

Gerência de Projetos de Software

Termo de Encerramento do Projeto: MRP

Componentes da Equipe: Daniel Estevam Pacheco de Souza, Eduardo Bandeira de Melo Guimarães, Eric Rodrigues Diniz, Lucas Machado de Oliveira Andrade, Mariana Eliza Alves Costa e Vítor Lion Guimarães Rodrigues

1. Objetivo do projeto

O objetivo do projeto é desenvolver um sistema de Planejamento de Recursos de Manufatura (MRP) que otimize a gestão de estoques e a comunicação entre os departamentos de produção, estoque e compras. O sistema busca integrar dados em tempo real, reduzir falhas como excesso ou falta de materiais, e melhorar a eficiência operacional nas empresas. Ele foi concebido para atender demandas específicas de planejamento e execução de produção, excluindo a gestão de compras, com acesso diferenciado para operadores (via dispositivos móveis) e gestores (via plataforma web).

2. Resumo do projeto

O projeto consistiu na concepção, desenvolvimento e implementação de um sistema MRP, utilizando uma arquitetura moderna baseada em microserviços. A solução foi composta por um backend desenvolvido em Node.js (arquitetura MVC) e um frontend utilizando Vue.js e Flutter. O sistema é capaz de:

- Gerenciar produtos e materiais;
- Planejar e monitorar produções em tempo real;
- Enviar e processar solicitações de estoque via filas de mensagens (Kafka);
- Garantir segurança e autenticação com JWT e criptografia (Hash e SSL/TLS);
- Realizar testes automatizados para validação contínua.

3. Artefatos entregues

- Modelagens:
 - Modelagem do banco de dados com suporte à AWS.
 - Diagrama de arquitetura utilizando padrões como Pub-Sub e integração com serviços RESTful.
- Documentos:
 - **Registro das Partes Interessadas:**

Este documento lista todas as partes interessadas no projeto, incluindo gerentes, professores, e papéis específicos como gestores de estoque, operadores e técnicos. Ele define claramente as expectativas, influências,

importância, apoio e observações para cada papel, garantindo o alinhamento entre stakeholders e objetivos do projeto. Contém tabelas detalhadas para identificação, responsabilidades e avaliações de expectativas e poder.

- **Declaração de Escopo:**

O documento descreve os objetivos do projeto MRP, requisitos funcionais e não funcionais, entregáveis, limitações, restrições e premissas. Ele abrange o desenvolvimento de um sistema para gerenciar recursos de manufatura, incluindo funcionalidades web e mobile, comunicação em tempo real via Kafka, e garantia de segurança com JWT e criptografia. Também detalha os marcos do projeto e estabelece as expectativas e limites do desenvolvimento (declaracao_de_escopo).

- **Documento de Arquitetura:**

Este documento detalha a arquitetura do sistema MRP, abordando problemas que o projeto pretende resolver, objetivos específicos, restrições arquiteturais, requisitos funcionais e não funcionais, e mecanismos arquiteturais. Inclui diagramas de visão geral, classes, componentes e entidade-relacionamento para ilustrar a estrutura e operação do sistema. Também apresenta uma visão clara de como os elementos interagem e suportam a funcionalidade pretendida (architecture_document).

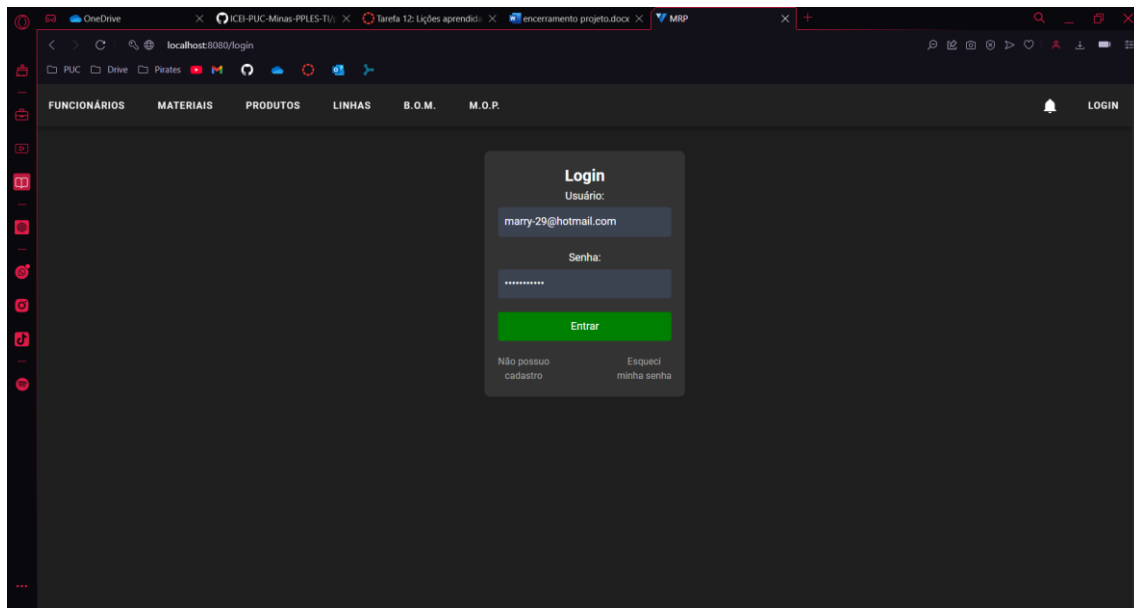
- **Plano de Testes:**

O plano de testes detalha como as funcionalidades do sistema serão verificadas. Ele abrange testes unitários, automatizados e manuais, bem como testes em ambiente beta. Define critérios de aceitação para funcionalidades e classifica a severidade de bugs. Inclui ferramentas como Jest e Playwright para validação de qualidade. O objetivo é garantir que o sistema atenda aos requisitos funcionais e não funcionais antes do lançamento (Plano de teste).

- **Protótipos Funcionais:**

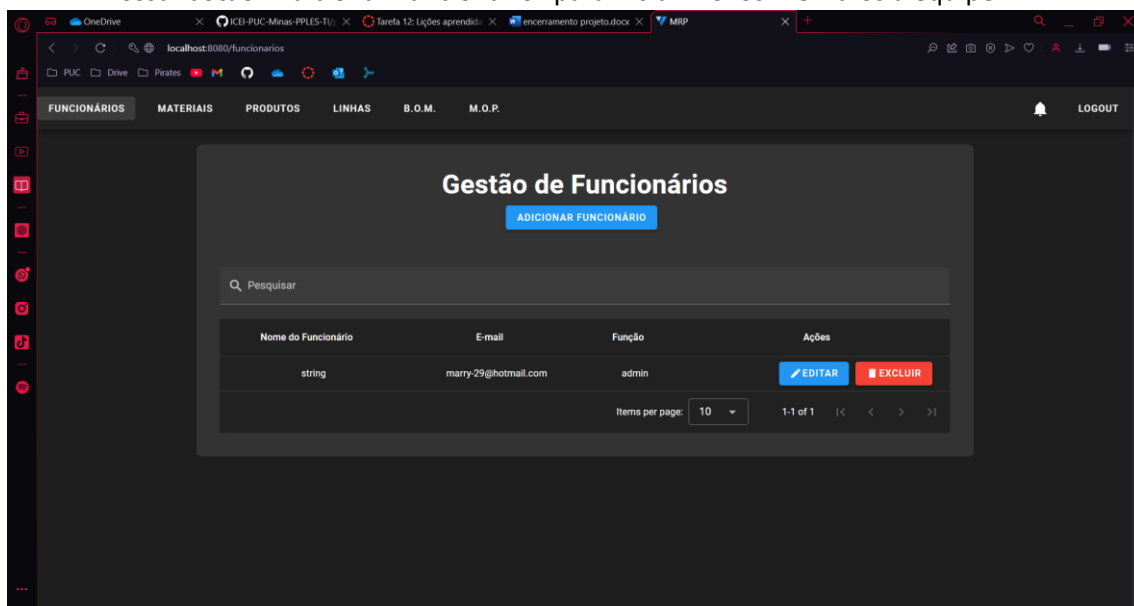
Login de Usuário:

- Permite que os usuários se autentiquem utilizando e-mail e senha.
- Inclui opções de "Esqueci minha senha" e "Não possuo cadastro" para recuperação de senha ou criação de nova conta.
- Fornece acesso restrito às funcionalidades com base no perfil do usuário.



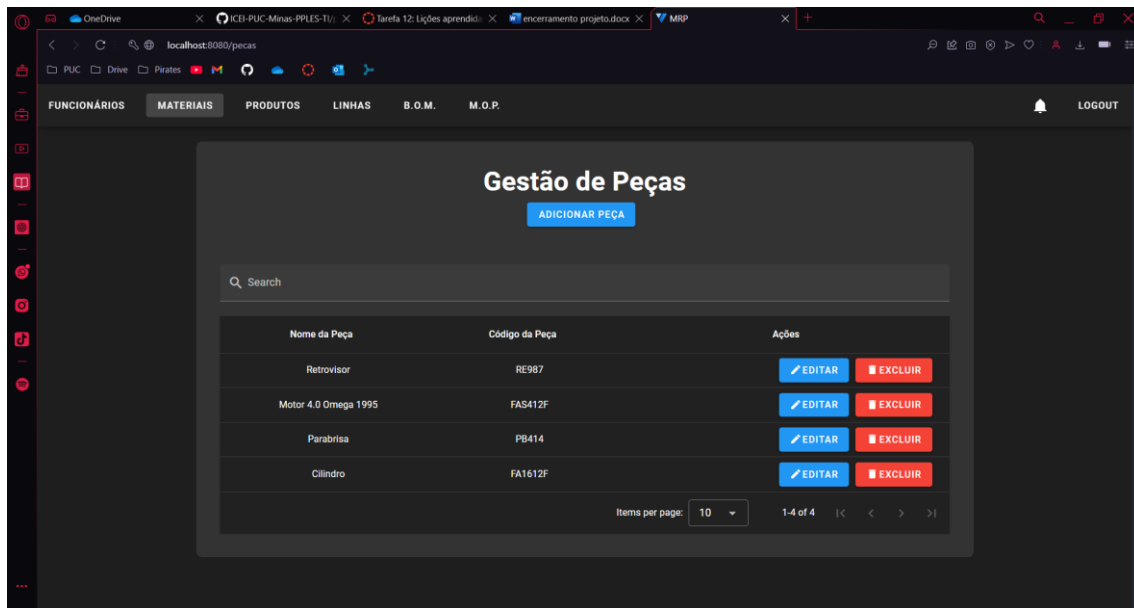
Gestão de Funcionários:

- Lista os funcionários cadastrados, incluindo nome, e-mail e função (ex.: admin, operador).
- Oferece ações como editar ou excluir os registros.
- Possui botão "Adicionar Funcionário" para incluir novos membros à equipe.



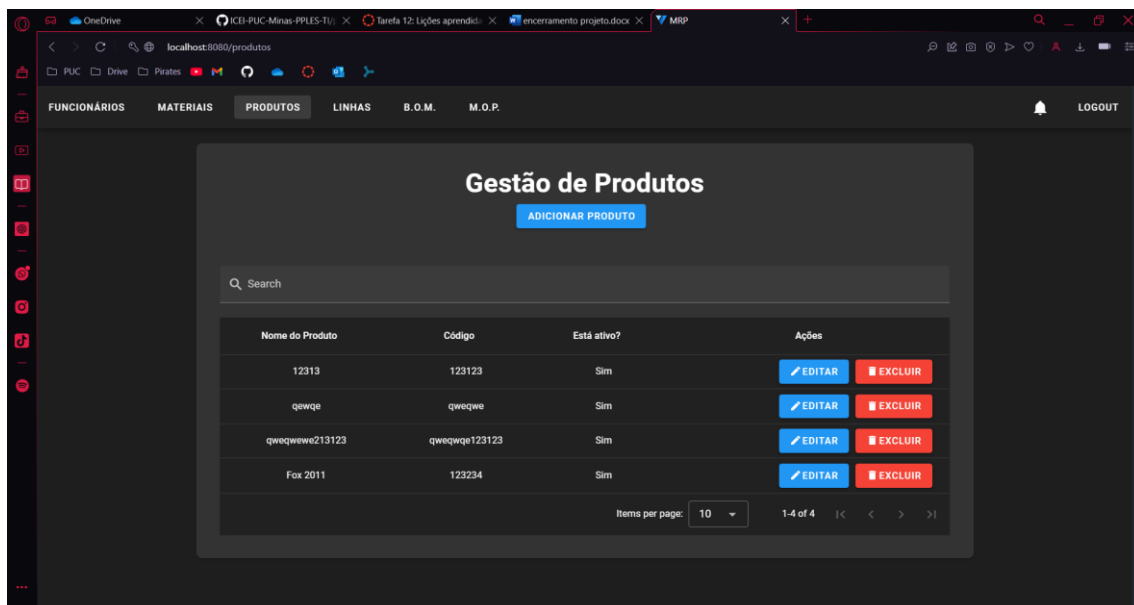
Gestão de Peças:

- Mostra uma lista de peças com nome e código de identificação.
- Permite editar ou excluir peças existentes.
- Inclui a opção "Adicionar Peça" para cadastro de novos materiais.



Gestão de Produtos:

- Exibe produtos cadastrados com nome, código e status (ativo ou inativo).
- Oferece opções de editar ou excluir registros existentes.
- Possui funcionalidade de adicionar novos produtos.



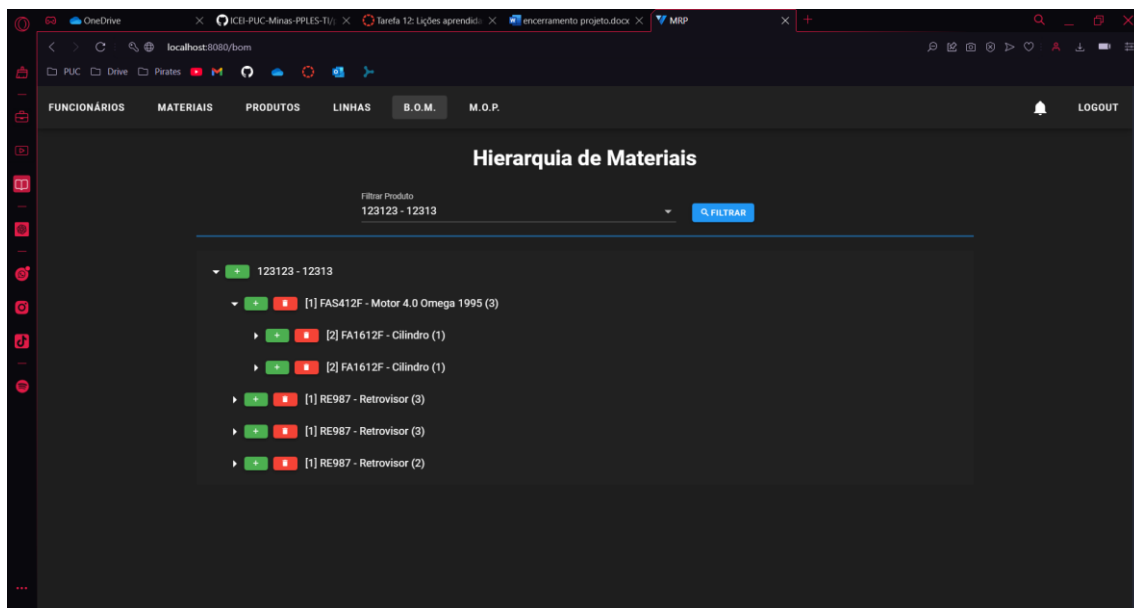
Gestão de Linhas de Produção:

- Permite visualizar e gerenciar linhas de produção, mostrando nomes das linhas.
- Inclui ações como editar ou excluir linhas.
- Possui botão "Adicionar Linha" para criação de novas linhas de produção.



Hierarquia de Materiais (B.O.M - Bill of Materials):

- Exibe a hierarquia de materiais relacionados a um produto específico.
- Permite visualizar e organizar componentes de produtos, detalhando quantidade e códigos de peças.
- Oferece filtragem por produto para facilitar a navegação.



Planejamento de Produção (M.O.P - Master Operation Plan):

- Disponibiliza um calendário para planejamento de produção.
- Oferece visualização por mês, semana ou dia, com eventos e tarefas programadas.
- Detalha informações sobre tarefas, como quantidade de itens, nome do produto e linha de produção alocada.

Planejamento da Produção

Mês Semana Dia

Dezembro 2024

	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira	Sábado	Domingo
48			27				1
49	2	3	4	5	6	7	8
50	9	10	11	12	13	14	15
51	16	17	18	19	20	21	22
52	23	24	25	26	27	28	29
1	30	31					

Visualização por mes

Planejamento da Produção

Mês **Semana** Dia

Semana 49 (Dezembro 2024)

Segunda-feira 2	Terça-feira 3	Quarta-feira 4	Quinta-feira 5	Sexta-feira 6	Sábado 7	Domingo 8
Sem eventos	Sem eventos	Sem eventos	Sem eventos	qweqwe Quantidade: 2 Produto: qewqe Linha: corsa	Sem eventos	Sem eventos

Visualização detalhada por semana

- Soluções de Software:
 - Backend (Node.js) com rotas de produção, planos de produção e controle de estoques.
 - Frontend (Vue.js e Flutter) com interfaces responsivas para operadores e gestores.
 - Integração com Kafka para mensagens assíncronas e comunicações em tempo real.

- Implementação de Processos Automatizados:

A metodologia dos testes no sistema MRP segue uma abordagem estruturada, garantindo a validação das funcionalidades desenvolvidas e a identificação de possíveis falhas antes do lançamento oficial. Abaixo está um detalhamento dos principais aspectos:

1. Escopo de Testes

Os testes abrangem todas as funcionalidades descritas na tabela de requisitos funcionais, incluindo

- Gerenciamento de produtos, materiais, hierarquias, e linhas de produção.
- Planejamento de operações, notificações e autenticação.
- Comunicação em tempo real e controle de inventário.
- Exclusão: Funcionalidades fora do escopo do sistema, como o cadastro de itens irrelevantes (e.g., livros), não serão testadas.

2. Níveis de Testes

A execução dos testes é distribuída em diferentes níveis:

- Testes Unitários:
 - Responsabilidade dos desenvolvedores.
 - Cobertura mínima de 60%.
 - Garantem que funções e métodos individuais funcionem corretamente.
- Testes Automatizados:
 - Realizados pelo time de qualidade utilizando Playwright.
 - Foco principal na funcionalidade de Login (end-to-end).
 - Simulam o comportamento real do usuário e garantem a integração entre componentes.
- Testes Manuais:
 - Executados pelo time de qualidade com base na documentação de cenários de teste.
 - Testam funcionalidades completas e fluxos do sistema.
- Versão Beta:
 - Disponibilização do sistema para 3 usuários pré-cadastrados.
 - Identificação de possíveis problemas em um ambiente próximo ao real.

3. Ambiente de Teste

Os testes são realizados em um ambiente de homologação, que replica o ambiente de produção. Esse ambiente inclui:

- Configurações idênticas ao ambiente final.
- Massa de dados previamente gerada pelo time de qualidade para simulações.

4. Ferramentas de Teste

As ferramentas utilizadas garantem eficiência e confiabilidade:

- Jest: Framework para testes unitários.
- Playwright: Ferramenta para testes automatizados (end-to-end).

5. Classificação de Bugs

Os bugs identificados são classificados em níveis de severidade:

- Blocker: Impedem o teste ou uso completo da funcionalidade; bloqueiam a entrega.
- Grave: Funcionalidade não opera como esperado, podendo causar efeitos irreversíveis.
- Moderada: Funcionalidade não atende completamente aos critérios de aceitação, mas permanece utilizável.
- Pequena: Impacto mínimo na funcionalidade; geralmente afetam a experiência do usuário, como erros ortográficos.

6. Critérios de Aceitação e "Definição de Pronto"

Uma funcionalidade será considerada pronta quando:

- Passar por todos os níveis de teste descritos.
- Não apresentar bugs com severidade acima de Moderada.
- Receber validação de negócio pelo time de produto.[

4. Conclusões

- Resultados Obtidos
 - Os resultados atingidos foram diretamente relacionados aos objetivos iniciais. O sistema MRP desenvolvido demonstrou ser uma ferramenta robusta para otimizar o planejamento e a execução de recursos de manufatura, melhorando a comunicação entre setores e reduzindo falhas operacionais. A integração de tecnologias como Kafka e JWT garantiu uma arquitetura escalável e segura.
- Principais Contribuições
 - Um sistema funcional e flexível para gestão de produção, testado e validado para atender demandas reais.
 - Ferramentas de automação que simplificaram processos manuais, como o planejamento de materiais e controle de estoques.
 - Incremento no aprendizado prático de ferramentas modernas, como Kafka, JWT, e Vue.js.
- Lições Aprendidas

- Tecnológicas: A escolha adequada de tecnologias, como Kafka, foi crucial para lidar com comunicação em tempo real e garantir alta disponibilidade.
- Gerenciais: A definição clara de papéis no time evitou retrabalho e melhorou a produtividade.
- Acadêmicas: O uso de conceitos de arquitetura de software, como microserviços e Pub-Sub, foi aplicado com sucesso em um cenário real.
- Pessoais: O trabalho em equipe e a comunicação foram essenciais para o cumprimento dos prazos e metas.
- Operacionais: A implementação de testes automatizados garantiu um sistema mais confiável e reduziu o tempo de manutenção.