

Empresa de Transporte de Mercadorias (Parte 1)

Transportes SML

up201806429@fe.up.pt

Diogo Miguel Ferreira Rodrigues

up201806554@fe.up.pt

Telmo Alexandre Espirito Santo Baptista

up201306340@fe.up.pt

Luís Paulo da Rocha Miranda

Faculdade de Engenharia da Universidade do Porto
Algoritmos e Estruturas de Dados (AEDA) - Turma 5, grupo 3

16 de novembro de 2019

Problema: Empresa de Transporte de Mercadorias

Modelar um problema recorrendo ao paradigma da orientação por objetos e usar a linguagem C++ para implementar a solução correspondente.

A empresa Transportes SML é especialista no transporte de mercadorias. A empresa possui um número fixo de camiões de diferentes tipos, específicos para o transporte de determinada mercadoria [...]

Os clientes [...] podem requisitar serviços de transporte de mercadorias, [...] (o que) pode obrigar ao uso de múltiplos camiões por parte da empresa.

Interessa conhecer os valores mensais que a empresa retira dos seus serviços de transporte, para cada um dos tipos de camião e no geral. Deve ser possível monitorar os clientes da empresa e serviços efetuados.

Os utilizadores da aplicação devem fazer login, e possuem permissões diferentes consoante o seu tipo:

- Funcionário (Employee)
 - Gestor (Manager)
 - Condutor (Driver)
- Cliente (Client)

Um só tipo de camiões (Truck), que pode carregar uma carga (Cargo) de um dos tipos, cujo preço de transporte varia:

- Normal (Normal)
- Animal (Animal)
- Refrigerada (Refrigerated)
- Perigoso (Dangerous)

Todos os utilizadores podem ver e editar as suas informações pessoais. O cliente pode ver e editar os seus serviços ativos, ou requisitar um novo. O condutor pode ver e editar os seus serviços ativos, bem como ver a lista de camiões disponíveis e as suas estatísticas mensais.. O gestor pode fazer tudo isto e ainda ver as listas de clientes e funcionários (completas ou

Quando um cliente requisita um serviço, são atribuídos automaticamente condutores e camiões. Algoritmo de ordenação mergesort, pesquisa linear

***** Falar sobre a organização das classes etc?? por exemplo: Foram implementadas as classes “Person” (da qual derivam “Client” e “Employee”, da qual derivam “Driver” e “Manager”), “Trucks”, “Cargo” e “Service”. **** - algoritmos relevantes - estava a pensar falarmos aqui da atribuição dos serviços

- diagrama de classes - diagrama de classes

clients.txt

Guarda a lista de todos os usuários no seguinte formato: Nome; Número de telemóvel; Nome de utilizador; Password; Morada (Rua e número; Código-Postal; Distrito e País); NIF;

drivers.txt/managers.txt

Guarda a lista de todos os condutores no mesmo formato dos clientes, com uma linha extra equivalente ao salário. trucks.txt

Guarda a lista de todos os camiões da empresa no seguinte formato: Matrícula; Data da matrícula; Combustível (de 0 a 6, de acordo com o tipo); Alcance (em km); Categoria; Número de cargas; Lista de cargas (Peso; Descrição; Preço por km; custo de refrigeração ou nível de perigo, caso se aplique); services.txt - falta completar

Aqui pra já so fiz a lista das exceções que vi serem tratadas, falta atualizar quando o trabalho estiver pronto e falar um pouco sobre elas provavelmente (nao tenho a certeza do q a prof quer aqui)

Lista de funcionalidades

Clients can request a service, with origin, destination, travel time,...

Show list of clients; add edit or remove clients;

Show list of trucks; add edit or remove trucks;

Login for clients, managers and drivers

Add, remove and edit truck

Add, remove and edit client, manager and driver

When a driver arrives to a place, he needs to have someone's contact for loading/unloading, which might or might not be the client for whom the service is being made

(tenho atualizado conforme tem sido feito, é capaz de ja ter mais alguma coisa desde ontem a noite, mas tambem mais vale so atualizar no fim)

AS VÁRIAS LISTAS E OPÇÕES DAS LISTAS (ORDENAÇÃO, FILTRAGEM,...)

Destaque de funcionalidade

Trabalho de grupo = colaboração + trabalho individual

Neste trabalho as classes estavam muito interligadas umas com as outras e com a estrutura da aplicação, pelo que foi difícil compartimentar tarefas e classes/funções

Diogo Rodrigues

- utils: quase tudo, incluindo
stringregex,
mergesort, findif
- Person e todas as classes derivadas
- Truck: reescreveu
- Cargo: reescreveu
- I/O ficheiros
- App: tabelas, opções das tabelas e mostrar detalhes de entidades
- Time, Address, Phonenummer, Temperature, VAT
- Service e função de alocação

Telmo Baptista

- utils: algumas funções
- App: input do utilizador; adição, edição e remoção de entidades
- Cargo: implementou
- Person: alguns métodos
- Time: algumas funções
- Truck: implementou

Luís Miranda

- Organização da app
- Permissões
- Dados
- Testes
- Apresentação