

Banco de dados - SEMEIA WEB

Equipe: Arthur Vinicius, Caio Sabino, Marcos Vinícius e Thauan Bezerra

Disciplina: Banco de Dados

Professor: Danilo Farias Soares da Silva

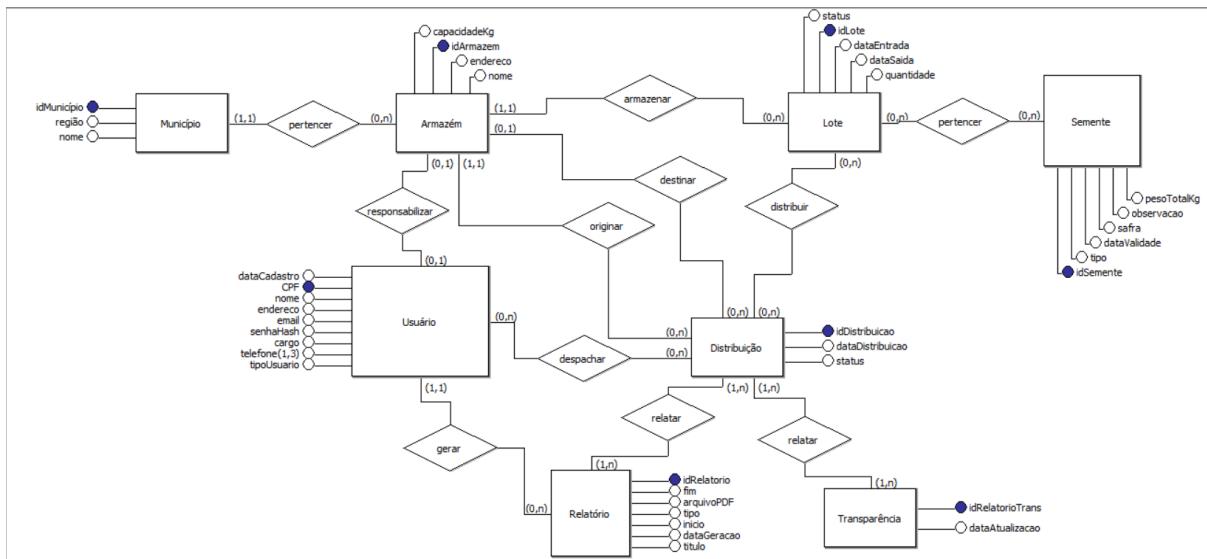
Instituição: Senac PE

O banco de dados do **Semeia Web** foi projetado para dar suporte ao controle, gestão e transparência do processo de distribuição de sementes em Pernambuco. Ele organiza entidades essenciais ao fluxo operacional, desde o cadastro de municípios e armazéns, até a gestão de lotes, distribuição, usuários e relatórios.

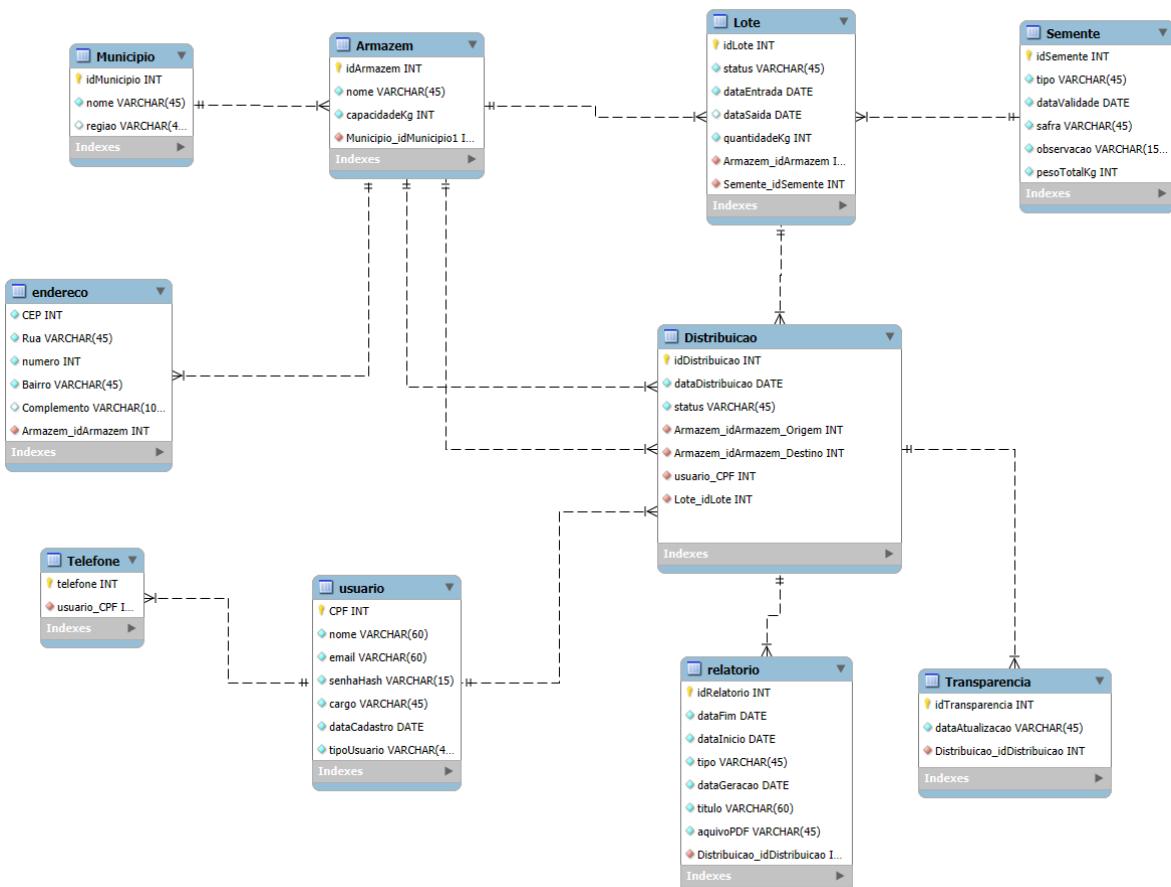
Entidades do Banco de Dados do Semeia Web

- **Município:** Guarda os municípios atendidos pelo programa, permitindo organizar a distribuição de sementes por região.
- **Armazém:** Representa os locais onde as sementes são armazenadas. Cada armazém está ligado a um município e recebe lotes de sementes.
- **Endereço:** Registra o endereço completo de cada armazém, garantindo localização precisa na plataforma.
- **Semente:** Define os tipos de sementes disponibilizadas pelo IPA, com informações necessárias para controle e rastreabilidade.
- **Lote:** Controla a entrada, saída e estoque de sementes em cada armazém. Cada lote representa uma quantidade específica de um tipo de semente.
- **Distribuição:** Registra as movimentações dos lotes entre armazéns e para os responsáveis pela distribuição, permitindo rastreamento completo.
- **Usuário:** Cadastra as pessoas que utilizam o sistema, como gestores, operadores e agentes responsáveis pelas distribuições.
- **Telefone:** Armazena números de telefone vinculados aos usuários, para facilitar contato e identificação.
- **Relatório:** Armazena relatórios gerados pelos usuários sobre atividades e distribuições realizadas.
- **Transparência:** Guarda informações disponibilizadas ao público, reforçando a transparência do processo de distribuição.

MODELAGEM ENTIDADE RELACIONAMENTO



MODELAGEM RELACIONAL



SCRIPT DE CRIAÇÃO - CREATE

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,N
O_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
CREATE SCHEMA IF NOT EXISTS `IPA` DEFAULT CHARACTER SET utf8 ;
USE `IPA` ;
```

```
CREATE TABLE IF NOT EXISTS `IPA`.`usuario` (
`CPF` VARCHAR(11) NOT NULL,
`nome` VARCHAR(60) NOT NULL,
`email` VARCHAR(60) NOT NULL,
`senhaHash` VARCHAR(32) NOT NULL,
`cargo` VARCHAR(45) NOT NULL,
`dataCadastro` DATE NOT NULL,
`tipoUsuario` VARCHAR(45) NOT NULL,
PRIMARY KEY (`CPF`));
```

```
CREATE TABLE IF NOT EXISTS `IPA`.`Municipio` (
`idMunicipio` INT NOT NULL,
`nome` VARCHAR(45) NOT NULL,
`regiao` VARCHAR(45) NULL,
PRIMARY KEY (`idMunicipio`));
```

```
CREATE TABLE IF NOT EXISTS `IPA`.`Armazem` (
`idArmazem` INT NOT NULL,
`nome` VARCHAR(45) NOT NULL,
`capacidadeKg` INT NOT NULL,
`Municipio_idMunicipio1` INT NOT NULL,
PRIMARY KEY (`idArmazem`),
UNIQUE INDEX `idArmazém_UNIQUE` (`idArmazem` ASC) VISIBLE,
INDEX `fk_Armazem_Municipio_idx` (`Municipio_idMunicipio1` ASC) VISIBLE,
CONSTRAINT `fk_Armazem_Municipio`
FOREIGN KEY (`Municipio_idMunicipio1`)
REFERENCES `IPA`.`Municipio` (`idMunicipio`)
ON DELETE NO ACTION
ON UPDATE NO ACTION);
```

```
CREATE TABLE IF NOT EXISTS `IPA`.`Semente` (
`idSemente` INT NOT NULL,
```

```

`tipo` VARCHAR(45) NOT NULL,
`dataValidade` DATE NOT NULL,
`safra` VARCHAR(45) NOT NULL,
`observacao` VARCHAR(150) NOT NULL,
`pesoTotalKg` INT NOT NULL,
PRIMARY KEY (`idSemente`));

```

```

CREATE TABLE IF NOT EXISTS `IPA`.`Lote` (
`idLote` INT NOT NULL,
`status` VARCHAR(45) NOT NULL,
`dataEntrada` DATE NOT NULL,
`dataSaida` DATE NULL,
`quantidadeKg` INT NOT NULL,
`Armazem_idArmazem` INT NOT NULL,
`Semente_idSemente` INT NOT NULL,
PRIMARY KEY (`idLote`),
INDEX `fk_Lote_Armazem1_idx` (`Armazem_idArmazem` ASC) VISIBLE,
INDEX `fk_Lote_Semente1_idx` (`Semente_idSemente` ASC) VISIBLE,
CONSTRAINT `fk_Lote_Armazem1`
FOREIGN KEY (`Armazem_idArmazem`)
REFERENCES `IPA`.`Armazem` (`idArmazem`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Lote_Semente1`
FOREIGN KEY (`Semente_idSemente`)
REFERENCES `IPA`.`Semente` (`idSemente`)
ON DELETE NO ACTION
ON UPDATE NO ACTION);

```

```

CREATE TABLE IF NOT EXISTS `IPA`.`Distribuicao` (
`idDistribuicao` INT NOT NULL,
`dataDistribuicao` DATE NOT NULL,
`status` VARCHAR(45) NOT NULL,
`Armazem_idArmazem_Origem` INT NOT NULL,
`Armazem_idArmazem_Destino` INT NOT NULL,
`usuario_CPF` VARCHAR(11) NOT NULL,
`Lote_idLote` INT NOT NULL,
PRIMARY KEY (`idDistribuicao`),
INDEX `fk_Distribuicao_Armazem1_idx` (`Armazem_idArmazem_Origem` ASC) VISIBLE,
INDEX `fk_Distribuicao_Armazem2_idx` (`Armazem_idArmazem_Destino` ASC) VISIBLE,
INDEX `fk_Distribuicao_usuario1_idx` (`usuario_CPF` ASC) VISIBLE,
INDEX `fk_Distribuicao_Lote1_idx` (`Lote_idLote` ASC) VISIBLE,
CONSTRAINT `fk_Distribuicao_Armazem1`
FOREIGN KEY (`Armazem_idArmazem_Origem`)
REFERENCES `IPA`.`Armazem` (`idArmazem`)
ON DELETE NO ACTION

```

```

ON UPDATE NO ACTION,
CONSTRAINT `fk_Distribuicao_Armazem2`
FOREIGN KEY (`Armazem_idArmazem_Destino`)
REFERENCES `IPA`.`Armazem` (`idArmazem`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Distribuicao_usuario1`
FOREIGN KEY (`usuario_CPF`)
REFERENCES `IPA`.`usuario` (`CPF`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Distribuicao_Lote1`
FOREIGN KEY (`Lote_idLote`)
REFERENCES `IPA`.`Lote` (`idLote`)
ON DELETE NO ACTION
ON UPDATE NO ACTION);

```

```

CREATE TABLE IF NOT EXISTS `IPA`.`relatorio` (
`idRelatorio` INT NOT NULL,
`dataFim` DATE NOT NULL,
`dataInicio` DATE NOT NULL,
`tipo` VARCHAR(45) NOT NULL,
`dataGeracao` DATE NOT NULL,
`titulo` VARCHAR(60) NOT NULL,
`arquivoPDF` VARCHAR(45) NOT NULL,
`Distribuicao_idDistribuicao` INT NOT NULL,
PRIMARY KEY (`idRelatorio`),
INDEX `fk_relatorio_Distribuicao1_idx` (`Distribuicao_idDistribuicao` ASC) VISIBLE,
CONSTRAINT `fk_relatorio_Distribuicao1`
FOREIGN KEY (`Distribuicao_idDistribuicao`)
REFERENCES `IPA`.`Distribuicao` (`idDistribuicao`)
ON DELETE NO ACTION
ON UPDATE NO ACTION);

```

```

CREATE TABLE IF NOT EXISTS `IPA`.`Transparencia` (
`idTransparencia` INT NOT NULL,
`dataAtualizacao` VARCHAR(45) NOT NULL,
`Distribuicao_idDistribuicao` INT NOT NULL,
PRIMARY KEY (`idTransparencia`),
INDEX `fk_Transparencia_Distribuicao1_idx` (`Distribuicao_idDistribuicao` ASC) VISIBLE,
CONSTRAINT `fk_Transparencia_Distribuicao1`
FOREIGN KEY (`Distribuicao_idDistribuicao`)
REFERENCES `IPA`.`Distribuicao` (`idDistribuicao`)
ON DELETE NO ACTION
ON UPDATE NO ACTION);

```

```
CREATE TABLE IF NOT EXISTS `IPA`.`Telefone` (
  `telefone` VARCHAR(15) NOT NULL,
  `usuario_CPF` VARCHAR(11) NOT NULL,
  PRIMARY KEY (`telefone`),
  INDEX `fk_Teléfono_usuario1_idx` (`usuario_CPF` ASC) VISIBLE,
  CONSTRAINT `fk_Teléfono_usuario1`
    FOREIGN KEY (`usuario_CPF`)
    REFERENCES `IPA`.`usuario` (`CPF`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

```
CREATE TABLE IF NOT EXISTS `IPA`.`endereco` (
  `CEP` INT NOT NULL,
  `Rua` VARCHAR(45) NOT NULL,
  `numero` INT NOT NULL,
  `Bairro` VARCHAR(45) NOT NULL,
  `Complemento` VARCHAR(100) NULL,
  `Armazem_idArmazem` INT NOT NULL,
  INDEX `fk_endereco_Armazem1_idx` (`Armazem_idArmazem` ASC) VISIBLE,
  CONSTRAINT `fk_endereco_Armazem1`
    FOREIGN KEY (`Armazem_idArmazem`)
    REFERENCES `IPA`.`Armazem` (`idArmazem`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

SCRIPT DE TRIGGERS - CREATE TRIGGERS

```
DELIMITER $$  
CREATE TRIGGER trg_distribuicao_move_lote  
AFTER INSERT ON Distribuicao  
FOR EACH ROW  
BEGIN  
    UPDATE Lote  
    SET Armazem_idArmazem = NEW.Armazem_idArmazem_Destino  
    WHERE idLote = NEW.Lote_idLote;  
END $$ DELIMITER ;  
  
DELIMITER $$  
CREATE TRIGGER trg_distribuicao_origem_destino_validacao  
BEFORE INSERT ON Distribuicao  
FOR EACH ROW  
BEGIN  
    IF NEW.Armazem_idArmazem_Origem = NEW.Armazem_idArmazem_Destino THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Origem e destino não podem ser o mesmo armazém.';  
    END IF;  
END $$ DELIMITER ;  
  
DELIMITER $$  
CREATE TRIGGER trg_usuario_definir_data_cadastro  
BEFORE INSERT ON usuario  
FOR EACH ROW  
BEGIN  
    IF NEW.dataCadastro IS NULL THEN  
        SET NEW.dataCadastro = CURDATE();  
    END IF;  
END$$  
DELIMITER ;  
  
DELIMITER $$  
CREATE TRIGGER trg_usuario_senha_tamanho  
BEFORE INSERT ON usuario  
FOR EACH ROW  
BEGIN  
    IF CHAR_LENGTH(NEW.senhaHash) < 8 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'A senha deve ter pelo menos 8 caracteres.';  
    END IF;  
END$$  
DELIMITER ;  
  
DELIMITER $$
```

```

CREATE TRIGGER trg_bloquear_distribuicao_vencida
BEFORE INSERT ON Distribuicao
FOR EACH ROW
BEGIN
    DECLARE validade DATE;

    SELECT s.dataValidade
        INTO validade
        FROM Semente s
        JOIN Lote l ON l.Semente_idSemente = s.idSemente
        WHERE l.idLote = NEW.Lote_idLote;

    IF validade < CURDATE() THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Distribuição bloqueada: lote contém semente vencida.';
    END IF;
END$$
DELIMITER ;

DELIMITER $$

CREATE TRIGGER trg_aviso_validade_proxima
BEFORE INSERT ON Distribuicao
FOR EACH ROW
BEGIN
    DECLARE validade DATE;

    SELECT dataValidade INTO validade
        FROM Semente s
        JOIN Lote l ON l.Semente_idSemente = s.idSemente
        WHERE idLote = NEW.Lote_idLote;

    IF DATEDIFF(validade, NEW.dataDistribuicao) < 30 THEN
        SIGNAL SQLSTATE '01000'
        SET MESSAGE_TEXT = 'Aviso: o lote está com menos de 30 dias para vencer.';
        -- IMPORTANTE: SQLSTATE '01000' = WARNING → NÃO impede o INSERT
    END IF;
END$$
DELIMITER ;

DELIMITER $$

CREATE TRIGGER tg_limite_capacidade_armazem
BEFORE INSERT ON Lote
FOR EACH ROW
BEGIN
    DECLARE capacidade INT;
    DECLARE ocupado INT;

    SELECT capacidadeKg INTO capacidade

```

```

FROM Armazem
WHERE idArmazem = NEW.Armazem_idArmazem;

SELECT COALESCE(SUM(quantidadeKg), 0) INTO ocupado
FROM Lote
WHERE Armazem_idArmazem = NEW.Armazem_idArmazem;

IF (ocupado + NEW.quantidadeKg) > capacidade THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Capacidade do armazém excedida。';
END IF;
END$$
DELIMITER ;

DELIMITER $$$
CREATE TRIGGER tg_no_municipio_duplicado
BEFORE INSERT ON Municipio
FOR EACH ROW
BEGIN
    IF EXISTS(SELECT 1 FROM Municipio WHERE nome = NEW.nome) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Já existe um município cadastrado com este nome。';
    END IF;
END$$
DELIMITER ;

DELIMITER $$$
CREATE TRIGGER trg_atualizar_transparencia
AFTER UPDATE ON Distribuicao
FOR EACH ROW
BEGIN
    UPDATE Transparencia
    SET dataAtualizacao = CURDATE()
    WHERE Distribuicao_idDistribuicao = NEW.idDistribuicao;
END$$
DELIMITER ;

DELIMITER $$$
CREATE TRIGGER trg_definir_data_entrada_lote
BEFORE INSERT ON Lote
FOR EACH ROW
BEGIN
    IF NEW.dataEntrada IS NULL THEN
        SET NEW.dataEntrada = CURDATE();
    END IF;
END$$
DELIMITER ;

```

```
DELIMITER $$  
CREATE TRIGGER trg_impedir_delete_armazem_com_lotes  
BEFORE DELETE ON Armazem  
FOR EACH ROW  
BEGIN  
    IF (SELECT COUNT(*)  
        FROM Lote  
        WHERE Armazem_idArmazem = OLD.idArmazem  
        AND dataSaida IS NULL) > 0 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Não é possível excluir o armazém: existem lotes ativos.';  
    END IF;  
END$$  
DELIMITER ;  
  
DELIMITER $$  
CREATE TRIGGER trg_auto_data_geracao_relatorio  
BEFORE INSERT ON relatorio  
FOR EACH ROW  
BEGIN  
    IF NEW.dataGeracao IS NULL THEN  
        SET NEW.dataGeracao = CURDATE();  
    END IF;  
END$$  
DELIMITER ;
```

SCRIPT DE PROCEDURES - CREATE PROCEDURE

```
# Registrar novo lote
DELIMITER $$

CREATE PROCEDURE sp_registrar_lote(
    IN p_idLote INT,
    IN p_status VARCHAR(45),
    IN p_dataEntrada DATE,
    IN p_quantidadeKg INT,
    IN p_idArmazem INT,
    IN p_idSemente INT)
BEGIN
    INSERT INTO Lote(idLote, status, dataEntrada, quantidadeKg,
    Armazem_idArmazem, Semente_idSemente) VALUES (p_idLote, p_status, p_dataEntrada,
    p_quantidadeKg, p_idArmazem, p_idSemente);
END $$ DELIMITER ;

CALL sp_registrar_lote(
    101,          -- p_idLote
    'DISPONÍVEL', -- p_status
    '2025-01-10', -- p_dataEntrada
    500,          -- p_quantidadeKg
    3,            -- p_idArmazem
    5             -- p_idSemente
);

# Registrar distribuição
DELIMITER $$

CREATE PROCEDURE sp_registrar_distribuicao(
    IN p_idDistribuicao INT,
    IN p_data DATE,
    IN p_status VARCHAR(45),
    IN p_origem INT,
    IN p_destino INT,
    IN p_usuario VARCHAR(11),
    IN p_lote INT)
BEGIN
    INSERT INTO Distribuicao(idDistribuicao, dataDistribuicao, status,
    Armazem_idArmazem_Origem, Armazem_idArmazem_Destino, usuario_CPF, Lote_idLote)
    VALUES (p_idDistribuicao, p_data, p_status, p_origem, p_destino, p_usuario, p_lote);

    UPDATE Lote
    SET status = 'EM MOVIMENTAÇÃO'
    WHERE idLote = p_lote;
END $$ DELIMITER ;

CALL sp_registrar_distribuicao(
```

```
25,          -- idDistribuicao
'2025-01-15',    -- data
'ENVIADO',      -- status
3,            -- armazém origem
5,            -- armazém destino
'10101010101', -- CPF do usuário
10           -- id do lote
);
```

```
# Finalizar distribuição
DELIMITER $$

CREATE PROCEDURE sp_finalizar_distribuicao(
    IN p_lote INT,
    IN p_dataSaida DATE)
BEGIN
    UPDATE Lote
    SET status = 'DISTRIBUÍDO',
        dataSaida = p_dataSaida
    WHERE idLote = p_lote;
END $$ DELIMITER ;
```

```
CALL sp_finalizar_distribuicao(
    10,          -- id do lote
    '2025-01-20' -- data de saída
);
```

```
# Troca o armazém do lote
DELIMITER $$

CREATE PROCEDURE sp_mover_lote(IN p_lote INT,IN p_novoArmazem INT)
BEGIN
    UPDATE Lote
    SET Armazem_idArmazem = p_novoArmazem
    WHERE idLote = p_lote;
END $$ DELIMITER ;
```

```
CALL sp_mover_lote(
    10,
    3
);
```

```
# Gera um relatório da distribuição
DELIMITER $$

CREATE PROCEDURE sp_gerar_relatorio(
    IN p_id INT,
    IN p_inicio DATE,
    IN p_fim DATE,
    IN p_tipo VARCHAR(45),
```

```

    IN p_titulo VARCHAR(60),
    IN p_arquivo VARCHAR(45),
    IN p_distribuicao INT)
BEGIN
    INSERT INTO relatorio(idRelatorio, dataInicio, dataFim, tipo, dataGeracao, titulo,
arquivoPDF, Distribuicao_idDistribuicao) VALUES (p_id, p_inicio, p_fim, p_tipo, CURDATE(),
p_titulo, p_arquivo, p_distribuicao);
END $$ DELIMITER ;

```

```

CALL sp_gerar_relatorio(
    25,          -- id do relatório
    '2025-01-01',      -- data início
    '2025-01-31',      -- data fim
    'DISTRIBUICAO',      -- tipo
    'Relatório de Janeiro', -- título
    'jan2025.pdf',      -- arquivo PDF
    10           -- id da distribuição
);

```

```

# Gera um entrada no portal da transparência
DELIMITER $$

CREATE PROCEDURE sp_atualizar_transparencia(
    IN p_id INT,
    IN p_distribuicao INT)
BEGIN
    INSERT INTO Transparencia(idTransparencia, dataAtualizacao,
Distribuicao_idDistribuicao) VALUES (p_id, DATE_FORMAT(NOW(), '%Y-%m-%d
%H:%i:%s'), p_distribuicao);
END $$ DELIMITER ;

```

```

CALL sp_atualizar_transparencia(
    25,      -- idTransparencia
    15      -- idDistribuicao
);

```

```

# Registro de usuário
DELIMITER $$

CREATE PROCEDURE sp_registrar_usuario(
    IN p_cpf VARCHAR(11),
    IN p_nome VARCHAR(60),
    IN p_email VARCHAR(60),
    IN p_senha VARCHAR(32),
    IN p_cargo VARCHAR(45),
    IN p_tipo VARCHAR(45))
BEGIN

```

```
INSERT INTO usuario(CPF, nome, email, senhaHash, cargo, dataCadastro, tipoUsuario)
VALUES (p_cpf, p_nome, p_email, p_senha, p_cargo, CURDATE(), p_tipo);
END $$ DELIMITER ;

CALL sp_registrar_usuario(
    '12345678902',
    'Maria Silva',
    'maria.silva@gmail.com',
    'senha123',
    'Gestor',
    'ADMIN'
);
```

SCRIPT DE FUNCTIONS - CREATE FUNCTION

```
# Idade das sementes
DELIMITER $$

CREATE FUNCTION fn_idade_semente(p_dataEntrada DATE)
RETURNS INT DETERMINISTIC
BEGIN
    RETURN DATEDIFF(CURDATE(), p_dataEntrada);
END $$ DELIMITER ;

SELECT
    idLote 'ID',
    dataEntrada 'Entrada',
    fn_idade_semente(dataEntrada) 'Idade em Dias'
FROM Lote;

# Sementes Vencidas
DELIMITER $$

CREATE FUNCTION fn_semente_vencida(p_dataValidade DATE)
RETURNS TINYINT DETERMINISTIC
BEGIN
    RETURN p_dataValidade < CURDATE();
END $$ DELIMITER ;

SELECT
    idSemente 'ID',
    tipo 'Semente',
    fn_semente_vencida(dataValidade) 'Vencimento'
FROM Semiente;

# Dias para o vencimento da semente
DELIMITER $$

CREATE FUNCTION fn_dias_para_vencer(p_dataValidade DATE)
RETURNS INT DETERMINISTIC
BEGIN
    RETURN DATEDIFF(p_dataValidade, CURDATE());
END $$ DELIMITER ;

SELECT
    idSemente 'ID',
    tipo 'Semente',
    dataValidade 'Validade',
    fn_dias_para_vencer(dataValidade) 'Prazo de Vencimento'
FROM Semiente;
```

```

# Armazém de um lote
DELIMITER $$

CREATE FUNCTION fn_armazem_lote(p_lote INT)
RETURNS VARCHAR(45) DETERMINISTIC
BEGIN
    DECLARE resultado VARCHAR(45);

    SELECT a.nome INTO resultado
    FROM Lote l
    INNER JOIN Armazem a ON l.Armazem_idArmazem = a.idArmazem
    WHERE l.idLote = p_lote;

    RETURN resultado;
END $$ DELIMITER ;

SELECT
    idLote 'ID',
    fn_armazem_lote(idLote) 'Armazém'
FROM Lote;

# Total distribuído para um armazém
DELIMITER $$

CREATE FUNCTION fn_total_distribuido_destino(p_armazem INT)
RETURNS INT DETERMINISTIC
BEGIN
    DECLARE total INT;

    SELECT SUM(l.quantidadeKg) INTO total
    FROM Distribuicao d
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
    WHERE d.Armazem_idArmazem_Destino = p_armazem;

    RETURN IFNULL(total, 0);
END $$ DELIMITER ;

SELECT
    idArmazem 'ID',
    nome 'Nome',
    fn_total_distribuido_destino(idArmazem) 'Total Distribuído'
FROM Armazem;

# Percentual de ocupação de um armazém
DELIMITER $$

CREATE FUNCTION fn_percentual_ocupacao(p_armazem INT)
RETURNS DECIMAL(5,2) DETERMINISTIC
BEGIN
    DECLARE capacidade INT;

```

```

DECLARE ocupado INT;

SELECT capacidadeKg INTO capacidade FROM Armazem WHERE idArmazem =
p_armazem;

SELECT SUM(quantidadeKg) INTO ocupado
FROM Lote
WHERE Armazem_idArmazem = p_armazem AND status = 'DISPONÍVEL';

RETURN ROUND((IFNULL(ocupado,0) / capacidade) * 100, 2);
END $$ DELIMITER ;

SELECT
idArmazem 'ID',
nome 'Nome',
capacidadeKg,
CONCAT(FORMAT(fn_percentual_ocupacao(idArmazem), 2), '%') `Taxa de Ocupação`
FROM Armazem;

# Responsável pela distribuição
DELIMITER $$

CREATE FUNCTION fn_nome_responsavel(p_idDistribuicao INT)
RETURNS VARCHAR(60) DETERMINISTIC
BEGIN
DECLARE nomeUser VARCHAR(60);

SELECT u.nome INTO nomeUser
FROM Distribuicao d
INNER JOIN usuario u ON d.usuario_CPF = u.CPF
WHERE d.idDistribuicao = p_idDistribuicao;

RETURN nomeUser;
END $$ DELIMITER ;

SELECT
d.idDistribuicao 'ID',
d.dataDistribuicao 'Data',
fn_nome_responsavel(d.idDistribuicao) 'Responsável'
FROM Distribuicao d;

```

SCRIPT DE CRIAÇÃO DE VIEWS - CREATE VIEWS

```
# Situação do estoque por armazém  
CREATE VIEW vw_estoque_armazem AS  
SELECT a.idArmazem,  
       a.nome AS nomeArmazem,  
       m.nome AS municipio,  
       m.regiao,  
       SUM(l.quantidadeKg) AS estoqueAtualKg  
FROM Lote l
```

```
      INNER JOIN Armazem a ON l.Armazem_idArmazem = a.idArmazem  
      INNER JOIN Municipio m ON a.Municipio_idMunicipio1 = m.idMunicipio  
      WHERE l.status = 'DISPONÍVEL'  
      GROUP BY a.idArmazem, a.nome, m.nome, m.regiao;
```

```
# Distribuições realizadas para determinada região
```

```
CREATE VIEW vw_distribuicoes_por_regiao AS  
SELECT m.regiao,  
       COUNT(*) AS totalDistribuicoes  
FROM Distribuicao d  
      INNER JOIN Armazem a ON d.Armazem_idArmazem_Destino = a.idArmazem  
      INNER JOIN Municipio m ON a.Municipio_idMunicipio1 = m.idMunicipio  
      GROUP BY m.regiao;
```

```
# Informações do lote
```

```
CREATE VIEW vw_rastreabilidade_lote AS  
SELECT l.idLote,  
       s.tipo AS tipoSemente,  
       l.dataEntrada,  
       l.dataSaida,  
       l.status,  
       a.nome AS armazemAtual,  
       m.nome AS municipio,  
       m.regiao  
FROM Lote l  
      INNER JOIN Semente s ON l.Semente_idSemente = s.idSemente  
      INNER JOIN Armazem a ON l.Armazem_idArmazem = a.idArmazem  
      INNER JOIN Municipio m ON a.Municipio_idMunicipio1 = m.idMunicipio;
```

```
# Sementes com prazo de validade curto
```

```
CREATE VIEW vw_sementes_proximas_validade AS  
SELECT idSemente,  
       tipo,  
       dataValidade,  
       DATEDIFF(dataValidade, CURDATE()) AS diasParaVencer  
FROM Semente
```

```

WHERE dataValidade <= DATE_ADD(CURDATE(), INTERVAL 60 DAY);

# Distribuições por usuário
CREATE VIEW vw_distribuicoes_por_usuario AS
SELECT u.nome,
       u.cargo,
       u.email,
       COUNT(d.idDistribuicao) AS totalDistribuicoes
  FROM Distribuicao d
    INNER JOIN usuario u ON d.usuario_CPF = u.CPF
   GROUP BY u.CPF;

# Armazém de destino e origem de todas as distribuições
CREATE VIEW vw_fluxo_armazens AS
SELECT d.idDistribuicao,
       d.dataDistribuicao,
       a1.nome AS armazemOrigem,
       a2.nome AS armazemDestino,
       d.status
  FROM Distribuicao d
    INNER JOIN Armazem a1 ON d.Armazem_idArmazem_Origem = a1.idArmazem
    INNER JOIN Armazem a2 ON d.Armazem_idArmazem_Destino = a2.idArmazem;

#Capacidade x Ocupação dos armazéns
CREATE VIEW vw_ocupacao_armazens AS
SELECT a.idArmazem,
       a.nome,
       a.capacidadeKg,
       IFNULL(SUM(l.quantidadeKg), 0) AS ocupacaoKg,
       ROUND((SUM(l.quantidadeKg) / a.capacidadeKg) * 100, 2) AS percentualOcupacao
  FROM Armazem a
    LEFT JOIN Lote l ON l.Armazem_idArmazem = a.idArmazem AND l.status =
'DISPONÍVEL'
   GROUP BY a.idArmazem;

#Quantidade de vezes que um lote foi distribuído
CREATE VIEW vw_historico_distribuicao_por_lote AS
SELECT d.Lote_idLote,
       d.idDistribuicao,
       d.dataDistribuicao,
       a1.nome AS origem,
       a2.nome AS destino,
       d.status
  FROM Distribuicao d
    INNER JOIN Armazem a1 ON d.Armazem_idArmazem_Origem = a1.idArmazem
    INNER JOIN Armazem a2 ON d.Armazem_idArmazem_Destino = a2.idArmazem;

#Relação dos relatórios gerados

```

```

CREATE VIEW vw_relatorios_completos AS
SELECT r.idRelatorio,
       r.titulo,
       r.tipo,
       r.dataGeracao,
       d.idDistribuicao,
       d.dataDistribuicao,
       a.nome AS armazemDestino
FROM relatorio r
    INNER JOIN Distribuicao d ON r.Distribuicao_idDistribuicao = d.idDistribuicao
    INNER JOIN Armazem a ON d.Armazem_idArmazem_Destino = a.idArmazem;

#Visão geral das movimentações do Programa
CREATE VIEW vw_transparencia_publica AS
SELECT t.idTransparencia,
       t.dataAtualizacao,
       d.idDistribuicao,
       d.dataDistribuicao,
       s.tipo AS semente,
       l.quantidadeKg,
       a1.nome AS origem,
       a2.nome AS destino
FROM Transparencia t
    INNER JOIN Distribuicao d ON t.Distribuicao_idDistribuicao = d.idDistribuicao
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
    INNER JOIN Semente s ON l.Semente_idSemente = s.idSemente
    INNER JOIN Armazem a1 ON d.Armazem_idArmazem_Origem = a1.idArmazem
    INNER JOIN Armazem a2 ON d.Armazem_idArmazem_Destino = a2.idArmazem;

```

SCRIPT DE INSERÇÃO - INSERT

```
INSERT INTO IPA.usuario (CPF, nome, email, senhaHash, cargo, dataCadastro, tipoUsuario) VALUES  
('111111111111', 'João Batista Silva', 'joao.silva@ipa.pe.gov.br',  
'5f4dcc3b5aa765d61d8327deb882cf99', 'Gestor de Projetos', '2024-01-10', 'Gestor do IPA'),  
(222222222222, 'Maria Clara Souza', 'maria.souza@ipa.pe.gov.br',  
'e99a18c428cb38d5f260853678922e03', 'Gestora de Operações', '2024-02-15', 'Gestor do IPA'),  
(333333333333, 'Ricardo Santos', 'ricardo.santos@ipa.pe.gov.br',  
'd8578edf8458ce06fbc5bb76a58c5ca4', 'Gestor Administrativo', '2024-03-21', 'Gestor do IPA'),  
(444444444444, 'Fernanda Lima', 'fernanda.lima@ipa.pe.gov.br',  
'25f9e794323b453885f5181f1b624d0b', 'Gestora de Logística', '2024-04-02', 'Gestor do IPA'),  
(555555555555, 'Carlos Eduardo', 'carlos.eduardo@ipa.pe.gov.br',  
'6512bd43d9caa6e02c990b0a82652dca', 'Gestor Regional', '2024-04-18', 'Gestor do IPA'),  
(666666666666, 'Paulo Henrique', 'paulo.henrique@armazem.pe.gov.br',  
'5ebe2294ecd0e0f08eab7690d2a6ee69', 'Gerente de Armazém', '2024-05-11', 'Gerente do Armazem'),  
(777777777777, 'Ana Paula Ferreira', 'ana.paula@armazem.pe.gov.br',  
'6c569aab7775ef8fc570e228c16b98', 'Gerente de Armazém', '2024-06-01', 'Gerente do Armazem'),  
(888888888888, 'Luiz Otávio Ramos', 'luiz.ramos@armazem.pe.gov.br',  
'8f14e45fceea167a5a36dedd4bea2543', 'Gerente de Armazém', '2024-06-25', 'Gerente do Armazem'),  
(999999999999, 'Juliana Beatriz', 'juliana.beatriz@armazem.pe.gov.br',  
'c9f0f895fb98ab9159f51fd0297e236d', 'Gerente de Armazém', '2024-07-04', 'Gerente do Armazem'),  
(10101010101, 'Rafael Moura', 'rafael.moura@armazem.pe.gov.br',  
'45c48cce2e2d7fbdea1afc51c7c6ad26', 'Gerente de Armazém', '2024-07-19', 'Gerente do Armazem'),  
(12121212121, 'Sônia Almeida', 'sonia.almeida@ipa.pe.gov.br',  
'd41d8cd98f00b204e9800998ecf8427e', 'Agente de Distribuição', '2024-08-03', 'Agente de Distribuicao'),  
(13131313131, 'Marcos Vinicius', 'marcos.vinicius@ipa.pe.gov.br',  
'a87ff679a2f3e71d9181a67b7542122c', 'Agente de Distribuição', '2024-08-20', 'Agente de Distribuicao'),  
(14141414141, 'Rita de Cássia', 'rita.cassia@ipa.pe.gov.br',  
'1679091c5a880faf6fb5e6087eb1b2dc', 'Agente de Distribuição', '2024-09-01', 'Agente de Distribuicao'),  
(15151515151, 'Eduardo Pereira', 'eduardo.pereira@ipa.pe.gov.br',  
'8f14e45fceea167a5a36dedd4bea2543', 'Agente de Distribuição', '2024-09-12', 'Agente de Distribuicao'),  
(16161616161, 'Patrícia Gomes', 'patricia.gomes@ipa.pe.gov.br',  
'c4ca4238a0b923820dcc509a6f75849b', 'Agente de Distribuição', '2024-09-30', 'Agente de Distribuicao'),
```

('17171717171', 'Lucas Andrade', 'lucas.andrade@ipa.pe.gov.br',
'c81e728d9d4c2f636f067f89cc14862c', 'Agente de Distribuição', '2024-10-11', 'Agente de
Distribuicao'),
(18181818181', 'Bianca Torres', 'bianca.torres@ipa.pe.gov.br',
'eccbc87e4b5ce2fe28308fd9f2a7baf3', 'Agente de Distribuição', '2024-10-28', 'Agente de
Distribuicao'),
(19191919191', 'Thiago Oliveira', 'thiago.oliveira@ipa.pe.gov.br',
'a87ff679a2f3e71d9181a67b7542122c', 'Agente de Distribuição', '2024-11-05', 'Agente de
Distribuicao'),
(20202020202', 'Larissa Martins', 'larissa.martins@ipa.pe.gov.br',
'e4da3b7fbcbce2345d7772b0674a318d5', 'Agente de Distribuição', '2024-11-18', 'Agente de
Distribuicao'),
(21212121212', 'Pedro Henrique', 'pedro.henrique@ipa.pe.gov.br',
'1679091c5a880faf6fb5e6087eb1b2dc', 'Agente de Distribuição', '2024-11-28', 'Agente de
Distribuicao'),
(23232323232', 'Camila Nogueira', 'camila.nogueira@ipa.pe.gov.br',
'c9f0f895fb98ab9159f51fd0297e236d', 'Agente de Distribuição', '2024-12-02', 'Agente de
Distribuicao'),
(24242424242', 'Matheus Barros', 'matheus.barros@ipa.pe.gov.br',
'45c48cce2e2d7fbdea1afc51c7c6ad26', 'Agente de Distribuição', '2024-12-10', 'Agente de
Distribuicao');

INSERT INTO IPA.Municipio (idMunicipio, nome, regiao) VALUES
(1, 'Recife', 'Região Metropolitana'),
(2, 'Olinda', 'Região Metropolitana'),
(3, 'Jaboatão dos Guararapes', 'Região Metropolitana'),
(4, 'Paulista', 'Região Metropolitana'),
(5, 'Cabo de Santo Agostinho', 'Região Metropolitana'),
(6, 'Ipojuca', 'Região Metropolitana'),
(7, 'Caruaru', 'Agreste'),
(8, 'Garanhuns', 'Agreste'),
(9, 'Santa Cruz do Capibaribe', 'Agreste'),
(10, 'Toritama', 'Agreste'),
(11, 'Bezerros', 'Agreste'),
(12, 'Vitória de Santo Antão', 'Zona da Mata'),
(13, 'Palmares', 'Zona da Mata'),
(14, 'Escada', 'Zona da Mata'),
(15, 'Rio Formoso', 'Zona da Mata'),
(16, 'Barreiros', 'Zona da Mata'),
(17, 'Petrolina', 'Sertão'),
(18, 'Juazeiro', 'Sertão'),
(19, 'Serra Talhada', 'Sertão'),
(20, 'Arcoverde', 'Sertão'),
(21, 'Salgueiro', 'Sertão'),
(22, 'Ouricuri', 'Sertão');

INSERT INTO IPA.Armazem (idArmazem, nome, capacidadeKg, Municipio_idMunicipio1)
VALUES

(1, 'Armazém Central do Recife', 500000, 1),
(2, 'Armazém de Olinda', 220000, 2),
(3, 'Armazém de Jaboatão', 300000, 3),
(4, 'Armazém de Paulista', 180000, 4),
(5, 'Armazém do Cabo de Santo Agostinho', 260000, 5),
(6, 'Armazém de Ipojuca', 240000, 6),
(7, 'Armazém de Caruaru', 350000, 7),
(8, 'Armazém de Garanhuns', 210000, 8),
(9, 'Armazém de Santa Cruz do Capibaribe', 150000, 9),
(10, 'Armazém de Toritama', 120000, 10),
(11, 'Armazém de Bezerros', 140000, 11),
(12, 'Armazém de Vitória de Santo Antão', 200000, 12),
(13, 'Armazém de Palmares', 160000, 13),
(14, 'Armazém de Escada', 130000, 14),
(15, 'Armazém de Rio Formoso', 110000, 15),
(16, 'Armazém de Barreiros', 100000, 16),
(17, 'Armazém de Petrolina', 400000, 17),
(18, 'Armazém de Juazeiro', 380000, 18),
(19, 'Armazém de Serra Talhada', 220000, 19),
(20, 'Armazém de Arcoverde', 190000, 20),
(21, 'Armazém de Salgueiro', 160000, 21),
(22, 'Armazém de Ouricuri', 150000, 22);

INSERT INTO IPA.Semente (idSemente, tipo, dataValidade, safra, observacao, pesoTotalKg) VALUES

(1, 'Milho', '2026-12-31', '2023/2024', 'Variedade BRS 4104 produzida pelo IPA', 8000),
(2, 'Feijão', '2026-10-31', '2023/2024', 'Feijão comum para agricultura familiar', 5000),
(3, 'Sorgo', '2026-11-30', '2023/2024', 'Sorgo granífero adaptado ao semiárido', 6000),
(4, 'Milheto', '2026-09-30', '2023/2024', 'Milheto IPA-BULK 1 categoria S1/S2', 4000),
(5, 'Cebola', '2026-08-31', '2023/2024', 'Cultivar Franciscana IPA-10', 3000),
(6, 'Capim Buffel', '2027-12-31', '2024/2025', 'Capim forrageiro resistente à seca', 10000),
(7, 'Palma Forrageira', '2027-06-30', '2024/2025', 'Palma para alimentação animal no semiárido', 12000);

INSERT INTO IPA.Lote (idLote, status, dataEntrada, dataSaida, quantidadeKg, Armazem_idArmazem, Semente_idSemente) VALUES

(1, 'Disponível', '2024-01-15', NULL, 1200, 1, 1),
(2, 'Disponível', '2024-01-20', NULL, 900, 2, 2),
(3, 'Parcialmente Distribuído', '2024-02-01', NULL, 700, 3, 3),
(4, 'Disponível', '2024-02-18', NULL, 1500, 4, 4),
(5, 'Esgotado', '2024-03-02', '2024-04-10', 0, 5, 5),
(6, 'Disponível', '2024-03-15', NULL, 2000, 6, 6),
(7, 'Parcialmente Distribuído', '2024-03-28', NULL, 300, 7, 7),
(8, 'Disponível', '2024-04-10', NULL, 1800, 8, 1),
(9, 'Disponível', '2024-04-20', NULL, 1600, 9, 2),
(10, 'Parcialmente Distribuído', '2024-05-01', NULL, 450, 10, 3),
(11, 'Disponível', '2024-05-12', NULL, 1300, 11, 4),
(12, 'Aguardando Conferência', '2024-05-25', NULL, 950, 12, 5),

```
(13, 'Disponível', '2024-06-03', NULL, 2200, 13, 6),
(14, 'Esgotado', '2024-06-10', '2024-07-01', 0, 14, 7),
(15, 'Disponível', '2024-06-20', NULL, 1700, 15, 1),
(16, 'Disponível', '2024-07-02', NULL, 1100, 16, 2),
(17, 'Parcialmente Distribuído', '2024-07-18', NULL, 550, 17, 3),
(18, 'Disponível', '2024-07-28', NULL, 2000, 18, 4),
(19, 'Disponível', '2024-08-05', NULL, 1400, 19, 5),
(20, 'Aguardando Conferência', '2024-08-12', NULL, 2300, 20, 6),
(21, 'Disponível', '2024-08-20', NULL, 900, 21, 7),
(22, 'Esgotado', '2024-09-01', '2024-10-05', 0, 22, 1);
```

```
INSERT INTO IPA.Distribuicao (idDistribuicao, dataDistribuicao, status,
Armazem_idArmazem_Origem, Armazem_idArmazem_Destino, usuario_CPF, Lote_idLote)
VALUES
(1, '2025-01-05', 'Concluída', 1, 5, '111111111111', 1),
(2, '2025-01-07', 'Concluída', 1, 7, '222222222222', 2),
(3, '2025-01-10', 'Pendente', 2, 1, '333333333333', 3),
(4, '2025-01-12', 'Em andamento', 3, 1, '444444444444', 4),
(5, '2025-01-14', 'Concluída', 2, 3, '555555555555', 1),
(6, '2025-01-16', 'Concluída', 1, 2, '666666666666', 2),
(7, '2025-01-18', 'Cancelada', 3, 2, '777777777777', 3),
(8, '2025-01-20', 'Pendente', 2, 3, '888888888888', 4),
(9, '2025-01-22', 'Concluída', 1, 12, '999999999999', 1),
(10, '2025-01-24', 'Concluída', 3, 1, '10101010101', 2),
(11, '2025-01-26', 'Em andamento', 1, 2, '12121212121', 3),
(12, '2025-01-28', 'Pendente', 2, 1, '13131313131', 4),
(13, '2025-01-30', 'Concluída', 3, 2, '14141414141', 1),
(14, '2025-02-02', 'Concluída', 2, 15, '15151515151', 2),
(15, '2025-02-04', 'Cancelada', 1, 15, '16161616161', 3),
(16, '2025-02-06', 'Pendente', 3, 1, '17171717171', 4),
(17, '2025-02-08', 'Concluída', 1, 2, '18181818181', 1),
(18, '2025-02-10', 'Concluída', 2, 3, '19191919191', 2),
(19, '2025-02-12', 'Em andamento', 17, 1, '20202020202', 3),
(20, '2025-02-15', 'Concluída', 1, 20, '21212121212', 4),
(21, '2025-02-18', 'Pendente', 2, 20, '23232323232', 1),
(22, '2025-02-20', 'Concluída', 3, 19, '24242424242', 2);
```

```
INSERT INTO IPA.relatorio (idRelatorio, dataFim, dataInicio, tipo, dataGeracao, titulo,
arquivoPDF, Distribuicao_idDistribuicao) VALUES
(1, '2025-01-10', '2025-01-01', 'Distribuição', '2025-01-11', 'Relatório de Distribuição 1',
'relatorio_1.pdf', 1),
(2, '2025-01-12', '2025-01-05', 'Movimentação', '2025-01-13', 'Movimentação de Lotes 2',
'relatorio_2.pdf', 2),
(3, '2025-01-15', '2025-01-07', 'Distribuição', '2025-01-16', 'Relatório de Distribuição 3',
'relatorio_3.pdf', 3),
(4, '2025-01-18', '2025-01-10', 'Atividade', '2025-01-19', 'Atividades do Armazém 4',
'relatorio_4.pdf', 4),
```

(5, '2025-01-20', '2025-01-12', 'Distribuição', '2025-01-21', 'Relatório Completo 5',
'relatorio_5.pdf', 5),
(6, '2025-01-22', '2025-01-14', 'Movimentação', '2025-01-23', 'Movimentação de Sementes
6', 'relatorio_6.pdf', 6),
(7, '2025-01-25', '2025-01-16', 'Distribuição', '2025-01-26', 'Relatório de Distribuição 7',
'relatorio_7.pdf', 7),
(8, '2025-01-26', '2025-01-18', 'Atividade', '2025-01-27', 'Registro de Atividades 8',
'relatorio_8.pdf', 8),
(9, '2025-01-28', '2025-01-19', 'Distribuição', '2025-01-29', 'Resumo Operacional 9',
'relatorio_9.pdf', 9),
(10, '2025-01-30', '2025-01-21', 'Movimentação', '2025-01-31', 'Movimentação de Lotes 10',
'relatorio_10.pdf', 10),
(11, '2025-02-02', '2025-01-25', 'Distribuição', '2025-02-03', 'Relatório Especial 11',
'relatorio_11.pdf', 11),
(12, '2025-02-04', '2025-01-27', 'Atividade', '2025-02-05', 'Registro de Atividades 12',
'relatorio_12.pdf', 12),
(13, '2025-02-06', '2025-01-29', 'Distribuição', '2025-02-07', 'Relatório Técnico 13',
'relatorio_13.pdf', 13),
(14, '2025-02-08', '2025-02-01', 'Movimentação', '2025-02-09', 'Movimentação entre
Armazéns 14', 'relatorio_14.pdf', 14),
(15, '2025-02-10', '2025-02-02', 'Distribuição', '2025-02-11', 'Relatório de Distribuição 15',
'relatorio_15.pdf', 15),
(16, '2025-02-12', '2025-02-03', 'Atividade', '2025-02-13', 'Relatório de Atividades 16',
'relatorio_16.pdf', 16),
(17, '2025-02-14', '2025-02-05', 'Distribuição', '2025-02-15', 'Resumo Operacional 17',
'relatorio_17.pdf', 17),
(18, '2025-02-16', '2025-02-07', 'Movimentação', '2025-02-17', 'Movimentação Geral 18',
'relatorio_18.pdf', 18),
(19, '2025-02-18', '2025-02-09', 'Distribuição', '2025-02-19', 'Relatório de Distribuição 19',
'relatorio_19.pdf', 19),
(20, '2025-02-20', '2025-02-11', 'Atividade', '2025-02-21', 'Registro de Atividades 20',
'relatorio_20.pdf', 20),
(21, '2025-02-22', '2025-02-13', 'Distribuição', '2025-02-23', 'Relatório Especial 21',
'relatorio_21.pdf', 21),
(22, '2025-02-25', '2025-02-15', 'Movimentação', '2025-02-26', 'Movimentação Total 22',
'relatorio_22.pdf', 22);

INSERT INTO IPA.Transparencia (idTransparencia, dataAtualizacao,
Distribuicao_idDistribuicao) VALUES
(1, '2025-01-05', 1),
(2, '2025-01-12', 2),
(3, '2025-01-15', 3),
(4, '2025-01-18', 4),
(5, '2025-01-20', 5),
(6, '2025-01-25', 6),
(7, '2025-02-01', 7),
(8, '2025-02-05', 8),
(9, '2025-02-10', 9),

```
(10, '2025-02-12', 10),
(11, '2025-02-15', 11),
(12, '2025-02-18', 12),
(13, '2025-02-20', 13),
(14, '2025-02-25', 14),
(15, '2025-03-01', 15),
(16, '2025-03-05', 16),
(17, '2025-03-10', 17),
(18, '2025-03-12', 18),
(19, '2025-03-15', 19),
(20, '2025-03-18', 20),
(21, '2025-03-20', 21),
(22, '2025-03-25', 22);
```

```
INSERT INTO IPA.Telefone (telefone, usuario_CPF) VALUES
(81990010001, 11111111111),
(81990010002, 22222222222),
(81990010003, 33333333333),
(81990010004, 44444444444),
(81990010005, 55555555555),
(81990010006, 66666666666),
(81990010007, 77777777777),
(81990010008, 88888888888),
(81990010009, 99999999999),
(81990010010, 10101010101),
(81990010011, 12121212121),
(81990010012, 13131313131),
(81990010013, 14141414141),
(81990010014, 15151515151),
(81990010015, 16161616161),
(81990010016, 17171717171),
(81990010017, 18181818181),
(81990010018, 19191919191),
(81990010019, 20202020202),
(81990010020, 21212121212),
(81990010021, 23232323232),
(81990010022, 24242424242);
```

```
INSERT INTO IPA.endereco (CEP, Rua, numero, Bairro, Complemento,
Armazem_idArmazem) VALUES
(50710000, 'Rua das Acáias', 120, 'Boa Vista', NULL, 1),
(50710001, 'Avenida Pernambuco', 450, 'Centro', 'Próximo à praça', 2),
(50710002, 'Rua Sete de Setembro', 89, 'Santo Antônio', NULL, 3),
(50710003, 'Rua Projetada A', 300, 'Jardim São Paulo', NULL, 4),
(50710004, 'Rua das Mangueiras', 55, 'Ibura', 'Galpão 2', 5),
(50710005, 'Rua Nova Esperança', 220, 'Cordeiro', NULL, 6),
(50710006, 'Rua do Progresso', 135, 'Afogados', NULL, 7),
(50710007, 'Avenida Brasil', 999, 'Imbiribeira', 'Bloco A', 8),
```

(50710008, 'Rua Dom Bosco', 43, 'Boa Vista', NULL, 9),
(50710009, 'Rua Antônio Falcão', 560, 'Boa Viagem', 'Fundos', 10),
(50710010, 'Rua São Miguel', 210, 'Campo Grande', NULL, 11),
(50710011, 'Rua Santa Luzia', 12, 'Água Fria', 'Depósito lateral', 12),
(50710012, 'Rua José Rufino', 870, 'Estâncio', NULL, 13),
(50710013, 'Rua Sargento Silvino', 34, 'Tamarineira', NULL, 14),
(50710014, 'Rua Belmiro Correia', 77, 'Casa Amarela', 'Galpão 1', 15),
(50710015, 'Rua do Futuro', 501, 'Aflitos', NULL, 16),
(50710016, 'Rua Amélia', 160, 'Graças', NULL, 17),
(50710017, 'Rua da Aurora', 680, 'Santo Amaro', 'Pavilhão B', 18),
(50710018, 'Rua do Sol', 25, 'Centro', NULL, 19),
(50710019, 'Rua José de Alencar', 310, 'Espinheiro', NULL, 20),
(50710020, 'Rua Quarenta e Oito', 90, 'Campo Grande', 'Anexo 3', 21),
(50710021, 'Rua Capitão Lima', 410, 'Santo Amaro', NULL, 22);

SCRIPT DE CONSULTA - SELECT

```
# Quantas distribuições cada agente realizou no período de janeiro 2025?
SELECT u.nome 'Agente',
u.CPF 'CPF',
u.email 'Email',
u.cargo 'Cargo',
COUNT(d.idDistribuicao) 'Distribuições'
FROM Distribuicao d
    INNER JOIN usuario u ON d.usuario_CPF = u.CPF
        WHERE d.dataDistribuicao BETWEEN '2025-01-01' AND '2025-01-31'
            GROUP BY u.CPF
                ORDER BY Distribuições DESC;

# Quantas distribuições cada armazém enviou (origem), ordenando do maior para o menor?
SELECT a.nome 'Armazem de Origem',
m.nome 'Município',
m.regiao 'Região',
COUNT(d.idDistribuicao) 'Total de Envios'
FROM Distribuicao d
    INNER JOIN Armazem a ON d.Armazem_idArmazem_Origem = a.idArmazem
    INNER JOIN Municipio m ON a.Municipio_idMunicipio1 = m.idMunicipio
        GROUP BY a.idArmazem
            ORDER BY 'Total de Envios' DESC;

# Listar agentes que realizaram mais de 3 distribuições.
SELECT u.nome,
u.CPF 'CPF',
u.email 'Email',
u.cargo 'Cargo',
COUNT(d.idDistribuicao) 'Total'
FROM usuario u
    INNER JOIN Distribuicao d ON u.CPF = d.usuario_CPF
        GROUP BY u.CPF
            HAVING COUNT(d.idDistribuicao) > 3;

# Qual o número de distribuições concluídas por mês em 2025?
SELECT
DATE_FORMAT(d.dataDistribuicao, '%M') 'Mês',
COUNT(*) 'Total'
FROM Distribuicao d
    WHERE d.status = 'Concluída'
        GROUP BY Mês
            ORDER BY MIN(d.dataDistribuicao);

# Distribuições agrupadas por tipo de semente (via lote).
SELECT
```

```
s.tipo 'Semente',
COUNT(d.idDistribuicao) 'Distribuições',
CONCAT(FORMAT(SUM(l.quantidadeKg), 2, 'de_DE'), ' kg') 'Quantidade'
FROM Distribuicao d
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
    INNER JOIN Semente s ON l.Semente_idSemente = s.idSemente
        GROUP BY s.tipo
            ORDER BY s.tipo;
```

```
# Quantidade de relatórios gerados por cada usuário responsável pela distribuição.
SELECT u.nome,
u.CPF 'CPF',
u.email 'Email',
u.cargo 'Cargo',
COUNT(r.idRelatorio) 'Relatórios Gerados'
FROM relatorio r
    INNER JOIN Distribuicao d ON r.Distribuicao_idDistribuicao = d.idDistribuicao
    INNER JOIN usuario u ON d.usuario_CPF = u.CPF
        GROUP BY u.CPF
            ORDER BY 'Relatórios Gerados' DESC;
```

```
#Listar o total de distribuições registradas na transparência por mês.
SELECT
DATE_FORMAT(t.dataAtualizacao, '%M') 'Mês',
COUNT(*) 'Total de Registros'
FROM Transparencia t
    INNER JOIN Distribuicao d ON t.Distribuicao_idDistribuicao = d.idDistribuicao
        GROUP BY Mês;
```

```
# Quantas distribuições cada armazém recebeu como destino.
SELECT a.nome 'Armazém de Destino',
COUNT(d.idDistribuicao) 'Distribuições Recebidas'
FROM Distribuicao d
    INNER JOIN Armazem a ON d.Armazem_idArmazem_Destino = a.idArmazem
        GROUP BY a.nome;
```

```
# Listar agentes de distribuição e a média de distribuições realizadas por mês.
SELECT u.nome,
u.CPF 'CPF',
u.cargo 'Cargo',
COUNT(d.idDistribuicao) / 12 AS mediaMensal
FROM usuario u
    INNER JOIN Distribuicao d ON u.CPF = d.usuario_CPF
        GROUP BY u.CPF;
```

```
# Total de movimentações por armazém sejam entradas e saídas
SELECT a.nome,
COUNT(d.idDistribuicao) Movimentacoes
```

```
FROM Armazem a
    INNER JOIN Distribuicao d ON a.idArmazem IN (d.Armazem_idArmazem_Origem,
d.Armazem_idArmazem_Destino)
        GROUP BY a.nome
            ORDER BY movimentacoes DESC;
```

Total de semente distribuída por tipo (via soma do peso dos lotes).

```
SELECT s.tipo 'Semente',
CONCAT(FORMAT(SUM(l.quantidadeKg), 2, 'de_DE'), ' kg') 'Total Distribuído'
FROM Distribuicao d
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
        INNER JOIN Semente s ON l.Semente_idSemente = s.idSemente
            GROUP BY s.tipo;
```

#Distribuições feitas por cada agente.

```
SELECT u.nome 'Agente',
u.CPF 'CPF',
u.cargo 'Cargo',
COUNT(d.idDistribuicao) 'Total'
FROM usuario u
    INNER JOIN Distribuicao d ON u.CPF = d.usuario_CPF
        GROUP BY u.CPF
            ORDER BY total DESC;
```

Total de kg distribuídos por tipo de semente em janeiro

```
SELECT
s.tipo 'Semente',
CONCAT(FORMAT(SUM(l.quantidadeKg), 2, 'de_DE'), ' kg') 'Total Distribuído'
FROM Distribuicao d
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
        INNER JOIN Semente s ON l.Semente_idSemente = s.idSemente
            WHERE d.dataDistribuicao BETWEEN '2025-01-01' AND '2025-01-31'
                GROUP BY s.tipo
                    ORDER BY 'Total Distribuído' DESC;
```

Total KG distribuído por município

```
SELECT m.nome Município,
CONCAT(FORMAT(SUM(l.quantidadeKg), 2, 'de_DE'), ' kg') 'Total Distribuído'
FROM Distribuicao d
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
        INNER JOIN Armazem a ON d.Armazem_idArmazem_Destino = a.idArmazem
            INNER JOIN Municipio m ON a.Municipio_idMunicipio1 = m.idMunicipio
                GROUP BY m.nome
                    ORDER BY 'Total Distribuído' DESC;
```

#Total de distribuições por região em determinado ano

```
SELECT m.regiao 'Região',
CONCAT(FORMAT(SUM(l.quantidadeKg), 2, 'de_DE'), ' kg') 'Total Distribuído'
```

```
FROM Distribuicao d INNER JOIN Armazem a ON d.Armazem_idArmazem_Destino =
a.idArmazem
```

```
    INNER JOIN Municipio m ON a.Municipio_idMunicipio1 = m.idMunicipio
```

```
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
```

```
        WHERE YEAR(d.dataDistribuicao) = 2025
```

```
            GROUP BY m.regiao
```

```
                ORDER BY 'Total Distribuído' DESC;
```

```
# Quantidade distribuída por agente
```

```
SELECT u.nome 'Agente',
```

```
CONCAT(FORMAT(SUM(l.quantidadeKg), 2, 'de_DE'), ' kg') 'Total Distribuído'
```

```
FROM Distribuicao d
```

```
    INNER JOIN Usuario u ON d.Usuario_CPF = u.CPF
```

```
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
```

```
        WHERE u.tipoUsuario = 'Agente de Distribuicao'
```

```
            GROUP BY u.nome
```

```
                ORDER BY 'Total Distribuído' DESC;
```

```
# Total KG distribuído por armazém
```

```
SELECT a.nome 'Armazém',
```

```
CONCAT(FORMAT(SUM(l.quantidadeKg), 2, 'de_DE'), ' kg') 'Total Distribuído'
```

```
FROM Distribuicao d
```

```
    INNER JOIN Armazem a ON d.Armazem_idArmazem_Origem = a.idArmazem
```

```
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
```

```
        GROUP BY a.nome
```

```
            ORDER BY 'Total Distribuído' DESC;
```

```
# Média de KG por distribuição por região
```

```
SELECT m.regiao 'Região',
```

```
CONCAT(FORMAT(AVG(l.quantidadeKg), 2, 'de_DE'), ' kg') 'Média Por Região'
```

```
FROM Distribuicao d
```

```
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
```

```
    INNER JOIN Armazem a ON d.Armazem_idArmazem_Destino = a.idArmazem
```

```
    INNER JOIN Municipio m ON a.Municipio_idMunicipio1 = m.idMunicipio
```

```
        GROUP BY m.regiao;
```

```
# KG distribuídos por região em um intervalo (dinâmico)
```

```
SELECT m.regiao,
```

```
CONCAT(FORMAT(SUM(l.quantidadeKg), 2, 'de_DE'), ' kg') 'Total Distribuído'
```

```
FROM Distribuicao d
```

```
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
```

```
    INNER JOIN Armazem a ON d.Armazem_idArmazem_Destino = a.idArmazem
```

```
    INNER JOIN Municipio m ON a.Municipio_idMunicipio1 = m.idMunicipio
```

```
        WHERE d.dataDistribuicao BETWEEN '2025-01-01' AND '2025-01-31'
```

```
            GROUP BY m.regiao;
```

```
# Quantidade total de KG distribuídos por município(Concluidas)
```

```
SELECT
```

```
m.nome 'Município Destino',
CONCAT(FORMAT(SUM(l.quantidadeKg), 2, 'de_DE'), ' kg') 'Total Distribuído'
FROM Distribuicao d
    INNER JOIN Lote l ON d.Lote_idLote = l.idLote
    INNER JOIN Armazem a ON d.Armazem_idArmazem_Destino = a.idArmazem
    INNER JOIN Municipio m ON a.Municipio_idMunicipio1 = m.idMunicipio
    WHERE d.idDistribuicao IN (
        SELECT idDistribuicao
        FROM Distribuicao
            WHERE status = 'Concluída'
        GROUP BY m.nome
        ORDER BY 'Total Distribuído' DESC;
```