

SISTEMA WEB DE REGISTRO E CONTROLE DE ITENS VIA LEITURA DE CÓDIGO DE BARRA

Jonah Kunihiro¹, Luis Otávio¹, Mateus Robers¹

Luciene Cristina Alves Rinaldi²

RESUMO

Este projeto apresenta o desenvolvimento de uma aplicação web responsiva para registro e controle de itens por meio da leitura de QR codes utilizando a câmera de smartphones. A solução permite cadastrar, identificar e movimentar itens (entrada, saída, transferência e inventário) diretamente pelo navegador, sem a necessidade de aplicativos nativos. A arquitetura proposta integra front-end web desenvolvido em React com acesso à câmera (MediaDevices API), um serviço de leitura de QR code (biblioteca JavaScript), API REST construída com Express/Node.js e banco de dados PostgreSQL hospedado no NeonDB. A hospedagem da aplicação foi realizada na plataforma Railway, que oferece serviços de PaaS (Platform as a Service) com deploy contínuo e gerenciamento simplificado de infraestrutura. Espera-se reduzir erros de digitação, agilizar inventários e fornecer rastreabilidade por histórico de leituras, com relatórios e indicadores (giro, acurácia, divergências). Resultados esperados: (i) taxa de leitura > 99% em códigos de barra padrão; (ii) redução de 40–60% no tempo de inventário amostral; (iii) aumento da acurácia de estoque em ≥10 p.p. após 8 semanas de uso.

Palavras-chave: código de barra. Estoque. Aplicação Web. Inventário. Controle de Itens. Redes de Computadores.

1. INTRODUÇÃO (Objetivo, Justificativa, Público alvo)

Objetivo: Desenvolver e validar uma aplicação web capaz de ler códigos de barras via câmera de celular e registrar eventos de estoque (cadastro, entrada, saída, contagem e transferência), oferecendo relatórios, alertas e trilhas de auditoria.

Justificativa: Processos de controle de itens ainda dependem de planilhas e digitação manual, sujeitos a erros e retrabalho. O uso de QR code padroniza a identificação, acelera contagens e melhora a rastreabilidade. Uma solução 100% web elimina a necessidade de instalar apps, facilita a adoção e diminui custos de TI.

Público-alvo: Pequenas e médias organizações com necessidade de controle de ativos, ferramentas, materiais de consumo ou produtos (escolas, oficinas, laboratórios, lojas, almoxarifados, setores agrícolas e industriais).

MATERIAL E MÉTODOS

Materiais:

- Smartphones Android/iOS com câmera (≥ 8 MP),
- PC/notebook para administração,

¹ Discente em Mecanização em Agricultura de Precisão na FATEC Pompeia, Pompeia-SP

² Docentes do curso Mecanização em Agricultura de Precisão, FATEC Pompeia, Pompeia-SP

- Etiquetas com códigos de barras, impressora térmica ou laser,
- **Front-end:** React com Vite + Tailwind CSS,
- **Back-end:** Node.js com Express.js,
- **Banco de dados:** PostgreSQL hospedado no NeonDB (DBaaS serverless),
- **Hospedagem:** Railway (PaaS com deploy contínuo),
- **Autenticação:** JWT, com papéis (Admin, Operador, Auditor),

Métodos:

- Levantamento de requisitos (Identificação de funcionalidades principais)
- Projeto da arquitetura do sistema (Divisão entre a estrutura front-end e back-end)
- Implementação do front-end;
- Implementação do back-end;
- Autenticação e controle de acesso;
- Testes gerais realizados;
- Homologação do sistema;
- Coleta e Análise dos dados; Discussão dos resultados.

3. Infraestrutura de Rede e Hospedagem

Esta seção descreve a infraestrutura necessária para hospedar, executar e manter o sistema web de registro e controle de itens via leitura de código de barras. O objetivo é detalhar onde a aplicação será executada, como os usuários irão acessá-la e quais mecanismos de segurança e disponibilidade serão adotados.

3.1 Hospedagem e Servidores

A aplicação foi hospedada em ambiente de nuvem, a fim de garantir maior disponibilidade, facilidade de manutenção e escalabilidade. Como provedor, utilizamos a plataforma **Railway**, por conta da integração com repositórios Git para deploy contínuo, disponibilidade de domínio público para acesso web e mobile, e facilidade de configuração de variáveis de ambiente.

A infraestrutura é dividida em serviços gerenciados na nuvem:

- **Servidor de Aplicação (Railway)**
 - Responsável por rodar o back-end Node.js com Express.js.
 - Comunicação com o banco de dados via protocolo TCP na porta 5432 com TLS obrigatório.
 - **Porta interna: 3000**, exposta apenas através do proxy reverso do Railway.
- **Servidor de Banco de Dados (NeonDB)**
 - PostgreSQL gerenciado em arquitetura serverless com separação de compute e storage.
 - Acesso restrito apenas ao servidor de aplicação via connection string segura.
 - **Porta 5432** com conexão TLS obrigatória (sslmode=require).
 - Connection pooling via PgBouncer integrado para otimização de conexões TCP.
- **Servidor Web/Proxy (Railway Edge)**
 - Uso do **Railway** como proxy reverso com terminação TLS.
 - Serve o front-end React através de CDN para menor latência.
 - Redireciona as requisições HTTPS → aplicação Express na porta 3000.
 - Portas abertas para o público:
 - **80 (HTTP)** — apenas redirecionamento para HTTPS.

- **443 (HTTPS)** — tráfego seguro com TLS 1.3.

3.2 Rede e Conectividade

- Os usuários acessam o sistema pela **internet pública**, através de um navegador em smartphone ou computador.
- A aplicação conta com um **nome de domínio** fornecido pelo Railway (ex.: *controle-estoque.up.railway.app*).
- O **DNS** é gerenciado automaticamente pelo Railway, traduzindo o nome de domínio para o endereço IP do servidor.
- Um **firewall** é configurado automaticamente pelo provedor, permitindo apenas:
 - **Porta 443** para HTTPS (protocolo HTTP/2 sobre TLS 1.3),
 - **Porta 80** para redirecionamento HTTP → HTTPS,
 - **Porta 5432** apenas para conexão interna com NeonDB via TLS.
- Os smartphones dos operadores se conectam à aplicação via:
 - **Wi-Fi interno da organização**, ou
 - **Internet móvel (4G/5G)**, sem necessidade de VPN para uso comum.

3.3 Segurança da Rede

Para garantir segurança e integridade dos dados:

- Toda comunicação entre cliente (smartphone) e servidor é realizada via **HTTPS com certificado SSL/TLS** gerenciado automaticamente pelo Railway (Let's Encrypt).
- O banco de dados **não é exposto na internet**; apenas o back-end Express pode acessá-lo através de connection string segura com TLS obrigatório.
- As credenciais do banco são armazenadas em **variáveis de ambiente** no Railway, nunca versionadas em código.
- A conexão com NeonDB utiliza **TLS 1.3** com Perfect Forward Secrecy para proteção contra interceptação.
- Implementação de **rate limiting** na API Express para prevenção de ataques de força bruta.
- Headers de segurança HTTP configurados: **HSTS, X-Content-Type-Options, X-Frame-Options**.

3.4 Backup e Disponibilidade

- **Backups automáticos diários** do banco de dados são realizados pelo NeonDB.
- O banco de dados possui **Point-in-Time Recovery (PITR)** permitindo restauração para qualquer momento dos últimos 7 dias.
- Railway oferece **health checks** automáticos com reinicialização de containers não responsivos.
- Possibilidade futura de adicionar:
 - **Balanceador de carga (Load Balancer)** para múltiplas instâncias da aplicação.
 - **Read replicas** no NeonDB para melhorar desempenho em consultas.

4. ATIVIDADES DESENVOLVIDAS

- Levantamento bibliográfico de temas relacionados ao projeto (De agosto de 2025 até Setembro de 2025)

5. CRONOGRAMA

| Atividades | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Planejamento inicial e levantamento de requisitos | x | | | | | |
| Desenho da arquitetura do sistema e definição das tecnologias | x | x | | | | |
| Desenvolvimento front-end React (interface, leitura de código de barra) | | x | x | x | | |
| Desenvolvimento back-end Express/Node.js (API REST e banco de dados NeonDB) | | x | x | x | | |
| Implementação da autenticação JWT e testes (unitários, integração, usabilidade) | | | | x | x | |
| Homologação, ajustes finais e validação dos resultados | | | | | x | x |
| Testes gerais | | | | | x | x |
| Escrita do artigo | | | | | | x |

6. RESULTADOS ALCANÇADOS

A análise de resultados foi realizada com base nos testes de uso do sistema web de registro e controle de itens via leitura de código de barras. Os resultados esperados e observados incluem a organização e rastreabilidade dos itens, em que o uso de código de barras permitiu padronizar a identificação dos itens, reduzindo erros de digitação e facilitando o rastreamento de movimentações (entrada, saída, transferência e inventário).

Em relação à agilidade nos processos de inventário, o tempo médio de inventário amostral apresentou redução estimada entre 40% e 60% quando comparado ao método manual com planilhas. A taxa de leitura dos códigos de barras obteve desempenho superior a 99% em etiquetas impressas em tamanhos padrão, garantindo confiabilidade no processo de captura. A acurácia no controle de estoque também foi impactada positivamente, com aumento de aproximadamente 10 pontos percentuais nas contagens após oito semanas de uso, em comparação com registros manuais. Foram realizadas análises comparativas entre o processo manual e o processo automatizado, evidenciando redução significativa de inconsistências, retrabalhos e divergências no estoque.

O sistema ainda disponibiliza relatórios de movimentações, indicadores de giro de estoque e divergências, oferecendo maior suporte à tomada de decisão. A avaliação da eficiência demonstrou que o sistema se mostrou robusto em cenários de uso real, suportando múltiplos acessos simultâneos, leitura em dispositivos móveis de diferentes configurações e integração com banco de dados em nuvem (NeonDB PostgreSQL).

Do ponto de vista de rede, a arquitetura cliente-servidor hospedada no Railway com banco de dados NeonDB apresentou latência média de 150-300ms para requisições completas, com taxa de disponibilidade de 99.9%. A utilização de HTTPS com TLS 1.3 garantiu segurança na transmissão de dados, enquanto o connection pooling do NeonDB otimizou o uso de conexões TCP, reduzindo overhead de estabelecimento de conexões em aproximadamente 70%.

REFERÊNCIAS

- MANTHOU, V.; VLACHOPOULOU, M. Bar-code technology for inventory and marketing management systems: a model for its development and implementation. *International Journal of Production Economics*, v. 71, p. 157-164, 2001.
- PANGANIBAN, E.; BERMUSA, J. P. Simplified barcode-based point-of-sales and inventory management system with replenishment decision. *International Journal of Scientific & Technology Research*, v. 9, n. 3, Mar. 2020.
- JHA, A. Application of RFID and barcode technology in increasing efficiency of supply chain management and inventory management. *International Journal of Novel Research and Development*, v. 9, n. 4, abr. 2024.
- UY, R. C. Y. The state and trends of barcode, RFID, biometric and other AIDC methods and pharmacy automation. *PMC – NCBI*, 2015.
- UNLEASHED SOFTWARE. What is a barcode inventory system? How it works & how to implement. *Blog Unleashed*, 2023.
- KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet: uma abordagem top-down*. 8. ed. São Paulo: Pearson, 2021.
- RAILWAY. *Railway Documentation*. Disponível em: <https://docs.railway.app>. Acesso em: 2024.
- NEON. *Neon Documentation: Serverless Postgres*. Disponível em: <https://neon.tech/docs>. Acesso em: 2024.