

Documentação Técnica — ONCOImmune

Protótipo Interativo Web e Diretrizes para App Nativo

1. Visão Geral e Escopo do Documento

Este documento descreve a arquitetura, as tecnologias e as funcionalidades do protótipo interativo ONCOImmune, construído para validar a experiência do usuário, o fluxo de navegação e o conceito visual da solução. Ele também estabelece as diretrizes técnicas para a evolução desse protótipo em um aplicativo nativo publicado na Apple App Store e no Google Play Store.

O protótipo atual é uma aplicação web responsiva, desenvolvida com React e TanStack Start, que simula telas, dados, notificações e ações do usuário. A versão final, destinada às lojas de aplicativos, deverá ser reescrita em uma tecnologia nativa ou cross-platform — React Native, Flutter ou outra stack equivalente — mantendo o design system, a identidade visual e os fluxos validados neste protótipo.

- Protótipo atual: aplicação web interativa (não é PWA nem app nativo).
- Objetivo do protótipo: validar jornada do paciente, usabilidade e requisitos com stakeholders e usuários antes da construção do app nativo.
- Produto final: aplicativo nativo para iOS e Android, publicado nas lojas oficiais.

2. Tecnologias do Protótipo Atual (Web)

2.1. Frontend

- React 19 — biblioteca principal para construção da interface.
- TypeScript — tipagem estática para maior segurança e manutenibilidade do código.
- TanStack Start — framework full-stack para React com SSR/SSG e server functions, utilizado para roteamento, carregamento de dados e execução de funções no servidor.
- TanStack Router — roteamento tipado baseado em arquivos, com loaders e layouts aninhados.
- TanStack Query — gerenciamento de estado assíncrono e cache de dados do servidor.
- Tailwind CSS v4 — estilização utilitária via CSS nativo com tema customizado.
- Radix UI + shadcn/ui — componentes acessíveis e primitivos de interface (dialogs, dropdowns, tabs, etc.).
- Lucide React — biblioteca de ícones.
- Sora — família tipográfica principal (títulos e corpo de texto).

2.2. Backend e Infraestrutura

- Supabase (Cloud) — backend as a service, responsável por autenticação, banco de dados PostgreSQL, storage e Row Level Security (RLS).
- PostgreSQL — banco de dados relacional para persistência de usuários, registros de sintomas, conteúdos salvos, perfil e cartão de identificação.
- Supabase Auth — autenticação por e-mail/senha e OAuth (Google/Apple), com JWT e sessões.
- Server Functions do TanStack Start — funções RPC executadas no edge (Cloudflare Workers), usadas para lógica de servidor segura.
- Cloudflare Workers — runtime serverless para SSR e server functions.

2.3. Persistência no Protótipo

Por se tratar de um protótipo sem banco de dados integrado em alguns fluxos, parte dos dados é armazenada localmente no navegador do usuário, usando localStorage e/ou IndexedDB. Isso permite simular funcionalidades offline, histórico de sintomas e conteúdos salvos sem depender de uma API completa.

- localStorage/IndexedDB — simulação de persistência local no protótipo.
- Supabase — será a fonte de verdade no app de produção, substituindo o armazenamento local.

3. Tecnologias Recomendadas para o App Nativo

O protótipo web valida a experiência, mas não pode ser publicado diretamente nas lojas como app nativo. Abaixo estão as opções recomendadas para a versão final, considerando reutilização de conhecimento, manutenção e performance.

3.1. Opção A — React Native (recomendada se o time já domina React)

- React Native — framework para compilar código React em apps nativos iOS/Android.
- Expo — plataforma que simplifica build, OTA updates, push notifications e acesso a APIs nativas.
- TypeScript — mantém a tipagem existente do protótipo.
- React Navigation — roteamento e navegação entre telas.
- TanStack Query — pode ser reaproveitado para cache de dados.
- Tailwind CSS via NativeWind — permite reaproveitar parte das classes utilitárias, com adaptações.
- Supabase JS Client — comunicação direta com o backend já definido.

3.2. Opção B — Flutter

- Flutter + Dart — framework da Google com renderização própria e alta performance.
- Riverpod/BLoC — gerenciamento de estado.
- Supabase Flutter SDK — integração com autenticação e banco de dados.
- Material 3 / Cupertino — componentes nativos de cada plataforma.

3.3. Backend mantido em ambas as opções

- Supabase (PostgreSQL + Auth + Storage) — backend permanece o mesmo; apenas o cliente muda (React Native ou Flutter).
- Server Functions/Edge Functions — podem continuar sendo usadas para lógicas sensíveis, webhooks e integrações.
- Push notifications — Firebase Cloud Messaging (Android) e APNs (iOS), via Supabase Edge Functions ou serviço dedicado.

4. Funcionalidades do Protótipo e do Produto Final

4.1. Onboarding e Autenticação

- Tela de login com e-mail/senha, recuperação de senha e criação de conta.
- Login social com Google (e Apple, no app nativo).
- Onboarding inicial explicando o propósito do app (orientar, monitorar e conectar o paciente oncológico).

4.2. Home / Calendário

- Calendário mensal com destaque para o dia atual e dias com registros.
- Indicadores visuais de registros por severidade (cores verde, amarelo e vermelho).
- Cards de próximos eventos, lembretes e conteúdos relevantes.
- Acesso rápido a registrar sintomas, conteúdos educativos e vídeos.

4.3. Registro de Sintomas

- Seleção de sintomas como coceira, erupção cutânea, febre, náusea, fadiga, diarreia, entre outros.
- Descrições educativas exibidas após a seleção de cada sintoma.
- Níveis de intensidade representados apenas por cores (verde, amarelo, vermelho), sem rótulos textuais de leve/moderado/grave.
- Armazenamento do histórico de registros por dia.
- Geração de simulação de PDF com imagem estática (no protótipo); no app nativo, geração real de PDF a partir dos dados do banco.

4.4. Cartão de Identificação

- Formulário com dados do paciente, diagnóstico, Imunoterápico(s) em uso, comorbidades e contatos de emergência.
- Visualização do cartão com QR Code (simulado no protótipo).
- Pré-visualização de como o PDF do cartão será gerado no futuro, usando imagens estáticas.
- No app nativo: geração real de PDF e possibilidade de compartilhamento nativo.

4.5. Conteúdos Educacionais

- Lista de conteúdos organizados por temas (introdução, efeitos colaterais, Tratamento do câncer, etc.).
- Tela de leitura com ações secundárias: "Ouvir conteúdo" (text-to-speech) e "Salvar para ler depois".
- Modo offline simulado via localStorage; no app nativo, sincronização com Supabase e cache local.

4.6. Vitrine de Vídeos

- Galeria de vídeos educativos selecionados.
- Player embutido ou redirecionamento para vídeo.
- No app nativo: player nativo, download para offline e notificações de novos conteúdos.

4.7. Chat e Suporte

- Chat com IA para tirar dúvidas sobre imunoterapia e cuidados.
- Opção de falar com um humano (equipe de apoio).
- No app nativo: push notifications para novas mensagens.

4.8. Perfil

- Visualização e edição de dados pessoais.
- Acesso a configurações, termos de uso e política de privacidade.
- Logout seguro.

5. Roadmap de Transição: Protótipo → App Nativo

- Fase 1 — Validação do protótipo web: testes de usabilidade, ajustes de fluxo e consolidação do design system.
- Fase 2 — Definição da stack nativa (React Native/Flutter) e setup dos repositórios iOS/Android.
- Fase 3 — Migração das telas e componentes, reaproveitando identidade visual, paleta de cores (ONCOImmune) e tipografia.
- Fase 4 — Integração completa com Supabase: autenticação, banco de dados, storage e RLS.
- Fase 5 — Implementação de features nativas: push notifications, câmera, galeria, compartilhamento de PDF, biometria.
- Fase 6 — Testes em dispositivos reais, ajustes de performance e publicação nas lojas.

6. Considerações sobre Publicação nas Lojas

- Contas de desenvolvedor: Apple Developer Program (iOS) e Google Books/Google Play Console (Android).
- Guidelines: seguir as diretrizes de privacidade, acessibilidade e conteúdo médico de cada loja.
- LGPD/GDPR: consentimento claro, política de privacidade e direito ao esquecimento.
- Classificação indicativa e disclaimer: o app é um auxiliar ao tratamento, não substitui orientação médica.
- Revisão médica: conteúdos educacionais devem ser validados por profissionais de saúde antes da publicação.

7. Design System e Identidade Visual

- Marca: ONCOImmune.
- Tagline: "Orienta. Monitora. Conecta."
- Paleta principal: azul escuro (#0C457A), azul vibrante (#1579F0) e teal (#05A8A9).
- Tipografia: Sora (web); no app nativo, usar fonte equivalente ou a mesma via custom fonts.
- Tom de voz: acolhedor, claro, seguro e acessível ao paciente oncológico.

8. Resumo para Stakeholders

O que está sendo construído agora é um protótipo interativo web, com dados simulados e persistência local, cujo objetivo é validar a jornada do usuário e o conceito do ONCOImmune. Após aprovação, o projeto será reescrito como aplicativo nativo para iOS e Android, utilizando React Native ou Flutter, mantendo o Supabase como backend e publicado nas lojas oficiais. Nenhuma mudança no protótipo web é necessária neste momento; este documento serve para alinhar a arquitetura futura desde o início.

Fim do documento