

Banco de Dados

Pontifícia Universidade Católica de Minas Gerais
Ciência da Computação

Trabalho Prático

Aluno: Otávio Augusto de Assis Ferreira Monteiro

Belo Horizonte
2025

1. Descrição Textual do Minimundo

A ONG "Comida que Abraça" atua no combate à vulnerabilidade social por meio da arrecadação e distribuição de doações, como alimentos não perecíveis, roupas e cobertores, para pessoas em situação de necessidade. Para otimizar suas operações e substituir métodos manuais como planilhas, que são propensos a erros e ineficiências, a organização necessita de um sistema de banco de dados. Este sistema permitirá um gerenciamento integrado e preciso de seus voluntários, doações, estoque e distribuições, garantindo maior transparência e eficácia em suas ações.

Para atender a essa necessidade, o sistema gerenciará um cadastro unificado de **Pessoas**, que serão identificadas por um **ID_Pessoa** único. Para cada pessoa, a ONG deve conhecer seu **CPF**, **Nome**, **Sobrenome**, **Telefone**, **Email**, **DataNasc** e a **DataCadastro** no sistema. Considerando que uma pessoa pode atuar como voluntária, o sistema também armazenará seu **StatusVoluntario** (Ativo/Inativo), sendo este campo nulo caso a pessoa seja exclusivamente uma doadora.

Além de pessoas físicas, a ONG colabora com **Instituições Parceiras**, que serão identificadas por um **ID_Instituicao** único. Para cada uma, é necessário conhecer o **NomeInstituicao**, **CNPJ**, um **Contato** principal, **Endereco** e **Telefone**.

As ações da ONG são frequentemente organizadas em **Campanhas**, identificadas por um **ID_Campanha**. Cada campanha possui um **NomeCampanha**, uma **Descricao** detalhada, uma **DataInicio** e um **Status** para controle (ex: Planejamento, Ativa, Finalizada). Uma Campanha pode ser apoiada ou patrocinada por uma **Instituicao_Parceira**.

Uma Pessoa pode participar como voluntária em múltiplas Campanhas ao longo do tempo, e uma Campanha, por sua vez, conta com múltiplos voluntários. Para gerenciar essa relação de muitos-para-muitos, cada participação é registrada na entidade **Voluntario_Campanha**, que vincula as duas tabelas através de suas

chaves estrangeiras: o **ID_Pessoa_Voluntario** (referenciando a pessoa) e o **ID_Campanha** (referenciando a campanha). Adicionalmente, especifica-se a **Funcao** desempenhada pelo voluntário naquele evento específico.

O recebimento de recursos é gerenciado pela entidade **Doacoes**. Cada doação recebe um **ID_Doacao** único e pode ser feita por uma **Pessoa** ou por uma **Instituicao_Parceira**. Para cada doação, o sistema registra seu **TipoDoacao** (Dinheiro, Item), o **ValorDoadado** (se aplicável), uma **Descricao** e a **DataDoacao**. Uma doação pode ser destinada a uma **Campanha** específica ou ser uma doação geral para a ONG, neste caso, o vínculo com a campanha é nulo.

Se a doação for de um item, ela estará obrigatoriamente vinculada a um **Produto** pré-cadastrado e terá sua **QuantidadeDoadada** registrada. A ONG mantém um catálogo de **Produtos** que pode receber e distribuir, cada um identificado por um **ID_Produto** e com **NomeProduto**, **Descricao**, **UnidadeMedida** (kg, unidade, etc.) e **Categoria**. O controle de inventário é feito pela entidade **Estoque**. Cada registro de estoque é identificado por um **ID_Estoque** único e está obrigatoriamente vinculado a um **Produto** através do **ID_Produto** (FK). Nele, são armazenados a **QuantidadeAtual** daquele item e a **DataUltimaAtualizacao** do registro.

Para a gestão financeira, as **Despesas** da ONG são registradas, cada uma com seu **ID_Despesa**, **Descricao**, **Valor**, **DataDespesa** e o **Fornecedor**. Uma despesa pode ser associada a uma **Campanha** para controle de custos de eventos específicos.

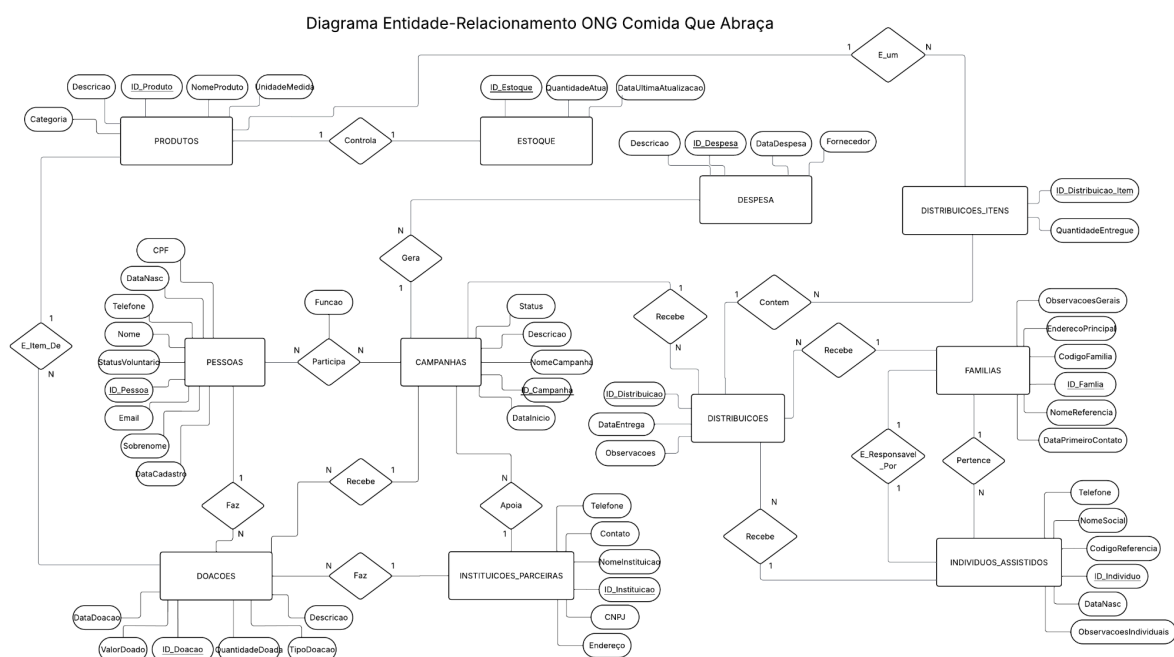
A assistência é direcionada a **Familias** e **Individuos_Assistidos**. Cada **Familia** é identificada por um **ID_Familia** e possui um **CodigoFamilia** de referência, um **NomeReferencia**, a **DataPrimeiroContato**, seu **EnderecoPrincipal** e **ObservacoesGerais**. Cada **Familia** é obrigatoriamente representada por um **Individuo_Assistido**, que é o responsável. Cada **Individuo_Assistido**, por sua vez, possui um **ID_Individuo** único, um

CodigoReferencia, **NomeSocial**, **Telefone** (se houver), **DataNasc** e **ObservacoesIndividuais**. Um indivíduo pode ou não pertencer a uma **Familia** já cadastrada.

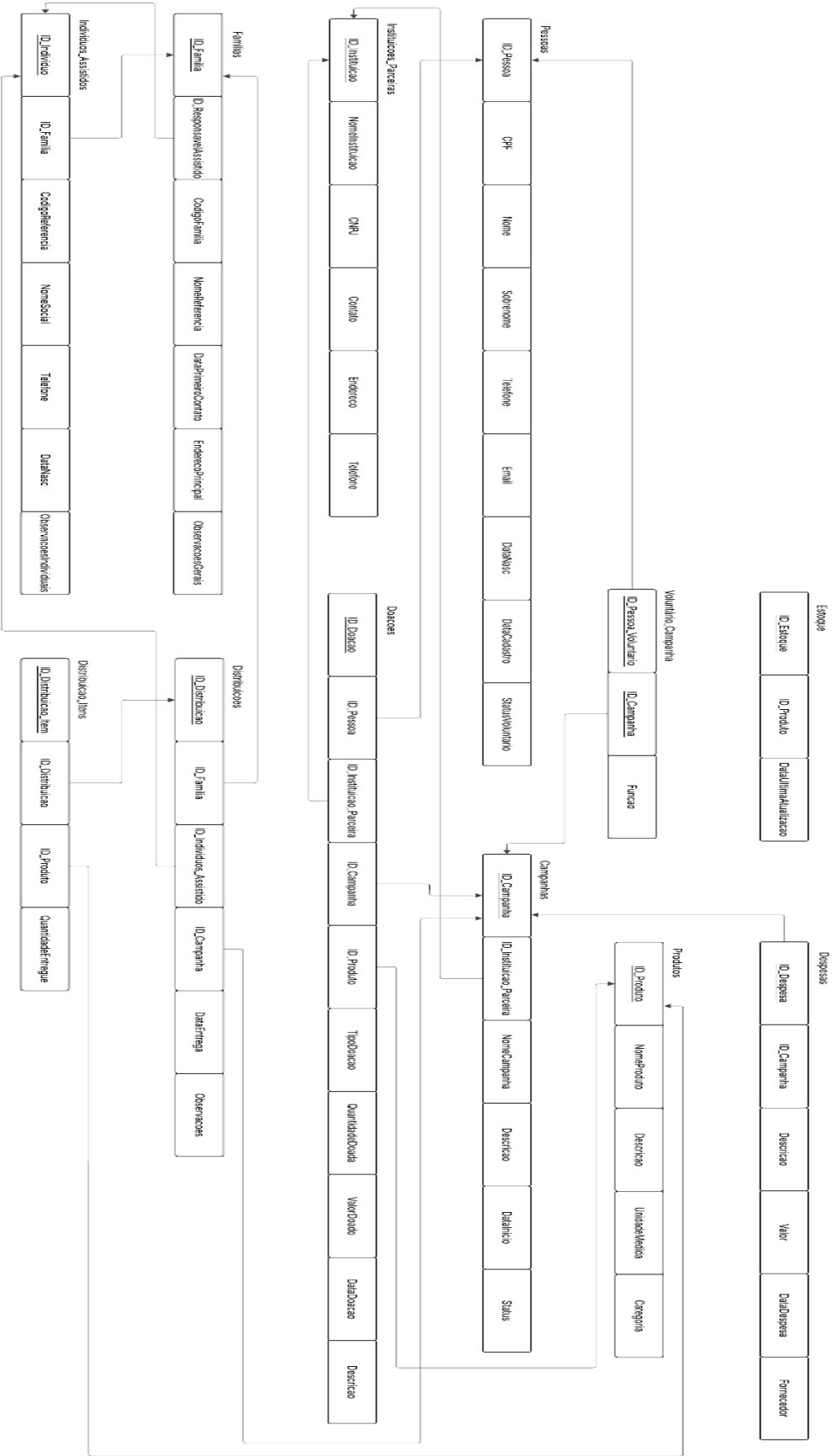
A transação de entrega de doações é registrada em **Distribuicoes**. Cada evento de distribuição é único, identificado por um **ID_Distribuicao**, e registra a **DataEntrega**, **Observacoes** e a quem se destina (uma **Familia** ou um **Individuo_Assistido**), podendo também estar vinculada a uma **Campanha**. O conteúdo de cada entrega é detalhado em **Distribuicao_Items**, cuja existência pressupõe a existência de uma Distribuição. Cada registro de item, identificado por seu próprio **ID_Distribuicao_Item**, representa um **Produto** entregue como parte de uma **Distribuicao**, especificando a **QuantidadeEntregue**

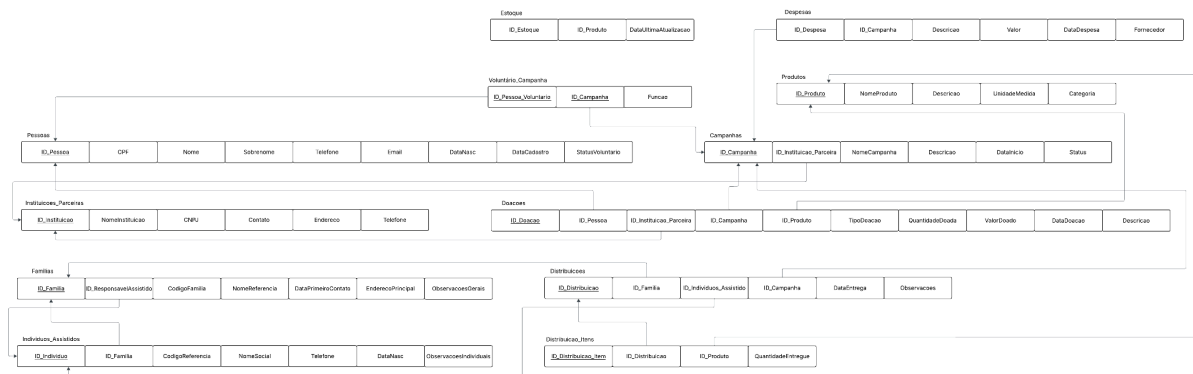
2. Diagramas

2.1. Modelo Conceitual: Diagrama Entidade-Relacionamento em Notação Peter Chen

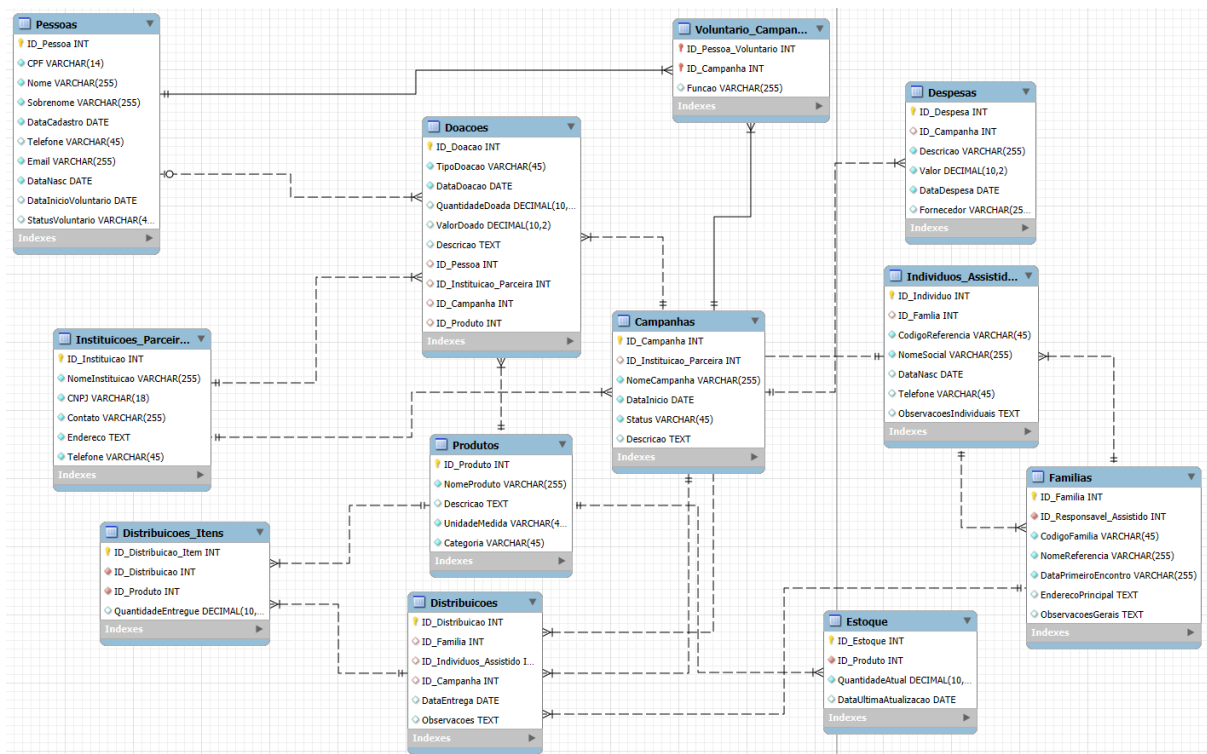


2.2. Modelo Lógico: Diagrama de Esquem





2.3. Modelo de Implementação: Diagrama de Implementação em Notação Pé-de-Galinha



3. Scripts SQL

3.1. Criação do Banco de Dados

```
CREATE DATABASE IF NOT EXISTS ong_comida_que_abraca;
USE ong_comida_que_abraca;
```

```
CREATE TABLE PESSOAS (
    ID_Pessoa INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
```

```

Nome VARCHAR(255) NOT NULL,
Sobrenome VARCHAR(255) NOT NULL,
DataCadastro DATETIME NOT NULL,
CPF VARCHAR(14),
Telefone VARCHAR(45),
Email VARCHAR(255),
DataNasc DATE,
DataInicioAtividade_Voluntario DATE,
Status_Voluntario VARCHAR(45)
);

```

```

CREATE TABLE INSTITUICOES_PARCEIRAS (
    ID_Instituicao INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    NomeInstituicao VARCHAR(255) NOT NULL,
    CNPJ VARCHAR(18),
    Contato VARCHAR(255),
    Endereco TEXT,
    Telefone VARCHAR(45)
);

```

```

CREATE TABLE CAMPANHAS (
    ID_Campanha INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    ID_Instituicao_Parceira INT,
    NomeCampanha VARCHAR(255) NOT NULL,
    DataInicio DATE NOT NULL,
    Status VARCHAR(45) NOT NULL,
    Descricao TEXT,
    FOREIGN KEY (ID_Instituicao_Parceira) REFERENCES
INSTITUICOES_PARCEIRAS(ID_Instituicao)
);

```

```

CREATE TABLE VOLUNTARIO_CAMPANHA (
    ID_Pessoa_Voluntario INT NOT NULL,
    ID_Campanha INT NOT NULL,

```



```
Funcao VARCHAR(255),  
PRIMARY KEY (ID_Pessoa_Voluntario, ID_Campanha),  
FOREIGN KEY (ID_Pessoa_Voluntario) REFERENCES PESSOAS(ID_Pessoa),  
FOREIGN KEY (ID_Campanha) REFERENCES CAMPANHAS(ID_Campanha)  
);
```

```
CREATE TABLE PRODUTOS (  
    ID_Produto INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    NomeProduto VARCHAR(255) NOT NULL,  
    UnidadeMedida VARCHAR(45) NOT NULL,  
    Descricao TEXT,  
    Categoria VARCHAR(45)  
);
```

```
CREATE TABLE ESTOQUE (  
    ID_Estoque INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    ID_Produto INT NOT NULL,  
    QuantidadeAtual DECIMAL(10, 2) NOT NULL,  
    DataUltimaAtualizacao DATETIME NOT NULL,  
    FOREIGN KEY (ID_Produto) REFERENCES PRODUTOS(ID_Produto)  
);
```

```
CREATE TABLE DOACOES (  
    ID_Doacao INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    ID_Pessoa INT,  
    ID_Instituicao_Parceira INT,  
    ID_Campanha INT,  
    ID_Produto INT,  
    TipoDoacao VARCHAR(45) NOT NULL,  
    DataDoacao DATETIME NOT NULL,  
    QuantidadeDoadada DECIMAL(10, 2),  
    ValorDoadado DECIMAL(10, 2),  
    Descricao TEXT,
```

```

FOREIGN KEY (ID_Pessoa) REFERENCES PESSOAS(ID_Pessoa),
        FOREIGN KEY (ID_Instituicao_Parceira) REFERENCES
INSTITUICOES_PARCEIRAS(ID_Instituicao),
FOREIGN KEY (ID_Campanha) REFERENCES CAMPANHAS(ID_Campanha),
FOREIGN KEY (ID_Produto) REFERENCES PRODUTOS(ID_Produto)
);

```

```

CREATE TABLE DESPESAS (
    ID_Despesa INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    ID_Campanha INT,
    Descricao VARCHAR(255) NOT NULL,
    Valor DECIMAL(10, 2) NOT NULL,
    DataDespesa DATE NOT NULL,
    Fornecedor VARCHAR(255),
    FOREIGN KEY (ID_Campanha) REFERENCES CAMPANHAS(ID_Campanha)
);

```

```

CREATE TABLE FAMILIAS (
    ID_Familia INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    ID_Responsavel_Assistido INT, -- Chave estrangeira será adicionada com ALTER
TABLE
    CodigoFamilia VARCHAR(45) NOT NULL,
    NomeReferencia VARCHAR(255) NOT NULL,
    DataPrimeiroContato DATE NOT NULL,
    EnderecoPrincipal TEXT,
    ObservacoesGerais TEXT
);

```

```

CREATE TABLE INDIVIDUOS_ASSISTIDOS (
    ID_Individuo INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    ID_Familia INT,
    CodigoReferencia VARCHAR(45) NOT NULL,
    NomeSocial VARCHAR(255) NOT NULL,

```

```
DataNasc DATE,  
Telefone VARCHAR(45),  
ObservacoesIndividuais TEXT,  
FOREIGN KEY (ID_Familia) REFERENCES FAMILIAS(ID_Familia)  
);
```

```
ALTER TABLE FAMILIAS ADD FOREIGN KEY (ID_Responsavel_Assistido)  
REFERENCES INDIVIDUOS_ASSISTIDOS(ID_Individuo);
```

```
CREATE TABLE DISTRIBUICOES (  
    ID_Distribuicao INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    ID_Familia INT,  
    ID_Individuo_Assistido INT,  
    ID_Campanha INT,  
    DataEntrega DATETIME NOT NULL,  
    Observacoes TEXT,  
    FOREIGN KEY (ID_Familia) REFERENCES FAMILIAS(ID_Familia),  
    FOREIGN KEY (ID_Individuo_Assistido) REFERENCES  
INDIVIDUOS_ASSISTIDOS(ID_Individuo),  
    FOREIGN KEY (ID_Campanha) REFERENCES CAMPANHAS(ID_Campanha)  
);
```

```
CREATE TABLE DISTRIBUICAO_ITENS (  
    ID_Distribuicao_Item INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    ID_Distribuicao INT NOT NULL,  
    ID_Produto INT NOT NULL,  
    QuantidadeEntregue DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (ID_Distribuicao) REFERENCES  
DISTRIBUICOES(ID_Distribuicao),  
    FOREIGN KEY (ID_Produto) REFERENCES PRODUTOS(ID_Produto)  
);
```

3.2. População do Banco de Dados

USE ong_comida_que_abraca;

```
INSERT INTO PRODUTOS (NomeProduto, UnidadeMedida, Categoria) VALUES
('Arroz Tipo 1', 'kg', 'Alimento Não Perecível'),
('Feijão Carioca', 'kg', 'Alimento Não Perecível'),
('Óleo de Soja', 'unidade', 'Alimento Não Perecível'),
('Cobertor de Casal', 'unidade', 'Vestuário e Cama'),
('Camisa de Algodão M', 'unidade', 'Vestuário e Cama');
```

```
INSERT INTO ESTOQUE (ID_Produto, QuantidadeAtual, DataUltimaAtualizacao)
VALUES
(1, 50.00, NOW()), -- Arroz
(2, 35.00, NOW()), -- Feijão
(3, 24.00, NOW()), -- Óleo
(4, 15.00, NOW()), -- Cobertor
(5, 30.00, NOW()); -- Camisa
```

```
INSERT INTO INSTITUICOES_PARCEIRAS (NomeInstituicao, CNPJ, Telefone)
VALUES
('Supermercados BH', '00.123.456/0001-00', '(31) 3333-4444'),
('Igreja da Boa Vontade', '11.222.333/0001-11', '(31) 98888-7777');
```

```
INSERT INTO PESSOAS (Nome, Sobrenome, DataCadastro, Email,
DataInicioAtividade_Voluntario, Status_Voluntario) VALUES
('Maria', 'Silva', NOW(), 'maria.s@email.com', '2024-01-10', 'Ativo'), -- Voluntária e
Doadora
('João', 'Pereira', NOW(), 'joao.p@email.com', NULL, NULL), -- Apenas Doador
```

('Ana', 'Souza', NOW(), 'ana.s@email.com', '2024-03-22', 'Ativo'), -- Apenas Voluntária

('Carlos', 'Gomes', NOW(), 'carlos.g@email.com', '2023-11-05', 'Inativo'); -- Voluntário Inativo

INSERT INTO CAMPANHAS (NomeCampanha, DataInicio, Status, Descricao, ID_Instituicao_Parceira) VALUES

('Inverno Solidário 2025', '2025-05-01', 'Ativa', 'Arrecadação de cobertores para o inverno.', 2),

('Natal Sem Fome 2025', '2025-11-01', 'Planejamento', 'Arrecadação de alimentos para cestas de Natal.', 1);

INSERT INTO VOLUNTARIO_CAMPANHA (ID_Pessoa_Voluntario, ID_Campanha, Funcao) VALUES

(1, 1, 'Coleta e Triagem'), -- Maria na campanha de inverno

(3, 1, 'Logística'); -- Ana na campanha de inverno

INSERT INTO DOACOES (ID_Pessoa, ID_Campanha, TipoDoacao, DataDoacao, ValorDoadado) VALUES

(2, 2, 'Dinheiro', NOW(), 150.00);

INSERT INTO DOACOES (ID_Instituicao_Parceira, ID_Produto, TipoDoacao, DataDoacao, QuantidadeDoadada) VALUES

(1, 1, 'Item', NOW(), 200.00); -- Supermercado doou 200kg de arroz

-- ATENÇÃO: Após um INSERT de doação de item, o ideal é ter um processo (Trigger ou na aplicação) para atualizar o ESTOQUE.

-- Ex: UPDATE ESTOQUE SET QuantidadeAtual = QuantidadeAtual + 200 WHERE ID_Produto = 1;

-- Primeiro a família, depois os indivíduos, e por fim atualiza o responsável.

```
INSERT INTO FAMILIAS (CodigoFamilia, NomeReferencia, DataPrimeiroContato)
VALUES ('FAM-001', 'Família Oliveira', '2025-02-15');
```

```
SET @last_fam_id = LAST_INSERT_ID();
```

```
INSERT INTO INDIVIDUOS_ASSISTIDOS (ID_Familia, CodigoReferencia,
NomeSocial, DataNasc) VALUES
```

```
(@last_fam_id, 'IND-001', 'Sandra Oliveira', '1985-06-20'),
```

```
(@last_fam_id, 'IND-002', 'Lucas Oliveira', '2010-09-10');
```

```
SET @resp_id = (SELECT ID_Individuo FROM INDIVIDUOS_ASSISTIDOS
WHERE NomeSocial = 'Sandra Oliveira');
```

```
UPDATE FAMILIAS SET ID_Responsavel_Assistido = @resp_id WHERE
ID_Familia = @last_fam_id;
```

```
INSERT INTO INDIVIDUOS_ASSISTIDOS (CodigoReferencia, NomeSocial)
VALUES ('IND-003', 'José das Neves');
```

```
INSERT INTO DISTRIBUICOES (ID_Familia, ID_Campanha, DataEntrega,
Observacoes) VALUES
```

```
(@last_fam_id, 1, NOW(), 'Entrega de cesta básica e cobertores.');
```

```
SET @last_dist_id = LAST_INSERT_ID();
```

```
INSERT INTO DISTRIBUICAO_ITENS (ID_Distribuicao, ID_Produto,
QuantidadeEntregue) VALUES
```

```
(@last_dist_id, 1, 5.00),
```

```
(@last_dist_id, 2, 2.00),
```

```
(@last_dist_id, 4, 2.00);
```

```
UPDATE ESTOQUE SET QuantidadeAtual = QuantidadeAtual - 5 WHERE  
ID_Produto = 1;
```