



PROJETO 1 – Domínio de Projeto

Link do Vídeo de Domínio de Projeto: <https://www.youtube.com/watch?v=Bgncli8oLyA>

Link para o GitHub com o projeto: https://github.com/quelfialho/Mata60_ProjetoBD

1. PROBLEMA

A TechStore é uma plataforma de e-commerce especializada na venda de produtos eletrônicos como smartphones, laptops, câmeras e outros. Atualmente a empresa está com dificuldade para gerenciar seu catálogo de produtos e para criar dashboards que tragam informações úteis para tomada de decisões, o que pode estar limitando os seus resultados.

Pensando em melhorar a gestão do negócio, os sócios da TechStore decidiram contratar uma empresa especializada para criação de um sistema de informação que consiga ajudá-los no gerenciamento dos produtos e fornecedores, além de fornecer dados relevantes para tomada de decisões e acompanhamento dos resultados.

2. REQUISITOS DO SISTEMA DE INFORMAÇÃO

RS1: O sistema deve permitir a manipulação do catálogo de produtos da empresa.

RS2: O sistema deve permitir a manipulação das categorias dos produtos da empresa.

RS3: O sistema deve permitir a manipulação dos fornecedores dos produtos da empresa.

RS4: O sistema deve permitir a manipulação dos clientes da empresa.

RS5: O sistema deve fornecer uma visualização do estoque que agregue a quantidade armazenada de um produto por cada um dos seus fabricantes ativos.

RS6: O sistema deve permitir que o cliente compre um produto independentemente do fornecedor, mas internamente deve ser possível rastrear a origem (fornecedor) do produto associado à venda.

RS7: Inicialmente o sistema deve permitir apenas a venda, sem limite de quantidade (desde que seja do mesmo fornecedor), de um mesmo produto em cada compra. Logo, caso o cliente queira comprar produtos diferentes (ou não tenha a quantidade desejada do produto em apenas um fornecedor), precisaria fazer compras separadas.

RS8: O sistema deve fornecer insumos para construção de um gráfico visando entender a distribuição das notas dadas pelos clientes nas avaliações dos produtos.

RS9: O sistema deve fornecer insumos para construção de um gráfico visando entender o perfil demográfico dos clientes que mais realizaram compras.

RS10: O sistema deve fornecer insumos para construção de um gráfico visando entender o interesse dos clientes em cada uma das categorias de produtos.

RS11: O sistema deve fornecer insumos para construção de um gráfico visando entender o ticket médio das vendas em cada uma das categorias de produtos.

RS12: O sistema deve fornecer insumos para construção de um gráfico visando entender quantos clientes distintos já realizaram compras em relação ao total de clientes.

RS13: O sistema deve fornecer insumos para construção de um gráfico visando entender qual o valor investido nos produtos que estão no estoque.

RS14: O sistema deve fornecer insumos para construção de um gráfico visando entender a distribuição dos produtos em estoque por cada categoria.

RS15: O sistema deve fornecer insumos para construção de um gráfico visando entender o faturamento por cliente.

RS16: O sistema deve fornecer insumos para construção de um gráfico visando entender a média do preço de aquisição de um produto em seus diferentes fornecedores.

3. DELIMITAÇÃO DO MINI-MUNDO PARA O BANCO DE DADOS

- **tbl_produtos:** tabela que concentra as informações dos itens individuais que compõem o acervo da empresa.
 - **id_produto** [int, serial, PK]: código único que identifica o produto.
 - **nome** [text, não nulo]: nome atribuído ao produto.
 - **ce_categoria** [FK, não nulo]: chave estrangeira que referencia uma categoria na *tbl_categorias*.
 - **marca** [text, não nulo]: identificação da marca do fabricante do produto.
 - **descricao** [text, não nulo]: descrição das características do produto.
 - **preco** [float, não nulo]: preço de venda do produto.
 - **deleted** [boolean, default false]: atributo para informar que um produto foi removido do sistema mas sem realmente removê-lo do banco para garantir a consistência dos demais registros.
- **tbl_categorias:** tabela que concentra as categorias dos produtos.
 - **id_categoria** [int, serial, PK]: código único que identifica a categoria.
 - **nome** [text, não nulo]: nome atribuído a uma categoria.
 - **descricao** [text, nulo]: descrição opcional da categoria.
 - **deleted** [boolean, default false]: atributo para informar que uma categoria foi removida do sistema mas sem realmente removê-la do banco para garantir a consistência dos demais registros.
- **tbl_fornecedores:** tabela que concentra os fornecedores de produtos da TechStore.
 - **id_fornecedor** [int, serial, PK]: código único que identifica um fornecedor.
 - **nome** [text, não nulo]: nome do fornecedor.
 - **cnpj** [text, não nulo]: código nacional de pessoa jurídica do fornecedor.

- **avaliacao [int (1 a 5), not null]**: campo que exibe a avaliação da satisfação da empresa TechStore com o fornecedor.
- **deleted [boolean, default false]**: atributo para informar que um fornecedor foi removido do sistema mas sem realmente removê-lo do banco para garantir a consistência dos demais registros.
- **tbl_clientes**: tabela que concentra os dados dos clientes da TechStore.
 - **id_cliente [int, serial, PK]**: código único que identifica um cliente.
 - **nome [text, não nulo]**: nome do cliente.
 - **cpf: [text, não nulo, único]**: código de pessoa física do cliente.
 - **email [text, não nulo, único]**: e-mail do cliente.
 - **realizou_compra [boolean, default false]**: atributo para identificar se o cliente já realizou uma compra na TechStore.
 - **sexo [char, não nulo]**: sexo biológico do cliente. M para masculino e F para feminino.
 - **deleted [boolean, default false]**: atributo para informar que um cliente foi removido do sistema mas sem realmente removê-lo do banco para garantir a consistência dos demais registros.
- **tbl_produtos_fornecedores**: tabela de relacionamento entre produtos e fornecedores que identifica informações pertinentes a essa relação.
 - **id_produto_fornecedor [int, serial, PK]**: código único que identifica a relação entre produto e fornecedor.
 - **ce_id_produto [FK, não nulo]**: chave estrangeira que identifica o produto na tabela **tbl_produtos**.
 - **ce_id_fornecedor [FK, não nulo]**: chave estrangeira que identifica o fornecedor na tabela **tbl_fornecedores**.
 - **quantidade_em_estoque [int, não nulo]**: quantidade do produto do fornecedor em estoque.
 - **preco_fornecimento [float, não nulo]**: preço de compra do produto com o fornecedor.
 - **deleted [boolean, default false]**: atributo para informar que a relação entre um produto e fornecedor foi removida do sistema mas sem realmente removê-la do banco para garantir a consistência dos demais registros.
- **tbl_vendas**: tabela que concentra as informações das vendas da TechStore.

- **id_venda [int, serial, PK]:** código único que identifica uma venda.
 - **ce_id_produto_fornecedor [FK, não nulo]:** chave estrangeira que identifica a relação entre um produto e um fornecedor na tabela **tbl_produtos_fornecedores**.
 - **ce_id_cliente [FK, não nulo]:** chave estrangeira que identifica um cliente na tabela **tbl_clientes**.
 - **quantidade [int, não nulo]:** número de itens comprados.
 - **valor_total [float, não nulo]:** valor total da venda.
 - **criado_em [timestamp, default now()]:** data da venda criada automaticamente.
- **tbl_avaliacoes:** tabela que concentra as informações das avaliações dos produtos feitas pelos clientes da TechStore.
 - **id_avaliacao [int, serial, PK]:** código único que identifica uma avaliação.
 - **ce_id_produto [FK, não nulo]:** chave estrangeira que identifica o produto na tabela **tbl_produtos**.
 - **ce_id_cliente [FK, não nulo]:** chave estrangeira que identifica um cliente na tabela **tbl_clientes**.
 - **nota [int (1 a 5), not null]:** campo que armazena a nota da satisfação do cliente com o produto adquirido na TechStore.
 - **comentario [text, null]:** campo opcional para armazenar comentários adicionais na avaliação
- **tbl_interesses:** tabela que concentra as informações do interesse dos clientes nas categorias de produtos da TechStore.
 - **id_interesse [int, serial, auto-incremento, PK]:** código único que identifica a relação entre cliente e produto.
 - **ce_id_cliente [FK, não nulo]:** chave estrangeira que referencia um cliente na tabela **tbl_clientes**.
 - **ce_id_categoria [FK, não nulo]:** chave estrangeira que referencia uma categoria na tabela **tbl_categoriass**.
 - **criado_em [timestamp, default now()]:** data que o cliente sinalizou interesse em uma categoria de produtos.
 - **tbl_enderecos:** tabela que concentra as informações dos endereços dos clientes da TechStore.

- **id_endereco** [int, serial, PK]: código único que identifica o endereço.
- **ce_id_cliente** [FK, não nulo]: chave estrangeira que referencia um cliente na tabela **tbl_clientes**.
- **logradouro** [text, não nulo]: nome da rua, avenida, ou outro tipo de logradouro.
- **numero** [text, não nulo]: número do endereço (pode incluir letras, como em apartamentos ou conjuntos).
- **complemento** [text, nulo]: informações adicionais sobre o endereço, como apartamento ou bloco.
- **bairro** [text, não nulo]: bairro onde o endereço está localizado.
- **cidade** [text, não nulo]: cidade onde o endereço está localizado.
- **estado** [text, não nulo]: estado onde o endereço está localizado.
- **cep** [text, não nulo]: código postal do endereço.
- **deleted** [boolean, default false]: atributo para informar que um endereço foi removido do sistema sem realmente removê-lo do banco para garantir a consistência dos demais registros.

RELACIONAMENTO DAS ENTIDADES

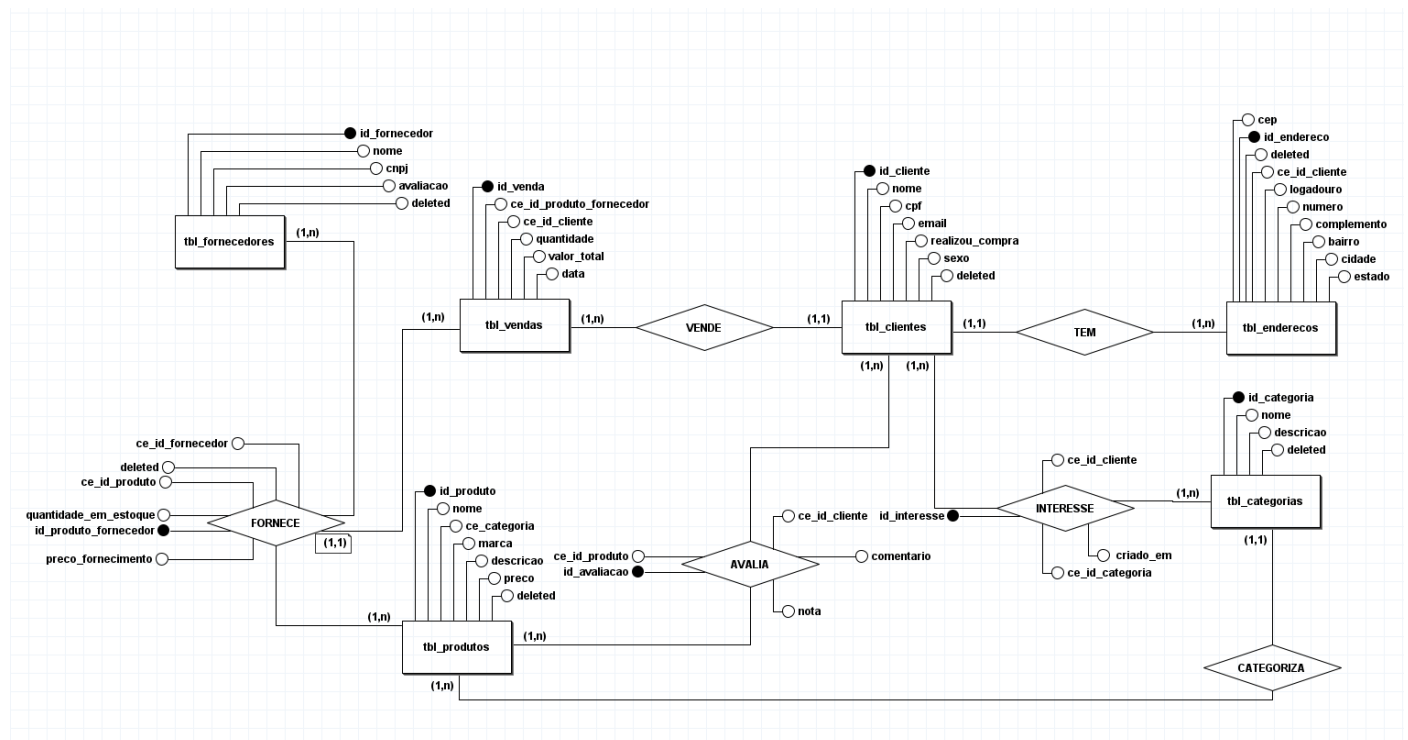
- **RE1:** Um produto pode ter apenas uma categoria. Uma categoria pode estar associada a vários produtos.
- **RE2:** Um produto pode estar associado a vários fornecedores. Um fornecedor pode estar associado a vários produtos.
- **RE3:** Uma venda está associada a uma relação produto/fornecedor específica. E uma relação produto/fornecedor específica pode estar associada a várias vendas.
- **RE4:** Uma venda está associada a um cliente. Um cliente pode realizar várias vendas.
- **RE5:** Um cliente pode ter vários endereços. Um endereço só pode ser associado a um cliente.
- **RE6:** Um cliente pode ter interesse em várias categorias. Uma categoria pode interessar vários clientes.
- **RE7:** Um cliente pode avaliar vários produtos. Um produto pode ser avaliado por vários clientes.

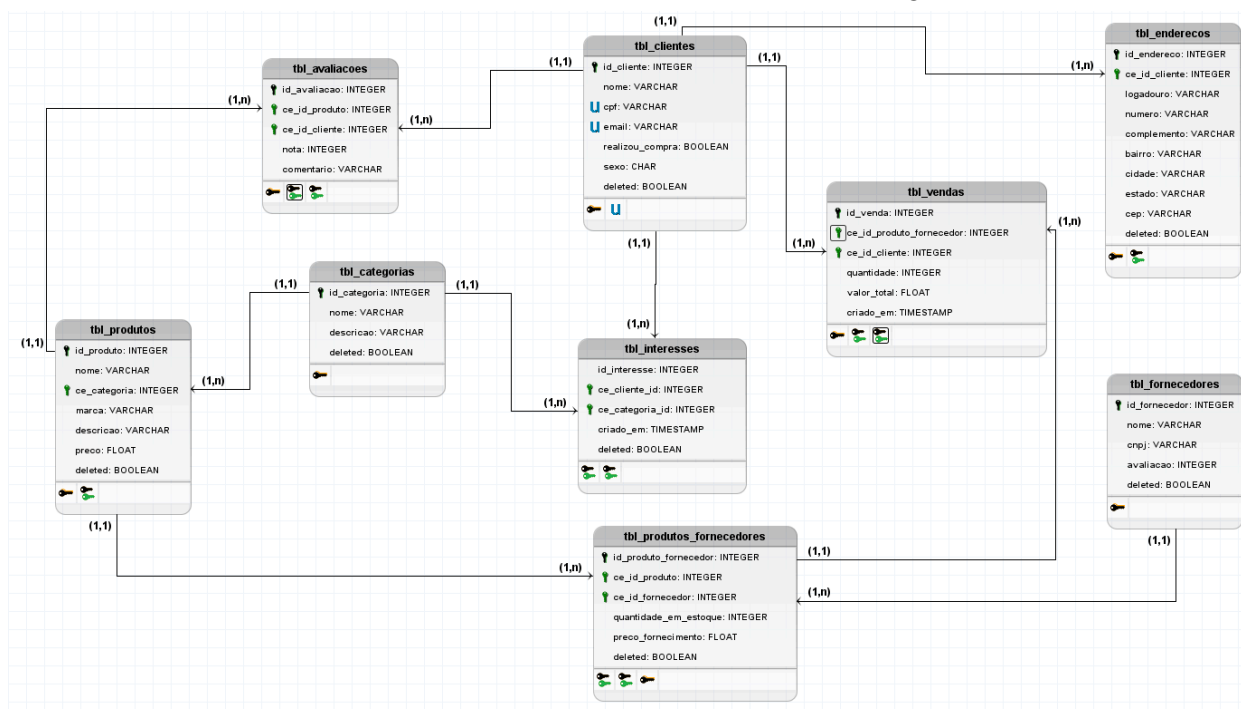
4. PROJETO CONCEITUAL

O projeto conceitual foi construído com base na ferramenta BrModelo apresentada no laboratório da disciplina. Em anexo estão disponíveis o arquivo conceitual do BrModelo e uma imagem para garantir a visualização do modelo do projeto.

Vale ressaltar que a partir da relação entre produtos e fornecedores será gerada uma tabela associativa, que será usada na tabela vendas como chave estrangeira de modo a associar uma venda a um produto e fornecedor específicos para atender ao RS4.

E, como o professor Robespierre comentou na sala, seria uma relação ternária que não está necessariamente prevista na estrutura da ferramenta, mas fez as adaptações necessárias com um comentário ao lado da relação "FORNECE" para indicar que uma venda está associada apenas a um elemento da relação entre produtos e fornecedores.





6. PROJETO FÍSICO

Abaixo teremos os comandos do tipo DDL (Data Definition Language) para um sistema de banco de dados PostgreSQL. Também disponível em: [Projeto Físico - Notion](#)

1. Tabela de categorias:

```
CREATE TABLE tbl_categorias (  
    id_categoria SERIAL PRIMARY KEY,  
    nome TEXT NOT NULL,  
    descricao TEXT,  
    deleted BOOLEAN DEFAULT FALSE  
);
```

2. Tabela de Produtos:

```
CREATE TABLE tbl_produtos (  
    id_produto SERIAL PRIMARY KEY,  
    nome TEXT NOT NULL,  
    ce_categoria INT NOT NULL REFERENCES tbl_categorias(id_categoria),  
    marca TEXT NOT NULL,  
    descricao TEXT NOT NULL,  
    preco FLOAT NOT NULL,  
    deleted BOOLEAN DEFAULT FALSE  
);
```

3. Tabela de Fornecedores:

```
CREATE TABLE tbl_fornecedores (  
    id_fornecedor SERIAL PRIMARY KEY,  
    nome TEXT NOT NULL,  
    cnpj TEXT NOT NULL,  
    avaliacao INT CHECK(avaliacao >= 1 AND avaliacao <= 5) NOT NULL,  
    deleted BOOLEAN DEFAULT FALSE  
);|
```

4. Tabela de Clientes:

```
CREATE TABLE tbl_clientes (  
    id_cliente SERIAL PRIMARY KEY,  
    nome TEXT NOT NULL,  
    cpf TEXT NOT NULL UNIQUE,  
    email TEXT NOT NULL UNIQUE,  
    realizou_compra BOOLEAN DEFAULT FALSE,  
    sexo CHAR(1) NOT NULL CHECK(sexo IN ('M', 'F')),  
    deleted BOOLEAN DEFAULT FALSE  
);|
```

5. Tabela que associa Produtos e Fornecedores:

```
CREATE TABLE tbl_produtos_fornecedores (  
    id_produto_fornecedor SERIAL PRIMARY KEY,  
    ce_id_produto INT NOT NULL REFERENCES tbl_produtos(id_produto),  
    ce_id_fornecedor INT NOT NULL REFERENCES tbl_fornecedores(id_fornecedor),  
    quantidade_em_estoque INT NOT NULL,  
    preco_fornecimento FLOAT NOT NULL,  
    deleted BOOLEAN DEFAULT FALSE  
);
```

6. Tabela de Vendas

```
CREATE TABLE tbl_vendas (  
    id_venda SERIAL PRIMARY KEY,  
    ce_id_produto_fornecedor INT NOT NULL REFERENCES tbl_produtos_fornecedores(id_produto_fornecedor),  
    ce_id_cliente INT NOT NULL REFERENCES tbl_clientes(id_cliente),  
    quantidade INT NOT NULL,  
    valor_total FLOAT NOT NULL,  
    criado_em TIMESTAMP DEFAULT NOW()  
);
```

7. Tabela de Avaliações:

```
CREATE TABLE tbl_avaliacoes (  
    id_avaliacao SERIAL PRIMARY KEY,  
    ce_id_produto INT NOT NULL REFERENCES tbl_produtos(id_produto),  
    ce_id_cliente INT NOT NULL REFERENCES tbl_clientes(id_cliente),  
    nota INT CHECK(nota >= 1 AND nota <= 5) NOT NULL,  
    comentario TEXT  
);
```

8. Tabela de Interesses:

```
CREATE TABLE tbl_interesses (  
    id_interesse SERIAL PRIMARY KEY,  
    ce_id_cliente INT NOT NULL REFERENCES tbl_clientes(id_cliente),  
    ce_id_categoria INT NOT NULL REFERENCES tbl_categorias(id_categoria),  
    criado_em TIMESTAMP DEFAULT NOW()  
);
```

9. Tabela de Endereços:

```
CREATE TABLE tbl_enderecos (  
    id_endereco SERIAL PRIMARY KEY,  
    ce_id_cliente INT NOT NULL REFERENCES tbl_clientes(id_cliente),  
    logradouro TEXT NOT NULL,  
    numero TEXT NOT NULL,  
    complemento TEXT,  
    bairro TEXT NOT NULL,  
    cidade TEXT NOT NULL,  
    estado TEXT NOT NULL,  
    cep TEXT NOT NULL,  
    deleted BOOLEAN DEFAULT FALSE  
);
```

7. IMPLANTAÇÃO

O projeto foi implementado em um banco de dados PostgreSQL rodando em um container Docker. Foi gerado um arquivo docker-compose para reprodução do banco de dados criado.

Além disso, foi criado um programa simples em JavaScript usando o framework NodeJs para facilitar o processo de build da aplicação, no qual ele gerencia as configuração de subida do container, criação das tabelas com os comandos DDL's necessários e popula o banco de dados com 100 instâncias em cada tabela, exceto a tabela de categorias que se enquadra como tabela de controle, conforme previsto no barema.

```
JS tabelas.js src X
src > JS tabelas.js > ...
1 const pool = require("../db");
2
3 class Tabelas {
4   async criar() {
5     await this.criaTabelaCategorias();
6     await this.criaTabelaProdutos();
7     await this.criaTabelaFornecedores();
8     await this.criaTabelaClientes();
9     await this.criaTabelaProdutosFornecedores();
10    await this.criaTabelaVendas();
11    await this.criaTabelaAvaliacoes();
12    await this.criaTabelaInteresses();
13    await this.criaTabelaEnderecos();
14  }
15  async criaTabelaCategorias() {
16    await pool.query(`
17      CREATE TABLE tbl_categorias (
18        id_categoria SERIAL PRIMARY KEY,
19        nome TEXT NOT NULL,
20        descricao TEXT,
21        deleted BOOLEAN DEFAULT FALSE
22      );
23    `);
24    console.log("Tabela tbl_categorias criada com sucesso!");
25    return;
26  }
27
28  async criaTabelaProdutos() {
```