

1 Documentação Técnica - PersonalMediCare

- 1.1 Índice
- 1.2 Visão Geral
 - 1.2.1 Objetivo Principal
- 1.3 Arquitetura Técnica
 - 1.3.1 Arquitetura Geral
 - 1.3.2 Padrões Arquiteturais
- 1.4 Tecnologias Utilizadas
 - 1.4.1 Frontend
 - 1.4.2 Backend e Serviços
 - 1.4.3 Inteligência Artificial
 - 1.4.4 Ferramentas de Desenvolvimento
- 1.5 Estrutura do Projeto
 - 1.5.1 Diretórios Principais
 - 1.5.2 Arquivos de Configuração
- 1.6 Funcionalidades Principais
 - 1.6.1 1. Gestão Familiar
 - 1.6.2 2. Controle de Medicamentos
 - 1.6.3 3. Farmacinha Digital
 - 1.6.4 4. Assistente Médica Virtual
 - 1.6.5 5. Análise de Documentos
 - 1.6.6 6. Relatórios e Documentos
- 1.7 Configuração e Deploy
 - 1.7.1 Variáveis de Ambiente
 - 1.7.2 Scripts de Build
 - 1.7.3 Perfis EAS Build
- 1.8 Segurança
 - 1.8.1 Autenticação
 - 1.8.2 Proteção de Dados
 - 1.8.3 Privacidade
- 1.9 APIs e Integrações
 - 1.9.1 Google Gemini AI
 - 1.9.2 Groq API
 - 1.9.3 Firebase Services
- 1.10 Banco de Dados
 - 1.10.1 Estrutura Firestore
 - 1.10.2 Índices e Consultas
- 1.11 Interface e UX
 - 1.11.1 Design System
 - 1.11.2 Navegação
 - 1.11.3 Responsividade
 - 1.11.4 Componentes Principais
- 1.12 Performance e Otimização
 - 1.12.1 Estratégias de Performance
 - 1.12.2 Monitoramento
- 1.13 Testes
 - 1.13.1 Estratégia de Testes
 - 1.13.2 Ferramentas
- 1.14 Manutenção e Evolução
 - 1.14.1 Versionamento

1.14.2 Roadmap
1.15 Conclusão

1 Documentação Técnica - PersonalMediCare

1.1 Índice

- 1. [Visão Geral](#)
 - 2. [Arquitetura Técnica](#)
 - 3. [Tecnologias Utilizadas](#)
 - 4. [Estrutura do Projeto](#)
 - 5. [Funcionalidades Principais](#)
 - 6. [Configuração e Deploy](#)
 - 7. [Segurança](#)
 - 8. [APIs e Integrações](#)
 - 9. [Banco de Dados](#)
 - 10. [Interface e UX](#)
-

1.2 Visão Geral

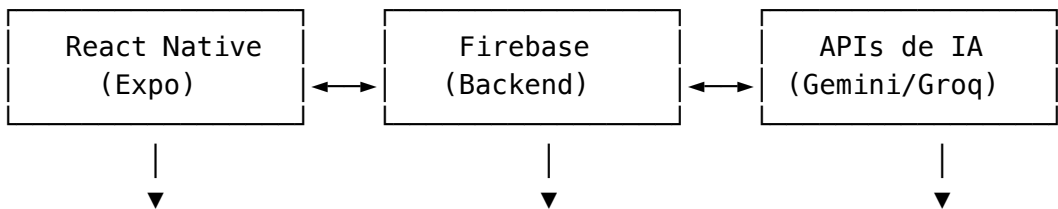
O **PersonalMediCare** é um aplicativo móvel desenvolvido em React Native com Expo, focado no gerenciamento inteligente de medicamentos para toda a família. O aplicativo combina tecnologias modernas de desenvolvimento mobile com inteligência artificial para oferecer uma experiência completa de cuidados médicos domiciliares.

1.2.1 Objetivo Principal

Facilitar o controle e gerenciamento de medicamentos familiares através de: - Lembretes inteligentes - Controle de estoque (Farmacinha Digital) - Assistente médica virtual com IA - Análise de documentos médicos - Relatórios para consultas

1.3 Arquitetura Técnica

1.3.1 Arquitetura Geral



iOS/Android
Nativo

Firestore
Storage

Webhooks
(N8N)

1.3.2 Padrões Arquiteturais

- **Context API:** Gerenciamento de estado global
 - **Component-Based:** Arquitetura baseada em componentes reutilizáveis
 - **Service Layer:** Camada de serviços para APIs externas
 - **Repository Pattern:** Abstração para acesso a dados
-

1.4 Tecnologias Utilizadas

1.4.1 Frontend

- **React Native:** Framework principal
- **Expo SDK 51:** Plataforma de desenvolvimento
- **TypeScript:** Linguagem de programação
- **React Navigation:** Navegação entre telas
- **Expo Router:** Roteamento baseado em arquivos

1.4.2 Backend e Serviços

- **Firebase Authentication:** Autenticação de usuários
- **Cloud Firestore:** Banco de dados NoSQL
- **Firebase Storage:** Armazenamento de arquivos
- **Firebase Functions:** Funções serverless

1.4.3 Inteligência Artificial

- **Google Gemini AI:** Análise de documentos e chat
- **Groq API:** Processamento de linguagem natural
- **N8N Webhooks:** Automação e integrações

1.4.4 Ferramentas de Desenvolvimento

- **EAS Build:** Sistema de build
 - **ESLint:** Linting de código
 - **Prettier:** Formatação de código
 - **TypeScript:** Tipagem estática
-

1.5 Estrutura do Projeto

1.5.1 Diretórios Principais

PersonalMediCare/	
├─ app/	# Rotas e telas (Expo Router)
│ └─ (auth)/	# Telas de autenticação
│ └─ (drawer)/	# Navegação drawer
│ └─ (tabs)/	# Navegação por abas
├─ components/	# Componentes reutilizáveis
├─ contexts/	# Contextos React
├─ services/	# Serviços e APIs
├─ screens/	# Telas principais
├─ db/	# Camada de dados
├─ types/	# Definições TypeScript
├─ utils/	# Utilitários
├─ constants/	# Constantes e configurações
├─ functions/	# Firebase Functions
└─ assets/	# Recursos estáticos

1.5.2 Arquivos de Configuração

- `app.json`: Configuração do Expo
 - `eas.json`: Configuração do EAS Build
 - `firebase.json`: Configuração do Firebase
 - `package.json`: Dependências e scripts
 - `tsconfig.json`: Configuração TypeScript
-

1.6 Funcionalidades Principais

1.6.1 1. Gestão Familiar

- **Perfis Múltiplos**: Gerenciamento de medicamentos para toda a família
- **Membros**: Adição e gerenciamento de membros familiares
- **Permissões**: Controle de acesso por membro

1.6.2 2. Controle de Medicamentos

- **Cadastro**: Registro completo de medicamentos
- **Dosagem**: Controle de doses e horários
- **Lembretes**: Notificações inteligentes
- **Histórico**: Acompanhamento de uso

1.6.3 3. Farmacinha Digital

- **Estoque**: Controle de quantidade disponível

- **Validade:** Alertas de vencimento
- **Compras:** Lista de medicamentos a comprar
- **Localização:** Organização por local de armazenamento

1.6.4 4. Assistente Médica Virtual

- **Chat IA:** Conversação natural sobre medicamentos
- **Informações:** Dados sobre medicamentos e interações
- **Orientações:** Dicas de uso e cuidados
- **Limitações:** Avisos sobre não substituir consultas médicas

1.6.5 5. Análise de Documentos

- **Upload:** Envio de receitas e exames
- **OCR:** Extração de texto de documentos
- **Análise IA:** Interpretação de conteúdo médico
- **Armazenamento:** Organização de documentos por membro

1.6.6 6. Relatórios e Documentos

- **Dossiê Médico:** Documento completo para consultas
 - **Histórico:** Relatórios de uso de medicamentos
 - **Estatísticas:** Análises de aderência ao tratamento
 - **Exportação:** Geração de PDFs
-

1.7 Configuração e Deploy

1.7.1 Variáveis de Ambiente

```
# Firebase
FIREBASE_API_KEY=
FIREBASE_AUTH_DOMAIN=
FIREBASE_PROJECT_ID=
FIREBASE_STORAGE_BUCKET=

# APIs de IA
GEMINI_API_KEY=
GROQ_API_KEY=

# Webhooks
N8N_WEBHOOK_URL=
```

1.7.2 Scripts de Build

```
{
  "scripts": {
    "start": "expo start",
```

```

    "android": "expo start --android",
    "ios": "expo start --ios",
    "web": "expo start --web",
    "build:android": "eas build --platform android",
    "build:ios": "eas build --platform ios",
    "submit:android": "eas submit --platform android",
    "submit:ios": "eas submit --platform ios"
  }
}

```

1.7.3 Perfis EAS Build

- **development:** Build para desenvolvimento
 - **preview:** Build para testes
 - **production:** Build para produção
-

1.8 Segurança

1.8.1 Autenticação

- **Firebase Auth:** Sistema robusto de autenticação
- **Múltiplos Provedores:** Email, Google, Apple
- **Tokens JWT:** Autenticação segura
- **Refresh Tokens:** Renovação automática

1.8.2 Proteção de Dados

- **Firestore Rules:** Regras de segurança no banco
- **Storage Rules:** Proteção de arquivos
- **Criptografia:** Dados sensíveis criptografados
- **HTTPS:** Comunicação segura

1.8.3 Privacidade

- **LGPD Compliance:** Conformidade com lei brasileira
 - **Consentimento:** Termos de uso e política de privacidade
 - **Anonimização:** Dados pessoais protegidos
 - **Auditoria:** Logs de acesso e modificações
-

1.9 APIs e Integrações

1.9.1 Google Gemini AI

```

// Configuração Gemini
const genAI = new GoogleGenerativeAI(GEMINI_API_KEY);

```

```
const model = genAI.getGenerativeModel({ model: "gemini-pro" });

// Uso para análise de medicamentos
const result = await model.generateContent(prompt);
```

1.9.2 Groq API

```
// Configuração Groq
const groq = new Groq({ apiKey: GROQ_API_KEY });

// Chat completion
const completion = await groq.chat.completions.create({
  messages: [{ role: "user", content: message }],
  model: "llama3-8b-8192"
});
```

1.9.3 Firebase Services

```
// Firestore
const db = getFirestore(app);

// Storage
const storage = getStorage(app);

// Authentication
const auth = getAuth(app);
```

1.10 Banco de Dados

1.10.1 Estrutura Firestore

1.10.1.1 Collections Principais

users

```
interface UserProfile {
  id: string;
  name: string;
  email: string;
  photoURL?: string;
  phone?: string;
  emergencyContact?: string;
  createdAt: Date;
  updatedAt: Date;
}
```

members

```
interface Member {  
  id: string;  
  userId: string;  
  name: string;  
  birthDate: Date;  
  relationship: string;  
  photoURL?: string;  
  allergies?: string[];  
  conditions?: string[];  
  createdAt: Date;  
}
```

medications

```
interface Medication {  
  id: string;  
  userId: string;  
  memberId: string;  
  name: string;  
  dosage: string;  
  frequency: string;  
  startDate: Date;  
  endDate?: Date;  
  instructions?: string;  
  sideEffects?: string[];  
  quantity: number;  
  expirationDate?: Date;  
  location?: string;  
}
```

treatments

```
interface Treatment {  
  id: string;  
  userId: string;  
  memberId: string;  
  name: string;  
  description?: string;  
  medications: string[];  
  startDate: Date;  
  endDate?: Date;  
  doctor?: string;  
  notes?: string;  
}
```

documents

```
interface Document {  
  id: string;  
  userId: string;  
  memberId?: string;  
  name: string;
```



```
type: 'prescription' | 'exam' | 'report' | 'other';
fileURL: string;
extractedText?: string;
analysis?: string;
uploadDate: Date;
tags?: string[];
}
```

1.10.2 Índices e Consultas

- **Composite Indexes:** Para consultas complexas
 - **Single Field Indexes:** Para consultas simples
 - **Array Contains:** Para busca em arrays
 - **Range Queries:** Para filtros por data
-

1.11 Interface e UX

1.11.1 Design System

- **Tema Claro/Escuro:** Suporte a múltiplos temas
- **Cores Consistentes:** Paleta de cores definida
- **Tipografia:** Fontes legíveis e hierarquia clara
- **Iconografia:** Ícones intuitivos e consistentes

1.11.2 Navegação

- **Drawer Navigation:** Menu lateral principal
- **Stack Navigation:** Navegação hierárquica
- **Tab Navigation:** Acesso rápido a funcionalidades
- **Deep Linking:** Links diretos para telas específicas

1.11.3 Responsividade

- **Adaptive Layout:** Adaptação a diferentes tamanhos
- **Safe Area:** Respeito às áreas seguras do dispositivo
- **Orientation:** Suporte a orientação portrait/landscape
- **Accessibility:** Suporte a recursos de acessibilidade

1.11.4 Componentes Principais

- **MedicationCard:** Cartão de medicamento
 - **MemberProfile:** Perfil de membro
 - **ChatBubble:** Bolha de conversa
 - **DocumentViewer:** Visualizador de documentos
 - **NotificationBell:** Sino de notificações
-

1.12 Performance e Otimização

1.12.1 Estratégias de Performance

- **Lazy Loading:** Carregamento sob demanda
- **Image Caching:** Cache de imagens
- **Data Pagination:** Paginação de dados
- **Offline Support:** Suporte offline básico

1.12.2 Monitoramento

- **Firebase Analytics:** Análise de uso
 - **Crashlytics:** Relatórios de crash
 - **Performance Monitoring:** Métricas de performance
 - **Error Tracking:** Rastreamento de erros
-

1.13 Testes

1.13.1 Estratégia de Testes

- **Unit Tests:** Testes unitários de funções
- **Integration Tests:** Testes de integração
- **E2E Tests:** Testes end-to-end
- **Manual Testing:** Testes manuais em dispositivos

1.13.2 Ferramentas

- **Jest:** Framework de testes
 - **React Native Testing Library:** Testes de componentes
 - **Detox:** Testes E2E
 - **Firebase Emulator:** Testes com Firebase local
-

1.14 Manutenção e Evolução

1.14.1 Versionamento

- **Semantic Versioning:** Versionamento semântico
- **Release Notes:** Notas de versão
- **Migration Guides:** Guias de migração
- **Backward Compatibility:** Compatibilidade retroativa

1.14.2 Roadmap

- **Novas Funcionalidades:** Planejamento de features

- **Melhorias de Performance:** Otimizações contínuas
 - **Atualizações de Dependências:** Manutenção de libs
 - **Feedback dos Usuários:** Incorporação de sugestões
-

1.15 Conclusão

O PersonalMediCare representa uma solução completa e moderna para o gerenciamento de medicamentos familiares, combinando tecnologias de ponta com uma experiência de usuário intuitiva. A arquitetura robusta e escalável permite futuras expansões e melhorias, mantendo sempre o foco na segurança e privacidade dos dados dos usuários.

A integração com serviços de IA oferece um diferencial competitivo significativo, proporcionando assistência inteligente e análise automatizada de documentos médicos. O projeto está bem estruturado para manutenção e evolução contínua, seguindo as melhores práticas de desenvolvimento mobile.

Versão da Documentação: 1.0

Data: Dezembro 2024

Autor: Equipe PersonalMediCare

Status: Produção