



## **BDAD: Empresa de Entregas**

### **Relatório**

#### **Grupo 2:**

João Martins – up201706978

Miguel Charchalis - up201506074

José Miguel Mações – up201806622

## Resumo

Neste trabalho define-se uma possível base de dados relativa a uma suposta empresa de entregas, recorrendo para isso a modelação UML, ao Modelo Relacional e também a SQL, esta última não só como linguagem de definição de dados mas também como linguagem de manipulação de dados.

# Índice

1. Contexto	4
2. Esquema UML	5
3. Modelo Relacional	6
3.1 Relações	6
3.2 Análise de Dependências Funcionais e Formas Normais	7
3.3 Restrições	11
4. Interrogações	14
5. Gatilhos	16
6. Conclusões	18

# 1. Contexto

De cada cliente interessa saber o nome, o NIF, o email, a data de nascimento, o número de telemóvel e a palavra passe. Isto tudo também é informação necessária relativa a cada condutor, para além do número de segurança social e a data em que começou a trabalhar na plataforma. Para além disso interessa saber as horas de início e de fim da sessão de trabalho do condutor.

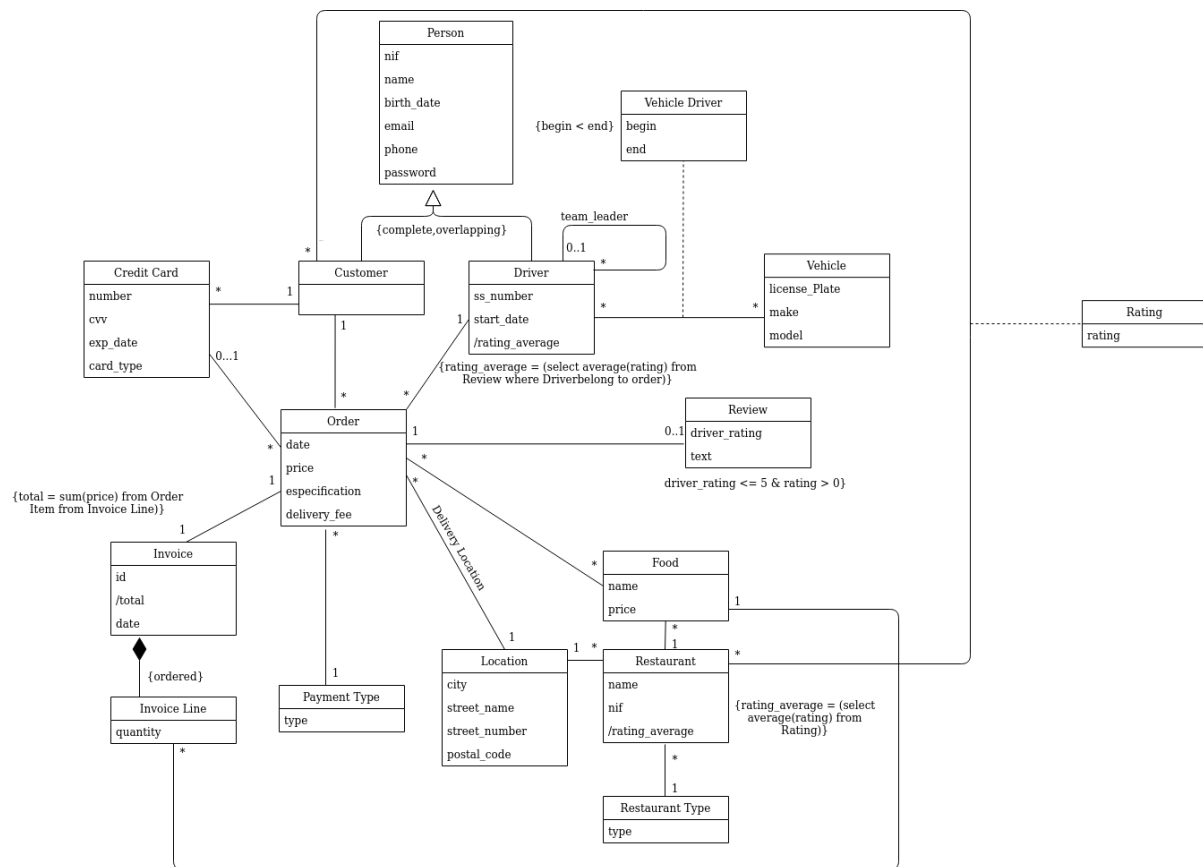
Cada cliente necessita de ter um cartão de crédito, do qual deve ser conhecido o número, o CVV, a data de validade e a rede (VISA, MasterCard, etc.)

Cada condutor deve ter a si associado um veículo, identificado pela matrícula, do qual interessa saber a marca e o modelo. Um condutor pode ser chefe de equipa, sendo responsável por outros condutores.

O cliente pode efetuar o pedido, que será entregue por um condutor e é constituído por uma data, um modo de pagamento e um preço, calculado a partir do preço da comida e da taxa de entrega. Para além disso terá uma avaliação que terá uma classificação, entre 1 e 5, e poderá ou não ter um texto. As classificações são usadas para calcular a média de classificações do condutor. O pedido estará também associado a um local de entrega, caracterizado pela cidade, nome da rua, número da rua, código postal.

O Restaurante tem como elementos identificativos o nome, o tipo, a localização, da qual interessa saber o mesmo que o local de entrega, e a classificação, é a média de classificações de clientes. Disponibiliza pratos, cada um com um nome e preço, que constituem os itens pedidos, juntamente com a quantidade de cada prato. Estes itens fazem parte do pedido e estão presentes na fatura, também associada ao pedido composta por um número identificativo, os itens pedidos e a sua quantidade, o preço total e a data.

## 2. Esquema UML



## 3. Modelo Relacional

### 3.1 Relações

Person (NIF, name, birth\_date, email, phone, password)

Customer (customerNIF -> Person)

Driver (driverNIF -> Person, ss\_number, start\_date, /rating\_average)

Vehicle (license\_plate, make, model)

VehicleDriver (driverNIF -> Driver, vehicle\_license\_plate -> Vehicle, begin, end)

Team (driverNIF -> Driver, leaderNIF -> Driver)

CreditCard (number, cvv, exp\_date, card\_type, customerNIF -> Customer)

Demand (demandID, date, /price, specification, delivery\_fee, customerNIF -> Customer, driverNIF -> Driver, locationID -> Location, paymentTypeID -> PaymentType, creditCardID -> CreditCard)

PaymentType (paymentTypeID, type)

Review (reviewID, rating, text, demandID -> Demand)

Rating (ratingID, rating, restaurantID -> Restaurant) - Class associação.

Food (foodID, name, price, restaurantID -> Restaurant)

Demanded (demandedID, foodID -> Food, demandID -> Demand, quantity)

Restaurant (restaurantID, name, NIF, locationID -> Location, /rating\_average, restaurantTypeID -> RestaurantType)

RestaurantType (restaurantTypeID, type)

Location (locationID, city, street\_name, street\_number, postal\_code)

Invoice (id, /total, date, demandID -> Demand)

InvoiceLine (invoice\_lineID, quantity, demandedID -> Demanded, invoiceID -> Invoice)

## 3.2 Análise de Dependências Funcionais e Formas Normais

### 3.2.1 Person

{NIF} -> {name, birth\_date, email, phone, password}

Não viola BCNF, pois o atributo da esquerda na dependência funcional constitui a chave da relação.

### 3.2.2 Customer

(Só um atributo)

### 3.2.3 Driver

{driverNIF} -> {ss\_number, start\_date, /rating\_average}

Não viola BCNF, pois o atributo da esquerda na dependência funcional constitui a chave da relação.

### 3.2.4 Vehicle

{license\_plate} -> {make, model}

Não viola BCNF, pois o atributo da esquerda na dependência funcional constitui a chave da relação.

### 3.2.5 VehicleDriver

{driverNIF, vehicle\_license\_plate} -> {begin, end}

Não viola BCNF, pois os dois atributos da esquerda na dependência funcional constituem chaves da relação.

### 3.2.6 Team

{diverNIF} -> {leaderNIF}

Não viola BCNF, pois o atributo da esquerda na dependência funcional constitui a chave da relação.

### 3.2.7 CreditCard

{number, customerNIF} -> {cvv, exp\_date, card\_type}

Viola BCNF pois duas pessoas podem usar o mesmo cartão, então a informação está repetida.

### 3.2.8 Demand (originalmente Order)

{demandID} -> {date, /price, specification, delivery\_fee, customerNIF, driverNIF, locationID, paymentTypeID, creditCardID}

{date} -> {delivery\_fee}

A segunda dependência funcional viola a BCNF porque o lado esquerdo (date) não é uma superchave, ou seja, o seu fecho não contém a totalidade dos atributos da relação.

Contudo, não viola a 3NF, uma vez que o atributo à direita (delivery\_fee) depende de demandID.

### 3.2.9 PaymentType

{paymentTypeID} -> {type}

Não viola BCNF, pois o atributo da esquerda na dependência funcional constitui a chave da relação.

### 3.2.10 Review

{reviewID} -> {rating, text, demandID}

Não viola BCNF, pois o atributo da esquerda na dependência funcional constitui a chave da relação.



### 3.2.11 Rating

{ratingID} -> {rating, restaurantID}

Não viola BCNF, pois o atributo da esquerda na dependência funcional constitui a chave da relação.

### 3.2.12 Food

{foodID} -> {name, price, restaurantID}

Viola BCNF porque o Restaurant, name e price dependem todos da foodID, mas não viola 3NF porque os atributos à direita que são parte de outra key.

### 3.2.13 Demanded

{demandedID} -> {foodID, demandID, quantity}

Viola BCNF porque a quantity, foodID e demandID dependem da demandedID, mas não viola 3NF porque todos os atributos da direita são parte de demandedID.

### 3.2.14 Restaurant

{restaurantID} -> {name, NIF, locationID, /rating\_average, restaurantTypeID}

Não viola BCNF, pois o atributo da esquerda na dependência funcional constitui a chave da relação.

### 3.2.15 RestaurantType

{restaurantTypeID} -> {type}

Não viola BCNF, pois o atributo da esquerda na dependência funcional constitui a chave da relação.

### 3.2.16 Location

{locationID} -> {city, street\_name, street\_number, postal\_code}

Não viola BCNF, pois o atributo da esquerda na dependência funcional constitui a chave da relação

### 3.2.17 Invoice

{id} -> {total, date, demandID}

Viola BCNF porque date depende da demandID, mas

não viola 3NF porque todos os atributos da direita são parte de outra key.

### 3.2.18 InvoiceLine

{invoice\_lineID} -> {quantity, demandedID, invoiceID}

Viola BCNF porque quantity depende de demandedID, mas

não viola 3NF porque todos os atributos da direita são parte de outra key.

### **3.3 Restrições**

#### **3.3.1 Person**

NIF, birth\_date, phone, password não podem ser nulos.

Comprimento de phone maior ou igual a 9.

NIF tem de ter tamanho 9.

#### **3.3.2 Customer**

#### **3.3.3 Driver**

Comprimento ss\_number maior ou igual a 11.

#### **3.3.4 Vehicle**

License\_plate, make, model não podem ser nulos.

Tamanho de license\_plate igual a 8 (a contar com "-").

#### **3.3.5 VehicleDriver**

Nenhum atributo pode ser nulo.

#### **3.3.6 Team**

Cada driver só tem um leader, ou nenhum, se for ele próprio leader.

#### **3.3.7 CreditCard**

Nenhum atributo pode ser nulo.

Comprimento de number maior ou igual a 13 e menor ou igual a 19.

Comprimento de cvv igual a 3.

### **3.3.8 Demand (originalmente Order)**

Date, delivery\_fee, price, customerNIF, driverNIF, locationID e paymentTypeID não podem ser nulos.

delivery\_fee maior ou igual a 0.

price maior ou igual 0 (não inclui delivery\_fee).

### **3.3.9 PaymentType**

Nenhum atributo pode ser nulo.

### **3.3.10 Review**

Pode não existir uma review numa demand, mas se existir não pode ter atributos nulos.

Não pode haver mais que uma review por demand.

Rating maior ou igual a 1 e menor ou igual a 5.

### **3.3.11 Rating**

Pode não existir um rating para um restaurant, mas se existir não pode ter atributos nulos.

Não pode haver mais que um rating por restaurant.

Rating maior ou igual a 1 e menor ou igual a 5.

### **3.3.12 Food**

foodID, name, price e restaurantID não podem ser nulos.

price tem de ser maior que 0.

### **3.3.13 Demanded**

#### **3.3.14 Restaurant**

restaurantID, name, NIF, locationID e restaurantTypeID não podem ser nulos.

NIF tem de ter tamanho 9.

Não pode haver 2 Restaurant com o mesmo restaurantID.

#### **3.3.15 RestaurantType**

Não pode haver atributos nulos.

#### **3.3.16 Location**

locationID, city, street\_name e postal\_code não podem ser nulos.

#### **3.3.17 Invoice**

invoiceID, total, date e demandID não podem ser nulos.

#### **3.3.18 InvoiceLine**

invoice\_lineID, demandedID e invoiceID não podem ser nulos.

## 4. Interrogações

### 4.1 Informação relativa a uma encomenda

Apresenta os vários itens da encomenda, como preço, nome do restaurante, local de entrega, etc.

### 4.2 Mês com mais faturação

Exibe o mês no qual a faturação dos restaurantes foi maior.

### 4.3 Lista dos 5 clientes com mais dinheiro gasto em encomendas

Apresenta uma lista do *top-5* de clientes ordenado pelo valor total gasto em encomendas, decendentemente.

### 4.4 Total de faturação por tipo de restaurante

Exibe uma lista do total de faturação agrupado por tipo de restaurante, ordenada decendentemente.

### 4.5 Lista de 5 restaurantes com maior número de encomendas

Apresenta uma listagem do *top-5* de restaurantes ordenado pelo número de encomendas a eles associadas.

### 4.6 Comidas nunca encomendadas

Mostra uma lista de todas as comidas que nunca foram associadas a nenhuma encomenda, ordenadas pelo nome do restaurante, ascendentemente.

### 4.7 Clientes que encomendaram apenas de um restaurante

Apresenta uma lista de todos os clientes que, sempre que encomendaram, foi do mesmo restaurante, bem como o restaurante em questão.

#### **4.8 Clientes que nunca encomendaram de um restaurante específico**

Exibe uma lista de todos os clientes que nunca fizeram nenhuma encomenda de um restaurante específico.

#### **4.9 Tipo de pagamento mais usado por cidade**

Mostra os tipos de pagamento mais utilizados agrupados por cidade, ordenados por ordem alfabética das cidades.

#### **4.10 Conclusão relativa a se compensa mais trabalhar em equipa ou a solo**

Apresenta uma conclusão relativa à questão de quem ganha mais serem os condutores que pertencem a uma equipa ou os que trabalham sozinhos.

## 5. Gatilhos

### 5.1 Cálculo automático da média do rating de um condutor após inserção de uma review

Aquando da inserção de uma Review, é calculada a média de rating do condutor associado a essa Review já tendo em conta o rating relativo à Review.

### 5.2 Cálculo automático da média do rating de um restaurante após inserção de um rating

Aquando da inserção de um Rating, é calculada a média de rating do restaurante associado a esse Rating já o tendo em conta.

### 5.3 Cálculo automático do preço de uma encomenda quando lhe é adicionado um item

Sempre que é adicionado um item (Demanded) à encomenda (Demand), o seu preço é adicionado ao total da encomenda, bem como é adicionado o valor da taxa de entrega.

### 5.4 Associação automática do cartão de crédito do cliente que fez a encomenda, se for esse o método de pagamento

Se o método de pagamento (PaymentType) de uma encomenda (Demand) for cartão de crédito, o cartão de crédito do cliente que fez a encomenda é associado à própria encomenda quando esta é inserida.

### 5.5 Atualização do total de uma fatura aquando da inserção de uma nova linha na fatura

Quando é inserido uma nova linha na fatura (InvoiceLine), é utilizado o valor do preço dessa linha e adicionado ao total da fatura (Invoice) a que a linha pertence.

### 5.6 Criação automática de uma linha de fatura quando é inserido um item na encomenda

Se é inserido um novo item (Demanded) na encomenda, é automaticamente criada uma linha de fatura (InvoiceLine) a si associada.



## 5.7 Criação automática de uma fatura quando é inserida uma encomenda

Se é inserido uma nova encomenda (Demand), é automaticamente criada uma fatura (Invoice) a si associada.

## 6. Conclusões

Na generalidade, este trabalho correu como esperado, sendo o único obstáculo a obrigação de serem criados mais *triggers* que os três pedidos só para o funcionamento da base de dados.