

# DOCUMENTAÇÃO DE BANCO DE DADOS

Documentação elaborada para composição da nota da Disciplina de Projeto de Banco de Dados, orientado pelo prof<sup>a</sup> Tacyana Batista.

Alunos: Rodrigo Antonio e Gustavo Rodrigues

## Sumário

Minimundo  
MER  
DER  
Scripts DDL  
Scripts DML

## **Minimundo:**

Com o intuito de gerenciar e expandir o comércio na cafeteria FlowCoffee, o dono pediu para que um desenvolvedor o ajudasse na elaboração de um banco de dados cumprindo alguns requisitos. Um cliente pode realizar vários pedidos e cada pedido pode ser associado a um funcionário responsável pelo atendimento e pode ter um ou mais produtos, a entrega (delivery) de um pedido só pode ser realizada por um único funcionário.

Ao entrar no estabelecimento o cliente deve se dirigir a um dos caixas disponíveis para iniciar a realização do seu pedido.

O funcionário que realiza os pedidos ao iniciar o atendimento deve perguntar se o cliente possui cadastro e se o cliente possuir ele continua normalmente, caso contrário ele deve criar um cadastro para ele com as seguintes informações: nome, cpf e endereço caso queira delivery.

O cliente pode escolher se deseja consumir no local ou se prefere a entrega via delivery, se tiver informado o endereço no cadastro, onde um funcionário será designado para levar o pedido até o endereço do cliente. Os clientes que optam por consumir no local aguardam no balcão, onde seus pedidos são chamados pela ordem do horário em que foram realizados, ou seja, quem fez o pedido primeiro tem prioridade na entrega do pedido em relação aos pedidos realizados posteriormente.

Os caixas da loja têm acesso aos produtos que contém todos os pratos e bebidas disponíveis e seus respectivos valores para facilitar o processo de realização dos pedidos. Cada pedido contém informações detalhadas sobre os produtos escolhidos pelo cliente, o valor total e o horário em que foi realizado.

Os funcionários podem ser dos tipos: atendentes de caixas e entregadores responsáveis pela entrega dos pedidos via delivery.

O sistema deve permitir consultar o histórico de pedidos de cada cliente, além de possibilitar o controle dos funcionários responsáveis pelos atendimentos e entregas, garantindo o acompanhamento completo do ciclo do pedido desde o registro até a entrega final.

## MER:

Cliente (id\_clientePK, nome\_cliente, endereço, cpf\_cliente)

Funcionários (id\_funcionárioPK, nome\_funcionário, cargo, cpf\_funcionário, senha, ativo)

Produtos (id\_produtoPK, nome\_produto, descricao, valor\_produto, categoria, ativo)

Pedidos (id\_pedidoPK, id\_clienteFK, id\_funcionárioFK, hora\_pedido, valor\_pedido, modalidade, pagamento, status)

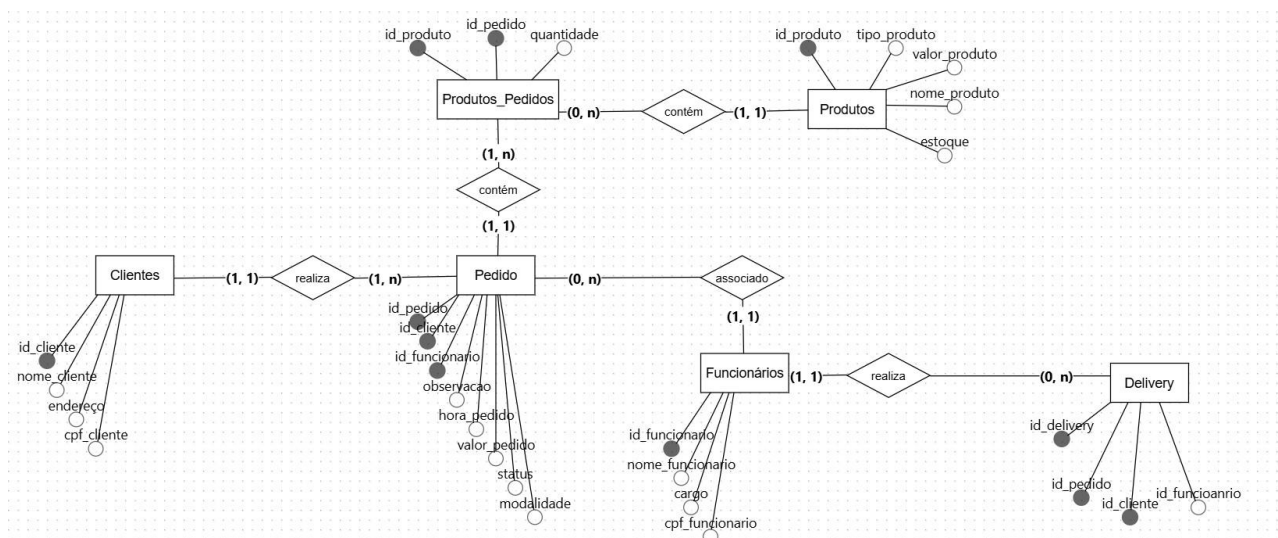
Delivery (id\_deliveryPK, id\_pedidoFK, id\_funcionarioFK,)

Produtos\_Pedidos (id\_produtoFK, id\_pedidoFK, quantidade)

## RELACIONAMENTOS E CARDINALIDADES:

- (1,1) Cliente -- realiza -- pedido (1,n)
- (0,n) Pedido -- é associado -- funcionário (1,1)
- (0,n) Produtos\_Pedidos -- Contem -- Produtos (1,1)
- (1,1) Pedidos -- Contém -- Produtos\_Pedidos (1,n)
- (1,1) Funcionário -- realiza -- delivery (0,n)

## DER:



## SCRIPTS DDL:

```
create table CLIENTES(  
    id_cliente serial primary key,  
    nome_cliente varchar(255) not null,  
    cpf_cliente char(11) unique not null,  
    endereco varchar(255)  
);  
  
create table FUNCIONARIOS(  
    id_funcionario serial primary key,  
    nome_funcionario varchar(255) not null,  
    cpf_funcionario char(11) not null unique,  
    cargo char(1) not null check (cargo in ('A', 'C', 'E')),  
    senha bytea not null,  
    ativo boolean not null default true  
);  
  
create table PRODUTOS(  
    id_produto serial primary key,  
    nome_produto varchar(255) not null unique,  
    descricao text,  
    valor_produto double precision not null,  
    categoria char(1) not null check (categoria in ('C', 'B', 'D', 'S')),  
    ativo boolean not null default true  
);  
  
create table PEDIDOS(  
    id_pedido serial primary key,  
    id_cliente integer not null,  
    constraint fk_cliente foreign key (id_cliente) references CLIENTES (id_cliente),  
    id_funcionario integer not null,  
    constraint fk_funcionario foreign key (id_funcionario) references FUNCIONARIOS  
    (id_funcionario),  
    hora_pedido timestamp not null default CURRENT_TIMESTAMP,  
    valor_pedido double precision not null default 0,  
    modalidade char(1) check (modalidade in ('L', 'E')),  
    pagamento char(1) not null check (pagamento in ('P', 'C', 'D')),  
    status char(1) not null check (status in ('P', 'C', 'X')) default 'P'  
);  
  
create table DELIVERY(  
    id_delivery serial primary key,  
    id_pedido integer not null,  
    constraint fk_pedido foreign key (id_pedido) references PEDIDOS (id_pedido),  
    id_funcionario integer,  
    constraint fk_funcionario foreign key (id_funcionario) references FUNCIONARIOS  
    (id_funcionario)  
);  
  
create table PRODUTOS_PEDIDOS(  
    id_pedido integer not null,
```

```
id_produto integer not null,  
primary key (id_pedido, id_produto),  
quantidade integer not null default 1,  
observacao text,  
constraint fk_pedido foreign key (id_pedido) references PEDIDOS (id_pedido),  
constraint fk_produto foreign key (id_produto) references PRODUTOS (id_produto)  
);
```

