

Saúde Conectada

1. Visão geral do material

Este documento reúne a estrutura do projeto, um resumo técnico do aplicativo e a transcrição organizada do código-fonte recebido. O objetivo é facilitar a leitura em contexto de evento, permitindo que outras pessoas entendam rapidamente a proposta do sistema e consultem os arquivos principais.

Nome do projeto	Arquivos organizados	Páginas/módulos	Formato principal
saude-na-palma	15	Splash, onboarding, home e módulos clínicos	React + Tailwind + rotas em JSX

2. Estrutura do projeto

A árvore abaixo resume a organização apresentada no material original.

```
saude-na-palma/  
├── index.html  
├── index.css  
├── tailwind.config.js  
├── App.jsx  
├── entities/  
│   └── Paciente.json  
├── pages/  
│   ├── Splash.jsx  
│   ├── Onboarding.jsx  
│   ├── Home.jsx  
│   ├── Triagem.jsx  
│   ├── PastaDigital.jsx  
│   ├── Agenda.jsx  
│   ├── FilaEspera.jsx  
│   ├── Receituario.jsx  
│   └── Perfil.jsx  
└── components/  
    └── SOSButton.jsx
```

3. Tecnologias

Tecnologia	Uso no aplicativo
React 18	Estrutura principal da interface
Tailwind CSS	Estilização utilitária e sistema visual
Framer Motion	Animações e transições
React Router DOM	Navegação entre páginas
Base44 SDK	Persistência de dados e integração de IA
TanStack Query	Gerenciamento de consultas e cache
Nunito (Google Fonts)	Tipografia do aplicativo

4. Funcionalidades destacadas

#	Módulo	Descrição
1	Tela inicial (Splash)	Abertura do app com identidade visual e redirecionamento para onboarding.
2	Onboarding em 4 missões	Cadastro guiado do paciente com dados pessoais, condições de saúde e contato de emergência.
3	Painel principal	Centraliza o acesso aos módulos do sistema.
4	SOS / Emergência	Confirmação de ajuda e compartilhamento de localização.
5	Triagem inteligente	Classifica sintomas e orienta o local de atendimento.
6	Pasta digital	Exames, laudos, receitas e vacinas em uma área única.
7	Agenda	Consultas, exames e lembretes de saúde.
8	Fila de espera	Acompanhamento de senha e previsão de atendimento.
9	Receituário	Prescrições e status de validade.

10	Perfil do paciente	Visão consolidada dos dados de saúde.
----	--------------------	---------------------------------------

5. Código-fonte organizado por arquivo

Cada arquivo foi separado em sua própria seção, com o respectivo caminho e conteúdo em bloco monoespaçado. Para melhorar a leitura e evitar páginas com muito espaço vazio, arquivos maiores foram divididos em blocos menores de continuação.

5.1 index.html

Caminho: index.html

```
<!doctype html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Saúde na Palma da Sua Mão</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

5.2 index.css

Caminho: index.css

```
@import url('https://fonts.googleapis.com/css2?family=Nunito:wght@400;500;600;700;800;900&display=swap');

@tailwind base;
@tailwind components;
@tailwind utilities;

@layer base {
  :root {
    --background: 140 30% 97%;
    --foreground: 160 30% 10%;
    --card: 0 0% 100%;
    --card-foreground: 160 30% 10%;
    --primary: 142 60% 35%; /* Verde saúde */
    --primary-foreground: 0 0% 100%;
    --secondary: 270 50% 55%; /* Roxo */
    --secondary-foreground: 0 0% 100%;
    --muted: 140 20% 93%;
    --muted-foreground: 160 15% 45%;
    --accent: 142 50% 90%;
    --accent-foreground: 142 60% 25%;
    --destructive: 0 84% 60%; /* Vermelho emergência */
  }
}
```

```

--destructive-foreground: 0 0% 98%;
--border: 140 20% 88%;
--radius: 1rem;
}
}

@layer base {
* { @apply border-border outline-ring/50; }
body { @apply bg-background text-foreground font-nunito; }
}

```

5.3 tailwind.config.js

Caminho: tailwind.config.js

```

module.exports = {
  darkMode: ["class"],
  content: ["/index.html", "/src/**/*.{ts,tsx,js,jsx}"],
  theme: {
    extend: {
      fontFamily: { nunito: ['Nunito', 'sans-serif'] },
      borderRadius: {
        lg: 'var(--radius)',
        md: 'calc(var(--radius) - 2px)',
        sm: 'calc(var(--radius) - 4px)'
      },
      colors: {
        background: 'hsl(var(--background))',
        foreground: 'hsl(var(--foreground))',
        primary: { DEFAULT: 'hsl(var(--primary))', foreground: 'hsl(var(--primary-foreground))' },
        secondary: { DEFAULT: 'hsl(var(--secondary))', foreground: 'hsl(var(--secondary-foreground))' },
        muted: { DEFAULT: 'hsl(var(--muted))', foreground: 'hsl(var(--muted-foreground))' },
        accent: { DEFAULT: 'hsl(var(--accent))', foreground: 'hsl(var(--accent-foreground))' },
        destructive: { DEFAULT: 'hsl(var(--destructive))', foreground: 'hsl(var(--destructive-foreground))' },
        border: 'hsl(var(--border))',
        card: { DEFAULT: 'hsl(var(--card))', foreground: 'hsl(var(--card-foreground))' },
      },
      keyframes: {
        'pulse-slow': {
          '0%, 100%': { transform: 'scale(1)', opacity: '1' },
          '50%': { transform: 'scale(1.08)', opacity: '0.9' },
        },
        'float': {
          '0%, 100%': { transform: 'translateY(0)' },
          '50%': { transform: 'translateY(-10px)' },
        },
      },
    },
  },
}

```

```
    animation: {
      'pulse-slow': 'pulse-slow 2s ease-in-out infinite',
      'float': 'float 3s ease-in-out infinite',
    }
  },
  plugins: [require("tailwindcss-animate")],
}
```

5.4 App.jsx

Caminho: App.jsx

```
import { Toaster } from "@components/ui/toaster"
import { QueryClientProvider } from '@tanstack/react-query'
import { queryClientInstance } from '@lib/query-client'
import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
import PageNotFound from './lib/PageNotFound';
import { AuthProvider, useAuth } from '@lib/AuthContext';
import UserNotRegisteredError from '@components/UserNotRegisteredError';
import Splash from './pages/Splash';
import Onboarding from './pages/Onboarding';
import Home from './pages/Home';
import Triagem from './pages/Triagem';
import PastaDigital from './pages/PastaDigital';
import Agenda from './pages/Agenda';
import FilaEspera from './pages/FilaEspera';
import Receituário from './pages/Receituário';
import Perfil from './pages/Perfil';

const AuthenticatedApp = () => {
  const { isLoadingAuth, isLoadingPublicSettings, authError, navigateToLogin } = useAuth();

  if (isLoadingPublicSettings || isLoadingAuth) {
    return (
      <div className="fixed inset-0 flex items-center justify-center">
        <div className="w-8 h-8 border-4 border-slate-200 border-t-slate-800 rounded-full animate-spin" />
      </div>
    );
  }

  if (authError) {
    if (authError.type === 'user_not_registered') return <UserNotRegisteredError />;
  }
```

```
if (authError.type === 'auth_required') { navigateToLogin(); return null; }  
}
```

Continuação 2/2

```
return (  
  <Routes>  
    <Route path="/" element={<Splash />} />  
    <Route path="/onboarding" element={<Onboarding />} />  
    <Route path="/home" element={<Home />} />  
    <Route path="/triagem" element={<Triagem />} />  
    <Route path="/pasta-digital" element={<PastaDigital />} />  
    <Route path="/agenda" element={<Agenda />} />  
    <Route path="/fila" element={<FilaEspera />} />  
    <Route path="/receituario" element={<Receituario />} />  
    <Route path="/perfil" element={<Perfil />} />  
    <Route path="*" element={<PageNotFound />} />  
  </Routes>  
);  
};  
  
function App() {  
  return (  
    <AuthProvider>  
      <QueryClientProvider client={queryClientInstance}>  
        <Router>  
          <AuthenticatedApp />  
        </Router>  
        <Toaster />  
      </QueryClientProvider>  
    </AuthProvider>  
  );  
}  
  
export default App;
```

5.5 entities/Paciente.json

Caminho: entities/Paciente.json

```
{  
  "name": "Paciente",  
  "type": "object",  
  "properties": {
```

```

"nome":      { "type": "string", "title": "Nome" },
"idade":     { "type": "number", "title": "Idade" },
"genero":    { "type": "string", "title": "Gênero",
              "enum": ["Masculino", "Feminino", "Outro", "Prefiro não informar"] },
"cidade":    { "type": "string", "title": "Cidade" },
"tipo_sanguineo": { "type": "string", "title": "Tipo Sanguíneo" },
"problemas_saude": { "type": "array", "title": "Problemas de Saúde",
                    "items": { "type": "string" } },
"medicamento_continuo": { "type": "string", "title": "Medicamento Contínuo" },
"alergias":   { "type": "string", "title": "Alergias" },
"contato_emergencia_nome": { "type": "string", "title": "Contato de Emergência - Nome" },
"contato_emergencia_tel": { "type": "string", "title": "Contato de Emergência - Telefone" },
"permite_localizacao": { "type": "boolean", "title": "Permite Localização" },
"onboarding_concluido": { "type": "boolean", "title": "Onboarding Concluído" }
},
"required": ["nome"]
}

```

5.6 pages/Splash.jsx

Caminho: pages/Splash.jsx

```

import { useEffect } from 'react';
import { useNavigate } from 'react-router-dom';
import { motion } from 'framer-motion';

export default function Splash() {
  const navigate = useNavigate();

  // Redireciona para o onboarding após 3 segundos
  useEffect(() => {
    const timer = setTimeout(() => navigate('/onboarding'), 3000);
    return () => clearTimeout(timer);
  }, [navigate]);

  return (
    <div className="min-h-screen bg-primary flex flex-col items-center justify-center px-8 text-white">
      <motion.div
        initial={{ scale: 0, opacity: 0 }}
        animate={{ scale: 1, opacity: 1 }}
        transition={{ duration: 0.6, ease: 'backOut' }}
        className="flex flex-col items-center gap-6 text-center"
      >
        { /* Ícone principal */ }
        <div className="w-32 h-32 bg-white/20 rounded-full flex items-center justify-center shadow-2xl">
          <div className="text-7xl animate-pulse-slow">👤</div>
        </div>
      </div>
    </div>
  );
}

```

```

<div>
  <h1 className="text-4xl font-black leading-tight">saúde</h1>
  <div className="w-16 h-1 bg-white/60 rounded-full mx-auto my-2" />
  <h2 className="text-2xl font-bold text-white/90">na palma da sua mão</h2>
</div>

```

Continuação 2/2

```

  { /* Indicador de carregamento */ }
  <motion.div initial={{ opacity: 0, y: 20 }} animate={{ opacity: 1, y: 0 }} transition={{ delay: 0.8 }}>
    <div className="flex gap-2 justify-center">
      <span className="w-2 h-2 bg-white/60 rounded-full animate-bounce" style={{ animationDelay: '0ms' }} />
      <span className="w-2 h-2 bg-white/60 rounded-full animate-bounce" style={{ animationDelay: '150ms' }} />
      <span className="w-2 h-2 bg-white/60 rounded-full animate-bounce" style={{ animationDelay: '300ms' }} />
    </div>
  </motion.div>
</motion.div>

<div className="absolute bottom-8 text-white/50 text-xs font-medium">
  Sua saúde, sempre perto de você 💖
</div>
</div>
);
}

```

5.7 pages/Onboarding.jsx

Caminho: pages/Onboarding.jsx

```

import { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import { motion, AnimatePresence } from 'framer-motion';
import { base44 } from '@api/base44Client';

// Definição das 4 missões com progresso
const MISSIONS = [
  { id: 1, title: 'Missão 1: Vamos te conhecer?', emoji: '👋', progress: 10 },
  { id: 2, title: 'Missão 2: Você tem algum problema de saúde?', emoji: '🏥', progress: 45 },
  { id: 3, title: 'Missão 3: Nos informe sobre você', emoji: '👤', progress: 60 },
  { id: 4, title: 'Missão 4: Contato para emergência?', emoji: '🚑', progress: 100 },
];

```

```

const PROBLEMAS = [
  { label: 'Cardíacos', emoji: '❤️' },
  { label: 'Respiratório', emoji: '🫁' },
  { label: 'Diabetes', emoji: '💧' },
  { label: 'Hipertensão', emoji: '🩺' },
  { label: 'Nenhum', emoji: '☑️' },
];

export default function Onboarding() {
  const navigate = useNavigate();
  const [step, setStep] = useState(0); // 0=boas-vindas, 1-4=missões, 5=final
  const [saving, setSaving] = useState(false);
  const [form, setForm] = useState({
    nome: "", idade: "", genero: "",
    problemas_saude: [],
    cidade: "", medicamento_continuo: "", alergias: "",
    contato_emergencia_nome: "", contato_emergencia_tel: "",
    permite_localizacao: false,
  });

```

Continuação 2/8

```

const toggleProblema = (label) => {
  if (label === 'Nenhum') {
    setForm(f => ({ ...f, problemas_saude: ['Nenhum'] }));
    return;
  }
  setForm(f => {
    const sem = f.problemas_saude.filter(p => p !== 'Nenhum');
    if (sem.includes(label)) return { ...f, problemas_saude: sem.filter(p => p !== label) };
    return { ...f, problemas_saude: [...sem, label] };
  });
};

const handleFinish = async () => {
  setSaving(true);
  await base44.entities.Paciente.create({
    ...form,
    idade: Number(form.idade),
    onboarding_concluido: true,
  });
  setSaving(false);
  setStep(5);
  setTimeout(() => navigate('/home'), 2500);
};

const currentMission = MISSIONS[step - 1];

return {

```

```

<div className="min-h-screen bg-background flex flex-col">
  <AnimatePresence mode="wait">

    { /* TELA DE BOAS-VINDAS */ }

```

Continuação 3/8

```

{step === 0 && {
  <motion.div key="welcome"
    initial={{ opacity: 0, y: 30 }} animate={{ opacity: 1, y: 0 }} exit={{ opacity: 0, y: -30 }}
    className="flex-1 flex flex-col items-center justify-center px-8 text-center gap-6"
  >
    <div className="text-6xl animate-pulse-slow">❤️</div>
    <div>
      <h1 className="text-3xl font-black text-foreground">Olá, bem vindo ao nosso atendimento digital!!!</h1>
      <p className="mt-4 text-muted-foreground text-lg font-medium">
        Vamos começar sua missão? Preencha o questionário a seguir.
      </p>
    </div>
    <button onClick={() => setStep(1)}
      className="mt-4 bg-primary text-primary-foreground px-10 py-4 rounded-2xl font-bold text-lg shadow-lg">
      Vamos lá! 🚀
    </button>
  </motion.div>
}}

{ /* MISSÃO 1 — Dados pessoais */ }
{step === 1 && {
  <MissionWrapper mission={currentMission} onNext={() => setStep(2)}
    canNext={!form.nome && !form.idade && !form.genero}>
    <Field label="👤 Como podemos te chamar?">
      <input className="input-field" placeholder="Seu nome"
        value={form.nome} onChange={e => setForm(f => ({ ...f, nome: e.target.value }))) />
    </Field>
    <Field label="🎂 Qual sua idade?">
      <input className="input-field" type="number" placeholder="Ex: 28"
        value={form.idade} onChange={e => setForm(f => ({ ...f, idade: e.target.value }))) />
    </Field>
    <Field label="🏳️ Qual seu gênero?">

```

Continuação 4/8

```

<div className="grid grid-cols-2 gap-2">
  { ['Masculino', 'Feminino', 'Outro', 'Prefiro não informar'].map(g => {
    <button key={g} onClick={() => setForm(f => ({ ...f, genero: g })))
      className={`py-2 px-3 rounded-xl text-sm font-semibold border-2 transition-all
        ${form.genero === g ? 'border-primary bg-accent text-primary' : 'border-border bg-white text-muted-foreground'} `}
    {g}
  })

```

```

    </button>
  }}}
</div>
</Field>
</MissionWrapper>
}}

{/* MISSÃO 2 — Problemas de saúde */}
{step === 2 && (
  <MissionWrapper mission={currentMission} onNext={() => setStep(3)}
  canNext={form.problemas_saude.length > 0}>
    <div className="space-y-3">
      {PROBLEMAS.map(p => (
        <button key={p.label} onClick={() => toggleProblema(p.label)}
          className={`w-full flex items-center gap-3 py-3 px-4 rounded-2xl border-2 transition-all font-semibold text-left
            ${form.problemas_saude.includes(p.label) ? 'border-primary bg-accent text-primary' : 'border-border bg-white text-foreground'}`}>
          <span className="text-xl">{p.emoji}</span>
          <span>{p.label}</span>
          <span className="ml-auto text-lg">{form.problemas_saude.includes(p.label) ? '👍' : '👎'}</span>
        </button>
      ))}
    </div>
  </MissionWrapper>
)}

{/* MISSÃO 3 — Cidade e medicamentos */}

```

Continuação 5/8

```

{step === 3 && (
  <MissionWrapper mission={currentMission} onNext={() => setStep(4)} canNext={!!form.cidade}>
    <Field label="📍 Em qual cidade você se encontra?">
      <input className="input-field" placeholder="Sua cidade"
        value={form.cidade} onChange={e => setForm(f => ({ ...f, cidade: e.target.value }))) />
    </Field>
    <Field label="💊 Você usa algum medicamento contínuo?">
      <input className="input-field" placeholder="Ex: Metformina, Losartana..."
        value={form.medicamento_continuo} onChange={e => setForm(f => ({ ...f, medicamento_continuo: e.target.value }))) />
    </Field>
    <Field label="⚠️ Tem alguma alergia?">
      <input className="input-field" placeholder="Ex: Dipirona, Penicilina..."
        value={form.alergias} onChange={e => setForm(f => ({ ...f, alergias: e.target.value }))) />
    </Field>
  </MissionWrapper>
)}

{/* MISSÃO 4 — Contato de emergência */}
{step === 4 && (
  <MissionWrapper mission={currentMission} onNext={handleFinish} loading={saving} isLast>
    <div className="bg-yellow-50 border border-yellow-200 rounded-2xl p-3 text-sm text-yellow-800 font-medium">

```

```

🏆 Essa é sua última missão!
</div>
<Field label="👤 Nome do contato">
  <input className="input-field" placeholder="Nome completo"
    value={form.contato_emergencia_nome} onChange={e => setForm(f => ({ ...f, contato_emergencia_nome: e.target.value })} />
</Field>
<Field label="☎ Telefone">
  <input className="input-field" placeholder="(00) 00000-0000"
    value={form.contato_emergencia_tel} onChange={e => setForm(f => ({ ...f, contato_emergencia_tel: e.target.value })} />
</Field>
<button onClick={() => setForm(f => ({ ...f, permite_localizacao: !f.permite_localizacao })} />

```

Continuação 6/8

```

className={w-full flex items-center gap-3 py-3 px-4 rounded-2xl border-2 transition-all font-semibold
  ${form.permite_localizacao ? 'border-primary bg-accent text-primary' : 'border-border bg-white text-foreground'}}>
<span className="text-xl">🔒 </span>
<span className="flex-1 text-left text-sm">Permitir acesso à minha localização?</span>
<span>{form.permite_localizacao ? '☑ Sim' : '☐ Não'}</span>
</button>
</MissionWrapper>
}}

{/* TELA FINAL — Perfil sendo criado */}
{step === 5 && (
  <motion.div key="final" initial={{ opacity: 0, scale: 0.8 }} animate={{ opacity: 1, scale: 1 }}
    className="flex-1 flex flex-col items-center justify-center px-8 text-center gap-6">
    <div className="text-8xl animate-float">🏆 </div>
    <div>
      <h2 className="text-3xl font-black text-foreground">Pronto!</h2>
      <p className="mt-3 text-xl font-semibold text-muted-foreground">Seu perfil de saúde está sendo criado.</p>
      <p className="mt-2 text-lg text-primary font-bold">Aproveite! 😊</p>
    </div>
  </motion.div>
)}

</AnimatePresence>
</div>
);
}

```

```

// Componente reutilizável: wrapper de missão com header e barra de progresso
function MissionWrapper({ mission, children, onNext, canNext = true, loading, isLast }) {
  return (
    <motion.div key={mission.id}
      initial={{ opacity: 0, x: 60 }} animate={{ opacity: 1, x: 0 }} exit={{ opacity: 0, x: -60 }}

```

Continuação 7/8

```

className="flex-1 flex flex-col min-h-screen">

{/* Header com progresso */}
<div className="bg-primary px-6 pt-14 pb-6">
  <div className="flex items-center gap-2 mb-3">
    <span className="text-white/70 text-sm font-semibold">● {mission.title}</span>
  </div>
  <div className="w-full bg-white/30 rounded-full h-3">
    <motion.div className="bg-white rounded-full h-3"
      initial={{ width: 0 }} animate={{ width: `${mission.progress}%` }}
      transition={{ duration: 0.8, ease: 'easeOut' }} />
  </div>
  <div className="text-white/80 text-xs mt-1 text-right font-bold">{mission.progress}%</div>
</div>

{/* Conteúdo */}
<div className="flex-1 px-6 py-6 space-y-4 overflow-y-auto">{children}</div>

{/* Botão de avançar */}
<div className="px-6 pb-8">
  <button onClick={onNext} disabled={!canNext || loading}
    className="w-full bg-primary text-primary-foreground py-4 rounded-2xl font-bold text-lg shadow-lg disabled:opacity-40">
    {loading ? '⌛ Salvando...' : isLast ? '✅ Concluir' : 'Próxima missão →'}
  </button>
</div>
</motion.div>
);
}

// Componente reutilizável: campo de formulário com label
function Field({ label, children }) {
  return (

```

Continuação 8/8

```

<div className="space-y-2">
  <label className="text-sm font-bold text-foreground">{label}</label>
  {children}
</div>
);
}

```

5.8 pages/Home.jsx

Caminho: pages/Home.jsx

```

import { useState, useEffect } from 'react';
import { useNavigate } from 'react-router-dom';
import { motion } from 'framer-motion';
import { base44 } from '@api/base44Client';
import SOSButton from '../components/SOSButton';

const MODULES = [
  { icon: '🔍', label: 'Triagem Inteligente', desc: 'Descubra onde se tratar', path: '/triagem', color: 'bg-emerald-50 border-emerald-200' },
  { icon: '📁', label: 'Pasta Digital', desc: 'Exames e receitas', path: '/pasta-digital', color: 'bg-blue-50 border-blue-200' },
  { icon: '📅', label: 'Agenda', desc: 'Consultas e lembretes', path: '/agenda', color: 'bg-purple-50 border-purple-200' },
  { icon: '👤', label: 'Fila de Espera', desc: 'Acompanhe sua senha', path: '/fila', color: 'bg-orange-50 border-orange-200' },
  { icon: '💊', label: 'Receituário', desc: 'Suas prescrições', path: '/receituario', color: 'bg-pink-50 border-pink-200' },
  { icon: '💬', label: 'Chat Médico', desc: 'Tire suas dúvidas', path: '/chat', color: 'bg-cyan-50 border-cyan-200' },
  { icon: '📍', label: 'Mapa de Saúde', desc: 'Unidades próximas', path: '/mapa', color: 'bg-yellow-50 border-yellow-200' },
  { icon: '👤', label: 'Meu Perfil', desc: 'Dados e histórico', path: '/perfil', color: 'bg-slate-50 border-slate-200' },
];

export default function Home() {
  const navigate = useNavigate();
  const [paciente, setPaciente] = useState(null);
  const [showSOS, setShowSOS] = useState(false);

  useEffect(() => {
    base44.entities.Paciente.list('-created_date', 1).then(data => {
      if (data?.length > 0) setPaciente(data[0]);
    });
  }, []);

  return (
    <div className="min-h-screen bg-background pb-8">
      { /* Header com saudação */ }
      <div className="bg-primary px-6 pt-12 pb-8 rounded-b-3xl shadow-lg">

```

Continuação 2/4

```

<div className="flex items-center justify-between">
  <div>
    <p className="text-white/70 text-sm font-medium">Olá! 🤖</p>
    <h1 className="text-white text-2xl font-black">{paciente?.nome || 'Bem-vindo!'}</h1>
    <p className="text-white/70 text-xs mt-1">{paciente?.cidade && `📍 ${paciente?.cidade}`}</p>
  </div>
  <div className="w-14 h-14 bg-white/20 rounded-full flex items-center justify-center text-3xl">🏠</div>
</div>

{ /* Tags rápidas de saúde */ }
{paciente && (
  <div className="mt-4 flex gap-2 flex-wrap">
    {paciente.tipo_sanguineo && (
      <span className="bg-white/20 text-white text-xs font-semibold px-3 py-1 rounded-full">🩸 {paciente.tipo_sanguineo}</span>

```

```

    }}
    {paciente.alergias && {
      <span className="bg-red-400/40 text-white text-xs font-semibold px-3 py-1 rounded-full"> ⚠ Alergia</span>
    }}
    {paciente.medicamento_continuo && {
      <span className="bg-white/20 text-white text-xs font-semibold px-3 py-1 rounded-full"> 💊 Medicação</span>
    }}
  </div>
}}
</div>

<div className="px-6 mt-6 space-y-6">
  { /* Botão SOS */ }
  <motion.button onClick={() => setShowSOS(true)} whileTap={{ scale: 0.95 }}
    className="w-full bg-destructive text-white py-5 rounded-3xl font-black text-xl flex items-center justify-center gap-3 shadow-xl"
    animate={{ boxShadow: ['0 8px 30px rgba(239,68,68,0.3)', '0 8px 40px rgba(239,68,68,0.5)', '0 8px 30px rgba(239,68,68,0.3)'] }}
    transition={{ duration: 1.5, repeat: Infinity }}>
     EMERGÊNCIA / SOS
  </motion.button>

```

Continuação 3/4

```

</motion.button>

{ /* Triagem em destaque */ }
<motion.button onClick={() => navigate('/triagem')}
  initial={{ opacity: 0, y: 20 }} animate={{ opacity: 1, y: 0 }}
  className="w-full bg-white border-2 border-primary rounded-3xl p-5 flex items-center gap-4 shadow-sm">
  <div className="w-14 h-14 bg-accent rounded-2xl flex items-center justify-center text-3xl"> 🚑 </div>
  <div className="text-left">
    <p className="font-black text-foreground text-lg">Triagem Inteligente</p>
    <p className="text-muted-foreground text-sm">Onde devo ir me tratar agora?</p>
  </div>
  <span className="ml-auto text-2xl">→</span>
</motion.button>

{ /* Grade de módulos */ }
<div>
  <h2 className="font-black text-foreground text-lg mb-3">Serviços</h2>
  <div className="grid grid-cols-2 gap-3">
    {MODULES.slice(1).map((mod, i) => {
      <motion.button key={mod.path} onClick={() => navigate(mod.path)}
        initial={{ opacity: 0, y: 20 }} animate={{ opacity: 1, y: 0 }}
        transition={{ delay: i * 0.05 }}
        className={` ${mod.color} border-2 rounded-2xl p-4 flex flex-col items-start gap-2`}>
        <span className="text-3xl">{mod.icon}</span>
        <div>
          <p className="font-bold text-foreground text-sm leading-tight">{mod.label}</p>
          <p className="text-muted-foreground text-xs">{mod.desc}</p>
        </div>
      </motion.button>
    })}
  </div>

```

```
    )})  
  </div>  
</div>
```

Continuação 4/4

```
    { /* Dica do dia */  
    <div className="bg-accent border border-primary/20 rounded-3xl p-5">  
      <p className="text-xs font-bold text-primary uppercase tracking-wide mb-2">💡 Dica do Dia</p>  
      <p className="text-foreground font-semibold text-sm">  
        Beba pelo menos 8 copos de água por dia. A hidratação é essencial para o bom funcionamento do seu corpo! 💧  
      </p>  
    </div>  
  </div>  
  
  {showSOS && <SOSButton onClose={() => setShowSOS(false)} />}  
</div>  
};  
}
```

5.9 components/SOSButton.jsx

Caminho: components/SOSButton.jsx

```
import { useState } from 'react';  
import { motion } from 'framer-motion';  
  
export default function SOSButton({ onClose }) {  
  const [confirmed, setConfirmed] = useState(false);  
  
  const handleConfirm = () => {  
    setConfirmed(true);  
    setTimeout(() => { onClose(); setConfirmed(false); }, 4000);  
  };  
  
  return (  
    <motion.div initial={{ opacity: 0 }} animate={{ opacity: 1 }} exit={{ opacity: 0 }}  
      className="fixed inset-0 bg-black/70 z-50 flex items-center justify-center px-6"  
      onClick={onClose}>  
      <motion.div initial={{ scale: 0.8, y: 40 }} animate={{ scale: 1, y: 0 }}  
        className="bg-white rounded-3xl p-8 w-full max-w-sm text-center shadow-2xl"  
        onClick={e => e.stopPropagation()}>  
  
        {!confirmed ? (  

```

```

<>
<div className="text-6xl mb-4">🆘</div>
<h2 className="text-2xl font-black text-foreground mb-2">Emergência!</h2>
<p className="text-muted-foreground mb-6 font-medium">
  Confirme para acionar ajuda e compartilhar sua localização.
</p>
<div className="space-y-3">
  <button onClick={handleConfirm}
    className="w-full bg-destructive text-white py-4 rounded-2xl font-black text-lg">
    ☒ SIM, PRECISO DE AJUDA
  </button>
  <button onClick={onClose}

```

Continuação 2/2

```

    className="w-full bg-muted text-muted-foreground py-3 rounded-2xl font-semibold">
      Cancelar
    </button>
  </div>
  { /* Telefones de emergência */ }
  <div className="mt-6 grid grid-cols-2 gap-3 text-sm">
    <a href_="tel:192" className="bg-red-50 border border-red-200 rounded-xl py-3 font-bold text-red-600">📞 SAMU 192</a>
    <a href_="tel:193" className="bg-orange-50 border border-orange-200 rounded-xl py-3 font-bold text-orange-600">🚒 Bombeiros 193</a>
  </div>
</>
): {
  <div className="py-4">
    <div className="text-6xl mb-4 animate-pulse">📶</div>
    <h2 className="text-2xl font-black text-primary mb-2">Ajuda acionada!</h2>
    <p className="text-muted-foreground font-medium">Sua localização está sendo compartilhada. Ajuda a caminho! ❤️</p>
  </div>
  }
</motion.div>
</motion.div>
);
}

```

5.10 pages/Triagem.jsx

Caminho: pages/Triagem.jsx

```

import { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import { motion } from 'framer-motion';

```

```

import { base44 } from '@api/base44Client';

const SINTOMAS = [
  '🤒 Febre', '🤯 Dor de cabeça', '🫁 Dificuldade de respirar',
  '❤️ Dor no peito', '🤢 Náusea/Vômito', '💩 Diarreia',
  '😓 Cansaço extremo', '💧 Sangramento', '💪 Dor muscular',
  '👁️ Visão turva', '🌀 Tontura', '🩹 Desmaio',
];

const INTENSIDADES = [
  { label: 'Leve', emoji: '🟢', value: 1 },
  { label: 'Moderada', emoji: '🟡', value: 2 },
  { label: 'Intensa', emoji: '🟠', value: 3 },
  { label: 'Insuportável', emoji: '🔴', value: 4 },
];

// Mapeamento de classificação para local de atendimento
const LOCAIS = {
  leve: { local: 'UBS – Unidade Básica de Saúde', emoji: '🏠', cor: 'bg-green-50 border-green-300', badge: 'LEVE', badgeCor: 'bg-green-500' },
  moderado: { local: 'Clínica de Saúde ou Médico de Família', emoji: '👨‍⚕️', cor: 'bg-yellow-50 border-yellow-300', badge: 'MODERADO', badgeCor: 'bg-yellow-500' },
  urgente: { local: 'UPA – Unidade de Pronto Atendimento', emoji: '🚑', cor: 'bg-orange-50 border-orange-300', badge: 'URGENTE', badgeCor: 'bg-orange-500' },
  emergencia: { local: 'SAMU 192 ou Hospital de Emergência', emoji: '🚒', cor: 'bg-red-50 border-red-300', badge: 'EMERGÊNCIA', badgeCor: 'bg-red-500' },
};

export default function Triage() {
  const navigate = useNavigate();
  const [sintomas, setSintomas] = useState([]);
  const [intensidade, setIntensidade] = useState(null);
  const [tempo, setTempo] = useState("");

```

Continuação 2/5

```

const [resultado, setResultado] = useState(null);
const [loading, setLoading] = useState(false);

const toggleSintoma = (s) =>
  setSintomas(prev => prev.includes(s) ? prev.filter(x => x !== s) : [...prev, s]);

// Chama a IA para classificar o caso
const calcularResultado = async () => {
  setLoading(true);
  const res = await base44.integrations.Core.InvokeLLM({
    prompt: `Você é um sistema de triagem médica. Classifique em: leve, moderado, urgente ou emergencia.
    Sintomas: ${sintomas.join(', ')}
    Intensidade: ${INTENSIDADES.find(i => i.value === intensidade)?.label}
    Tempo: ${tempo}
    Responda JSON: { "classificacao": "leve|moderado|urgente|emergencia", "orientacao": "texto em português simples" },
    response_json_schema: {
      type: 'object',

```

```

properties: {
  classificacao: { type: 'string' },
  orientacao: { type: 'string' },
},
},
});
setResultado(res);
setLoading(false);
};

const rec = resultado ? LOCAIS[resultado.classificacao] || LOCAIS.moderado : null;

return (
  <div className="min-h-screen bg-background pb-8">
    <div className="bg-primary px-6 pt-12 pb-6 rounded-b-3xl">

```

Continuação 3/5

```

<button onClick={() => navigate('/home')} className="text-white/70 font-semibold mb-3">← Voltar</button>
<h1 className="text-white text-2xl font-black">👁️ Triage Inteligente</h1>
<p className="text-white/70 text-sm mt-1">Descubra onde você deve ser atendido</p>
</div>

<div className="px-6 mt-6 space-y-6">
  {!resultado ? (
    <>
      {/* Seleção de sintomas */}
      <div>
        <h2 className="font-black text-foreground mb-3">Quais são seus sintomas?</h2>
        <div className="grid grid-cols-2 gap-2">
          {SINTOMAS.map(s => (
            <button key={s} onClick={() => toggleSintoma(s)}
              className={`py-2 px-3 rounded-xl text-sm font-semibold border-2 text-left transition-all
                ${sintomas.includes(s) ? 'border-primary bg-accent text-primary' : 'border-border bg-white text-foreground'}`}>
              {s}
            </button>
          ))}
        </div>
      </div>

      {/* Intensidade */}
      <div>
        <h2 className="font-black text-foreground mb-3">Intensidade da dor/desconforto</h2>
        <div className="grid grid-cols-2 gap-2">
          {INTENSIDADES.map(i => (
            <button key={i.value} onClick={() => setIntensidade(i.value)}
              className={`py-3 px-4 rounded-xl font-bold border-2 transition-all flex items-center gap-2
                ${intensidade === i.value ? 'border-primary bg-accent text-primary' : 'border-border bg-white text-foreground'}`}>
              <span className="text-lg">{i.emoji}</span> {i.label}
            </button>
          ))}
        </div>
      </div>
    </>
  ) : (
    <div>
      <h2 className="font-black text-foreground mb-3">Quais são seus sintomas?</h2>
      <div className="grid grid-cols-2 gap-2">
        {SINTOMAS.map(s => (
          <button key={s} onClick={() => toggleSintoma(s)}
            className={`py-2 px-3 rounded-xl text-sm font-semibold border-2 text-left transition-all
              ${sintomas.includes(s) ? 'border-primary bg-accent text-primary' : 'border-border bg-white text-foreground'}`}>
            {s}
          </button>
        ))}
      </div>

      <div>
        <h2 className="font-black text-foreground mb-3">Intensidade da dor/desconforto</h2>
        <div className="grid grid-cols-2 gap-2">
          {INTENSIDADES.map(i => (
            <button key={i.value} onClick={() => setIntensidade(i.value)}
              className={`py-3 px-4 rounded-xl font-bold border-2 transition-all flex items-center gap-2
                ${intensidade === i.value ? 'border-primary bg-accent text-primary' : 'border-border bg-white text-foreground'}`}>
            <span className="text-lg">{i.emoji}</span> {i.label}
          </button>
        ))}
      </div>
    </div>
  )}
</div>

```

Continuação 4/5

```

    )}}
  </div>
</div>

  { /* Tempo dos sintomas */
  <div>
    <h2 className="font-black text-foreground mb-3">Há quanto tempo?</h2>
    <div className="grid grid-cols-3 gap-2">
      {[ 'Hoje', 'Alguns dias', 'Semanas' ].map(t => {
        <button key={t} onClick={() => setTempo(t)}
          className={`py-2 rounded-xl text-sm font-semibold border-2 transition-all
            ${tempo === t ? 'border-primary bg-accent text-primary' : 'border-border bg-white text-foreground'} `}>
          {t}
        </button>
      )}}
    </div>

    <button onClick={calcularResultado} disabled={sintomas.length === 0 || lintensidade || loading}
      className="w-full bg-primary text-primary-foreground py-4 rounded-2xl font-black text-lg shadow-lg disabled:opacity-40">
      {loading ? ' ⌛ Analisando com IA...' : ' 🏥 Fazer Triagem'}
    </button>
  </>
): {
  /* Resultado da triagem */
  <motion.div initial={{ opacity: 0, y: 30 }} animate={{ opacity: 1, y: 0 }} className="space-y-4">
    <div className={` ${rec.cor} border-2 rounded-3xl p-6 text-center`}>
      <span className={` ${rec.badgeCor} text-white text-xs font-black px-3 py-1 rounded-full`}>{rec.badge}</span>
      <div className="text-6xl mt-4 mb-3">{rec.emoji}</div>
      <h2 className="text-xl font-black text-foreground">Procure:</h2>
      <p className="text-lg font-bold text-foreground mt-1">{rec.local}</p>
    </div>

    <div className="bg-yellow-50 border border-yellow-200 rounded-2xl p-4 text-xs text-yellow-800 font-medium">
      ⚠️ Esta triagem é apenas uma orientação inicial e não substitui avaliação médica profissional.
    </div>

    <button onClick={() => { setResultado(null); setSintomas([]); setIntensidade(null); setTempo(""); }}

```

Continuação 5/5

```

    <div className="bg-white border-2 border-border rounded-2xl p-4">
      <p className="font-bold text-foreground mb-2">📍 Orientação:</p>
      <p className="text-muted-foreground text-sm leading-relaxed">{resultado.orientacao}</p>
    </div>

    <div className="bg-yellow-50 border border-yellow-200 rounded-2xl p-4 text-xs text-yellow-800 font-medium">
      ⚠️ Esta triagem é apenas uma orientação inicial e não substitui avaliação médica profissional.
    </div>

    <button onClick={() => { setResultado(null); setSintomas([]); setIntensidade(null); setTempo(""); }}

```

```

        className="w-full bg-muted text-muted-foreground py-3 rounded-2xl font-semibold">
        Fazer nova triagem
      </button>
    </motion.div>
  )}
</div>
</div>
};
}

```

5.11 pages/PastaDigital.jsx

Caminho: pages/PastaDigital.jsx

```

import { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import { motion } from 'framer-motion';

const CATEGORIAS = [
  { icon: '📋', label: 'Exames', count: 3 },
  { icon: '📄', label: 'Laudos', count: 1 },
  { icon: '🍽️', label: 'Receitas', count: 2 },
  { icon: '💉', label: 'Vacinas', count: 8 },
  { icon: '🏠', label: 'Internações', count: 0 },
  { icon: '🔪', label: 'Cirurgias', count: 0 },
];

const DOCS_DEMO = [
  { id: 1, tipo: 'Exame', nome: 'Hemograma Completo', data: '15/03/2026', status: 'Disponível', emoji: '📋' },
  { id: 2, tipo: 'Exame', nome: 'Glicemia em Jejum', data: '10/03/2026', status: 'Disponível', emoji: '💧' },
  { id: 3, tipo: 'Receita', nome: 'Amoxicilina 500mg', data: '02/04/2026', status: 'Válida', emoji: '🍽️' },
  { id: 4, tipo: 'Receita', nome: 'Dipirona 500mg', data: '01/03/2026', status: 'Vencida', emoji: '💉' },
];

export default function PastaDigital() {
  const navigate = useNavigate();
  const [tab, setTab] = useState('Todos');

  return (
    <div className="min-h-screen bg-background pb-8">
      <div className="bg-blue-600 px-6 pt-12 pb-6 rounded-b-3xl">
        <button onClick={() => navigate('/home')} className="text-white/70 font-semibold mb-3">← Voltar</button>
        <h1 className="text-white text-2xl font-black">📁 Pasta Digital de Saúde</h1>
        <p className="text-white/70 text-sm mt-1">Todos os seus documentos em um só lugar</p>
      </div>
    </div>
  );
}

```

Continuação 2/3

```
<div className="px-6 mt-6 space-y-5">
  {/* Grid de categorias */}
  <div className="grid grid-cols-3 gap-3">
    {CATEGORIAS.map(cat => {
      <div key={cat.label} className="bg-white border-2 border-border rounded-2xl p-3 text-center">
        <div className="text-2xl">{cat.icon}</div>
        <p className="text-xs font-bold text-foreground mt-1">{cat.label}</p>
        <p className="text-lg font-black text-primary">{cat.count}</p>
      </div>
    })}
  </div>

  {/* Filtros por tipo */}
  <div className="flex gap-2 overflow-x-auto pb-1">
    {[ 'Todos', 'Exames', 'Receitas', 'Vacinas' ].map(t => {
      <button key={t} onClick={() => setTab(t)}
        className={`px-4 py-2 rounded-xl text-sm font-bold whitespace-nowrap transition-all
          ${tab === t ? 'bg-primary text-white' : 'bg-white border border-border text-muted-foreground'}`}>
        {t}
      </button>
    })}
  </div>

  {/* Lista de documentos */}
  <div className="space-y-3">
    {DOCS_DEMO.filter(d => tab === 'Todos' || d.tipo === tab.slice(0, -1)).map((doc, i) => {
      <motion.div key={doc.id} initial={{ opacity: 0, x: -20 }} animate={{ opacity: 1, x: 0 }}
        transition={{ delay: i * 0.07 }}
        className="bg-white border-2 border-border rounded-2xl p-4 flex items-center gap-4">
        <div className="w-12 h-12 bg-accent rounded-xl flex items-center justify-center text-2xl">{doc.emoji}</div>
        <div className="flex-1">
          <p className="font-bold text-foreground text-sm">{doc.nome}</p>
```

Continuação 3/3

```
          <p className="text-muted-foreground text-xs">{doc.tipo} • {doc.data}</p>
        </div>
        <span className={`text-xs font-bold px-2 py-1 rounded-lg
          ${doc.status === 'Disponível' || doc.status === 'Válida' ? 'bg-green-100 text-green-700' : 'bg-red-100 text-red-700'}`}>
          {doc.status}
        </span>
      </motion.div>
    })}
  </div>

  <button className="w-full border-2 border-dashed border-primary text-primary py-4 rounded-2xl font-bold text-lg">
    + Adicionar documento
```

```

    </button>
  </div>
</div>
);
}

```

5.12 pages/Agenda.jsx

Caminho: pages/Agenda.jsx

```

import { useNavigate } from 'react-router-dom';
import { motion } from 'framer-motion';

const CONSULTAS = [
  { id: 1, medico: 'Dr. Carlos Silva', especialidade: 'Clínico Geral', tipo: 'Consulta', data: '10/04/2026', hora: '09:30', local: 'UBS Centro', emoji: '👤', status: 'confirmado' },
  { id: 2, medico: 'Laboratório Central', especialidade: 'Hemograma', tipo: 'Exame', data: '12/04/2026', hora: '07:00', local: 'Lab Saúde+', emoji: '🔬', status: 'confirmado' },
  { id: 3, medico: 'Dra. Ana Souza', especialidade: 'Cardiologista', tipo: 'Retorno', data: '20/04/2026', hora: '14:00', local: 'Clínica CardioVida', emoji: '❤️', status: 'pendente' },
];

export default function Agenda() {
  const navigate = useNavigate();

  return (
    <div className="min-h-screen bg-background pb-8">
      <div className="bg-purple-600 px-6 pt-12 pb-6 rounded-b-3xl">
        <button onClick={() => navigate('/home')} className="text-white/70 font-semibold mb-3">← Voltar</button>
        <h1 className="text-white text-2xl font-black">📅 Agenda de Saúde</h1>
        <p className="text-white/70 text-sm mt-1">Suas consultas e lembretes</p>
      </div>

      <div className="px-6 mt-6 space-y-4">
        <div className="flex justify-between items-center">
          <h2 className="font-black text-foreground text-lg">Próximos compromissos</h2>
          <button className="text-primary font-bold text-sm">+ Novo</button>
        </div>

        {CONSULTAS.map((c, i) => (
          <motion.div key={c.id} initial={{ opacity: 0, y: 20 }} animate={{ opacity: 1, y: 0 }}
            transition={{ delay: i * 0.1 }}
            className="bg-white border-2 border-border rounded-2xl p-4">
              <div className="flex items-start gap-3">
                <div className="w-12 h-12 bg-accent rounded-xl flex items-center justify-center text-2xl shrink-0">{c.emoji}</div>

```

```

<div className="flex-1">
  <div className="flex items-center justify-between">
    <p className="font-black text-foreground">{c.medico}</p>
    <span className={`text-xs font-bold px-2 py-1 rounded-lg
      ${c.status === 'confirmado' ? 'bg-green-100 text-green-700' : 'bg-yellow-100 text-yellow-700'}`}>
      {c.status === 'confirmado' ? '✅ Confirmado' : '⌚ Pendente'}
    </span>
  </div>
  <p className="text-muted-foreground text-sm">{c.especialidade} • {c.tipo}</p>
  <div className="mt-2 flex gap-3 text-xs text-muted-foreground font-medium">
    <span>📅 {c.data}</span>
    <span>🕒 {c.hora}</span>
    <span>📍 {c.local}</span>
  </div>
</div>
</div>
</motion.div>
  )}
}

<div className="bg-accent border border-primary/20 rounded-2xl p-4">
  <p className="font-bold text-primary mb-2">🔔 Lembrete</p>
  <p className="text-sm text-foreground font-medium">Sua vacina da gripe vence em 15 dias. Agende no posto mais próximo!</p>
</div>
</div>
</div>
);
}

```

5.13 pages/FilaEspera.jsx

Caminho: pages/FilaEspera.jsx

```

import { useState, useEffect } from 'react';
import { useNavigate } from 'react-router-dom';
import { motion } from 'framer-motion';

export default function FilaEspera() {
  const navigate = useNavigate();
  const [senha] = useState('A-047');
  const [posicao, setPosicao] = useState(8);
  const [atual] = useState('A-039');

  // Simula avanço da fila a cada 8 segundos
  useEffect(() => {
    const interval = setInterval(() => {
      setPosicao(p => p > 1 ? p - 1 : p);
    }, 8000);
  }, []);
}

```

```

return () => clearInterval(interval);
}, []);

const tempoEstimado = posicao * 7;

return (
  <div className="min-h-screen bg-background pb-8">
    <div className="bg-orange-500 px-6 pt-12 pb-6 rounded-b-3xl">
      <button onClick={() => navigate('/home')} className="text-white/70 font-semibold mb-3">← Voltar</button>
      <h1 className="text-white text-2xl font-black">🚦 Fila de Espera</h1>
      <p className="text-white/70 text-sm mt-1">Acompanhe sua posição em tempo real</p>
    </div>

    <div className="px-6 mt-6 space-y-5">
      {/* Sua senha */}
      <div className="bg-white border-2 border-orange-200 rounded-3xl p-6 text-center">
        <p className="text-muted-foreground font-semibold text-sm mb-2">Sua senha</p>

```

Continuação 2/3

```

<p className="text-6xl font-black text-orange-500">{senha}</p>
<div className="mt-4 flex justify-center gap-6 text-sm font-semibold text-muted-foreground">
  <div className="text-center">
    <p className="text-2xl font-black text-foreground">{posicao}</p>
    <p>na fila</p>
  </div>
  <div className="w-px bg-border" />
  <div className="text-center">
    <p className="text-2xl font-black text-foreground">~{tempoEstimado}min</p>
    <p>estimado</p>
  </div>
</div>
</div>

{/* Atendimento atual */}
<div className="bg-green-50 border-2 border-green-200 rounded-2xl p-4 flex items-center gap-3">
  <div className="w-10 h-10 bg-green-500 rounded-xl flex items-center justify-center text-white font-black text-sm">{atual}</div>
  <div>
    <p className="font-bold text-foreground text-sm">Atendendo agora</p>
    <p className="text-muted-foreground text-xs">Guichê 3 • UBS Centro</p>
  </div>
  <div className="ml-auto w-3 h-3 bg-green-500 rounded-full animate-pulse" />
</div>

{/* Barra de progresso */}
<div className="bg-white border-2 border-border rounded-2xl p-4">
  <div className="flex justify-between text-sm font-bold text-muted-foreground mb-2">
    <span>Progresso</span>
    <span>{Math.round(((40 - posicao) / 40) * 100)}%</span>
  </div>

```

```
<div className="w-full bg-muted rounded-full h-4">
  <motion.div className="bg-orange-500 rounded-full h-4"
```

Continuação 3/3

```
    animate={{ width: `${((40 - posicao) / 40) * 100}%` }}
    transition={{ duration: 1 }} />
  </div>
</div>

<div className="bg-blue-50 border border-blue-200 rounded-2xl p-4 text-sm text-blue-800 font-medium">
  📄 Você receberá uma notificação quando faltarem 3 pessoas para o seu atendimento.
</div>
</div>
</div>
);
}
```

5.14 pages/Receituario.jsx

Caminho: pages/Receituario.jsx

```
import { useNavigate } from 'react-router-dom';
import { motion } from 'framer-motion';

const RECEITAS = [
  { id: 1, medicamento: 'Amoxicilina 500mg', medico: 'Dr. Carlos Silva', dose: '1 comprimido de 8 em 8h por 7 dias', validade: '02/05/2026', status: 'valida' },
  { id: 2, medicamento: 'Omeprazol 20mg', medico: 'Dra. Ana Souza', dose: '1 comprimido em jejum por 30 dias', validade: '10/04/2026', status: 'valida' },
  { id: 3, medicamento: 'Dipirona 500mg', medico: 'Dr. Carlos Silva', dose: 'Tomar quando necessário', validade: '01/03/2026', status: 'vencida' },
];

export default function Receituario() {
  const navigate = useNavigate();

  return (
    <div className="min-h-screen bg-background pb-8">
      <div className="bg-pink-600 px-6 pt-12 pb-6 rounded-b-3xl">
        <button onClick={() => navigate('/home')} className="text-white/70 font-semibold mb-3">← Voltar</button>
        <h1 className="text-white text-2xl font-black">🏠 Receituário Digital</h1>
        <p className="text-white/70 text-sm mt-1">Suas prescrições médicas</p>
      </div>

      <div className="px-6 mt-6 space-y-4">
        {RECEITAS.map((r, i) => (
          <motion.div key={r.id} initial={{ opacity: 0, y: 20 }} animate={{ opacity: 1, y: 0 }}>
```

```

transition={{ delay: i * 0.1 }}
className={`bg-white border-2 rounded-2xl p-5 ${r.status === 'valida' ? 'border-green-200' : 'border-red-200 opacity-70'}}`}>
<div className="flex items-start justify-between mb-3">
  <div className="flex items-center gap-2">
    <span className="text-2xl"><img alt="pill icon" data-bbox="212 141 228 157"/></span>
    <p className="font-black text-foreground">{r.medicamento}</p>
  </div>
  <span className={`text-xs font-bold px-2 py-1 rounded-lg
    ${r.status === 'valida' ? 'bg-green-100 text-green-700' : 'bg-red-100 text-red-700'}}`>

```

Continuação 2/2

```

    {r.status === 'valida' ? '✅ Válida' : '❌ Vencida'}
  </span>
</div>
<p className="text-sm text-muted-foreground mb-1"><img alt="doctor icon" data-bbox="308 351 324 367"/> {r.medico}</p>
<p className="text-sm font-medium text-foreground mb-2"><img alt="pill icon" data-bbox="334 368 350 384"/> {r.dose}</p>
<p className="text-xs text-muted-foreground">Validade: {r.validade}</p>
</motion.div>
)}}

<button className="w-full border-2 border-dashed border-pink-400 text-pink-600 py-4 rounded-2xl font-bold text-lg">
  + Adicionar receita
</button>
</div>
</div>
);
}

```

5.15 pages/Perfil.jsx

Caminho: pages/Perfil.jsx

```

import { useState, useEffect } from 'react';
import { useNavigate } from 'react-router-dom';
import { base44 } from '@api/base44Client';

export default function Perfil() {
  const navigate = useNavigate();
  const [paciente, setPaciente] = useState(null);

  useEffect(() => {
    base44.entities.Paciente.list('-created_date', 1).then(data => {
      if (data?.length > 0) setPaciente(data[0]);
    });
  });
}

```

```

}, []);

const info = [
  { label: 'Nome',      value: paciente?.nome,      emoji: '👤' },
  { label: 'Idade',     value: paciente?.idade ? `${paciente.idade} anos` : null,      emoji: '🎂' },
  { label: 'Gênero',    value: paciente?.genero,      emoji: '🧑' },
  { label: 'Cidade',    value: paciente?.cidade,      emoji: '🏡' },
  { label: 'Tipo Sanguíneo', value: paciente?.tipo_sanguineo,      emoji: '🩸' },
  { label: 'Alergias',   value: paciente?.alergias,      emoji: '⚠️' },
  { label: 'Medicação Contínua', value: paciente?.medicamento_continuo,      emoji: '💊' },
  { label: 'Contato Emergência', value: paciente?.contato_emergencia_nome
    ? `${paciente.contato_emergencia_nome} • ${paciente.contato_emergencia_tel}` : null,      emoji: '☎️' },
];

return (
<div className="min-h-screen bg-background pb-8">
  <div className="bg-slate-700 px-6 pt-12 pb-6 rounded-b-3xl">
    <button onClick={() => navigate('/home')} className="text-white/70 font-semibold mb-3">← Voltar</button>
    <h1 className="text-white text-2xl font-black">👤 Meu Perfil</h1>
    <p className="text-white/70 text-sm mt-1">Seus dados de saúde</p>
  </div>
</div>

```

Continuação 2/3

```

</div>

<div className="px-6 mt-6 space-y-4">
  { /* Avatar */ }
  <div className="flex flex-col items-center py-4">
    <div className="w-24 h-24 bg-accent rounded-full flex items-center justify-center text-5xl border-4 border-primary">👤 </div>
    <h2 className="text-2xl font-black text-foreground mt-3">{paciente?.nome || 'Usuário'}</h2>
    {paciente?.problemas_saude?.length > 0 && {
      <div className="flex gap-2 mt-2 flex-wrap justify-center">
        {paciente.problemas_saude.map(p => {
          <span key={p} className="bg-accent text-primary text-xs font-semibold px-2 py-1 rounded-full">{p}</span>
        })}
      </div>
    )}
  </div>
</div>

{ /* Informações */ }
<div className="bg-white border-2 border-border rounded-2xl divide-y divide-border">
  {info.map(item => item.value ? {
    <div key={item.label} className="flex items-center gap-3 px-4 py-3">
      <span className="text-xl w-8">{item.emoji}</span>
      <div>
        <p className="text-xs text-muted-foreground font-medium">{item.label}</p>
        <p className="text-sm font-bold text-foreground">{item.value}</p>
      </div>
    </div>
  )}
</div>

```

```
    ): null}}  
</div>  
  
<button onClick={() => navigate('/onboarding')}  
  className="w-full border-2 border-primary text-primary py-3 rounded-2xl font-bold">  
  ✎ Editar perfil
```

Continuação 3/3

```
    </button>  
  </div>  
</div>  
);  
}
```