido para inserir um código, nome, apelido e coordenadas x e y.

Voluntário
Se a opção escolhida for criar um novo voluntário, é pedido para inserir um código, nome, apelido, coordenadas x e y, raio de entrega e,por fim, pergunta se este novo voluntário tem certificado de transporte de medicamentos.

Transportadora
Se a opção escolhida for criar uma nova transportadora, é pedido para inserir um código, nome, NIF, coordenadas x e y, raio de entrega, preço por km e por peso, velocidade média a que a transportadora se desloca para uma entrega e questiona se esta faz multiplas encomendas e se possui certificado de transporte de medicamentos.

Loja
Se a opção escolhida for criar uma nova loja, é pedido para inserir um código, nome, tempo médio de atendimento, coordenadas x e y e pergunta se esta nova loja tem informação de fila. Caso a resposta a esta ultima pergunta seja sim, pede o tempo medio da fila.

4 Diagrama de Classes

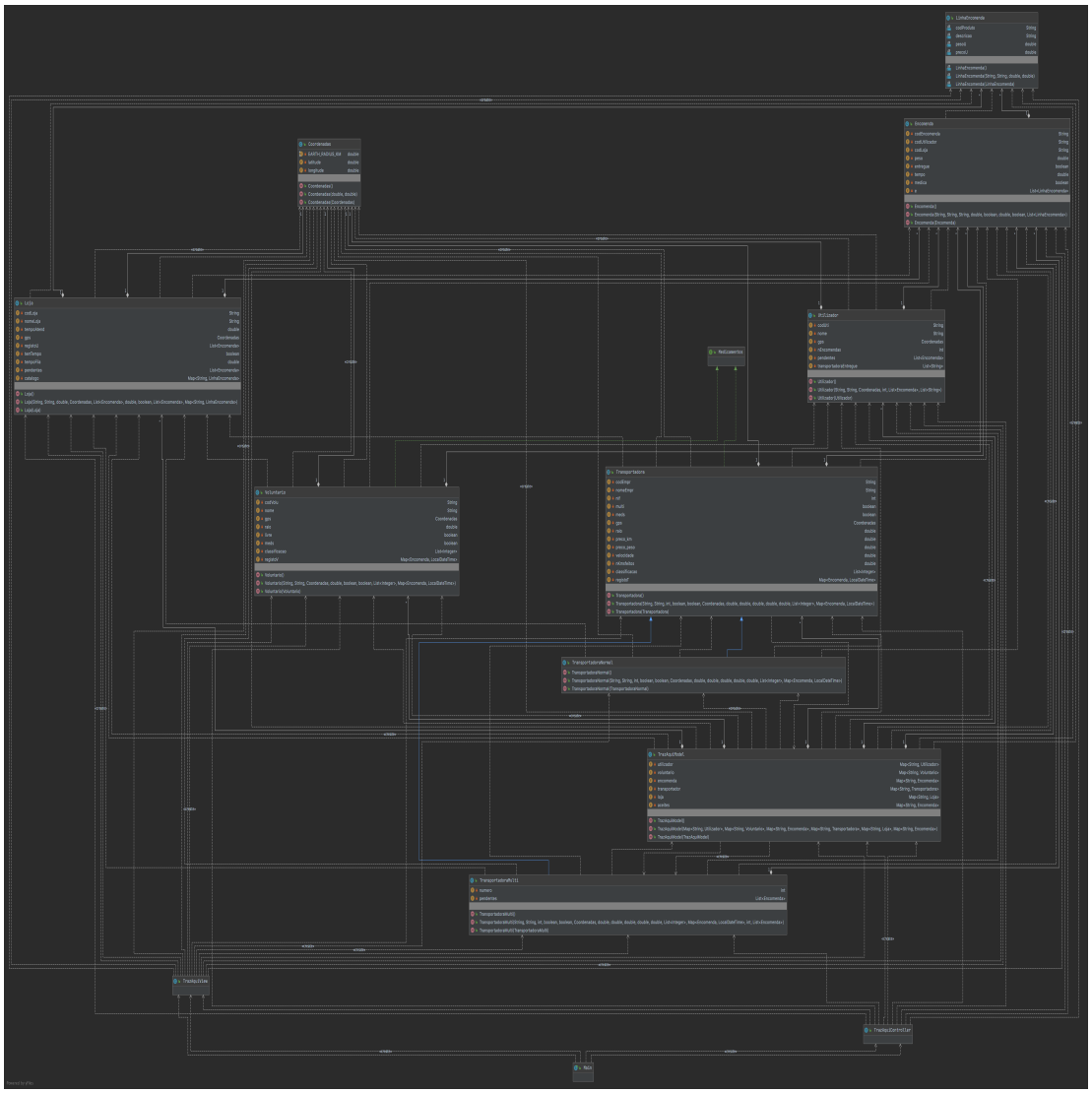

Figure 4: Diagrama de Classes

5 Conclusão

Após a realização deste trabalho, concluimos que a implementação deste tipo de plataformas através da utilização da linguagem Java é bastante mais acessível principalmente em termos de abstração, modularidade e reutilização de código. Por outro lado, também conseguimos entender melhor conceitos como encapsulamento de dados, otimização de código e utilização de classes abstratas e interfaces. Deste modo, concluindo que este projeto foi bastante bem conseguido e aumentou imenso o nosso conhecimento nesta nova linguagem.