**Vibea+ — Especificação Técnica (v1.0)**

Documento técnico para orientar o desenvolvimento, testes e publicação do Vibea+ em **iOS** e **Android**, contemplando arquitetura, modelos de dados, APIs, permissões, políticas de loja, UX, segurança e planos de lançamento.

**0) Sumário rápido**

* **Stack sugerida:** Flutter (Dart) + Firebase (Auth, Firestore, Storage, Cloud Functions, FCM) + RevenueCat (assinaturas)
* **Arquitetura App:** Camadas limpas (Domain / Data / Presentation), padrão **MVVM** com **Riverpod**/**Bloc**
* **Modos:** Deslizar · Próximos · Caminhos · Mensagens
* **Tema:** **Preto** como base (neumorfismo sutil), com cores de modo
* **Perfil:** **Gênero obrigatório** + preferências do que exibir
* **Privacidade e Segurança:** Regras Firestore, validações em Cloud Functions, moderação de mídia, controle de localização
* **Lojas:** App Store + Google Play (diretrizes, privacidade, dados, billing)

**1) Objetivo & Escopo**

Criar um aplicativo de relacionamento com **três modos de descoberta** e foco em **personalização, privacidade e performance**. O documento define **requisitos funcionais e não-funcionais**, **modelagem de dados**, **fluxos**, **permissões**, **integrações**, **testes** e **publicação** para iOS/Android.

**1.1 Requisitos chave (MVP)**

* Cadastro por **telefone** e/ou redes sociais
* Perfil completo: nome, idade, **gênero (obrigatório)**, preferências de exibição por gênero, fotos (5–6), bio, objetivos
* Modos: **Deslizar**, **Próximos** (tempo real), **Caminhos** (rastreamento contínuo), **Mensagens**
* Notificações: match, cruzamentos, mensagens
* Filtros básicos (idade, distância, objetivos)
* Denúncia de perfis e bloqueio de usuários
* Monetização freemium (assinatura + consumíveis)

**1.2 Fora do MVP (planejados)**

* Verificação por documento/selfie
* Videochamada; Stories/Clipes
* IA para sugestões de abertura de conversa

**2) Stack & Integrações**

**2.1 Mobile**

* **Framework:** **Flutter** (3.x)
* **Arquitetura:** Clean Architecture (Domain/Data/Presentation)
* **Gerência de estado:** **Riverpod** (ou Bloc, caso preferido)
* **Navegação:** go\_router
* **UI kit:** Material 3 custom com **tema escuro** (preto) + tokens de cor por modo
* **Animações:** Impeller/Flutter animations
* **Mapas:** Google Maps SDK (Android/iOS) + Geocoding
* **Localização:** geolocator + foreground service (Android) + Background Modes (iOS)

**2.2 Backend & Serviços**

* **Firebase**: Auth, Firestore, Storage, Cloud Functions (Node 20), FCM, Analytics, Remote Config
* **Billing:** **RevenueCat** (unifica StoreKit e Play Billing; planos/experimentos)
* **Moderação de Mídia:** Cloud Vision API (rótulos/segurança) + bloqueio por política
* **CDN de imagens:** Firebase Storage + resizes em CF (thumbs)

**2.3 Observabilidade**

* **Crashlytics**, **Analytics** de eventos (funis de cadastro, match rate, retenção), **Performance**

**3) UX/UI (tema preto + neumorfismo sutil)**

**3.1 Fundos & elevação**

* Fundo **#000000**; cartões em #0B0B0B; sombras suaves para efeito “extrudado”
* Borda 1–2 px com opacidade baixa para destacar contornos

**3.2 Cores por modo (tokens)**

* **Deslizar:** acento cinza-grafite ~#1F1F1F (foco no perfil)
* **Próximos:** acento neutro claro ~#E0E0E0 (ícones/indicadores)
* **Caminhos:** acento vibrante único (ex.: **#7C5CFC**) para sensação de jornada

**3.3 Acessibilidade**

* Contraste AA, tamanhos de fonte escaláveis (Dynamic Type), labels para leitores de tela, áreas de toque 44×44

**4) Fluxos principais**

**4.1 Onboarding & Cadastro**

1. Splash → Permissões (notificação, localização) → Login (telefone / Apple / Google)
2. Criação de Perfil **obrigatória** (antes de liberar modos):
   * Nome (obrig.), Idade (>=18), **Gênero (obrig.)**
   * Preferências de exibição por gênero (multiseleção)
   * Objetivos (multi)
   * Fotos (1 principal + 4–5 adicionais)
   * Bio (opcional)
3. Tutorial rápido dos três modos

**4.2 Descoberta**

* **Deslizar:** stack de cartões (like, dislike, super like); limites no free
* **Próximos:** grade/lista por distância (tempo real)
* **Caminhos:** feed de cruzamentos (coordenada+timestamp agregado/privado)

**4.3 Match & Mensagens**

* Match quando like mútuo; cria **Thread**; push de “É um Match!”
* Mensagens: texto, imagens (com moderação); confirmação de leitura opcional

**4.4 Denúncia/Moderação**

* Denunciar usuário/conversa/mídia → fila de revisão + ação automática (severidade alta)

**4.5 Assinaturas & Compras**

* Planos: Premium mensal/trimestral/anual (ex.: ver curtidas, filtros avançados, Caminhos sem limite)
* Consumíveis: **Vibes** para Super Like e Boost

**5) Modelagem de Dados (Firestore — proposta)**

Nomes e campos podem ajustar conforme analítica; chaves em inglês para consistência.

/Users/{userId}

displayName: string

birthDate: timestamp

gender: enum("man","woman","nonbinary","trans","other")

showGenders: string[] // gêneros a exibir

bio: string

objectives: string[] // namoro, casual, amizade, etc.

heightCm: number

bodyType: enum("athletic","curvy","slim","large","fit","average")

tribe: string[] // bear, twink, daddy, discreet, etc.

hairColor: string | null

eyeColor: string | null

hasTattoos: boolean | null

hasPiercings: boolean | null

interests: string[]

photos: {

mainUrl: string

gallery: string[] // urls

}

locVisibility: enum("off","approx","precise")

lastActiveAt: timestamp

createdAt: timestamp

updatedAt: timestamp

isPremium: boolean

flags: { isVerified: boolean, banned: boolean }

/Swipes/{userId}/items/{targetUserId}

action: enum("like","dislike","superLike")

createdAt: timestamp

/Matches/{pairId}

users: [userIdA, userIdB]

createdAt: timestamp

lastMessageAt: timestamp

active: boolean

/Conversations/{threadId}

matchId: pairId

users: [userIdA, userIdB]

createdAt: timestamp

/Conversations/{threadId}/Messages/{messageId}

from: userId

type: enum("text","image")

text: string | null

imageUrl: string | null

createdAt: timestamp

readBy: string[]

/Geo/{userId}/pings/{pingId}

lat: number

lng: number

accuracy: number

createdAt: timestamp

/Crossings/{userId}/events/{eventId}

otherUserId: string

at: timestamp

approxAreaHash: string // geohash arredondado

/Reports/{reportId}

from: userId

targetType: enum("user","message","photo")

targetId: string

reason: string

createdAt: timestamp

status: enum("open","review","closed")

/Premium/Products (Remote Config)

plans: ...

**5.1 Índices & Regras (resumo)**

* Índices por **lastActiveAt**, localização aproximada (geohash), **isPremium**
* Regras: cada usuário só lê/edita seus dados; coleções de descoberta retornam campos limitados; mensagens restritas a participantes

**6) APIs (Cloud Functions / REST opcional)**

**6.1 Endpoints sugeridos (se REST adicional)**

* POST /swipe → registra like/dislike/superLike; retorna se houve match
* GET /discover → paginação por geohash/último ativo/filtros
* GET /nearby → usuários próximos com distância aproximada
* POST /crossing → registra cruzamento (server-side match por geohash/timebox)
* POST /report → denúncia
* POST /boost → consome Vibe (valida saldo)

**6.2 Cloud Functions críticas**

* **onSwipeWrite**: verifica like reverso → cria **Match** e **Conversation**
* **onImageUpload**: rodar moderação; gerar thumbs; atualizar perfil
* **onLocationPing**: agrega geohash, gera **Crossings** sob privacidade
* **onPurchaseUpdated**: atualizar **isPremium** via RevenueCat webhook

**7) Localização & Background**

**7.1 Android (Manifest & serviços)**

* Permissões: ACCESS\_FINE\_LOCATION, ACCESS\_COARSE\_LOCATION, ACCESS\_BACKGROUND\_LOCATION (se Caminhos ativo), POST\_NOTIFICATIONS, FOREGROUND\_SERVICE, FOREGROUND\_SERVICE\_LOCATION
* Foreground Service com notificação persistente quando **Caminhos** estiver ativo
* Intervalos adaptativos (p. ex., 5–10 min parado; 1–3 min em movimento); *debounce* por distância (≥50–100 m)

**7.2 iOS (Info.plist & Background Modes)**

* Chaves: NSLocationWhenInUseUsageDescription, NSLocationAlwaysAndWhenInUseUsageDescription, NSPhotoLibraryAddUsageDescription, NSCameraUsageDescription, NSUserTrackingUsageDescription (se usar ads/atribuição)
* Background Modes: **Location Updates**, **Remote notifications**, **Background fetch** (se necessário)
* Uso criterioso para aprovação: ligar/desligar Caminhos pelo usuário; explicitar benefício

**7.3 Estratégias de bateria/privacidade**

* Amostrar localização de forma adaptativa
* Em **locVisibility = approx**, armazenar apenas geohash reduzido
* Explicar claramente os usos na tela de consentimento

**8) Notificações Push**

* **FCM** (Android/iOS) + APNs
* Tópicos/eventos: match.created, message.new, crossing.detected, promo
* Deep links → abrir Thread/Perfil/Modo
* Silenciar conversas, DND por horários

**9) Segurança, Privacidade & Moderação**

* **Auth hardening:** verificação de telefone + Sign in with Apple/Google; proteção reCAPTCHA/DeviceCheck
* **Regras Firestore** de menor privilégio; validação em Cloud Functions
* **Criptografia:** TLS em trânsito; conteúdo sensível não indexável
* **Idade mínima:** 18+ com confirmação; (opcional) verificação por selfie/documento em fases
* **Conteúdo impróprio:** moderação automática (Vision API) + revisão humana
* **Bloqueio & Denúncia** com SLA interno
* **Logs de auditoria** para ações administrativas

**10) Monetização**

* **RevenueCat** para gerenciar planos e recibos (StoreKit 2 / Google Play Billing)
* Planos: mensal, trimestral, anual; *intro offers*, *win-back* via Remote Config
* **Consumíveis:** Vibes (pacotes); **Boost** (tempo limitado de destaque)

**11) Analytics (eventos sugeridos)**

* signup\_start, signup\_complete
* profile\_completed
* permission\_location\_granted
* mode\_view (deslizar/próximos/caminhos)
* swipe\_like, swipe\_dislike, super\_like
* match\_created
* message\_sent
* crossing\_detected
* purchase\_started, purchase\_success, purchase\_refund

Funis: Cadastro → Perfil 100% → 1º Like → 1º Match → 1ª Mensagem → Retenção D1/D7/D30

**12) Testes & Qualidade**

* **Unitários:** viewmodels, usecases (ex.: matching)
* **Widget/Golden tests:** cartões, listas, telas
* **Integração/E2E:** onboarding, cadastro, discovery, chat
* **Beta:** TestFlight (iOS) e Internal Testing (Play)
* **QA checklist:** permissões, localização background, perdas de rede, consumo de bateria, moderação de mídia, restores de compra

**13) Publicação nas Lojas**

**13.1 App Store (iOS)**

* Bundle ID, Perfis, Certificados, Capabilities (Background Modes, Pus

📌 **13.1 App Store (continuação)**

* **Push Notifications:**
  + Configurar **APNs Key** no Apple Developer
  + Integração via Firebase Cloud Messaging (FCM) com APNs
* **Privacidade:**
  + Preencher **Privacy Nutrition Labels** no App Store Connect
  + Declarar permissões (Câmera, Localização em segundo plano, Microfone para futuros recursos de vídeo)
* **Revisão Apple:**
  + Garantir link para política de privacidade e termos de uso
  + Garantir **Sign in with Apple** se login com redes sociais for usado

📌 **13.2 Google Play (Android)**

* **Pacote e Assinatura:**
  + Configurar pacote no formato **com.vibea.app**
  + Upload de chave de assinatura (Play App Signing)
* **Política de dados:**
  + Preencher **Data Safety Section** no Google Play Console
* **Requisitos Android 14+:**
  + Permissões de localização em segundo plano explicitamente justificadas
  + Ajustes de background services (WorkManager)

📌 **13.3 Requisitos de Lançamento**

* Teste em **TestFlight** (iOS) e **Internal Testing** (Google Play)
* Testes Beta com **Firebase App Distribution**
* Preparar **ASO (App Store Optimization):** título, descrição, screenshots, vídeos promocionais

📌 **14) Marketing e Crescimento**

* Landing Page (Next.js)
* Estratégia de aquisição: redes sociais, Google Ads, TikTok Ads
* Implementação de **Firebase Analytics** + **A/B Testing**
* Programa de indicação com deep link

📌 **15) Segurança e Conformidade**

* **Criptografia:** Dados sensíveis criptografados em trânsito (TLS 1.3) e em repouso
* **Controle de Idade:** Verificação mínima 18 anos
* **Moderação de Conteúdo:** Cloud Vision API ou serviço de moderação
* **LGPD / GDPR Compliance**

📌 **16) Roadmap Futuro**

* **Fase 2:** Stories, Videochamadas, IA para matches
* **Fase 3:** Eventos ao vivo, integrações externas (Spotify, Instagram)
* **Fase 4:** Sugestões inteligentes baseadas em comportamento

SCRIPT DO APP:  
  
junte as especificções tecnicar com esse script:

Proposta de App: Vibea+

1. A Ideia Central do Produto

O Vibea+ é um aplicativo de relacionamento único, que se diferencia por oferecer ao usuário a flexibilidade de escolher entre três modos de descoberta de perfis. Em vez de se limitar a uma única forma de interação, o app se adapta à intenção do usuário no momento. O objetivo é dar total controle sobre a experiência, permitindo que a pessoa descubra perfis de forma rápida, por proximidade imediata ou com base em cruzamentos de caminhos da vida real.

2. Funcionalidades e Fluxo do Usuário

Fluxo de Cadastro e Perfil:

O processo de cadastro será simples e direto, usando um número de celular ou conta de rede social. Após o cadastro inicial, o usuário será obrigado a criar seu perfil antes de ter acesso aos modos de descoberta.

Detalhes do Perfil: O perfil deve ser completo para enriquecer a experiência em todos os modos. O usuário precisará preencher os seguintes dados:

Fotos: Upload de 5 a 6 fotos que representem o usuário. Uma foto de perfil principal é obrigatória.

Dados Pessoais: Nome (obrigatório), idade (obrigatório, pode ser verificado por documento) e uma bio/descrição pessoal.

Preferências de Relacionamento:

Objetivos: Uma seleção de múltiplas opções pré-definidas para o usuário escolher, como "Namoro sério", "Algo casual", "Amizade", "Apenas conversa", "O que rolar". Isso é crucial para filtrar as intenções dos usuários.

Sexualidade e Gênero: Opções de auto-identificação e a escolha de quais gêneros e orientações o usuário gostaria de ver.

Detalhes Físicos e Pessoais (Campos detalhados):

Altura: Campo numérico com opção de unidade (cm ou metros).

Tipo Físico: Uma seleção de opções como "Atlético(a)", "Curvy", "Magro(a)", "Grande", "Forte", "Médio(a)".

"Tribo" ou Estilo: Uma seleção de opções ou tags para a comunidade LGBTQIA+ como "Urso", "Twink", "Daddy", "Musculoso", "Discreto", etc.

Características: Opcionais, como cor do cabelo, cor dos olhos, tipo de corpo (por exemplo, "com tatuagens", "com piercings", etc.).

Interesses e Estilo de Vida: Tags ou palavras-chave sobre hobbies, gostos musicais, filmes, etc.

Integração: Opcionalmente, pode-se integrar o perfil com redes sociais como Instagram e Spotify para exibir fotos e gostos musicais.

Navegação entre Modos:

A navegação do app será baseada em uma barra de abas na parte inferior da tela, que permitirá ao usuário alternar facilmente entre os três modos de descoberta e a tela de mensagens.

Ícones de Navegação:

"Deslizar" (ícone de cartas ou perfis).

"Próximos" (ícone de grade ou mapa).

"Caminhos" (ícone de um mapa com traços).

"Mensagens" (ícone de balão de fala).

Notificações e Interações:

As notificações serão essenciais para a experiência.

Notificação de Match: Quando o interesse mútuo é identificado no modo "Deslizar", o app emite uma notificação animada de "É um Match!".

Notificação de Cruzamento: Quando o usuário está navegando no Modo "Caminhos", o app pode usar notificações para avisar que ele cruzou com alguém com um interesse em comum ou de um perfil que ele marcou como favorito, por exemplo.

3. Arquitetura Técnica e Modelagem de Dados

Banco de Dados Unificado:

A base de dados será única para todos os usuários, independentemente do modo que eles utilizam. As interações de cada modo serão salvas em tabelas diferentes, mas relacionadas a um único perfil de usuário.

Tabela de Usuários: Contém todos os dados do perfil (nome, idade, fotos, bio, interesses, etc.).

Tabela de Interações: Armazena os "likes" e "descartes" do modo "Deslizar".

Tabela de Geolocalização: Armazena os registros de tempo e localização para o modo "Caminhos".

Tabela de Conversas: Armazena todas as mensagens trocadas.

Tecnologias-Chave:

Geolocalização: O aplicativo precisará de duas lógicas diferentes:

Rastreamento Contínuo: Para o modo "Caminhos", que registra os cruzamentos em segundo plano.

Localização em Tempo Real: Para o modo "Próximos", que precisa de dados de distância exatos e atualizados.

4. Design e Prototipagem (UI/UX)

Design Neumórfico: A interface deve seguir o conceito de Neumorfismo. Os elementos de UI serão suaves e extrudados, parecendo atravessar a superfície. Isso será conseguido com o uso de sombras sutis e uma paleta de cores monocromática, criando uma sensação física e tátil. A profundidade será mínima, garantindo uma estética clean e moderna. O app terá versões claras e escuras.

Feedback Visual por Cores: Para diferenciar cada modo, a cor de fundo da interface mudará de acordo com a função ativa, oferecendo uma experiência visual coesa e intuitiva.

Modo Deslizar: A tela principal será em uma cor escura, como um chumbo ou grafite, para focar a atenção no perfil em exibição.

Modo Próximos: A interface terá um tom neutro ou claro.

Modo Caminhos: A tela usará um tom claro ou vibrante para dar a sensação de jornada e movimento.

5. Requisitos Adicionais

Língua: O aplicativo deve ser desenvolvido e funcionar completamente em português.

Ícone Discreto: O ícone do aplicativo na tela inicial do celular deve ser discreto, sem ser óbvio ou chamativo, para preservar a privacidade do usuário.

Segurança de Acesso: Além do login por número de celular ou redes sociais, o aplicativo oferecerá a opção de o usuário criar uma senha para acessar sua conta. Também será implementado um fluxo de "Esqueci a senha" para permitir a recuperação por e-mail ou número de celular, garantindo que o usuário possa sempre acessar seu perfil de forma segura.

6. Viabilidade de Negócio e Estratégia de Mercado

Estratégia de Monetização: A receita do Vibea+ será baseada em um modelo freemium. As funcionalidades básicas serão gratuitas, enquanto recursos adicionais serão oferecidos por meio de:

Assinatura Premium: Permite ver quem curtiu seu perfil, ter acesso a filtros de busca avançados e navegar no modo "Caminhos" sem limites.

Compras no Aplicativo (In-App Purchases): Venda de "Vibes" (uma moeda virtual) que podem ser usadas para enviar um "Super Like" ou dar um "Boost" no perfil.

Público-Alvo e Posicionamento: O Vibea+ se posiciona como a solução para o usuário que busca flexibilidade e controle. Ele atende tanto à intenção de encontros rápidos (Modo Próximos) quanto a de conexões significativas (Modo Caminhos). O público-alvo são jovens adultos (20-35 anos) que valorizam a variedade e a personalização em aplicativos de relacionamento.

Segurança e Privacidade: A segurança do usuário é a principal prioridade. O aplicativo deve ter um sistema robusto que inclui:

Moderação de Perfis: Para combater perfis falsos e de spam.

Mecanismo de Denúncia: Ferramenta fácil para denunciar comportamentos inadequados.

Controle de Localização: Oferecer opções de privacidade para que o usuário possa controlar quem vê sua localização e quando.

7. Plano de Lançamento e Testes de Viabilidade

Testes Internos: Antes de qualquer lançamento, a equipe de desenvolvimento irá realizar testes de funcionalidade e de segurança para garantir a estabilidade do app.

Programa Beta (Testes de Viabilidade): Uma versão do aplicativo será disponibilizada para um grupo seleto de usuários (testadores beta) em uma área geográfica limitada. O objetivo é validar a viabilidade e a aceitação de todos os recursos em um ambiente real. O feedback desses usuários será usado para identificar bugs, avaliar a experiência de uso e fazer ajustes finais antes do lançamento oficial.

Lançamento: Após a otimização com base no feedback do programa beta, o aplicativo será lançado para o público em geral.

* **Testes de Carga e Performance:** Testar como o app se comporta com um grande número de usuários simultâneos, especialmente nos modos "Próximos" e "Caminhos", que dependem de dados de geolocalização em tempo real. Isso é crucial para evitar lentidão e falhas.
* **Testes de Usabilidade (UX/UI):** Realizar sessões com usuários reais, observando como eles navegam entre os modos e interagem com a interface. Isso pode revelar problemas de design ou fluxos de usuário confusos que os testes técnicos não capturam.

**2. Estratégias de Crescimento e Retenção**

A estratégia de monetização é clara, mas o plano de crescimento pode ser mais aprofundado:

* **Viralidade:** Como o aplicativo pode se tornar viral? O programa de indicação com deep link é um bom começo. Que tal gamificação? Recompensar os usuários por convidar amigos pode impulsionar o crescimento.
* **Retenção de Usuários:** Além das notificações de match e mensagens, como o app vai manter os usuários engajados a longo prazo? Campanhas de marketing por e-mail, notificações push personalizadas com base no comportamento do usuário (por exemplo, "Vimos que você cruzou com alguém com interesses em comum! Dê uma olhada no perfil.") ou a introdução de novos recursos de forma regular seriam estratégias valiosas.

**3. Gerenciamento de Conteúdo e Comunidade**

O documento aborda a moderação de conteúdo, mas para o crescimento da comunidade, a presença de uma equipe dedicada à moderação e suporte pode ser um diferencial.

* **Guia de Conduta:** Criar um guia de conduta claro para a comunidade, detalhando o que é e o que não é permitido. Isso ajuda a educar os usuários e facilita o trabalho de moderação.
* **Canais de Suporte:** Definir canais de suporte ao usuário (e-mail, chat no app) para que eles possam relatar problemas, tirar dúvidas e fazer denúncias de forma mais eficiente.

 **Especificações de Performance**

* Definir meta de **tempo de carregamento** (ex.: menos de 2s para telas principais).
* Tamanho máximo de imagens enviadas e compressão.
* Cache local para reduzir chamadas ao servidor.

 **Acessibilidade Detalhada**

* Adicionar suporte a **VoiceOver/TalkBack** completo.
* Incluir testes de acessibilidade na checklist de QA.

 **Fluxos de Segurança Avançada**

* Falta detalhar **rate limiting** (para evitar abusos/spam).
* Plano para **armazenamento seguro de tokens** e **refresh automático**.

 **Analytics e Métricas**

* Já tem eventos, mas faltam **KPIs claros** (retenção, taxa de match, churn).

 **CI/CD** (Integração e Deploy Contínuos)

* Não há menção a pipeline de build (Fastlane, Codemagic ou GitHub Actions).

 **Suporte Offline**

* Não há definição sobre funcionamento básico sem internet (ex.: ver conversas salvas).

 **LGPD/GDPR**

* Menciona conformidade, mas poderia incluir **fluxo de exclusão de conta e portabilidade de dados**.

**🔹 1. Planejamento e Definição**

1. Definir **escopo do MVP** (funcionalidades essenciais).
2. Criar **personas e fluxos de usuário** com base no público-alvo.
3. Elaborar **wireframes e protótipo navegável** (Figma).
4. Validar conceito com grupo pequeno de usuários.

**🔹 2. Arquitetura e Stack**

1. Escolher **stack final**: Flutter + Firebase + RevenueCat.
2. Estruturar **arquitetura limpa (MVVM)**.
3. Configurar **repositórios, Git e CI/CD** (GitHub Actions/Fastlane).
4. Criar **pipeline de build e deploy** (TestFlight, Play Console).

**🔹 3. Modelagem de Dados e Backend**

1. Implementar **estrutura de coleções Firestore**:
   * Users, Swipes, Matches, Conversations, Geo, Crossings, Reports, Premium.
2. Configurar **Cloud Functions**:
   * onSwipeWrite, onImageUpload, onLocationPing, onPurchaseUpdated.
3. Definir **regras de segurança** e testes automatizados.
4. Implementar **cache local** e suporte offline básico.

**🔹 4. Cadastro, Login e Perfil**

1. Criar fluxo de **onboarding**: permissões, login (telefone, Google, Apple).
2. Construir **tela de criação de perfil**:
   * Nome, idade, gênero, objetivos, fotos, bio, preferências.
3. Implementar **verificação de idade** (18+) e segurança de login.
4. Adicionar **opções de integração com Instagram/Spotify**.

**🔹 5. Modos de Descoberta**

1. **Modo Deslizar**: stack de cartões com like/dislike/superlike.
2. **Modo Próximos**: lista/grade em tempo real com distância.
3. **Modo Caminhos**: cruzamentos via geohash e tracking adaptativo.
4. Otimizar para **bateria e privacidade** (visibilidade aproximada).

**🔹 6. Match e Mensagens**

1. Criar fluxo de **notificação de match**.
2. Desenvolver **chat com imagens moderadas** e status de leitura.
3. Adicionar **bloqueio e denúncia** com fluxo de revisão.

**🔹 7. Monetização e Compras**

1. Integrar **RevenueCat** para planos Premium e consumíveis.
2. Implementar **Super Like, Boost** e filtros avançados pagos.
3. Configurar **experimentos de preço e ofertas** via Remote Config.

**🔹 8. Moderação e Segurança Avançada**

1. Adicionar **moderação automática** com Vision API.
2. Configurar **rate limiting** e proteção contra abuso.
3. Criar **logs de auditoria** e ferramentas internas.
4. Garantir **LGPD/GDPR** com exclusão de conta e portabilidade de dados.

**🔹 9. Analytics e Observabilidade**

1. Integrar **Crashlytics e Firebase Analytics**.
2. Definir **KPIs**: retenção, taxa de match, engajamento.
3. Montar dashboards de métricas e alertas.

**🔹 10. UI/UX e Acessibilidade**

1. Implementar **tema preto e neumorfismo sutil**.
2. Diferenciar modos com **cores únicas**.
3. Testar **acessibilidade**: TalkBack, VoiceOver, contraste AA.

**🔹 11. Testes e QA**

1. Testes **unitários, widget, integração**.
2. **Testes de carga** para escalabilidade.
3. **Beta fechado** (TestFlight e Google Play Internal Testing).
4. Ajustes com base no feedback.

**🔹 12. Publicação nas Lojas**

1. Preparar **App Store Connect** (Bundle ID, Privacy Labels).
2. Configurar **Google Play Console** (Data Safety, assinatura).
3. Criar **assets promocionais** (screenshots, vídeo).

**🔹 13. Lançamento e Crescimento**

1. Lançar versão **MVP** em regiões limitadas.
2. Ativar **programa de indicação com deep link**.
3. Estratégia de **ads e social media**.
4. Planejar roadmap: Stories, IA, integrações externas.

**4. Geolocalização e Privacidade do Usuário**

**Objetivo:** Permitir que o Vibea+ sugira perfis próximos ao usuário, oferecendo experiências de conexão mais rápidas e relevantes, sem comprometer a segurança e a privacidade.

**Funcionalidades principais:**

1. **Localização em tempo real (opcional):**
   * O usuário pode ativar a geolocalização para ver perfis próximos.
   * A localização exata nunca é compartilhada com outros usuários; apenas a distância aproximada é exibida.
2. **Configurações de privacidade:**
   * O usuário pode escolher ocultar a localização, usar localização aproximada ou desativar completamente a função.
   * Opção de “modo invisível” que permite navegar no app sem mostrar presença.
3. **Segurança dos dados:**
   * A posição geográfica é criptografada e armazenada de forma segura.
   * Nenhuma informação de localização será vendida ou compartilhada com terceiros.
4. **Sugestões inteligentes:**
   * O app pode usar a geolocalização para sugerir eventos próximos, encontros ou conexões em tempo real.
   * Algoritmos de proximidade podem levar em conta a distância, preferências e histórico de interações.
5. **Pipeline CI/CD:**  
   Você menciona GitHub Actions/Fastlane, mas não detalha builds, testes automatizados e deploy contínuo. Seria útil ter fluxos claros de staging, QA e produção.
6. **Suporte offline parcial:**  
   Não há definição de como o app deve funcionar sem internet — pelo menos para ver conversas salvas, perfis em cache ou notificações pendentes.
7. **KPIs e métricas de engajamento:**  
   Eventos de analytics estão listados, mas faltam indicadores claros de sucesso, como: taxa de match, churn, tempo médio de sessão, retenção D1/D7/D30.
8. **Rate limiting e prevenção de abuso:**  
   Foi mencionado apenas de forma superficial. Seria bom detalhar limites por usuário, por IP, para ações como swipes, mensagens ou envio de imagens.
9. **Fluxo de exclusão e portabilidade de dados:**  
   Para LGPD/GDPR, seria ideal explicitar como o usuário pode excluir conta, exportar dados e revogar consentimentos.
10. **Detalhamento de notificações inteligentes:**  
    As sugestões baseadas em cruzamentos e comportamento poderiam ser explicadas com mais clareza: frequência, filtros, como evitar spam de push.
11. **Gamificação e engajamento a longo prazo:**  
    Você mencionou viralidade e retenção, mas ainda poderia incluir badges, conquistas ou streaks para aumentar engajamento contínuo.
12. **Testes de performance para geolocalização:**  
    Testes de bateria, consumo de dados e performance em background são críticos, especialmente para o modo Caminhos.

**Vibea+ — Documento Técnico Consolidado (v1.1)**

**0. Sumário Executivo**

* **Stack sugerida:** Flutter (Dart) + Firebase (Auth, Firestore, Storage, Cloud Functions, FCM) + RevenueCat (assinaturas)
* **Arquitetura App:** Clean Architecture (Domain / Data / Presentation), padrão MVVM com Riverpod/Bloc
* **Modos de descoberta:** Deslizar · Próximos · Caminhos · Mensagens
* **Tema:** Preto como base (neumorfismo sutil) com cores de modo
* **Perfil:** Gênero obrigatório + preferências de exibição
* **Privacidade e segurança:** Regras Firestore, Cloud Functions, moderação de mídia, controle de localização
* **Publicação:** App Store + Google Play (diretrizes, privacidade, billing)
* **Monetização:** Freemium (assinatura + consumíveis)
* **Objetivo:** Conectar usuários com flexibilidade, privacidade e experiência personalizada.

**1. Objetivo & Escopo**

**Objetivo:** Criar um app de relacionamento com três modos de descoberta, priorizando personalização, privacidade e performance.

**Escopo MVP:**

* Cadastro por telefone/redes sociais
* Perfil completo: nome, idade, gênero, preferências, fotos, bio, objetivos
* Modos de descoberta: Deslizar, Próximos, Caminhos
* Notificações: match, cruzamentos, mensagens
* Filtros básicos (idade, distância, objetivos)
* Denúncia e bloqueio de perfis
* Monetização freemium (assinatura + consumíveis)

**Fora do MVP (planejado):**

* Verificação por documento/selfie
* Videochamada, Stories/Clipes
* IA para sugestões de abertura de conversa

**2. Arquitetura e Stack**

**Mobile:** Flutter 3.x, MVVM + Riverpod/Bloc, Material 3 custom, go\_router para navegação, Google Maps SDK, geolocator.  
**Backend & Serviços:** Firebase (Auth, Firestore, Storage, Cloud Functions, FCM, Analytics, Remote Config), RevenueCat para billing, Cloud Vision API para moderação de mídia, CDN de imagens via Firebase Storage.  
**Observabilidade:** Crashlytics, Firebase Analytics, dashboards de performance.

**3. UX/UI (Tema Preto + Neumorfismo Sutil)**

* Fundo preto (#000000), cartões #0B0B0B, sombras suaves para efeito “extrudado”
* Cores por modo:
  + **Deslizar:** cinza-grafite (#1F1F1F)
  + **Próximos:** neutro claro (#E0E0E0)
  + **Caminhos:** vibrante (#7C5CFC)
* Acessibilidade: contraste AA, tamanhos escaláveis, labels para leitores de tela, áreas de toque ≥44×44

**4. Fluxos Principais**

**4.1 Onboarding & Cadastro**

1. Splash → Permissões (notificação, localização) → Login (telefone/Apple/Google)
2. Criação de Perfil obrigatória: nome, idade (>=18), gênero, preferências, objetivos, fotos, bio
3. Tutorial rápido dos três modos

**4.2 Descoberta**

* **Deslizar:** stack de cartões (like, dislike, superlike), limites no free
* **Próximos:** grade/lista por distância (tempo real)
* **Caminhos:** feed de cruzamentos (geohash + timestamp agregado)

**4.3 Match & Mensagens**

* Match cria Thread e notificação “É um Match!”
* Mensagens: texto, imagens (moderação), confirmação de leitura opcional

**4.4 Denúncia / Moderação**

* Denunciar usuário/conversa/mídia → fila de revisão + ação automática (severidade alta)

**4.5 Monetização**

* Planos Premium (mensal, trimestral, anual)
* Consumíveis: Vibes para Super Like, Boost
* Experimentos e ofertas via Remote Config

**5. Modelagem de Dados (Firestore)**

* **/Users/{userId}** → perfil completo, fotos, preferências, localização, flags
* **/Swipes/{userId}/items/{targetUserId}** → ação (like/dislike/superlike)
* **/Matches/{pairId}** → usuários, timestamps, active
* **/Conversations/{threadId}/Messages/{messageId}** → mensagens com moderação
* **/Geo/{userId}/pings/{pingId}** → localização em tempo real
* **/Crossings/{userId}/events/{eventId}** → cruzamentos
* **/Reports/{reportId}** → denúncias
* **/Premium/Products** → planos via Remote Config
* Índices: lastActiveAt, geohash, isPremium
* Regras: menor privilégio, mensagens restritas a participantes

**6. APIs & Cloud Functions**

* REST opcional: /swipe, /discover, /nearby, /crossing, /report, /boost
* Cloud Functions críticas: onSwipeWrite, onImageUpload, onLocationPing, onPurchaseUpdated

**7. Geolocalização & Background**

* Android: Foreground Service, permissões de localização e notificações
* iOS: Info.plist (NSLocation\*, NSCamera\*), Background Modes (Location Updates, Remote notifications)
* Estratégias: localização adaptativa, visibilidade aproximada, explicação clara ao usuário

**8. Notificações Push**

* FCM + APNs
* Tópicos/eventos: match.created, message.new, crossing.detected, promo
* Deep links para Thread/Perfil/Modo
* Silenciamento de conversas e DND por horários

**9. Segurança, Privacidade & Moderação**

* Auth robusto: telefone, Apple/Google, reCAPTCHA/DeviceCheck
* Criptografia TLS, regras Firestore, validação Cloud Functions
* Moderação: Vision API + revisão humana
* Logs de auditoria
* Controle de idade 18+, fluxos de verificação (selfie/documento)

**10. Monetização**

* RevenueCat: planos Premium, consumíveis (Vibes, Boost)
* Experimentos de preço e ofertas via Remote Config

**11. Analytics & KPIs**

* Eventos: signup\_start, profile\_completed, swipe\_like, match\_created, message\_sent, crossing\_detected, purchase\_success
* KPIs: taxa de match, retenção D1/D7/D30, churn, tempo médio de sessão
* Dashboards de métricas e alertas

**12. Testes & QA**

* Unitários: viewmodels, usecases
* Widget/Golden tests: cartões, listas
* Integração/E2E: onboarding, cadastro, discovery, chat
* Testes de carga/performance, geolocalização, consumo de bateria
* Beta: TestFlight (iOS) e Internal Testing (Play)
* Checklist QA: permissões, offline, restores, moderação

**13. Publicação nas Lojas**

* App Store: Bundle ID, Privacy Labels, Push Notifications (APNs + FCM), Sign in with Apple
* Google Play: pacote, Data Safety, assinaturas, permissões background explicitadas
* Assets promocionais: screenshots, vídeo, ASO

**14. Marketing & Crescimento**

* Landing Page (Next.js)
* Estratégia Ads: redes sociais, Google Ads, TikTok Ads
* Programa de indicação com deep link, gamificação (badges, streaks)
* Campanhas push personalizadas

**15. Roadmap Futuro**

* Fase 2: Stories, Videochamadas, IA para matches
* Fase 3: Eventos ao vivo, integrações externas (Spotify, Instagram)
* Fase 4: Sugestões inteligentes baseadas em comportamento

**16. Detalhes Adicionais**

* CI/CD: GitHub Actions/Fastlane
* Suporte offline parcial: conversas salvas, perfis em cache
* Fluxo de exclusão e portabilidade de dados (LGPD/GDPR)
* Rate limiting e prevenção de abuso
* Notificações inteligentes detalhadas
* Testes de UX com usuários reais

**Fluxograma de Desenvolvimento & Fluxo do Usuário – Vibea+**

**1️⃣ Onboarding e Cadastro**

Start → Splash Screen → Permissões (Notificação, Localização) → Login (Tel / Google / Apple) → Criação de Perfil obrigatória

* Criação de Perfil: Nome, Idade, Gênero → Preferências (Gênero/Objetivos) → Fotos → Bio → Integração opcional (Instagram/Spotify)
* Status: MVP

**2️⃣ Modos de Descoberta**

Perfil Completo → Tela Principal → [Barra de Navegação Inferior]

├─ Deslizar → Stack de cartões → Like / Dislike / Super Like → Atualiza Swipes / Possível Match

├─ Próximos → Lista/Grade usuários próximos (tempo real) → Atualiza Swipes / Crossings

└─ Caminhos → Rastreamento contínuo → Geohash → Cruzamentos privados → Sugestões inteligentes

* Notificações: Match, Cruzamento, Mensagem
* Privacidade: locVisibility (off / approx / precise)
* Status: MVP

**3️⃣ Match & Mensagens**

Mutual Like → Cria Thread → Notificação “É um Match!” → Chat (Texto / Imagem moderada) → Status de leitura opcional

* Bloqueio e Denúncia → Moderação automática + revisão humana
* Status: MVP

**4️⃣ Monetização**

Plano Premium → Filtros avançados, Curtidas, Caminhos sem limite

Consumíveis → Vibes (Super Like / Boost)

* RevenueCat → Integração StoreKit / Play Billing
* Status: MVP

**5️⃣ Segurança & Privacidade**

Login seguro (telefone/Apple/Google + senha opcional) → Autenticação reforçada → TLS → Firestore regras restritivas → Moderação de conteúdo → LGPD/GDPR: exclusão/portabilidade

* Fluxo de rate limiting para evitar abuso
* Status: MVP

**6️⃣ UI/UX & Acessibilidade**

Tema escuro + Neumorfismo → Cores diferenciadas por modo → Contraste AA → VoiceOver/TalkBack → Áreas de toque ≥ 44px

* Status: MVP

**7️⃣ Testes & QA**

Unitários / Widget / Integração → Testes de performance (modos Próximos & Caminhos) → Beta fechado → Ajustes → Publicação nas lojas

* Status: MVP

**8️⃣ Pipeline CI/CD**

GitHub Actions / Fastlane → Builds Staging → QA → Deploy TestFlight / Play Internal Testing → Produção

* Status: MVP

**9️⃣ Marketing & Crescimento**

Landing Page → Programa de Indicação → Ads (Google / TikTok / Social Media) → Gamificação (Badges / Streaks)

* Status: Futuro

**🔟 Roadmap Futuro**

Fase 2: Stories / Videochamadas / IA para sugestões

Fase 3: Eventos ao vivo / Integrações externas (Spotify, Instagram)

Fase 4: Sugestões inteligentes baseadas em comportamento

**Plano de Testes Detalhado para o Vibea+**

**1. Testes Funcionais (O que o aplicativo faz)**

* **Fluxo de Cadastro e Login**
  + Testar login com número de telefone e código SMS.
  + Testar login com redes sociais (Apple/Google).
  + Testar o fluxo de "Esqueci a Senha".
  + Verificar o preenchimento obrigatório e opcional do perfil (nome, idade, gênero, fotos, etc.).
* **Modos de Descoberta**
  + **Modo Deslizar:**
    - Verificar o carregamento de perfis e a função de like/dislike/super like.
    - Testar a aplicação de filtros (idade, distância, objetivos).
  + **Modo Próximos:**
    - Verificar se a lista de usuários é atualizada em tempo real com base na localização.
    - Testar a precisão do cálculo de distância.
  + **Modo Caminhos:**
    - Testar se o rastreamento em segundo plano funciona corretamente.
    - Verificar se o aplicativo notifica os cruzamentos de caminho conforme esperado.
* **Interações e Mensagens**
  + Testar o fluxo de match (quando dois usuários dão like um no outro).
  + Verificar se as notificações de match são exibidas corretamente.
  + Testar o envio e recebimento de mensagens de texto e imagens.
  + Testar as funções de bloquear e denunciar um perfil.
* **Monetização**
  + Testar a compra de consumíveis ("Vibes").
  + Verificar a compra de planos de assinatura.
  + Garantir que os recursos premium (filtros avançados, acesso ilimitado ao modo Caminhos, etc.) sejam liberados após a compra.
* **Acessibilidade**
  + Testar o suporte a leitores de tela como VoiceOver (iOS) e TalkBack (Android).
  + Verificar se os elementos de interface têm tamanho de toque adequado (maior que 44px).

**2. Testes de Não Funcionais (Como o aplicativo funciona)**

* **Performance e Estabilidade**
  + **Teste de Carga:** Simular múltiplos usuários nos modos "Próximos" e "Caminhos" para garantir que o banco de dados e o backend não sobrecarreguem.
  + **Velocidade de Carregamento:** Medir o tempo de carregamento das telas principais para garantir que estejam abaixo de 2 segundos.
  + **Otimização de Imagens:** Verificar a compressão de imagens enviadas e o impacto no tempo de carregamento.
* **Segurança**
  + **Autenticação:** Testar vulnerabilidades no fluxo de login para garantir que não haja acesso não autorizado.
  + **Regras de Segurança do Firestore:** Tentar ler, escrever e apagar dados sem a permissão adequada para garantir que as regras estejam funcionando.
  + **Moderação:** Tentar enviar imagens inapropriadas para verificar se o sistema de moderação automática (Cloud Vision API) funciona.
  + **Conformidade:** Garantir que o aplicativo está em conformidade com as leis de proteção de dados (LGPD/GDPR).
* **Usabilidade (UX/UI)**
  + Testar a navegação entre os modos e a transição das cores de tema.
  + Garantir que o design Neumórfico se mantenha consistente em todas as telas.
* **Ambientes**
  + Testar a instalação e o desempenho do aplicativo em diferentes dispositivos iOS e Android, e em versões de sistema operacional mais antigas e mais recentes.

**Testes de Fluxo (End-to-End)**

Estes testes simulam o comportamento real do usuário, do início ao fim:

1. Um novo usuário se cadastra, preenche o perfil completo e começa a deslizar os perfis.
2. Um usuário com perfil completo entra no modo "Próximos" e dá like em alguém que está perto.
3. Dois usuários se dão like mutuamente, recebem a notificação de match e iniciam uma conversa.
4. Um usuário compra um plano premium e tem acesso a filtros e recursos extras.
5. Um usuário usa o modo "Caminhos", cruza com outro usuário e é notificado.

Vibea+ — Documento Técnico Consolidado e Proposta Integrada

0. Sumário Executivo

Objetivo: Aplicativo de relacionamento flexível, permitindo três modos de descoberta, priorizando privacidade, personalização e experiência do usuário.

Stack sugerida:

* Flutter (Dart)
* Firebase (Auth, Firestore, Storage, Cloud Functions, FCM)
* RevenueCat (assinaturas)

Arquitetura: Clean Architecture (Domain/Data/Presentation) + MVVM com Riverpod/Bloc

Modos de descoberta: Deslizar · Próximos · Caminhos · Mensagens

Tema: Preto base (neumorfismo sutil) + cores por modo

Perfil: Gênero obrigatório + preferências de exibição

Privacidade & Segurança: Regras Firestore, validação em Cloud Functions, moderação de mídia, controle de localização, criptografia TLS, LGPD/GDPR

Publicação: App Store + Google Play (diretrizes, privacidade, billing)

Monetização: Freemium (assinatura + consumíveis)

1. Objetivo & Escopo

Objetivo: Criar app de relacionamento com três modos de descoberta, priorizando personalização, privacidade e performance.

1.1 MVP

* Cadastro por telefone/redes sociais
* Perfil completo: nome, idade, gênero, fotos (5–6), bio, preferências, objetivos
* Modos: Deslizar, Próximos (tempo real), Caminhos (geohash + timestamp)
* Notificações: match, cruzamentos, mensagens
* Filtros básicos: idade, distância, objetivos
* Denúncia e bloqueio de perfis
* Monetização: assinatura + consumíveis

1.2 Fora do MVP

* Verificação por documento/selfie
* Videochamada, Stories/Clipes
* IA para sugestões de abertura de conversa

2. Planejamento e Arquitetura

* Definir MVP, personas, fluxos e wireframes (Figma)
* Validar conceito com grupo pequeno de usuários
* Stack final: Flutter + Firebase + RevenueCat
* Arquitetura limpa: MVVM + Riverpod/Bloc
* CI/CD: GitHub Actions / Fastlane (builds, testes automatizados, deploy)
* Observabilidade: Crashlytics, Firebase Analytics, dashboards de performance

3. UX/UI

* Tema preto + Neumorfismo sutil
* Fundos & Elevação: fundo #000000, cartões #0B0B0B, sombras suaves, bordas 1–2px
* Cores por modo:
  + Deslizar: cinza-grafite #1F1F1F
  + Próximos: neutro claro #E0E0E0
  + Caminhos: vibrante #7C5CFC
* Acessibilidade: contraste AA, fontes escaláveis, labels para leitores de tela, áreas de toque ≥44×44
* Navegação: barra inferior → Deslizar, Próximos, Caminhos, Mensagens

4. Fluxos Principais

4.1 Onboarding & Cadastro

* Splash → permissões (notificação, localização) → login (telefone / Apple / Google)
* Criação de perfil obrigatória: nome, idade (>=18), gênero, objetivos, fotos (5–6), bio, preferências
* Tutorial rápido dos três modos

4.2 Descoberta

* Deslizar: stack de cartões, like/dislike/superlike, limites free
* Próximos: lista/grade por distância em tempo real
* Caminhos: feed de cruzamentos (geohash + timestamp), tracking adaptativo e privacidade

4.3 Match & Mensagens

* Match cria Thread e notificação “É um Match!”
* Chat: texto, imagens (moderação), confirmação de leitura opcional

4.4 Denúncia / Moderação

* Denunciar usuário/conversa/mídia → fila de revisão + ação automática (severidade alta)

4.5 Monetização

* Planos Premium: mensal, trimestral, anual
* Consumíveis: Vibes (Super Like), Boost
* Experimentos de preço via Remote Config

5. Modelagem de Dados (Firestore)

Coleções principais:

* /Users/{userId}: perfil completo, fotos, preferências, localização, flags, isPremium
* /Swipes/{userId}/items/{targetUserId}: ação (like/dislike/superlike)
* /Matches/{pairId}: usuários, timestamps, active
* /Conversations/{threadId}/Messages/{messageId}: mensagens
* /Geo/{userId}/pings/{pingId}: localização em tempo real
* /Crossings/{userId}/events/{eventId}: cruzamentos
* /Reports/{reportId}: denúncias
* /Premium/Products: planos via Remote Config

Índices e regras:

* lastActiveAt, geohash, isPremium
* Privilégio mínimo, mensagens restritas a participantes

6. APIs & Cloud Functions

* REST opcional: /swipe, /discover, /nearby, /crossing, /report, /boost
* Cloud Functions críticas: onSwipeWrite, onImageUpload, onLocationPing, onPurchaseUpdated

7. Geolocalização & Privacidade

* Android: Foreground Service, permissões localização
* iOS: Info.plist + Background Modes
* Estratégias: localização adaptativa, visibilidade aproximada, explicação clara ao usuário
* Sugestões inteligentes: eventos, cruzamentos, proximidade com interesses em comum

8. Notificações Push

* FCM + APNs
* Eventos: match.created, message.new, crossing.detected, promo
* Deep links para Thread/Perfil/Modo
* Silenciamento e DND por horários

9. Segurança & Moderação

* Auth robusto: telefone, Apple/Google, reCAPTCHA/DeviceCheck
* Criptografia TLS, regras Firestore, validação Cloud Functions
* Moderação: Vision API + revisão humana
* Idade mínima 18+
* Bloqueio/Denúncia com SLA, logs de auditoria

10. Monetização

* RevenueCat: planos Premium e consumíveis
* Super Like, Boost, filtros avançados pagos
* Experimentos de preço e ofertas via Remote Config

11. Analytics & KPIs

* Eventos: signup\_start, profile\_completed, swipe\_like, match\_created, message\_sent, crossing\_detected, purchase\_success
* KPIs: taxa de match, churn, tempo médio de sessão, retenção D1/D7/D30
* Dashboards de métricas e alertas

12. Testes & QA

* Unitários: viewmodels, usecases
* Widget/Golden tests: cartões, listas
* Integração/E2E: onboarding, cadastro, discovery, chat
* Testes de carga/performance
* Beta: TestFlight e Internal Testing
* Checklist QA: permissões, offline, restores, moderação

13. Publicação nas Lojas

* App Store: Bundle ID, certificados, Privacy Labels, Sign in with Apple
* Google Play: Pacote com.vibea.app, Data Safety, background services
* Assets promocionais: screenshots, vídeo

14. Marketing e Crescimento

* Landing Page (Next.js)
* Ads: redes sociais, Google, TikTok
* Programa de indicação com deep link
* A/B Testing, campanhas de retenção

15. Roadmap Futuro

* Fase 2: Stories, Videochamadas, IA para matches
* Fase 3: Eventos ao vivo, integrações externas (Spotify, Instagram)
* Fase 4: Sugestões inteligentes baseadas em comportamento

16. Requisitos Adicionais & Melhorias Integradas

* Rate limiting e prevenção de abuso (swipes, mensagens, upload)
* Suporte offline parcial (ver conversas salvas, perfis em cache)
* Fluxo de exclusão e portabilidade de dados (LGPD/GDPR)
* Suporte completo a acessibilidade: VoiceOver/TalkBack, contraste AA
* Meta de performance: telas carregam <2s, compressão de imagens e cache local
* Integração opcional com Instagram/Spotify

 **Interação:** A coleção Swipes ou Interação deve ser restrita para que um usuário só possa criar ou ler seus próprios swipes, e não os de outros. A regra read: false e write: false pode ser o ponto de partida, liberando apenas para o usuário autenticado em sua própria sub-coleção.

 **GeoPing e Cruzamento:** A documentação do Vibea+ já afirma que o app utiliza uma estratégia de privacidade, armazenando "apenas geohash reduzido" e que as informações de cruzamento são "privadas". Você deve garantir que as regras do Firestore reflitam isso, permitindo apenas que o usuário autenticado leia os seus próprios pings e cruzamentos.