Universidade do Minho

Relatório

TrazAqui

Programação Orientada a Objetos

Mestrado Integrado em Engenharia Informática

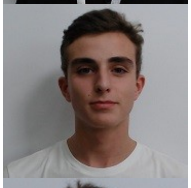
2º ano – 2º Semestre

2019/2020

Projeto realizado pelo grupo 28:

Rúben Correia Cerqueira A89593

Júlio Miguel De Sá Lima Magalhães Alves A89468



Joel Salgueiro Martins A89575

Conteúdo

1. Introdução

No âmbito da Unidade Curricular de Programação Orientada a Objetos, foi-nos proposto a realização de um projeto que consiste na simulação de uma aplicação de entrega de encomendas ao domicílio.

Nesta aplicação, é possível várias entidades fazer login na aplicação para gerar o fluxo desta, ou seja, um utilizador faz login e pede uma encomenda, após esta ação um voluntário ou empresa transportadora que esteja no raio de ação da loja em que a encomenda pedida se encontra pode-se disponibilizar a entregar, sendo no fim, classificada pelo respetivo utilizador.

A realização deste projeto foi um bom meio de estudo e consolidação dos conhecimentos adquiridos nesta UC.

2. Descrição do Projeto

2.1 Model

2.1.1 Utilizador

O Utilizador é a classe que representa uma pessoa normal a usar a aplicação, que pode realizar encomendas, aceitá-las, ver estatísticas, entre outros.

Esta classe contém informações acerca da localização, nome, código e histórico de encomendas.

A localização é necessária para calcular distâncias para os preços das transportadoras, o código serve para identificar o utilizador no sistema, o histórico de encomendas é para ter acesso às encomendas que foram anteriormente enviadas e o nome para a apresentação de resultados.

2.1.2 Loja

A Loja é uma representação de um setor que recebe pedidos de encomendas de um utilizador e sinaliza os estafetas que há uma encomenda para ser entregue.

Tal como o utilizador, a loja tem as informações de código, nome, localização e pedidos de encomenda.

Código será para identificar a loja, o nome será usado para apresentação de resultados, localização para verificar se esta se encontra no raio de ação dos estafetas e para calcular distâncias por parte das transportadoras e os pedidos de encomenda para sinalizar quais as encomendas estão prontas para serem entregues.

Também adicionamos outro tipo de loja, chamado LojaFilaEspera, que consiste na subclasse da Loja, ou seja, contém todos os campos mencionados, para os mesmos objetivos, mas têm informação acerca da fila de espera da loja e o tempo estimado de atendimento a cada entidade.

2.1.3 Estafeta

Decidimos usar esta classe como classe abstrata que aglomera os voluntários e as transportadoras uma vez que estas têm o mesmo propósito que é fazer a entrega de encomendas.

Esta classe irá ter informações como código, nome, localização, raio de ação, encomendas entregues, pedidosEncomenda, classificação, disponibilidade e certificação.

O código será para identificar a entidade no sistema, o nome para apresentar resultados, localização para determinar distâncias, raio para identificar as lojas em que pode atuar, encomendas entregues para ter acesso a estatísticas acerca do Estafeta, pedidos de encomenda para encomendas em standby, classificação é a média das classificações dadas pelos utilizadores, disponibilidade para indicar se a entidade está disposta a entregar encomendas e certificação se o estafeta tem a capacidade de fazer o transporte de encomendas médicas.

2.1.3.1 Voluntário

O voluntário é uma subclasse de Estafeta, porém não possui nenhuma informação adicional, sendo a sua única diferença relativamente à Transportadora que este não acarreta nenhum custo ao utilizador pela entrega da encomenda.

2.1.3.2 Transportadora

A classe Transportadora é uma subclasse de Estafeta e, como dito no ponto 2.3.1, a sua única diferença relativamente ao Voluntário é que acarreta um custo pela entrega ao utilizador e, por isso, acrescenta essa informação relativamente ao Estafeta e também acrescenta a informação relativa ao seu NIF e ao número de quilómetros percorridos.

O preço por quilómetro corresponde ao preço por cada quilómetro percorrido para fazer a entrega, o NIF é para saber o NIF da transportadora e o número de quilómetros percorridos serve para fins estatísticos como por exemplo para se fazer o top 10 transportadoras.

2.1.4 Encomenda

Esta classe representa a informação contida por cada encomenda presente no sistema.

A Encomenda é também constituída pelo seu código, o seu peso, a data de quando foi pedida, o utilizador que a pediu, o estafeta que a transportou, a loja de onde foi pedida , um indicador se a encomenda é médica ou não e uma lista de linhas de encomenda.

Cada encomenda é constituída por uma ou mais LinhasEncomenda que representam os dados (preço, quantidade, a descrição do produto, a fragilidade e o código) de cada produto pedido na Encomenda.

2.1.5 Estado

O estado é a classe principal do Model pois é onde é tratada toda a informação do sistema.

Aqui temos armazenados os dados sobre todos os utilizadores, estafetas e lojas do sistema através de HashMaps e também é aqui que temos informação sobre o login.

Esta vai ser a classe com que o Controller vai comunicar para obter os dados necessários para a View apresentar.

2.1.6 FileIO

Esta classe foi desenhada de forma a tratar das operações Input/Output de ficheiros, tais como, ler dos ficheiros de logs, escrever e ler ficheiros binários e consultar e registar contas do sistema.

2.2 Controller

É na classe Menu que está situado o Controller. Aqui são processadas as informações Input do utilizador e para onde o Model envia as respostas, sendo depois enviadas para a View.

O Menu é responsável por tratar do login das diferentes entidades que vão usar o sistema e este tem funcionalidades diferentes para cada um.

Para o Utilizador, o Menu dispõe de funcionalidades como efetuar uma encomenda, ver o histórico de encomendas (que pode ser filtrado por voluntários, transportadora e por intervalo de tempo) , aceitar uma entrega de uma encomenda proposta por uma Transportadora e avaliar o estafeta responsável pela entrega.

Para a loja, possui funcionalidades como ver as encomendas que estão à espera de ser levantadas por um estafeta e consultar o tamanho da fila de espera.

Para o Voluntário, possui funcionalidades como mudar de disponibilidade, escolher uma encomenda para ir levantar, alterar o raio de ação e consultar a sua classificação.

Para a Transportadora, possui funcionalidades como mudar a disponibilidade, determinar o preço de uma encomenda, transportar uma encomenda, consultar o total faturado num determinado intervalo de tempo, alterar o raio de ação, alterar o preço por quilómetro, ver a sua classificação.

Todas estas entidades são capazes de consultar o top 10 utilizadores, o top 10 transportadoras, terminar sessão e encerrar o programa.

2.3 View

A View está definida na classe UI, e é aqui que se apresenta o output pedido pelo utilizador do programa.

2.4 Entrada

Esta interface aglomera as várias entidades do sistema e é uma das responsáveis pela abstração e flexibilidade do programa. É aqui que se processam as informações do login, uma vez que pode ser qualquer uma das entidades a fazer login.

2.5 Exceptions

Para este programa apenas criamos 3 exceções, ExistingCodeException, InvalidInputException e LojaInexistenteException, pois todos os erros com os quais nos fomos defrontando se resumiam a estes 3 tipos de exceção.

3. Conclusão

Em suma, esta aplicação cumpre todos os requisitos propostos, seguindo o padrão MVC (Model-View-Controller).

Como em qualquer outra aplicação, alguns aspetos poderiam ser tidos em consideração, nomeadamente implementar a noção de tempo real no programa, a noção de rota e adicionar novas entidades.

4. Diagrama de classes