

#### CONTROLE DE ESTOQUE

A empresa *Smart Software* está desenvolvendo uma aplicação de Controle de Estoque para seus clientes, os quais se caracterizam como microempresas.

O sistema não vai emitir cupom fiscal, embora ele gerencie entradas e saídas. No levantamento de requisitos, foram definidas as seguintes Entidades:

- USUARIOS – quem utiliza o sistema. O *tipoDeUsuario* pode ser 0-normal, 1-gerente, 2-admin.
- CLIENTES – quem compra produtos da empresa
- SAIDAS – armazena as NF de saída
- ITENS\_SAIDA – os produtos das NF de saída
- FORNECEDORES – de quem a empresa compra produtos para seu estoque
- ENTRADAS – armazena as NF de entrada
- ITENS\_ENTRADA – os produtos das NF de entrada
- PRODUTOS – catálogo de produtos e quantidade em estoque
- CATEGORIAS – classificação dos produtos.

São restrições do sistema:

- Toda vez que um produto entra ou sai do estoque, a *quantidadeEstoque* deve ser atualizada na Entidade PRODUTOS;
- A cada entrada, o *precoCusto* deve ser atualizado considerando o valor mais alto entre *precoCusto* (atual) e *valorUnitario* (da entrada).
- A cada entrada, o *precoVenda* deve ser atualizado considerando a fórmula  $precoVenda = precoCusto * 2.1$ .
- Demais restrições são representadas pelas cardinalidades apresentadas no Diagrama Conceitual (Figura 1).

A Figura 1 apresenta o Diagrama Conceitual para o banco de dados do Controle de Estoque.

A Figura 2 apresenta o Diagrama Lógico para o banco de dados do Controle de Estoque.

O Quadro 1 apresenta o código SQL DDL para implementar o BD do Controle de Estoque.

# BANCO DE DADOS II

## Estudo de Caso

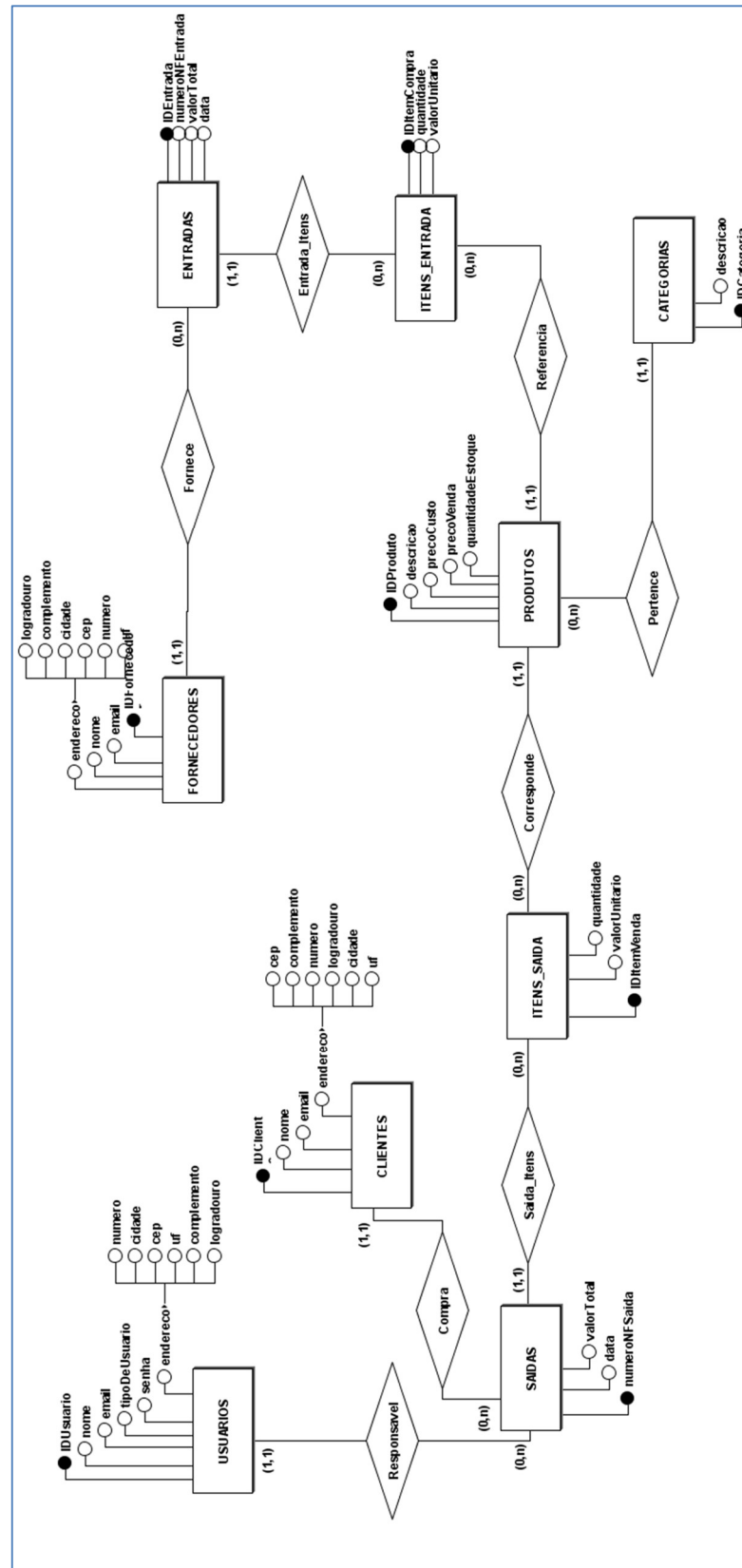


Figura 1 – Diagrama Conceitual do banco de dados para o Controle de Estoque.

## BANCO DE DADOS II

### Estudo de Caso

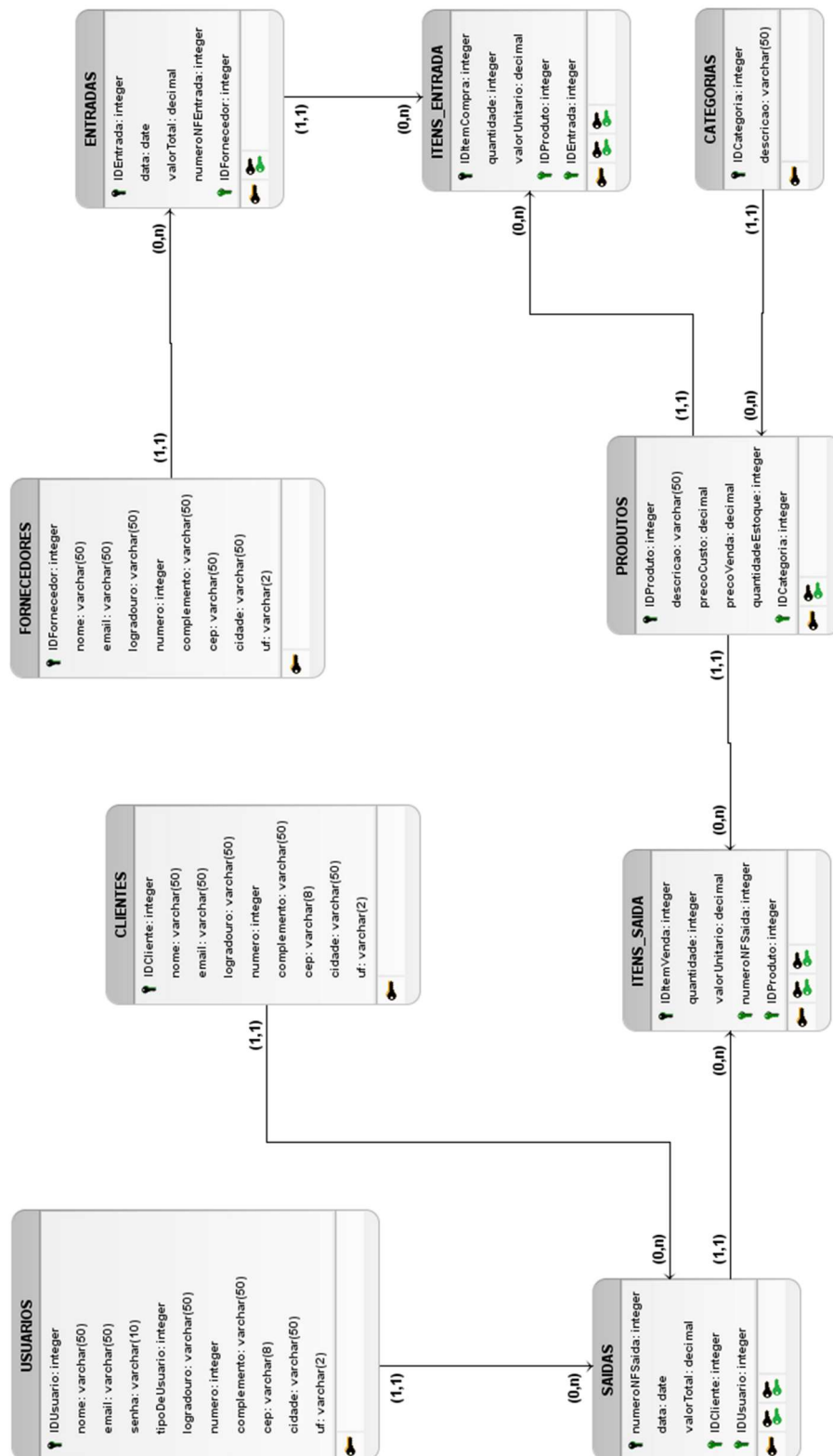


Figura 2 – Diagrama Lógico do banco de dados para o Controle de Estoque.

**Quadro 1 – Código SQL DDL para criação do BD do Controle de Estoque**

```
/* Sistema do Controle de Estoque
Projeto Físico do Banco de Dados
Versão 1.0
Autor: Angelo Augusto Frozza
*/

/*
TABELAS PRIMÁRIAS
- Contém apenas Chave Primária (PK)
*/

CREATE TABLE usuarios (
    IDUsuario SERIAL PRIMARY KEY, -- Definição da PK
    nome VARCHAR(50) NOT NULL, -- Campos obrigatórios
    email VARCHAR(50) NOT NULL,
    senha VARCHAR(10) NOT NULL,
    tipoDeUsuario INTEGER NOT NULL DEFAULT 0, -- Usuários: 0-normal (DEFAULT),
                                           -- 1-gerente, 2-admin
    logradouro VARCHAR(50), -- Campos opcionais
    numero INTEGER,
    complemento VARCHAR(50),
    cep VARCHAR(8),
    cidade VARCHAR(50),
    uf VARCHAR(2)
);

CREATE TABLE fornecedores (
    IDFornecedor SERIAL PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL,
    logradouro VARCHAR(50),
    numero INTEGER,
    complemento VARCHAR(50),
    cep VARCHAR(50),
    cidade VARCHAR(50),
    uf VARCHAR(2)
);

CREATE TABLE clientes (
    IDCliente SERIAL PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL,
    logradouro VARCHAR(50) NOT NULL,
    numero VARCHAR(50) NOT NULL,
    complemento VARCHAR(50) NOT NULL,
    cep VARCHAR(8) NOT NULL,
    cidade VARCHAR(50) NOT NULL,
    uf VARCHAR(2) NOT NULL
);

CREATE TABLE categorias (
    IDCategoria SERIAL PRIMARY KEY,
    descricao VARCHAR(50) NOT NULL
);
```

```
/*
TABELAS SECUNDÁRIAS
- Contem Chaves Estrangeiras (FK) para os relacionamentos
*/

CREATE TABLE produtos (
    IDProduto SERIAL PRIMARY KEY,
    Descricao VARCHAR(50) NOT NULL,
    PrecoCusto DECIMAL DEFAULT 0, -- Inicialização de valores no cadastro
    precoVenda DECIMAL DEFAULT 0,
    quantidadeEstoque INTEGER DEFAULT 0,
    IDCategoria INTEGER NOT NULL, -- Campo da FK é obrigatório
    FOREIGN KEY (IDCategoria) -- Definição da FK
        REFERENCES CATEGORIAS (IDCategoria)
        ON DELETE CASCADE
);

CREATE TABLE entradas (
    numeroNFEntrada SERIAL PRIMARY KEY,
    data DATE NOT NULL,
    valorTotal DECIMAL,
    IDFornecedor INTEGER NOT NULL,
    FOREIGN KEY (IDFornecedor)
        REFERENCES FORNECEDORES (IDFornecedor)
        ON DELETE CASCADE
);

CREATE TABLE saidas (
    numeroNFSaida SERIAL PRIMARY KEY,
    data DATE NOT NULL,
    valorTotal DECIMAL NOT NULL,
    IDCliente INTEGER NOT NULL,
    IDUsuario INTEGER NOT NULL,
    FOREIGN KEY (IDCliente)
        REFERENCES CLIENTES (IDCliente)
        ON DELETE CASCADE,
    FOREIGN KEY (IDUsuario)
        REFERENCES USUARIOS (IDUsuario)
        ON DELETE CASCADE
);

CREATE TABLE itens_saida (
    IDItemVenda SERIAL PRIMARY KEY,
    quantidade INTEGER NOT NULL,
    valorUnitario DECIMAL NOT NULL,
    numeroNFSaida INTEGER NOT NULL,
    IDProduto INTEGER NOT NULL,
    FOREIGN KEY (numeroNFSaida)
        REFERENCES SAIDAS (numeroNFSaida)
        ON DELETE CASCADE,
    FOREIGN KEY (IDProduto)
        REFERENCES PRODUTOS (IDProduto)
        ON DELETE CASCADE
);
```

## BANCO DE DADOS II

### Estudo de Caso

---

```
CREATE TABLE itens_entrada (  
    IDItemCompra      SERIAL PRIMARY KEY,  
    quantidade         INTEGER NOT NULL,  
    valorUnitario      DECIMAL NOT NULL,  
    IDProduto          INTEGER NOT NULL,  
    numeroNFEntrada    INTEGER NOT NULL,  
    FOREIGN KEY (IDProduto)  
        REFERENCES PRODUTOS (IDProduto)  
        ON DELETE CASCADE,  
    FOREIGN KEY (numeroNFEntrada)  
        REFERENCES ENTRADAS (numeroNFEntrada)  
        ON DELETE CASCADE  
);  
  
/*  
TABELAS TERCIÁRIAS  
- Implementam (FK) os relacionamentos n:n  
  
O ESTUDO DE CASO DO CONTROLE DE ESTOQUE NÃO POSSUI RELACIONAMENTOS N:N  
*/
```