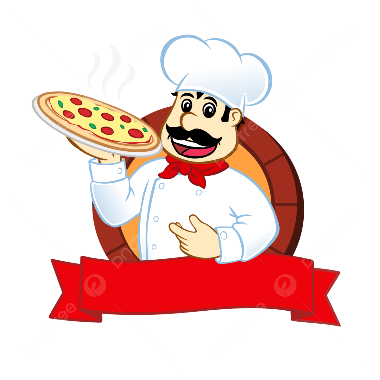
**Pizzaria do Messi**

****

## Integrantes: **Guthierry Silva De Souza,**

## **Joaõ Vitor Revoeda Bontempo,**

Matheus de Sousa Holanda

## Instituição: **UDF**

## Data: **09/06/2025**

## 

Sumário

## Introdução

## Desenvolvimento

### Cenário

### 2.2 Modelagem conceitual

### 2.3 Modelagem lógica

### 2.4 Modelagem Física (DDL)

### 2.5 Dados (DML)

## Resultados

## Conclusão

**1. Introdução**

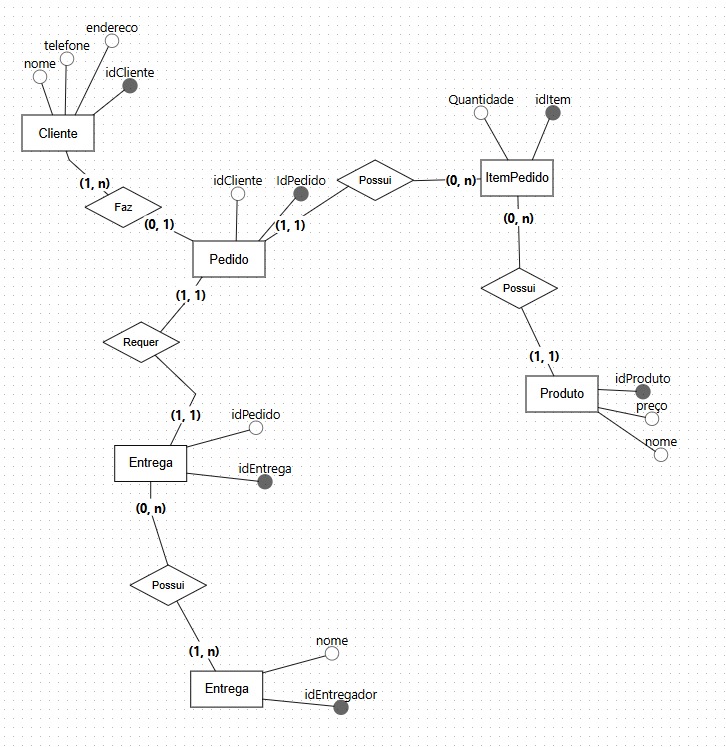
## Este relatório apresenta o desenvolvimento de um banco de dados para a Pizzaria do Messi, com o objetivo de organizar as informações dos pedidos, clientes, produtos, entregadores e entregas realizadas. A modelagem de dados é uma etapa essencial no projeto de sistemas de informação, pois define como os dados serão estruturados, armazenados e relacionados entre si. O trabalho foi desenvolvido utilizando o MySQL Workbench para a criação do modelo conceitual, lógico e físico.

**2. Desenvolvimento**

# ****2.1 Cenário****

A Pizzaria do Messi realiza vendas de pizzas diretamente aos clientes, registrando os pedidos realizados, os produtos solicitados, os entregadores responsáveis e os dados dos clientes. A estrutura do banco de dados deve permitir o controle eficiente dessas informações, facilitando a geração de relatórios, análise de vendas e logística de entregas.

# ****2.2Modelagem conceitual****



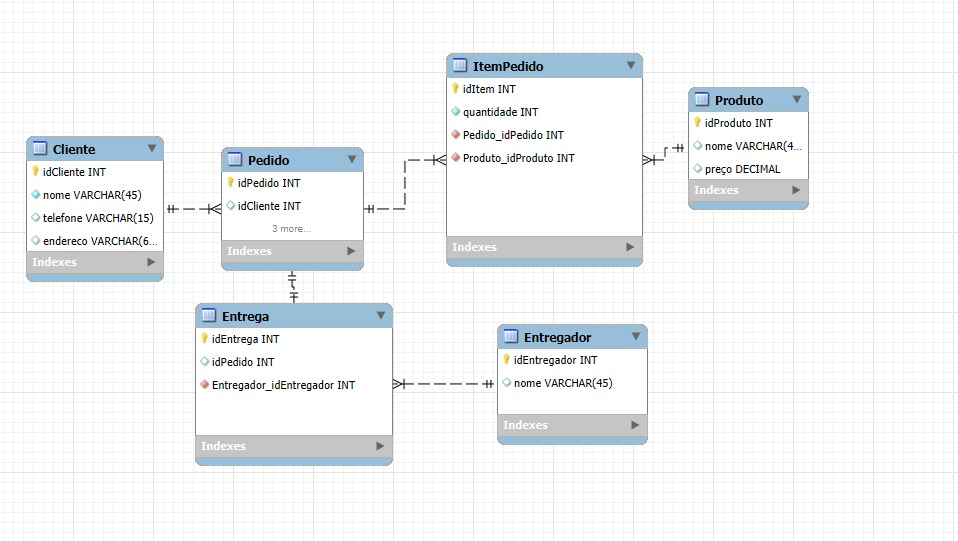
### ****Explicação da Modelagem Conceitual****

A imagem representa o **diagrama entidade-relacionamento** do banco de dados da Pizzaria do Messi, estruturando as principais entidades envolvidas na operação da pizzaria.

* **Cliente**: armazena dados pessoais como nome, telefone e endereço. Está relacionado com a entidade **Pedido**, pois cada pedido é feito por um cliente.
* **Pedido**: representa uma compra feita pelo cliente. Está ligado aos **Itens do Pedido**, pois cada pedido pode conter vários produtos, e também à **Entrega**, pois cada pedido é entregue por um entregador.
* **Produto**: armazena informações das pizzas ou itens vendidos, como nome e preço. Está relacionado com **ItemPedido**, que define quais produtos foram comprados em cada pedido.
* **ItemPedido**: entidade associativa que liga os **Pedidos** aos **Produtos**, indicando quais produtos foram comprados em quais pedidos e em que quantidade.
* **Entregador**: representa os funcionários responsáveis pelas entregas dos pedidos.
* **Entrega**: associa cada pedido ao entregador responsável. Indica quem entregou o quê.

Esse modelo permite registrar e consultar de forma organizada todas as informações de clientes, pedidos, produtos vendidos e entregas feitas, garantindo integridade e eficiência no gerenciamento dos dados.

# ****2.3 Modelagem Lógica****

****

**Tabela: Cliente**

idCliente INT AUTO\_INCREMENT PRIMARY KEY,

    nome VARCHAR(100) NOT NULL,

    telefone VARCHAR(15),

    endereco VARCHAR(200)

**Tabela: Produto**

idProduto INT AUTO\_INCREMENT PRIMARY KEY,

    nome VARCHAR(100) NOT NULL,

    preco DECIMAL NOT NULL

**Tabela: Pedido**

idPedido INT AUTO\_INCREMENT PRIMARY KEY,

    idCliente INT,

    total DECIMAL,

    FOREIGN KEY (idCliente) REFERENCES Cliente(idCliente)

**Tabela: ItensPedido**

idItem INT AUTO\_INCREMENT PRIMARY KEY,

    idPedido INT,

    idProduto INT,

    quantidade INT NOT NULL,

    FOREIGN KEY (idPedido) REFERENCES Pedido(idPedido),

    FOREIGN KEY (idProduto) REFERENCES Produto(idProduto)

);

**Tabela: Entregador**

 idEntregador INT AUTO\_INCREMENT PRIMARY KEY,

    nome VARCHAR(100) NOT NULL

**Tabela: Entrega**

idEntrega INT AUTO\_INCREMENT PRIMARY KEY,

    idPedido INT,

    idEntregador INT,

    FOREIGN KEY (idPedido) REFERENCES Pedido(idPedido),

    FOREIGN KEY (idEntregador) REFERENCES Entregador(idEntregador)

# ****2.4 Modelagem Física (DDL)****

Abaixo estão os scripts de criação das tabelas (DDL):

CREATE TABLE Cliente (

    idCliente INT PRIMARY KEY AUTO\_INCREMENT,

    nome VARCHAR(45),

    telefone VARCHAR(15),

    endereco VARCHAR(60)

);

CREATE TABLE Produto (

    idProduto INT PRIMARY KEY AUTO\_INCREMENT,

    nome VARCHAR(45),

    preco DECIMAL(10,2)

);

CREATE TABLE Pedido (

    idPedido INT PRIMARY KEY AUTO\_INCREMENT,

    idCliente INT,

    total DECIMAL(10,2),

    FOREIGN KEY (idCliente) REFERENCES Cliente(idCliente)

);

CREATE TABLE ItemPedido (

    idItem INT PRIMARY KEY AUTO\_INCREMENT,

    quantidade INT,

    Pedido\_idPedido INT,

    Produto\_idProduto INT,

    FOREIGN KEY (Pedido\_idPedido) REFERENCES Pedido(idPedido),

    FOREIGN KEY (Produto\_idProduto) REFERENCES Produto(idProduto)

);

CREATE TABLE Entregador (

    idEntregador INT PRIMARY KEY AUTO\_INCREMENT,

    nome VARCHAR(45)

);

CREATE TABLE Entrega (

    idEntrega INT PRIMARY KEY AUTO\_INCREMENT,

    idPedido INT,

    Entregador\_idEntregador INT,

    FOREIGN KEY (idPedido) REFERENCES Pedido(idPedido),

    FOREIGN KEY (Entregador\_idEntregador) REFERENCES Entregador(idEntregador)

);

# ****2.5 Dados (DML)****

Os dados abaixo foram inseridos no banco para testes:

INSERT INTO Cliente (nome, telefone, endereco) VALUES

('Cliente1', '11999990000', 'Casa1'),

('Cliente2', '11888887777', 'Casa2'),

('Cliente3', '11777777777', 'Casa3');

INSERT INTO Produto (nome, preco) VALUES

('Calabresa', 30.00),

('Frango com Catupiry', 30.00),

('Banana', 30.00),

('Quatro Queijos', 30.00);

INSERT INTO Pedido (idCliente, total) VALUES

(1, 73.00),

(2, 70.00);

INSERT INTO ItemPedido (idPedido, idProduto, quantidade) VALUES

(1, 1, 1),  -- Pedido1

(1, 2, 1),  -- Pedido2

(2, 4, 2);  -- Pedido3

INSERT INTO Entregador (nome) VALUES

('Motoboy1'),

('Motoboy2'),

('Motoboy3');

INSERT INTO Entrega (idPedido, idEntregador) VALUES

(1, 1),

(2, 2);

**3. Resultados**

Exemplos de consultas SQL com respostas esperadas:

**Consulta 1: Listar todos os clientes**

#### sql

#### SELECT \* FROM **Cliente;**

**Consulta 2: Listar todos os pedidos e seus valores**

#### sql

#### SELECT \* FROM **Pedido;**

**Consulta 3: Consultar os produtos de um pedido específico**

#### sql

#### SELECT **p.nome, ip.quantidade**

#### FROM **Produto p**

#### JOIN **ItensPedido ip ON p.idProduto = ip.Produto\_idProduto**

#### WHERE **ip.Pedido\_idPedido = 1;**

**Consulta 4: Ver entregador responsável por cada entrega**

#### sql

#### SELECT **e.idPedido, en.nome**

#### FROM **Entrega e**

#### JOIN **Entregador en ON e.Entregador\_idEntregador = en.idEntregador;**

A seguir está um exemplo de resultado de uma consulta SQL que exibe o número do pedido, o nome do cliente, o valor total do pedido e o entregador responsável:

sql

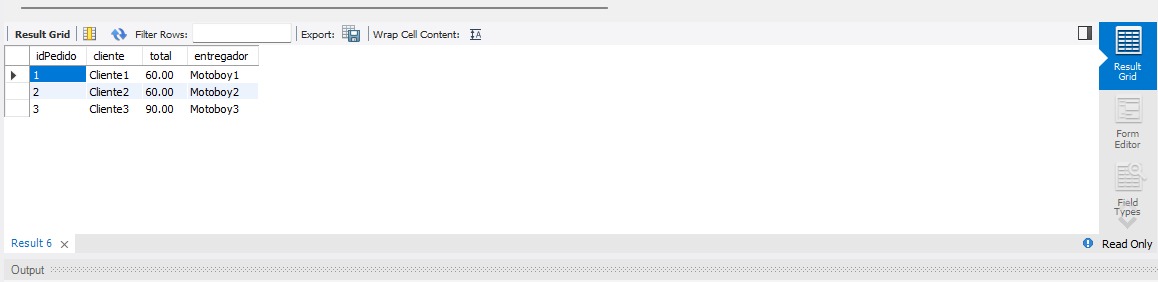
SELECT p.idPedido, c.nome AS cliente, p.total, e.nome AS entregador

FROM Pedido p

JOIN Cliente c ON p.idCliente = c.idCliente

JOIN Entrega en ON p.idPedido = en.idPedido

JOIN Entregador e ON en.Entregador\_idEntregador = e.idEntregador;

****

**Explicação:**  
A consulta faz uma junção entre as tabelas Pedido, Cliente, Entrega e Entregador para retornar um resumo completo de cada pedido. Ela mostra o nome do cliente que realizou o pedido, o valor total e o entregador que realizou a entrega. Essa consulta é útil para gerar relatórios de entregas e verificar a distribuição de pedidos entre entregadores.

**4. Conclusão**

## A criação do banco de dados para a Pizzaria do Messi permitiu a organização eficiente dos dados relacionados aos pedidos, clientes, entregas e produtos. Através da modelagem conceitual, lógica e física, foi possível representar todas as entidades e seus relacionamentos de forma clara e estruturada. O uso de SQL para consultas demonstrou a capacidade do sistema em fornecer informações úteis para a gestão do negócio. Com isso, a pizzaria poderá melhorar sua operação e atendimento aos clientes.