

Sistema de Gerenciamento de Estoque para Retentores de Microempresas

Grupo: PowerDevs

**Curso: Análise e Desenvolvimento de
Sistemas**

Composição do Grupo

- Product Owner: **Angela Perez Barcellos**;
- Scrum Master: **Bruno dos Santos Carvalho – RGM: 38997126**;

Equipe:

- **Luis Mario Lira Rodrigues – RGM: 40811956**;
- **Vitoria Olivia Rodrigues – RGM: 39921794**;
- **Geovana Santos de Carvalho – RGM: 38814480**;
- **Nicole Geovanna Pereira de Farias – RGM: 38821923**;
- **Luis Felipe Domingos de Santana – RGM: 38153009**;
- **Gabriel Henrique Sardinha – RGM: 39017915**;
- **Guilherme Ribeiro Anaia – RGM: 39045242**;
- **Pedro Renato Lopes Caires – RGM: 38808943**;
- **Victor Lira Bonifácio dos Santos – RGM: 38363160**;
- **Rafael Gustavo Dias dos Reis – RGM: 37589521**;

Análise de Requisitos

Descrição: Inicialmente fomos recebidos por uma funcionária do setor de RH da empresa Borved que nos apresentou ao diretor, ele nos deu uma explicação bem detalhada sobre o funcionamento da empresa. Resumidamente é uma empresa de fabricação de borracha que utiliza matrizes ou moldes para a criação de retentores para vedação customizada, primeiramente o cliente faz o pedido de acordo com a matriz específica, os funcionários verificam a disponibilidade da matéria-prima para a fabricação dos retentores através de uma tabela no Excel e após a confirmação começa a produção.

Justificativa: Um dos maiores problemas citados foi a falta de praticidade na administração do estoque, a logística da comunicação com a fábrica e a conexão entre os setores, também foi citado uma tentativa de utilizar o sistema da Totvs para gerenciamento de estoque mas devido a complexidade da utilização do sistema ele não foi utilizado.presa, diminuindo o tempo de produção e melhorando a satisfação do cliente.

Objetivos do Projeto

- Oferecer um melhor controle de estoque com relação as matrizes e os produtos;
- Criar uma melhor comunicação entre os setores de vendas, gerencia e produção;
- Desenvolver um sistema de fácil entendimento e usabilidade;
- Proporcionar satisfação ao cliente e à empresa com um sistema que entrega o produto com melhor agilidade.

Requisitos Funcionais

O sistema deve:

- Gerenciar o estoque através do setor administrativo;
- Possuir interfaces distintas para os setores de vendas, gerencia e produção;
- Informar ao usuário a situação do seu pedido;
- Integrar os setores de vendas, gerencia e produção;

- Informar a produção o que será necessário para desenvolver o produto;
- Permitir que a gerência aprove ou não qualquer pedido.

Requisitos Não Funcionais

O sistema deve:

- Ser intuitivo e de fácil utilização;
- Informar ao setor administrativo sobre a falta de matéria-prima;
- Possuir os detalhes referentes as matrizes nos pedidos;
- Ser capaz de suportar grandes pedidos.

Escopo do produto

O sistema deve possuir:

- Banco de dados contendo informações sobre o estoque de matéria-prima e informações referentes aos funcionários e suas funções;
- Interface para vendas contendo informações sobre o produto para o cliente;
- Interface para pré-produção do produto acessada pela administração;
- Interface para a produção contendo a matriz específica e a quantidade de matéria-prima que o produto deve possuir;
- Sistema para a conexão entre o banco de dados e as interfaces;

Backlog Selecionado

- O banco de dados deverá gerenciar o estoque contendo a quantidade e falta de matéria-prima e informações referentes aos funcionários como seu nome, id da empresa, senha e sua função. Essa etapa será desenvolvida por: Bruno Carvalho, Victor Lira, Vitória Olivia e Luis Mario;
- A interface para vendas deve exibir ao usuário as informações detalhadas sobre o produto e nela o usuário deve realizar o login informando também onde o produto deverá ser entregue. Essa etapa será desenvolvida por: Geovanna Carvalho;
- A interface para pré-produção deverá informar a administração os pedidos realizados pelos clientes contendo as informações dos pedidos e se há ou não falta de matéria-prima, a administração pode então aprovar ou reprovar o pedido dependendo da ausência dos itens necessários para produção, caso seja aprovado o pedido segue para a produção, caso não seja o usuário será notificado do motivo do cancelamento. Essa etapa será desenvolvida por: Nicole Farias;
- A interface para a produção deve exibir para os funcionários onde se encontra a matriz específica e os detalhes necessários para a criação do produto, além dos detalhes de como ele deve ser produzido. Essa etapa será desenvolvida por: Luis Felipe;
- A conexão entre o banco de dados e as interfaces deve ser feita de acordo com a logística da empresa e as especificações da gerência e do usuário. Essa etapa será desenvolvida por: Rafael Reis, Pedro Henrique e Gabriel Sardinha;

Backlog de Impedimentos

Viabilidade técnica:

Pequenas empresas de produção industrial possuem um controle de estoque e equipamento para gerenciá-lo, portanto é necessário um sistema simples que auxilie esse gerenciamento, pois alguns sistemas como o da Totvs são muito complexos para pequenas empresas.

Viabilidade econômica:

Por ser um sistema simples de gerenciamento de estoque com armazenamento em nuvem é importante que pequenas empresas o implementem a sua administração, pois assim poderão economizar tempo ao consultar o estoque e se comunicar entre setores, fazendo assim com que tenham um investimento pequeno com um lucro maior.

Viabilidade organizacional:

Devido a dificuldade de gerenciamento de estoque que algumas microempresas possuem, acreditamos que seja necessária a criação de um sistema que melhor administre os produtos e informe a produção o que é preciso para desenvolvê-lo.

Cronograma do projeto

04/08 - Primeira visita a empresa de fabricação de retentores

11/08 - Sprint para confeccionar o planejamento escopo do projeto

12/08 à 03/09 – Sprint para o desenvolvimento do escopo do projeto (inclui o levantamento e análise de requisitos)

04/09 - Entrega do planejamento e iniciação

11/09 - Entrega do levantamento e análise de requisitos

12/09 à 17/09 - Sprint para o desenvolvimento da modelagem conceitual

18/09 - Entrega da modelagem conceitual

19/09 à 24/09 - Sprint para o desenvolvimento da modelagem lógica

25/09 - Entrega da modelagem lógica

26/09 à 08/10 - Sprint para o desenvolvimento da modelagem física

09/10 - Entrega da modelagem física

10/10 à 15/10 - Sprint para o desenvolvimento da documentação final e apresentação do trabalho

16/10 - Apresentação do trabalho

Planos de comunicação

Todos os sprints a serem realizados deverão ser feitos através de reuniões presenciais após o horário de aula nas salas de reunião no bloco A ou online através do Discord e referente as reuniões com o CEO da empresa elas serão realizadas através de visitas técnicas ou intermédio de troca de informações através de um membro do grupo próximo ao CEO da empresa.

Visão

```
CREATE VIEW pedido_por_cliente  
AS ppc  
SELECT cnpj, status  
FROM cliente, pedido  
WHERE status = "execução";
```

```
CREATE VIEW quantidade_por_cliente  
AS qpc  
SELECT cnpj, quantidade  
FROM cliente, inclui  
WHERE q=> 3;
```

Índices

STATUS =
“execução”

Pedido

CNPJ	STATUS
01.633.883/0001-30	execução
57.850.426/0001-51	execução

Quantidade => 3

Inclui

CNPJ	QUANTIDADE
01.633.883/0001-30	3
57.850.426/0001-51	5

Regra de negócio

Cliente:

O Cliente tem todo direito Código de Defesa do Consumidor (CDC) estabelecido no código do consumidor Lei nº 8.078.

Em caso de peças com defeitos só abrir contato que a troca é feita.

Matéria prima:

Cada matéria prima deve ser trabalhada da forma correta, sendo rigoroso o uso correto dos EPI's e das máquinas e matrizes certas, caso algo como temperatura, corte ou definição errada de seguimento aconteça a peça será danificada e será perdida instantaneamente.

Matrizes:

Cada matriz tem sua série que define seu diâmetro, largura, altura e profundidade.

Para cada matriz é importante se lembrar de tomar cuidado para não ser danificada, um pequeno amasso pode ser o suficiente para perder a matriz.

Outras definições:

Toda ordem de trabalho tem que ser seguidas de maneiras correta, a partir do desenho que mostra as definições da peça, sua matriz e o material que será usado para trabalho.

Após uma compra, venda de produto ou ordem de serviço, o setor RH é acionado, levando as demais informações para o Projetista ou para o setor de Administração.

Toda peça deve ser milimetricamente calculada pelo projetista, que será o responsável pelo desenho do projeto.

Funcionários:

Uso de vestimenta apropriada: Camisa, Calça fechada preferência Jeans, Calçado fechado. (É Oferecido uniforme, Camisa e calça.)

Uso de EPI's, durante manuseio de maquinários. O mesmo deve usar EPI adequados sendo eles, Botas de serviço, Luvas de pano e Óculos de proteção.

Proibido uso de celulares durante manuseio de maquinários, risco de acidente de trabalho.

Após chegada de fim do expediente organizar o ambiente de trabalho para o próximo dia.

Estritamente proibido qualquer tipo de preconceito racial, xenofóbico, homofóbico e social.

Seguir as regras da escala de trabalho sendo ela 5/2 Trabalhar 5 dias úteis (Segunda a Sexta) e tem o direito de folga Sábado e domingo. Entrada das 08:00 saída 18:00, equivalente a 10:00 horas de expediente. O Funcionário pode ser contatado para fazer banco de hora em finais de semana.

Scripts SQL e Configurações de Performance

```
SET NAMES utf8mb4;
```

```
CREATE DATABASE IF NOT EXISTS `fabrica`
```

```
    DEFAULT CHARACTER SET utf8mb4
```

```
    DEFAULT COLLATE utf8mb4_unicode_ci;
```

```
USE `fabrica`;
```

```
-- =====
```

```
-- 1) TABELAS (DDL) + ÍNDICES NECESSÁRIOS
```

```
-- =====
```

```
-- =====
```

```
-- TABELA: Cliente
```

```
-- =====
```

```
-- cnpj: CHAR(14) PK | nome_emp: VARCHAR(45) | endereco: VARCHAR(80)
```

```
CREATE TABLE `Cliente` (
```

```
    `cnpj`    CHAR(14)    NOT NULL,
```

```
    `nome_emp` VARCHAR(45) NOT NULL,
```

```
    `endereco` VARCHAR(80) NOT NULL,
```

```
    PRIMARY KEY (`cnpj`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- =====
-- TABELA: Telefone
-- =====

-- num_telefone: numerico(12) PK | cnpj_fk: CHAR(14) FK->Cliente

CREATE TABLE `Telefone` (
    `num_telefone` DECIMAL(12,0) NOT NULL,
    `cnpj_fk`     CHAR(14)     NOT NULL,
    PRIMARY KEY (`num_telefone`),
    KEY `IDX_Telefone_cnpj_fk` (`cnpj_fk`),
    CONSTRAINT `FK_Telefone_Cliente`
        FOREIGN KEY (`cnpj_fk`) REFERENCES `Cliente`(`cnpj`)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- =====
-- TABELA: Email
-- =====

-- end_email: VARCHAR(50) PK | cnpj_fk: CHAR(14) FK->Cliente

CREATE TABLE `Email` (
    `end_email` VARCHAR(50) NOT NULL,
    `cnpj_fk`   CHAR(14)   NOT NULL,
    PRIMARY KEY (`end_email`),
```

```
KEY `IDX_Email_cnpj_fk`(`cnpj_fk`),
CONSTRAINT `FK_Email_Cliente`
FOREIGN KEY (`cnpj_fk`) REFERENCES `Cliente`(`cnpj`)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- =====
```

```
-- TABELA: Funcionario
```

```
-- =====
```

```
-- id_func: AUTO_INCREMENT PK | nome_func: VARCHAR(45) | cargo:
VARCHAR(45)
```

```
CREATE TABLE `Funcionario` (
```

```
 `id_func` INT      NOT NULL AUTO_INCREMENT,
```

```
 `nome_func` VARCHAR(45) NOT NULL,
```

```
 `cargo`    VARCHAR(45) NOT NULL,
```

```
 PRIMARY KEY (`id_func`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- =====
```

```
-- TABELA: Pedido
```

```
-- =====
```

```
-- id_pedido: AUTO_INCREMENT PK | data: DATE | id_func_fk: FK->Funcionario
```

```
-- status: VARCHAR(20) | cnpj_fk: FK->Cliente
```

```
CREATE TABLE `Pedido` (
```

```

`id_pedido` INT      NOT NULL AUTO_INCREMENT,
`data`      DATE     NOT NULL,
`id_func_fk` INT      NOT NULL,
`status`    VARCHAR(20) NOT NULL,
`cnpj_fk`   CHAR(14)  NOT NULL,
PRIMARY KEY (`id_pedido`),
KEY `IDX_Pedido_id_func_fk` (`id_func_fk`),
KEY `IDX_Pedido_cnpj_fk`   (`cnpj_fk`),
CONSTRAINT `FK_Pedido_Funcionario`
FOREIGN KEY (`id_func_fk`) REFERENCES `Funcionario`(`id_func`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `FK_Pedido_Cliente`
FOREIGN KEY (`cnpj_fk`) REFERENCES `Cliente`(`cnpj`)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```
-- =====
```

```
-- TABELA: Matriz
```

```
-- =====
```

```
-- id_matriz: CHAR(7) PK
```

```
CREATE TABLE `Matriz` (
```

```
 `id_matriz` CHAR(7) NOT NULL,
```

```
PRIMARY KEY (`id_matriz`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- =====
```

```
-- TABELA: M_prima (Matéria-prima)
```

```
-- =====
```

```
-- id_m_prima: AUTO_INCREMENT PK | nome_m_prima: VARCHAR(60)
```

```
CREATE TABLE `M_prima` (
```

```
    `id_m_prima` INT      NOT NULL AUTO_INCREMENT,
```

```
    `nome_m_prima` VARCHAR(60) NOT NULL,
```

```
    PRIMARY KEY (`id_m_prima`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- =====
```

```
-- TABELA: Contem (associativa M_prima x Matriz)
```

```
-- =====
```

```
-- PK composta (id_m_prima_fk, id_matriz_fk)
```

```
-- Índice extra em id_matriz_fk para buscas por matriz
```

```
CREATE TABLE `Contem` (
```

```
    `id_m_prima_fk` INT      NOT NULL,
```

```
    `id_matriz_fk` CHAR(7) NOT NULL,
```

```
    PRIMARY KEY (`id_m_prima_fk`, `id_matriz_fk`),
```

```
    KEY `IDX_Contem_id_matriz_fk` (`id_matriz_fk`),
```

```
    CONSTRAINT `FK_Contem_M_prima`
```

```

FOREIGN KEY (`id_m_prima_fk`) REFERENCES `M_prima`(`id_m_prima`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `FK_Contem_Matriz`
FOREIGN KEY (`id_matriz_fk`) REFERENCES `Matriz`(`id_matriz`)
ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

-- =====

-- TABELA: Inclui (associativa Pedido x Matriz)

-- =====

-- PK composta (id_pedido_fk, id_matriz_fk)

-- Índice extra em id_matriz_fk para buscas por matriz

CREATE TABLE `Inclui` (

 `id_pedido_fk` INT NOT NULL,

 `id_matriz_fk` CHAR(7) NOT NULL,

 `quantidade` INT NOT NULL,

 PRIMARY KEY (`id_pedido_fk`, `id_matriz_fk`),

 KEY `IDX_Inclui_id_matriz_fk` (`id_matriz_fk`),

 CONSTRAINT `FK_Inclui_Pedido`

 FOREIGN KEY (`id_pedido_fk`) REFERENCES `Pedido`(`id_pedido`)

 ON DELETE CASCADE

 ON UPDATE CASCADE,

```
CONSTRAINT `FK_Inclui_Matriz`  
    FOREIGN KEY (`id_matriz_fk`) REFERENCES `Matriz`(`id_matriz`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Documentação Técnica

Tabela Clientes:

A tabela Clientes foi criada para armazenar as informações cadastrais dos consumidores que utilizam o sistema. Ela representa o principal ponto de acesso do cliente às funcionalidades disponíveis, permitindo que realizem suas compras e consultem seus pedidos.

O acesso do cliente é restrito às operações relacionadas as vendas, não possuindo permissão para modificar dados internos da empresa. Essa tabela também é responsável por manter o histórico e a integridade dos dados cadastrais de cada cliente, garantindo que cada registro seja único e devidamente identificado por uma chave primária (id_cliente).

Tabela Pedidos:

A tabela Pedidos tem como objetivo registrar e controlar todas as vendas realizadas pela empresa.

Cada pedido recebe um identificador único (id_pedido), garantindo rastreabilidade e integridade das transações.

Além das informações básicas da venda, esta tabela está associada a outras entidades, como Clientes, Funcionários e Matriz, possibilitando o relacionamento entre o cliente que realizou a compra, o funcionário responsável e os produtos incluídos no pedido.

Tabela Funcionário:

A tabela Funcionário foi desenvolvida para representar os colaboradores da empresa que possuem acesso administrativo ao sistema.

Os funcionários têm permissões ampliadas em relação aos clientes, podendo gerenciar o estoque, matrizes de produtos e matérias-primas, além de aprovar ou rejeitar pedidos.

Por motivos de segurança, o acesso a esta tabela requer login e autenticação, visto que os registros contêm dados sensíveis e as operações possuem impacto direto nas demais entidades do sistema.

Tabela M_prima (Matéria-prima):

A tabela M_prima armazena todas as informações referentes às matérias-primas utilizadas na produção dos produtos da empresa.

Entre os dados registrados, incluem-se o nome da matéria-prima, descrição, quantidade disponível e unidade de medida, permitindo o controle e rastreamento dos insumos.

Tabela Matriz:

A tabela Matriz contém os dados das matrizes que compõem a base dos produtos oferecidos pela empresa.

Cada matriz é identificada por um número único (id_matriz) e pode estar associada a diferentes matérias-primas.

Essa estrutura permite compreender a composição de cada produto e otimiza o controle de produção e estoque.

Tabela Associativa Contem:

A tabela Contem é uma tabela associativa responsável por representar o relacionamento muitos-para-muitos entre Matriz e M_prima.

Ela armazena as chaves estrangeiras (id_matriz e id_m_prima), garantindo a integridade referencial entre as duas entidades e permitindo que uma matriz utilize várias matérias-primas, e uma mesma matéria-prima seja usada em diferentes matrizes.

Tabela Associativa Inclui:

A tabela Inclui é uma tabela associativa que estabelece o relacionamento muitos-para-muitos entre Matriz e Pedidos.

Além das chaves estrangeiras (id_matriz e id_pedido), esta tabela contém o campo quantidade, que registra o número de unidades de cada produto incluído no pedido. Essa estrutura permite detalhar o conteúdo de cada compra e facilitar o controle de vendas e estoque.

Tabelas Email e Telefone:

As tabelas Email e Telefone foram criadas para armazenar informações de contato dos usuários.

Por se tratar de dados multivvalorados, cada cliente ou funcionário pode possuir mais de um e-mail ou telefone cadastrado.

Essas tabelas estão relacionadas com Clientes e Funcionários através de chaves estrangeiras, garantindo flexibilidade na atualização e consistência das informações de contato.

Plano de Segurança

Cliente:

Os clientes possuem acesso limitado ao sistema, sendo autorizados apenas a interagir com os pedidos realizados. Eles podem visualizar informações detalhadas sobre as matrizes e seus componentes, mas não possuem permissão para criar, editar ou excluir matrizes. Esse controle garante a segurança e integridade dos dados internos, evitando alterações não autorizadas.

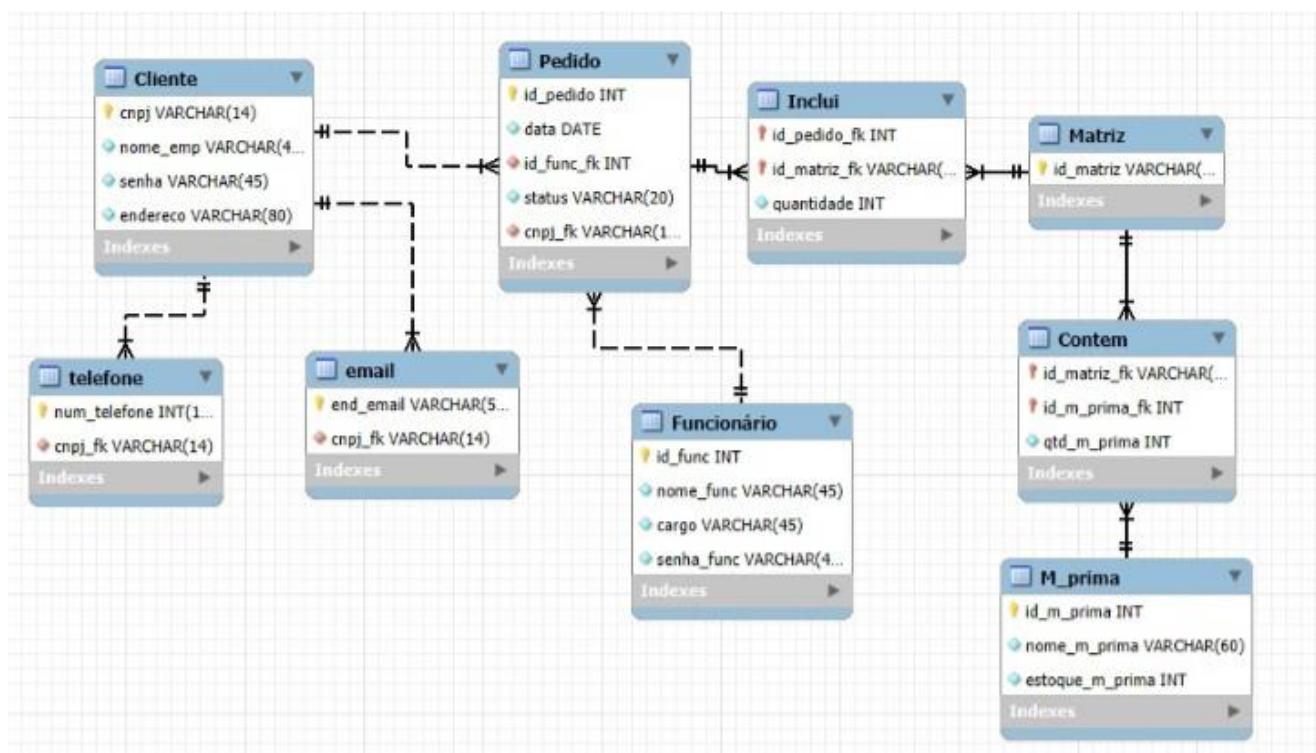
Funcionários e CEO:

Os funcionários com nível gerencial ou administrativo, bem como o CEO da empresa, possuem acesso ampliado dentro do sistema. Eles podem consultar e

gerenciar pedidos realizados pelos clientes, com a possibilidade de cancelar transações quando necessário. Em caso de cancelamento, o sistema exige o registro do motivo, que pode incluir falta de matéria-prima, defeito em matrizes ou problemas técnicos.

Esses usuários também possuem permissão para gerenciar as entidades Matriz e M_prima, podendo criar, editar ou excluir registros conforme as necessidades operacionais da empresa.

Diagrama de Entidade e Relacionamento



Modelo de Entidade e Relacionamento

