

**Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores**

**Partilha de Trotinetes Elétricas**

|  |  |  |
| --- | --- | --- |
| Autores: | 50485 | Gustavo Bueno Antunes |
|  | 49504  51283 | Jorge Palácios da Silva  João Nunes Chuço |
|  |

Relatório para a Unidade Curricular de Programação da Licenciatura em Engenharia Informática e de Computadores

Professor: Nuno Datia

18 – 12 – 2025

<< Esta página foi intencionalmente deixada em branco >>

Resumo

No domínio da Unidade Curricular de Introdução a Sistemas de Informação, fomos encarregues de estabelecer uma ligação ao Sistema de Gerenciamento de Base de Dados, SGBD, uma interface entre vários usuários e a base de dados, utilizando Java Database Connectivity, JDBC, uma API baseada em X/Open SQL CLI que permite o acesso, em java, a base de dados relacionais e a execução em comandos SQL.

Para tal, foi utilizado o texto com os requisitos do sistema, apresentados na primeira fase do projeto, e o modelo de dados que foi implementado na segunda fase.

Abstract

Within the scope of the curricular unit Introduction of Information Systems, we carried out establishing a connection to the Data Base Management System, DBMS, an interface between users and the database, using Java Database Connectivity, an API based on X/Open SQL CLI that allows access, in java, the relational database and the execution in SQL commands.

To carry out this step, the text with the system requirements, presented in the first phase of the project, and the data model in the second phase were used.

Índice

[1. Introdução 1](#_Toc185428342)

[2. Restrições 1](#_Toc185428343)

[3. Model 2](#_Toc185428344)

[4. App 2](#_Toc185428345)

[5. Conclusão 5](#_Toc185428346)

# Introdução

Pretende-se criar um sistema informático para gestão e partilha de trotinetes elétricas. Para tal é necessário desenvolver uma aplicação Java que permita realizar as seguintes operações: Opção para introduzir novos clientes; Opção para listar as ordens de reposição nuam dada estação; Opção para iniciar/parar uma viagem; Opção para atualizar o estado das docas; Opção para analisar o grau de satisfação do utilizador; Opção para listar as três estações com maior taxa de ocupação. O modelo ER utilizado foi fornecido no enunciado do segundo trabalho.

# Restrições

Para as restrições de integridade apenas implementamos quatro que foram dadas inicialmente pelo enunciado. A primeira restrição é para verificar que não existe nenhum viagem a decorrer num usuário. A segunda restrição é para verificar que a scooter está disponível na determinada estação. A terceira é para verificar se o valor que está na *String* é um inteiro válido. E a quarta é para verficar se a viagem possui uma avaliação. Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamenteUma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra, software

Descrição gerada automaticamente

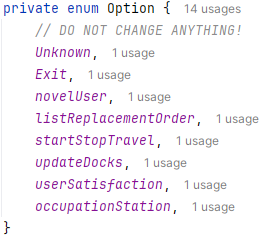
Listagem 1 Implementação das Restrições

# Model

O ficheiro *Model* é onde se encontram as funções todas que são usadas na aplicação que se deseja desenvolver.

A função *listOrders* cria uma lista de pedidos baseado num critério que dado como parâmetro. A função *listReplacementOrders* faz uma lista de pedidos de reposição para uma estação específica num dado período de tempo. A função *travel* processa a operação de viagem de um usuário, usa como função auxiliar a função *getClientId* que obtém o *ID* do cliente. Essa função irá usufruir das duas funções *startTravel* e *stopTravel*, que servirão para começar e parar uma viagem, respetivamente. A função *occupationStation* lista as três estações com maior taxa de ocupação.

# App

 O ficheiro App.java constroi a app chamando as funções presentes no Model.java. Dentro da class App, tem uma enum class Option que enumera as opções que o utilizador têm ao iniciar a app e tem um função App que escolhendo uma opção dentro da enum class, é executada a função destinada a esse propósito. Tem também uma função DisplayMenu utilizada para apresentar as opções ao utilizador. Listagem 2 Função UI

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamenteListagem 3 Função UI

Listagem 4 Função utilizada para a apresentação das opções ao utilizador

Para o uso da aplicação é necessário usar alguns comandos por cada opção:

Opção 2- Inserir informação de utilizador, ou seja, email, nº contribuinte e nome.

Opção 3- Inserir id da estação, e o intervalo de tempo.

Opção 4- Na opção *start* é preciso inserir o id do cliente, id da scooter e id da estação.

Opção 5- É necessário inserir o estado da scooter junto com o seu ID no número da doca correto e ID da determinada estação.

Opção 6- Inserir o id do utilizador.

# Conclusão