



universidade  
de aveiro

# Relatório - UberEats

BD 18/19

André Neves 84948  
Marlene Bastos 76346

## Introdução

Este projeto foi desenvolvido na unidade curricular Bases de Dados.

UberEats é uma aplicação destinada à gestão de um serviço de entregas ao domicílio, no ramo da restauração. É destinada apenas a uso administrativo. A interface dispõe vários estabelecimentos de restauração e respetivos produtos para venda.

Oferece funcionalidades sobre encomendas, motoristas e clientes num panorama geral. No entanto, não são as únicas entidades presentes que serão apresentadas mais à frente.

Para ajuda a visualização do funcionamento da base de dados, foi desenvolvida uma interface em `c#`, utilizando Visual Studio.

## Requisitos

Um **Cliente** realiza encomendas aos estabelecimentos aderentes e são entregues por um motorista. Cada cliente pode utilizar um código promocional para descontar ao preço final, ou então proporcionar o seu para que outro cliente o utilize.

Um **Motorista** depois de aceitar uma encomenda dirige-se ao estabelecimento em causa e entrega a encomenda ao cliente. Tem a si associado a matrícula e a marca do veículo e uma avaliação.

Um **Estabelecimento** é caracterizados por nome, tipo (hamburgueria, pizzeria, etc), morada, horário, produtos, avaliação e identificador único.

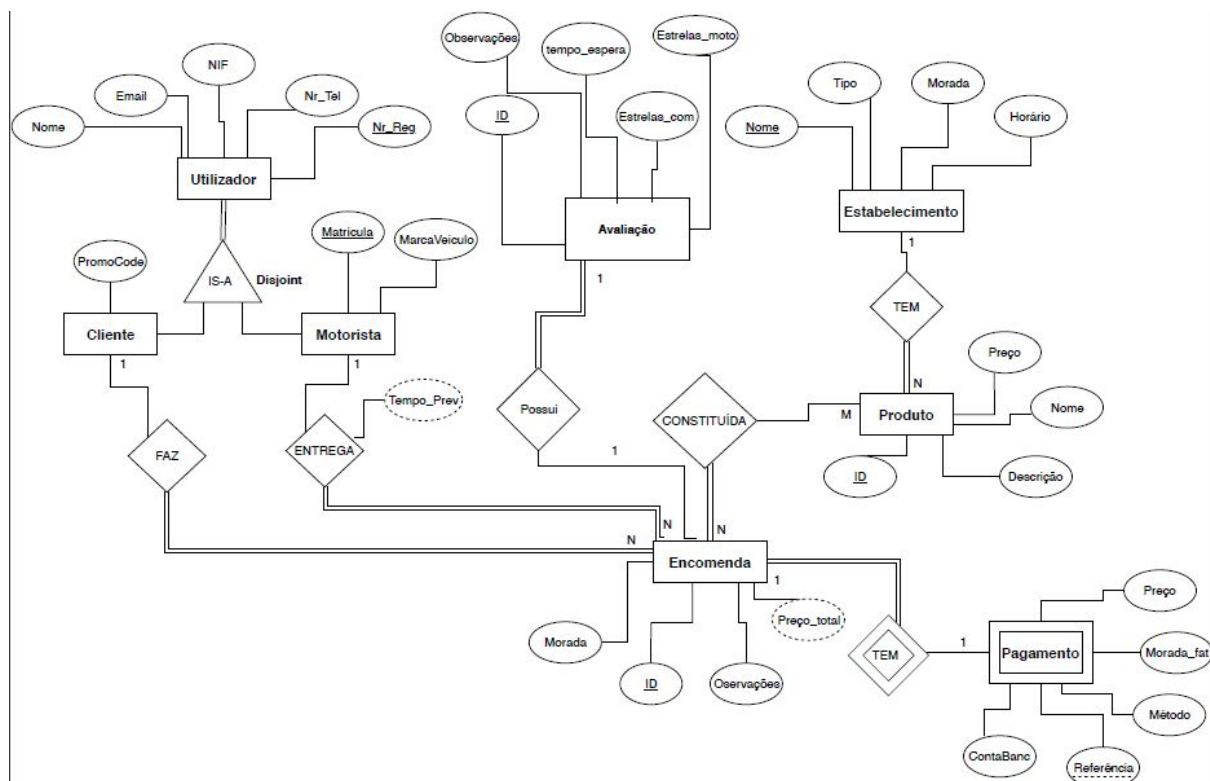
Um **Produto** é caracterizado por um nome, identificador único, preço e descrição.

Uma **Encomenda** é realizada por um cliente. É constituída por um ou mais produtos, preço total, morada, observações (caso o cliente tenha pedidos especiais, instruções, etc), identificador único e tempo previsto de entrega.

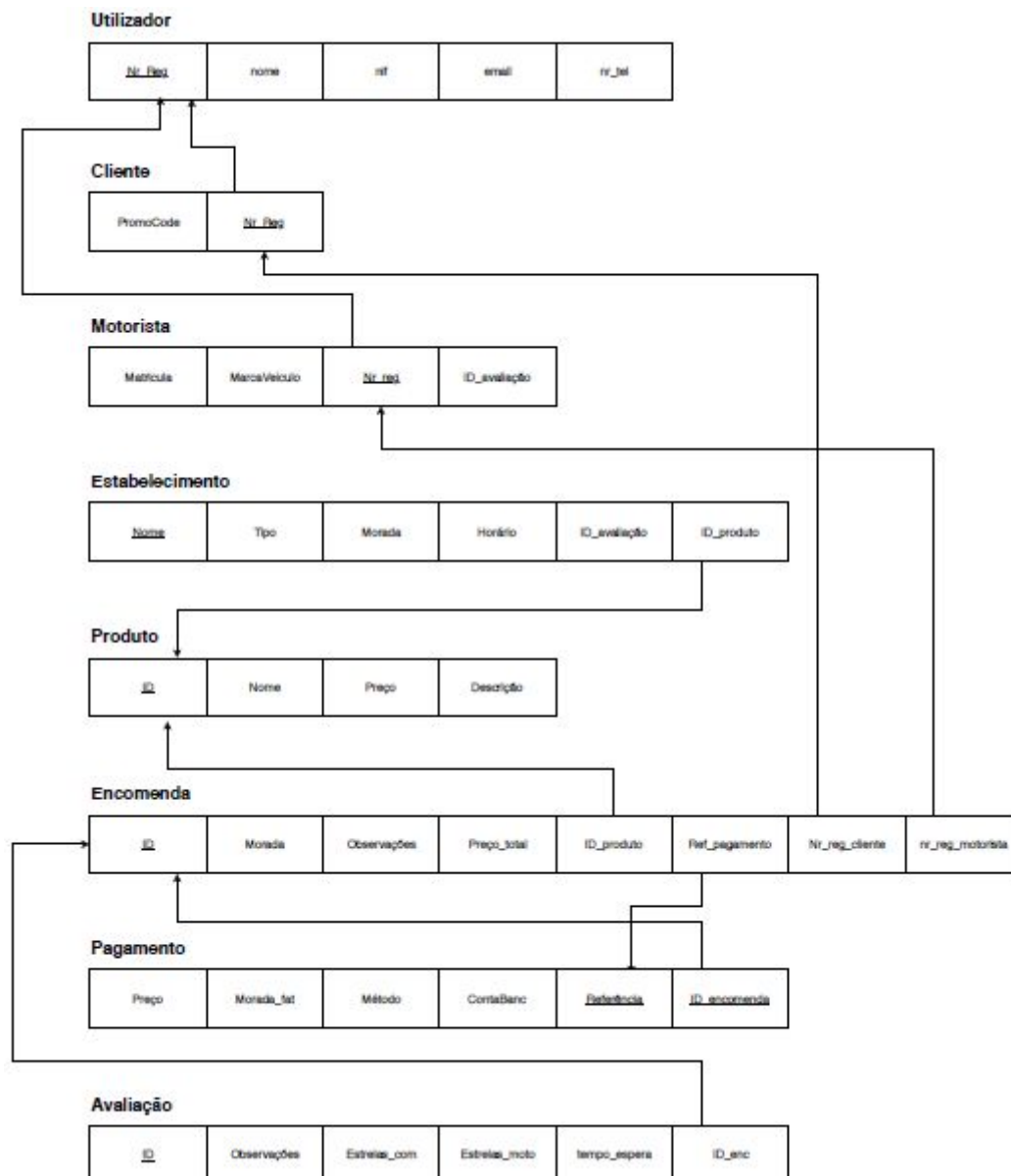
Uma **Avaliação** é atribuída a motoristas e estabelecimentos, caracterizada por número de estrelas (0 a 10) e descrição. No entanto, é opcional.

O **Pagamento** possui um preço, método de pagamento e pode ter uma conta bancária associada e morada para facturação.

## Diagrama Entidade-Relação (DER)



# Modelo Entidade Relação (MER)



A transformação do diagrama entidade relação foi direta. Como se tratava de um projeto com alguma simplicidade, as relações já se encontravam todas na forma BCNF. Posto isto, não foi necessário fazer o processo de Normalização.

## Triggers

Foram criados dois triggers, um na tabela Cliente e outro na tabela Motorista, ambos para controlar a inserção de um novo elemento. Estes triggers lidam com a situação de poder já existir na tabela um cliente ou um motorista igual aos que se tentam adicionar, de forma a não duplicar dados.

## Stored Procedures

Todas as interações de inserção, edição e eliminação entre a base de dados e interface gráfica foram feitas através de Stored Procedures.

## UDF's e Views

Não utilizamos views pois não sentimos essa necessidade. Ao invés, utilizamos User Defined Functions de dois tipos:

- **retornam inteiros:** utilizadas para retornar número total de objetos de uma certa entidade;
- **retornam tabelas:** utilizadas para retornar tabelas de objetos específicos que queiramos obter.

## SQL DDL

Exemplo de criação de uma tabela:

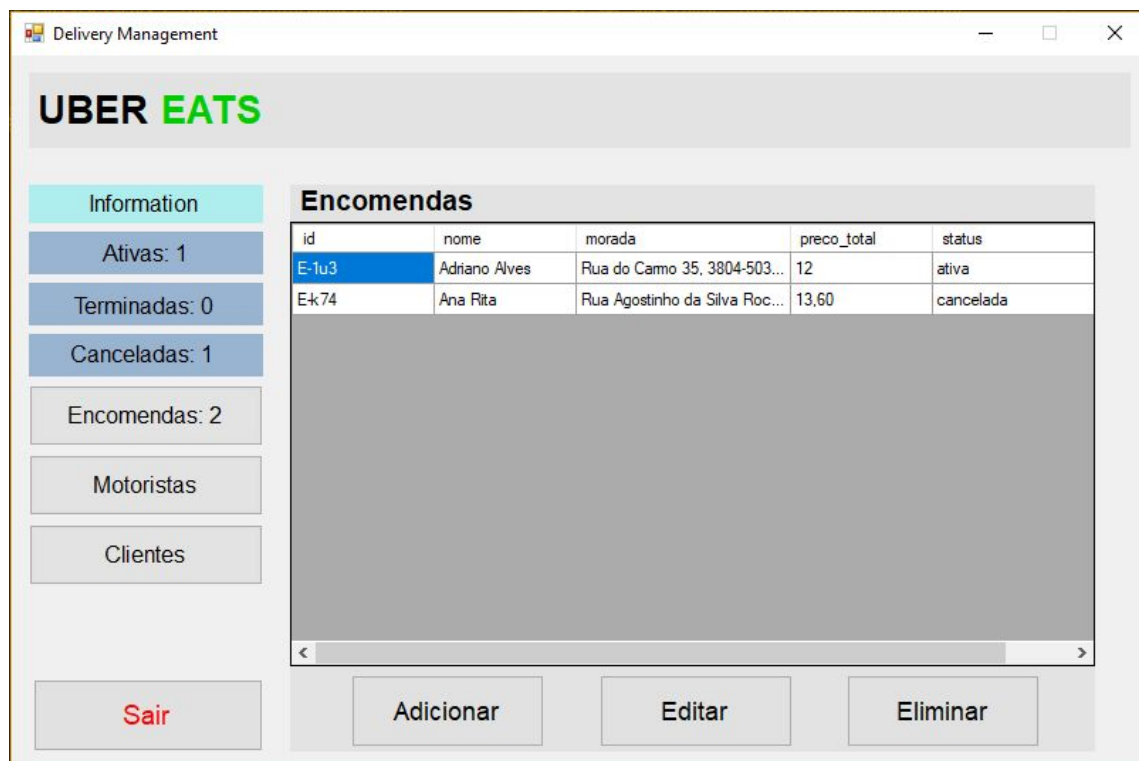
```
]create table ServEntr.Encomenda(  
  id          varchar(5),  
  morada      varchar(255) not null unique,  
  obs         varchar(255),  
  preco_total decimal(6,2),  
  ref_pagamento int,  
  nr_reg_cliente int not null unique,  
  nr_reg_motorista int not null unique,  
  id_avaliacao int,  
  primary key(id),  
  foreign key(nr_reg_cliente) references ServEntr.Cliente(nr_reg)  
    on delete cascade, -- ao apagar o cliente, apaga as encomendas dele  
  foreign key(nr_reg_motorista) references ServEntr.Motorista(nr_reg)  
    on delete cascade, -- ao apagar o motorista, apaga as encomendas dele  
  foreign key(id_avaliacao) references ServEntr.Avaliacao(id));  
  
alter table ServEntr.Encomenda add status varchar(15);
```

# SQL DML

Exemplo de criação de um Stored Procedure:

```
-- INSERE NOVO CLIENTE --
create proc ServEntr.NewCliente (@nome varchar(255), @nif int, @nr_tel int, @email varchar(255), @morada varchar(255))
as
begin
    declare @id as int;
    select @id = max(nr_reg)+1 from ServEntr.Cliente;
    insert into ServEntr.Cliente(nr_reg, nome, nif, nr_tel, email, morada) values (@id, @nome, @nif, @nr_tel, @email, @morada);
end
GO
```

## Interface Gráfica



Delivery Management

UBER EATS

Information

Ativas: 1

Terminadas: 0

Canceladas: 1

Encomendas: 2

Motoristas

Cientes

Sair

Motoristas

nr_reg	nome	nif	nr_tel	marcaveiculo	matricula
3	Tânia Coelho	254846227	934654229	Renault	10-47-NK
4	Rita Martins	354268559	961542278	Renault	UJ-87-46
5	Dany Costa	845461227	934656115	Mercedes-Benz	50-47-KL
6	Osvaldo Carolino	254811089	913452189	BMW	20-CN-54
7	Pedro Pedrosa	754211989	962157789	Skoda	89-15-XD
10	Maria Cunha	123456789	915462514	Renault	14-BG-78

Adicionar

Editar

Eliminar

Delivery Management

UBER EATS

Information

Ativas: 1

Terminadas: 0

Canceladas: 1

Encomendas: 2

Motoristas

Cientes

Sair

Cientes

nr_reg	nome	nif	nr_tel	email	morada
1	Alberto Santos	187293726	910926123	albsantos@hotm...	Rua de Adria
2	Adriano Alves	654812350	915846235	aalves@hotmail....	Rua do Cam
5	Ana Rita	858412350	936420805	anarita@gmail.com	Rua Agostin
15	Cátia Andreia	548712693	918436022	lenelinda@ua.pt	Rua Aguinch
18	Anastácio Joaquim	125487963	965546651	anastjoaqu@hot...	Rua do Grav

Adicionar

Editar

Eliminar

Delivery Management

UBER EATS

Information

Ativas: 1

Terminadas: 0

Canceladas: 1

Encomendas: 2

Motoristas

Clientes

Sair

Nova Encomenda

Estabelecimento:

Cliente:

Produtos:

Obs:

Método: 

PayPal

MBWay

Preço:  €

Cancelar

OK