

ONCOImmune

Painel Administrativo

Orienta · Monitora · Conecta

Documentação Técnica Detalhada

Stack, arquitetura e funcionalidades

Versão do documento: 1.0
Julho de 2026

1. Resumo Executivo

O ONCOImmune é um painel administrativo web desenvolvido para acompanhar pacientes em tratamento de imunoterapia oncológica. A aplicação oferece uma interface centralizada para profissionais de saúde visualizarem registros de sintomas, gerenciarem alertas, acompanharem pacientes, disponibilizarem conteúdos educativos e vídeos, além de comunicação via chat de suporte.

O projeto foi construído como um protótipo funcional de alta fidelidade, utilizando tecnologias modernas de frontend e preparado para integração com backend Supabase (Cloud) em ambiente de produção.

2. Stack Tecnológica

2.1. Frontend

- Framework: React 19 + TanStack Start v1 (full-stack React framework com SSR/SSG)
- Roteamento: TanStack Router (file-based routing)
- Gerenciamento de estado servidor: TanStack Query v5
- Build / bundler: Vite 8 com configuração @lovable.dev/vite-tanstack-config
- Estilização: Tailwind CSS v4 com tema customizado via CSS variables
- Componentes de UI: shadcn/ui (Radix UI primitives) + Tailwind
- Ícones: Lucide React
- Tipagem: TypeScript 5.8
- Validação de formulários: Zod + React Hook Form
- Notificações: Sonner (toast notifications)
- Gráficos: Recharts
- Exportação de dados: SheetJS (xlsx)
- Data/calendário: date-fns e react-day-picker
- Fonte tipográfica: Sora (Google Fonts)

2.2. Backend e Infraestrutura Recomendada

- Backend: Supabase (Cloud)
 - Banco de dados: PostgreSQL
 - Autenticação: Supabase Auth com JWT e RLS (Row Level Security)
 - API Data: PostgREST (REST automática sobre PostgreSQL)
 - Storage: Supabase Storage para imagens e arquivos
 - Edge Functions: Supabase Edge Functions (Cloudflare Workers) para lógica customizada
 - Realtime: Supabase Realtime para notificações e chat em tempo real
- Hospedagem: Lovable Cloud / Edge deploy (Cloudflare Workers)
- Estado atual do projeto: protótipo com dados mockados e localStorage para persistência local de estados de status, fixação e prontuários.

2.3. Ferramentas de Desenvolvimento

- Linting: ESLint 9 + typescript-eslint + eslint-plugin-react-hooks
- Formatação: Prettier 3
- Runtime local: Bun / Node
- Servidor de desenvolvimento: Vite dev server

3. Arquitetura da Aplicação

3.1. Modelo Arquitetural

A aplicação segue uma arquitetura SPA (Single Page Application) com Server-Side Rendering (SSR) via TanStack Start. O roteamento é baseado em arquivos (file-based routing), onde cada arquivo .tsx dentro de src/routes representa uma rota pública ou protegida.

3.2. Estrutura de Pastas

Pasta/Arquivo	Responsabilidade
---------------	------------------

src/routes	Definição de todas as rotas da aplicação
src/routes/__root.tsx	Layout raiz, providers globais (QueryClient), metadados e SEO
src/routes/_authenticated.tsx	Layout protegido: verifica autenticação e renderiza sidebar
src/components	Componentes reutilizáveis: header, sidebar, diálogos, sheets
src/components/ui	Componentes base do shadcn/ui (Button, Card, Dialog, etc.)
src/lib	Utilitários, mocks, autenticação e helpers
src/assets	Assets estáticos gerenciados pela Lovable Asset Pipeline
src/styles.css	Tema global, tokens de cores, gradientes e utilitários
src/server.ts	Entry point SSR customizado com tratamento de erros
src/start.ts	Configuração do TanStack Start e middleware de erro
vite.config.ts	Configuração do Vite com plugin TanStack Start

3.3. Fluxo de Autenticação

Atualmente o protótipo utiliza autenticação simulada baseada em localStorage. O token é armazenado na chave admin-token e as rotas sob _authenticated.tsx verificam a presença do token antes de renderizar o conteúdo protegido.

Para produção, recomenda-se substituir pela autenticação do Supabase (Auth) com JWT, refresh token e RLS, garantindo que cada usuário acesse apenas dados autorizados.

4. Design System e Identidade Visual

A identidade visual do ONCOImmune foi baseada na paleta "Confiança", projetando segurança, tecnologia e cuidado. A fonte escolhida foi a Sora, uma sans-serif moderna e legível.

4.1. Paleta de Cores

Token	Hex	Uso
Primary	#0c457a	Títulos, botões primários, ênfase de marca
Secondary	#1579f0	Links, hover, gradientes, áreas interativas
Accent	#05a8a9	Destaques, badges de sucesso, elementos de apoio
Foreground	#2a3344	Texto principal
Background	#ecf0f6	Fundo da aplicação e superfícies
Alert Red	#b42332 / #fde2e5	Alertas graves, status pendente, ações destrutivas
Alert Amber	#8a5a10 / #fff1d6	Alertas moderados, atenção

Alert Green	#0f7a5a / #d9f2ea	Alertas leves, status resolvido, sucesso
-------------	----------------------	--

4.2. Componentes Visuais

- Gradientes: primary (navy -> blue -> teal), coral, purple, amber
- Cards elevados com sombra suave, bordas arredondadas (rounded-2xl) e hover sutil
- Glassmorphism no header e painéis com backdrop-blur
- Mesh gradient background para profundidade visual
- Badges e tags com cores semânticas por severidade

5. Funcionalidades Detalhadas

5.1. Login e Autenticação

Tela de login com campos de email e senha. No protótipo, qualquer credencial redireciona para o dashboard após salvar um token fictício no localStorage. O layout utiliza card centralizado com branding ONCOImmune e gradiente de fundo.

5.2. Dashboard Administrativo

A tela inicial apresenta uma visão gerencial completa do acompanhamento dos pacientes:

- Hero band com saudação personalizada e mensagem da marca
- Funil de pacientes com 5 indicadores: cadastrados no app, registraram sintomas, alerta vermelho, alerta amarelo e alerta verde
- Filtro de período (1, 7, 15, 30, 60, 90 dias ou todo o período)
- Gráfico de pizza com distribuição de alertas por nível
- Lista de alertas por nível com scroll e cards de pacientes
- Gráficos demográficos: distribuição por sexo e faixa etária
- Imunoterápicos mais utilizados (gráfico de barras)
- Diagnósticos mais frequentes
- Média de registros por paciente e frequência de acesso
- Lista de cliques em conteúdos educativos e vídeos
- Botão Exportar Planilha com popup de seleção de período, formato (Excel/CSV) e filtros específicos: imunoterápico, grau, sexo, faixa etária e paciente

5.3. Registros de Sintomas

Listagem centralizada de todos os registros clínicos enviados pelos pacientes:

- Cards com data, paciente, severidade, sintomas e status
- Filtros: busca por paciente, nível de alerta, status (pendente/em andamento/resolvido), período e paciente
- Clique abre o sheet de detalhes do registro
- Exportação de planilha com data, nível, detalhe, sintoma, intensidade, condutas e anotações

5.4. Detalhes do Registro (Sheet)

Painel lateral que exhibe os detalhes completos de um registro:

- Cabeçalho com data e identificação visual
- Box informativo do alerta conforme severidade (verde/amarelo/vermelho)
- Dropdown de status do atendimento (pendente, em andamento, resolvido)
- Lista de sintomas com nível de intensidade
- Condutas sugeridas com ícones
- Anotações do paciente/responsável
- Rodapé com data do registro e data de resolução (simulada)
- Botão Gerar PDF (protótipo)

5.5. Alertas / Notificações

Tela dedicada ao gerenciamento de alertas clínicos:

- Lista de alertas com cor de borda por nível (verde/amarelo/vermelho)
- Container destacado em vermelho quando status = pendente
- Motivos dinâmicos: "O atendimento está pendente há mais de 24 horas" ou "O paciente não faz registro há 1 mês"
- Filtros por nível, status e período
- Dropdown de status no próprio alerta
- Ícone de olho para visualizar detalhes no sheet
- Integração com sininho de notificações do header

5.6. Pacientes

Gestão completa dos pacientes cadastrados:

- Grid de cards com foto, nome, idade, datas de cadastro e último registro
- Busca por nome
- Fixação de pacientes (pin) com persistência no localStorage
- Abas "Todos" e "Fixados"
- Clique no card abre a tela de detalhes do paciente
- Ordenação alfabética

5.7. Detalhes do Paciente

Tela individual do paciente dividida em 4 abas:

- Dados: informações cadastradas pelo paciente no app (dados pessoais, diagnóstico, imunoterapia, outros tratamentos, alergias e emergência)
- Registros: lista de registros do paciente + calendário interativo com marcação dos dias com registros por severidade
- Prontuário: histórico de anotações clínicas com possibilidade de cadastrar novas entradas
- Dashboard: KPIs individuais, gráfico de registros por data/tipo, últimos acessos, vídeos assistidos e conteúdos visualizados
- Exportação de dados pessoais e de registros do paciente via planilha

5.8. Profissionais

Cadastro e gestão de médicos e enfermeiras:

- Abas separadas por tipo: Médicos e Enfermeiras
- Tabela com nome, registro (CRM/COREN), especialidade, contato, pacientes vinculados e status
- Ações: editar e ativar/inativar

- Modal de cadastro com todos os campos

5.9. Conteúdos Educativos

CMS simples para gestão de artigos educativos:

- Grid de cards com thumbnail, título e texto completo
- Ações via dropdown: editar e excluir
- Sheet lateral para criar/editar conteúdo com título, thumb e texto completo
- Dados iniciais preenchidos com conteúdos sobre câncer e tratamento

5.10. Vídeos

Biblioteca de vídeos educativos:

- Grid de cards com thumbnail, ícone de play, título e descrição
- Modal de edição com player embed, título, descrição, URL e thumbnail
- Vídeos iniciais sobre imunoterapia, tratamento, cuidados e sistema imunológico

5.11. Chat de Suporte

Interface de chat para atendimento humano:

- Layout de duas colunas: lista de conversas e janela de mensagens
- Filtros por status: todas, abertas, em atendimento, resolvidas
- Indicador de online/offline
- Contador de mensagens não lidas
- Envio de mensagens com resposta automática simulada
- Anexo (ícone de clip)

5.12. Cadastro de Acessos

Gerenciamento de usuários do painel administrativo:

- Tabela com nome, email, perfil, status e último acesso
- Perfis: Administrador, Médico, Recepcionista e Suporte
- Modal de cadastro/edição com senha temporária
- Switch para ativar/inativar acesso
- Busca por nome ou email

5.13. Meu Perfil

Tela de configurações do usuário logado:

- Header com avatar, nome, email, especialidade e badge de administrador
- Aba Dados pessoais: nome, email, telefone, especialidade, bio
- Aba Senha: alteração de senha com confirmação
- Aba Preferências: emails de resumo, notificações push e alertas críticos

5.14. Compartilhar App

Página com links de download para Google Play Store e Apple App Store, além de botões para copiar os links de compartilhamento.

5.15. Componente de Exportação (ExportDialog)

Diálogo reutilizável para exportação de planilhas:

- Seleção de formato: Excel (.xlsx) ou CSV (.csv)
- Seleção de período via calendários (data inicial e final)
- Validação de período
- Suporte a filtros customizados por contexto (usado no Dashboard)
- No protótipo, exibe mensagem de sucesso simulando a exportação

6. Fluxos de Dados e Persistência

No estado atual do protótipo, os dados são majoritariamente estáticos ou persistidos no `localStorage` do navegador.

6.1. Dados Mockados (`src/lib/mock-*.ts`)

- `mock-pacientes.ts`: 10 pacientes padrão com dados clínicos e responsáveis
- `mock-registros.ts`: 19 registros de sintomas com severidade, sintomas, condutas e anotações
- `mock-profissionais.ts`: 10 profissionais (médicos e enfermeiras)
- `mock-acessos.ts`: 4 usuários do painel
- `mock-conteudos.ts`: 3 conteúdos educativos
- `mock-videos.ts`: 4 vídeos educativos
- `mock-chats.ts`: conversas de suporte
- `mock-prontuarios.ts`: prontuários clínicos com persistência local
- `mock-alertas.ts`: status de alertas e motivos
- `mock-registro-status.ts`: status de atendimento dos registros

6.2. Persistência Local (`localStorage`)

- `admin-token`: token de autenticação simulada
- `onco.pacientes.fixados`: lista de pacientes fixados
- `onco.alertas.status`: mapa de status dos alertas
- `onco.registros.status`: mapa de status dos registros
- `onco.prontuarios`: prontuários clínicos cadastrados

6.3. Integração Futura com Supabase

Para transformar o protótipo em produção, recomenda-se:

- Substituir os mocks por tabelas PostgreSQL no Supabase
- Criar tabelas: `pacientes`, `registros`, `profissionais`, `usuarios`, `conteudos`, `videos`, `prontuarios`, `alertas`, `acessos`, `mensagens_chat`
- Habilitar RLS em todas as tabelas com políticas por perfil de usuário
- Usar Supabase Auth para login real com email/senha ou OAuth
- Utilizar Supabase Realtime para chat e notificações em tempo real
- Migrar exportações para edge functions quando necessário
- Armazenar imagens e PDFs no Supabase Storage

7. Segurança e Boas Práticas

- Autenticação: migrar de `localStorage` para Supabase Auth com JWT seguro

- Autorização: implementar RLS no PostgreSQL para garantir que usuários acessem apenas dados permitidos
- Dados sensíveis: criptografar dados de saúde em repouso e em trânsito (SSL/TLS)
- LGPD/GDPR: garantir consentimento, anonimização e direito ao esquecimento
- Validação: usar Zod para validar todas as entradas de formulários
- Auditoria: registrar logs de acesso e alterações em registros médicos
- Roles: armazenar papéis em tabela separada (user_roles) e nunca no client-side

8. Deploy e Publicação

O projeto está configurado para deploy via Lovable Cloud, com build pelo Vite e runtime em edge (Cloudflare Workers). Os comandos disponíveis são:

- npm run dev: servidor de desenvolvimento
- npm run build: build de produção
- npm run build:dev: build em modo desenvolvimento
- npm run preview: preview do build
- npm run lint: verificação de código
- npm run format: formatação com Prettier

9. Próximos Passos Recomendados

- Ativar Lovable Cloud / Supabase no projeto
- Criar schema de banco de dados no PostgreSQL
- Migrar mocks para tabelas reais com seeds
- Implementar autenticação real e RLS
- Criar roles e permissões (admin, médico, enfermeira, recepcionista)
- Conectar chat de suporte ao Supabase Realtime
- Habilitar geração real de PDFs e planilhas
- Implementar notificações push e email
- Adicionar testes automatizados (unitários e E2E)
- Realizar auditoria de segurança e conformidade LGPD

10. Conclusão

O ONCOImmune é um protótipo visual e funcionalmente robusto, pronto para evoluir para uma aplicação de produção. Com base em React 19, TanStack Start e Tailwind CSS, o frontend oferece uma experiência moderna e responsiva. A integração futura com Supabase (Lovable Cloud) como backend permitirá persistência segura, autenticação, autorização granular e comunicação em tempo real.

Este documento serve como referência técnica completa para desenvolvedores, gestores de produto e stakeholders envolvidos no projeto.