

# Relatório de Desenvolvimento — FoodSafe

**Projeto:** Implementação de Avatar com Foto no Drawer

**Aluno:** João Vítor Herzer De Sousa

**Turma:** CC4 / Aplicações para dispositivos móveis

**Data:** 23 de Outubro de 2025

---

## Resumo:

Este projeto implementou um sistema de gerenciamento de foto de perfil no aplicativo **FoodSafe**, seguindo os requisitos da **LGPD** e boas práticas de desenvolvimento **Flutter**.

A solução permite aos usuários **adicionar, visualizar e remover fotos de perfil**, com **armazenamento local** e **fallback para iniciais do nome**.

O desenvolvimento foi **assistido por IA**, resultando em uma implementação robusta, com **testes completos** e considerações de **privacidade, acessibilidade e desempenho**.

---

## Introdução

### Contexto do PRD

O projeto teve como objetivo expandir a funcionalidade do *drawer* do FoodSafe, substituindo o **CircleAvatar** básico (com iniciais) por um **sistema completo de gestão de fotos de perfil**.

### Objetivos principais:

- Implementação de upload, seleção e remoção de fotos
- Conformidade com a LGPD
- Fallback automático para iniciais
- Preparação para futura integração com o **Supabase Storage**

### Objetivos Específicos

- Desenvolver interface intuitiva para gestão de fotos
  - Implementar armazenamento local seguro
  - Garantir performance ( $\leq 100\text{ms}$  para carregamento)
  - Estabelecer suíte de testes completa
  - Assegurar conformidade com LGPD
- 

## Metodologia

### Uso de IA

O desenvolvimento foi assistido pelo **GitHub Copilot**, utilizando uma abordagem estruturada.

### Análise Inicial

- Exploração do código existente
- Identificação de pontos de integração
- Planejamento da arquitetura

### Desenvolvimento Iterativo

- Geração de código assistida
- *Review* e refinamento das implementações
- Correção de bugs em tempo real

### Ferramentas Utilizadas

- **GitHub Copilot** – geração e refatoração de código
  - **VS Code** – IDE principal
  - **Flutter SDK** – desenvolvimento e testes
-

# Desenvolvimento

## Decisões Arquiteturais

### Estrutura de Camadas:

- Repository Pattern para abstração de dados
- Service Layer para lógica de negócios
- Widget Composition para UI reutilizável

## Iterações Principais

### Primeira Iteração

- Implementação básica do `UserAvatar`
- Integração com `SharedPreferences`
- Correção de bugs de visibilidade da foto

### Segunda Iteração

- Correção de persistência de fotos
- Implementação de estado reativo
- Adição de testes unitários

### Terceira Iteração

- Implementação LGPD
- Refinamento da UX
- Expansão de testes

---

## Validações

### Testes Implementados

### Testes Unitários:

- `local_photo_store_test.dart`
- `preferences_service_test.dart`
- `profile_repository_test.dart`

### Testes de Widget:

- `user_avatar_test.dart`
- `user_avatar_error_test.dart`

### Acessibilidade (A11Y)

- Implementação de `semanticsLabel`
- Área de toque  $\geq 48dp$
- Suporte a *screen readers*

### LGPD

- Armazenamento local por padrão
- Consentimento explícito para futura integração em nuvem
- Funcionalidade de remoção de dados

### Desempenho

- Otimização no carregamento de imagens
- Compressão eficiente
- Gestão de estado otimizada

---

## Recursos Flutter Utilizados

## Widgets Principais

- `CircleAvatar`
- `ValueListenableBuilder`
- `ImageProvider`
- `BottomSheet`

## Pacotes

- `shared_preferences`
- `path_provider`
- `image_picker`
- `mockito` (*testes*)

## Permissões

- Câmera
- Armazenamento
- Galeria de fotos

---

## Resultados e Discussão

### Sucessos

- Implementação completa do sistema de fotos
- Cobertura de testes robusta
- Conformidade com LGPD
- Performance dentro dos parâmetros esperados

## Limitações

- Necessidade de integração futura com Supabase
  - *Crop* de imagem básico
  - Sincronização multi-dispositivo pendente
- 

## Conclusão e Próximos Passos

### Conclusões

O projeto atingiu seus objetivos principais, entregando uma **solução robusta e bem testada** para gestão de fotos de perfil no FoodSafe, seguindo boas práticas Flutter e diretrizes LGPD.

### Próximos Passos

- Implementação do **Supabase Storage**
  - Adição de **recursos avançados de edição de imagem**
  - Implementação de **sincronização multi-dispositivo**
  - Expansão da **telemetria opt-in**
- 

## Apêndice

### Principais Interações com IA

- Análise inicial de código e planejamento
- *Debugging* de visibilidade de fotos
- Implementação de testes
- Refinamento de conformidade LGPD

### Exemplos de Código Gerados

[Inclua aqui screenshots ou links para as interações mais significativas com o Copilot.]

## **Métricas de Desenvolvimento**

- **Arquivos alterados:** ~10
- **Linhas de código modificadas:** ~500
- **Testes implementados:** 15+