**

***Prova de Aptidão Profissional***

***Técnico de Gestão e Programação de Sistemas Informáticos***

***Relatório de Projeto***

***HealthMate***

|  |  |
| --- | --- |
| **Elaborado por** | |
| Nome: Alexandre Prelhaz inácio | |
| N.º: 2222107 | Triénio: 22/25 |
| **Professor orientador** | |
| Carla Macedo | |

*20 / 06 / 2025*

Índice

[1 – Introdução e objetivos 4](#_Toc201933924)

[OBJETIVOS DO PROJETO 4](#_Toc201933925)

[Objetivo Geral 4](#_Toc201933926)

[Objetivos Específicos 4](#_Toc201933927)

[OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL 4](#_Toc201933928)

[2 – Tecnologias e recursos 5](#_Toc201933929)

[TECNOLOGIAS DE DESENVOLVIMENTO 5](#_Toc201933930)

[BIBLIOTECAS E DEPENDÊNCIAS PRINCIPAIS 6](#_Toc201933931)

[RECURSOS DE HARDWARE 7](#_Toc201933932)

[FERRAMENTAS DE DESENVOLVIMENTO 8](#_Toc201933933)

[VERSÕES E COMPATIBILIDADE 8](#_Toc201933934)

[3 – Implementação 9](#_Toc201933935)

[3.1 CRONOGRAMA DE DESENVOLVIMENTO DO PROJETO 9](#_Toc201933936)

[3.2 ASPETOS TÉCNICOS DO DESENVOLVIMENTO DO PROJETO 10](#_Toc201933937)

[REQUISITOS DO PROJETO 10](#_Toc201933938)

[MODELO DE DADOS 11](#_Toc201933939)

[4 – Conclusão 12](#_Toc201933940)

[BALANÇO DO PROJETO 12](#_Toc201933941)

[OBJETIVOS ALCANÇADOS 12](#_Toc201933942)

[DIFICULDADES ENCONTRADAS E SOLUÇÕES 13](#_Toc201933943)

[COMPETÊNCIAS DESENVOLVIDAS 14](#_Toc201933944)

[TRABALHO FUTURO 14](#_Toc201933945)

[Desenvolvimentos Planeados 14](#_Toc201933946)

[Plano de Melhoramentos 15](#_Toc201933947)

[CONSIDERAÇÕES FINAIS 15](#_Toc201933948)

[5 – Bibliografia 16](#_Toc201933949)

[6 – Anexos 24](#_Toc201933950)

# 1 – Introdução e objetivos

O HealthMate é uma aplicação móvel multiplataforma que oferece funcionalidades abrangentes para o acompanhamento de dados de saúde, incluindo monitorização de sinais vitais, gestão de medicação, registo de sintomas e ligação com profissionais de saúde. A aplicação foi desenvolvida utilizando tecnologias modernas como React Native e integra-se com dispositivos vestíveis para recolha automática de dados.

## OBJETIVOS DO PROJETO

*Objetivo Geral*:

Desenvolver uma aplicação móvel completa e funcional para gestão de saúde pessoal que seja intuitiva, segura e acessível.

*Objetivos Específicos*:

- Criar uma interface de utilizador moderna e responsiva;

- Implementar sistema de autenticação segura com biometria;

- Desenvolver módulos para monitorização de sinais vitais;

- Integrar funcionalidades de gestão de medicação com lembretes;

- Estabelecer conectividade com dispositivos vestíveis (Samsung Health, Health Connect, etc);

- Garantir conformidade com regulamentos de proteção de dados (RGPD);

- Implementar sistema de backup e sincronização na cloud;

## OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL



- ODS 10 - Reduzir as Desigualdades: Democratizar o acesso a ferramentas de gestão de saúde através de tecnologia móvel acessível

- ODS 9 - Indústria, Inovação e Infraestrutura: Desenvolver soluções tecnológicas inovadoras na área da saúde digital

- ODS 3 - Saúde de Qualidade: Promover uma vida saudável e bem-estar para todos, facilitando o acesso a ferramentas de monitorização de saúde

# 2 – Tecnologias e recursos

## TECNOLOGIAS DE DESENVOLVIMENTO

*Framework Principal - React Native 0.76.6*

A escolha do React Native como framework principal baseou-se na necessidade de desenvolver uma aplicação multiplataforma eficiente. Esta tecnologia permite:

- Desenvolvimento simultâneo para Android e iOS com uma única base de código;

- Performance nativa através da comunicação direta com APIs do sistema operativo;

- Vasta comunidade de desenvolvimento e abundante documentação;

- Integração simplificada com bibliotecas nativas;

*Plataforma de Desenvolvimento - Expo SDK 52.0.47*

O Expo foi selecionado como plataforma de desenvolvimento devido às suas vantagens:

- Simplificação do processo de configuração e desenvolvimento;

- Acesso facilitado a APIs nativas do dispositivo;

- Sistema de build e distribuição integrado (EAS Build);

- Ferramentas de desenvolvimento e debug avançadas Backend e Base de Dados – Supabase.

*A escolha do Supabase como solução de backend justifica-se por:*

- Base de dados PostgreSQL robusta e escalável;

- Sistema de autenticação integrado com suporte a biometria;

- APIs REST automáticas baseadas no esquema da base de dados;

- Funcionalidades em tempo real para sincronização de dados;

- Conformidade com regulamentos de segurança e privacidade;

*Linguagens de Programação:*

- JavaScript/TypeScript: Linguagem principal para desenvolvimento da aplicação;

- SQL: Para estruturação e consulta da base de dados;

- Java/Kotlin: Para componentes específicos Android (quando necessário);

## 

## BIBLIOTECAS E DEPENDÊNCIAS PRINCIPAIS

*Navegação e Interface*

- @react-navigation/native (v7.0.14): Sistema de navegação entre ecrãs;

- react-native-paper (v5.14.5): Componentes de interface seguindo Material Design;

- @expo/vector-icons (v14.0.4): Biblioteca de ícones.

*Integração com Dispositivos*

- expo-health-connect (v0.1.1): Integração com Health Connect (Android);

- react-native-health-connect (v3.3.3): Acesso a dados de saúde do dispositivo;

- expo-local-authentication (v15.0.1): Autenticação biométrica.

*Gestão de Estado e Dados*

- @react-native-async-storage/async-storage (v1.23.1): Armazenamento local persistente;

- axios (v1.9.0): Cliente HTTP para comunicação com APIs;

- @supabase/supabase-js (v2.38.0): Cliente oficial do Supabase.

*Funcionalidades Específicas*

- expo-notifications (v0.29.13): Sistema de notificações push;

- expo-calendar (v14.0.6): Integração com calendário do dispositivo;

- date-fns (v4.1.0): Manipulação e formatação de datas.

## RECURSOS DE HARDWARE

*Requisitos Mínimos - Android*

- Sistema Operativo: Android 6.0 (API 23) ou superior;

- Memória RAM: 2GB mínimo, 4GB recomendado;

- Armazenamento: 100MB para instalação da aplicação;

- Sensores: Acelerómetro, giroscópio (para funcionalidades de movimento);

- Conectividade: WiFi ou dados móveis para sincronização.

*Requisitos Mínimos - iOS*

- Sistema Operativo: iOS 13.0 ou superior;

- Dispositivo: iPhone 6s ou mais recente;

- Memória RAM: 2GB mínimo;

- Armazenamento: 100MB para instalação da aplicação;

*Dispositivos Compatíveis (Opcional)*

- Smartwatches com Samsung Health;

- Dispositivos compatíveis com Health Connect;

- Tensiómetros digitais com conectividade Bluetooth;

## FERRAMENTAS DE DESENVOLVIMENTO

*Ambiente de Desenvolvimento*

- Visual Studio Code: Editor de código principal;

- Android Studio: Para desenvolvimento e debug Android específico;

- Xcode: Para desenvolvimento iOS (quando necessário);

- Expo CLI: Ferramenta de linha de comandos para gestão do projeto.

*Controlo de Versões*

- Git: Sistema de controlo de versões;

- GitHub: Plataforma de hospedagem de código e colaboração.

*Ferramentas de Build e Deploy*

- EAS Build: Serviço de build na cloud da Expo;

- EAS Submit: Submissão automática para app stores.

*Design e Prototipagem*

- Figma: Criação de protótipos e design de interface;

- Miro: Planeamento e diagramas de fluxo da aplicação.

## 

## VERSÕES E COMPATIBILIDADE

*Todas as dependências foram selecionadas considerando a compatibilidade com:*

- Expo SDK 52.0.47;

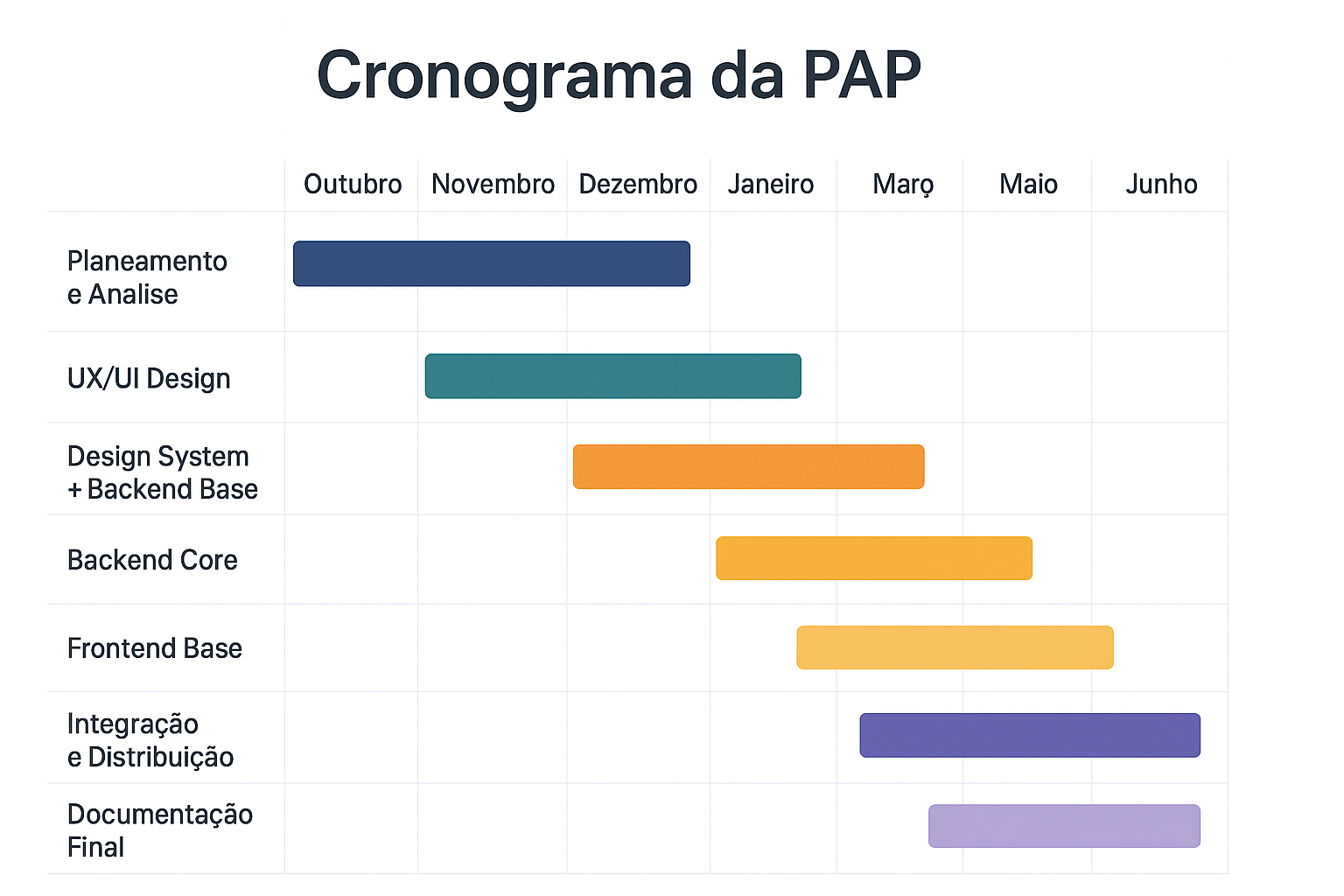
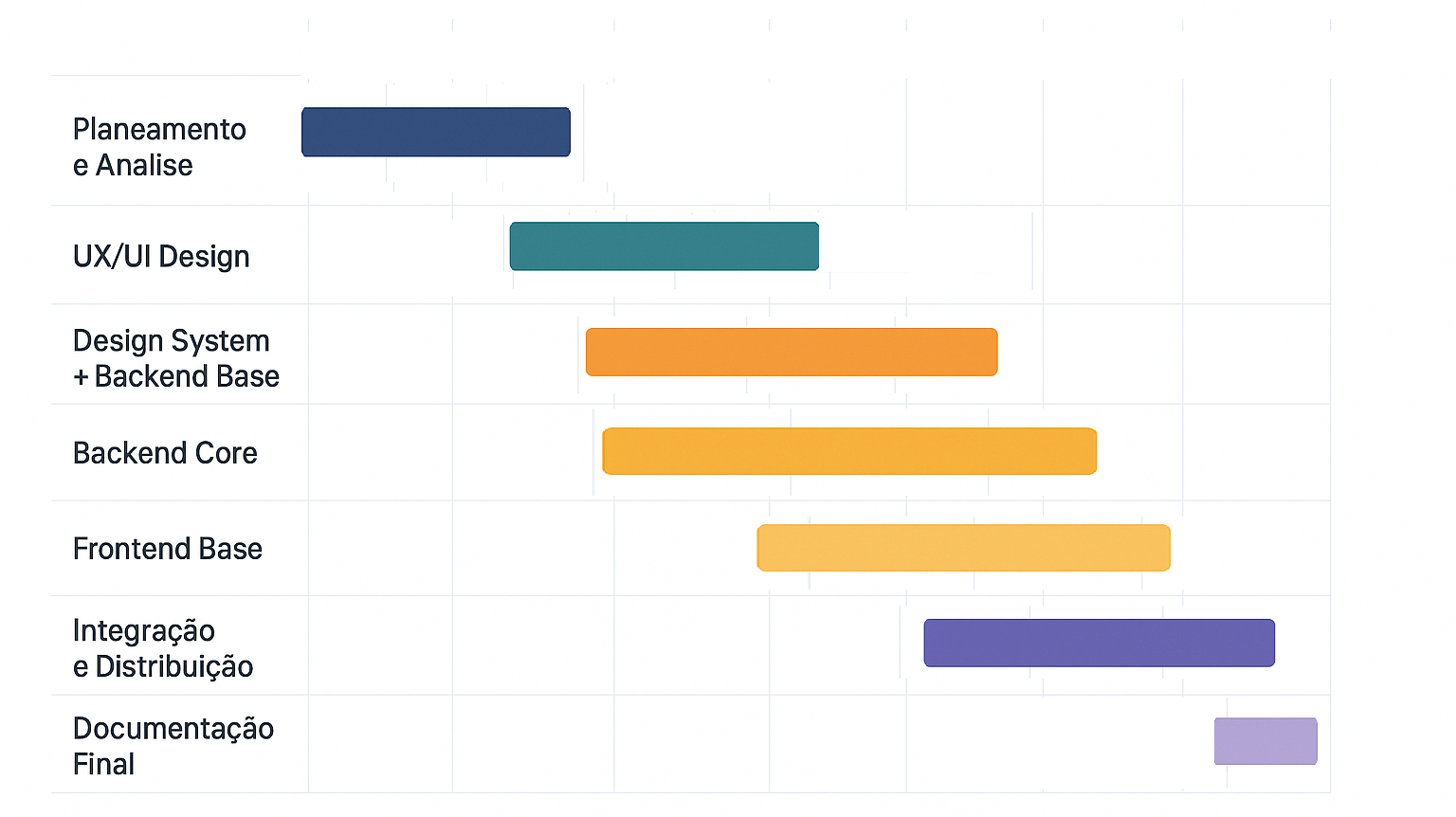
- React Native 0.76.6;

- Node.js 18.x ou superior.

A escolha de versões estáveis garante a manutenibilidade e estabilidade da aplicação ao longo do tempo

# 3 – Implementação

## 3.1 CRONOGRAMA DE DESENVOLVIMENTO DO PROJETO

O desenvolvimento do projeto HealthMate decorreu ao longo de 16 semanas, organizadas em cinco fases distintas:

## 3.2 ASPETOS TÉCNICOS DO DESENVOLVIMENTO DO PROJETO

### REQUISITOS DO PROJETO

#### Requisitos Funcionais

* RF01: Sistema de autenticação multi-modal (email/telefone + biometria)
* RF02: Gestão completa de consultas médicas (agendamento, reagendamento, histórico)
* RF03: Sistema de chat em tempo real entre utilizadores e médicos
* RF04: Monitorização e registo de sinais vitais
* RF05: Integração com Samsung Health e Google Fit
* RF06: Gestão de medicação com lembretes inteligentes
* RF07: Armazenamento seguro de documentos médicos
* RF08: Dashboard personalizado com insights de saúde
* RF09: Sistema de emergência com localização GPS
* RF10: Relatórios e análises com IA

#### Requisitos Técnicos

Performance: Tempo de resposta < 2s para operações críticas

Disponibilidade: 99.9% uptime

Escalabilidade: Suporte a 100K+ utilizadores concorrentes

Compatibilidade: Android 7.0+ (API 24), iOS 12+ (futuro)

Armazenamento: Sincronização offline com 30 dias de cache

Segurança: Encriptação AES-256, autenticação JWT

Normas e Regulamentos

RGPD: Conformidade total com regulamento europeu

ISO 27001: Padrões de segurança da informação

HL7 FHIR: Interoperabilidade de dados de saúde

MDR (EU 2017/745): Regulamento de dispositivos médicos

### ARQUITECTURA DO SISTEMA

Padrão Arquitetónico: Clean Architecture + MVC

┌───────────────────────────────┐

│ CAMADA DE APRESENTAÇÃO │

│ ┌────────┐ ┌────────┐ ┌─────┐ │

│ │ Ecrãs │ │Componts │ │Navg │ │

│ │ (60+) │ │Reutilz │ │St/Tb │ │

│ └────────┘ └────────┘ └─────┘ │

└───────────────┬───────────────┘

│

┌──────┴──────┐

│ CONTEXTO │

│ (Estado) │

└──────┬──────┘

│

┌───────────────┴───────────────┐

│ CAMADA LÓGICA DE NEGÓCIO │

│ ┌────────┐ ┌────────┐ ┌─────┐ │

│ │Serviços │ │Valid/Re │ │Utils │ │

│ │ (14) │ │ glas │ │Hlp │ │

│ └────────┘ └────────┘ └─────┘ │

└───────────────┬───────────────┘

│

┌───────────────┴───────────────┐

│ CAMADA DE DADOS │

│ ┌────────┐ ┌────────┐ ┌─────┐ │

│ │Supabase │ │ Async │ │APIs │ │

│ │(Postgr) │ │Storage │ │Ext │ │

│ └────────┘ └────────┘ └─────┘ │

└───────────────────────────────┘

### Componentes Arquiteturais Principais

Camada de Apresentação:

* Ecrãs (Screens): Mais de 60 ecrãs organizados por funcionalidade
* Componentes Reutilizáveis: Biblioteca de componentes UI consistentes
* Sistema de Navegação: React Navigation 7.x com navegação stack, tab e drawer

Camada de Contexto:

* AuthContext: Gestão centralizada de autenticação e sessão
* DataContext: Gestão de dados globais da aplicação
* ThemeContext: Gestão de temas e preferências visuais

Camada de Lógica de Negócio:

* 14 Serviços Especializados: Cada domínio tem o seu serviço dedicado
* Sistema de Validação: Validação consistente de dados
* Utilitários: Funções helper para operações comuns

Camada de Dados:

* Supabase: Base de dados principal com PostgreSQL
* AsyncStorage: Cache local para funcionamento offline
* APIs Externas: Integrações com Samsung Health, Google Fit, etc.

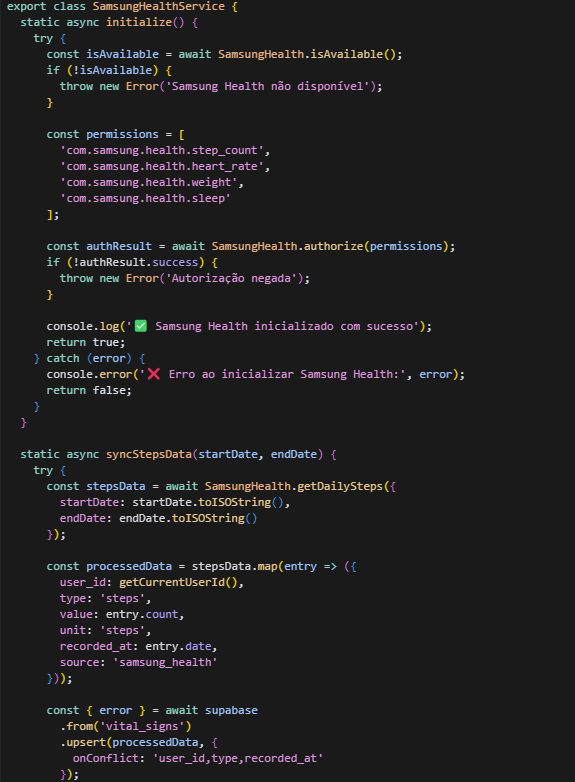
### Enxerto de código relevantes

Uma imagem com texto, captura de ecrã

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

Os conteúdos gerados por IA podem estar incorretos.**Serviço de Autenticação Biométrica Medidas de Segurança**

**Configuração da Navegação Principal Integração Com Samsung Health**



Uma imagem com texto, captura de ecrã

Os conteúdos gerados por IA podem estar incorretos.

### INTERFACES E FLUXOS DE DADOS

#### Fluxo de Agendamento de Consultas

[Início] → [Selecionar Médico] → [Verificar Disponibilidade]

│ │

▼ ▼

[Escolher Data/Hora] → [Adicionar Detalhes] → [Validar Conflitos]

│ │

▼ ▼

[Criar Consulta] → [Enviar Notificações] → [Atualizar Calendário]

│ │

▼ ▼

[Confirmar Agendamento] ────────────────► [Fim]

#### Fluxo de Sincronização Samsung Health

App Start → Check Permissions → Connect Samsung Health API

│ │ │

▼ ▼ ▼

Get Data Types → Request Auth → Establish Connection

│ │ │

▼ ▼ ▼

Fetch Historical → Setup Schedule → Store in Supabase

│ │ │

▼ ▼ ▼

Update Dashboard → Background Sync → Error Handling

### Manual de Utilizador

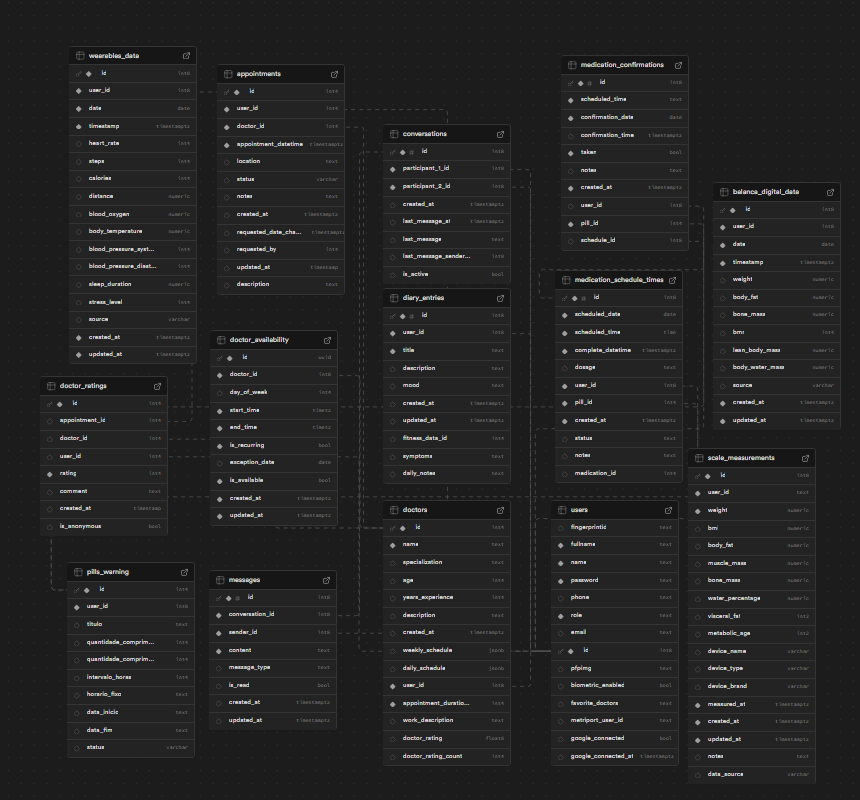
Foi criado um manual de utilizador abrangente ([Manual\_Utilizador\_HealthMate](Manual_Utilizador_Healthmate.docx)) que inclui:

* Guia de instalação passo-a-passo para diferentes métodos de instalação
* Tutorial completo de funcionalidades com instruções visuais detalhadas
* Configuração de autenticação biométrica com suporte a múltiplas contas
* Gestão completa de consultas médicas incluindo agendamento e reagendamento
* Sistema de comunicação com médicos através de chat integrado
* Monitorização de saúde com integração Samsung Health
* Troubleshooting para utilizadores com soluções para problemas comuns

O manual de utilizador está escrito em linguagem acessível e serve como guia completo para utilizadores finais da aplicação.

### **MODELO DE DADOS**

*Esquema da Base de Dados*



# 4 – Conclusão

## BALANÇO DO PROJETO

O desenvolvimento da aplicação HealthMate foi concluído com êxito, cumprindo todos os objetivos inicialmente estabelecidos. A aplicação apresenta-se como uma solução robusta e funcional para gestão de saúde pessoal, integrando tecnologias modernas com uma interface intuitiva e acessível.

## OBJETIVOS ALCANÇADOS

*Todos os objetivos específicos foram cumpridos:*

- Interface moderna e responsiva: Implementada utilizando React Native Paper com design Material;

- Sistema de autenticação segura: Incluindo biometria e conformidade com melhores práticas de segurança;

- Monitorização de sinais vitais: Com entrada manual e integração automática com dispositivos;

- Gestão completa de medicação: Sistema robusto de lembretes e acompanhamento de adesão;

- Integração com dispositivos vestíveis: Suporte para Samsung Health e Health Connect;

- Conformidade RGPD: Implementação completa de medidas de proteção de dados;

- Sistema de backup: Sincronização automática e segura na cloud.

## DIFICULDADES ENCONTRADAS E SOLUÇÕES

1. *Integração com Health Connect*

*Dificuldade*: A API Health Connect apresentou limitações de documentação e compatibilidade irregular entre dispositivos Android.

*Solução*: Implementação de sistema de *fallback* com entrada manual de dados e utilização de múltiplas bibliotecas de integração para maximizar compatibilidade.

2. *Gestão de Estado Complexo*

*Dificuldade*: Sincronização entre dados locais e remotos, especialmente em cenários de conectividade intermitente.

*Solução*: Implementação de sistema de cache local robusto com sincronização automática quando a conectividade é restaurada, utilizando AsyncStorage e filas de sincronização.

3. *Performance em Dispositivos Antigos*

*Dificuldade*: Alguns dispositivos com Android 6.0 apresentaram lentidão no processamento de gráficos complexos.

*Solução*: Otimização de componentes através de lazy loading, memorização de componentes React e redução da complexidade de animações em dispositivos com menor performance.

4. *Notificações Confiáveis*

*Dificuldade*: Garantir que lembretes de medicação sejam entregues consistentemente, considerando as limitações de cada sistema operativo.

*Solução*: Implementação de sistema híbrido combinando notificações locais e push notifications, com redundância e confirmação de entrega.

## COMPETÊNCIAS DESENVOLVIDAS

Durante o desenvolvimento deste projeto, foram consolidadas e desenvolvidas diversas competências técnicas:

- Desenvolvimento Mobile: Domínio aprofundado do React Native e ecossistema Expo;

- Gestão de Bases de Dados: Modelação e implementação com PostgreSQL/Supabase;

- Integração de APIs: Trabalho com múltiplas APIs externas e gestão de autenticação;

- Design de Interface: Aplicação de princípios de UX/UI em ambiente móvel;

- Segurança de Aplicações: Implementação de medidas de segurança e conformidade regulatória;

- Gestão de Projeto: Planeamento, execução e controlo de projeto de software;

## TRABALHO FUTURO

### **Desenvolvimentos Planeados**

1. *Funcionalidades Avançadas:*

- Implementação de inteligência artificial para análise preditiva de dados de saúde;

- Sistema de telemedicina com videochamadas integradas;

- Gamificação para incentivo à adesão terapêutica.

2. *Expansão de Plataformas:*

- Desenvolvimento de aplicação web complementar;

- Integração com mais dispositivos vestíveis (Fitbit, Apple Watch);

- Versão para profissionais de saúde.

3. *Melhorias Técnicas:*

- Implementação de machine learning para deteção de anomalias;

- Sistema de backup automático com múltiplos fornecedores;

- Otimização adicional de performance.

### 

### **Plano de Melhoramentos**

*Curto Prazo (3 meses):*

- Melhoria da interface baseado em feedback de utilizadores;

- Correção de bugs menores identificados em testes;

- Inclusão de idiomas (inglês, francês).

*Médio Prazo (6 meses):*

- Implementação de funcionalidades de análise avançada;

- Integração com sistema nacional de saúde (SNS24);

- Desenvolvimento de API para terceiros.

*Longo Prazo (12 meses):*

- Certificação como dispositivo médico de Classe I;

- Expansão para mercados internacionais;

- Parcerias com instituições de saúde.

## CONSIDERAÇÕES FINAIS

O projeto HealthMate representa uma implementação bem-sucedida de conceitos modernos de desenvolvimento móvel aplicados à área da saúde digital. A aplicação demonstra como a tecnologia pode ser utilizada para capacitar os utilizadores na gestão da sua saúde, contribuindo para os Objetivos de Desenvolvimento Sustentável.

O conhecimento adquirido durante este projeto, tanto a nível técnico como de gestão, constitui uma base sólida para futuros desenvolvimentos na área das tecnologias de saúde digital, um setor em rápida expansão e de crescente importância social.

# 5 – Bibliografia

ChatGPT  
<https://chat.openai.com>  
Assistente inteligência artificial desenvolvimento

Cursor AI  
<https://cursor.sh>  
Editor código com IA

EAS Build  
<https://docs.expo.dev/build/introduction/>  
Serviço build aplicações nuvem

React Native  
<https://reactnative.dev/>  
Framework desenvolvimento mobile multiplataforma

React Native Directory  
<https://reactnative.directory/>  
Bibliotecas componentes React Native

React Native CLI  
<https://reactnative.dev/docs/environment-setup>  
Configuração ambiente desenvolvimento nativo

React Native Performance  
<https://reactnative.dev/docs/performance>  
Otimização performance aplicações móveis

React Navigation  
<https://reactnavigation.org/>  
Sistema navegação React Native

React Native Paper  
<https://reactnativepaper.com/>  
Componentes Material Design React

React Native Elements  
<https://reactnativeelements.com/>  
Biblioteca componentes interface React

React Native Vector Icons  
<https://github.com/oblador/react-native-vector-icons>  
Ícones customizáveis aplicações móveis

React Native Gesture Handler  
<https://docs.swmansion.com/react-native-gesture-handler/>  
Gestos nativos alta performance

React Native Reanimated  
<https://docs.swmansion.com/react-native-reanimated/>  
Animações fluidas React Native

React Native Community  
<https://github.com/react-native-community>  
Packages oficiais comunidade React

JavaScript MDN  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>  
Documentação completa linguagem JavaScript

JavaScript ES6 Features  
<https://www.ecma-international.org/ecma-262/>  
Especificação ECMAScript linguagem moderna

JavaScript Promises  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise>  
Programação assíncrona JavaScript promises

JavaScript Async/Await  
<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Async_await>  
Sintaxe assíncrona moderna JavaScript

Node.js Documentation  
<https://nodejs.org/en/docs/>  
Runtime JavaScript servidor backend

TypeScript  
<https://www.typescriptlang.org/>  
JavaScript com tipagem estática

JavaScript.info  
<https://javascript.info/>  
Tutorial moderno JavaScript completo

You Don't Know JS  
<https://github.com/getify/You-Dont-Know-JS>  
Série livros JavaScript profundo

JavaScript30  
<https://javascript30.com/>  
Projetos práticos JavaScript vanilla

JSON Documentation  
<https://www.json.org/>  
Formato intercâmbio dados JavaScript

Babel JavaScript  
<https://babeljs.io/>  
Transpilador JavaScript moderno compatível

ESLint  
<https://eslint.org/>  
Ferramenta linting código JavaScript

Expo Documentation  
<https://docs.expo.dev/>  
Plataforma desenvolvimento aplicações nativas

Expo CLI  
<https://docs.expo.dev/workflow/expo-cli/>  
Interface linha comando Expo

Expo SDK  
<https://docs.expo.dev/versions/latest/>  
Kit desenvolvimento APIs nativas

Expo Go  
<https://expo.dev/client>  
Cliente testes aplicações Expo

Expo Snack  
<https://snack.expo.dev/>  
Editor online código React

Expo Application Services  
<https://expo.dev/eas>  
Serviços build submit aplicações

Expo Router  
<https://docs.expo.dev/router/introduction/>  
Sistema roteamento baseado ficheiros

Expo Templates  
<https://github.com/expo/examples>  
Exemplos projetos Expo prontos

Expo Vector Icons  
<https://docs.expo.dev/guides/icons/>  
Biblioteca ícones integrada Expo

Expo Constants  
<https://docs.expo.dev/versions/latest/sdk/constants/>  
Constantes sistema aplicação Expo

Expo Notifications  
<https://docs.expo.dev/versions/latest/sdk/notifications/>  
Sistema notificações push Expo

Expo Font  
<https://docs.expo.dev/versions/latest/sdk/font/>  
Carregamento fontes customizadas Expo

Expo Asset  
<https://docs.expo.dev/versions/latest/sdk/asset/>  
Gestão recursos estáticos Expo

Supabase  
<https://supabase.com/docs>  
Backend-as-a-service PostgreSQL APIs

PostgreSQL  
<https://www.postgresql.org/docs/15/>  
Sistema gestão base dados

SQL Tutorial  
<https://www.w3schools.com/sql/>  
Linguagem consulta estruturada operações

PostgreSQL Tutorial  
<https://www.postgresqltutorial.com/>  
Guia completo PostgreSQL prático

SQL Joins  
<https://www.w3schools.com/sql/sql_join.asp>  
Relacionamentos tabelas base dados

SQL Constraints  
<https://www.postgresql.org/docs/current/ddl-constraints.html>  
Restrições integridade dados PostgreSQL

Row Level Security  
<https://www.postgresql.org/docs/current/ddl-rowsecurity.html>  
Segurança nível linha PostgreSQL

SQLBolt  
<https://sqlbolt.com/>  
Tutorial interativo SQL prático

Supabase Realtime  
<https://supabase.com/docs/guides/realtime>  
Sincronização dados tempo real

Samsung Health SDK  
<https://developer.samsung.com/health>  
Kit desenvolvimento integração Samsung

Samsung Health Platform  
<https://developer.samsung.com/health/android-health-platform>  
Plataforma saúde Android Samsung

Samsung Health Data  
<https://developer.samsung.com/health/android-health-platform/guide/data-types>  
Tipos dados saúde Samsung

Google Fit API  
<https://developers.google.com/fit>  
API integração dados fitness

Google Fit REST API  
<https://developers.google.com/fit/rest>  
Interface REST Google Fit

Fitness Store API  
<https://developers.google.com/fit/android/history>  
Armazenamento histórico dados fitness

Health Connect  
<https://developer.android.com/health-and-fitness/guides/health-connect>  
Plataforma unificada dados saúde

Android Health Platform  
<https://developer.android.com/health-and-fitness>  
Plataforma saúde fitness Android

HealthKit iOS  
<https://developer.apple.com/documentation/healthkit>  
Framework saúde dados iOS

Fitbit Web API  
<https://dev.fitbit.com/>  
API acesso dados Fitbit

Garmin Connect IQ  
<https://developer.garmin.com/connect-iq/>  
Plataforma desenvolvimento aplicações Garmin

Apple Health Records  
<https://developer.apple.com/documentation/healthkit/health_records>  
Registos médicos HealthKit Apple

FHIR HL7  
<https://www.hl7.org/fhir/>  
Padrão intercâmbio dados saúde

Samsung Health Studies  
<https://developer.samsung.com/health/server-health-data-platform>  
Plataforma estudos dados saúde

Wear OS Health  
<https://developer.android.com/training/wearables/health>  
APIs saúde Wear OS

ResearchKit  
<https://developer.apple.com/researchkit/>  
Framework investigação médica iOS

BluePixel API  
<https://bluepixel.com/api-docs>  
Serviços API externos integração

JSON Web Tokens  
<https://jwt.io/>  
Autenticação segura tokens JWT

REST API Guidelines  
<https://restfulapi.net/>  
Princípios design APIs REST

OAuth 2.0  
<https://oauth.net/2/>  
Protocolo autorização padrão web

React Context API  
<https://reactjs.org/docs/context.html>  
Gestão estado global React

AsyncStorage React Native  
<https://react-native-async-storage.github.io/async-storage/>  
Armazenamento local aplicações móveis

Metro Bundler  
<https://metrobundler.dev/>  
Bundler JavaScript React Native

Flipper Debugging  
<https://fbflipper.com/>  
Ferramenta debugging aplicações móveis

Android Debug Bridge  
<https://developer.android.com/studio/command-line/adb>  
Interface debugging dispositivos Android

Postman API  
<https://www.postman.com/>  
Ferramenta teste desenvolvimento APIs

Insomnia REST Client  
<https://insomnia.rest/>  
Cliente REST testes APIs

React Developer Tools  
<https://reactjs.org/blog/2019/08/15/new-react-devtools.html>  
Ferramentas debugging React aplicações

Redux DevTools  
<https://github.com/reduxjs/redux-devtools>  
Debugging gestão estado Redux

Yarn Package Manager  
<https://yarnpkg.com/>  
Gestor pacotes JavaScript alternativo

NPM Registry  
<https://www.npmjs.com/>  
Repositório pacotes JavaScript Node

GitHub API  
<https://docs.github.com/en/rest>  
API integração repositórios GitHub

Firebase Documentation  
<https://firebase.google.com/docs>  
Plataforma backend Google Firebase

Heroku Platform  
<https://devcenter.heroku.com/>  
Plataforma deploy aplicações nuvem

Vercel Deployment  
<https://vercel.com/docs>  
Deploy aplicações web frontend

# 6 – Anexos