

Enunciado — Implementação do PRD “Avatar com Foto no Drawer” com uso de IA (FoodSafe) — COMPLETO

Atualização: inclui **apresentação obrigatória para a turma** e todos os itens consolidados (escopo, entregáveis, critérios, checklist, diretrizes técnicas e modelo de relatório).

1) Objetivo

Implementar, no app **FoodSafe**, o PRD do **Avatar com Foto no Drawer** utilizando **assistentes de IA** durante todo o ciclo (planejamento, geração/ajuste de código, testes, revisão e documentação). O resultado deve **substituir o CircleAvatar com iniciais** pela **foto do usuário**, mantendo **fallback** para iniciais e respeitando **LGPD** e **Acessibilidade** (Material 3).

2) Escopo mínimo a entregar (MVP)

1. **Fluxo completo de foto local:** adicionar, alterar e remover foto (câmera e galeria), com **preview** antes de salvar.
2. **Persistência local:** guardar `userPhotoPath` (e `userPhotoUpdatedAt`) em `SharedPreferences`; salvar o arquivo no diretório do app; carregar no Drawer.
3. **Qualidade de imagem: compressão** (meta $\leq \sim 200\text{KB}$) e **remoção de EXIF/GPS**; uso de `cacheWidth/height` ao exibir.
4. **A11Y & UX:** área do avatar clicável com $\geq 48\text{dp}$, `semanticsLabel` /tooltip, foco visível; **mensagem curta de privacidade** (“sua foto fica neste dispositivo, você pode remover quando quiser”).
5. **Testes:** pelo menos **1 unit test** (persistência/armazenamento) e **1 widget test** (fallback e exibição da foto).

Fase 2 — opcional (não obrigatória aqui): upload para nuvem (Supabase Storage) com consentimento, crop/editor, sync multi-dispositivo.

3) Como usar IA (obrigatório)

Você **deve** empregar IA (Copilot/ChatGPT/Gemini/Code Assistant etc.) para: - **Planejar** os PRs (#1 Infra/Chaves; #2 UI Drawer; #3 Seleção/Compressão; #4 Testes & A11Y); - **Gerar/refatorar** código (services/repository/store), **escrever testes** e **mensagens de UX/A11Y**; - **Checar** permissões Android/iOS e montar **checklists**; - **Validar** critérios de aceite (Given/When/Then) e riscos.

Registre as interações relevantes com IA (prompts e respostas). Trechos grandes podem ser resumidos, mas inclua **prints/links/arquivos**.

4) Entregáveis

1. **Repositório** (link): commits claros e PRs pequenos (idealmente 4 PRs conforme acima).
 2. **Relatório Final (PDF/MD)** — “Uso de IA na implementação do PRD” (estrutura sugerida no item 7).
 3. **Demonstração em vídeo (2–3 min)**: adicionar/trocar/remover foto; Drawer com fallback; mensagem de privacidade; menção rápida às permissões.
 4. **Checklist de conformidade** preenchido (item 6.4).
 5. **Apresentação para a turma (obrigatória)**:
 6. **Tempo**: 5–7 min por equipe/aluno + 3 min de Q&A.
 7. **Formato: ao vivo** (demo no emulador/dispositivo) + **slides curtos** (4–6 lâminas) cobrindo: 1) **Objetivo & PRD** (resumo em 4–6 bullets); 2) **Arquitetura/Recursos Flutter usados** (widgets, pacotes, permissões, padrão de estado); 3) **Uso de IA** (2–3 prompts representativos; o que foi aproveitado/ajustado/descartado e por quê); 4) **Validações** (compressão/EXIF, A11Y, testes, desempenho do Drawer); 5) **Lições aprendidas & próximos passos** (Supabase Storage, crop etc.).
 8. **Evidências**: slide final com **links** (repo, PRs, vídeo, relatório e checklist).
-

5) Critérios de avaliação (0–100)

- **Implementação funcional (40 pts)**: fluxos add/troca/remover, fallback sólido, persistência local, fluidez do Drawer.
- **Qualidade técnica (20 pts)**: compressão/EXIF, estrutura (Repository/Store/Service), permissões Android/iOS, testes automatizados.
- **Evidências do uso de IA (20 pts)**: qualidade dos prompts, senso crítico ao aproveitar/ajustar, rastreabilidade das decisões.
- **Relatório reflexivo (15 pts)**: profundidade, dificuldades & ações, correções e validações, aprendizados.
- **Apresentação/Demo (5 pts)**: clareza dos slides, foco no PRD, domínio na demo, Q&A.

Bônus até +5 pts por métricas simples de performance (tempo extra do Drawer) e A11Y adicional.

6) Diretrizes técnicas

6.1 Arquitetura sugerida

- `ProfileRepository`: orquestra `get/set/remove` da foto e dados de perfil.
- `LocalPhotoStore`: salvar/ler/apagar arquivo, **comprimir** e **remover EXIF**.
- `PreferencesService`: ler/gravar `userPhotoPath` e `userPhotoUpdatedAt` (`SharedPreferences`).
- **Gerência de estado**: `ChangeNotifier` / `Provider` / `Riverpod` / `Bloc` (padrão já adotado no projeto).

6.2 UI & UX

- **Drawer Header**: `CircleAvatar(backgroundImage: ImageProvider?)`; `child` com iniciais **apenas no fallback**.

- **Ação:** ícone de editar (ou toque no avatar) abrindo **BottomSheet/Dialog** com **Câmera, Galeria, Remover** (se houver foto).
- **Carregamento:** `Image.file(..., cacheWidth: 256, cacheHeight: 256)` ou equivalente para uso eficiente de memória.

6.3 Permissões

- **Android:** `CAMERA`; `READ_MEDIA_IMAGES` (API 33+); para APIs antigas, `READ_EXTERNAL_STORAGE` (quando necessário). Ajustar o `AndroidManifest.xml` e *scoped storage* conforme a versão.
- **iOS:** `NSCameraUsageDescription` e `NSPhotoLibraryUsageDescription` no `Info.plist`.

6.4 Checklist de conformidade (preencha e entregue)

- [] Adicionar foto (câmera/galeria) funciona
- [] Remover foto apaga arquivo local e limpa preferências
- [] Fallback para iniciais quando sem foto **ou** em erro
- [] Compressão $\leq \sim 200\text{KB}$ (ou justificativa técnica)
- [] EXIF/GPS removido
- [] Drawer sem lentidão perceptível (meta: $+\leq 100\text{ms}$)
- [] Ações acessíveis ($\geq 48\text{dp}$, rótulos/semantics, foco)
- [] 1 unit test e 1 widget test passando
- [] Relatório com prompts/respostas/decisões incluído
- [] Slides prontos e apresentados para a turma

6.5 Pacotes (exemplos)

- `image_picker` (câmera/galeria)
- `flutter_image_compress` (ou similar) para compressão/remover metadados
- `shared_preferences` (persistência das chaves)
- (Opcional Fase 2) `cached_network_image`, `image_cropper`

7) Modelo de estrutura do Relatório (sugestão)

1. **Capa** (Projeto, aluno, turma, data)
 2. **Resumo** (5–7 linhas)
 3. **Introdução** (PRD e objetivos)
 4. **Metodologia** (como usou IA, ferramentas)
 5. **Desenvolvimento** (decisões, prompts, iterações, correções)
 6. **Validações** (testes, A11Y, LGPD, desempenho)
 7. **Recursos Flutter** usados (widgets, pacotes, permissões)
 8. **Resultados e discussão** (o que funcionou/limitações)
 9. **Conclusão e próximos passos**
 10. **Apêndice** (prints/links de prompts e respostas da IA)
-

8) Critérios de aceite (Given/When/Then)

1. **Given** usuário sem foto; **When** toca no avatar e escolhe Galeria e confirma; **Then** o Drawer exibe a foto e persiste localmente.
 2. **Given** usuário com foto; **When** toca em **Remover foto** e confirma; **Then** volta a iniciais e o arquivo é apagado.
 3. **Given** foto grande (>10MB); **When** confirma o envio; **Then** o app **comprime** para $\leq \sim 200\text{KB}$ (ou valor justificado), remove EXIF e exibe sem travar.
-

9) Entrega

- **Prazo:** conforme agenda da disciplina.
- **Formato:** link do repositório (com PRs/commits), **Relatório** (PDF/MD), **Vídeo** (2-3 min), **Checklist** preenchido e **slides** da apresentação.
- **Nome do release/tag:** `avatar-photo-drawer-mvp`.

Dúvidas técnicas e de escopo: registre no README os **trade-offs** assumidos e as justificativas.