

PRD — Avatar com Foto no Drawer (FoodSafe)

1) Resumo executivo

Adicionar suporte a **foto do usuário** no Drawer do app FoodSafe, substituindo o CircleAvatar atual com iniciais. A solução deve respeitar a **LGPD**, permitir **upload/seleção** de imagem, **compressão**, **remoção** e **fallback** para iniciais quando não houver foto, além de prever uma **evolução opcional** para armazenamento em nuvem (Supabase Storage) com consentimento explícito.

2) Objetivos

- Exibir a foto do usuário no Drawer e em locais correlatos do app.
- Permitir que o usuário **adicione/altere/remova** sua foto com experiência clara e acessível.
- Tratar **privacidade** e **consentimento**: local por padrão; nuvem opcional com aceite.
- Garantir **desempenho** (cache, compressão) e **qualidade** (sem alongar tempo de abertura do Drawer).

Métricas de sucesso

- $\geq 95\%$ de carregamentos do Drawer em ≤ 100 ms adicionais vs. baseline (cache local).
 - 0 crash reports originados da tela/fluxo de foto em 7 dias após rollout.
 - $\geq 80\%$ dos usuários completam o fluxo de adicionar foto sem erro na primeira tentativa.
-

3) Contexto e referências internas

- MVP atual armazena `userName` / `userEmail` em **SharedPreferences** e desenha iniciais no **CircleAvatar**.
 - Guia vigente: Perfil, Drawer e Consentimentos (Flutter + LGPD).
 - Roadmap institucional inclui **Supabase Storage**; erro anterior em função SQL será considerado no plano de risco.
-

4) Escopo (MVP)

Incluso 1. UI para **ver/definir/alterar/remover** foto do perfil. 2. Armazenamento **local** da imagem (path + cache) e chave em SharedPreferences (`userPhotoPath`). 3. **Fallback automático** para iniciais quando não houver foto ou em erro. 4. **Compressão** básica (máx. 512 x 512 px; qualidade ~80; remoção de metadados EXIF sensíveis). 5. A11Y: `semanticsLabel` / `alt text`, foco, área de toque $\geq 48dp$. 6. Testes: unit (repositório) + widget (Drawer renderiza foto | iniciais).

Fora do MVP (elegíveis para fase 2) - Upload para **Supabase Storage** com URL pública/assinada. - **Crop** avançado (quadrado/circular, zoom) e editor. - Multi-device sync da foto via backend.

5) Não-objetivos

- Não implementar backend novo neste ciclo.
 - Não coletar foto sem consentimento; não ativar upload automático para nuvem no MVP.
-

6) Perfis e jornadas

- **Estudante/Usuário final:** deseja personalizar sua conta com uma foto.
- **Professor/Administrador** (futuro): poderá ver foto via painel (fora do MVP).

Fluxos principais

1. **Adicionar foto:** Drawer → toque no avatar → escolher **Câmera** ou **Galeria** → preview → salvar.
2. **Trocar foto:** Drawer → menu “Alterar foto” → repetir fluxo de seleção.
3. **Remover foto:** Drawer → “Remover foto” → confirmar → voltar a iniciais.

7) UX & UI (guias práticos)

- **Drawer Header:** `CircleAvatar` com `backgroundImage: ImageProvider?` e `child` com iniciais para fallback.
 - **Ação rápida:** ícone de editar (lápis) sobreposto ao avatar (botão acessível).
 - **Dialog de escolha:** `BottomSheet` ou `Dialog` com opções: “Câmera”, “Galeria”, “Remover foto” (quando houver).
 - **Aviso de privacidade** (texto curto): “Sua foto fica apenas neste dispositivo. Você pode remover quando quiser.” (se nuvem desativada).
 - **Erro:** mensagem breve com opção “Tentar novamente”.
-

8) Dados e armazenamento

Chaves (SharedPreferences)

- `userName: string`
- `userEmail: string`
- `userPhotoPath: string | null` (caminho local)
- (Fase 2) `userPhotoUrl: string | null` (URL remota – Supabase)
- `userPhotoUpdatedAt: int | null` (epoch ms para invalidar cache)

Arquivos

- Diretório app (Documentos/Aplicativo). Nome: `avatar.jpg` (ou `.webp` se compressão suportar).
 - Tamanho alvo pós-compressão: $\leq 200 \text{ KB}$ (preferir WebP quando disponível).
 - Remover **EXIF** (especialmente GPS) antes de persistir.
-

9) Privacidade, LGPD e consentimentos

- **Base legal:** consentimento para dado pessoal (imagem) — opção **local** (padrão, sem compartilhamento).
 - **Transparência:** texto curto no fluxo + link para Políticas.
 - **Revogação:** “Remover foto” apaga arquivo local e limpa chave(s).
 - **Nuvem (fase 2):** só habilitar após aceite explícito (checkbox/termo). Sem aceite, **não** enviar.
-

10) Arquitetura & técnica (Flutter)

Camadas - `ProfileRepository` (abstração: `getPhoto()`, `setPhoto(File)`, `removePhoto()`, `getName()`, `getEmail()` ...) - `LocalPhotoStore`: salva arquivo, comprime, remove EXIF, retorna path. - `PreferencesService`: grava/lê `userPhotoPath` e metadados. - `AvatarController` (`ChangeNotifier`/`Riverpod`/`Bloc`): estado reativo do avatar.

UI - `Drawer` consome `AvatarController`. Se `photoProvider != null`, usa `backgroundImage`; senão, iniciais. - Usar `Image.file` com `cacheWidth/height` (p.ex. 256) para reduzir consumo.

Permissões - Android: `android.permission.CAMERA` (se usar câmera), `READ_MEDIA_IMAGES` (API 33+), `READ_EXTERNAL_STORAGE` (APIs antigas). - iOS: `NSCameraUsageDescription`, `NSPhotoLibraryUsageDescription` no Info.plist.

Bibliotecas sugeridas (MVP) - `image_picker` (câmera/galeria) - `flutter_image_compress` ou suporte nativo a WebP/HEIF quando disponível - (Fase 2) `cached_network_image` (URL remota) - (Fase 2) `image_cropper` (edição/corte)

11) Evolução opcional (Supabase Storage)

- Bucket: `user-avatars` (privado). Upload com `content-type` correto e política de acesso mediante token.
 - Estratégia de URL: **assinada (tempo limitado)**; cache + `If-None-Match`.
 - Vínculo com perfil do usuário (id/suid) para path: `avatars/{userId}.webp`.
 - Limpar versões antigas ao atualizar (ou manter versionamento controlado).
-

12) Estados de erro & fallback

- Falha ao abrir câmera/galeria → `snackbar` com `retry`.
 - Falha na compressão → salvar original se ≤ 1 MB; senão, informar erro.
 - Falha ao ler arquivo (path inválido) → limpar preferência e voltar a iniciais.
-

13) Performance

- Carregar provider do avatar **preguiçosamente** (lazy) após o header do `Drawer` montar.
- Reduzir dimensões na leitura (`cacheWidth/height`).

- Evitar trabalho no thread UI (usar `compute` /isolates para compressão quando necessário).
-

14) Telemetria (opt-in)

- `profile_photo_added`, `profile_photo_removed`, `profile_photo_changed`, com motivo (camera/gallery), tempo do fluxo e resultado (ok/erro). **Somente eventos agregados, sem imagem.**
-

15) Testes & critérios de aceite

Unit

- `LocalPhotoStore`: salva, comprime, remove metadados, retorna path.
- `PreferencesService`: grava/lê/limpa `userPhotoPath`.

Widget

- Drawer com `userPhotoPath` nulo → mostra **iniciais**.
- Drawer com `userPhotoPath` válido → mostra **foto** (sem child de iniciais).
- Ação "Remover foto" limpa estado e volta a iniciais.

A11Y

- Botão do avatar com `tooltip` / `semanticsLabel`.
- Área de toque $\geq 48dp$, foco visível.

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"; **Then** volta a iniciais e o arquivo é apagado.
 3. **Given** foto grande ($>10MB$); **When** confirma; **Then** app comprime para $\leq 200KB$ (ou próximo) e exibe sem travar.
-

16) Segurança

- Remover **EXIF/GPS**.
 - Não enviar a imagem a terceiros/serviços no MVP.
 - Em Fase 2 (nuvem): upload autenticado; URLs assinadas; política de remoção.
-

17) Riscos e mitigação

- **Permissões** quebrando build (Android/iOS): checklist de manifest/Info.plist em review.
- **Imagens enormes** → compressão + redução de dimensões.

- **Inconsistência de path** (arquivos removidos pelo SO): validar no boot do app e autocorrigir (limpar chave).
 - **Supabase Storage** indisponível (fase 2): manter local como padrão.
-

18) Plano de entrega (MVP em 4 PRs curtos)

1. **PR #1 — Infra & chaves:** `PreferencesService` + `ProfileRepository` + `LocalPhotoStore`; chaves `userPhotoPath` / `userPhotoUpdatedAt`.
2. **PR #2 — UI Drawer:** atualizar header do Drawer para ler provider da foto; fallback para iniciais.
3. **PR #3 — Fluxo de seleção:** ação no avatar (câmera/galeria/remover) + compressão + EXIF strip.
4. **PR #4 — Testes & A11Y:** unit + widget + checklist de acessibilidade.

(Fase 2 — Opcional) 5. **PR #5 — Supabase Storage:** client, upload, URL assinada, política, toggle por consentimento. 6. **PR #6 — Cropper & editor:** recorte quadrado/circular, rotação.

19) Checklist de revisão

- ☐ Drawer renderiza foto quando disponível
 - ☐ Remoção volta a iniciais e apaga arquivo
 - ☐ Compressão \leq 200KB e sem EXIF
 - ☐ A11Y (rotulagem, foco, 48dp)
 - ☐ Testes verdes (unit + widget)
 - ☐ Texto de privacidade revisado
-

20) Itens para decisão (PO/Arquiteto)

- **Formato preferido** pós-compressão: JPEG vs WebP (suporte alvo).
 - **Ativar nuvem na Fase 1?** (recomendação: não; deixar para Fase 2)
 - **Biblioteca de compressão** oficial do projeto.
-

21) Anexos & referências

- Guia “Perfil, Drawer e Consentimentos (Flutter + LGPD)”.
- Padrões de Acessibilidade (Material 3) e privacidade por padrão.