

# Documento de Arquitetura do Banco de Dados – Projeto Gestor Simplificado

Versão: 1.1

Data: 25 de setembro de 2025

Última atualização em: 08 de outubro de 2025

Autor: Felipe da Silva

## 1. Visão Geral

Este documento descreve a estrutura do banco de dados para a aplicação "Gestor Simplificado". O objetivo é criar um modelo de dados que suporte todas as funcionalidades planejadas, como autenticação de usuários, gestão de produtos, registro de vendas e despesas, e a geração de relatórios financeiros. A tecnologia base é o PostgreSQL.

## 2. Diagrama de Entidade-Relacionamento (ERD)

O diagrama abaixo ilustra as entidades principais e seus relacionamentos.

```
USER {  
    int id PK  
    string name  
    string email UK  
    string password  
}
```

```
PRODUCT {  
    int id PK  
    string name  
    string description  
    decimal price  
    int userId FK  
}
```

```
EXPENSE {  
    int id PK  
    string description  
    string category
```

```
    decimal amount
    datetime date
    int userId FK
}
```

```
SALE {
    int id PK
    decimal totalAmount
    datetime saleDate
    int userId FK
}
```

```
SaleItem {
    int id PK
    int quantity
    decimal priceAtSale
    int saleId FK
    int productId FK
}
```

```
USER ||--o{ PRODUCT : "possui"
USER ||--o{ EXPENSE : "registra"
USER ||--o{ SALE : "realiza"
SALE ||--|{ SaleItem : "contém"
PRODUCT ||--|{ SaleItem : "é vendido em"
```

---

### 3. Definição das Entidades (Tabelas)

A seguir, detalhamos cada uma das tabelas, seus campos e regras.

### Tabela: User

Armazena as informações dos usuários que utilizam o sistema. Cada usuário é "dono" de seus próprios dados.

Atributo	Tipo de Dado	Descrição / Regras	Exemplo
id	INTEGER	Chave Primária (PK), Auto-incremento.	1
name	VARCHAR(255)	Nome do usuário. (Opcional)	"Felipe Alves"
email	VARCHAR(255)	E-mail de login. Deve ser Único (UK).	"felipe@email.com"
password	VARCHAR(255)	Senha do usuário. <b>Deve ser armazenada com hash.</b>	"\$2b\$10\$..."
createdAt	DATETIME	Data e hora de criação do registro. Automático.	"2025-09-25 15:30:00"
updatedAt	DATETIME	Data e hora da última atualização. Automático.	"2025-09-25 15:30:00"

### Tabela: Product

Armazena os produtos ou serviços que um usuário cadastra para vender.

Atributo	Tipo de Dado	Descrição / Regras	Exemplo
id	INTEGER	Chave Primária (PK), Auto-incremento.	101
name	VARCHAR(255)	Nome do produto. (Obrigatório)	"Consultoria de Software"
description	TEXT	Descrição detalhada do produto. (Opcional)	"1 hora de consultoria..."
price	DECIMAL(10, 2)	Preço padrão do produto.	150.00
userId	INTEGER	Chave Estrangeira (FK) que referencia User(id).	1
createdAt	DATETIME	Data e hora de criação.	"2025-09-25 15:35:00"
updatedAt	DATETIME	Data e hora da última atualização.	"2025-09-25 15:35:00"

### Tabela: Expense

Armazena os registros de despesas de um usuário.

Atributo	Tipo de Dado	Descrição / Regras	Exemplo
id	INTEGER	Chave Primária (PK), Auto-incremento.	201
description	VARCHAR(255)	Descrição da despesa. (Obrigatório)	"Aluguel do escritório"
category	VARCHAR(255)	Categoria para agrupar despesas.	"Custos Fixos"
amount	DECIMAL(10, 2)	Valor da despesa. (Obrigatório, > 0)	1200.50
date	DATETIME	Data em que a despesa ocorreu.	"2025-09-20 10:00:00"
userId	INTEGER	Chave Estrangeira (FK) que referencia User(id).	1
createdAt	DATETIME	Data e hora de criação.	"2025-09-25 15:40:00"

### Tabela: Sale

Armazena a "capa" de uma venda, como o valor total e a data. Os itens vendidos

ficam em outra tabela.

Atributo	Tipo de Dado	Descrição / Regras	Exemplo
id	INTEGER	Chave Primária (PK), Auto-incremento.	301
totalAmount	DECIMAL(10, 2)	Valor total da venda. (Calculado)	450.00
saleDate	DATETIME	Data e hora em que a venda foi realizada.	"2025-09-24 18:00:00"
userId	INTEGER	Chave Estrangeira (FK) que referencia User(id).	1
createdAt	DATETIME	Data e hora de criação.	"2025-09-25 15:45:00"

### **Tabela: SaleItem (Tabela de Junção)**

Esta tabela é o "coração" do relacionamento entre Vendas e Produtos. Ela resolve o

relacionamento **Muitos-para-Muitos**.

Atributo	Tipo de Dado	Descrição / Regras	Exemplo
id	INTEGER	Chave Primária (PK), Auto-incremento.	401
quantity	INTEGER	Quantidade do produto vendido nesta venda.	3
priceAtSale	DECIMAL(10, 2)	Preço do produto no momento da venda.	150.00
saleId	INTEGER	Chave Estrangeira (FK) que referencia Sale(id).	301
productId	INTEGER	Chave Estrangeira (FK) que referencia Product(id).	101

#### 4. Descrição dos Relacionamentos

- **User ↔ Product (Um-para-Muitos):** Um usuário pode ter muitos produtos cadastrados, mas cada produto pertence a um único usuário.
- **User ↔ Expense (Um-para-Muitos):** Um usuário pode registrar muitas despesas, mas cada despesa pertence a um único usuário.
- **User ↔ Sale (Um-para-Muitos):** Um usuário pode realizar muitas vendas, mas cada venda pertence a um único usuário.
- **Sale ↔ Product (Muitos-para-Muitos):** Este é o relacionamento mais complexo.
  - Uma venda pode conter muitos produtos diferentes.
  - Um produto pode estar presente em muitas vendas diferentes.
  - Para resolver isso, criamos a tabela de junção **SaleItem**. Ela conecta uma venda a um produto e armazena informações específicas dessa conexão, como a quantity vendida e o priceAtSale (para garantir a integridade histórica dos preços).