



MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

IF Sul-Rio-Grandense - Campus Passo Fundo

Bacharelado em Ciência da Computação

TRABALHO FINAL BANCO DE DADOS I

Trabalho apresentado ao curso Bacharel em Ciência da Computação do Instituto Federal Sul-Rio-Grandense Campus Passo Fundo, como parte dos requisitos necessários à aprovação na disciplina de Banco de Dados I.

Helen Zanco Neis

Marco Antônio Felipi

Marjory Ane Toazza

Passo Fundo, 17/08/2024.

## SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>2</b>
<b>2. DOMÍNIO</b>	<b>3</b>
2.2. Regras de Negócio e Integridade	3
2.3. Modelo e tipos de Dados utilizado	3
<b>3. REQUISITOS E FUNCIONALIDADES</b>	<b>4</b>
3.1. Planejamento do Levantamento de Requisitos	4
3.2. Coleta de Informações	4
3.3. Documentação dos Requisitos	4
3.4. Definição das Funcionalidades Principais	5
<b>4. DESENVOLVIMENTO</b>	<b>6</b>
4.1 MODELO CONCEITUAL	6
4.2 MODELO LÓGICO	6
4.3 MODELO FÍSICO	8
<b>5. CONCLUSÃO</b>	<b>12</b>

## **1. INTRODUÇÃO**

Este trabalho tem como objetivo explorar os conceitos fundamentais dos bancos de dados, incluindo suas estruturas, modelos e tecnologias. Serão temas envolvidos como o modelo conceitual, modelo lógico e físico no PostgreSQL com a extensão pgadmin 4. Este trabalho analisa como esses componentes são utilizados para criar e gerenciar um sistema de vendas, oferecendo uma compreensão prática da implementação de um sistema de banco de dados desde o planejamento inicial até a execução

## 2. DOMÍNIO

O domínio escolhido pelo nosso grupo foi um sistema de venda de materiais, similar ao da empresa JR materiais de construção. O nosso sistema é genérico e pode se encaixar facilmente em qualquer contexto de vendas. O propósito Geral é o gerenciamento do fluxo de informações principalmente vendas de produtos na loja, clientes, vendedores, inventário de produtos, pedidos, fluxo de giro e caixa. O contexto de uso é criar um sistema de gestão empresarial, alimentando o banco de dados e conservando informações cruciais dentro da empresa, melhorando a gestão e facilitando o armazenamento de dados da empresa, evitando erros e produzindo resultados.

### 2.1. Entidades e Relacionamentos

- **Entidades e atributos:**

Cliente - cod\_cliente, nome, email, endereco numero - rua, CPF CNPJ, telefone

Vendedor - cod\_vendedor, nome, email, comissao, aux\_vendedor

Produto - cod\_produto, desconto, quantidade, nome\_tipo, preco, descricao

Categoria - cod\_categoria, tipo\_categoria

Pedido - cod\_pedido, valor\_total, forma\_pagamento, data\_pedido

Caixa - cod\_caixa, forma\_pagamento, total\_caixa, entrega

- **Relacionamentos:** Cliente atendido pelo vendedor, Cliente irá pagar no caixa. O Vendedor vai requisitar um Produto. O Vendedor passará informações ao caixa. O vendedor terá um auxiliar. O Produto compõem o Pedido. O Produto pertence a uma Categoria. O pedido será direcionado ao caixa.

### 2.2. Regras de Negócio e Integridade

- **Regras de Integridade:** Utilizamos chaves estrangeiras e integridade de domínio, códigos identificadores.
- **Restrições:** As informações que o vendedor deve passar ao caixa são: cod\_pedido, forma de pagamento parcelas, cod\_vendedor.

### 2.3. Modelo e tipos de Dados utilizado

- **Tipos de Dados:** será utilizado os mais variados dados VARCHAR para texto, INT para inteiros e PRIMARY KEY, DATE para datas, DECIMAL para comissões e descontos, TEXT para descrições produto.
- **Modelo de dados:** Modelo Conceitual, Modelo Lógico e Modelo Físico.

### 3. REQUISITOS E FUNCIONALIDADES

#### 3.1. Planejamento do Levantamento de Requisitos

- **Definir Objetivos:** O objetivo é a criação de um sistema que realize vendas de produtos e tenha o controle dos dados das entidades submetidas. Esperamos que contemple detalhadamente as funcionalidades de acordo com o contexto do negócio estudado.
- **Partes interessadas:** Vendedores, gerentes, clientes, estoquistas, quem compra os produtos, almoxarifado, fornecedores, financeiro.

#### 3.2. Coleta de Informações

- **Observação:** Observamos os usuários no ambiente em que o sistema é utilizado e tentamos replicar ao máximo de acordo com a logística utilizada pela empresa. Isso nos forneceu insights sobre como eles interagem com o sistema existente e suas necessidades reais.

#### 3.3. Documentação dos Requisitos

- **Requisitos Funcionais com classificação:** Abaixo estão listados as funcionalidades que o sistema deve oferecer e sua classificação de prioridade (essencial-prioridade alta, opcional-pode ou não conter e desejáveis - esperado obter como resultado).
  - Efetuar vendas - essenciais
  - Atualização de quantidades de estoque - essenciais
  - Cadastro de clientes - opcionais
  - Acompanhamento do desempenho da empresa - opcionais
  - Visualização do lucro - essenciais
  - Melhor gestão - desejáveis
  - Armazenamento de dados - essenciais
  - Maior desempenho - desejáveis
- **Avaliação de Impacto:** Abaixo estão listados o impacto de cada requisito no sistema e no negócio, priorizando os que oferecem maior valor ou são críticos para o sucesso do projeto.
  - Efetuar vendas - essenciais: A funcionalidade de realizar vendas é a principal função do sistema, sem que o sistema não possa operar como pretendido. Permite a transferência de produtos ou serviços, essencial para o fluxo de receita.
  - Atualização de quantidades de estoque - essenciais: A atualização automática dos itens de estoque é crucial para manter a precisão das informações disponíveis sobre os produtos, evitando a venda de itens fora de estoque ou a falta de produtos.
  - Cadastro de clientes - opcionais: Embora a funcionalidade de cadastro de clientes não seja essencial para a operação básica do sistema, ela permite um gerenciamento mais eficiente dos relacionamentos com clientes e facilita o atendimento personalizado.
  - Acompanhamento do desempenho da empresa - opcionais: A função de acompanhar o desempenho fornece insights sobre como o negócio está saindo, mas não é vital para a operação diária do sistema.
  - Visualização do lucro - essenciais: A capacidade de visualizar e analisar lucros é crucial para monitorar a saúde financeira do negócio e tomar decisões informadas sobre estratégias e investimentos.

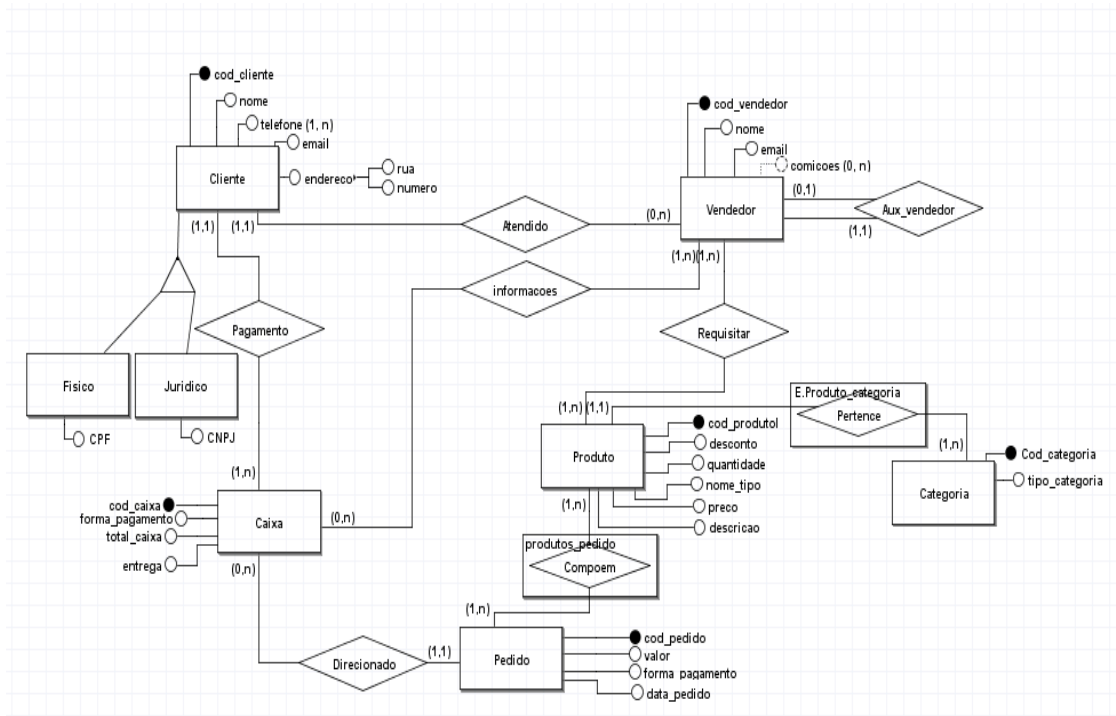
- Melhor gestão - desejáveis: Mais eficiente redução de problemas operacionais, mas não são fundamentais para o funcionamento básico do sistema.
- Armazenamento de dados - essenciais: O armazenamento de dados é essencial para garantir a persistência e integridade das informações, o que permite a continuidade das operações e o atendimento aos requisitos de conformidade e segurança.
- Maior desempenho - desejáveis: Melhorar o desempenho, aumentar a eficiência e a velocidade das operações, resultando em uma experiência mais segura para o usuário.

### 3.4. Definição das Funcionalidades Principais

- **Funcionalidades Essenciais:** É esperado que se possa realizar vendas dos produtos sem erros no estoque e também armazenar as informações com mais facilidade. Garante uma base de dados confiável das informações da empresa. Os vendedores terão mais desempenho para realizar vendas e poderão receber comissões por elas. Os clientes terão suas informações e endereços cadastrados caso queiram receber o material diretamente em casa, terão também um histórico de compra caso haja algum erro com o recebimento. As vendas estarão armazenadas na base de dados facilitando o financeiro da empresa, visando margem de lucro e perda e podendo ser consultadas em caso de erros. O estoque estará sempre em dia, uma vez que quando realizada a venda será dada baixa automaticamente no estoque automaticamente.

## 4. DESENVOLVIMENTO

### 4.1 MODELO CONCEITUAL



### 4.2 MODELO LÓGICO

#### Clientes

# cod_cliente	Integer	not null
nome	varchar, 100	not null
email	Varchar, 100	
Rua	Varchar, 50	Not null
Numero	Varchar, 10	Not null
Cpf	Varchar, 11	Unique not null
Cnpj	Varchar, 14	Unique not null
Telefone	Varchar, 15	Not null

#### Vendedores

# cod_vendedor	Integer	not null
nome	Varchar, 100	not null
email	Varchar, 100	
Comissao	Decimal, (10,2)	
auxiliar	boolean	

### Pedidos

# cod produto	integer	not null
desconto	float	
nome	varchar, 100	not null
Quantidade	Integer	not null
Preco	Decimal (10,2)	Not null
Descricao	text	not null

### Produtos

# cod pedido	integer	not null
valor total	Decimal (10,2)	not null
data pedido	DATE	
forma pagamento	Varchar, 50	Not null
@cod_cliente		cod_cliente referencia clientes(cod_cliente)
@cod_vendedor		cod_vendedor referencia vendedores(cod_vendedor)

### Categorias

# cod categoria	integer	not null
tipo categoria	Varchar, 50	
@cod_produto	Integer	cod_produto referencia produtos(cod_produto)

### Caixas

# cod caixa	integer	not null
forma pagamento	Varchar, 50	not null- editora referencia codigo.editora
total caixa	Decimal(10,2)	not null
Entrega	Varchar, 50	Not null
@cod_cliente	integer	cod_cliente referencia clientes(cod_cliente)
@cod_vendedor	integer	cod_vendedor referencia vendedores (cod_vendedor)
@cod_pedido	integer	cod_pedido referencia pedidos(cod_pedido)



### 4.3 MODELO FÍSICO

#### 1. Cria tabelas

<pre>CREATE TABLE clientes (   cod_cliente INT PRIMARY KEY NOT NULL,   nome VARCHAR(100) NOT NULL,   email VARCHAR(100),   numero VARCHAR(10),   rua VARCHAR(50),   CPF VARCHAR(11) UNIQUE,   CNPJ VARCHAR(14) UNIQUE,   telefone VARCHAR(15) );</pre>	<pre>CREATE TABLE vendedores (   cod_vendedor INT PRIMARY KEY NOT NULL,   nome VARCHAR(100) NOT NULL,   email VARCHAR(100),   comissao DECIMAL(10, 2),   auxiliar bool );</pre>
<pre>CREATE TABLE produtos (   cod_produto INT PRIMARY KEY NOT NULL,   desconto float,   quantidade INT NOT NULL,   nome_tipo VARCHAR(100) NOT NULL,   preco DECIMAL(10, 2) NOT NULL,   descricao TEXT );</pre>	<pre>CREATE TABLE categorias (   cod_categoria INT PRIMARY KEY NOT NULL,   tipo_categoria VARCHAR(50),   cod_produto INT,   FOREIGN KEY (cod_produto)   REFERENCES produtos(cod_produto) );</pre>
<pre>CREATE TABLE pedidos (   cod_pedido INT PRIMARY KEY NOT NULL,   valor_total DECIMAL(10, 2) NOT NULL,   forma_pagamento VARCHAR(50) NOT NULL,   data_pedido DATE,   cod_cliente INT,   cod_vendedor INT,   FOREIGN KEY (cod_cliente)   REFERENCES clientes(cod_cliente),   FOREIGN KEY (cod_vendedor)   REFERENCES vendedores(cod_vendedor) );</pre>	<pre>CREATE TABLE caixas(   cod_caixa INT PRIMARY KEY NOT NULL,   forma_pagamento VARCHAR(50) NOT NULL,   total_caixa DECIMAL(10, 2),   entrega VARCHAR(50) NOT NULL,   cod_cliente INT,   cod_vendedor INT,   cod_pedido INT,   FOREIGN KEY (cod_cliente)   REFERENCES clientes(cod_cliente),   FOREIGN KEY (cod_vendedor)   REFERENCES vendedores(cod_vendedor),</pre>

	FOREIGN KEY (cod_pedido) REFERENCES pedidos(cod_pedido) );
--	---

## 2. Inserts

<p>INSERT INTO clientes (cod_cliente, nome, email, numero, rua, cpf, telefone)</p> <p>VALUES</p> <p>(1, 'João Silva', 'joao.silva@gmail.com', '286', 'Luiz Ceron', '03786097102', '5554991846755'),</p> <p>(2, 'Maria Oliveira', 'maria.oliveira@gmail.com', '52', 'Leopoldo Magalhas', '04275643298', '5551992345678'),</p> <p>(3, 'Pedro Souza', 'pedro.souza@hotmail.com', '128', 'Matheus Franscico', '82567354392', '5552991740244'),</p> <p>(4, 'Ana Pereira', 'ana.pereira@gmail.com', '1232', 'Alcides Rebechi', '21175463291', '5552984645432'),</p> <p>(5, 'Carlos Lima', 'carlos.lima@hotmail.com', '98', 'Roberval Jeronimo', '23678390872', '5554984039321'),</p> <p>(6, 'Fernanda Alves', 'fernanda.alves@hotmail.com', '904', 'Patolino Xavier', '73678690843', '5554992878765'),</p> <p>(7, 'Rafael Castro', 'rafael.castro@gmail.com', '729', 'Pernalonga Coelho', '23467585795', '5551992837465'),</p> <p>(8, 'Juliana Costa', 'juliana.costa@hotmail.com', '187', 'Guarani', '95847362789', '5551990382637'),</p> <p>(9, 'Ricardo Nunes', 'ricardo.nunes@gmail.com', '105', 'Romero Britto', '19384732028', '5554998364536'),</p> <p>(10, 'Patrícia Fernandes', 'patricia.fernandes@gmail.com', '345', 'Homer Simpson', '78364597802', '5551984657865'),</p> <p>(11, 'Sérgio Barbosa', 'sergio.barbosa@hotmail.com', '99', 'Presidente Vargas', '37263453828', '5554981726326'),</p> <p>(12, 'Luciana Martins', 'luciana.martins@gmail.com', '10', 'Avenida Brasil', '27364056840', '555499983654');</p>	<p>INSERT INTO vendedores (cod_vendedor, nome, email, comissao, auxiliar)</p> <p>VALUES</p> <p>(1, 'José Rodrigues', 'jose.rodrigues@gmail.com', 5.00, '0'),</p> <p>(2, 'Carlos Ribeiro', 'carlos.ribeiro@hotmail.com', 7.50, '1'),</p> <p>(3, 'Paula Gonçalves', 'paula.goncalves@gmail.com', 4.00, '0'),</p> <p>(4, 'Roberta Almeida', 'roberta.almeida@gmail.com', 6.00, '1'),</p> <p>(5, 'Marcelo Dias', 'marcelo.dias@gmail.com', 8.00, '0'),</p> <p>(6, 'Cláudia Moura', 'claudia.moura@gmail.com', 3.50, '0'),</p> <p>(7, 'Henrique Souza', 'henrique.souza@hotmail.com', 5.50, '0'),</p> <p>(8, 'Rosa Mendes', 'rosa.mendes@hotmail.com', 6.50, '1'),</p> <p>(9, 'Adriano Silva', 'adriano.silva@hotmail.com', 7.00, '0'),</p> <p>(10, 'Carolina Santos', 'carolina.santos@gmail.com', 4.50, '1'),</p> <p>(11, 'Lucio Castro', 'lucio.castro@gmail.com', 6.75, '1'),</p> <p>(12, 'Gustavo Neves', 'gustavo.neves@gmail.com', 5.25, '0');</p>
<p>INSERT INTO produtos (cod_produto, desconto, quantidade, nome_tipo, preco, descricao)</p>	<p>INSERT INTO pedidos (cod_pedido, valor_total, forma_pagamento, data_pedido, cod_cliente, cod_vendedor)</p>

VALUES (1, 0, 100, 'Cimento', 40.00, 'Cimento'), (2, 0, 200, 'Tijolo', 30.00, 'Tijolo'), (3, 0, 150, 'Telha', 20.00, 'Telha'), (4, 0, 120, 'Tinta', 40.00, 'Tinta'), (5, 0, 300, 'Trelça', 50.00, 'Trelça'), (6, 0, 10, 'Furadeira', 200.00, 'Furadeira'), (7, 0, 90, 'Estaca de madeira', 5.00, 'Estaca de madeira'), (8, 0, 110, 'Cano PVC', 15.00, 'Cano PVC'), (9, 0, 130, 'Fio elétrico', 10.00, 'Fio elétrico'), (10, 0, 160, 'Martelo', 10.00, 'Martelo'), (11, 0, 70, 'Piso', 50.00, 'Piso'), (12, 0, 80, 'Janela', 35.00, 'Janela');	VALUES (1, 100.00, 'Cartão de Crédito', '2024-08-01', 1, 1), (2, 150.00, 'Boleto', '2024-08-02', 2, 2), (3, 200.00, 'Dinheiro', '2024-08-03', 3, 3), (4, 250.00, 'Cartão de Débito', '2024-08-04', 4, 4), (5, 300.00, 'Pix', '2024-08-05', 5, 5), (6, 350.00, 'Cartão de Crédito', '2024-08-06', 6, 6), (7, 400.00, 'Boleto', '2024-08-07', 7, 7), (8, 450.00, 'Dinheiro', '2024-08-08', 8, 8), (9, 500.00, 'Cartão de Débito', '2024-08-09', 9, 9), (10, 550.00, 'Pix', '2024-08-10', 10, 10), (11, 600.00, 'Cartão de Crédito', '2024-08-11', 11, 11), (12, 650.00, 'Boleto', '2024-08-12', 12, 12);
INSERT INTO categorias (cod_categoria, tipo_categoria, cod_produto) VALUES (1, 'Ferramentas Manuais', 10), (2, 'Acabamentos', 3), (3, 'Metais', 5), (4, 'Tijolos e Blocos', 2), (5, 'Cimento e Argamassa', 1), (6, 'Tubos e Conexões', 8), (7, 'Elétrica e Iluminação', 9), (8, 'Portas e Janelas', 12), (9, 'Revestimentos e Pisos', 11), (10, 'Madeiras e Compensados', 7), (11, 'Ferramentas Elétricas', 11), (12, 'Tintas e Vernizes', 4);	INSERT INTO caixas (cod_caixa, forma_pagamento, total_caixa, entrega, cod_cliente, cod_vendedor, cod_pedido) VALUES (1, 'Cartão de Crédito', 100.00, 'Sim', 1, 1, 1), (2, 'Boleto', 150.00, 'Não', 2, 2, 2), (3, 'Dinheiro', 200.00, 'Sim', 3, 3, 3), (4, 'Cartão de Débito', 250.00, 'Não', 4, 4, 4), (5, 'Pix', 300.00, 'Sim', 5, 5, 5), (6, 'Cartão de Crédito', 350.00, 'Não', 6, 6, 6), (7, 'Boleto', 400.00, 'Sim', 7, 7, 7), (8, 'Dinheiro', 450.00, 'Não', 8, 8, 8), (9, 'Cartão de Débito', 500.00, 'Sim', 9, 9, 9), (10, 'Pix', 550.00, 'Não', 10, 10, 10), (11, 'Cartão de Crédito', 600.00, 'Sim', 11, 11, 11), (12, 'Boleto', 650.00, 'Não', 12, 12, 12);

### 3. Consultas

--Faz a média do valor total entre os caixas SELECT AVG(total_caixa) AS media_total_caixas FROM caixas;	--Soma o valor de todos os pedidos SELECT SUM(valor_total) AS valor_total_pedidos FROM pedidos;
--Seleciona o pedido com menor valor SELECT MIN(valor_total) AS pedido_menor_valor FROM pedidos;	--Retorna se os vendedores que possuem auxiliar SELECT cod_vendedor, nome FROM vendedores WHERE auxiliar=true;
--Retorna se os vendedores que não possuem auxiliar SELECT cod_vendedor, nome FROM vendedores WHERE auxiliar=false;	-- Mostra todos os pedidos na data selecionada SELECT * FROM pedidos WHERE data_pedido BETWEEN '2024-08-01' AND '2024-08-02';
-- Mostra todos os pedidos fora da data selecionada SELECT * FROM pedidos WHERE data_pedido NOT BETWEEN '2024-08-01' AND '2024-08-02';	-- Atualiza valor produto aplicando desconto UPDATE produtos SET desconto = desconto * 1.1 WHERE cod_produto = 101;
-- Deleta vendedor que foi demitido DELETE FROM vendedores WHERE cod_vendedor = 5;	-- Média de vendas por vendedor SELECT v.nome AS vendedor, AVG(p.valor_total) AS media_vendas FROM vendedores v JOIN pedidos p ON v.cod_vendedor = p.cod_vendedor GROUP BY v.nome;
--Valor total de vendas em um intervalo de datas SELECT SUM(p.valor_total) AS total_vendas FROM pedidos p WHERE p.data_pedido BETWEEN '2024-08-01' AND '2024-08-31';	--Total de vendas por forma de pagamento e cliente SELECT c.nome AS cliente, p.forma_pagamento, SUM(p.valor_total) AS total_vendas

	<pre>FROM clientes c JOIN pedidos p ON c.cod_cliente = p.cod_cliente GROUP BY c.nome, p.forma_pagamento;</pre>
--	--

## 5. CONCLUSÃO

Podemos concluir que com a execução deste trabalho obtivemos um vasto conhecimento sobre Banco de Dados e como o projeto demonstrou a importância de uma estruturação eficiente para garantir a integridade e funcionalidade do sistema. Através da criação de modelos conceitual, lógico e físico, e da implementação no PostgreSQL, foi possível organizar informações cruciais para a gestão de vendas, estoque e fluxo de caixa. Observa-se a importância em aprender o funcionamento e execução, a fim de avançar, explorando conteúdos mais complexos e expandindo conhecimentos prévios sobre o assunto.