

Mestrado Integrado

Engenharia Informática e Computação

Algoritmos e Estruturas de Dados

Tema 5 - Empresa de mudanças (Parte 1)

Turma 2 – Grupo 4

Gonçalo Regueiras dos Santos | 201603265 | up201603265@fe.up.pt

João Lourenço Teixeira Vieira | 201603190 | up201603190@fe.up.pt

João Miguel Mendes Ribeiro Agulha | 201607930 | up201607930@fe.up.pt

24 de novembro de 2017



Universidade do Porto

FEUP Faculdade de
Engenharia

Índice

Descrição do Tema.....	1
Descrição da solução implementada	2
Diagramas UML.....	3
Lista de Casos de Utilização	4
Principais Dificuldades no Desenvolvimento	5
Indicação do Esforço de Cada Elemento.....	6
Referências bibliográficas	7

Descrição do Tema

O presente trabalho permite a uma empresa de mudanças gerir informaticamente a sua rede de clientes e os serviços que a mesma presta.

A gestão engloba tanto serviços prestados a empresas como a clientes particulares, oferecendo, para o mesmo tipo de serviços, modalidades de pagamento diferente, consoante o tipo de cliente.

Todo o processo de tratamento do(s) objeto(s) a transportar são geridos pela empresa, sendo possível uma localização precisa e constante do(s) mesmo(s).

Descrição da solução implementada

A empresa (*Company* - classe principal do programa) tem como atributos vetores para todos os clientes (classe *Client*), para todos os Serviços (class *Services*) e para todos os pagamentos (classe *Payment*), o NIB, a entidade e a referência para os pagamentos. Guarda ainda a data (classe *Data*) e a hora (classe *Hour*).

Cada serviço contém o endereço de origem e destino, o volume da encomenda (em m³), o preço e a data e hora de início e fim de todas as etapas do serviço (classes *Packaging*, *Shipping* e *Delivery*). Contem ainda um indicativo de se é ou não pagamento no fim do mês (class *EOMPayment*)

Cada cliente tem um nome, um endereço (classe *Address*), o seu NIF e dois vetores, um para os serviços requisitados e outro para os pagamentos correspondentes.

A classe *Address* guarda a morada, o número da porta, a cidade, o distrito e o país de um endereço.

Os clientes estão divididos em 3 tipos, particular (classe *Personal*), empresarial (classe *Business*) e não registados (classe *Unregistered*). Os clientes particulares e não registados têm acesso aos mesmos meios de pagamentos (classe *Payment*), pagamento por Multibanco (classe *DebitCard*) ou transferência bancária (classe *BankTransfer*). Os clientes empresariais, além destes meios de pagamento, podem ainda pagar por cartão de crédito (classe *CreditCard*).

Os clientes empresariais usufruem de dois serviços exclusivos, o armazenamento das encomendas nos armazéns da empresa e o pagamento no final do mês. O serviço de armazenamento é gratuito pelos primeiros 5 dias, após este prazo, passam a ter um custo diário. Se o cliente desejar, pode no final do mês pagar todos os serviços que ficaram pendentes ao longo do mês.

Por motivos de simplificação, quando o cliente (empresarial) requisita o armazenamento da encomenda nos armazéns da empresa, esse tempo é incluído no tempo de transporte, isto é, o número de dias entre o fim do shipping e o início do delivery representa o número de dias de armazenamento. Também por simplificação, quando a cidade de um endereço indicado pelo Cliente é uma capital de distrito de Portugal, considera-se as coordenadas geográficas do centro da cidade como sendo as do endereço (*Address*) introduzido pelo cliente.

A distância entre a origem e o destino do serviço é calculada através de um algoritmo de *Haversine* [1], utilizando as coordenadas GPS dos dois pontos.

O número de dias entre duas datas é calculado convertendo essas datas para o formato *Rata Die* [2], que representa uma data pelo número (inteiro) de dias desde o dia 1 de janeiro do ano 1. Assim, a diferença de dias entre duas datas torna-se uma operação trivial, permitindo-nos abstrair do número de dias num mês e se o ano é bissexto ou não.

Quanto a organização geral do programa, baseamo-nos no padrão de arquitetura de software, mais precisamente, MVC (Model-View-Controller), que implementados para cada caso de uso. O *modelo* (*model*) consiste nos dados da aplicação. O *view* pode ser qualquer saída de representação dos dados, como uma tabela ou um diagrama bem como a interação direta com o utilizador. E, finalmente, o controller é aquele que faz a integração, ou por outras palavras, a ponte entre o model e o view. Por esta razão, é o controller o responsável por se adaptar aos outros dois anteriormente referidos.

Diagramas UML



Figura 2 - Diagrama UML:Company Class

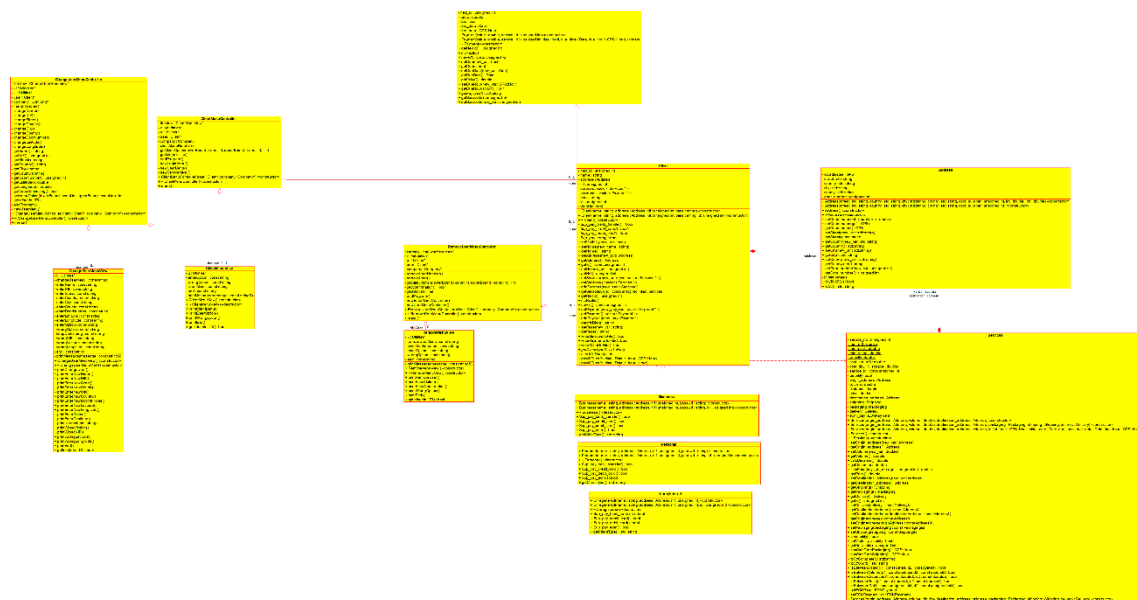


Figura 2 - Diagrama UML:Client Class

Mais imagens do diagrama serão disponibilizadas em anexo, assim como o ficheiro *proj.xmi*, com o mesmo.

Lista de Casos de Utilização

- Criar novo Cliente
- Login de Cliente já existente
- Requisitar novo serviço
- Ver serviço requisitado
- Listar serviços contratados
- Pagar serviço contratado
- Pagar os serviços do mês que ficaram por pagar (restrito a clientes empresariais)
- Ordenar serviços já requisitados
- Filtrar serviços já requisitados em função de critérios introduzidos pelo utilizador

Principais Dificuldades no Desenvolvimento

Gonçalo Santos:

Ao longo do desenvolvimento do projeto, foram surgindo alguns desafios, tal como o cálculo da distância entre dois locais, definidos por coordenadas geográficas ou operações aritméticas com datas.

João Vieira:

O projeto não se apresenta muito complicado ao nível técnico. Penso que a componente de gestão humana intragrupo se revela o ponto mais desafiante.

João Agulha:

Pelo facto de o projeto ter sido realizado utilizando um padrão de arquitetura de software, mais precisamente, MVC (Model-View-Controller), a integração destes foi se duvida a parte mais exigente e motivadora.

Indicação do Esforço de Cada Elemento

Todos os elementos concordam que o esforço foi igualmente repartido.

Referências bibliográficas

[1]

Baum, Peter. 2017. "Date Algorithms." Abril 28: 42-43.

[2]

Veness, Chris. n.d. *Calculate distance, bearing and more between Latitude/Longitude points*. Accessed Outubro 25, 2017. <https://www.movable-type.co.uk/scripts/latlong.html>.