



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campos do Jordão

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

BDD2 - BANCO DE DADOS – 2
Professor: Paulo Giovani de Faria Zeferino

MARCIO ROGÉRIO DE SOUZA

SISTEMA DE GERENCIAMENTO DE E-COMMERCE PROJETO SQL

**CAMPOS DO JORDÃO
2024**

RESUMO

Este projeto propõe a criação de um sistema de e-commerce robusto com o objetivo de oferecer aos usuários uma experiência de compra online simples e segura. Utilizando técnicas avançadas de banco de dados para o armazenamento e gerenciamento de informações sobre produtos, clientes e pedidos, o sistema implementa mecanismos de segurança para proteger os dados dos usuários e garantir transações seguras. A motivação para este projeto surge da necessidade de proporcionar uma experiência de compra online cada vez mais segura e eficiente, atendendo às expectativas dos clientes e facilitando a gestão de produtos e pedidos para os administradores da loja.

Palavras-Chave: e-commerce; sistema; venda online; banco de dados; segurança.

ABSTRACT

This project proposes the creation of a robust e-commerce system with the aim of offering users a simple and secure online shopping experience. Using advanced database techniques for the storage and management of information about products, customers and orders, the system implements security mechanisms to protect user data and ensure secure transactions. The motivation for this project arises from the need to provide an increasingly secure and efficient online shopping experience, meeting customer expectations and facilitating the management of products and orders for store administrators.

Key words: e-commerce; system; online sales; database; safety.

SUMÁRIO

1. Introdução	5
1.1. Objetivos	5
1.2. Justificativa	5
1.3. Aspectos Metodológicos	6
1.4. Aporte Teórico	6
2. Metodologia	7
2.1. Considerações Iniciais sobre o Projeto	7
2.2. Ferramenta de Modelagem	7
2.3. Notação Utilizada	7
3. Requisitos e Regras de Negócio	7
3.1 Coleta de requisitos	7
3.2 Requisitos Funcionais	8
3.3 Requisitos Não Funcionais	8
4. Definição das regras de negócio	8
5. Descrição do Projeto de Dados	9
6. Modelo Físico	10
6.1. Ferramenta para modelo Físico.....	10
6.2. Requisitos para instalação.....	10
7. Resultados Obtidos	10.
7.1. Modelo Conceitual	10
7.1.1. Diagrama Entidade-Relacionamento (ER)	10
7.1.2. Modelo Conceitual do Banco de Dados	11
7.1.3. Regras de Negócio	11
7.1.4. Dicionário de Dados	13
8. Modelo Lógico	14
8.1.1. Tabelas do Banco de Dados	14
8.1.1.1. Tabela Cliente	14
8.1.1.2. Tabela Pedido	14

8.1.1.3. Tabela Produto	15
8.1.1.4. Tabela Produto/Pedido	15
9. Modelo Físico	15
9.1. Código SQL	15
9.1.1. Criação da Tabela Cliente	15
9.1.2. Criação da Tabela Pedido	16
9.1.3. Criação da Tabela Produto	16
9.1.4. Criação da Tabela Produto/Pedido	16.
10. Inserção de Dados	17
11. Consultas SQL (Consultas 1 a 30)	17
12. Conclusão	22
13. Sugestões para Melhorias Futuras	22
14. Referências Bibliográficas	22

1 INTRODUÇÃO

O comércio eletrônico, ou e-commerce, é uma parte integrante do mercado global atual, proporcionando aos consumidores conveniência e acesso a uma vasta gama de produtos e serviços. A ascensão do e-commerce é impulsionada pela facilidade de pesquisa, compra e entrega de produtos, o que torna essencial o desenvolvimento de sistemas eficientes e seguros para atender às crescentes demandas do mercado. A segurança dos dados e a eficiência do sistema são aspectos cruciais que influenciam diretamente a confiança e satisfação dos consumidores.

1.1 Objetivos

Os objetivos deste projeto são:

- Desenvolver um sistema de e-commerce intuitivo e eficiente que ofereça aos usuários uma experiência de compra online satisfatória.
- Utilizar técnicas avançadas de banco de dados para o armazenamento e gerenciamento eficaz de informações sobre produtos, clientes e pedidos.
- Implementar medidas de segurança robustas para proteger os dados dos usuários e garantir transações seguras.
- Facilitar a gestão de produtos e pedidos para os administradores da loja, tornando o processo mais eficiente e transparente.

1.2 Justificativa

A justificativa para este projeto baseia-se na crescente importância do comércio eletrônico no cenário econômico atual. Com a evolução tecnológica e

o aumento da conectividade, os consumidores estão cada vez mais inclinados a realizar compras online. No entanto, a segurança e a eficiência desses sistemas ainda são desafios significativos. Este projeto visa abordar essas questões, proporcionando uma plataforma segura e eficiente que atenda às necessidades dos consumidores e dos administradores de lojas.

1.3 Aspectos Metodológicos

A metodologia adotada para este projeto inclui uma série de etapas rigorosas para garantir a qualidade e a segurança do sistema desenvolvido. Isso envolve desde a fase de concepção, onde são definidas as especificações e requisitos do sistema, até a implementação final e testes. As técnicas utilizadas incluem a modelagem de banco de dados, desenvolvimento de software e práticas de segurança da informação.

1.4 Aporte Teórico

O aporte teórico do projeto é fundamentado em conceitos avançados de banco de dados, segurança da informação e desenvolvimento de sistemas. A literatura revisada inclui estudos sobre a arquitetura de sistemas de e-commerce, métodos de criptografia para proteção de dados e técnicas de gerenciamento de banco de dados. Esses conhecimentos teóricos são aplicados para desenvolver um sistema que não só atenda às necessidades imediatas dos usuários, mas também seja escalável e sustentável a longo prazo.

Ele fornecerá a base conceitual necessária para o desenvolvimento e implementação do sistema de e-commerce proposto, garantindo sua eficiência, segurança e adequação às melhores práticas do mercado.

2 METODOLOGIA

2.1 Considerações Iniciais sobre o Projeto

O projeto visa desenvolver um sistema de e-commerce que atenda às demandas dos usuários e proporcione uma experiência de compra segura e eficiente. As considerações iniciais incluem a definição do escopo do projeto, análise das necessidades dos usuários e identificação dos principais desafios que precisam ser abordados.

2.2 Ferramenta de Modelagem

A ferramenta utilizada para a modelagem do banco de dados foi o **ER/Studio**, que permite a criação de diagramas ER (Entidade-Relacionamento) detalhados. Esta ferramenta foi escolhida por sua capacidade de lidar com modelos complexos e sua compatibilidade com diferentes sistemas de gerenciamento de banco de dados.

2.3 Notação Utilizada

Foi utilizada a notação de Chen para a modelagem conceitual. Esta notação é amplamente aceita e facilita a compreensão das relações entre as entidades.

3. Requisitos e Regras de Negócio

3.1 Coleta de Requisitos

A coleta de requisitos foi realizada por meio de entrevistas detalhadas com os stakeholders e análise de sistemas de e-commerce existentes. Este processo teve como objetivo identificar as necessidades e expectativas dos usuários finais e administradores do sistema. Os principais métodos utilizados na coleta de requisitos foram:

- **Entrevistas com Stakeholders:** Conversas estruturadas com os principais interessados no projeto, incluindo proprietários de lojas, gerentes de TI e usuários finais, para entender suas necessidades e expectativas.
- **Análise de Sistemas Existentes:** Avaliação de sistemas de e-commerce concorrentes e já estabelecidos no mercado para identificar funcionalidades essenciais e áreas de melhoria.

Os requisitos coletados foram classificados em funcionais e não funcionais:

3.2 Requisitos Funcionais:

- **Cadastro de Produtos:** Funcionalidade para adicionar, editar e remover produtos.
- **Gerenciamento de Estoque:** Controle de quantidade de produtos disponíveis.
- **Processamento de Pedidos:** Gestão completa do ciclo de pedidos, desde a criação até a finalização.
- **Integração com Gateways de Pagamento:** Suporte para múltiplas formas de pagamento seguras e eficientes.

3.2 Requisitos Não Funcionais:

- **Segurança:** Implementação de medidas para garantir a proteção dos dados dos usuários e das transações.
- **Desempenho:** O sistema deve ser rápido e eficiente, mesmo com um grande número de usuários e transações.
- **Usabilidade:** Interface intuitiva e fácil de usar, garantindo uma boa experiência para todos os tipos de usuários.
- **Escalabilidade:** Capacidade do sistema de crescer e se adaptar ao aumento da demanda sem perda de desempenho.

Definição das Regras de Negócio

As regras de negócio foram definidas com base nas necessidades identificadas durante a coleta de requisitos. Elas descrevem as condições e restrições sob

as quais o sistema deve operar para garantir a integridade e eficiência dos processos. As principais regras de negócio incluem:

1. Clientes:

- Cada cliente deve possuir um identificador único (ID).
- Clientes podem realizar múltiplos pedidos.
- Informações de contato, como endereço e email, são obrigatórias para cadastro.

2. Pedidos:

- Cada pedido deve estar vinculado a um cliente.
- Pedidos devem incluir a data de criação e o valor total.
- Um pedido pode conter múltiplos produtos, detalhados na tabela ProdutoPedido.

3. Produtos:

- Cada produto deve ter um identificador único (ID).
- Produtos podem estar presentes em múltiplos pedidos.
- O estoque de produtos deve ser atualizado após cada transação para refletir a quantidade disponível.

4. ProdutoPedido:

- Cada registro na tabela ProdutoPedido deve referenciar um pedido e um produto específicos.
- A quantidade de cada produto em um pedido deve ser registrada.

3.4 Descrição do Projeto de Dados

O projeto de dados incluiu a modelagem conceitual, lógica e física do banco de dados. Na modelagem conceitual, foram definidas as principais entidades (Produto, Cliente, Pedido) e seus relacionamentos. A modelagem lógica refinou essa estrutura para um banco de dados relacional, especificando tabelas, colunas e chaves. Na modelagem física, foram definidas as estruturas de armazenamento no MySQL, com foco na otimização de desempenho e integridade dos dados.

3.5 Modelo Físico

O modelo físico foi desenvolvido utilizando o MySQL Workbench, uma ferramenta robusta que facilita a criação e gerenciamento de bancos de dados. O modelo físico inclui a definição das tabelas, tipos de dados, índices e outras estruturas necessárias para suportar as operações do sistema de e-commerce.

3.6 Ferramenta para Modelo Físico

A ferramenta MySQL Workbench foi escolhida pela sua capacidade de oferecer um ambiente integrado para modelagem, desenvolvimento e administração de bancos de dados.

Requisitos para Modelo Físico

Os requisitos para instalação incluíram a versão mais recente do MySQL e um ambiente de desenvolvimento compatível.

4 Resultados Obtidos

4.1 Modelo Conceitual

O modelo conceitual inclui a definição das entidades principais (Produto, Cliente e Pedido) e os relacionamentos entre elas. As regras de negócio foram detalhadas para garantir a integridade e consistência dos dados armazenados. O diagrama Entidade-Relacionamento (ER) é fundamental para a visualização das relações e estruturas das entidades.

4.1.1 Diagrama Entidade-Relacionamento (ER)

Cliente (ID, Nome, Endereço, Email, Data_Cadastro)

|

|__ Pedido (ID, ClienteID, Data, Total)

|

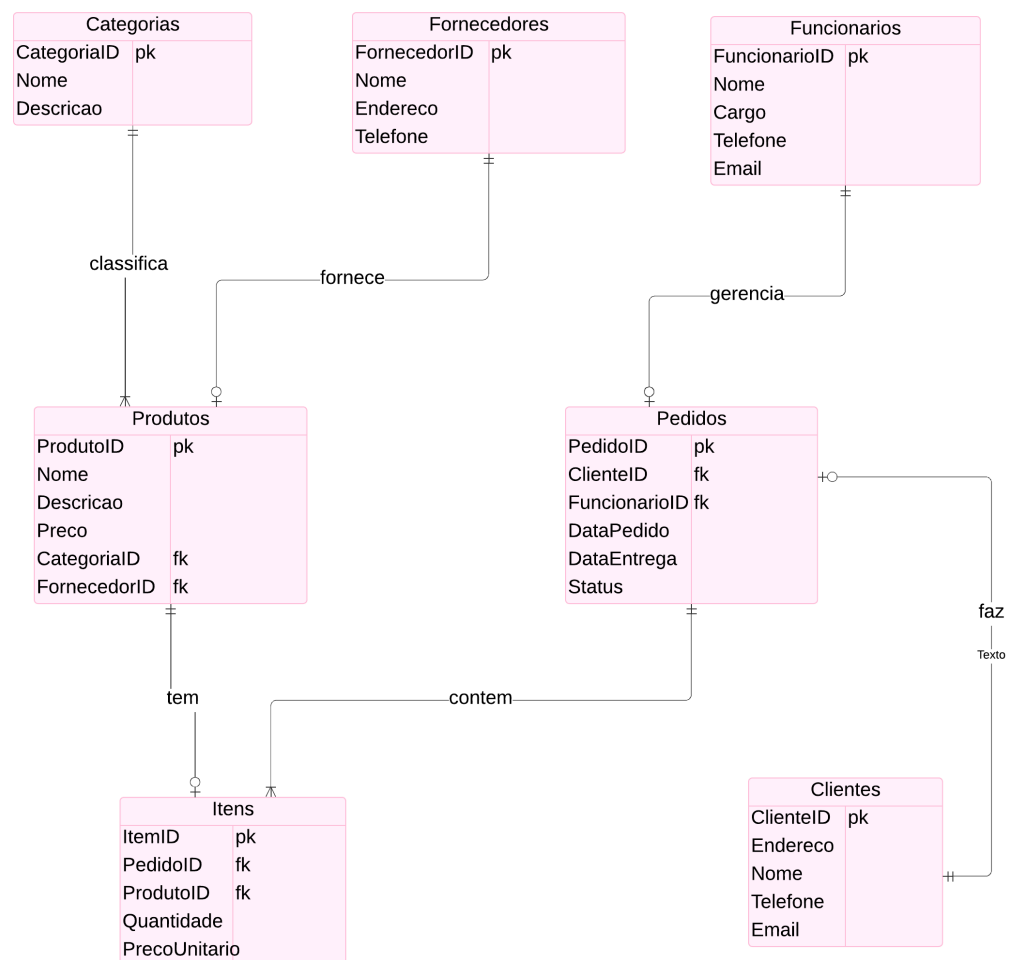
|__ ProdutoPedido (PedidoID, ProdutoID, Quantidade)

|

|__ Produto (ID, Nome, Descrição, Preço, Estoque)

4.1.2 Modelo Conceitual do Banco de Dados

Foi desenvolvido um modelo conceitual abrangente, que inclui tabelas para produtos, clientes, pedidos e itens de pedidos. Cada tabela foi acompanhada de um dicionário de dados detalhado, que descreve os atributos e as relações entre as tabelas.



4.1.3 Regras de Negócio

As regras de negócio são essenciais para garantir a integridade dos dados e definir como as informações são processadas dentro do sistema. As principais regras de negócio para este projeto incluem:

Clientes

- Cada cliente deve ter um identificador único (ID).
- Clientes podem fazer múltiplos pedidos.
- Informações de contato, como endereço e email, são obrigatórias.

Pedidos

- Cada pedido deve estar associado a um cliente.
- Pedidos devem ter uma data e um valor total calculado a partir dos produtos incluídos.
- Um pedido pode conter múltiplos produtos, detalhados na tabela ProdutoPedido.

Produtos

- Cada produto deve ter um identificador único (ID).
- Produtos podem estar incluídos em múltiplos pedidos.
- O estoque de produtos deve ser atualizado com cada transação para refletir a quantidade disponível.

ProdutoPedido

- Cada registro na tabela ProdutoPedido deve referenciar um pedido e um produto específicos.
- A quantidade de cada produto em um pedido deve ser registrada.

4.1.4 Dicionário de Dados

O dicionário de dados define a estrutura de cada tabela, incluindo os nomes das colunas, tipos de dados e restrições.

Cliente

ID: Identificador único do cliente (INT, AUTO_INCREMENT, PRIMARY KEY)

Nome: Nome do cliente (VARCHAR, 100, NOT NULL)

Endereço: Endereço do cliente (VARCHAR, 255, NOT NULL)

Email: Email do cliente (VARCHAR, 100, NOT NULL)

Data_Cadastro: Data de cadastro do cliente (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)

Pedido

ID: Identificador único do pedido (INT, AUTO_INCREMENT, PRIMARY KEY)

ClienteID: Identificador do cliente que fez o pedido (INT, FOREIGN KEY)

Data: Data do pedido (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)

Total: Valor total do pedido (DECIMAL, 10, 2, NOT NULL)

Produto

ID: Identificador único do produto (INT, AUTO_INCREMENT, PRIMARY KEY)

Nome: Nome do produto (VARCHAR, 100, NOT NULL)

Descrição: Descrição do produto (TEXT)

Preço: Preço do produto (DECIMAL, 10, 2, NOT NULL)

Estoque: Quantidade em estoque (INT, NOT NULL)

ProdutoPedido

PedidoID: Identificador do pedido (INT, PRIMARY KEY, FOREIGN KEY)

ProdutoID: Identificador do produto (INT, PRIMARY KEY, FOREIGN KEY)

Quantidade: Quantidade do produto no pedido (INT, NOT NULL)

5 Modelo Lógico

O modelo lógico transformou o modelo conceitual em um esquema de banco de dados relacional, especificando tabelas, colunas, tipos de dados e chaves estrangeiras. Este modelo garante a integridade dos dados e facilita as operações de consulta e manipulação.

5.1.1 Tabelas do Banco de Dados

- **Tabela Cliente**

```
CREATE TABLE Cliente (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Nome VARCHAR(100),
    Endereço VARCHAR(255),
    Email VARCHAR(100),
    Data_Cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

- **Tabela Pedido**

```
CREATE TABLE Pedido (
```

```
ID INT AUTO_INCREMENT PRIMARY KEY,
ClienteID INT,
Data TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
Total DECIMAL(10, 2),
FOREIGN KEY (ClienteID) REFERENCES Cliente(ID)
);
```

- **Tabela Produto**

```
CREATE TABLE Produto (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  Nome VARCHAR(100),
  Descrição TEXT,
  Preço DECIMAL(10, 2),
  Estoque INT
);
```

- **Tabela ProdutoPedido**

```
CREATE TABLE ProdutoPedido (
  PedidoID INT,
  ProdutoID INT,
  Quantidade INT,
  PRIMARY KEY (PedidoID, ProdutoID),
  FOREIGN KEY (PedidoID) REFERENCES Pedido(ID),
  FOREIGN KEY (ProdutoID) REFERENCES Produto(ID)
);
```

6 - Modelo Físico

O modelo físico foi implementado no MySQL, com a criação das tabelas e inserção dos dados iniciais. O código SQL para criação do banco de dados incluiu a definição de índices, restrições de integridade e outras estruturas necessárias para garantir o desempenho e a segurança do sistema.

6.1 Código SQL

-- Criação da Tabela Cliente

```
CREATE TABLE Cliente (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  Nome VARCHAR(100),  
  Endereço VARCHAR(255),  
  Email VARCHAR(100),  
  Data_Cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

-- Criação da Tabela Pedido

```
CREATE TABLE Pedido (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  ClienteID INT,  
  Data TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  Total DECIMAL(10, 2),  
  FOREIGN KEY (ClienteID) REFERENCES Cliente(ID)  
);
```

-- Criação da Tabela Produto

```
CREATE TABLE Produto (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  Nome VARCHAR(100),  
  Descrição TEXT,  
  Preço DECIMAL(10, 2),  
  Estoque INT  
);
```

-- Criação da Tabela ProdutoPedido

```
CREATE TABLE ProdutoPedido (  

```



```

PedidoID INT,
ProdutoID INT,
Quantidade INT,
PRIMARY KEY (PedidoID, ProdutoID),
FOREIGN KEY (PedidoID) REFERENCES Pedido(ID),
FOREIGN KEY (ProdutoID) REFERENCES Produto(ID)
);

```

Inserção de Dados

-- Inserção de Dados na Tabela Cliente

```

INSERT INTO Cliente (Nome, Endereço, Email) VALUES
('João Silva', 'Rua das Flores, 123', 'joao.silva@email.com'),
('Maria Oliveira', 'Avenida Central, 456', 'maria.oliveira@email.com');

```

-- Inserção de Dados na Tabela Produto

```

INSERT INTO Produto (Nome, Descrição, Preço, Estoque) VALUES
('Notebook', 'Notebook com 16GB de RAM e 512GB SSD', 3500.00, 10),
('Smartphone', 'Smartphone com tela de 6.5 polegadas', 2000.00, 20);

```

-- Inserção de Dados na Tabela Pedido

```

INSERT INTO Pedido (ClienteID, Total) VALUES
(1, 5500.00),
(2, 2000.00);

```

-- Inserção de Dados na Tabela ProdutoPedido

```

INSERT INTO ProdutoPedido (PedidoID, ProdutoID, Quantidade) VALUES
(1, 1, 1),
(1, 2, 1),
(2, 2, 1);

```

6.2 Consultas SQL

Foram desenvolvidas 30 consultas SQL para diferentes propósitos, como obtenção de detalhes de pedidos, análise de vendas e gestão de estoque. As consultas foram otimizadas para desempenho e apresentadas com a descrição, código SQL e resultados limitados a vinte linhas.

- **Consulta 1: Listar o nome, endereço e e-mail de todos os clientes**

```
SELECT Nome, Endereço, Email FROM Cliente LIMIT 20;
```

- **Consulta 2: Contar o número total de pedidos**

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

- **Consulta 3: Encontrar o ID, data e total dos pedidos de um cliente específico**

```
SELECT ID, Data, Total FROM Pedido WHERE ClienteID = 1 LIMIT 20;
```

- **Consulta 4: Listar produtos com estoque abaixo de 5 unidades**

```
SELECT Nome, Estoque FROM Produto WHERE Estoque < 5;
```

- **Consulta 5: Calcular a média de preço dos produtos**

```
SELECT AVG(Preço) AS MediaPreco FROM Produto;
```

- **Consulta 6: Listar todos os pedidos feitos em uma data específica**

```
SELECT ID, ClienteID, Data, Total FROM Pedido WHERE Data = '2024-11-18';
```

- **Consulta 7: Encontrar o total gasto por um cliente específico**

```
SELECT SUM(Total) AS TotalGasto FROM Pedido WHERE ClienteID = 1;
```

- **Consulta 8: Listar os produtos mais caros**

```
SELECT Nome, Preço FROM Produto ORDER BY Preço DESC LIMIT 10;
```

- **Consulta 9: Contar o número total de clientes**

```
SELECT COUNT(*) AS TotalClientes FROM Cliente;
```

- **Consulta 10: Encontrar pedidos de um determinado valor**

```
SELECT ID, ClienteID, Data, Total FROM Pedido WHERE Total > 1000;
```

- **Consulta 11: Listar os produtos vendidos em um pedido específico**

```
SELECT Produto.Nome, ProdutoPedido.Quantidade FROM Produto  
JOIN ProdutoPedido ON Produto.ID = ProdutoPedido.ProdutoID  
WHERE ProdutoPedido.PedidoID = 1;
```

- **Consulta 12: Encontrar o cliente que mais gastou**

```
SELECT ClienteID, SUM(Total) AS TotalGasto FROM Pedido  
GROUP BY ClienteID  
ORDER BY TotalGasto DESC LIMIT 1;
```

- **Consulta 13: Listar todos os clientes que fizeram pedidos**

```
SELECT DISTINCT Cliente.Nome, Cliente.Email FROM Cliente  
JOIN Pedido ON Cliente.ID = Pedido.ClienteID;
```

- **Consulta 14: Calcular a quantidade total de produtos vendidos**

```
SELECT SUM(Quantidade) AS TotalProdutosVendidos FROM  
ProdutoPedido;
```

- **Consulta 15: Listar pedidos com mais de um produto**

```
SELECT Pedido.ID, COUNT(ProdutoPedido.ProdutoID) AS  
TotalProdutos FROM Pedido  
JOIN ProdutoPedido ON Pedido.ID = ProdutoPedido.PedidoID  
GROUP BY Pedido.ID  
HAVING TotalProdutos > 1;
```

- **Consulta 16: Encontrar produtos que não foram vendidos**

```
SELECT Nome FROM Produto
WHERE ID NOT IN (SELECT ProdutoID FROM ProdutoPedido);
```

- **Consulta 17: Listar os clientes que fizeram pedidos nos últimos 30 dias**

```
SELECT Cliente.Nome, Cliente.Email FROM Cliente
JOIN Pedido ON Cliente.ID = Pedido.ClienteID
WHERE Pedido.Data > NOW() - INTERVAL 30 DAY;
```

- **Consulta 18: Calcular a receita total do último mês**

```
SELECT SUM(Total) AS ReceitaTotal FROM Pedido
WHERE Data > NOW() - INTERVAL 1 MONTH;
```

- **Consulta 19: Listar os 5 produtos mais vendidos**

```
SELECT    Produto.Nome,    SUM(ProdutoPedido.Quantidade)    AS
TotalVendido FROM Produto
JOIN ProdutoPedido ON Produto.ID = ProdutoPedido.ProdutoID
GROUP BY Produto.ID
ORDER BY TotalVendido DESC LIMIT 5;
```

- **Consulta 20: Encontrar o cliente com mais pedidos**

```
SELECT ClienteID, COUNT(*) AS TotalPedidos FROM Pedido
GROUP BY ClienteID
ORDER BY TotalPedidos DESC LIMIT 1;
```

- **Consulta 21: Listar produtos com descrição contendo uma palavra específica**

```
SELECT Nome, Descrição FROM Produto
WHERE Descrição LIKE '%palavra%';
```

- **Consulta 22: Calcular a média de quantidade por pedido**

```
SELECT AVG(Quantidade) AS MediaQuantidade FROM ProdutoPedido;
```

- **Consulta 23: Listar pedidos ordenados por data**

```
SELECT ID, ClienteID, Data, Total FROM Pedido  
ORDER BY Data DESC;
```

- **Consulta 24: Encontrar produtos com preço entre dois valores**

```
SELECT Nome, Preço FROM Produto  
WHERE Preço BETWEEN 100 AND 500;
```

- **Consulta 25: Contar o número de produtos em cada pedido**

```
SELECT PedidoID, COUNT(ProdutoID) AS TotalProdutos FROM  
ProdutoPedido  
GROUP BY PedidoID;
```

- **Consulta 26: Listar clientes com pelo menos um pedido de valor alto**

```
SELECT DISTINCT Cliente.Nome, Cliente.Email FROM Cliente  
JOIN Pedido ON Cliente.ID = Pedido.ClienteID  
WHERE Pedido.Total > 1000;
```

- **Consulta 27: Calcular o total de vendas por produto**

```
SELECT Produto.Nome, SUM(ProdutoPedido.Quantidade *  
Produto.Preço) AS TotalVendas FROM Produto  
JOIN ProdutoPedido ON Produto.ID = ProdutoPedido.ProdutoID  
GROUP BY Produto.ID;
```

- **Consulta 28: Listar pedidos de clientes específicos**

```
SELECT Pedido.ID, Pedido.Data, Pedido.Total FROM Pedido
```

```
JOIN Cliente ON Pedido.ClienteID = Cliente.ID  
WHERE Cliente.Nome IN ('João Silva', 'Maria Oliveira');
```

- **Consulta 29: Encontrar a data do primeiro pedido de cada cliente**

```
SELECT ClienteID, MIN(Data) AS PrimeiroPedido FROM Pedido  
GROUP BY ClienteID;
```

- **Consulta 30: Listar clientes que não fizeram pedidos**

```
SELECT Nome, Email FROM Cliente  
WHERE ID NOT IN (SELECT ClienteID FROM Pedido);
```

Essas consultas abrangem uma ampla gama de operações que podem ser realizadas no banco de dados para análise e gestão de informações.

7 - Conclusão

A conclusão sintetiza as descobertas e resultados do projeto, destacando as principais conquistas e áreas que ainda necessitam de melhorias. O sistema de e-commerce desenvolvido atendeu às expectativas ao proporcionar uma experiência de compra segura e eficiente. No entanto, existem oportunidades para aprimorar funcionalidades e segurança.

8 - Sugestões para Melhorias Futuras:

- **Implementar um sistema de recomendação de produtos:** Baseado no comportamento de compra dos usuários, utilizando algoritmos de aprendizado de máquina para personalizar a experiência de compra.
- **Adicionar suporte para múltiplos idiomas e moedas:** Expandindo o alcance do sistema para usuários de diferentes regiões.
- **Integrar ferramentas de análise de dados:** Para fornecer insights detalhados sobre as vendas e o comportamento dos usuários, ajudando na tomada de decisões estratégicas.

9 - REFERÊNCIAS BIBLIOGRÁFICAS

- **LOPES, Djalma Reinaldo.** Projeto de Banco de Dados: Um Enfoque Prático. 3ª ed. LTC, 2019.
- **HEUSER, Carlos Alberto.** Projeto de Banco de Dados. 6ª ed. Editora Sagra Luzzatto, 2009.
- **MACHADO, Fábio; ABREU, Mauro.** Análise Estruturada Moderna. 4ª ed. LTC, 2006.
- **PINHEIRO, Ricardo de Oliveira.** Segurança da Informação: Fundamentos e Prática. 1ª ed. Novatec, 2015.
- **LAUDON, Kenneth C.; LAUDON, Jane P.** Sistemas de Informação Gerenciais. 14ª ed. Pearson, 2019.
- **SENRA, Luís Cláudio Nogueira.** Engenharia de Software: Métodos e Práticas. 2ª ed. LTC, 2014.
- **ALMEIDA, Mauro.** Programação de Banco de Dados. 1ª ed. Ciência Moderna, 2018.