Uma imagem com Gráficos, design gráfico, símbolo, Tipo de letra

Os conteúdos gerados por IA podem estar incorretos.

Manual do Programador

Healthmate



20 de junho de 2025

Alexandre Prelhaz Inácio

2222107

Conteúdo

[INTRODUÇÃO E VISÃO GERAL 2](#_Toc201959674)

[Sobre o HealthMate 2](#_Toc201959675)

[Missão da Aplicação 2](#_Toc201959676)

[Funcionalidades Principais 2](#_Toc201959677)

[GESTÃO DE MEDICAÇÃO INTELIGENTE 2](#_Toc201959678)

[PLATAFORMA MÉDICA INTEGRADA 3](#_Toc201959679)

[MONITORIZAÇÃO AVANÇADA DE SAÚDE 3](#_Toc201959680)

[SISTEMA DE COMUNICAÇÃO MÉDICA 3](#_Toc201959681)

[DIÁRIO MÉDICO DIGITAL 4](#_Toc201959682)

[RELATÓRIOS E ANÁLISES AVANÇADAS 4](#_Toc201959683)

[Arquitectura Tecnológica Avançada 4](#_Toc201959684)

[Stack Tecnológico Completo 4](#_Toc201959685)

[Arquitectura de Sistema Distribuído 5](#_Toc201959686)

[Modelos de Utilizador e Casos de Uso 6](#_Toc201959687)

[Propostas de Valor Únicas 6](#_Toc201959688)

[Métricas de Desempenho e Objectivos 7](#_Toc201959689)

[Roadmap de Evolução 7](#_Toc201959690)

[ARQUITECTURA DO SISTEMA 9](#_Toc201959691)

[Arquitectura Multi-Camada Avançada 9](#_Toc201959692)

[Padrões Arquitecturais Implementados 10](#_Toc201959693)

[Visão Geral do Código do Projeto 11](#_Toc201959694)

[Arquitetura e Tecnologias 12](#_Toc201959695)

[Sistema de Autenticação 13](#_Toc201959696)

[13](#_Toc201959697)

[Serviços Principais 14](#_Toc201959698)

[Configuração Supabase 15](#_Toc201959699)

[Navegação e Estrutura de Ecrãs 16](#_Toc201959700)

[Componentes Inteligentes 17](#_Toc201959701)

[Integração com Dispositivos 18](#_Toc201959702)

[Padrões de Tratamento de Erros 18](#_Toc201959703)

[Comandos de Desenvolvimento Úteis 19](#_Toc201959704)

[CONCLUSÃO 20](#_Toc201959705)

# INTRODUÇÃO E VISÃO GERAL

## Sobre o HealthMate

O **HealthMate** é uma aplicação móvel abrangente e inovadora de gestão de saúde pessoal, desenvolvida com **React Native** e **Expo SDK 52**. Esta solução digital representa um ecossistema completo de saúde digital que integra múltiplas funcionalidades numa única plataforma.

## Missão da Aplicação

Democratizar o acesso à gestão de saúde pessoal através de tecnologia avançada, permitindo aos utilizadores tomarem controlo total da sua saúde com ferramentas intuitivas e dados precisos.

Funcionalidades Principais

### GESTÃO DE MEDICAÇÃO INTELIGENTE

- Sistema avançado de agendamento de medicamentos com algoritmos de recorrência

- Notificações inteligentes baseadas em horários personalizados

- Cálculo automático de próximas doses e gestão de stock

- Histórico completo de aderência à medicação

- Alertas para medicamentos em falta ou expirados

### PLATAFORMA MÉDICA INTEGRADA

- Agendamento de consultas com médicos registados

- Sistema de consultas personalizadas com médicos externos

- Dashboard médico para profissionais de saúde

- Gestão de disponibilidade médica avançada

- Sistema de avaliação e feedback de médicos

### MONITORIZAÇÃO AVANÇADA DE SAÚDE

- Integração nativa com Samsung Health e Google Health Connect

- Monitorização em tempo real de sinais vitais (frequência cardíaca, SpO2, tensão arterial)

- Acompanhamento de composição corporal (peso, IMC, gordura corporal)

- Análise de padrões de sono e qualidade do descanso

- Tracking de actividade física e calorias queimadas

- Gestão de hidratação e nutrição

### SISTEMA DE COMUNICAÇÃO MÉDICA

- Chat directo entre pacientes e profissionais de saúde

- Sistema de mensagens seguro e encriptado

- Partilha de documentos médicos e exames

- Telemedicina básica com suporte a videoconferência (em desenvolvimento)

### DIÁRIO MÉDICO DIGITAL

- Registo detalhado de sintomas e condições de saúde

- Tracking de humor e bem-estar mental

- Anotações médicas com suporte a voz e imagem

- Histórico cronológico de eventos de saúde

### RELATÓRIOS E ANÁLISES AVANÇADAS

- Dashboards interactivos com visualizações gráficas

- Relatórios exportáveis para profissionais de saúde

- Análise de tendências e padrões de saúde

- Alertas preditivos baseados em dados históricos

Arquitectura Tecnológica Avançada

### Stack Tecnológico Completo

**FRONTEND MÓVEL**

- **React Native 0.76.6**: Framework principal para desenvolvimento multiplataforma

- **Expo SDK 52:** Plataforma de desenvolvimento com ferramentas avançadas

- **TypeScript: Tipagem** estática para maior robustez do código

- **React Navigation 7**: Sistema de navegação avançado

- **React Native Paper**: Biblioteca de componentes Material Design

- **React Native Reanimated 3**: Animações performáticas

- **Expo Linear Gradient**: Gradientes avançados para interface moderna

**BACKEND E INFRAESTRUTURA**

- **Supabase:** Backend-as-a-Service completo

- **PostgreSQL**: Base de dados relacional robusta

- **Row Level Security (RLS):** Segurança avançada ao nível da linha

- **Real-time Subscriptions:** Actualizações em tempo real

- **Functions**: Processamento serverless

- **Storage**: Armazenamento seguro de ficheiros

**INTEGRAÇÃO E CONECTIVIDADE**

- **Expo Health Connect**: Integração nativa com dados de saúde Android

- **Samsung Health Service**: API específica para dispositivos Samsung

**- React Native BLE**: Conectividade Bluetooth Low Energy

- **Expo Notifications:** Sistema completo de notificações

- **Expo** Calendar: Integração com calendário nativo

- **Expo Local Authentication:** Biometria (Face ID/Touch ID)

**ARMAZENAMENTO E CACHE**

- **AsyncStorage**: Armazenamento persistente local

- **Expo** **SecureStore**: Armazenamento seguro de credenciais

- **SQLite**: Base de dados local para cache

Arquitectura de Sistema Distribuído

┌─────────────────────────────────────────────────────────────┐

│ CAMADA DE APRESENTAÇÃO │

│ (React Native + Expo SDK 52) │

├────────────────────────────────────────────────────────────┤

│ CAMADA DE NAVEGAÇÃO │

│ (React Navigation + Context Providers) │

├────────────────────────────────────────────────────────────┤

│ CAMADA DE LÓGICA DE NEGÓCIO │

│ (Services + Custom Hooks + State Management) │

├────────────────────────────────────────────────────────────┤

│ CAMADA DE INTEGRAÇÃO │

│ (Health APIs + Bluetooth + Notifications + Camera) │

├────────────────────────────────────────────────────────────┤

│ CAMADA DE COMUNICAÇÃO │

│ (HTTP Client + WebSocket + Real-time) │

├────────────────────────────────────────────────────────────┤

│ CAMADA DE DADOS │

│ (Supabase PostgreSQL + AsyncStorage + Cache) │

├────────────────────────────────────────────────────────────┤

│ INFRAESTRUTURA EXTERNA │

│ (Samsung Health + Google Health + External APIs) │

└──────────────────────────────────────────────────────────────┘

Modelos de Utilizador e Casos de Uso

### Propostas de Valor Únicas

**Diferenciadores Técnicos**

INTEGRAÇÃO NATIVA AVANÇADA

* Conectividade directa com mais de 50 tipos de dispositivos de saúde
* Sincronização automática de dados sem intervenção do utilizador
* Suporte para protocolos médicos padronizados (HL7 FHIR em desenvolvimento)

INTELIGÊNCIA ARTIFICIAL INTEGRADA

* Algoritmos de detecção de padrões de saúde
* Previsão de episódios de saúde baseada em dados históricos
* Recomendações personalizadas de medicação e estilo de vida

SEGURANÇA E PRIVACIDADE MÉDICA

* Encriptação end-to-end para todos os dados médicos
* Compliance com GDPR e HIPAA
* Auditoria completa de acesso a dados sensíveis

EXPERIÊNCIA DE UTILIZADOR SUPERIOR

* Interface adaptativa baseada no perfil do utilizador
* Acessibilidade completa para utilizadores com limitações
* Suporte multi-idioma com localização cultural

Métricas de Desempenho e Objectivos

KPIs Técnicos

* Tempo de resposta médio: < 300ms para operações locais
* Sincronização de dados: < 5 segundos para dados críticos
* Disponibilidade: 99.9% uptime
* Crash rate: < 0.1% das sessões
* Battery usage: < 2% por dia de utilização normal

KPIs de Utilizador

* Aderência à medicação: Melhoria de 40% em utilizadores activos
* Satisfação médica: Rating médio > 4.5/5
* Retenção de utilizadores: > 80% após 30 dias
* Engagement diário: > 5 minutos de utilização média

### Roadmap de Evolução

Versão Actual (1.1.0)

* Dashboard completo de saúde
* Gestão avançada de medicação
* Sistema de consultas médicas
* Integração com dispositivos vestíveis

Próximas Versões (1.2.0 - 2.0.0)

* IA Médica: Assistente virtual com processamento de linguagem natural
* Telemedicina: Videoconsultas integradas com partilha de ecrã
* Blockchain: Sistema descentralizado de registos médicos
* IoT Avançado: Integração com sensores ambientais e casa inteligente
* Realidade Aumentada: Visualização de dados de saúde em AR

# ARQUITECTURA DO SISTEMA

Arquitectura Multi-Camada Avançada

O HealthMate implementa uma arquitectura robusta e escalável baseada em múltiplas camadas especializadas, cada uma com responsabilidades bem definidas e interfaces claras.

Diagrama Arquitectural Completo

Camada de Apresentação

* Screens (50+ screens)
* Components (Reusable UI)
* Navigation (Stack/Tab/Drawer)

Camada de Gestão de Estado

* React Context (Global State)
* AsyncStorage (Persistence)
* Custom Hooks (Business Logic)

Camada de Serviços

* User Services (Auth, Profile)
* Health Services (Vitals, Devices)
* Medical Services (Appointments, Docs)

Camada de Integração

* Health Connect (Android)
* Samsung Health (Native)
* BLE Devices (Scales, Monitors)

Camada de Comunicação

* HTTP/REST API (Supabase)
* WebSockets (Live Updates)
* Real-time Sync (Notifications)

Camada de Dados

* PostgreSQL (Supabase)
* Local Storage (AsyncStorage)
* File Storage (Images, Documents)

Infraestrutura Externa

* Cloud Services (Push, Storage)
* Device APIs (Health, Camera)
* Third-party APIs (Maps, Analytics)

Padrões Arquitecturais Implementados

Clean Architecture Principles

Separação de Responsabilidades

* Presentation Layer: Interfaces de utilizador e lógica de apresentação
* Business Layer: Regras de negócio e casos de uso
* Data Layer: Acesso a dados e persistência
* Infrastructure Layer: Integrações externas e frameworks

Dependency Inversion

Interfaces abstratas para inversão de dependências

interface IHealthDataRepository

* getVitalSigns(userId: string): Promise<VitalSigns[]>
* saveVitalSigns(data: VitalSigns): Promise<void>

interface INotificationService

* scheduleNotification(notification: NotificationData): Promise<void>
* cancelNotification(id: string): Promise<void>

Implementações concretas

SupabaseHealthRepository implementa IHealthDataRepository (implementação específica do Supabase)  
ExpoNotificationService implementa INotificationService (implementação específica do Expo)

Domain-Driven Design (DDD)

Bounded Contexts

* User Management: Autenticação, perfis, preferências
* Health Monitoring: Sinais vitais, dispositivos, métricas
* Medical Care: Consultas, médicos, tratamentos
* Medication Management: Medicamentos, horários, adherência
* Communication: Chat, notificações, partilha de dados

# Visão Geral do Código do Projeto

O **HealthMate** é uma aplicação móvel abrangente de gestão de saúde desenvolvida em React Native com Expo. A aplicação foi concebida para centralizar todos os aspetos relacionados com a saúde do utilizador numa única plataforma intuitiva.

**Funcionalidades Principais:**

* **Gestão de Medicamentos**: Sistema inteligente de lembretes e acompanhamento
* **Consultas Médicas**: Plataforma integrada para marcação e gestão de consultas
* **Monitorização de Saúde**: Acompanhamento de sinais vitais e indicadores de saúde
* **Diário Médico**: Registo detalhado de sintomas e condições de saúde
* **Comunicação**: Sistema de chat integrado para comunicação com profissionais de saúde
* **Relatórios**: Análise e visualização de dados de saúde

**Porquê é Importante:**

Esta aplicação resolve um problema real na gestão pessoal de saúde, oferecendo uma solução centralizada que melhora a adesão aos tratamentos e facilita a comunicação com profissionais de saúde.

# Arquitetura e Tecnologias

// package.json - Dependências principais

{

"react-native": "0.76.6",

"@expo/vector-icons": "^14.0.4",

"@react-navigation/native": "^6.1.18",

"@supabase/supabase-js": "^2.39.7",

"expo": "~52.0.11"

}

**Arquitetura em Camadas:**

1. **Camada de Apresentação**: Componentes React Native
2. **Camada de Lógica de Negócio**: Serviços e contextos
3. **Camada de Dados**: Supabase (PostgreSQL)
4. **Camada de Integração**: APIs externas e dispositivos

**Porquê Esta Arquitetura:**

* **Separação de Responsabilidades**: Facilita manutenção e testes
* **Escalabilidade**: Permite adicionar novas funcionalidades sem afetar o código existente
* **Reutilização**: Componentes e serviços podem ser reutilizados em diferentes partes da aplicação

# Sistema de Autenticação

# Uma imagem com texto, captura de ecrã Os conteúdos gerados por IA podem estar incorretos.

**Porquê o AuthContext é Crucial:**

* **Estado Global**: Mantém o estado de autenticação acessível em toda a aplicação
* **Persistência**: Guarda dados no AsyncStorage para manter sessão ativa
* **Segurança**: Centraliza a lógica de autenticação num local controlado
* **Singleton Pattern**: DataUser garante uma única instância dos dados do utilizador

# Serviços Principais



**Porquê os Serviços São Importantes:**

* **Separação de Lógica**: Mantém a lógica de negócio separada dos componentes UI
* **Reutilização**: Podem ser utilizados por múltiplos componentes
* **Testabilidade**: Mais fácil de testar isoladamente
* **Manutenibilidade**: Mudanças na lógica ficam centralizadas

Uma imagem com texto, captura de ecrã

Os conteúdos gerados por IA podem estar incorretos.

# Configuração Supabase

// supabase.js

import { createClient } from '@supabase/supabase-js';

const supabaseUrl = 'https://your-project.supabase.co';

const supabaseAnonKey = 'your-anon-key';

export const supabase = createClient(supabaseUrl, supabaseAnonKey, {

auth: {

storage: AsyncStorage,

autoRefreshToken: true,

persistSession: true,

detectSessionInUrl: false,

},

});

**Porquê Supabase:**

* **Backend-as-a-Service**: Reduz complexidade de infraestrutura
* **Real-time**: Atualizações em tempo real para chat e notificações
* **Autenticação Integrada**: Sistema robusto de autenticação
* **PostgreSQL**: Base de dados relacional poderosa com funcionalidades avançadas

# Navegação e Estrutura de Ecrãs

**Porquê Esta Estrutura:**

* Uma imagem com texto, captura de ecrã, Tipo de letra

  Os conteúdos gerados por IA podem estar incorretos.**Hierarquia Clara**: Organização lógica dos ecrãs
* **Navegação Intuitiva**: Fluxo natural para o utilizador
* **Modularidade**: Cada ecrã é independente mas conectado

# Componentes Inteligentes

Uma imagem com texto, captura de ecrã

Os conteúdos gerados por IA podem estar incorretos.**Porquê Este Componente é Crítico:**

* **Algoritmo Inteligente**: Calcula automaticamente próximas tomas
* **Interface Intuitiva**: Permite ao utilizador marcar tomas facilmente
* **Atualizações em Tempo Real**: Atualiza automaticamente a cada minuto
* **Persistência**: Regista todas as tomas na base de dados

Uma imagem com texto, captura de ecrã, software

Os conteúdos gerados por IA podem estar incorretos.

# Integração com Dispositivos

**Porquê a Integração com Dispositivos:**

* Uma imagem com texto, captura de ecrã

  Os conteúdos gerados por IA podem estar incorretos.**Dados Precisos**: Obtém dados diretamente dos sensores dos dispositivos
* **Automatização**: Reduz a necessidade de entrada manual de dados
* **Histórico Completo**: Mantém registo contínuo dos dados de saúde
* **Análise Avançada**: Permite identificar padrões e tendências

# Uma imagem com texto, captura de ecrã Os conteúdos gerados por IA podem estar incorretos.Padrões de Tratamento de Erros

**Porquê o Tratamento de Erros é Vital:**

* **Experiência do Utilizador**: Mensagens claras em vez de erros técnicos
* **Debugging**: Logs detalhados para resolução de problemas
* **Robustez**: Aplicação continua a funcionar mesmo com erros
* **Confiança**: Utilizadores confiam mais numa aplicação que trata erros bem

# Comandos de Desenvolvimento Úteis

# Instalação de dependências

npm install

# Desenvolvimento

npx expo start

# Executar no Android

npx expo run:android

# Executar no iOS

npx expo run:ios

# Build para produção

npx eas build --platform all

# Actualizar dependências

npx expo install --fix

# Limpar cache

npx expo start --clear

# Gerar APK local

npx eas build --platform android –local

{

"scripts": {

"start": "expo start",

"android": "expo run:android",

"ios": "expo run:ios",

"web": "expo start --web",

"build:android": "eas build --platform android",

"build:ios": "eas build --platform ios",

"preview": "eas build --profile preview",

"lint": "eslint .",

"test": "jest"

}

}

# CONCLUSÃO

O **HealthMate** é uma aplicação robusta e escalável que segue as melhores práticas de desenvolvimento React Native. Este manual fornece uma base sólida para programadores que trabalhem no projecto, permitindo:

- **Compreensão rápida** da arquitectura e estrutura

- **Desenvolvimento** **eficiente** com padrões estabelecidos

- **Manutenção facilitada** com documentação detalhada

- **Evolução controlada** com boas práticas implementadas

Esta aplicação não só resolve problemas reais dos utilizadores, mas também estabelece uma base tecnológica que pode evoluir com as necessidades futuras da área da saúde digital.

**Recursos Adicionais**

- Documentação Expo\* <https://docs.expo.dev>

- Documentação React Native: <https://reactnative.dev>

- Documentação Supabase: <https://supabase.com/docs>

- React Navigation: <https://reactnavigation.org>