

# Relatório do Banco de Dados

## Dia 06/03

Em primeiro plano foi feita a escolha do tema do banco de dados, um E-commerce, que vai gerenciar pagamentos, pedidos, informações dos clientes, produtos e estoque, entre outras funções, e foram criadas as 10 tabelas iniciais (Produto, Endereço, Cliente, Carinho, Afiliado, Armazém, Vendedor, Avaliação, Credencial, Pagamento).

## Dia 13/03

Após analisarmos melhor, fizemos a troca da tabela credencial que guardava as informações do usuário pela tabela armazém, que vai guardar as informações de onde fica o produto, cidade, capacidade.

Ademais, incluímos uma nova tabela de dados bancários para guardamos os dados bancários do vendedor e afiliado do E-Commerce.

## Dia 14/03

Fizemos a Primeira versão do modelo lógico

Produto(IDProduto, Nome, Preço, Qtd\_estoque, IDCategoria, IDVendedor)  
IDCategoria referencia Categoria  
IDvendedor referencia Vendedor

Endereço(IDEndereço, IDCliente, Rua, Cidade, CEP, Numero)  
IDCliente referencia Cliente

Cliente (IDCliente, Nome, CPF, Data De Nascimento, Telefone, E-mail, Senha)

Carrinho (IDCliente, Nome, IDProdutos, Preço, Status, IDPagamento, IDAfiliado, DataPagamento, IDClientes)  
IDPagamento referencia Pagamento  
IDAfiliado referencia Afiliado  
IDClientes referencia Clientes

Afiliado (IDAfiliado, Nome, CNPJ/CPF, DadosBancarios, PorcentagemComissão)

Pagamento(IDAfiliado, Nome, Valor, Tipo)

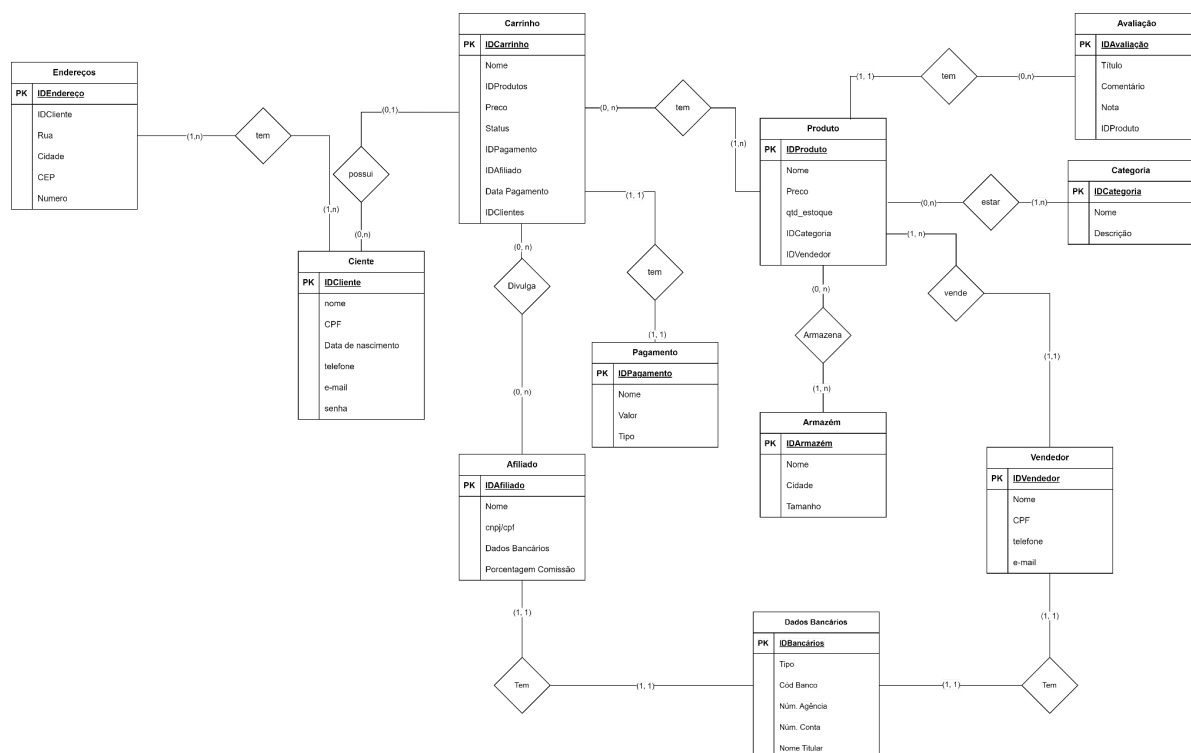
DadosBancarios(IDBancarios, Tipo, CodBanco, NumAgência, NumConta, NomeTitular)

Vendedor(DAfiliado, Nome, CPF, Telefone, Email)

Armazem(DAfiliado, Nome, Cidade, Tamanho)

Avaliação(IDAvaliação, Titulo, Comentário, Nota, IDProduto)  
IDProduto referencia Produto

---



## Normalização do banco de dados:

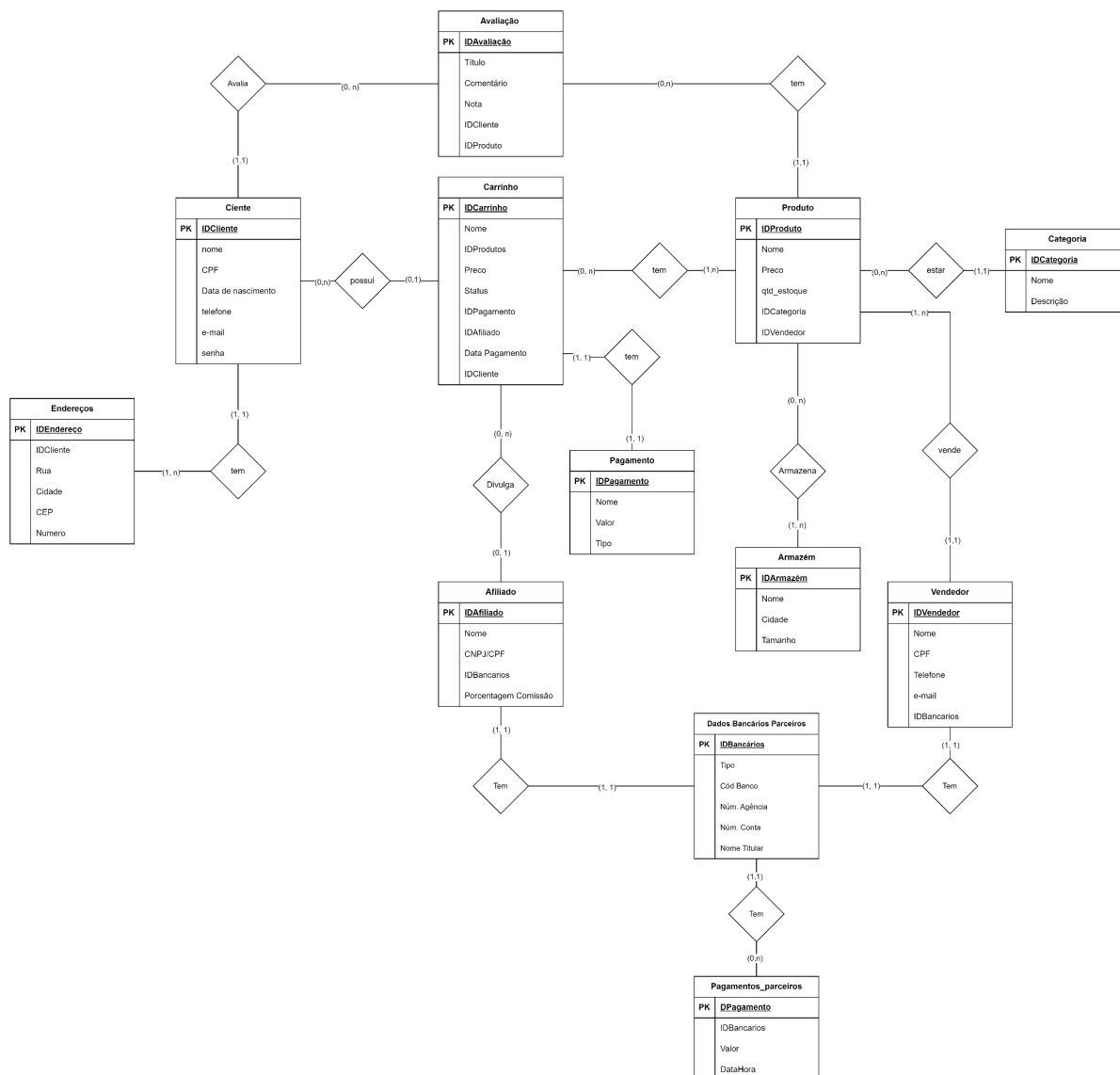
As tabelas Carrinho e Pagamentos são unificadas no banco de dados por terem cardinalidade do tipo 1:1

O relacionamento carrinho-produto é do tipo (n, n) logo gera uma nova tabela de associação Carrinho\_produto

O relacionamento carrinho-armazém é do tipo (n, n) logo gera uma nova tabela de associação Armazem\_produto

Além disso:

Como nosso banco não possibilitava calcular pagamentos feitos a vendedores e a afiliados, foi criado uma tabela denominada Pagamento\_parceiros que registra a DataHora, o Valor dos Pagamentos e para quem foi feito esse pagamento.



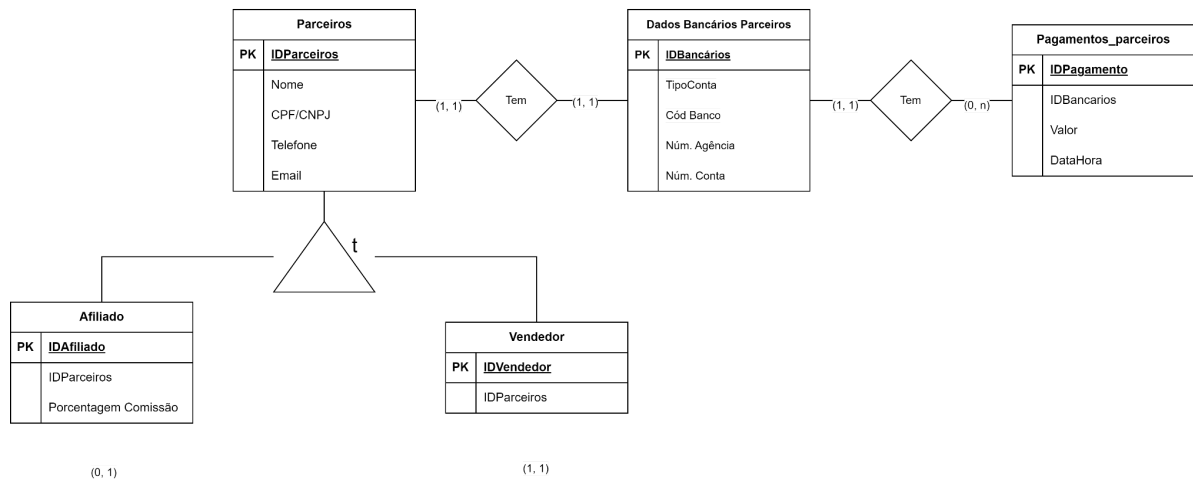
**Dia 27/03**

## – Especializações:

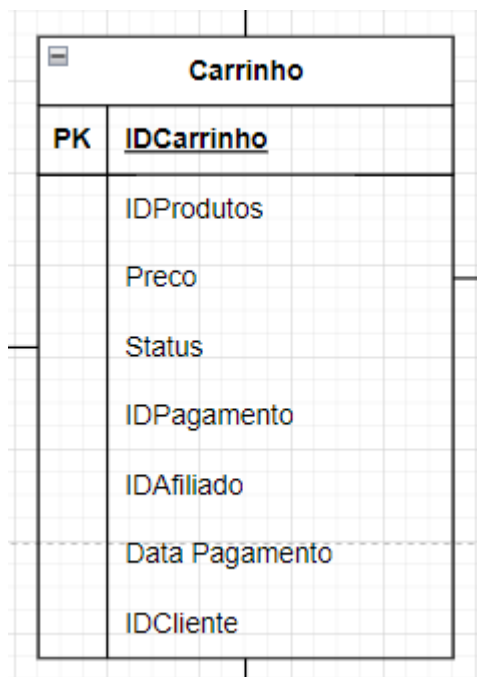
Foi criado uma generalização nas entidades afiliados e vendedor para a tabela parceiros, como cada tabela tem seus próprios relacionamentos distintos, optamos por usar uma tabela por entidade especializada.

As tabelas dadosBancarios e Parceiros são unificadas no banco de dados por terem cardinalidade do tipo 1:1.

## – Uso de Herança: Vendedor e Afiliado herdam de parceiros



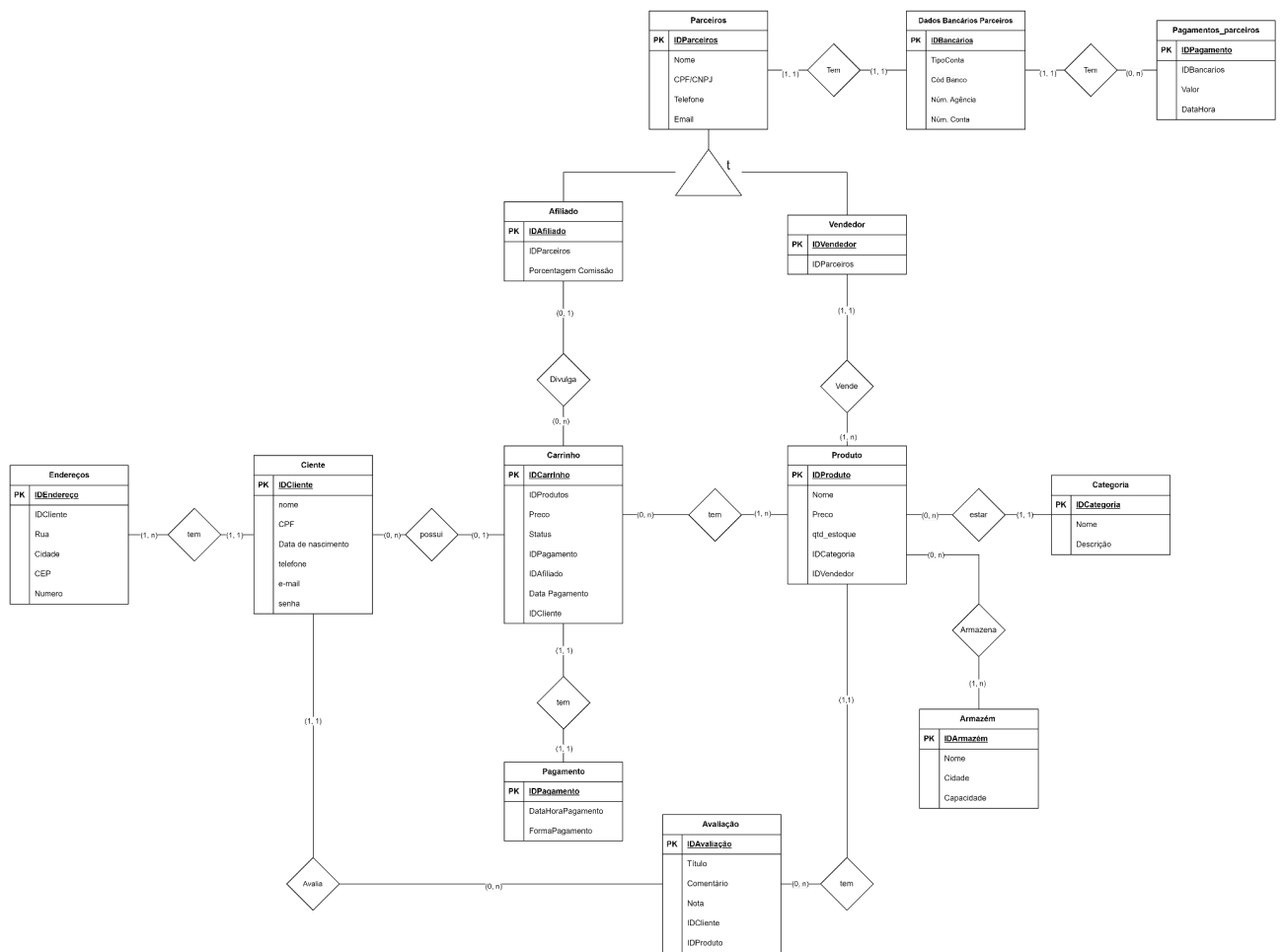
## Tabela com dados temporais: Carrinho



**Dia 03/04**

Colocamos AUTO\_INCREMENT em todas as chaves primárias no banco de dados

**Diagrama Entidade-Relacionamento finalizado:**



**Modelo Lógico Textual finalizado:**

Carrinho (IDCliente, IDProdutos, Preço, Status, IDAfiliado, DataPagamento, IDClientes, FormaPagamento, DataHoraPagamento)  
IDAfiliado referencia Afiliado  
IDCliente referencia Cliente

Armazem(IDArmazem, Nome, Cidade, Capacidade)

Produto(IDProduto, Nome, Preço, Qtd\_estoque, IDCategoria, IDVendedor)  
IDCategoria referencia Categoria  
IDvendedor referencia Vendedor

Categoria(IDCategoria, Nome, Descrição)

Carrinho\_produto(IDCarrinho\_produto, IDProduto, IDCarrinho, quantidade)  
IDProduto referencia produto  
IDCarrinho referencia Carrinho

Armazem\_produto(IDArmazem\_produto, IDProduto, IDArmazem, quantidade)  
IDProduto referencia produto  
IDArmazem referencia Armazem

Endereço(IDEndereço, IDCliente, Rua, Cidade, CEP, Numero)  
IDCliente referencia Cliente

Cliente (IDCliente, Nome, CPF, Data De Nascimento, Telefone, E-mail, Senha)

Parceiros(IDParceiros, Nome, CPF/CNPJ, Telefone, Email, TipoConta, CodBanco, NumAgência, NumConta)

Afiliado (IDAfiliado, PorcentagemComissão, IDParceiros)  
IDParceiros referencia Parceiros

Vendedor(IDVendedor, IDParceiros)  
IDParceiros referencia Parceiros

Parceiros\_Pagamentos(IDPagamento, IDParceiros, Valor, DataHora)  
IDParceiros referencia Parceiros

Avaliação(IDAvaliação, Titulo, Comentário, Nota, IDProduto, IDCliente)  
IDProduto referencia Produto  
IDCliente referencia Cliente

## Modelo Físico:

```
CREATE DATABASE seu_pai;
```

```
USE seu_pai;
```

```
CREATE TABLE cliente (  
    id_cliente INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(50) NOT NULL,  
    cpf VARCHAR(50) NOT NULL,  
    data_nasc DATE NOT NULL,  
    telefone VARCHAR(50) NOT NULL,  
    email VARCHAR(50) NOT NULL,  
    senha VARCHAR(30) NOT NULL  
);
```

```
CREATE TABLE endereco (  
    id_endereco INT AUTO_INCREMENT PRIMARY KEY,  
    id_cliente INT,  
    FOREIGN KEY (id_cliente) REFERENCES cliente (id_cliente),  
    rua VARCHAR(50) NOT NULL,  
    cidade VARCHAR(50) NOT NULL,  
    CEP VARCHAR(50) NOT NULL,  
    numero INT NOT NULL  
);
```

```
CREATE TABLE parceiros (  
    id_parceiros INT AUTO_INCREMENT PRIMARY KEY,  
    cpf_cnpj VARCHAR(50),  
    nome VARCHAR(50),  
    tipo_conta VARCHAR(50),  
    cod_banco INT NOT NULL,  
    num_agencia INT NOT NULL,  
    num_conta INT NOT NULL,  
    cpf VARCHAR(50),  
    telefone VARCHAR(50),  
    email VARCHAR(50)  
);
```

```
CREATE TABLE armazem (  
    id_armazem INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(50),  
    cidade VARCHAR(50),  
    capacidade INT NOT NULL  
);
```

```

CREATE TABLE categoria (
    id_categoria INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(50),
    descricao VARCHAR(100)
);

CREATE TABLE afiliado (
    id_afiliado INT PRIMARY KEY,
    comissao DECIMAL(10, 2) NOT NULL,
    id_parceiros INT NOT NULL,
    FOREIGN KEY (id_parceiros) REFERENCES parceiros (id_parceiros)
);

CREATE TABLE vendedor (
    id_vendedor INT AUTO_INCREMENT PRIMARY KEY,
    id_parceiros INT NOT NULL,
    FOREIGN KEY (id_parceiros) REFERENCES parceiros (id_parceiros)
);

CREATE TABLE produto (
    id_produto INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    preco DECIMAL(10, 2) NOT NULL,
    qtd_estoque INT NOT NULL,
    id_categoria INT NOT NULL,
    id_vendedor INT NOT NULL,
    FOREIGN KEY (id_categoria) REFERENCES categoria (id_categoria),
    FOREIGN KEY (id_vendedor) REFERENCES vendedor (id_vendedor)
);

CREATE TABLE carrinho (
    id_carrinho INT AUTO_INCREMENT PRIMARY KEY,
    id_cliente INT NOT NULL,
    preco DECIMAL(10, 2) NOT NULL,
    status VARCHAR(30),
    data_hora_pagamento DATE NOT NULL,
    id_afiliado INT NOT NULL,
    FOREIGN KEY (id_cliente) REFERENCES cliente (id_cliente),
    FOREIGN KEY (id_afiliado) REFERENCES afiliado (id_afiliado)
);

```



```
CREATE TABLE avaliacao (  
    id_avaliacao INT AUTO_INCREMENT PRIMARY KEY,  
    TITULO VARCHAR(50),  
    comentario VARCHAR(50),  
    nota INT NOT NULL,  
    id_produto INT NOT NULL,  
    id_cliente INT NOT NULL,  
    FOREIGN KEY (id_produto) REFERENCES produto (id_produto),  
    FOREIGN KEY (id_cliente) REFERENCES cliente (id_cliente)  
);
```

```
CREATE TABLE parceiros_pagamentos (  
    id_pagamento INT AUTO_INCREMENT PRIMARY KEY,  
    id_parceiros INT NOT NULL,  
    valor DECIMAL(10, 2) NOT NULL,  
    data DATE NOT NULL,  
    FOREIGN KEY (id_parceiros) REFERENCES parceiros (id_parceiros)  
);
```

```
CREATE TABLE armazem_produto (  
    id_armazem_produto INT AUTO_INCREMENT PRIMARY KEY,  
    id_produto INT NOT NULL,  
    id_armazem INT NOT NULL,  
    quantidade INT NOT NULL,  
    FOREIGN KEY (id_produto) REFERENCES produto (id_produto),  
    FOREIGN KEY (id_armazem) REFERENCES armazem (id_armazem)  
);
```

```
CREATE TABLE carrinho_produto (  
    id_carrinho_produto INT AUTO_INCREMENT PRIMARY KEY,  
    id_produto INT NOT NULL,  
    id_carrinho INT NOT NULL,  
    quantidade INT NOT NULL,  
    FOREIGN KEY (id_produto) REFERENCES produto (id_produto),  
    FOREIGN KEY (id_carrinho) REFERENCES carrinho (id_carrinho)  
);
```

---