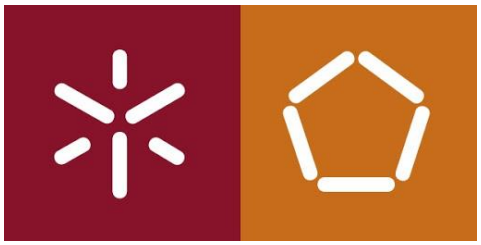


Universidade do Minho

Mestrado Integrado em Engenharia Informática

Programação Orientada aos Objetos

TrazAqui

Artur Drohobytskyy | a75874 | Grupo 61

Junho, 2020

Índice

- Introdução.....pág. 3
- Arquitetura.....pág. 4 – pág. 5
- Instruções.....pág. 6 – pág. 7
- Conclusão.....pág. 8

Introdução

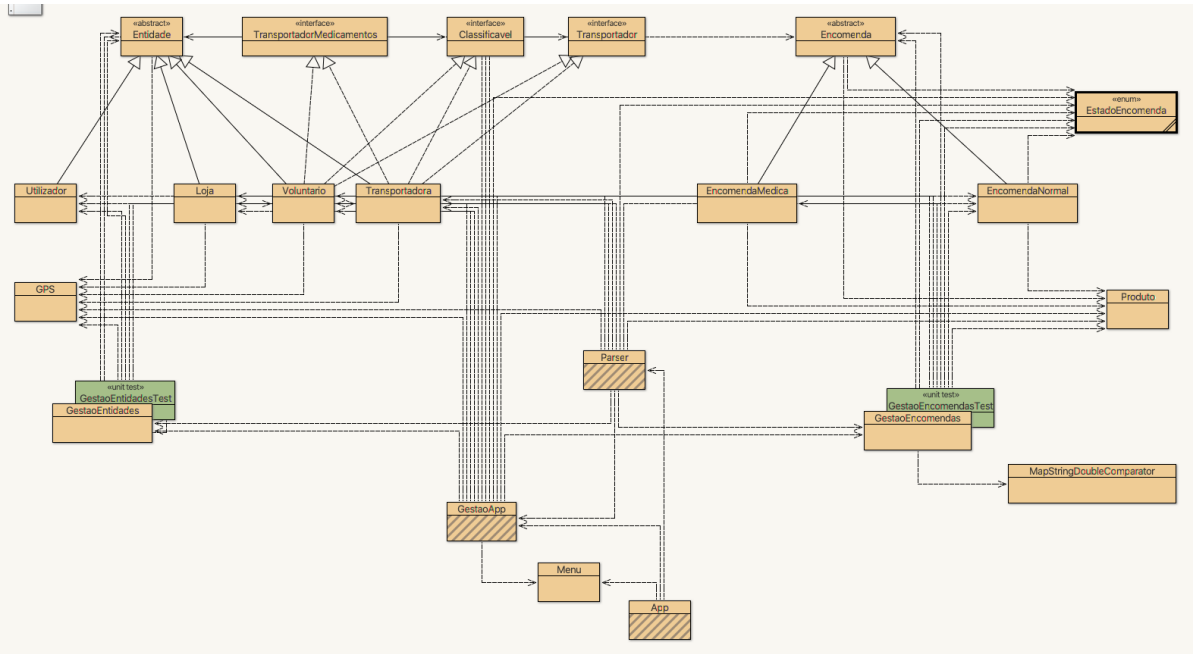
No âmbito da unidade curricular **Programação Orientada aos Objetos**, pretende-se criar uma aplicação de gestão de encomendas, utilizando a linguagem de programação **Java** e **IDE BlueJ.**

O objetivo principal é entender a importância e as grandes vantagens na utilização deste paradigma (**POO**).

O trabalho proposto consiste na implementação de uma solução que satisfaz os requisitos do enunciado e a utilização das técnicas e conhecimento adquiridos nas aulas teóricas desta disciplina.

Arquitetura

Diagrama de classes foi elaborado no **IDE BlueJ** e está apresentado na imagem abaixo:



1. As decisões tomadas em relação a estrutura do projeto
    - Criação das classes abstratas principais tais como **Entidade** e **Encomenda** e as respetivas variáveis de instância que foram pensadas de acordo com o enunciado
    - Em seguida foram criadas as classes concretas que estendem as classes base como **Utilizador**, **Loja**, **Voluntario**, **Transportadora**, **EncomendaMedica**, e **EncomendaNormal**
    - **GPS** e **Produto** para serem utilizados como variáveis de instância nas classes referidas acima
    - **Parser** para carregar os dados para testes a partir de um ficheiro *CSV*
    - **GestaoEntidades** e **GestaoEncomendas** que implementam a lógica relativa ao processo de gestão de *entidades* e as *encomendas*
    - **GestApp** tem como variáveis de instância **GestaoEntidades** e **GestaoEncomendas** e serve de camada de acesso a toda gestão da aplicação
    - **Menu** é usado como interface em modo texto para utilizador
    - **App tem GestApp** como variável de instância que só é instanciado uma vez durante a execução do programa, mantendo o estado em memoria
    - Durante o desenvolvimento do projeto foi necessário criar e implementar ***Interfaces*** e ***Enums*** para ajudar a resolver problemas que seria muito mais complicado de solucionar sem ter essas opções disponíveis

2. Uso de classes Abstratas e Interfaces levou a reutilização de código e uso das potencialidades de POO como Herança, Encapsulamento e Polimorfismo
3. Uso de **Enums** melhorou bastante o controlo de estado de encomenda
4. Uso de **Streams** e funções de ordem superior para simplificar a sintaxe e tornar o programa mais flexível e extensível para possíveis alterações
5. Foram criados testes unitários para as classes que implementam maior a logica de negócio
6. Melhorias a fazer:
    - Pensar numa forma diferente de guardar os dados para poder ter uma classe que faça esse trabalho para separar as responsabilidades e simplificar os testes
    - Uso de predicados para reduzir o número de funções e torná-las mais genéricas
    - Criar mais testes unitários
    - Seria interessante implementar padrões de desenho como Facade, Startegy etc.
    - Seguir os princípios S.O.L.I.D
    - Implementar interface gráfica com padrão de desenho MVC MVVM MVP etc.

Instruções

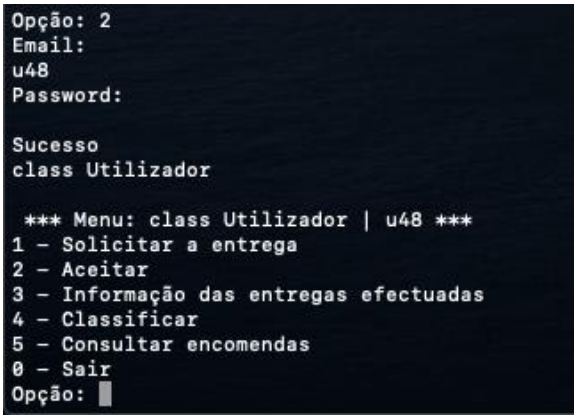
Neste momento a interface disponível é em modo texto

É possível efetuar as seguintes operações:

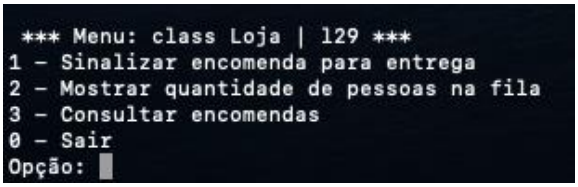
1. **Parse:** é feito o carregamento de dados para testar aplicação
2. **Login:** autenticação email e password (para simplificar os testes, email é o *código da entidade* password é *String vazia* para os dados carregados do ficheiro)
    - *Utilizador*
        1. **Solicitar a entrega**
        2. **Aceitar**
        3. **Informação das entregas efetuadas**
        4. **Classificar** (transportadores)
        5. **Consultar encomendas**
    - *Loja*
        1. **Sinalizar encomenda para entrega**
        2. **Mostrar quantidade de pessoas na fila**
        3. **Consultar encomendas**
    - *Voluntario*
        1. **Sinalizar disponibilidade**
        2. **Levantar encomenda do utilizador na loja**
        3. **Transportar encomenda ao destino**
        4. **Consultar encomendas**
    - *Transportadora*
        1. **Sinalizar disponibilidade**
        2. **Determinar o preço do transporte**
        3. **Transportar encomenda**
        4. **Consultar encomendas**
        5. **Consultar total faturado**
3. **Registo:** permite criação de uma nova entidade (*Utilizador, Loja, Voluntario* ou *Transportadora*)
4. **Top 10 utilizadores:** lista 10 utilizadores que mais usam o sistema
5. **Top 10 transportadores:** lista 10 transportadores com mais utilizam o sistema

1. Menu principal

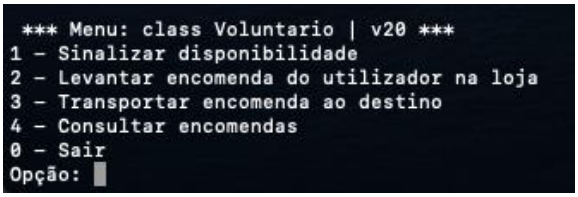
2. Menu utilizador



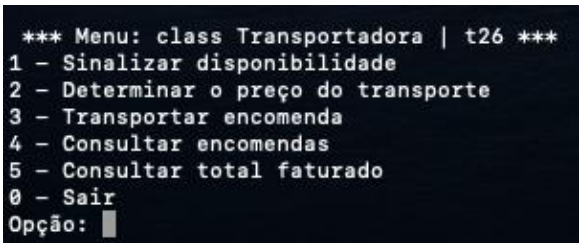
3. Menu loja



4. Menu voluntario



5. Menu Transportadora



Conclusão

Posso concluir que foi uma excelente experiência em participar no desenvolvimento deste trabalho prático assim como a aprendizagem do conteúdo das aulas teóricas.

Acrescenta um valor de extrema importância para o percurso académico e profissional, pois aprendi bastante sobre uma vasta variedade de temas no paradigma de Programação Orientada aos Objetos.

Entendi a importância de uso de *Classes Abstratas* e *Interfaces* que proporcionam *Encapsulamento, Herança e Polimorfismo,* por sua vez possibilitam a reutilização do código, testes, legibilidade etc..

API 8 do Java traz uma novidade que são os S*treams* e que o seu uso permite uma elegância na escrita do código e para além disso a possibilidade de integrar a programação paralela com esforço muito reduzido, permitindo assim a otimização na execução de uma aplicação.

Pratiquei o desenho diagrama de classes que foi muito útil para a aprendizagem de estruturação de um projeto.

Aprendi a importância em escrever os testes unitários e como se pode prevenir os bugs de lógica logo a partida e prevenir as futuras falhas no caso de haver alteração no código

Penso que o trabalho correu de forma muito positiva, foram realizados todos os requisitos básicos da especificação com algumas dificuldades a meio do caminho, mas que no final foram ultrapassadas que ajudou a aprender muitos conceitos novos que irão facilitar bastante no desempenho de funções relacionadas com este tema.