

PRD - Product Requirements Document

Check Point - Plataforma de Presença Social

Versão: 1.0

Data: Janeiro 2026

Status: Em Desenvolvimento

Autor: Product Team

Índice

1. [Visão Geral](#)
 2. [Objetivos e Métricas de Sucesso](#)
 3. [Personas e Casos de Uso](#)
 4. [Requisitos Funcionais](#)
 5. [Requisitos Não-Funcionais](#)
 6. [Arquitetura de Informação](#)
 7. [Especificação de Telas](#)
 8. [Especificação de APIs](#)
 9. [Regras de Negócio](#)
 10. [Estados e Erros](#)
 11. [Notificações](#)
 12. [Integrações](#)
 13. [Segurança e Privacidade](#)
 14. [Analytics e Tracking](#)
 15. [Cronograma e Fases](#)
-

1. Visão Geral

1.1 Problema

70% das conexões feitas em eventos se perdem porque não há forma fácil de reconectar com pessoas que você conheceu. Além disso, pessoas deixam de ir a eventos por não saber quem estará lá.

1.2 Solução

Check Point é uma camada de presença social que permite:

- **Check-in Social:** Marcar presença em eventos/locais
- **Discovery:** Ver quem está/estaré no mesmo lugar
- **ReMatch:** Reconectar com pessoas que encontrou
- **Dashboard B2B:** Métricas para organizadores

1.3 Proposta de Valor

Para	Proposta
Usuários	"Nunca mais perca o contato de quem você conheceu"
Organizadores	"Aumente engajamento e transforme eventos em comunidade"

1.4 Escopo do Documento

Este PRD cobre o **MVP (Minimum Viable Product)** com duração de 4 meses de desenvolvimento.

2. Objetivos e Métricas de Sucesso

2.1 Objetivos do Produto

ID	Objetivo	Prazo
OBJ-001	Validar que usuários querem saber quem está em eventos	MVP
OBJ-002	Validar que ReMatch tem valor (pessoas reconectam)	MVP
OBJ-003	Validar que organizadores pagam pela ferramenta	MVP+3 meses
OBJ-004	Criar efeito de rede em eventos específicos	6 meses

2.2 Métricas de Sucesso (MVP - 3 meses)

Métrica	Meta	Como Medir
Usuários cadastrados	500	COUNT(users)

Métrica	Meta	Como Medir
MAU (Monthly Active Users)	200	Users com ≥ 1 ação/mês
Check-ins totais	1.000	COUNT(checkins)
Check-ins/usuário ativo	5	AVG(checkins per MAU)
ReMatch requests enviados	150	COUNT(rematch_requests)
Taxa de aceite ReMatch	25%	Aceitos/Total requests
Eventos cadastrados	20	COUNT(events)
NPS	>40	Survey in-app

2.3 North Star Metric

⭐ "Conexões com Contexto por Semana"
 = Número de ReMatches aceitos entre pessoas
 que estiveram no mesmo evento/local

Meta MVP: 50/semana

Meta Ano 1: 500/semana

3. Personas e Casos de Uso

3.1 Personas

Persona 1: Marina - A Networker Social

Atributo	Detalhe
Idade	28 anos
Profissão	UX Designer em startup
Localização	São Paulo - Vila Madalena
Comportamento	Frequenta 3-4 eventos/mês
Redes Sociais	Instagram (2h/dia), LinkedIn
Dispositivo	iPhone 14, sempre atualizado
Dor Principal	"Conheci tanta gente legal mas perdi o contato"
Motivação	Ampliar rede profissional e social
Barreira	Não quer parecer "stalker"

Jobs-to-be-Done:

1. Quando vou a um evento, quero saber quem mais está lá para encontrar pessoas interessantes
2. Depois do evento, quero reconectar com quem conversei para não perder o contato
3. Antes de decidir ir, quero ver se pessoas do meu círculo vão

Persona 2: Lucas - O Hesitante Social

Atributo	Detalhe
Idade	24 anos
Profissão	Dev Backend, trabalha remoto
Localização	São Paulo - Pinheiros
Comportamento	Quer ir em eventos mas tem insegurança
Redes Sociais	Instagram (passivo), Twitter/X
Dispositivo	Android (Pixel)
Dor Principal	"Não vou se for sozinho"
Motivação	Sair do isolamento do home office
Barreira	Ansiedade social, medo de não conhecer ninguém

Jobs-to-be-Done:

1. Antes de comprar ingresso, quero saber se amigos ou conhecidos vão
2. Quando chego no evento, quero uma "muleta social" para começar conversas
3. Quero ir em eventos onde sei que terei conexões em comum com pessoas

Persona 3: Carla - A Organizadora

Atributo	Detalhe
Idade	35 anos
Profissão	Produtora de eventos de tecnologia
Localização	São Paulo - Itaim
Comportamento	Organiza 2-3 eventos/mês
Ferramentas	Sympla, Notion, Mailchimp
Dispositivo	MacBook + iPhone
Dor Principal	"Muita gente compra e não vai, não consigo criar comunidade"
Motivação	Aumentar engajamento e criar eventos memoráveis
Barreira	Pouco tempo, precisa ser fácil de usar

Jobs-to-be-Done:

1. Quero aumentar a taxa de comparecimento dos meus eventos
2. Quero que participantes façam networking e voltem em próximos eventos
3. Quero dados sobre quem está no evento e como estão interagindo

3.2 Casos de Uso Principais

UC-001: Fazer Check-in em Evento

Campo	Descrição
ID	UC-001
Nome	Fazer Check-in em Evento
Autor Principal	Usuário (Marina, Lucas)
Pré-condições	Usuário logado, evento existe, usuário está próximo do local
Pós-condições	Check-in registrado, usuário aparece na lista do evento

Fluxo Principal:

1. Usuário abre o app
2. Sistema mostra eventos próximos baseado em GPS
3. Usuário seleciona o evento
4. Sistema valida localização (raio de 200m)
5. Usuário confirma check-in
6. Usuário define visibilidade (público/amigos/privado)
7. Usuário opcionalmente adiciona status
8. Sistema registra check-in
9. Sistema mostra confirmação com animação
10. Sistema atualiza lista de pessoas no evento

Fluxos Alternativos:

3a. Evento não aparece na lista:

1. Usuário usa busca
2. Usuário digita nome do evento
3. Sistema mostra resultados
4. Continua no passo 3

4a. Usuário fora do raio de localização:

1. Sistema mostra erro "Você parece estar longe do evento"
2. Sistema oferece opções: verificar GPS / usar QR code
3. Se QR code: usuário escaneia código no local
4. Sistema valida código e continua no passo 5

4b. GPS desabilitado:

1. Sistema solicita permissão de localização
2. Se negado: sistema oferece check-in via QR code
3. Continua no passo 5 ou encerra

Campo	Descrição
ID	UC-002
Nome	Descobrir Pessoas no Evento
Ator Principal	Usuário
Pré-condições	Usuário com check-in ativo no evento
Pós-condições	Usuário viu lista de pessoas presentes

Fluxo Principal:

1. Usuário acessa tela do evento (já com check-in)
2. Sistema mostra lista de pessoas com check-in
3. Lista ordenada por: conexões em comum > recência
4. Cada pessoa mostra: foto, nome, @instagram, nº conexões em comum, tempo do check-in
5. Usuário pode filtrar: todos / amigos / amigos de amigos
6. Usuário toca em uma pessoa
7. Sistema abre perfil da pessoa
8. Perfil mostra: foto, bio, conexões em comum, eventos em comum

Fluxos Alternativos:

2a. Nenhuma pessoa com check-in:

1. Sistema mostra empty state "Seja o primeiro a fazer check-in!"
2. Sistema sugere compartilhar evento para atrair pessoas

5a. Usuário não tem amigos no app:

1. Filtro "amigos" mostra empty state
2. Sistema sugere conectar Instagram para importar

UC-003: Enviar ReMatch

Campo	Descrição
ID	UC-003
Nome	Enviar Solicitação de ReMatch
Ator Principal	Usuário
Pré-condições	Ambos estiveram no mesmo evento, dentro de 7 dias
Pós-condições	Request enviado, destinatário notificado

Fluxo Principal:

1. Usuário acessa aba "ReMatch" ou histórico de evento
2. Sistema mostra pessoas de eventos passados (últimos 7 dias)
3. Pessoas agrupadas por evento
4. Usuário seleciona pessoa para reconectar
5. Sistema abre modal de ReMatch
6. Usuário opcionalmente adiciona mensagem (max 140 chars)
7. Usuário confirma envio
8. Sistema registra request
9. Sistema envia notificação ao destinatário
10. Sistema mostra confirmação "Solicitação enviada!"

Fluxos Alternativos:

5a. Usuário já enviou request para esta pessoa:

1. Sistema mostra "Solicitação já enviada, aguardando resposta"
2. Não permite enviar novamente

7a. Usuário atingiu limite diário:

1. Sistema mostra "Você atingiu o limite de 10 solicitações hoje"
2. Sistema informa quando poderá enviar novamente

Campo	Descrição
ID	UC-004
Nome	Responder Solicitação de ReMatch
Autor Principal	Usuário
Pré-condições	Usuário tem request pendente
Pós-condições	Request respondido, se aceito: match criado

Fluxo Principal:

1. Usuário recebe notificação "Alguém do [Evento] quer conectar"
2. Usuário toca na notificação
3. Sistema abre tela de requests pendentes
4. Usuário vê: foto, nome, evento em comum, mensagem (se houver)
5. Usuário escolhe: Aceitar / Ignorar / Bloquear
6. Se Aceitar:
 - Sistema cria conexão mútua
 - Sistema mostra celebração "É um Match!  "
 - Sistema mostra botão para ver perfil Instagram
7. Se Ignorar:
 - Sistema remove da lista
 - Nenhuma notificação ao remetente
8. Se Bloquear:
 - Sistema remove da lista
 - Sistema impede futuras interações

UC-005: Criar Evento (Organizador)

Campo	Descrição
ID	UC-005
Nome	Criar Evento como Organizador
Autor Principal	Organizador (Carla)
Pré-condições	Usuário com conta de organizador
Pós-condições	Evento criado e disponível para check-ins

Fluxo Principal:

1. Organizador acessa dashboard web
2. Clica em "Criar Evento"
3. Preenche informações obrigatórias:
 - Nome do evento
 - Data e horário (início/fim)
 - Local (endereço ou online)
 - Descrição
4. Preenche informações opcionais:
 - Imagem de capa
 - Categorias/tags
 - Capacidade máxima
 - Link para ingressos
5. Configura check-in:
 - Tipo: geolocalizado / QR code / ambos
 - Raio de geolocalização (50-500m)
 - Permitir check-in antecipado (0-24h antes)
6. Configura privacidade:
 - Visibilidade: público / privado / unlisted
 - Lista de participantes: visível / oculta
7. Clica em "Publicar"
8. Sistema cria evento
9. Sistema gera QR code

10. Sistema mostra página do evento com link compartilhável

UC-006: Ver Dashboard do Evento (Organizador)

Campo	Descrição
ID	UC-006
Nome	Visualizar Dashboard do Evento
Autor Principal	Organizador
Pré-condições	Evento existe e pertence ao organizador
Pós-condições	Organizador viu métricas do evento

Fluxo Principal:

1. Organizador acessa dashboard web
 2. Seleciona evento na lista
 3. Sistema mostra métricas em tempo real:
 - Total de check-ins
 - Pessoas presentes agora
 - Pico de presença
 - ReMatches originados do evento
 4. Sistema mostra lista de participantes (se configurado)
 5. Sistema mostra gráfico de check-ins ao longo do tempo
 6. Organizador pode exportar dados (CSV)
 7. Organizador pode baixar QR code em alta resolução
-

4. Requisitos Funcionais

4.1 Módulo: Autenticação (AUTH)

ID	Requisito	Prioridade	Critério de Aceite
AUTH-001	Login com Instagram OAuth	P0	Usuário consegue logar usando conta Instagram
AUTH-002	Login com Email (backup)	P1	Usuário consegue logar com email + senha
AUTH-003	Importar dados do Instagram	P0	Sistema importa foto, nome, @, bio
AUTH-004	Criar sessão persistente	P0	Usuário permanece logado por 30 dias
AUTH-005	Logout	P0	Usuário consegue sair da conta
AUTH-006	Recuperação de senha	P1	Usuário recebe email para resetar
AUTH-007	Deletar conta	P0	Usuário pode deletar conta e dados (LGPD)

Detalhamento AUTH-001: Login com Instagram

Fluxo Técnico:

1. App abre WebView com URL de autorização Instagram
2. Usuário autoriza o app
3. Instagram redireciona com código de autorização
4. Backend troca código por access_token
5. Backend busca dados do usuário (me?fields=id,username,name,profile_picture_url)
6. Backend cria/atualiza usuário no banco
7. Backend gera JWT token
8. App armazena JWT no SecureStorage
9. App navega para Home

Permissões necessárias:

- user_profile (básico)
- user_media (futuro, para stories)

Tratamento de erros:

- Instagram indisponível: mostrar opção de email
- Usuário cancela: voltar para tela de login
- Token expirado: solicitar reautorização

4.2 Módulo: Perfil de Usuário (PROFILE)

ID	Requisito	Prioridade	Critério de Aceite
PROFILE-001	Visualizar meu perfil	P0	Usuário vê todas suas informações
PROFILE-002	Editar nome	P0	Usuário pode alterar nome
PROFILE-003	Editar foto	P0	Usuário pode trocar foto
PROFILE-004	Editar bio	P0	Usuário pode editar bio (max 140 chars)
PROFILE-005	Editar cidade	P1	Usuário pode definir cidade
PROFILE-006	Vincular/desvincular Instagram	P0	Usuário pode reconectar Instagram
PROFILE-007	Vincular LinkedIn	P2	Usuário pode adicionar LinkedIn
PROFILE-008	Ver histórico de check-ins	P0	Usuário vê todos os check-ins passados
PROFILE-009	Ver minhas conexões	P0	Usuário vê lista de matches
PROFILE-010	Ver estatísticas	P1	Total check-ins, eventos, conexões

Campos do Perfil:

```
typescript
```

```

interface UserProfile {
    // Obrigatórios
    id: string;           // UUID
    name: string;          // 2-50 chars
    photo_url: string;     // URL da foto

    // Opcionais
    bio?: string;          // max 140 chars
    city?: string;          // max 50 chars

    // Redes sociais
    instagram_username?: string; // sem @@
    instagram_verified: boolean; // se veio do OAuth
    linkedin_url?: string;      // URL completa

    // Metadados
    created_at: datetime;
    updated_at: datetime;
    last_active_at: datetime;

    // Estatísticas (computed)
    total_checkins: number;
    total_events: number;
    total_connections: number;
}

```

4.3 Módulo: Check-in (CHECKIN)

ID	Requisito	Prioridade	Critério de Aceite
CHECKIN-001	Listar eventos próximos	P0	Sistema mostra eventos em raio de 10km
CHECKIN-002	Buscar evento por nome	P0	Usuário encontra evento digitando
CHECKIN-003	Validar localização GPS	P0	Check-in só funciona dentro do raio
CHECKIN-004	Check-in via QR code	P0	Usuário pode escanear QR do evento
CHECKIN-005	Definir visibilidade	P0	Usuário escolhe público/amigos/privado
CHECKIN-006	Adicionar status	P1	Usuário pode adicionar texto/emoji
CHECKIN-007	Check-out manual	P0	Usuário pode sair do check-in
CHECKIN-008	Check-out automático	P0	Sistema remove após X horas

ID	Requisito	Prioridade	Critério de Aceite
CHECKIN-009	Ver check-in ativo	P0	Usuário vê onde está com check-in
CHECKIN-010	Histórico de check-ins	P0	Usuário vê check-ins passados
CHECKIN-011	Límite de check-in ativo	P0	Máximo 1 check-in ativo por vez

Modelo de Dados Check-in:

typescript

```
interface Checkin {
    id: string;           // UUID
    user_id: string;      // FK para User
    event_id?: string;    // FK para Event (se for evento)
    place_id?: string;    // FK para Place (se for lugar)

    // Timestamps
    started_at: datetime; // Quando fez check-in
    ended_at?: datetime;  // Quando fez check-out (null se ativo)
    expires_at: datetime; // Quando expira automaticamente

    // Configurações
    visibility: 'public' | 'friends' | 'private';
    status_text?: string;   // max 140 chars
    status_emoji?: string;  // emoji único

    // Validação
    validated_by: 'gps' | 'qr_code' | 'manual';
    location?: {            // Se GPS
        latitude: number;
        longitude: number;
        accuracy: number;
    };

    // Metadados
    created_at: datetime;
    updated_at: datetime;
}
```

Regras de Negócio Check-in:

RN-CHECKIN-001: Raio de validação GPS

- Padrão: 200 metros do centro do evento
- Configurável pelo organizador: 50m a 500m
- Se accuracy do GPS > 100m, solicitar QR code

RN-CHECKIN-002: Expiração automática

- Eventos: expira no horário de fim do evento + 1 hora
- Se evento não tem fim: expira em 8 horas
- Lugares (sem evento): expira em 4 horas

RN-CHECKIN-003: Check-in único

- Usuário só pode ter 1 check-in ativo
- Novo check-in automaticamente encerra o anterior

RN-CHECKIN-004: Check-in antecipado

- Eventos podem permitir check-in X horas antes
- Padrão: 30 minutos antes do início
- Configurável: 0 a 24 horas

RN-CHECKIN-005: Check-in pós-evento

- Não permitido após término do evento
- Grace period: 1 hora após término

4.4 Módulo: Discovery (DISCOVERY)

ID	Requisito	Prioridade	Critério de Aceite
DISC-001	Listar pessoas no evento	P0	Ver quem tem check-in ativo
DISC-002	Ordenar por relevância	P0	Amigos > amigos de amigos > outros
DISC-003	Mostrar conexões em comum	P0	Ver quantos amigos em comum
DISC-004	Filtrar por tipo	P1	Todos / amigos / amigos de amigos
DISC-005	Ver perfil de pessoa	P0	Abrir perfil com detalhes
DISC-006	Respeitar privacidade	P0	Não mostrar check-ins privados
DISC-007	Indicar tempo de check-in	P1	"Há 30 min", "Há 2h"
DISC-008	Buscar pessoa no evento	P2	Buscar por nome dentro do evento

Algoritmo de Ordenação:

Score de Relevância =

```
(conexões_em_comum * 100) +  
(é_amigo_direto ? 500 : 0) +  
(é_amigo_de_amigo ? 200 : 0) +  
(eventos_em_comum_anteriores * 50) +  
(100 - minutos_desde_checkin) // mais recente = mais relevante
```

Ordenação: DESC por score

Límite por página: 20 pessoas

Cálculo de Conexões em Comum:

Fontes de conexão:

1. Amigos importados do Instagram (seguindo mutuamente)
2. Matches dentro do Check Point
3. (Futuro) Conexões LinkedIn

Graus de separação mostrados:

- 1º grau: "Amigo" (conexão direta)
- 2º grau: "X amigos em comum"
- 3º+ grau: não mostrar indicador

4.5 Módulo: ReMatch (REMATCH)

ID	Requisito	Prioridade	Critério de Aceite
REMATCH-001	Listar pessoas para ReMatch	P0	Ver pessoas de eventos passados
REMATCH-002	Agrupar por evento	P0	Organizar lista por evento
REMATCH-003	Janela de 7 dias	P0	Só mostrar últimos 7 dias
REMATCH-004	Enviar solicitação	P0	Enviar request com mensagem opcional
REMATCH-005	Receber solicitação	P0	Ver requests pendentes
REMATCH-006	Aceitar solicitação	P0	Criar match quando aceito
REMATCH-007	Ignorar solicitação	P0	Remover sem notificar
REMATCH-008	Bloquear pessoa	P0	Impedir interações futuras
REMATCH-009	Limite diário	P0	Máximo 10 requests/dia

ID	Requisito	Prioridade	Critério de Aceite
REMATCH-010	Ver matches existentes	P0	Listar conexões feitas
REMATCH-011	Link para Instagram	P0	Abrir perfil Instagram do match

Modelo de Dados ReMatch:

typescript

```
interface RematchRequest {
    id: string;           // UUID
    sender_id: string;    // FK para User
    receiver_id: string;  // FK para User
    event_id: string;     // FK para Event (contexto)

    message?: string;     // max 140 chars

    status: 'pending' | 'accepted' | 'ignored' | 'blocked';

    created_at: datetime;
    responded_at?: datetime;
}

interface Connection {
    id: string;           // UUID
    user_a_id: string;    // FK para User
    user_b_id: string;    // FK para User

    origin_event_id: string; // Evento onde se conheceram
    origin_request_id: string; // Request que originou

    created_at: datetime;
}
```

Regras de Negócio ReMatch:

RN-REMATCH-001: Elegibilidade

- Ambos usuários devem ter feito check-in no mesmo evento
- Evento deve ter ocorrido nos últimos 7 dias
- Usuários não podem já estar conectados
- Nenhum dos dois pode ter bloqueado o outro

RN-REMATCH-002: Limite diário

- Máximo 10 requests enviados por dia
- Reset à meia-noite (timezone do usuário)
- Requests ignorados não são reembolsados

RN-REMATCH-003: Notificação de request

- Não revelar nome do remetente na notificação
- Texto: "Alguém do [Evento] quer conectar com você"
- Revelar nome apenas quando abrir o request

RN-REMATCH-004: Comportamento do bloqueio

- Usuário bloqueado não pode enviar novos requests
- Usuário bloqueado não vê o bloqueador na discovery
- Bloqueio é unidirecional (A bloqueia B, B ainda vê A)
- Bloquear remove qualquer request pendente

RN-REMATCH-005: Expiração de requests

- Requests pendentes expiram em 14 dias
- Requests expirados são marcados como 'expired'
- Usuário pode reenviar após expiração

4.6 Módulo: Eventos (EVENTS)

ID	Requisito	Prioridade	Critério de Aceite
EVENT-001	Criar evento	P0	Organizador cria evento
EVENT-002	Editar evento	P0	Organizador edita informações
EVENT-003	Cancelar evento	P1	Organizador cancela evento
EVENT-004	Gerar QR code	P0	Sistema gera QR único
EVENT-005	Definir configurações de check-in	P0	Raio, tipo, antecedência
EVENT-006	Definir privacidade	P0	Público/privado/unlisted
EVENT-007	Ver lista de participantes	P0	Organizador vê quem fez check-in

ID	Requisito	Prioridade	Critério de Aceite
EVENT-008	Exportar participantes	P1	Download CSV
EVENT-009	Ver métricas em tempo real	P0	Dashboard com contadores
EVENT-010	Múltiplos organizadores	P2	Adicionar co-organizadores

Modelo de Dados Evento:

```
typescript
```

```
interface Event {  
    id: string;           // UUID  
    organizer_id: string; // FK para User  
  
    // Informações básicas  
    name: string;         // 3-100 chars  
    description?: string; // max 2000 chars  
    image_url?: string;   // URL da capa  
  
    // Data e hora  
    starts_at: datetime;  
    ends_at?: datetime;   // opcional  
    timezone: string;     // ex: "America/Sao_Paulo"  
  
    // Localização  
    location_type: 'physical' | 'online' | 'hybrid';  
    address?: string;      // para físico  
    latitude?: number;  
    longitude?: number;  
    online_url?: string;   // para online  
  
    // Configurações de check-in  
    checkin_type: 'gps' | 'qr' | 'both';  
    checkin_radius_meters: number; // 50-500, default 200  
    checkin_early_minutes: number; // 0-1440, default 30  
  
    // Privacidade  
    visibility: 'public' | 'private' | 'unlisted';  
    show_attendee_list: boolean;  
    allow_rematch: boolean;  
  
    // Metadados  
    status: 'draft' | 'published' | 'cancelled' | 'finished';  
    category?: string;      // tech, music, fitness, etc  
    capacity?: number;  
    external_url?: string;  // link para ingressos  
  
    // QR Code  
    qr_code_token: string;  // token único para QR  
  
    // Timestamps  
    created_at: datetime;  
    updated_at: datetime;  
    published_at?: datetime;  
}
```

4.7 Módulo: Dashboard Organizador (DASHBOARD)

ID	Requisito	Prioridade	Critério de Aceite
DASH-001	Login web	P0	Organizador acessa via browser
DASH-002	Listar meus eventos	P0	Ver todos eventos criados
DASH-003	Métricas em tempo real	P0	Atualização automática
DASH-004	Gráfico temporal	P1	Check-ins ao longo do tempo
DASH-005	Lista de participantes	P0	Nome, foto, hora do check-in
DASH-006	Exportar dados	P1	CSV com participantes
DASH-007	Download QR code	P0	PNG/SVG em alta resolução
DASH-008	Configurações do evento	P0	Editar evento existente

Métricas do Dashboard:

```
typescript

interface EventMetrics {
    // Contadores básicos
    total_checkins: number;      // Check-ins únicos
    current_attendees: number;   // Com check-in ativo agora
    peak_attendees: number;      // Máximo simultâneo
    peak_time: datetime;         // Quando foi o pico

    // ReMatch
    rematches_originated: number; // ReMatches entre participantes
    rematch_rate: number;        // % que enviou pelo menos 1

    // Engajamento
    avg_duration_minutes: number; // Tempo médio de presença
    return_rate: number;          // % que veio em evento anterior

    // Série temporal
    checkins_by_hour: {
        hour: datetime;
        count: number;
    }[];
}
```

4.8 Módulo: Configurações (SETTINGS)

ID	Requisito	Prioridade	Critério de Aceite
SETT-001	Visibilidade padrão	P0	Definir padrão de check-ins
SETT-002	Quem pode enviar ReMatch	P1	Todos / amigos de amigos / ninguém
SETT-003	Notificações push	P1	Ligar/desligar por tipo
SETT-004	Notificações email	P1	Ligar/desligar
SETT-005	Modo invisível	P2	Fazer check-in sem aparecer
SETT-006	Idioma	P2	Português / Inglês
SETT-007	Tema	P2	Claro / Escuro / Sistema
SETT-008	Exportar meus dados	P0	LGPD compliance
SETT-009	Deletar conta	P0	LGPD compliance

Modelo de Configurações:

```
typescript
```

```

interface UserSettings {
    user_id: string;

    // Privacidade
    default_checkin_visibility: 'public' | 'friends' | 'private';
    rematch_from: 'everyone' | 'friends_of_friends' | 'nobody';
    invisible_mode: boolean;

    // Notificações Push
    push_enabled: boolean;
    push_friend_checkin: boolean;
    push_rematch_request: boolean;
    push_rematch_accepted: boolean;
    push_eventReminder: boolean;

    // Notificações Email
    email_enabled: boolean;
    email_weekly_digest: boolean;
    email_event_updates: boolean;

    // Aparência
    language: 'pt-BR' | 'en';
    theme: 'light' | 'dark' | 'system';

    updated_at: datetime;
}

```

5. Requisitos Não-Funcionais

5.1 Performance

ID	Requisito	Especificação
PERF-001	Tempo de carregamento inicial	< 3 segundos em 4G
PERF-002	Tempo de resposta API	< 500ms p95
PERF-003	Tempo de check-in	< 2 segundos end-to-end
PERF-004	Atualização de lista discovery	< 1 segundo
PERF-005	Suporte a usuários simultâneos	1.000 concurrent (MVP)

5.2 Disponibilidade

ID	Requisito	Especificação
AVAIL-001	Uptime	99.5% mensal
AVAIL-002	RPO (Recovery Point Objective)	1 hora
AVAIL-003	RTO (Recovery Time Objective)	4 horas
AVAIL-004	Backup de dados	Diário, retenção 30 dias

5.3 Segurança

ID	Requisito	Especificação
SEC-001	Criptografia em trânsito	TLS 1.2+ obrigatório
SEC-002	Criptografia em repouso	AES-256 para dados sensíveis
SEC-003	Autenticação	JWT com expiração 30 dias
SEC-004	Rate limiting	100 req/min por IP
SEC-005	Proteção contra injection	Prepared statements, input validation
SEC-006	LGPD compliance	Exportação e deleção de dados

5.4 Escalabilidade

ID	Requisito	Especificação
SCAL-001	Arquitetura	Stateless, horizontalmente escalável
SCAL-002	Database	PostgreSQL com read replicas
SCAL-003	Cache	Redis para sessões e dados frequentes
SCAL-004	CDN	Assets estáticos via CDN

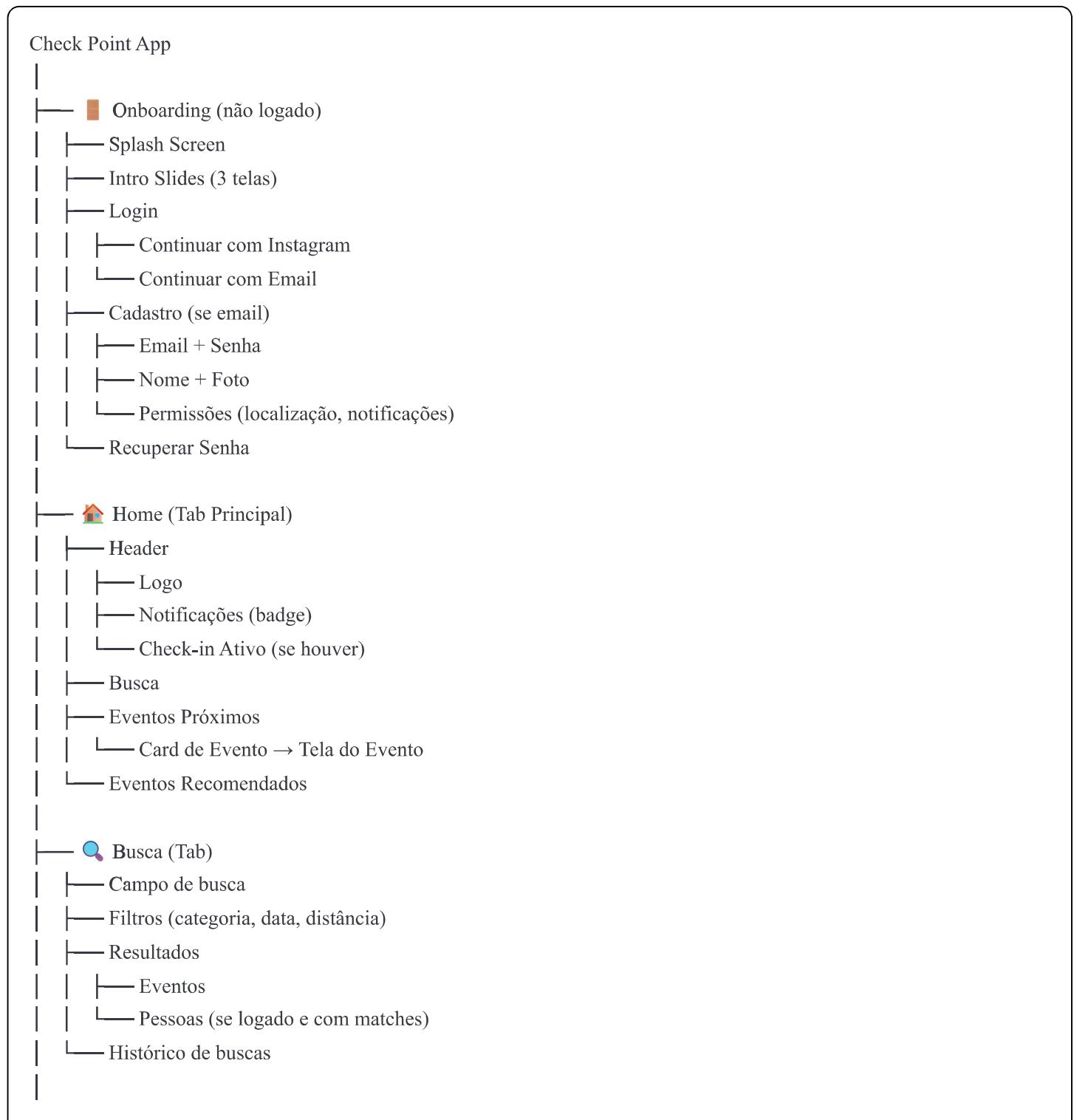
5.5 Compatibilidade

ID	Requisito	Especificação
COMPAT-001	iOS	iOS 14+

ID	Requisito	Especificação
COMPAT-002	Android	Android 8+ (API 26)
COMPAT-003	Browsers (dashboard)	Chrome, Safari, Firefox (últimas 2 versões)
COMPAT-004	Resoluções mobile	320px a 428px largura

6. Arquitetura de Informação

6.1 Mapa do Aplicativo (Mobile)



└──  Check-in (Tab Central / FAB)

└── Mapa com eventos próximos

└── Lista de eventos próximos

└── Scanner QR Code

└── Confirmar Check-in

└── Visibilidade

└── Status (opcional)

└── Confirmar

└──  ReMatch (Tab)

└── Solicitações Pendentes

└── Card de Request → Aceitar/Ignorar/Bloquear

└── Pessoas para ReMatch (por evento)

└── Card de Pessoa → Enviar Request

└── Conexões Feitas

└── Card de Match → Ver Perfil / Instagram

└──  Perfil (Tab)

└── Meu Perfil

└── Foto

└── Nome

└── Bio

└── Redes sociais

└── Estatísticas

└── Editar Perfil

└── Histórico de Check-ins

└── Minhas Conexões

└── Configurações

└── Privacidade

└── Notificações

└── Conta

└── Sobre

└── Logout

└──  Tela do Evento

└── Header (imagem, nome, data)

└── Informações

└── Data/hora

└── Local (mapa)

└── Descrição

└── Ação Principal

└── Fazer Check-in (se não fez)

└── Ver Check-in (se já fez)

└── Pessoas no Evento

└── Contador

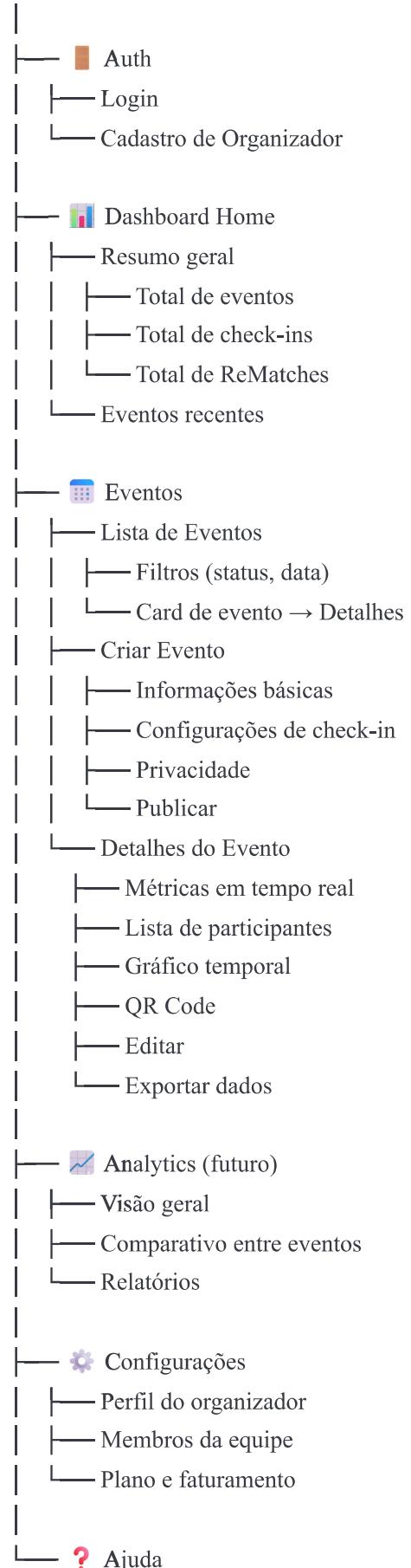
└── Lista (resumo)

└── Ver Todos → Lista Completa

- └ Compartilhar
- └  Lista de Pessoas no Evento
 - ├ Filtros (todos/amigos/amigos de amigos)
 - ├ Lista de pessoas
 - └ Card de Pessoa → Perfil
 - └ Busca por nome
- └  Perfil de Outra Pessoa
 - ├ Foto
 - ├ Nome
 - ├ Bio
 - ├ Eventos em comum
 - ├ Conexões em comum
 - ├ Redes sociais (se conectados)
 - └ Ações
 - ├ Enviar ReMatch (se elegível)
 - ├ Ver Instagram (se conectados)
 - └ Bloquear/Reportar
- └  Notificações
 - ├ Lista de notificações
 - └ ReMatch requests
 - └ Matches aceitos
 - └ Amigos com check-in
 - └ Lembretes de eventos
 - └ Marcar como lidas
- └  Configurações
 - ├ Privacidade
 - └ Visibilidade padrão
 - └ Quem pode enviar ReMatch
 - └ Modo invisível
 - ├ Notificações
 - └ Push
 - └ Email
 - ├ Conta
 - └ Alterar email
 - └ Alterar senha
 - └ Redes vinculadas
 - └ Exportar dados
 - └ Deletar conta
 - └ Sobre
 - └ Versão do app
 - └ Termos de uso

6.2 Mapa do Dashboard Web (Organizador)

Check Point Dashboard



- Documentação
- Tutoriais
- Suporte

7. Especificação de Telas

7.1 Telas de Onboarding

T-001: Splash Screen

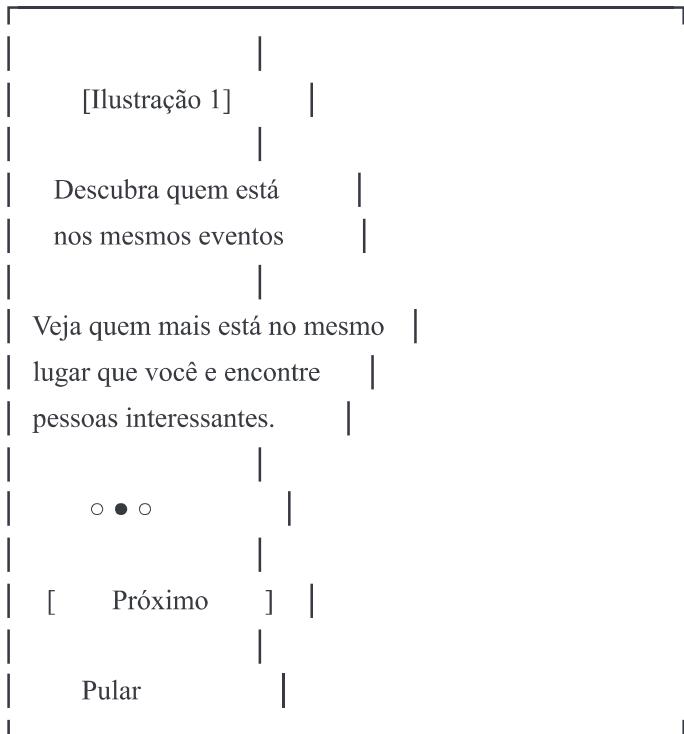
Comportamento:

- Duração: 1.5s ou até verificar auth
- Se logado: navega para Home
- Se não logado: navega para Intro

Elementos:

- Logo: 120px, centralizado
- Nome: Inter Bold 24px
- Loading: indicador animado
- Background: gradient primary

T-002: Intro Slides



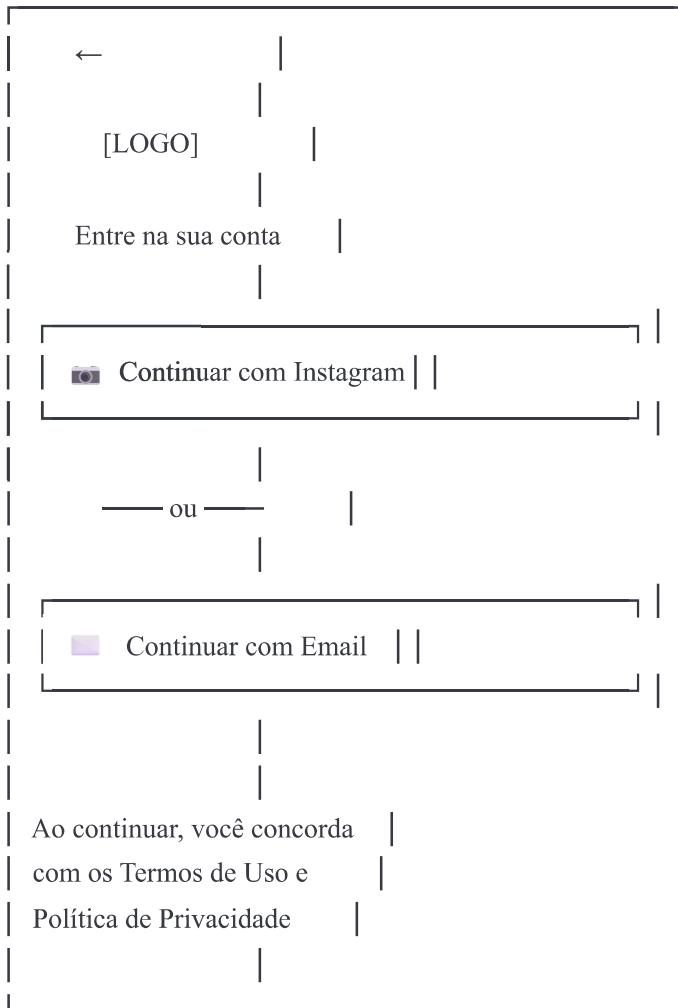
Slides:

1. "Descubra quem está nos mesmos eventos"
2. "Conecte-se com quem você conheceu"
3. "Nunca mais perca uma conexão"

Elementos:

- Ilustração: 200px altura
- Título: Inter Bold 24px
- Descrição: Inter Regular 16px, #64748B
- Indicadores: 8px círculos
- Botão Próximo: primary button
- Link Pular: ghost link

T-003: Login



Elementos:

- Back button: se veio de algum lugar
- Logo: 80px
- Título: Inter SemiBold 22px
- Botão Instagram: #E4405F background
- Botão Email: secondary button
- Links: 14px, #7C3AED underline

T-004: Cadastro - Step 1 (Email/Senha)

← Criar conta |

Step 1 de 3 |

Qual seu email? |

email@example.com ||

Crie uma senha |

•••••••• •||

Mínimo 8 caracteres |

Confirme a senha |

•••••••• •||

[Continuar] |

Validações:

- Email: formato válido, não duplicado
- Senha: min 8 chars, 1 maiúscula, 1 número
- Confirmar: deve ser igual

Estados de erro:

- Email inválido: "Digite um email válido"
- Email em uso: "Este email já está cadastrado"
- Senha fraca: "Senha deve ter pelo menos..."
- Senhas diferentes: "As senhas não coincidem"

T-005: Cadastro - Step 2 (Perfil)

← Criar conta

Step 2 de 3

Adicionar foto

Como quer ser chamado?

Seu nome

Fale um pouco sobre você

87/140

[Continuar]

This diagram illustrates a user interface flow for account creation. It starts with a back arrow and a 'Criar conta' button at the top. Below is a progress bar indicating 'Step 2 de 3' is complete. A large rectangular input field for adding a photo is shown, containing a camera icon and a plus sign. The next section asks for a name, with a placeholder 'Seu nome'. Following this is a bio input field with the placeholder 'Fale um pouco sobre você', which includes a character counter showing '87/140'. At the bottom is a 'Continuar' button enclosed in brackets.

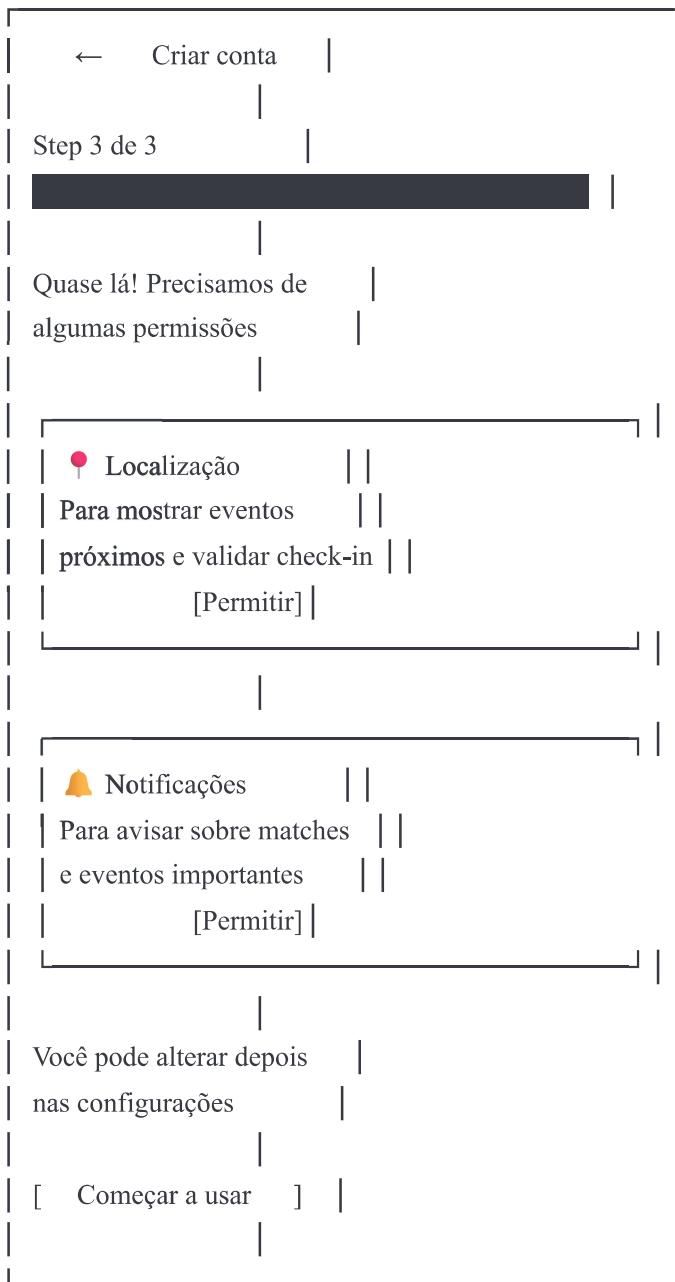
Elementos:

- Foto: 100px círculo, tap para selecionar
- Nome: obrigatório, 2-50 chars
- Bio: opcional, max 140 chars, contador

Ação foto:

- Bottom sheet: Tirar foto / Escolher da galeria
- Crop circular
- Upload e preview

T-006: Cadastro - Step 3 (Permissões)

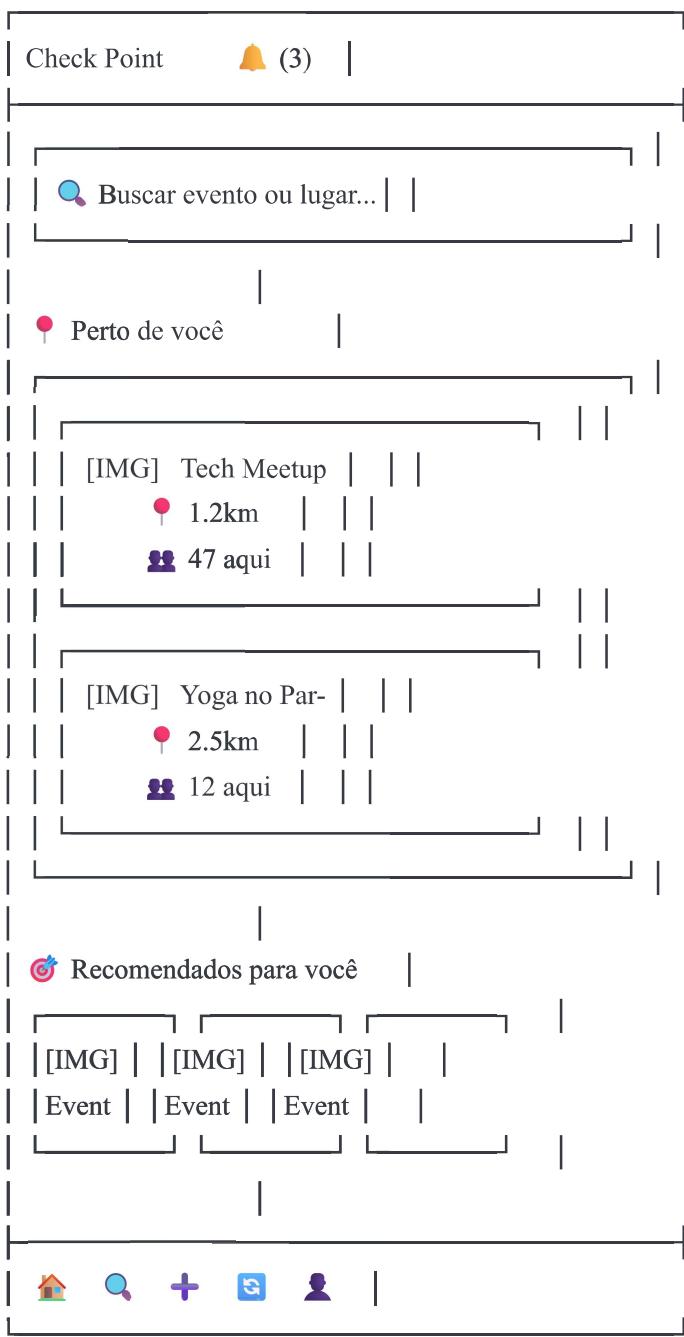


Comportamento:

- Localização: obrigatória para funcionar
- Notificações: opcional, mas recomendada
- Se negar localização: mostrar explicação
- Botão fica enabled mesmo negando notificações

7.2 Telas Principais (Home)

T-010: Home



Elementos:

- Header: logo + badge notificações
- Busca: input que navega para tela de busca
- Seção Perto: scroll horizontal de cards
- Seção Recomendados: baseado em histórico

Card de Evento (horizontal):

- Imagem: 80x80px, border-radius 12px
- Nome: Inter SemiBold 16px, max 2 linhas
- Distância: Inter Regular 14px, #64748B
- Contador: Inter Medium 14px, #7C3AED
- Tap: navega para T-030 (Tela do Evento)

Tab Bar:

- Home (filled quando ativo)

- Busca
- Check-in (FAB destacado)
- ReMatch
- Perfil

T-011: Home com Check-in Ativo

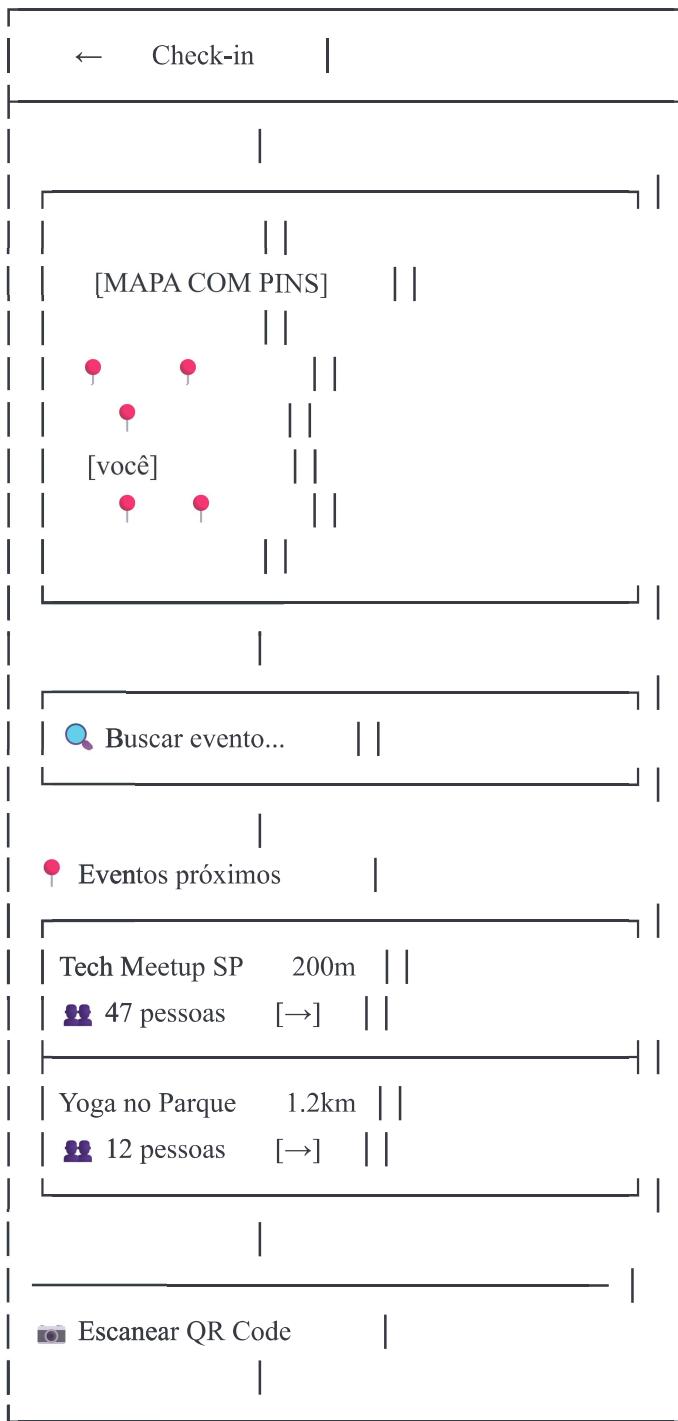
The screenshot shows the mobile application's home screen. At the top, there is a banner titled "Check Point" with a bell icon indicating 3 notifications. Below the banner, a pink location pin icon is followed by the text "Você está em". To its right is the text "Tech Meetup SP" and "Há 45 min · 47 pessoas". A "[Ver →]" button is located at the bottom right of this section. Below this, there is a search bar with a magnifying glass icon and the placeholder text "Buscar evento ou lugar...". Further down, the text "...resto igual T-010..." is visible. At the very bottom, there is a navigation bar with icons for home, search, a checked box, a blue square with a white symbol, and a profile.

Banner Check-in Ativo:

- Background: gradient primary (util)
- Mostra evento, tempo, contador
- Tap no banner: vai para tela do evento
- Botão Ver: vai para tela do evento
- FAB muda de para

7.3 Telas de Check-in

T-020: Fazer Check-in (Mapa)



Elementos:

- Mapa: Google Maps / Apple Maps
- Pin você: diferenciado, cor primária
- Pins eventos: cor secundária
- Raio de alcance: círculo sutil
- Lista: eventos ordenados por distância
- Link QR: abre scanner de câmera

T-021: Confirmar Check-in

x Check-in |

[IMG EVENTO] || |

Tech Meetup SP |

📍 WeWork Paulista |

📅 Hoje, 19:00 - 22:00 |

Quem pode ver seu check-in? |

Público	Amigos	Privado	
•			

Adicionar status (opcional) |

O que você está fazendo? || |

0/140 || |

[Fazer Check-in] |

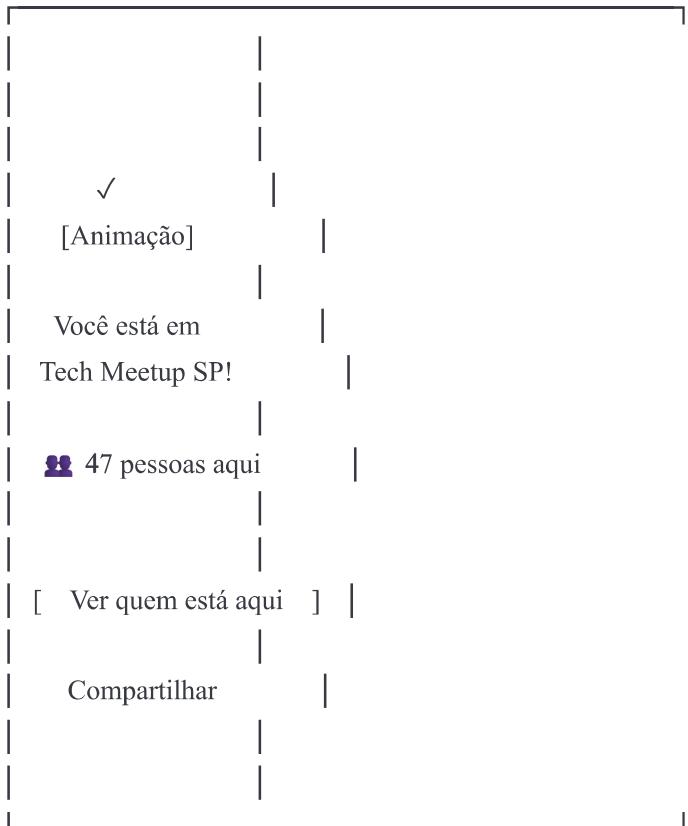
Elementos:

- Imagem evento: 100% largura, 150px altura
- Info evento: nome, local, horário
- Seletor visibilidade: 3 opções, radio
- Status: textarea opcional
- Botão: primary, full width

Visibilidades:

- Público: qualquer pessoa no app vê
- Amigos: só conexões veem
- Privado: só você vê (para histórico)

T-022: Check-in Confirmado (Success)



Elementos:

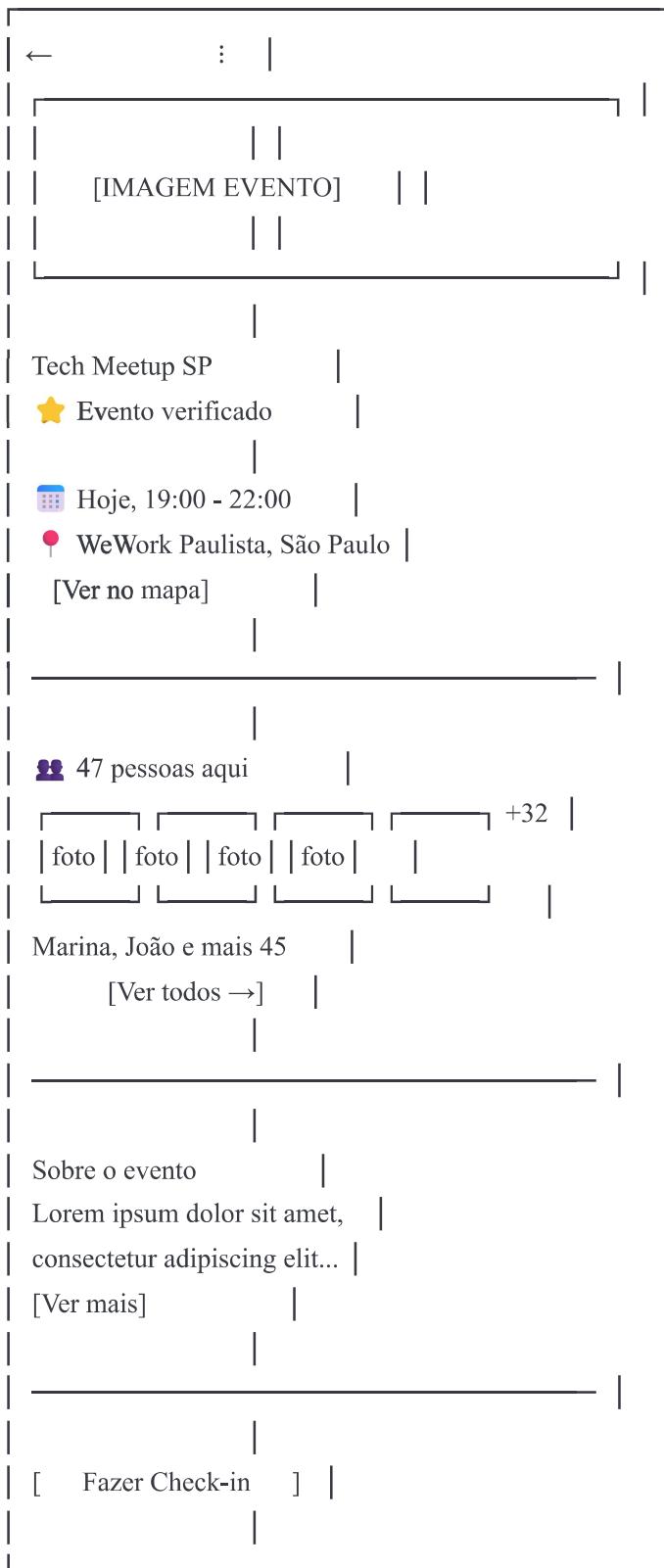
- Animação: checkmark + confetti
- Duração: 2 segundos
- Auto-dismiss: não, espera ação
- Botão primário: ver discovery
- Botão secundário: compartilhar stories

Compartilhar:

- Gera imagem para Stories
- Template com logo, evento, @ do usuário
- Deep link para o app

7.4 Telas de Discovery

T-030: Tela do Evento



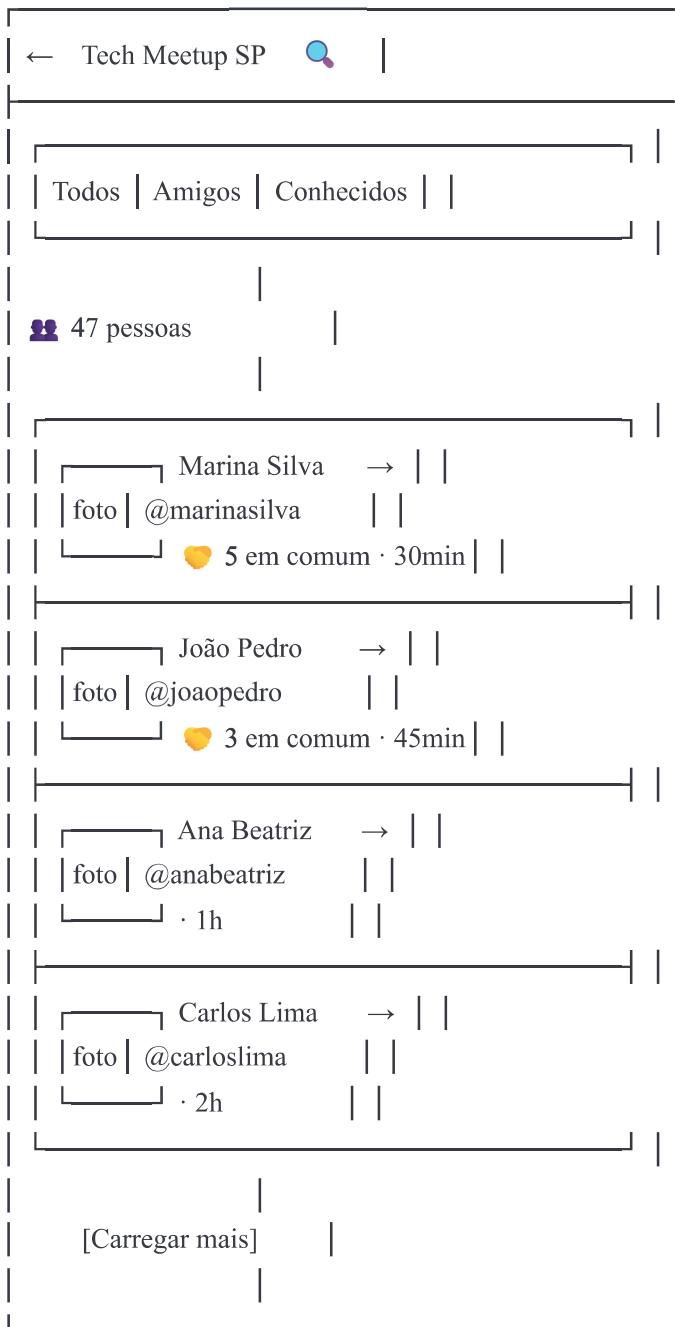
Elementos:

- Header: back + menu (compartilhar, reportar)
- Imagem: parallax scroll effect
- Badge verificado: eventos parceiros
- Info: data, local com link mapa
- Pessoas: avatares + contador + CTA
- Descrição: truncada com "ver mais"
- CTA: check-in (ou "você está aqui" se já fez)

Estados do CTA:

- Não fez check-in: "Fazer Check-in"
- Já fez check-in: "Você está aqui ✓" (vai para discovery)
- Evento futuro: "Lembrar-me" (salva evento)
- Evento passado: "Ver quem estava" (se habilitado)

T-031: Lista de Pessoas no Evento



Elementos:

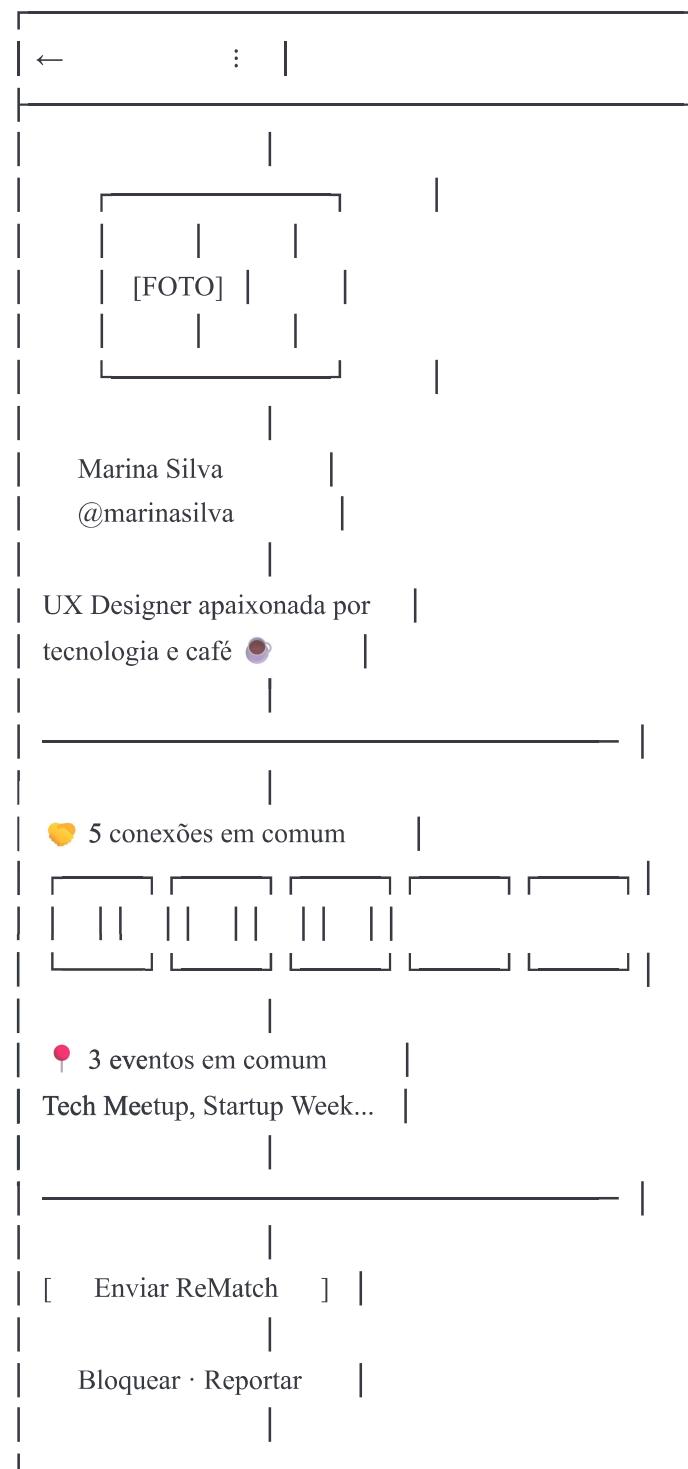
- Tabs filtro: todos / amigos / conhecidos
- Contador total
- Lista ordenada por relevância
- Card pessoa: foto, nome, @, conexões, tempo
- Busca no header: filtrar por nome
- Paginação: 20 por página

Card de Pessoa:

- Foto: 48px círculo
- Nome: Inter SemiBold 16px
- Instagram: Inter Regular 14px, #7C3AED
- Conexões: se > 0, mostrar com ícone

- Tempo: "Há 30min", "Há 2h"
- Tap: vai para perfil da pessoa

T-032: Perfil de Outra Pessoa



Elementos:

- Foto: 100px círculo
- Nome: Inter Bold 22px
- Instagram: link para perfil
- Bio: Inter Regular 16px
- Conexões em comum: avatares clicáveis
- Eventos em comum: lista resumida
- CTA: ReMatch se elegível

Estados do CTA:

- Pode enviar: "Enviar ReMatch"
- Já enviou: "Solicitação enviada" (disabled)

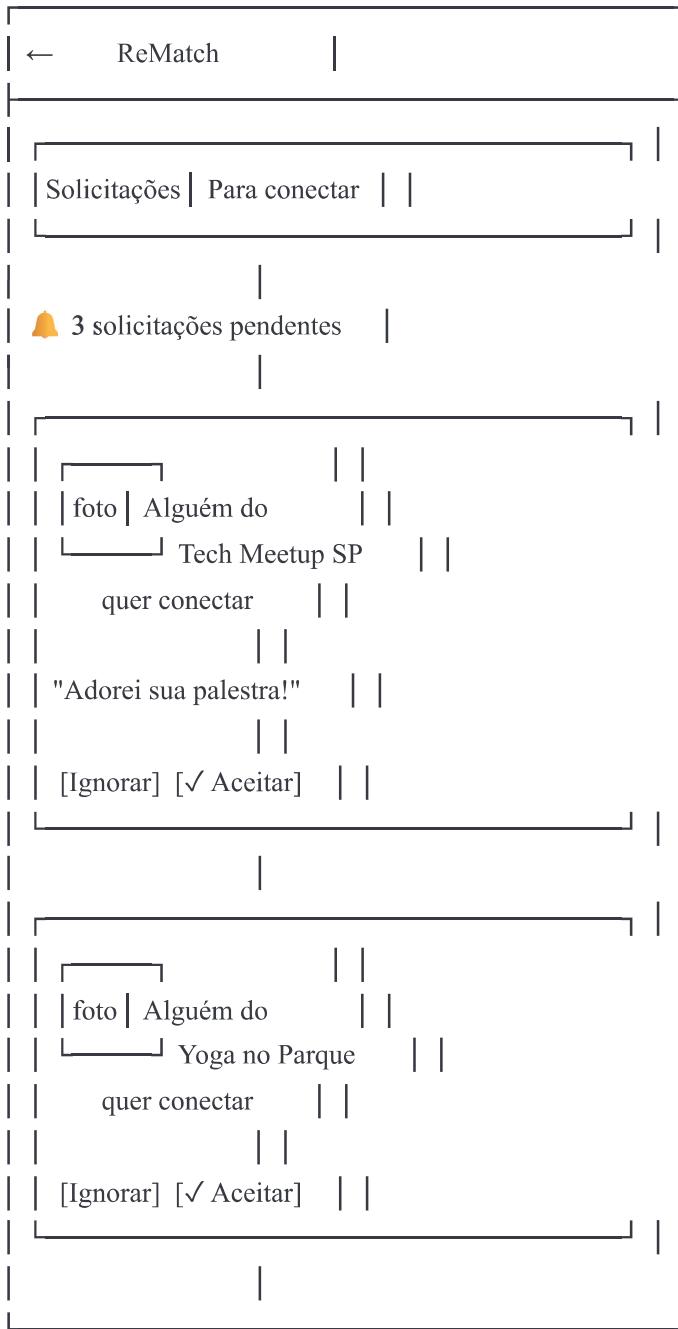
- Já conectados: "Vocês estão conectados ✓"
- Sem evento comum: CTA não aparece

Menu (:):

- Compartilhar perfil
 - Bloquear
 - Reportar
-

7.5 Telas de ReMatch

T-040: ReMatch - Solicitações

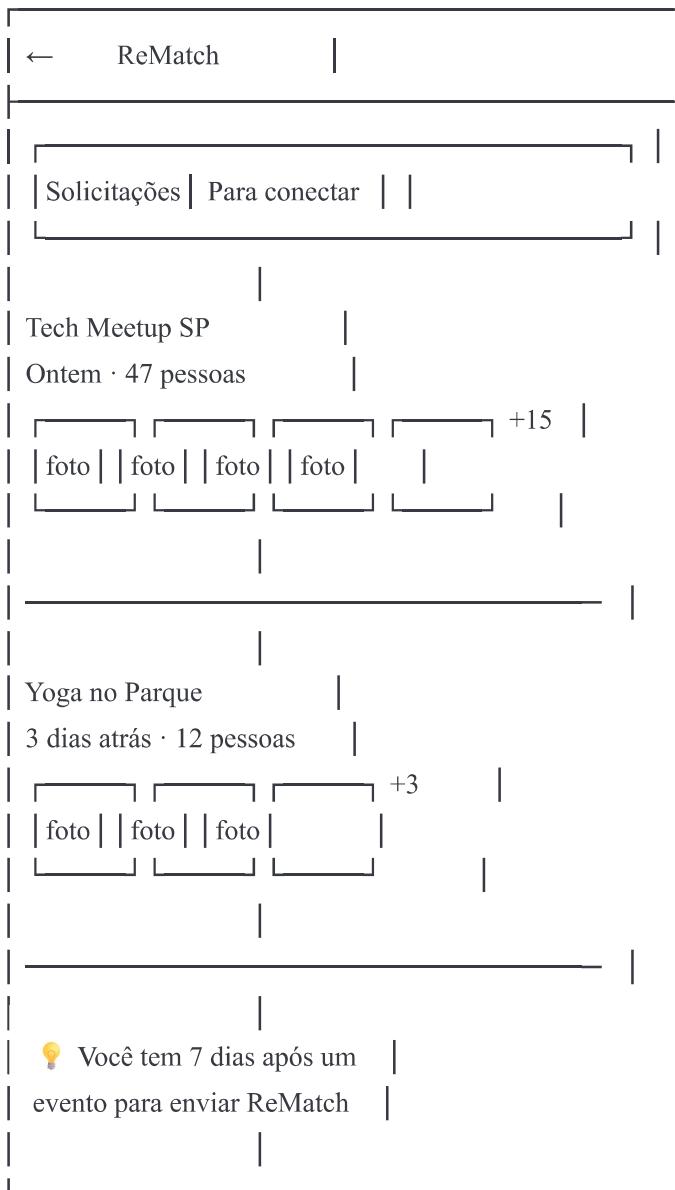


Comportamento:

- Não revelar nome até aceitar
- Mostrar evento em comum
- Mostrar mensagem se houver
- Tap no card: ver mais detalhes
- Swipe left: ignorar
- Swipe right: aceitar

Botões:

- Ignorar: secondary, remove silenciosamente
- Aceitar: primary, cria match



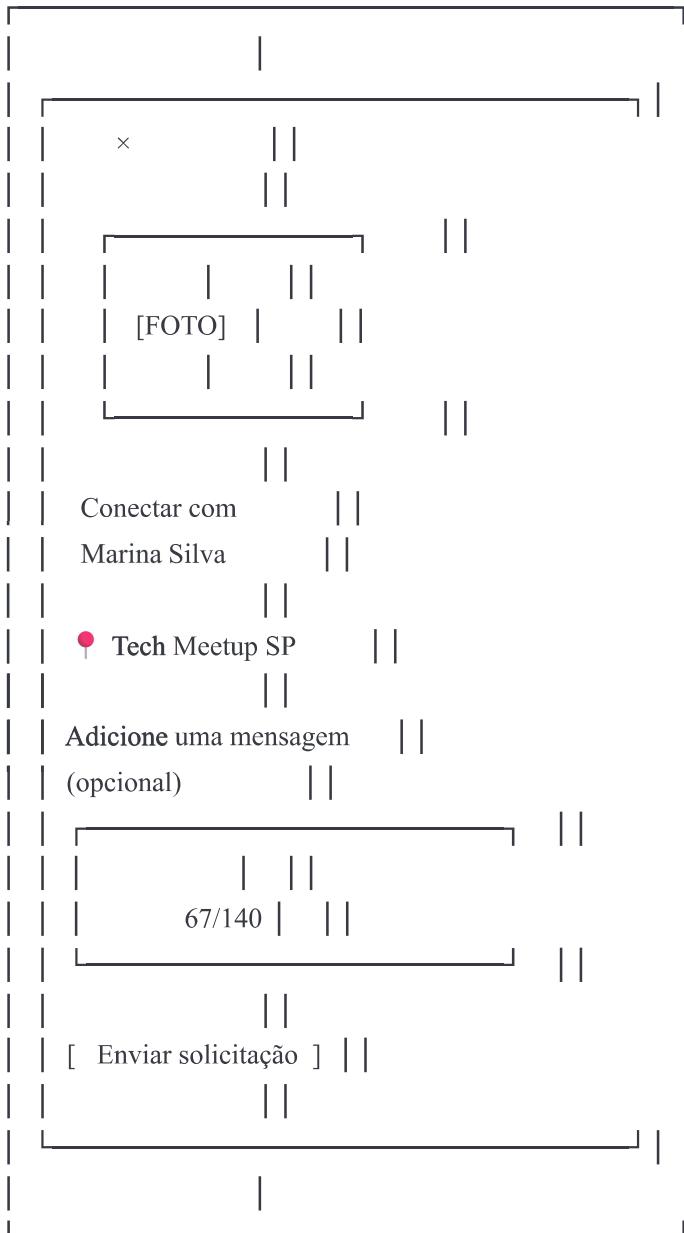
Elementos:

- Agrupado por evento
- Eventos ordenados por data (mais recente primeiro)
- Só mostra eventos dos últimos 7 dias
- Avatares são clicáveis (vai para perfil)
- Card do evento é clicável (expande lista)

Tap no evento:

- Expande para mostrar todas as pessoas
- Cada pessoa tem botão "Conectar"

T-042: Enviar ReMatch (Modal)



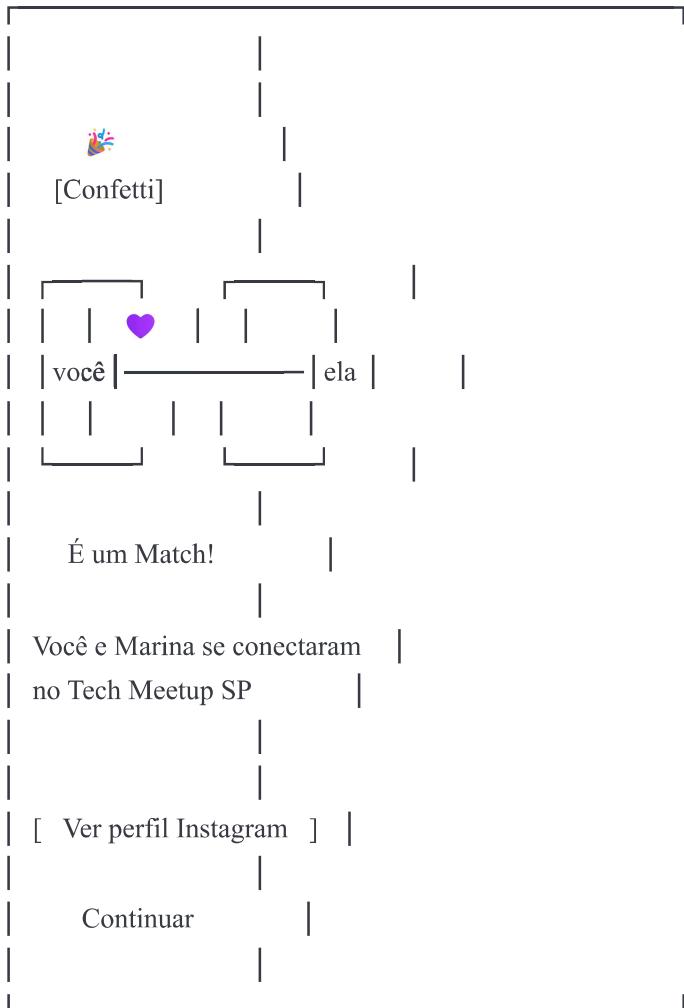
Elementos:

- Modal bottom sheet
- Foto da pessoa: 80px
- Evento em comum: contexto
- Textarea: mensagem opcional
- Contador de caracteres
- Botão enviar: primary

Após enviar:

- Fechar modal
- Toast: "Solicitação enviada!"
- Pessoa sai da lista "para conectar"
- Aparece em "aguardando resposta"

T-043: Match Confirmado (Celebration)



Elementos:

- Animação de confetti
- Fotos aproximando
- Coração pulsando
- Nome da pessoa revelado
- Evento de contexto
- Botão Instagram: abre @
- Continuar: fecha modal

Duração:

- Animação: 1.5s
- Fica até usuário interagir

7.6 Telas de Perfil e Configurações

T-050: Meu Perfil



Elementos:

- Header: back + settings (engrenagem)
- Foto: 100px, editável
- Info: nome, @, bio
- Stats: cards com números
- Menu: navegação para sub-telas
- Logout: botão ghost

x Editar Perfil Salvar

[FOTO] 

Nome
João Silva

Bio
Dev Backend, coffee addict  32/140

Cidade
São Paulo, SP

Redes Sociais

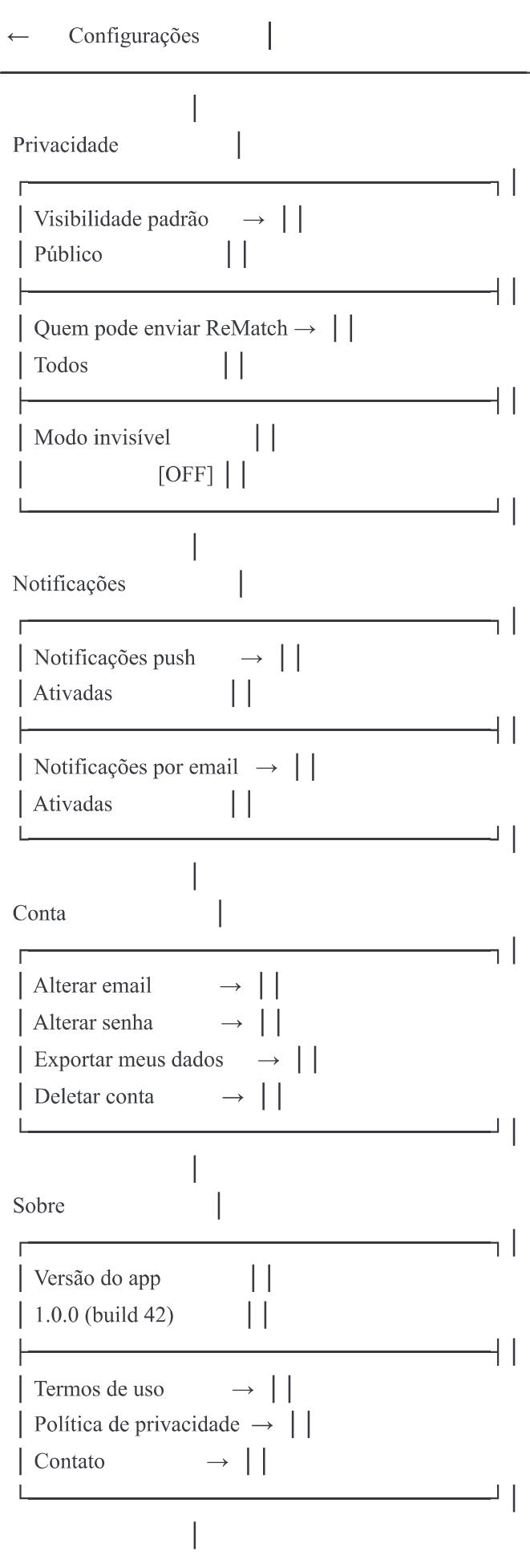
 Instagram
@joaosilva [Alterar]

 LinkedIn
Não conectado [Conectar]

Elementos:

- Header: cancelar (x) + salvar
- Campos editáveis
- Validação em tempo real
- Redes sociais: status de conexão
- Salvar: só habilitado se mudou algo

T-052: Configurações



Elementos:

- Seções agrupadas
- Itens com valor atual mostrado
- Switches para toggles
- Arrows para sub-telas
- Deletar conta: confirmation flow

8. Especificação de APIs

8.1 Visão Geral

Base URL: <https://api.checkpoint.app/v1>
Autenticação: Bearer JWT
Content-Type: application/json
Rate Limit: 100 requests/minuto por usuário

8.2 Endpoints de Autenticação

POST /auth/instagram

```
typescript

// Request
{
  code: string;      // Código do OAuth Instagram
  redirect_uri: string; // URI de redirecionamento
}

// Response 200
{
  user: User;
  token: string;      // JWT
  is_new_user: boolean;
}

// Response 401
{
  error: "invalid_code";
  message: "Código de autorização inválido ou expirado";
}
```

POST /auth/email/signup

```
typescript
```

```
// Request
{
  email: string;
  password: string;
  name: string;
  photo_url?: string;
}

// Response 201
{
  user: User;
  token: string;
}

// Response 400
{
  error: "email_taken" | "invalid_email" | "weak_password";
  message: string;
}
```

POST /auth/email/login

typescript

```
// Request
{
  email: string;
  password: string;
}

// Response 200
{
  user: User;
  token: string;
}

// Response 401
{
  error: "invalid_credentials";
  message: "Email ou senha incorretos";
}
```

POST /auth/logout

typescript

```
// Request: apenas header Authorization
```

```
// Response 204: No Content
```

POST /auth/refresh

```
typescript
```

```
// Request
```

```
{  
    refresh_token: string;  
}
```

```
// Response 200
```

```
{  
    token: string;  
    refresh_token: string;  
}
```

8.3 Endpoints de Usuário

GET /users/me

```
typescript
```

```
// Response 200
{
  id: string;
  name: string;
  email: string;
  photo_url: string;
  bio?: string;
  city?: string;
  instagram_username?: string;
  instagram_verified: boolean;
  linkedin_url?: string;
  stats: {
    total_checkins: number;
    total_events: number;
    total_connections: number;
  };
  settings: UserSettings;
  created_at: string;
}
```

PATCH /users/me

typescript

```
// Request (todos opcionais)
```

```
{
  name?: string;
  photo_url?: string;
  bio?: string;
  city?: string;
}
```

```
// Response 200
```

```
{
  ...User
}
```

```
// Response 400
```

```
{
  error: "validation_error";
  fields: {
    name?: "Nome muito curto";
    bio?: "Bio muito longa";
  };
}
```

GET /users/:id

typescript

```
// Response 200
{
  id: string;
  name: string;
  photo_url: string;
  bio?: string;
  instagram_username?: string; // só se conectados
  mutual_connections: number;
  mutual_events: Event[]; // resumo
  connection_status: "none" | "pending_sent" | "pending_received" | "connected";
}

// Response 404
{
  error: "user_not_found";
}
```

DELETE /users/me

typescript

```
// Request
{
  password: string; // confirmação
  reason?: string; // opcional, para feedback
}
```

// Response 204: conta deletada

```
// Response 401
{
  error: "invalid_password";
}
```

8.4 Endpoints de Check-in

GET /checkins/nearby

typescript

```
// Query params
?latitude=number
&longitude=number
&radius_km=number    // default 10
```

```
// Response 200
```

```
{
  events: Event[];
  places: Place[];
}
```

POST /checkins

typescript

```
// Request
{
  event_id?: string;
  place_id?: string;
  visibility: "public" | "friends" | "private";
  status_text?: string;
  location?: {
    latitude: number;
    longitude: number;
    accuracy: number;
  };
  qr_token?: string;    // see check-in via QR
}
```

```
// Response 201
```

```
{
  checkin: Checkin;
  event: Event;
  attendees_count: number;
}
```

```
// Response 400
```

```
{
  error: "already_checked_in" | "event_ended" | "too_far" | "invalid_qr";
  message: string;
}
```

DELETE /checkins/:id (checkout)

typescript

```
// Response 200
{
  checkin: Checkin;      // com ended_at preenchido
  duration_minutes: number;
}
```

GET /checkins/active

typescript

```
// Response 200
{
  checkin?: Checkin;
  event?: Event;
  attendees_count?: number;
}

// Response 200 (sem check-in ativo)
{
  checkin: null;
}
```

GET /checkins/history

typescript

```
// Query params
?page=number
&per_page=number    // default 20

// Response 200
{
  checkins: Checkin[];
  pagination: {
    page: number;
    per_page: number;
    total: number;
    has_more: boolean;
  };
}
```

8.5 Endpoints de Eventos

GET /events/:id

typescript

```
// Response 200
{
  event: Event;
  attendees_count: number;
  my_checkin?: Checkin;
  sample_attendees: User[]; // até 5, com conexões primeiro
}
```

GET /events/:id/attendees

typescript

```
// Query params
?filter=all|friends|fof //friends of friends
&page=number
&per_page=number
```

```
// Response 200
{
  attendees: [
    user: User;
    checkin_at: string;
    mutual_connections: number;
  ];
  pagination: Pagination;
}
```

GET /events/search

typescript

```
// Query params
?q=string           // busca por nome
&category=string
&date_from=string
&date_to=string
&latitude=number
&longitude=number
&radius_km=number

// Response 200
{
  events: Event[];
  pagination: Pagination;
}
```

8.6 Endpoints de ReMatch

GET /rematches/requests

```
typescript

// Response 200
{
  pending: RematchRequest[];    // recebidos, aguardando resposta
  sent: RematchRequest[];       // enviados, aguardando resposta
}
```

GET /rematches/available

```
typescript

// Response 200
{
  events: {
    event: Event;
    users: User[];      // pessoas para conectar
  }[];
}
```

POST /rematches

```
typescript
```

```

// Request
{
  receiver_id: string;
  event_id: string;
  message?: string;
}

// Response 201
{
  request: RematchRequest;
}

// Response 400
{
  error: "already_sent" | "already_connected" | "no_common_event" | "limit_reached" | "user_blocked";
  message: string;
}

```

PATCH /rematches/:id

```

typescript

// Request
{
  action: "accept" | "ignore" | "block";
}

// Response 200 (accept)
{
  request: RematchRequest;
  connection: Connection;
  user: User;           // dados completos do match
}

// Response 200 (ignore/block)
{
  request: RematchRequest;
}

```

GET /connections

```

typescript

```

```
// Response 200
{
  connections: {
    user: User;
    event: Event;          // onde se conheceram
    connected_at: string;
  }[];
}
```

8.7 Endpoints de Organizador

POST /organizer/events

typescript

```
// Request
{
  name: string;
  description?: string;
  image_url?: string;
  starts_at: string;
  ends_at?: string;
  timezone: string;
  location_type: "physical" | "online" | "hybrid";
  address?: string;
  latitude?: number;
  longitude?: number;
  online_url?: string;
  checkin_type: "gps" | "qr" | "both";
  checkin_radius_meters?: number;
  checkin_early_minutes?: number;
  visibility: "public" | "private" | "unlisted";
  show_attendee_list: boolean;
  allow_rematch: boolean;
  category?: string;
  capacity?: number;
  external_url?: string;
}
```

```
// Response 201
```

```
{
  event: Event;
  qr_code_url: string;
}
```

GET /organizer/events

```
typescript
```

```
// Response 200
{
  events: Event[];
}
```

GET /organizer/events/:id/metrics

```
typescript
```

```
// Response 200
{
  metrics: EventMetrics;
  attendees: {
    user: User;
    checkin_at: string;
    checkout_at?: string;
  }[];
}
```

GET /organizer/events/:id/export

typescript

```
// Response 200 (CSV)
Content-Type: text/csv
Content-Disposition: attachment; filename="event-attendees.csv"

name,email,instagram,checkin_at,checkout_at
"João Silva","joao@email.com","@joaosilva","2026-01-15T19:30:00Z","2026-01-15T22:15:00Z"
...
```

8.8 Endpoints de Configurações

GET /settings

typescript

```
// Response 200
{
  settings: UserSettings;
}
```

PATCH /settings

typescript

```
// Request (todos opcionais)
{
  default_checkin_visibility?: "public" | "friends" | "private";
  rematch_from?: "everyone" | "friends_of_friends" | "nobody";
  invisible_mode?: boolean;
  push_enabled?: boolean;
  push_friend_checkin?: boolean;
  push_rematch_request?: boolean;
  push_rematch_accepted?: boolean;
  push_event_reminder?: boolean;
  email_enabled?: boolean;
  email_weekly_digest?: boolean;
  theme?: "light" | "dark" | "system";
  language?: "pt-BR" | "en";
}
```

```
// Response 200
```

```
{
  settings: UserSettings;
}
```

8.9 Endpoints de Notificações

GET /notifications

typescript

```
// Query params
?page=number
&unread_only=boolean

// Response 200
{
  notifications: {
    id: string;
    type: "rematch_request" | "rematch_accepted" | "friend_checkin" | "event_reminder";
    title: string;
    body: string;
    data: object;          // dados específicos do tipo
    read: boolean;
    created_at: string;
  }[];
  unread_count: number;
  pagination: Pagination;
}
```

POST /notifications/read

```
typescript

// Request
{
  notification_ids: string[]; // ou ["all"] para marcar todas
}

// Response 200
{
  read_count: number;
}
```

POST /notifications/register-device

```
typescript
```

```

// Request
{
  token: string;           // FCM ou APNs token
  platform: "ios" | "android";
}

// Response 200
{
  device_id: string;
}

```

9. Regras de Negócio

9.1 Regras de Check-in

ID	Regra	Descrição
RN-001	Check-in único	Usuário só pode ter 1 check-in ativo por vez
RN-002	Validação GPS	Check-in só permitido dentro do raio configurado
RN-003	Accuracy mínimo	Se accuracy GPS > 100m, exigir QR code
RN-004	Expiração evento	Check-in expira 1h após término do evento
RN-005	Expiração padrão	Se evento sem término, expira em 8h
RN-006	Check-in antecipado	Permitido até X minutos antes (config evento)
RN-007	Check-in pós-evento	Não permitido após término + 1h
RN-008	QR code único	Cada evento tem token único para QR
RN-009	QR reutilizável	Mesmo QR funciona múltiplas vezes
RN-010	Novo check-in	Fazer novo check-in encerra o anterior

9.2 Regras de Discovery

ID	Regra	Descrição
RN-011	Visibilidade	Respeitar config de visibilidade do check-in
RN-012	Ordenação	Amigos > amigos de amigos > outros

ID	Regra	Descrição
RN-013	Conexões	Mostrar nº de conexões em comum
RN-014	Privado	Check-ins privados não aparecem para outros
RN-015	Bloqueio	Usuário bloqueado não aparece na lista
RN-016	Modo invisível	Se ativo, usuário não aparece (só vê)

9.3 Regras de ReMatch

ID	Regra	Descrição
RN-017	Elegibilidade	Ambos devem ter check-in no mesmo evento
RN-018	Janela tempo	Só até 7 dias após o evento
RN-019	Limite diário	Máximo 10 requests por dia
RN-020	Reset limite	Limite reseta à meia-noite (TZ usuário)
RN-021	Não duplicar	Não pode enviar se já enviou e pendente
RN-022	Reenvio	Pode reenviar se anterior foi ignorado há +7 dias
RN-023	Expiração	Requests pendentes expiram em 14 dias
RN-024	Bloqueio	Bloqueado não pode enviar requests
RN-025	Notificação	Não revelar nome na notificação
RN-026	Match	Aceitar cria conexão bidirecional

9.4 Regras de Organizador

ID	Regra	Descrição
RN-027	Criar evento	Qualquer usuário pode criar eventos
RN-028	Editar evento	Só organizador pode editar
RN-029	Limite grátis	Tier free: até 3 eventos ativos
RN-030	QR download	Organizador pode baixar QR em alta res
RN-031	Exportar	Organizador pode exportar lista CSV

ID	Regra	Descrição
RN-032	Cancelar	Cancelar evento notifica participantes

9.5 Regras de Privacidade (LGPD)

ID	Regra	Descrição
RN-033	Exportar dados	Usuário pode exportar todos seus dados
RN-034	Deletar conta	Deletar remove todos os dados em 30 dias
RN-035	Consentimento	Consentimento explícito para localização
RN-036	Retenção	Dados de check-in mantidos por 1 ano
RN-037	Anonimização	Após deleção, dados agregados anonimizados

10. Estados e Erros

10.1 Estados de Loading

Tela	Estado	Comportamento
Home	Carregando eventos	Skeleton de 3 cards
Lista pessoas	Carregando	Skeleton de 5 linhas
Perfil	Carregando	Skeleton completo
Check-in	Validando GPS	Spinner + "Validando localização..."
ReMatch	Enviando	Botão disabled + spinner

10.2 Estados Empty

Tela	Estado	Mensagem	Ação
Home	Sem eventos próximos	"Nenhum evento perto de você"	"Criar evento"
Discovery	Sem pessoas	"Seja o primeiro!"	"Compartilhar evento"
ReMatch	Sem solicitações	"Nenhuma solicitação"	-

Tela	Estado	Mensagem	Ação
ReMatch	Sem eventos recentes	"Faça check-in para conectar"	"Ver eventos"
Conexões	Sem conexões	"Suas conexões aparecerão aqui"	-
Histórico	Sem check-ins	"Seus check-ins aparecerão aqui"	"Ver eventos"

10.3 Códigos de Erro

Código	HTTP	Descrição	Ação Usuário
auth_required	401	Token inválido/expirado	Redirecionar para login
forbidden	403	Sem permissão	Mostrar mensagem
not_found	404	Recurso não existe	Voltar
validation_error	400	Dados inválidos	Mostrar campos
rate_limited	429	Muitas requests	Aguardar
server_error	500	Erro interno	Tentar novamente
maintenance	503	Em manutenção	Mostrar tela

10.4 Erros Específicos

Contexto	Erro	Mensagem
Check-in	too_far	"Você parece estar longe do evento. Verifique sua localização ou use o QR code."
Check-in	event_ended	"Este evento já terminou."
Check-in	already_checked_in	"Você já está neste evento."
Check-in	gps_disabled	"Ative a localização para fazer check-in."
Check-in	invalid_qr	"QR code inválido ou expirado."
ReMatch	limit_reached	"Você atingiu o limite de 10 solicitações hoje. Tente novamente amanhã."
ReMatch	already_sent	"Você já enviou uma solicitação para esta pessoa."
ReMatch	no_common_event	"Vocês precisam ter estado no mesmo evento."
ReMatch	user_blocked	"Não foi possível enviar a solicitação."

Contexto	Erro	Mensagem
Cadastro	email_taken	"Este email já está cadastrado."
Cadastro	weak_password	"Senha deve ter pelo menos 8 caracteres, 1 maiúscula e 1 número."

10.5 Estados Offline

Comportamento offline:

- Mostrar banner no topo
- Permitir ver dados em cache
- Desabilitar ações que precisam de rede
- Queue de ações para quando voltar online

11. Notificações

11.1 Push Notifications

Tipo	Título	Corpo	Ação
rematch_request	"Nova solicitação 💕"	"Alguém do [Evento] quer conectar"	Abrir requests
rematch_accepted	"É um Match! 🎉"	"Você e [Nome] estão conectados"	Abrir perfil
friend_checkin	"[Nome] está em um evento"	"[Nome] fez check-in em [Evento]"	Abrir evento
event_reminder	"Evento começando!"	"[Evento] começa em 30 minutos"	Abrir evento
event_update	"Atualização de evento"	"[Evento] foi atualizado"	Abrir evento

11.2 In-App Notifications

typescript

```
interface InAppNotification {  
  id: string;  
  type: NotificationType;  
  title: string;  
  body: string;  
  image_url?: string;    // foto do usuário/evento  
  action_url: string;   // deep link  
  read: boolean;  
  created_at: string;  
}
```

11.3 Email Notifications

Tipo	Assunto	Trigger
Boas-vindas	"Bem-vindo ao Check Point! 🎉"	Cadastro
Match	"Você tem um novo Match! 🎾"	ReMatch aceito
Digest semanal	"Sua semana no Check Point"	Domingo 10h
Inatividade	"Sentimos sua falta!"	14 dias sem uso
Recuperar senha	"Redefinir sua senha"	Solicitação

12. Interações

12.1 Instagram

Funcionalidade	API	Escopo
Login OAuth	Instagram Basic Display	<code>user_profile</code>
Importar perfil	GET /me	foto, nome, username
Importar seguindo	GET /me/following	lista de @ (futuro)
Share Stories	Instagram Sharing	compartilhar check-in

12.2 Google Maps

Funcionalidade	API
Mapa no app	Maps SDK (iOS/Android)
Geocoding	Geocoding API
Places autocomplete	Places API
Distância	Distance Matrix API

12.3 Firebase

Serviço	Uso
Authentication	Backup para login email
Cloud Messaging	Push notifications
Analytics	Eventos e funis
Crashlytics	Crash reporting
Remote Config	Feature flags

12.4 Outras

Serviço	Uso
SendGrid	Emails transacionais
Mixpanel	Analytics avançado
Sentry	Error tracking
AWS S3	Storage de imagens
CloudFlare	CDN e proteção

13. Segurança e Privacidade

13.1 Autenticação

- JWT com expiração de 30 dias
- Refresh token com expiração de 90 dias
- Revogação de tokens no logout
- Senha com hash bcrypt (cost 12)
- Rate limit de login: 5 tentativas / 15 min

13.2 Autorização

Recurso	Quem pode acessar
Próprio perfil	Usuário
Perfil de outros	Usuários logados
Instagram de match	Apenas matches
Lista de evento	Conforme config evento
Dashboard evento	Apenas organizador
Métricas evento	Apenas organizador

13.3 Proteção de Dados

Dado	Classificação	Proteção
Senha	Secreto	Hash + salt
Localização	Sensível	Não armazenar histórico
Email	PII	Criptografado
Tokens	Secreto	Secure storage
Fotos	PII	CDN com signed URLs

13.4 Rate Limiting

Endpoint	Limite
Geral	100/min por usuário
Login	5/15min por IP

Endpoint	Limite
Cadastro	3/hora por IP
ReMatch	10/dia por usuário
Upload	10/hora por usuário

13.5 LGPD Compliance

Requisito	Implementação
Consentimento	Opt-in explícito
Acesso	Endpoint /users/me/export
Retificação	Endpoint PATCH /users/me
Deleção	Endpoint DELETE /users/me
Portabilidade	Export em JSON

14. Analytics e Tracking

14.1 Eventos Core

```
javascript
```

```

// Onboarding
analytics.track('signup_started', { method: 'instagram' | 'email' });
analytics.track('signup_completed', { method, has_photo, source });
analytics.track('profile_completed', { fields_filled: [...] });

// Check-in
analytics.track('checkin_started', { source: 'nearby' | 'search' | 'qr' });
analytics.track('checkin_completed', { event_id, visibility, has_status });
analytics.track('checkout_completed', { event_id, duration_minutes });

// Discovery
analytics.track('event_viewed', { event_id, source });
analytics.track('attendee_list_viewed', { event_id, count });
analytics.track('profile_viewed', { user_id, context: 'event' | 'match' | 'connection' });

// ReMatch
analytics.track('rematch_list_viewed', { pending_count, available_count });
analytics.track('rematch_sent', { event_id, has_message });
analytics.track('rematch_responded', { response: 'accept' | 'ignore' | 'block' });
analytics.track('instagram_clicked', { context: 'profile' | 'match' });

// Engagement
analytics.track('app_opened', { days_since_last });
analytics.track('notification_opened', { type });
analytics.track('share_completed', { type: 'event' | 'checkin' | 'profile' });

```

14.2 User Properties

```

javascript

analytics.setUserProperties({
  total_checkins: number,
  total_events: number,
  total_connections: number,
  account_age_days: number,
  last_checkin_days_ago: number,
  has_instagram: boolean,
  default_visibility: string,
  platform: 'ios' | 'android',
});

```

14.3 Funis

Funil	Etapas
Onboarding	Download → Signup Started → Signup Completed → Profile Completed → First Checkin
Check-in	Event Viewed → Checkin Started → Checkin Completed
ReMatch	Profile Viewed → Rematch Sent → Rematch Accepted
Conversion	Signup → D1 Return → D7 Return → D30 Return

15. Cronograma e Fases

15.1 Visão Geral (16 semanas)

Semana 1-2: Setup & Infraestrutura

Semana 3-4: Auth & Perfil

Semana 5-7: Check-in Core

Semana 8-9: Discovery

Semana 10-12: ReMatch

Semana 13-14: Dashboard Organizador

Semana 15-16: Polish & Launch

15.2 Detalhamento por Fase

Fase 0: Setup (Semana 1-2)

Task	Responsável	Entrega
Setup repositório	Backend	Repo Git configurado
CI/CD básico	DevOps	Pipeline funcionando
Ambiente dev	DevOps	AWS/GCP provisionado
Database schema	Backend	Migrations criadas
API boilerplate	Backend	Hello World
App boilerplate	Mobile	Projeto RN/Flutter
Design system	Design	Componentes Figma

Fase 1: Auth & Perfil (Semana 3-4)

Task	Responsável	Entrega
OAuth Instagram	Backend	Endpoint funcionando
Login email	Backend	Endpoints funcionando
Telas de login	Mobile	UI implementada
Telas de cadastro	Mobile	UI implementada
Perfil CRUD	Backend	Endpoints funcionando
Telas de perfil	Mobile	UI implementada

Fase 2: Check-in (Semana 5-7)

Task	Responsável	Entrega
Eventos CRUD	Backend	Endpoints funcionando
Validação GPS	Backend	Lógica implementada
Check-in/out	Backend	Endpoints funcionando
QR code	Backend	Geração e validação
Mapa eventos	Mobile	Integração Maps
UI check-in	Mobile	Fluxo completo
Tela evento	Mobile	UI implementada

Fase 3: Discovery (Semana 8-9)

Task	Responsável	Entrega
Lista attendees	Backend	Endpoint com ordenação
Conexões em comum	Backend	Cálculo implementado
Lista pessoas	Mobile	UI com filtros
Perfil outro user	Mobile	UI implementada
Busca eventos	Backend	Endpoint com filtros
UI busca	Mobile	Implementada

Fase 4: ReMatch (Semana 10-12)

Task	Responsável	Entrega
Requests CRUD	Backend	Endpoints funcionando
Connections	Backend	Lógica de match
Regras negócio	Backend	Limites, expiração
Tab ReMatch	Mobile	UI completa
Enviar request	Mobile	Modal implementado
Celebração match	Mobile	Animação
Deep link Instagram	Mobile	Funcionando

Fase 5: Dashboard (Semana 13-14)

Task	Responsável	Entrega
Auth web	Backend	Login organizador
Criar evento web	Frontend	Form completo
Métricas	Backend	Agregações
Dashboard UI	Frontend	Gráficos e lista
Export CSV	Backend	Endpoint funcionando
QR download	Frontend	Alta resolução

Fase 6: Polish (Semana 15-16)

Task	Responsável	Entrega
Bug fixes	Todos	Issues resolvidas
Performance	Todos	Targets atingidos
Empty states	Mobile	Todos implementados
Onboarding flow	Mobile	Refinado
Emails	Backend	Templates prontos
Landing page	Frontend	Publicada
App stores	Mobile	TestFlight/Play Console

Apêndices

A. Glossário

Termo	Definição
Check-in	Ação de marcar presença em um evento ou local
Check-out	Ação de encerrar um check-in
Discovery	Funcionalidade de ver pessoas no mesmo local
ReMatch	Funcionalidade de reconectar com pessoas de eventos passados
Request	Solicitação de conexão enviada via ReMatch
Match/Coneção	Quando duas pessoas aceitam se conectar
Organizador	Usuário que cria e gerencia eventos
Visibilidade	Configuração de quem pode ver um check-in

B. Referências

- Figma: [link para design]

- API Docs: [link para Swagger]
 - Repositório: [link para GitHub]
 - Trello/Jira: [link para board]
-

PRD v1.0 - Check Point Última atualização: Janeiro 2026