Universidade do Minho

Escola de Engenharia

Licenciatura em Engenharia Informática

Unidade Curricular de

Programação Orientada a Objetos

Ano Letivo de 2022/2023

Relatório

Vintage: Sistema de Gestão de Vendas e Encomendas

Carlos Eduardo Culolo Dantas da Costa

A88551

Guilherme Gil Rocha Gonçalves

A88280

Miguel Nogueira Rodrigues

A89499

Grupo nº74

10 de Maio de 2023

ÍNDICE

OBJETIVOS

O objetivo deste projeto é criar uma aplicação que permita a gestão de todo um sistema de pedidos e entregas de encomendas, como adicionalmente a gestão de utilizadores que efetuem compras ou vendas, cuja estrutura deve permitir uma eventual expansão controlada, tendo sempre em mente as normas de programação orientada a objetos.

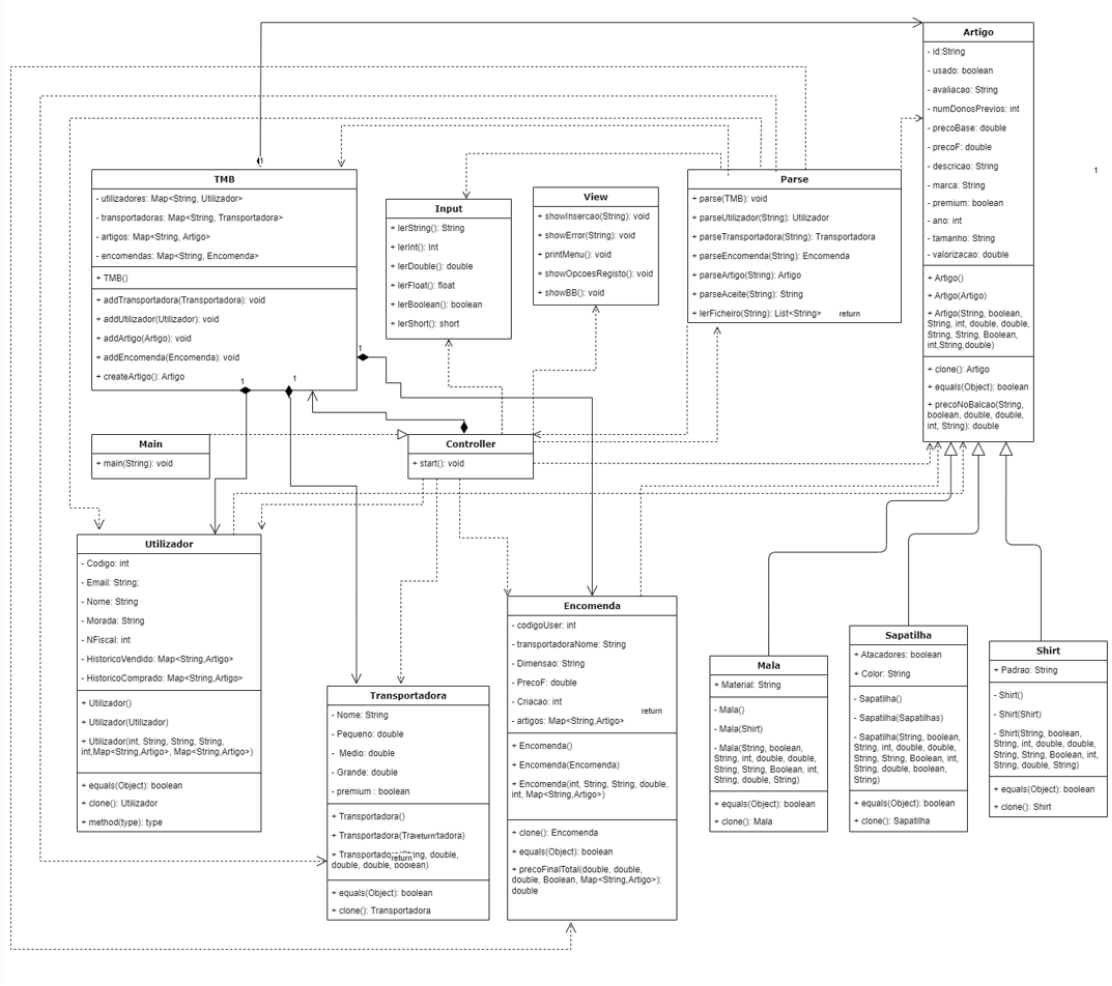
O programa deve permitir não só a gestão do sistema, como também deve permitir a introdução por parte do administrador e usuários de novos dados (por exemplo, quando uma nova Transportadora é introduzida ao sistema, ou adição de um artigo, ou o registo de um novo utilizador, ou a realização de uma nova encomenda), devendo ainda permitir guardar e carregar o estado da aplicação a qualquer momento.

APLICAÇÃO

Classes

1. Main: Cria um novo *Controller*, e inicializa o programa.
2. Artigo: superclasse que contém a informação comum ás subclasses e o método de calcular o preço do artigo no momento de venda na loja.
    a. Mala: Contém a informação específica relativamente à mala.
    b. Shirt: Contém a informação específica relativamente à t-shirt.
    c. Sapatilha: Contém a informação específica relativamente à sapatilha.
3. Encomenda: Contém toda a informação sobre a encomenda e o método de calcular o preço total a pagar do artigo.
4. Utilizador: Contém toda a informação sobre o utilizador.
5. Transportadora: Contém toda a informação sobre a transportadora.
6. TMB: Contém a informação sobre o sistema e sobre todas as entidades que dele fazem parte (*Maps* de utilizadores, artigos, transportadora e encomendas).
7. Input: A classe Input lê e verifica a validez de vários tipos de valores (String, int, boolean, etc) com a ajuda da classe *Scanner* do package *java.util*.
8. View: Esta interface é responsável por dar os *prints* necessários à apresentação de informação ao utilizador do programa.
9. Parse: É a interface responsável por ler os dados do ficheiro.

Diagrama de Classes



Para consultar o diagrama na sua totalidade, consultar o ficheiro *diagrama* no .zip do projeto.

FUNCIONALIDADES

Este programa é capaz de realizar as seguintes ações:

1. Guardar e carregar um estado do sistema, onde o estado é gravado no ficheiro *DB.bin*
2. Possui a capacidade de criar utilizadores (vendedores e/ou compradores), artigos, transportadores e fazer encomendas.
3. As prévias funcionalidades, como todos os métodos, já implementam o conceito de *Premium* ao usar artigos

SOLUÇÕES ADOTADAS

De um modo geral, as classes principais que contém as entidades do sistemas foram definidas tendo a eficiência e também a fácil interpretação do código .

Para a existência de maior simplicidade, todas os atributos que utilizassem a data apenas registam e utilizam o ano. Deste modo, os métodos de calculo de preços de artigos e encomendas tornam-se mais eficientes.

Assim, a estrutura utilizada para grupos de dados (nomeadamente listas de artigos) foram os *Maps*. Esta estrutura é mais eficiente do que *Lists* e permite aceder a dados específicos sem a necessidade de percorrer a estrutura inteira.

CONCLUSÃO

Um dos principais desafios encontrados durante a realização deste projeto foi a criação de uma interface *user*-programa que fosse simples de utilizar, e também que implementa todas as especificações base pedidas.

Em suma, apesar dos obstáculos encontrados durante a realização do projeto, acha-se que este está a um nível de qualidade satisfatório para um base do projeto geral. Apesar disto, sabe-se que há sempre espaço para melhorias e adições de novas funcionalidades.