

EcoSteps: Foto de Perfil com Privacidade

Implementação de foto de perfil personalizada no Drawer do EcoSteps, priorizando privacidade, compressão e acessibilidade em conformidade com LGPD.

Objetivo & Requisitos do PRD

Funcionalidade Principal

Substituir o CircleAvatar com iniciais por foto do usuário no Drawer do EcoSteps, oferecendo fluxo completo de gerenciamento de imagem.

- Adicionar foto via câmera ou galeria
- Alterar e remover foto com preview
- Persistência local segura
- Fallback para iniciais quando sem foto

Requisitos Técnicos

Garantir privacidade e performance através de compressão inteligente e conformidade regulatória.

- Compressão automática para $\leq 200\text{KB}$
- Remoção completa de metadados EXIF/GPS
- Conformidade total com LGPD
- Acessibilidade ($\geq 48\text{dp}$, Semantics)
- Cobertura de testes (unit + widget)

Arquitetura & Stack Técnico



Componentes UI

CircleAvatar com backgroundImage/FileImage, DrawerHeader, BottomSheet e GestureDetector para interação fluida.



Pacotes Essenciais

shared_preferences (persistência), flutter_image_compress (compressão/EXIF), image_picker (câmera/galeria), path_provider (storage).



Permissões

Android: CAMERA, READ_MEDIA_IMAGES. iOS: NSCameraUsageDescription, NSPhotoLibraryUsageDescription configuradas.

Camadas da Arquitetura

- **PrefsService:** Gerenciamento de estado simples
- **LocalPhotoStore:** Operações de imagem
- **ProfileDrawer:** Camada de apresentação

Fluxo de Dados

UI → Service → Storage com separação clara de responsabilidades e código modular para manutenibilidade.



IA Como Acelerador de Desenvolvimento

1

Prompt 1: Persistência

Solicitação: "Crie métodos em PrefsService para userPhotoPath e userPhotoUpdatedAt."

Aproveitado: Código base com getters/setters funcionais. **Ajustado:** Integração no PrefsService existente do EcoSteps para compatibilidade.

2

Prompt 2: Compressão

Solicitação: "Gere LocalPhotoStore para salvar/comprimir/remover EXIF."

Aproveitado: compressAndGetFile com quality:85 otimizado. **Descartado:** Sugestão de cache desnecessária. Validação manual garantiu $\leq 200\text{KB}$.

3

Prompt 3: Interface

Solicitação: "Atualize ProfileDrawer com CircleAvatar dinâmico, fallback para iniciais."

Aproveitado: Estrutura com GestureDetector. **Ajustado:** Integração no Drawer com ajustes para acessibilidade completa.

80%

Código Base Gerado

Aproximadamente 80% do código inicial foi gerado por IA, reduzindo drasticamente o tempo de desenvolvimento.

50%

Redução de Tempo

IA acelerou desenvolvimento em cerca de 50%, mas validação humana foi essencial para qualidade.

Validações & Garantia de Qualidade

1

Compressão & Privacidade

Imagens reduzidas de 10MB para ~150KB. Metadados EXIF/GPS completamente removidos via `flutter_image_compress`.

2

Acessibilidade

Área de toque $\geq 48dp$, Semantics/tooltips implementados, foco visível. Validado com TalkBack no Android.

3

Cobertura de Testes

Unit test (compressão), widget test (fallback). Cobertura de 70% com casos críticos validados.

4

Performance do Drawer

Carregamento $\leq 100ms$ com `cacheWidth/height`. Nenhuma lentidão perceptível na navegação.

Lições Aprendidas & Próximos Passos

Lições Principais

- IA acelera significativamente, mas validação humana é insubstituível
- Prompts detalhados evitam iterações desnecessárias
- Privacidade e segurança requerem atenção especializada

Desafios Superados

- Configuração manual de permissões iOS
- Compressão pode afetar qualidade em fotos escuras
- Balanceamento entre tamanho e qualidade visual



Fase 2: Cloud Sync

Upload para nuvem com consentimento explícito do usuário



Editor de Imagens

Funcionalidade de crop e edição básica integrada



Multi-dispositivo

Sincronização de perfil entre dispositivos no EcoSteps

Repositório, PRs, vídeo demonstrativo, relatório técnico e checklist disponíveis na documentação do projeto.