

Aluno: Altair Silva Filho

Matrícula: 202403254409

Unidade: Palhoça – SC

Curso: Desenvolvimento Full Stack

Disciplina: Vamos manter as informações?

Período: 2025.1

Relatório da Missão Prática – Nível 2 – Mundo 3

Vamos Manter as Informações?

1º Procedimento | Criando o Banco de Dados

Objetivo

O projeto visa criar um modelo de negócios para uma loja, onde operadores, que serão usuários do programa, podem executar operações de compra e venda de produtos, sendo para isto necessária a criação de banco de dados para as entidades envolvidas no processo.

Neste procedimento, o objetivo é criar uma ferramenta para a modelagem para a base de dados relacionais e o banco de dados utilizando como base o Server Management Studio.

Códigos do Projeto

Diagrama Visual (DER)

Modelo Loja.png

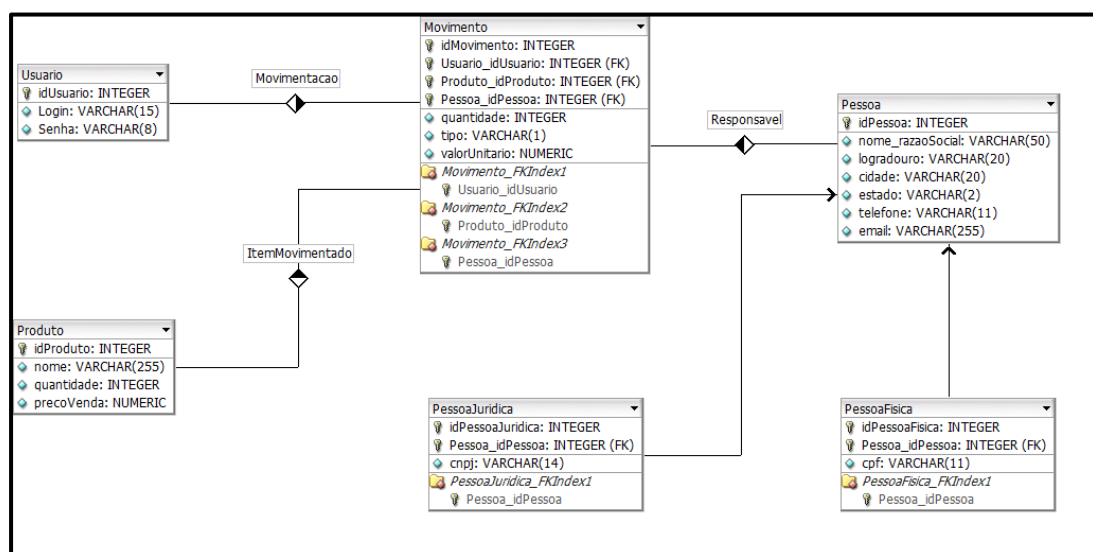


Figura 1 - Diagrama Visual

Script para a criação do Banco de Dados

Banco Loja.sql

```
CREATE SEQUENCE Seq_Id_TablePessoa  
AS INT  
START WITH 1  
INCREMENT BY 1;
```

```
CREATE TABLE Produto (  
  idProduto INTEGER NOT NULL ,  
  nome VARCHAR(255) ,  
  quantidade INTEGER ,  
  precoVenda NUMERIC ,  
  PRIMARY KEY(idProduto));
```

```
CREATE TABLE Pessoa (  
  idPessoa INTEGER NOT NULL ,  
  nome_razaoSocial VARCHAR(50) NOT NULL ,  
  logradouro VARCHAR(20) NOT NULL ,  
  cidade VARCHAR(20) NOT NULL ,  
  estado VARCHAR(2) NOT NULL ,  
  telefone VARCHAR(11) NOT NULL ,  
  email VARCHAR(255) NOT NULL ,  
  PRIMARY KEY(idPessoa));
```

```
CREATE TABLE Usuario (  
  idUsuario INTEGER NOT NULL ,  
  Login VARCHAR(15) ,  
  Senha VARCHAR(8) ,  
  PRIMARY KEY(idUsuario));
```

```
CREATE TABLE PessoaJuridica (  
  idPessoaJuridica INTEGER NOT NULL ,  
  Pessoa_idPessoa INTEGER NOT NULL ,  
  cnpj VARCHAR(14) ,  
  PRIMARY KEY(idPessoaJuridica, Pessoa_idPessoa) ,  
  FOREIGN KEY(Pessoa_idPessoa)  
  REFERENCES Pessoa(idPessoa));
```

```
CREATE INDEX PessoaJuridica_FKIndex1 ON PessoaJuridica (Pessoa_idPessoa);
```

```
CREATE INDEX IFK_Rel_09 ON PessoaJuridica (Pessoa_idPessoa);
```

```
CREATE TABLE PessoaFisica (  
  idPessoaFisica INTEGER NOT NULL ,  
  Pessoa_idPessoa INTEGER NOT NULL ,  
  cpf VARCHAR(11) ,  
  PRIMARY KEY(idPessoaFisica, Pessoa_idPessoa) ,  
  FOREIGN KEY(Pessoa_idPessoa)  
    REFERENCES Pessoa(idPessoa));
```

```
CREATE INDEX PessoaFisica_FKIndex1 ON PessoaFisica (Pessoa_idPessoa);
```

```
CREATE INDEX IFK_Rel_08 ON PessoaFisica (Pessoa_idPessoa);
```

```
CREATE TABLE Movimento (  
  idMovimento INTEGER NOT NULL ,  
  Usuario_idUsuario INTEGER NOT NULL ,  
  Produto_idProduto INTEGER NOT NULL ,  
  Pessoa_idPessoa INTEGER NOT NULL ,  
  quantidade INTEGER ,  
  tipo VARCHAR(1) ,  
  valorUnitario NUMERIC ,  
  PRIMARY KEY(idMovimento, Usuario_idUsuario, Produto_idProduto, Pessoa_idPessoa) ,  
  FOREIGN KEY(Usuario_idUsuario)  
    REFERENCES Usuario(idUsuario),  
  FOREIGN KEY(Produto_idProduto)  
    REFERENCES Produto(idProduto),  
  FOREIGN KEY(Pessoa_idPessoa)  
    REFERENCES Pessoa(idPessoa));
```

```
CREATE INDEX Movimento_FKIndex1 ON Movimento (Usuario_idUsuario);  
CREATE INDEX Movimento_FKIndex2 ON Movimento (Produto_idProduto);  
CREATE INDEX Movimento_FKIndex3 ON Movimento (Pessoa_idPessoa);
```

```
CREATE INDEX IFK_Movimentacao ON Movimento (Usuario_idUsuario);  
CREATE INDEX IFK_ItemMovimentado ON Movimento (Produto_idProduto);  
CREATE INDEX IFK_Responsavel ON Movimento (Pessoa_idPessoa);
```

Resultados da execução dos códigos

A janela do resultado da criação do Banco de Dados no SQL Server Management é mostrada abaixo.

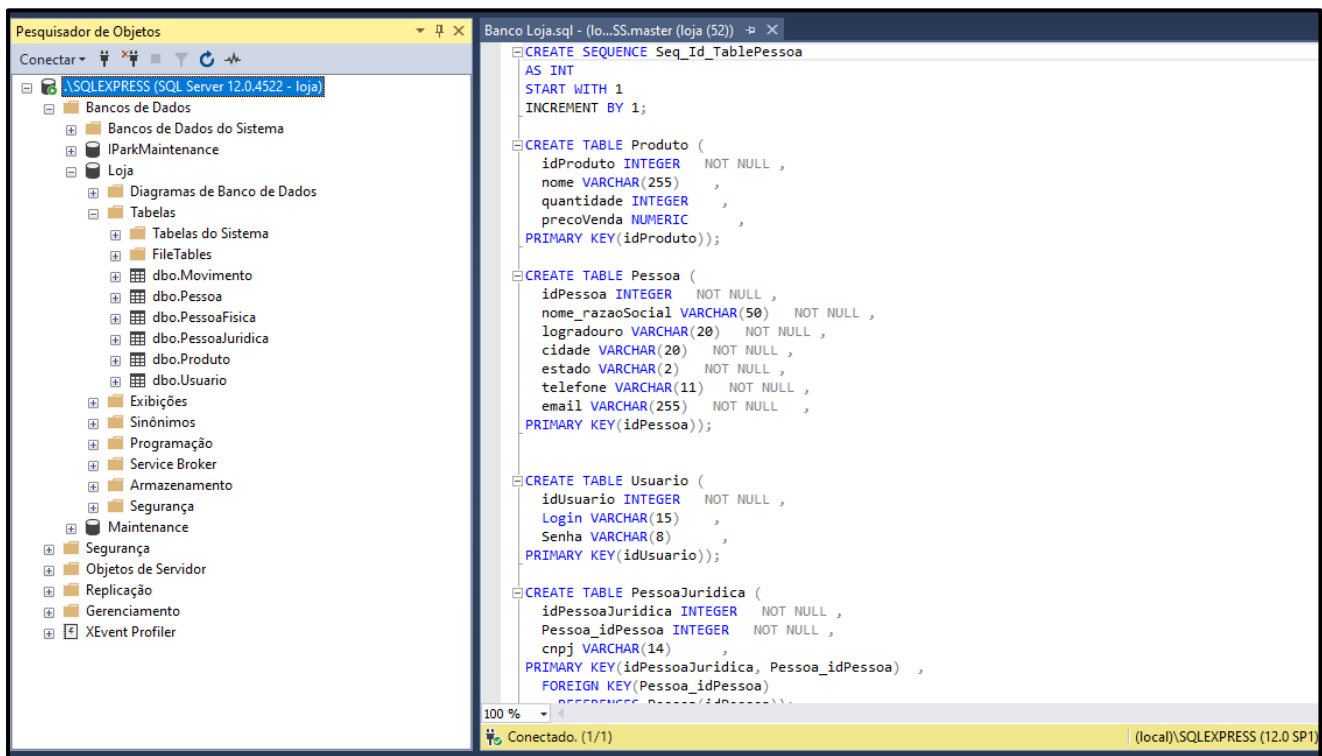


Figura 2 - Janela do SQL com o resultado da criação do BD

Análise e Conclusão

A missão mostrou como se pode implementar as diferentes cardinalidades. No projeto foi feito o uso das cardinalidades 1 x 1 e 1 x N. A cardinalidade 1 x 1 pode ser vista, por exemplo, na relação das tabelas “Pessoa” com “PessoaFisica”. A cardinalidade 1 x N foi utilizada nas relações entre as demais tabelas, como mostra a figura1. Elas foram implementadas através de chaves primárias (PK) e chaves estrangeiras (FK).

O projeto também mostrou como criar métodos para mapear a relação de herança no banco. Isto foi feito nas relações entre a Tabela “Pessoa” com as tabelas especializadas “PessoaFisica” e “PessoaJuridica”, que herdam características comuns a todas as pessoas, como nome, endereço, telefone e e-mail.

Por fim, podemos ver como o Server Management Studio permite a melhoria da produtividade nas tarefas relacionadas ao gerenciamento do banco de dados. A possibilidade de interação do programa com a ferramenta de modelagem, torna o desenvolvimento muito mais rápido e prático, minimizando os erros.

Armazenado no Github: <https://github.com/altairsf/MissaoN2M3.git>