

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLO-
GIA DE SÃO PAULO
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
CJOBDD2**

TIAGO JARRUGE SARAIVA

**SISTEMA DE PEDIDOS ONLINE PARA CANTINAS:
CANTINAON**

**PROFESSOR PAULO GIOVANI DE FARIA ZEFERINO
CAMPOS DO JORDÃO
2025**

RESUMO

O projeto **CantinaON** propõe a implementação de um banco de dados para um sistema de pedidos online voltado para cantinas escolares. O objetivo principal é solucionar problemas recorrentes nesses ambientes, como filas demoradas, desorganização nos atendimentos e dificuldade no controle de estoque. Por meio da digitalização do processo de pedidos, espera-se promover mais agilidade, praticidade e satisfação tanto para alunos e professores quanto para os funcionários das cantinas. O sistema foi modelado com base em requisitos previamente definidos, e o banco de dados foi estruturado utilizando a ferramenta **MySQL**, contemplando tabelas específicas para compradores, vendedores, produtos e pedidos, além das relações entre essas entidades.

Palavras-Chave: banco de dados, MySql, cantinaON, produtos, pedidos, compradores, vendedores, CON.

SUMÁRIO

1	INTRODUÇÃO	4
1.1	Objetivos	4
1.2	Justificativa	4
1.3	Aspectos Metodológicos	5
1.4	Aporte Teórico	5
2	METODOLOGIA	6
3	RESULTADOS OBTIDOS	7
3.1	Modelo conceitual	7
3.1.1	Regras de negócio	7
3.1.2	Dicionário de dados	9
3.2	Modelo lógico	10
3.3	Modelo físico	11
3.4	Inserção das informações na tabela	11
3.5	Exemplos de consulta	12
4	CONCLUSÃO	15
	REFERÊNCIAS	16

1 INTRODUÇÃO

Com o crescimento das tecnologias da informação, cada vez mais processos do cotidiano estão sendo automatizados e integrados a plataformas digitais. No ambiente escolar, as cantinas ainda funcionam, na maioria dos casos, de forma manual, o que pode gerar filas, desorganização nos pedidos e dificuldades no controle de estoque e vendas. Diante disso, surgiu a proposta de desenvolvimento de um sistema de pedidos online voltado para cantinas, com o intuito de melhorar a experiência dos usuários e otimizar o funcionamento das cantinas.

1.1 Objetivos

Objetivo Geral:

Desenvolver um sistema de pedidos online para cantinas escolares, com foco na organização dos pedidos, redução de filas e controle eficiente de produtos e vendas.

Objetivos Específicos:

- Realizar a modelagem conceitual e lógica do banco de dados necessário para o sistema.
- Implementar o banco de dados em um SGBD relacional.
- Criar funcionalidades básicas para cadastro de produtos, pedidos, usuários e visualização dos pedidos.
- Garantir a integridade dos dados por meio do uso de chaves primárias e estrangeiras, e normalização das tabelas.
- Produzir consultas SQL que permitam relatórios e análises das vendas.

1.2 Justificativa

A escolha deste tema se justifica pela necessidade real de modernização nos processos de venda em cantinas escolares. Atualmente, o atendimento exclusivamente presencial contribui para a formação de filas e perda de tempo, além de dificultar o controle financeiro e logístico da cantina. A proposta do sistema busca atender a uma demanda prática, contribuindo para a melhoria da experiência dos usuários e oferecendo uma ferramenta acessível para a gestão do negócio.

1.3 Aspectos Metodológicos

A metodologia adotada neste projeto seguiu as seguintes etapas:

1. Levantamento de requisitos do sistema.
2. Elaboração do modelo entidade-relacionamento (MER) para representar as entidades e seus relacionamentos.
3. Conversão do modelo conceitual em modelo lógico e, posteriormente, em modelo físico.
4. Implementação do banco de dados em um SGBD relacional
5. Testes de inserção, consulta e integridade dos dados.
6. Documentação das estruturas criadas, comandos SQL utilizados e resultados obtidos.

O projeto foi desenvolvido com base em práticas recomendadas de modelagem de banco de dados e organização de sistemas de informação.

1.4 Aporte Teórico

O desenvolvimento deste projeto foi fundamentado nos conceitos de banco de dados relacionais, modelagem entidade-relacionamento (ER), normalização de dados e linguagem SQL. Além disso, foram aplicados os princípios da normalização para evitar redundâncias e inconsistências nos dados. A implementação foi realizada em um sistema gerenciador de banco de dados relacional (SGBDR), permitindo garantir a integridade, segurança e eficiência na manipulação das informações.

2. METODOLOGIA

O desenvolvimento do projeto seguiu uma abordagem prática, com foco na criação de um sistema de banco de dados para um serviço de pedidos online voltado para cantinas. A metodologia adotada envolveu etapas de levantamento de requisitos, modelagem do banco de dados e implementação técnica utilizando ferramentas adequadas.

Inicialmente, foi realizada a coleta das regras de negócio com base no funcionamento esperado do sistema. Foram definidos os principais elementos do domínio: compradores (alunos ou usuários que realizam os pedidos, que têm acesso aos produtos apenas da cantina de sua instituição), vendedores (responsável pela cantina cadastrada na aplicação), produtos (comidas e bebidas disponíveis), pedidos (representando o carrinho de compras) e as relações entre esses elementos. A modelagem das entidades e seus relacionamentos considerou as seguintes regras:

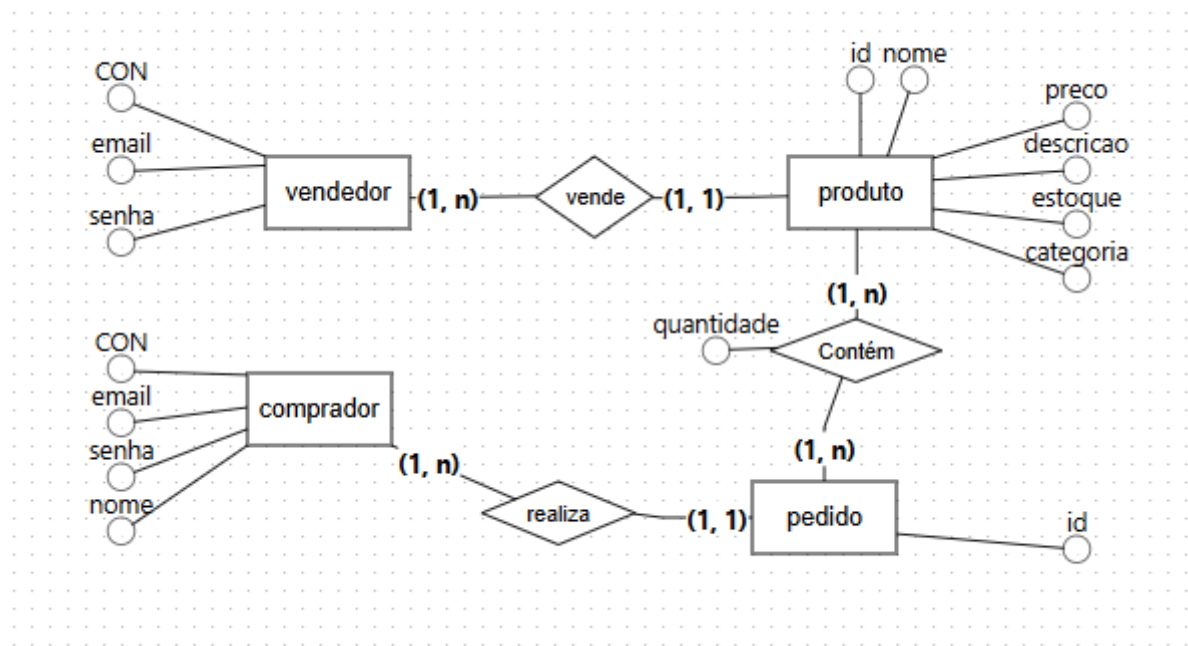
- Cada pedido pertence a um único comprador (relação 1:1).
- Cada pedido pode conter vários produtos e cada produto pode estar em vários pedidos (relação N:N).
- Cada produto é cadastrado por um único vendedor (relação 1:N).

A partir desses requisitos, foi elaborada a modelagem conceitual e lógica do banco de dados, utilizando o MySQL como ferramenta tanto para o desenho do modelo quanto para a implementação do modelo físico. As tabelas foram criadas com base nas entidades definidas, e as chaves primárias e estrangeiras foram aplicadas para garantir a integridade referencial do sistema.

Além disso, foram realizadas consultas SQL para testes de inserção, atualização, exclusão e busca de dados, assegurando o funcionamento correto do banco de dados. A normalização foi aplicada para evitar redundâncias e inconsistências nos dados.

3. RESULTADOS OBTIDOS

3.1 MODELO CONCEITUAL



3.1.1 Regras de negócio

Cadastro de vendedores:

- Os vendedores são os representantes da cantina de cada instituto escolar. O cadastro do vendedor é realizado pela própria escola e funciona da seguinte forma:
 - Um código **único** do **cantinaON**, que é gerado automaticamente pelo site e é composto de 5 caracteres alfanuméricos. Este código é chamado de **código CON** ou apenas **CON** pelo site, para ter uma pronuncia e escrita mais compacta.
 - Um **e-mail único**, que pode ser utilizado pelo site para notificar a escola de informações potencialmente importantes, como atualizações, logins em outros dispositivos, etc.
 - Uma **senha** para aumentar a segurança do usuário.

Cadastro de compradores:

- Os compradores são os alunos, professores e outros funcionários da instituição que desejarem fazer compras na cantina.
- Cada comprador possui:

- Um **código do cantinaON**, que é um código **único** de 9 caracteres alfanuméricos gerado pelo vendedor numa página que só pode ser acessada por ele. Todos os códigos gerados para os compradores possuem os 5 primeiros dígitos idênticos ao código do vendedor, o que permite com que o sistema atrele os compradores apenas para sua respectiva cantina. A distribuição do código para os alunos é de responsabilidade das autoridades da instituição.
- Um **e-mail único** para ser usado para recuperar a senha ou o CON e informar acessos de outros dispositivos.
- Uma **senha de acesso**, para aumentar a segurança.
- O **nome completo**, para identificação do usuário.
- Instituição associada via código CON:
 - O sistema identifica o **instituto escolar** com base nos 5 primeiros caracteres do código (que é igual nos CONs de todos os compradores, e igual ao CON do vendedor).
 - Os compradores só poderão visualizar e fazer pedidos de produtos cadastrados por **vendedores do mesmo instituto**.
- **Pedidos:**
 - Cada **pedido** é realizado por **um único comprador**.
 - Um comprador pode realizar **vários pedidos** ao longo do tempo.
 - Um pedido pode conter **um ou mais produtos** diferentes.
- **Produtos:**
 - Apenas **vendedores (representantes da cantina)** podem cadastrar produtos no sistema.
 - Cada produto pertence a **um único vendedor**, e portanto está associado a **um único instituto**.
 - Cada produto deve conter: **nome, descrição, preço, quantidade em estoque** e uma **categoria** (salgado, doce, lanche ou bebida).
- **Relacionamento entre pedidos e produtos:**
 - Um **pedido** pode conter **vários produtos**.
 - Um **produto** pode estar presente em **vários pedidos**.
 - A quantidade de cada produto no pedido é registrada, e assume **1 como valor padrão**.
- **Estoque:**
 - A quantidade de um produto pedida não pode ultrapassar a quantidade disponível no estoque.
- **Autenticação:**
 - Todos os usuários devem se autenticar com **CON, e-mail e senha válidos** para acessar o sistema.

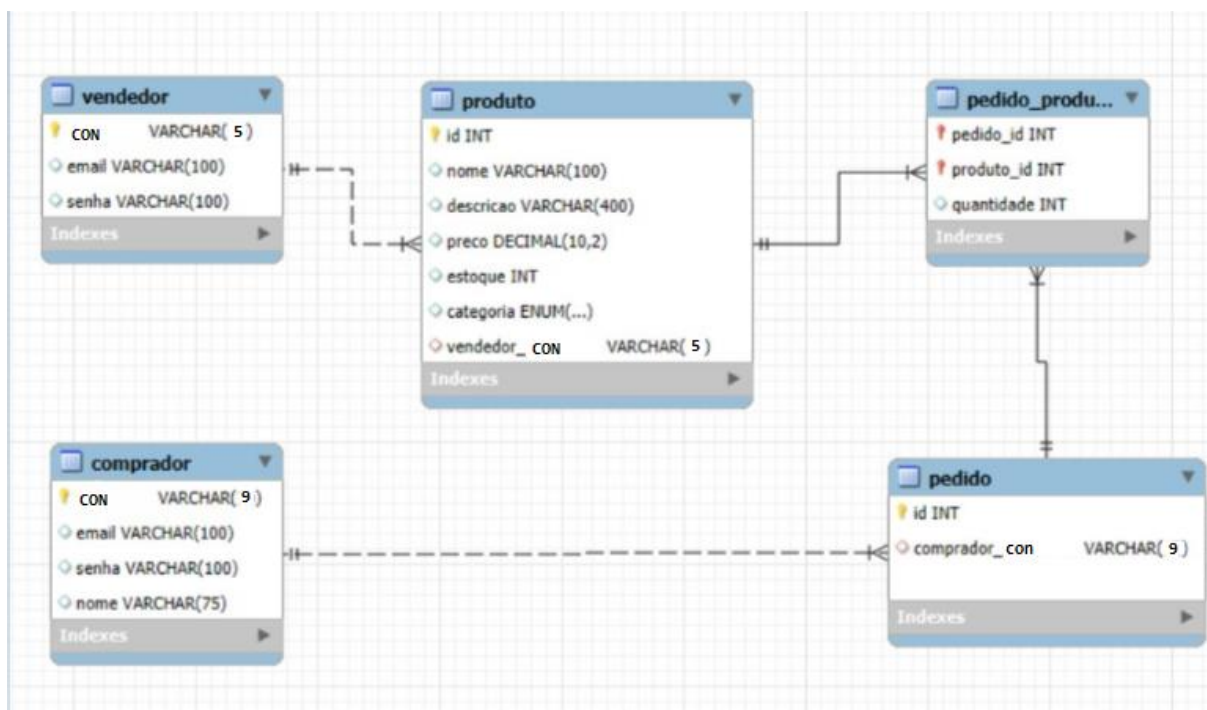
- **Categorias de produtos:**

- Os produtos devem ser classificados em **quatro categorias fixas**: *salgado, doce, lanche e bebida*.

3.1.2 Dicionário de dados

	A	B	C	D	E
1	Tabela	Campo	Tipo de Dado	Restrição	Descrição
2	Comprador	CON	VARCHAR(9)	PK	Prontuário do comprador
3	Comprador	email	VARCHAR(100)	UNIQUE	Email do comprador
4	Comprador	senha	VARCHAR(100)		Senha do comprador
5	Comprador	nome	VARCHAR(75)		Nome completo do comprador
6	Pedido	id	INT	PK, AUTO_INCREMENT	Identificador único do pedido
7	Pedido	comprador_prontuario	VARCHAR(10)	FK → Comprador(prontuario)	Prontuário do comprador que realizou o pedido
8	Vendedor	CON	VARCHAR(5)	PK	Prontuário do vendedor
9	Vendedor	email	VARCHAR(100)	UNIQUE	Email do vendedor
10	Vendedor	senha	VARCHAR(100)		Senha do vendedor
11	Produto	id	INT	PK, AUTO_INCREMENT	Identificador único do produto
12	Produto	nome	VARCHAR(100)		Nome do produto
13	Produto	descricao	VARCHAR(400)		Descrição detalhada do produto
14	Produto	preco	DECIMAL(10,2)		Preço do produto
15	Produto	estoque	INT		Quantidade em estoque do produto
16	Produto	categoria	ENUM('salgado', 'doce', 'lanche', 'bebida')		Categoria do produto
17	Produto	vendedor_prontuario	VARCHAR(10)	FK → Vendedor(prontuario)	Prontuário do vendedor que cadastrou o produto
18	Pedido_Produto	pedido_id	INT	PK, FK → Pedido(id)	Identificador do pedido
19	Pedido_Produto	produto_id	INT	PK, FK → Produto(id)	Identificador do produto no pedido
20	Pedido_Produto	quantidade	INT	DEFAULT 1	Quantidade do produto no pedido

3.2 MODELO LÓGICO



3.3 MODELO FÍSICO

-- Tabela do comprador (O usuário que realiza os pedidos)

```
CREATE TABLE Comprador (
    CON VARCHAR(9) PRIMARY KEY, -- Código do usuário
    email VARCHAR(100) UNIQUE,
    senha VARCHAR(100),
    nome VARCHAR(75)
);
```

-- Tabela do pedido realizado pelo Comprador

```
CREATE TABLE Pedido (
    id INT PRIMARY KEY AUTO_INCREMENT,
    comprador_prontuario VARCHAR(10),
    FOREIGN KEY (comprador_prontuario) REFERENCES
    Comprador(prontuario)
);
```

-- Tabela do vendedor (O usuário que insere os pedidos no sistema)

```
CREATE TABLE Vendedor (
    CON VARCHAR(5) PRIMARY KEY, -- Código do usuário
    email VARCHAR(100) UNIQUE,
    senha VARCHAR(100)
);
```

-- Tabela do produto (Os alimentos a venda) com referência ao vendedor

```
CREATE TABLE Produto (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(100),
    descricao VARCHAR(400),
```

```

    precos DECIMAL(10,2),
    estoque INT,
    categoria ENUM('salgado', 'doce', 'lanche', 'bebida'),
    vendedor_prontuario VARCHAR(10), FOREIGN KEY
    (vendedor_prontuario) REFERENCES Vendedor(prontuario)
);

-- Tabela intermediária para associar pedidos e produtos
CREATE TABLE Pedido_Produto (
    pedido_id INT,
    produto_id INT,
    quantidade INT DEFAULT 1,
    PRIMARY KEY (pedido_id, produto_id),
    FOREIGN KEY (pedido_id) REFERENCES Pedido(id),
    FOREIGN KEY (produto_id) REFERENCES Produto(id)
);

```

3.4 INSERÇÃO DAS INFORMAÇÕES NAS TABELAS

As tabelas serão preenchidas no sistema da seguinte forma:

Comprador:

- As chaves desta entidade são preenchidas durante o cadastro do usuário, seja no aplicativo ou no site. Ela armazena os dados dos compradores, que são os responsáveis por realizar os pedidos.

Pedido:

- A tabela Pedido representa o "carrinho de compras" do comprador. Ela registra quais produtos foram solicitados e qual comprador fez o pedido, sendo essencial para o controle de compras realizadas.

Vendedor:

- Esta tabela é preenchida durante o cadastro do **vendedor** (exceto o CON, que é gerado automaticamente), que é o representante da cantina de uma escola específica. Com base no prontuário do vendedor, o sistema identifica a qual instituição ele pertence.

Produto:

- Esta tabela é preenchida durante o registro de produtos no sistema, feito pelo vendedor. Ela contém informações importantes sobre os alimentos que estão à venda, como nome, descrição, preço, categoria e estoque, permitindo ao comprador conhecer os itens disponíveis.

3.5 EXEMPLOS DE CONSULTAS

1. Selecionar compradores de um instituto específico (ex. CON do vendedor = 1t6J0)

```

SELECT * FROM Comprador
WHERE prontuario LIKE '1t6J0%';

```

2. Listar todos os pedidos de um determinado comprador:

```
SELECT * FROM Pedido
WHERE comprador_prontuario = '1t6J0rK5s';
```

3. Listar os produtos de um pedido específico:

```
SELECT Produto.nome, Produto.preco, Pedido_Produto.quantidade FROM
Pedido_Produto JOIN Produto ON Produto.id = Pedido_Produto.pro-
duto_id WHERE Pedido_Produto.pedido_id = 5;
```

4. Listar todos os compradores cadastrados ordenados por nome:

```
SELECT * FROM Comprador
ORDER BY nome;
```

5. Mostrar todos os produtos da categoria “salgado”:

```
SELECT * FROM Produto
WHERE categoria = 'salgado';
```

6. Listar os produtos com preço acima de R\$10,00:

```
SELECT * FROM Produto
WHERE preco > 10.00;
```

7. Listar produtos com estoque abaixo de 5 unidades:

```
SELECT * FROM Produto
WHERE estoque < 5;
```

8. Mostrar a quantidade total de pedidos realizados no sistema:

```
SELECT COUNT(*) AS total_pedidos
FROM Pedido;
```

9. Mostrar os produtos cadastrados por um vendedor específico:

```
SELECT * FROM Produto
WHERE vendedor_CON = '1t6J0';
```

10. Listar todos os pedidos contendo um determinado produto:

```
SELECT pedido_id
FROM Pedido_Produto
WHERE produto_id = 2;
```

11. Mostrar a soma total de itens em um pedido específico:

```
SELECT SUM(quantidade) AS total_itens
FROM Pedido_Produto
WHERE pedido_id = 7;
```

12. Calcular o valor total de um pedido (soma dos preços × quantidades):

```
SELECT SUM(p.preco * pp.quantidade) AS total_pedido
FROM Pedido_Produto pp
JOIN Produto p ON pp.produto_id = p.id
WHERE pp.pedido_id = 3;
```

13. Listar os nomes e preços de todos os produtos em estoque (estoque > 0):

```
SELECT nome, preco
FROM Produto
WHERE estoque > 0;
```

14. Atualizar o estoque de um produto após um pedido (exemplo: diminuir 3 unidades):

```
UPDATE Produto
SET estoque = estoque - 3
WHERE id = 4;
```

15. Buscar comprador por e-mail:

```
SELECT * FROM Comprador
WHERE email = 'exemplo@email.com';
```

16. Listar o número de produtos por categoria:

```
SELECT categoria, COUNT(*) AS total
FROM Produto
GROUP BY categoria;
```

17. Verificar o CON de um aluno:

```
SELECT CON AS código_CON
FROM Comprador
WHERE email = 'aluno@if.com' AND senha = 'senha123';
```

18. Listar os pedidos e os nomes dos compradores:

```
SELECT p.id AS pedido_id, c.nome
FROM Pedido p
```

```
JOIN Comprador c ON p.comprador_CON = c.prontuario;
```

19. Contar quantos produtos cada vendedor cadastrou:

```
SELECT vendedor_CON, COUNT(*) AS total_produtos FROM Produto GROUP  
BY vendedor_CON;
```

20. Mostrar todos os produtos da categoria “salgado”:

```
SELECT * FROM Produto  
WHERE categoria = 'salgado';
```

4. CONCLUSÃO

A partir do desenvolvimento de um banco de dados, este projeto pôde ser concluído com sucesso, cumprindo sua proposta de auxiliar cantinas escolares. Como sugestões de melhorias para o futuro, destaca-se a possibilidade de inserir um campo na tabela de produtos para o armazenamento de imagens, permitindo que uma foto do item seja exibida ao ser cadastrado. Além disso, seria interessante implementar um sistema de pagamentos integrado, o que exigiria a criação de uma tabela específica para armazenar informações como o status do pagamento, horário da transação, entre outros dados relevantes.

REFERÊNCIAS

CORONEL, Carlos; MORRIS, Steven. **Database Systems, Design, implementation, & Management**. Boston: Cengage Learning, 2019.

MySQL Documentation: <https://dev.mysql.com/doc/>

Stack Overflow: <https://stackoverflow.com/>