

Sistema de Gerenciamento de Estoque de Receitas

Alan R. da Costa

Análise e Desenvolvimento de Sistemas
Laboratório de Desenvolvimento de Software

Problema Identificado

- ▶ **Desperdício de ingredientes** devido à falta de controle preciso do estoque
- ▶ **Inconsistência na produção** de receitas por ausência de padronização
- ▶ **Dificuldade em calcular** quantas porções podem ser produzidas com o estoque disponível
- ▶ **Ausência de histórico** das produções realizadas
- ▶ **Falta de controle de acesso** em ambientes compartilhados

Objetivo do Projeto

- ▶ Controle preciso do estoque de ingredientes
- ▶ Padronização e armazenamento de receitas
- ▶ Cálculo automático de viabilidade de produção
- ▶ Histórico detalhado de atividades
- ▶ Controle de acesso por níveis de usuário
- ▶ Interface responsiva e intuitiva

Metodologia

Gestão Ágil - Scrum

- ▶ **Sprints de 2 semanas** para cada tarefa principal
- ▶ **Daily meetings** simulados através de autoavaliação diária
- ▶ **Sprint Reviews** ao final de cada etapa
- ▶ **Retrospectivas** para melhorias contínuas

Desenvolvimento Orientado a Objetos

- ▶ Aplicação dos **princípios SOLID**
- ▶ **Encapsulamento** de dados por usuário
- ▶ **Abstração** de funcionalidades em módulos
- ▶ **Polimorfismo** no tratamento de diferentes tipos de usuário

Requisitos Funcionais

- ▶ **RF01:** Sistema deve permitir cadastro e autenticação de usuários
- ▶ **RF02:** Usuários devem gerenciar ingredientes com controle de estoque
- ▶ **RF03:** Sistema deve permitir criação e edição de receitas
- ▶ **RF04:** Cálculo automático de viabilidade de receitas baseado no estoque
- ▶ **RF05:** Registro de execução de receitas com atualização automática do estoque
- ▶ **RF06:** Histórico detalhado de todas as atividades
- ▶ **RF07:** Diferentes níveis de permissão (Admin/Usuário)

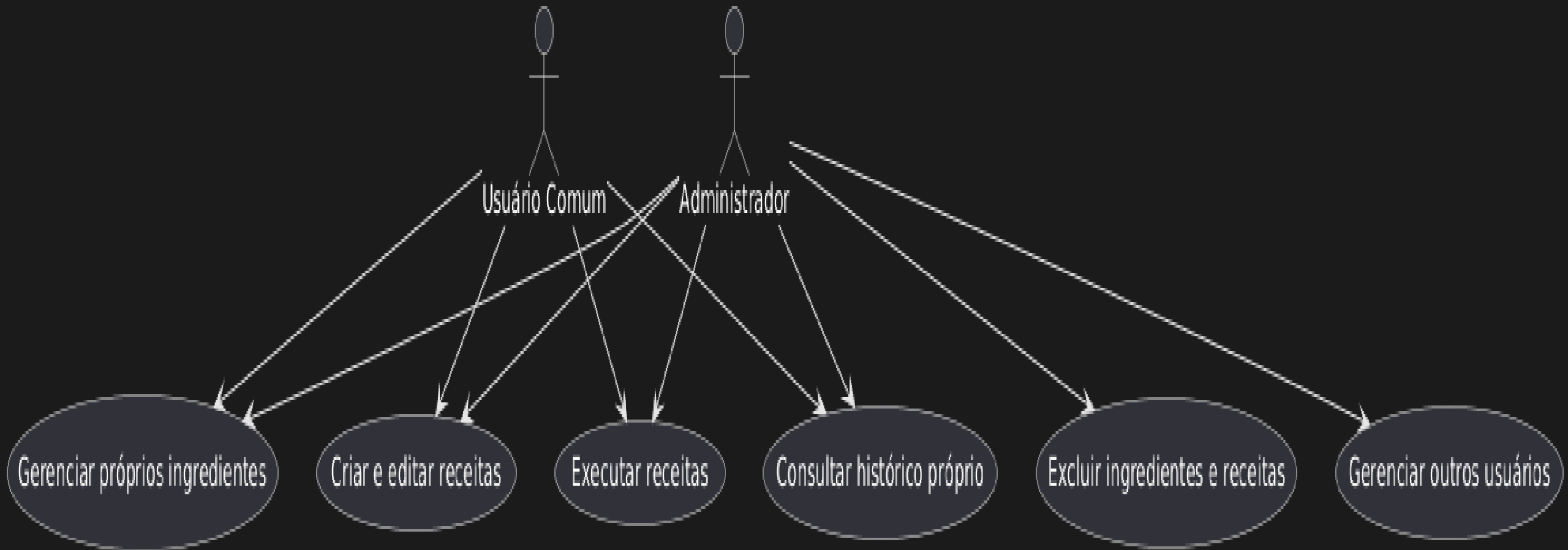
Requisitos Não Funcionais

- ▶ **RNF01:** Interface responsiva para desktop, tablet e mobile
- ▶ **RNF02:** Persistência de dados local (localStorage)
- ▶ **RNF03:** Tempo de resposta inferior a 2 segundos
- ▶ **RNF04:** Compatibilidade com navegadores modernos
- ▶ **RNF05:** Acessibilidade seguindo padrões WCAG

Arquitetura

- ▶ **Frontend:** HTML5 + CSS3 + JavaScript Vanilla
- ▶ **Armazenamento:** localStorage (navegador)
- ▶ **Arquitetura:** SPA (Single Page Application)

Diagramas Caso de Uso





Gerenciar Usuários

+ Adicionar Usuário

Administrador



Email: admin@sistema.com

Celular: (11) 99999-9999

Usuário: admin

Tipo: ADMIN

Alan



Email: alana@gamilc.om

Celular: a

Usuário: a

Tipo: USUÁRIO



Ingredientes

Receitas

Histórico

Ingredientes

Buscar ingredientes...

+ Adicionar Ingrediente

f



Unidade: Gramas

Estoque: 319 g

Farinha



Unidade: Gramas

Estoque: 1000 g

Ovos



Unidade: Unidade

Estoque: 30 un

Açúcar



Unidade: Gramas

Estoque: 1000 g

Nescau



Unidade: Gramas

Estoque: 1000 g

Essencia de Baunilha



Unidade: Mililitros

Estoque: 900 ml



Receitas

+ Adicionar Receita

f



Fazer



Descrição: f

Rendimento: 3

Pode fazer: 159 RECEITAS

Ingredientes:

f

2 g

Brownie



Fazer



Descrição: .

Rendimento: 25

Pode fazer: 1 RECEITA

Ingredientes:

Farinha

400 g

Ovos

6 un

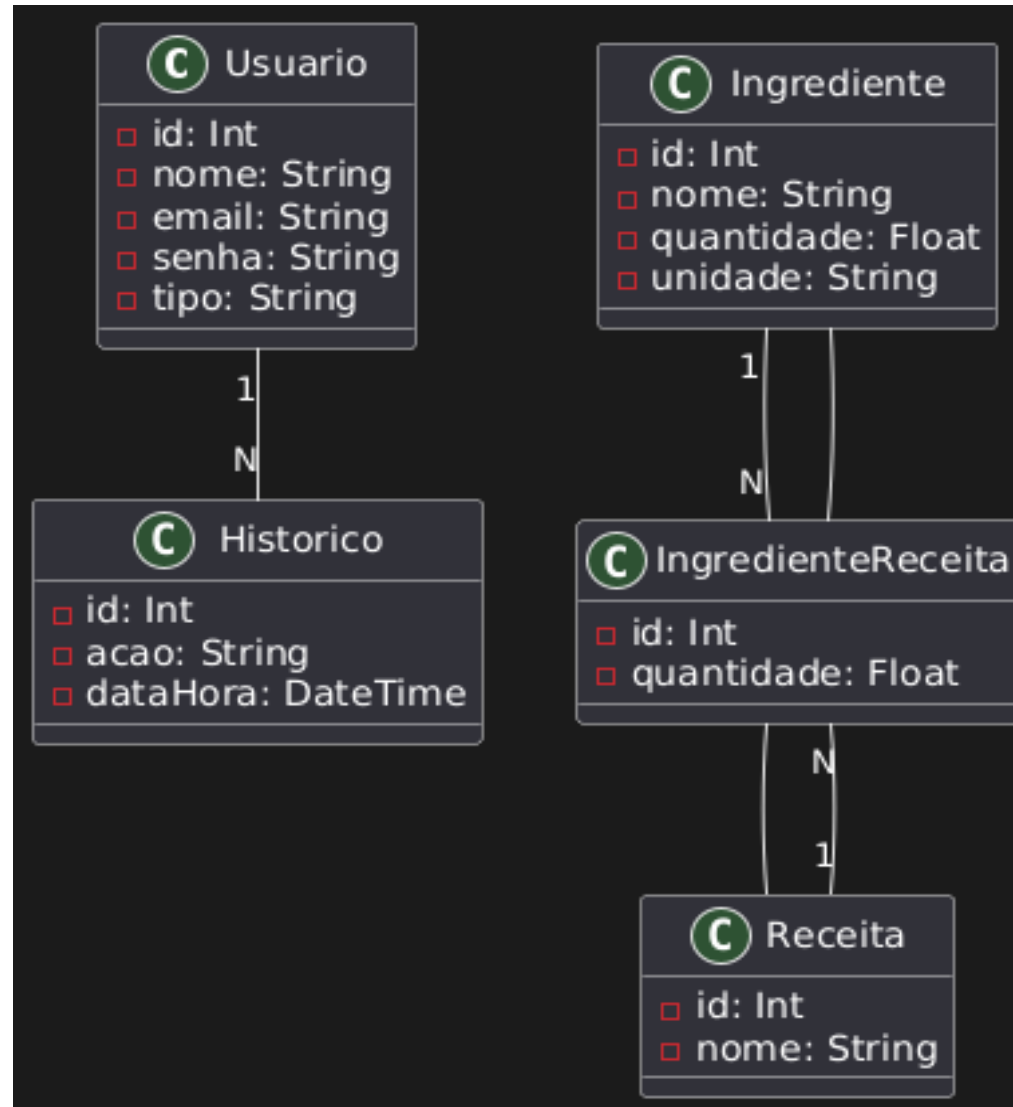
Açúcar

450 g

Nescau

600 g

Diagramas Classe



Diagramas Atividades

Brownie



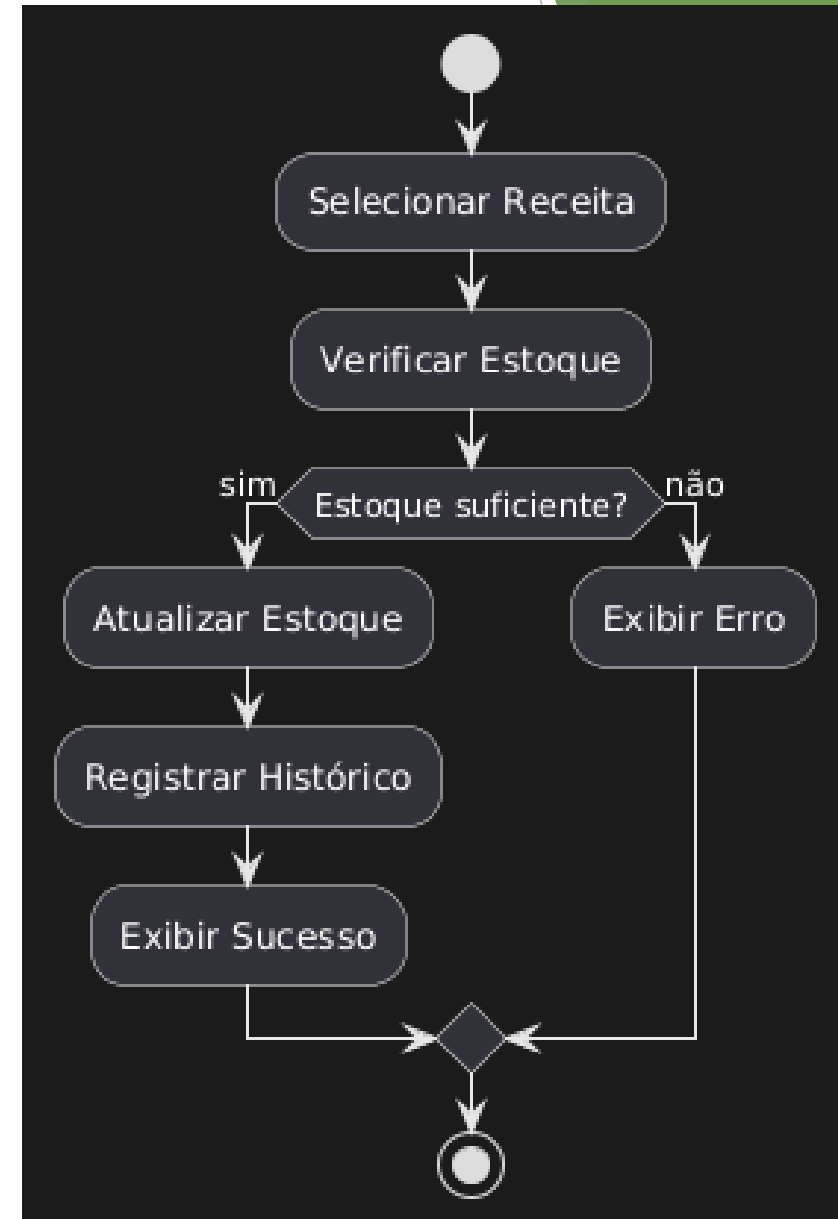
Descrição: .

Rendimento: 25

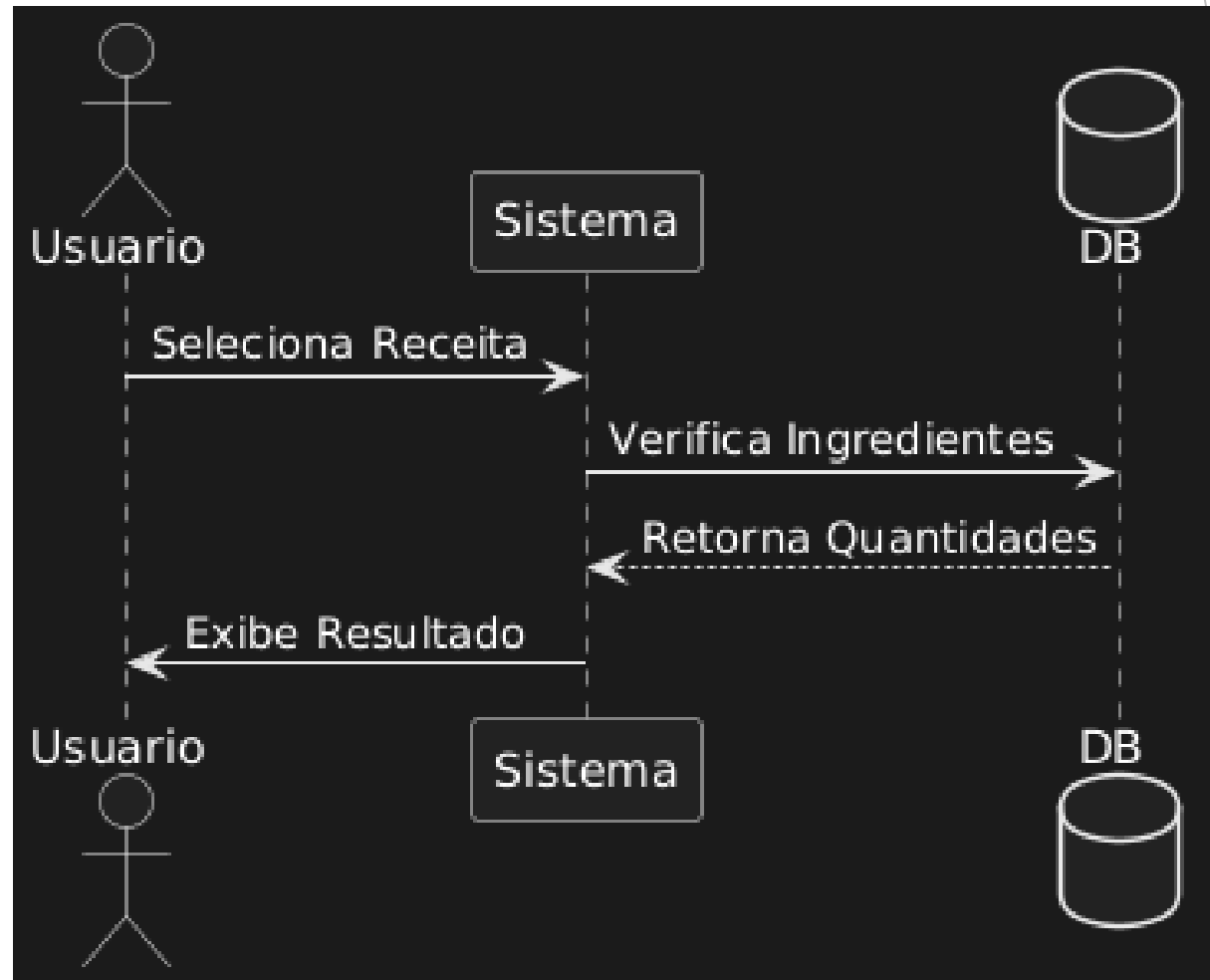
Pode fazer: 1 RECEITA

Ingredientes:

Farinha	400 g
Ovos	6 un
Açúcar	450 g
Nescau	600 g



Diagramas Sequencia



Seleção e Configuração de Frameworks e Tecnologias

Frontend Framework: JavaScript Vanilla

- ▶ **Justificativa:** Máxima compatibilidade, zero dependências, performance otimizada
- ▶ **Alternativas Consideradas:** React, Vue.js (descartados por complexidade desnecessária)

Persistência: localStorage

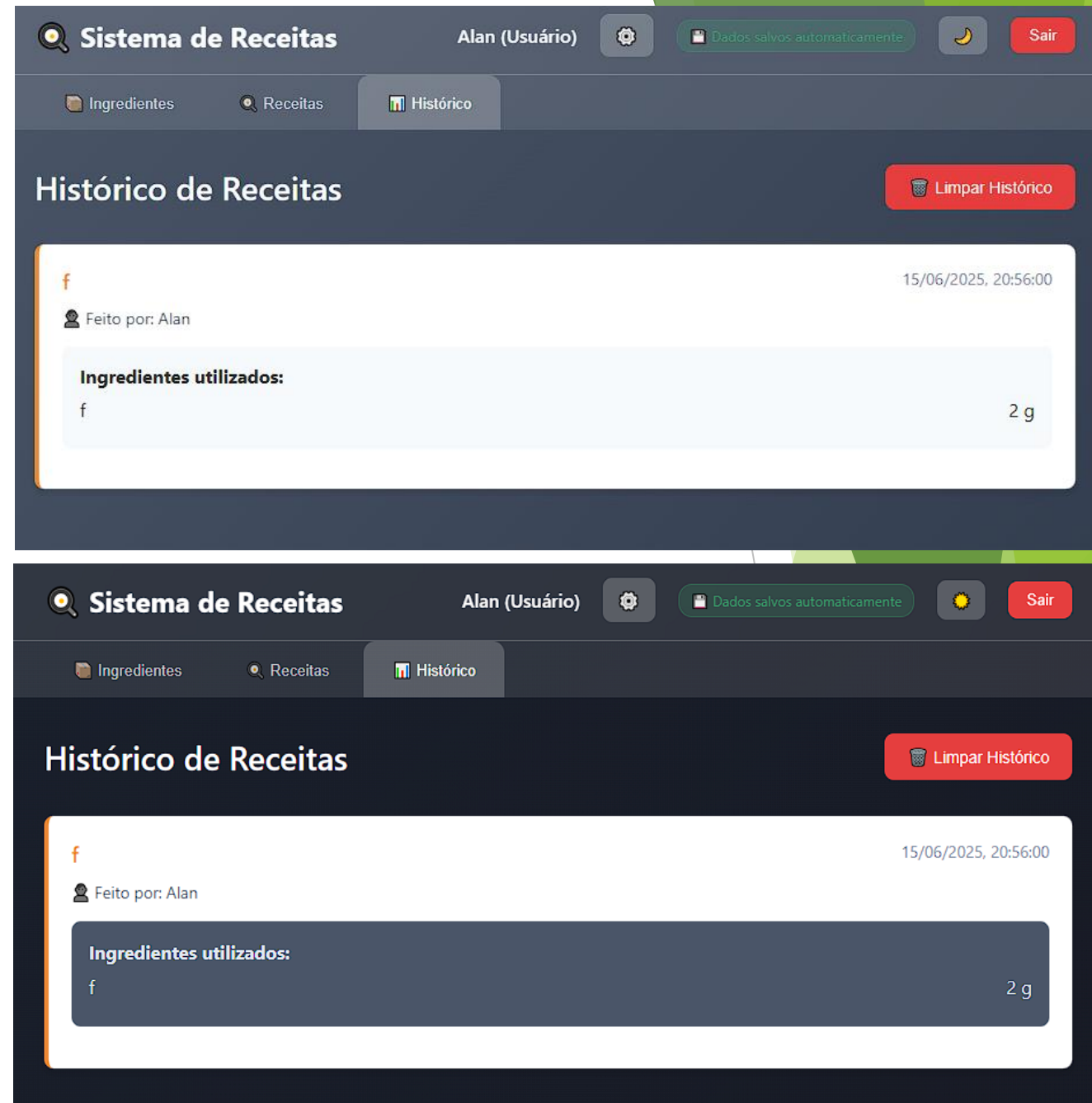
- ▶ **Justificativa:** Simplicidade, disponibilidade offline, sem necessidade de servidor
- ▶ **Alternativas Consideradas:** IndexedDB, WebSQL (localStorage suficiente para o escopo)

CSS Framework: CSS Vanilla com Variáveis

- ▶ **Justificativa:** Controle total sobre estilos, tamanho reduzido

Acessibilidade

- ▶ Labels semânticos em formulários
- ▶ Contraste adequado entre cores
- ▶ Navegação por teclado
- ▶ Feedback visual para ações



Persistência

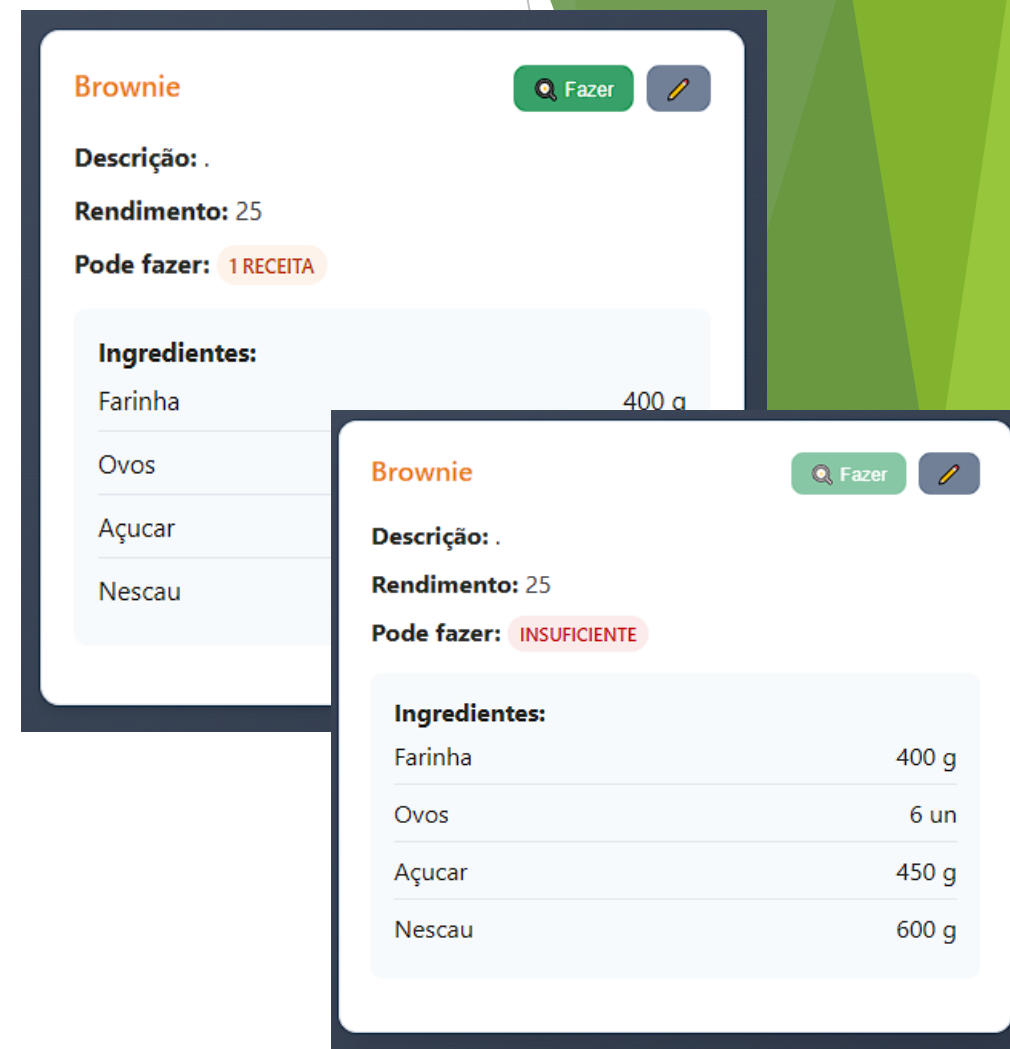
Modelo de Dados localStorage

```
class PersistenceManager {
  salvarDados() {
    const dados = {
      usuarios: dadosApp.usuarios,
      ingredientes: dadosApp.ingredientes,
      receitas: dadosApp.receitas,
      historico: dadosApp.historico,
      temaEscuro: dadosApp.temaEscuro
    };
    localStorage.setItem('sistemaReceitas_dados', JSON.stringify(dados));
  }

  carregarDados() {
    const dados = localStorage.getItem('sistemaReceitas_dados');
    if (dados) {
      return JSON.parse(dados);
    }
    return null;
  }
}
```

Testes Implementados

- ▶ **Testes Unitários:** Funções de cálculo e validação
- ▶ **Testes de Integração:** Fluxo completo de fazer receita
- ▶ **Testes de Interface:** Responsividade e interações
- ▶ **Testes de Persistência:** Salvar/carregar dados
- ▶ **Testes de Permissão:** Controle de acesso por usuário



Tecnologias Utilizadas

Desenvolvimento

- ▶ **Frontend:** HTML5, CSS3, JavaScript ES6+
- ▶ **Persistência:** localStorage, JSON
- ▶ **Arquitetura:** SPA (Single Page Application)

Ferramentas de Desenvolvimento

- ▶ **IDE:** Visual Studio Code
- ▶ **Servidor Local:** Live Server

Gestão de Projeto

- ▶ **Metodologia:** Scrum

Testes e Qualidade

- ▶ **Testes:** JavaScript nativo (console.assert)
- ▶ **Validação:** HTML5 Validator
- ▶ **Performance:** Lighthouse
- ▶ **Acessibilidade:** WAVE

ChefStock

Sistema de Gerenciamento de Estoque de Receitas

Obrigado pela atenção!

Estou disponível para perguntas

Alan R.