

Universidade Estácio

Campus: POLO FSP-RO

Curso: Desenvolvimento Full Stack

Disciplina: Missão prática do 2º nível de conhecimento

Turma: 2025

Semestre Letivo: 1

Integrantes: Eliton Rodrigues de Oliveira

Relatório Discente de Acompanhamento

Título da Prática

Criação do Banco de Dados para o Sistema de Movimentação de Produtos

Objetivo da Prática

Esta prática tem como objetivo modelar e criar um banco de dados para um sistema de movimentação de produtos. O sistema deve permitir o cadastro de usuários, produtos e transações de compra e venda, diferenciando pessoas físicas e jurídicas. Além disso, será utilizado o SQL Server Management Studio (SSMS) para criar e gerenciar o banco de dados, garantindo boas práticas na modelagem e implementação.

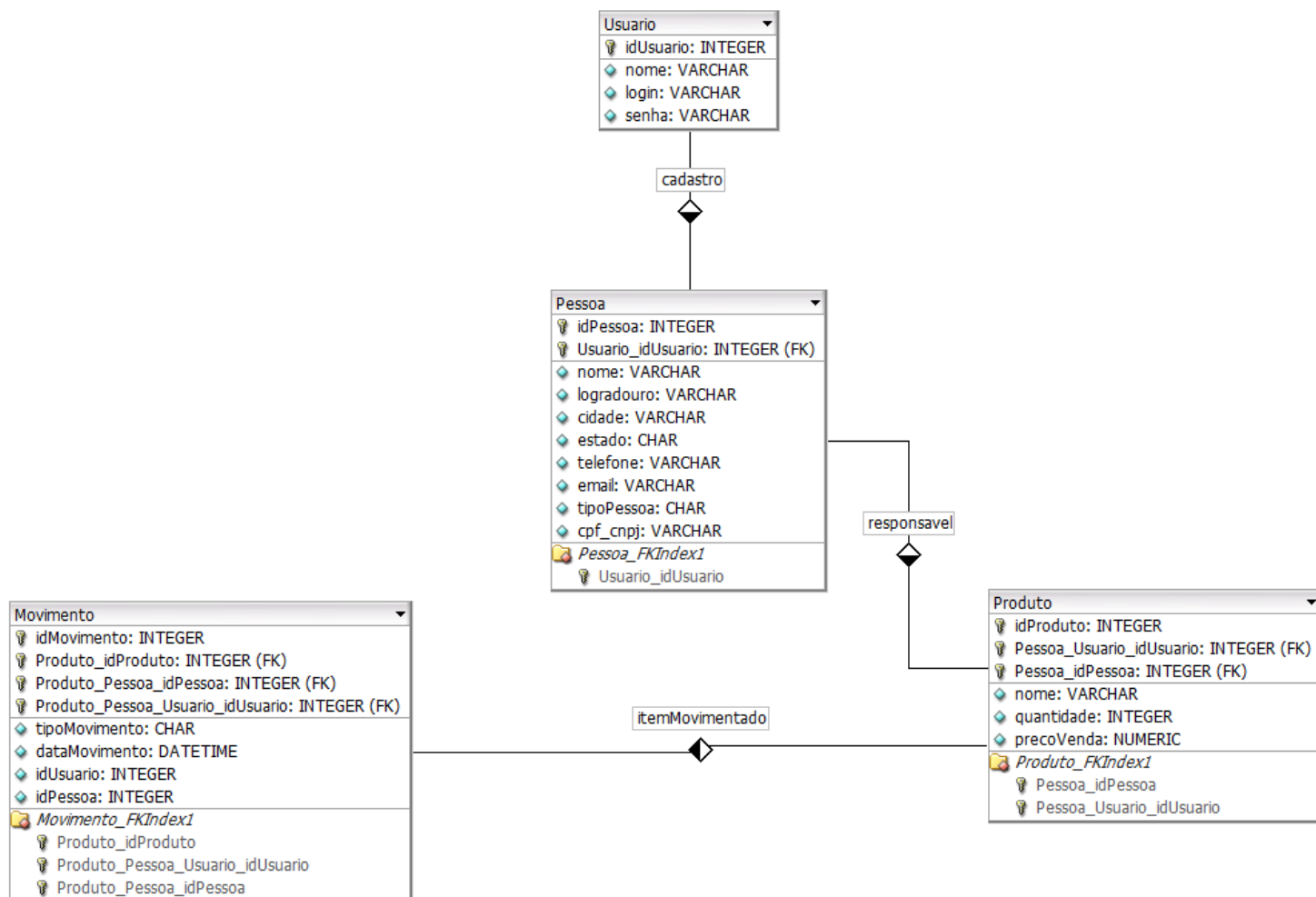
Desenvolvimento

1. Modelagem do Banco de Dados

Utilizamos o **DB Designer Fork** para criar o modelo de dados. O sistema contém as seguintes entidades principais:

- **Usuario:** Identifica os operadores do sistema.
- **Pessoa:** Superclasse para representar tanto pessoas físicas quanto jurídicas.

- **PessoaFisica:** Armazena CPFs e relaciona-se com a entidade Pessoa.
- **PessoaJuridica:** Armazena CNPJs e relaciona-se com a entidade Pessoa.
- **Produto:** Contém informações sobre os produtos comercializados.
- **Movimento:** Registra entradas (compras) e saídas (vendas).
- **ItemMovimentado:** Relaciona Movimento e Produto, armazenando quantidades e valores.



2. Criação do Banco de Dados no SQL Server

A base de dados foi criada com o seguinte comando:

CREATE TABLE Usuario (

idUsuario INT IDENTITY(1,1) PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

senha VARCHAR(100) NOT NULL

);

CREATE TABLE Pessoa (

```
idPessoa INT PRIMARY KEY,  
  
nome VARCHAR(100) NOT NULL,  
  
endereco VARCHAR(255) NOT NULL,  
  
telefone VARCHAR(20) NOT NULL  
  
);  
  
CREATE TABLE PessoaFisica (  
  
    idPessoa INT PRIMARY KEY,  
  
    cpf VARCHAR(14) UNIQUE NOT NULL,  
  
    FOREIGN KEY (idPessoa) REFERENCES Pessoa(idPessoa)  
  
);  
  
CREATE TABLE PessoaJuridica (  
  
    idPessoa INT PRIMARY KEY,  
  
    cnpj VARCHAR(18) UNIQUE NOT NULL,  
  
    FOREIGN KEY (idPessoa) REFERENCES Pessoa(idPessoa)  
  
);  
  
CREATE TABLE Produto (  
  
    idProduto INT IDENTITY(1,1) PRIMARY KEY,  
  
    nome VARCHAR(100) NOT NULL,  
  
    quantidade INT NOT NULL,  
  
    precoVenda DECIMAL(10,2) NOT NULL  
  
);  
  
CREATE TABLE Movimento (  
  
    idMovimento INT IDENTITY(1,1) PRIMARY KEY,  
  
    idUsuario INT NOT NULL,  
  
    idPessoa INT NOT NULL,  
  
    tipoMovimento CHAR(1) CHECK (tipoMovimento IN ('E', 'S')),  
  
    valorTotal DECIMAL(10,2) NOT NULL,
```

```
FOREIGN KEY (idUserario) REFERENCES Usuario(idUsuario),  
  
FOREIGN KEY (idPessoa) REFERENCES Pessoa(idPessoa)  
  
);  
  
CREATE TABLE ItemMovimentado (  
  
    idItemMovimentado INT IDENTITY(1,1) PRIMARY KEY,  
  
    idMovimento INT NOT NULL,  
  
    idProduto INT NOT NULL,  
  
    quantidade INT NOT NULL,  
  
    precoUnitario DECIMAL(10,2) NOT NULL,  
  
    FOREIGN KEY (idMovimento) REFERENCES Movimento(idMovimento),  
  
    FOREIGN KEY (idProduto) REFERENCES Produto(idProduto)  
  
);
```

3. Testes e Resultados

Para validar a estrutura, realizamos as seguintes operações:

- Inserção de usuários (operadores)

```
SELECT * FROM Usuario;
```

100 %

 Resultados  Mensagens

	idUsuario	nome	email	senha
1	1	Administrador	admin@empresa.com	senha123
2	2	Operador1	op1@empresa.com	senha456
3	3	Operador2	op2@empresa.com	senha789

- Inserção de produtos

```
SELECT * FROM produto;
```

100 %



Resultados



Mensagens

	idProduto	nome	quantidade	precoVenda
1	1	Notebook	10	3500.00
2	2	Mouse	50	45.00
3	3	Teclado	30	120.00
4	4	Monitor	15	800.00
5	5	Impressora	10	1200.00
6	6	Cadeira Gamer	5	950.00
7	7	Headset	20	300.00
8	8	Smartphone	15	2500.00
9	9	Tablet	12	1800.00

- Inserção de clientes (pessoas físicas e jurídicas)

```
FROM Pessoa P
JOIN PessoaFisica PF ON P.idPessoa = PF.idPessoa;

SELECT P.idPessoa, P.nome, PJ.cnpj
FROM Pessoa P
JOIN PessoaJuridica PJ ON P.idPessoa = PJ.idPessoa;
```

100 %

Resultados Mensagens

	idPessoa	nome	cpf
1	1	João Silva	12345678901
2	5	Carlos Almeida	22233344455
3	3	Maria Oliveira	98765432100

	idPessoa	nome	cnpj
1	2	Empresa X	11222333444455
2	4	Empresa Y	55667788990011
3	6	Tech Solutions LTDA	99887766554433

- Registros de movimentações (entradas e saídas de produtos)

```

SELECT M.idMovimento, U.nome AS Usuario, P.nome AS Pessoa,
       M.tipoMovimento, M.dataMovimento, M.valorTotal
FROM Movimento M
JOIN Usuario U ON M.idUsuario = U.idUsuario
JOIN Pessoa P ON M.idPessoa = P.idPessoa;

```

100 %

Resultados

Mensagens

	idMovimento	Usuario	Pessoa	tipoMovimento	dataMovimento	valorTotal
1	14	Administrador	Empresa X	E	2025-03-30 16:36:49.313	14000.00
2	15	Operador1	João Silva	S	2025-03-30 16:36:49.323	7000.00
3	16	Administrador	Empresa X	E	2025-03-30 16:37:41.400	2250.00
4	17	Operador1	João Silva	S	2025-03-30 16:37:41.423	7000.00
5	18	Administrador	Empresa X	E	2025-03-30 16:37:41.427	17500.00
6	19	Operador1	João Silva	S	2025-03-30 16:37:41.427	900.00
7	20	Administrador	Empresa X	E	2025-03-30 16:37:41.430	1500.00
8	21	Operador1	João Silva	S	2025-03-30 16:37:41.430	450.00
9	22	Administrador	Empresa X	E	2025-03-30 16:37:41.430	6000.00
10	23	Operador1	João Silva	S	2025-03-30 16:37:41.430	2400.00
11	24	Administrador	Empresa X	E	2025-03-30 16:37:41.430	5000.00
12	25	Operador1	João Silva	S	2025-03-30 16:37:41.433	2000.00
13	26	Operador2	Empresa X	E	2025-03-30 16:44:02.233	7000.00
14	27	Operador2	João Silva	S	2025-03-30 16:44:11.000	3500.00
15	28	Operador2	Empresa X	E	2025-03-30 16:44:22.190	450.00
16	29	Operador2	João Silva	S	2025-03-30 16:44:22.207	135.00
17	30	Operador2	Empresa X	E	2025-03-30 16:46:30.773	250.00

Os resultados esperados foram obtidos com sucesso, garantindo a correta relação entre as tabelas e integridade referencial.

Análise e Conclusão

Como são implementadas as diferentes cardinalidades?

- **1x1 (Pessoa - PessoaFisica/PessoaJuridica):** Implementado por meio de chaves primárias compartilhadas.
- **1xN (Usuário - Movimento):** Cada operador pode realizar várias movimentações.

- **NxN (Produto - Movimento):** Implementado pela tabela intermediária ItemMovimentado.

Que tipo de relacionamento deve ser utilizado para representar a herança?

No SQL Server, a herança foi implementada por meio de tabelas separadas (PessoaFisica e PessoaJuridica), cada uma referenciando a tabela principal Pessoa.

Como o SQL Server Management Studio melhora a produtividade?

O SSMS melhora a produtividade ao oferecer:

- Interface gráfica para gerenciamento de tabelas, consultas e índices.
- Ferramentas de modelagem e visualização do banco de dados.
- Facilidade para execução de scripts e otimização de consultas.

Repositório Git

O projeto está armazenado no seguinte repositório Git: [🌐 GitHub - Elitonr65/banco_estoque](https://github.com/Elitonr65/banco_estoque)