

CENTRO UNIVERSITÁRIO DE BRASÍLIA (CEUB)
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
DISCIPLINA DE PROJETO INTEGRADOR II

Rafael Irvine
Raul Falluh
Rodrigo Castro
Rodrigo Lemos

SISTEMA DE COISAS DE GARAGEM

Brasília, Agosto de 2025

SUMÁRIO

Introdução

1.1 Objetivo do Projeto

1.2 Escopo

1.3 Público-Alvo

Requisitos do Sistema

2.1 Requisitos Funcionais

2.2 Requisitos Não Funcionais

Arquitetura e Tecnologias

3.1 Diagrama de Arquitetura

3.2 Tecnologias Utilizadas

Design e Interface do Usuário

4.1 Protótipo de Telas

4.2 Padrões de Design

Modelagem do Sistema

5.1 Diagrama de Casos de Uso

5.2 Diagrama de Classes

5.3 Diagrama de Sequência

5.4 Diagrama Entidade-Relacionamento

Desenvolvimento e Metodologia

6.1 Metodologia de Desenvolvimento

6.2 Estrutura do Código

6.3 Repositório de Código-Fonte

Gestão de Tarefas e Kanban

7.1 Uso do Kanban

7.2 Entrega do Kanban

Testes e Validação

8.1 Estratégia de Testes

8.2 Ferramentas de Teste

Implantação e Manutenção

9.1 Processo de Implantação

9.2 Plano de Manutenção

9.3 Versão Executável ou Deploy na Nuvem

9.4 Documentação Técnica

9.5 Manual do Usuário/Treinamento

Conclusão

Anexos

1. Introdução

1.1 Objetivo do Projeto

O Coisas de Garagem é uma plataforma web que transforma *garage sales* tradicionais em experiências digitais modernas. O sistema conecta vendedores e compradores, oferecendo uma forma prática, segura e eficiente de cadastrar itens usados, expor via QR Codes, e realizar o arremate (compra) com rapidez.

1.2 Escopo

- Cadastro de produtos com fotos e descrições.
- Geração automática de QR Codes para cada item.
- Busca e acesso ao produto por scanner de QR Code (câmera do dispositivo) ou inserção de código manual.
- Exibição de detalhes do produto ao comprador e fluxo de arremate simplificado.
- Marcação automática de item como vendido após confirmação de compra.
- Dashboard para vendedores com estatísticas de vendas e receita.

- Geração de PDF contendo os QR Codes dos itens para impressão.
- Histórico de compras para compradores.

1.3 Público-Alvo

- Vendedores (pessoas físicas) que desejam organizar e vender itens usados com praticidade e controle.
- Compradores que procuram descobrir itens em *garage sales* de forma rápida, com acesso direto via QR Code e histórico de compras organizado.

2. Requisitos do Sistema

2.1 Requisitos Funcionais

Para Vendedores

- RF-01: Cadastrar produtos com título, descrição, preço e fotos.
- RF-02: Gerar QR Code único por produto.
- RF-03: Visualizar dashboard com estatísticas (vendas, receita, itens vendidos/ativos).
- RF-04: Gerar PDF com QR Codes para impressão.
- RF-05: Excluir produtos de forma segura.

Para Compradores

- RF-06: Escanear QR Code com a câmera do dispositivo.
- RF-07: Buscar item por código manual.
- RF-08: Visualizar detalhes do produto (fotos, descrição, preço, status).
- RF-09: Realizar arremate informando dados pessoais mínimos.

- RF-10: Consultar histórico de compras.
- RF-11: Receber notificações em tempo real (quando disponíveis).

Fluxo de Venda (Regra de Negócio)

- RN-01: Vendedor cadastra o produto.
- RN-02: Sistema gera QR Code único.
- RN-03: Comprador escaneia ou insere o código manual.
- RN-04: Sistema exibe detalhes do produto.
- RN-05: Comprador realiza o arremate.
- RN-06: Produto muda automaticamente para Vendido.

2.2 Requisitos Não Funcionais

- Usabilidade: interface simples, fluxo curto de compra, mensagens claras.
- Responsividade: funcionamento adequado em diferentes tamanhos de tela (mobile-first).

- Compatibilidade: compatível e testado em Chrome, Firefox, Edge e Safari (versões recentes), bem como dispositivos Android e iOS.
- Desempenho: tempo de carregamento adequado para uso em rede móvel.
- Segurança: QR Codes únicos; proteção contra edição manual de códigos; tratamento seguro de dados pessoais.
- Observabilidade: registro de eventos essenciais para acompanhamento de uso (ex.: escaneamentos e arremates).

3. Arquitetura e Tecnologias

3.1 Diagrama de Arquitetura

3.2 Tecnologias Utilizadas

Frontend

- React
- QRCode.js — geração de QR Codes
- jsQR — leitura de QR Codes
- Font Awesome — ícones

(Outras camadas — backend, banco de dados, storage, serviços de fila, etc. — não foram definidas nesta versão inicial e permanecem em aberto.)

4. Design e Interface do Usuário

4.1 Protótipo de Telas

4.2 Padrões de Design

- Responsividade total.
- Animações suaves para indicar estados (carregando, sucesso, erro).
- Interatividade clara (feedback a cliques, *hover*, estados ativos).
- Arquitetura modular no frontend para facilitar manutenção e expansão.
- Acessibilidade: uso de labels, contraste adequado e navegação por teclado.

5. Modelagem do Sistema

5.1 Diagrama de Casos de Uso

5.2 Diagrama de Classes

5.3 Diagrama de Sequência

5.4 Diagrama Entidade-Relacionamento (ER)

6. Desenvolvimento e Metodologia

6.1 Metodologia de Desenvolvimento

6.2 Estrutura do Código

6.3 Repositório de Código-Fonte

7. Gestão de Tarefas e Kanban

7.1 Uso do Kanban

7.2 Entrega do Kanban

8. Testes e Validação

8.1 Estratégia de Testes

8.2 Ferramentas de Teste

9. Implantação e Manutenção

9.1 Processo de Implantação

9.2 Plano de Manutenção

9.3 Versão Executável ou Deploy na Nuvem

9.4 Documentação Técnica

9.5 Manual do Usuário/Treinamento

10. Conclusão

O Coisas de Garagem propõe digitalizar *garage sales* com foco em praticidade, segurança e experiência mobile. As funcionalidades centrais — cadastro de itens, QR Codes por produto, scanner integrado, arremate simples e dashboard do vendedor — formam um MVP sólido para validação com usuários reais. As seções em aberto deste documento servirão para detalhar modelagem, metodologia, testes e implantação à medida que o projeto evoluir.

11. Anexos

11.1 Compatibilidade

- Chrome (últimas versões)
- Firefox (últimas versões)
- Edge (últimas versões)
- Safari (últimas versões)
- Dispositivos móveis Android e iOS

11.2 Recursos Avançados (Planejados/Disponíveis)

Para Vendedores

- Gráficos de desempenho de vendas.
- Geração de PDF com QR Codes para impressão.
- Exclusão segura de produtos.

Para Compradores

- Scanner de QR Code com troca de câmera.
- Notificações em tempo real.
- Histórico de compras detalhado.

11.3 Próximas Melhorias (Roadmap)

- Autenticação de usuários.
- Integração com APIs de geolocalização.
- Sistema de avaliação de produtos.
- Compartilhamento em redes sociais.

11.4 Equipe Responsável

Nome	Função
Raul Falluh	Frontend Developer
Rodrigo Lemos	Backend Developer
Rodrigo Castro	Database Developer
Rafael Irvine	Backend Developer & Documentador