

Relatório de Levantamento de Requisitos do Sistema de Controle de Estoque com IoT e Dashboard Web

1. Introdução

Este documento tem como objetivo apresentar o levantamento de requisitos para o desenvolvimento de um sistema de controle de estoque automatizado, baseado na tecnologia IoT (Internet das Coisas) e com interface web. O projeto visa atender mercados autônomos e convencionais, oferecendo uma solução eficiente para rastreabilidade de produtos, controle de estoque e redução de perdas operacionais.

O sistema integra hardware (leitor RFID e microcontrolador ESP32) e software (API, banco de dados e dashboard web) para fornecer uma visão em tempo real da situação do estoque, além de gerar alertas e relatórios.

2. Objetivos do Sistema

- Automatizar o controle de estoque.
 - Aumentar a rastreabilidade dos produtos nas prateleiras.
 - Reduzir perdas financeiras causadas por vencimento de produtos ou ruptura de estoque.
 - Fornecer informações precisas e atualizadas para gerentes e reposidores de estoque.
 - Oferecer uma solução escalável, segura e de fácil acesso.
-

3. Escopo do Sistema

O sistema é composto por três principais camadas:

1. Camada IoT:

- Leitor RFID instalado nas prateleiras.
- Microcontrolador ESP32 para captura dos dados e envio à nuvem.

2. Backend (API e WebSocket):

- Recepção dos dados do ESP32 via protocolo HTTP ou WebSocket.
- Processamento dos dados, persistência no banco de dados e distribuição das informações ao frontend.

3. Frontend (Dashboard Web):

- Interface para visualização dos dados de estoque em tempo real.
 - Geração de relatórios e emissão de alertas.
-

4. Perfis de Usuários

Perfil

Permissões

Administrador

Gestão total do sistema, configurações, manutenção e correções técnicas.

Gerente de Estoque

Visualização completa do dashboard, geração de relatórios e recepção de alertas.

Repositor

Consulta do estoque e acompanhamento dos alertas de falta de produtos.

5. Requisitos Funcionais

5.1. Dashboard

- Exibir estoque atual por categoria e quantidade.
- Mostrar alertas de:
 - Produtos em falta.
 - Produtos próximos da data de vencimento.
- Apresentar gráficos de controle (entrada/saída futura e tendências).
- Fornecer resumo geral com:
 - Quantidade total de produtos.
 - Quantidade de alertas ativos.
 - Produtos mais críticos.

5.2. Relatórios

- Geração de relatórios em formato PDF contendo:
 - Estoque atual.
 - Produtos com validade próxima.
 - Produtos em falta.
 - Data e hora da geração.

5.3. Notificações

- Notificar usuários sobre:
 - Produtos em falta.
 - Produtos próximos do vencimento.

5.4. Integração com IoT

- Receber dados de leitores RFID conectados ao ESP32.
- Processar eventos de entrada e saída de produtos automaticamente.

5.5. Gestão de Usuários

- Cadastro, edição e remoção de usuários.
- Controle de permissões por perfil (Administrador, Gerente e Repositor).

5.6. Segurança

- Autenticação de usuários.
 - Comunicação segura via HTTPS.
 - Validação de dados provenientes do ESP32.
-

6. Requisitos Não Funcionais

- **Responsividade:** compatível com desktop e navegadores mobile.
- **PWA:** possibilidade de implementação para oferecer:
 - Instalação como app.
 - Funcionamento offline limitado.
 - Notificações push (futuro).
- **Desempenho:** atualização quase em tempo real dos dados via WebSocket.
- **Escalabilidade:** sistema projetado para fácil expansão na nuvem.
- **Disponibilidade:** operação contínua, dependente da hospedagem na nuvem.

- **Segurança:** comunicação criptografada, autenticação robusta e proteção dos dados.
-

7. Comunicação IoT

- **Protocolo Utilizado:**
 - Preferência por **HTTP POST** ou **WebSocket**, visando simplicidade no MVP.
 - MQTT descartado devido à necessidade de um broker dedicado, o que eleva a complexidade e os custos de implantação.
 - **Fluxo de Dados:**
 - O ESP32 envia os dados capturados via HTTP para o backend.
 - O backend processa e atualiza os dados no banco de dados.
 - O frontend recebe atualizações em tempo real por meio de WebSocket.
-

8. Tecnologias Recomendadas

Camada

Tecnologias Sugeridas

IoT

ESP32 wROOM32 + Leitor RFID + Protocolo HTTP ou WebSocket

Backend/API

Node.js (Express ou NestJS) ou Python (FastAPI) + [Socket.io](https://socket.io/)

Banco de Dados

PostgreSQL (recomendado), alternativas: MySQL, SQLite (dev), ou Firebase (NoSQL)

Frontend

React.js + Vite ou Next.js + Tailwind CSS + Shadcn UI

Hospedagem

Render, Railway, Vercel, Netlify, [Fly.io](https://fly.io), ou Firebase Hosting

9. Arquitetura do Sistema

- **ESP32 → API Backend:** comunicação via HTTP POST com payload contendo dados do RFID.
 - **API Backend → Banco de Dados:** armazenamento dos dados recebidos.
 - **API Backend → Frontend:** envio de atualizações via WebSocket.
 - **Frontend:** exibição dos dados em tempo real no dashboard, geração de relatórios e alertas.
-

10. Considerações Finais

O sistema proposto visa oferecer uma solução robusta, escalável e eficiente para controle de estoque no setor varejista, especialmente em mercados autônomos. A utilização de IoT integrada a um dashboard web permitirá automação de processos, redução de perdas e melhoria na gestão do estoque.

A definição de tecnologias prioriza simplicidade para o desenvolvimento do MVP, mas mantém a possibilidade de expansão e aprimoramento futuro.