

MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA  
CAMPUS GUARABIRA

## **RELATÓRIO TÉCNICO**

Projeto Integrador em Sistemas para Internet (PISI)

### **Integrantes**

Natália dos Santos Gomes  
João Igor dos Santos Barbosa

### **Orientador**

Prof. Dr. Rodrigo Leone Alves

### **Co-orientador**

Prof. Dr. Nasson Paulo Sales Neves

MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA  
CAMPUS GUARABIRA

## **RELATÓRIO TÉCNICO**

Projeto Integrador em Sistemas para Internet (PISI)

**Natália dos Santos Gomes**

**João Igor dos Santos Barbosa**

### **Orientador**

Prof. Dr. Rodrigo Leone Alves

### **Co-orientador**

Prof. Dr. Nasson Paulo Sales Neves

Relatório técnico apresentado à Coordenação do Curso Superior de Tecnologia em Sistemas para Internet, no âmbito do IFPB - Campus Guarabira, para a conclusão da disciplina de Projeto Integrador em Sistemas para Internet (PISI), em cumprimento às exigências do Projeto Pedagógico do Curso.

Guarabira/PB

2024.2

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IFPB - GUARABIRA

R633r

Gomes, Natália dos Santos

Relatório Técnico: Projeto Integrador em Sistemas para Internet (PISI)  
/ Natália dos Santos Gomes; João Igor dos Santos Barbosa.- Guarabira,  
2025.

36 f.: il; color.

Trabalho de Conclusão de Curso (Superior de Tecnologia em  
Sistemas para Internet). – Instituto Federal da Paraíba, Campus Guarabira,  
2025.

"Orientação: Prof. Dr. Rodrigo Leone Alves"

"Co-orientador: Prof. Dr. Nasson Paulo Sales Neves."

Referências.

1. Sistemas para Internet. 2. Tecnologia assistiva. 3. Comunicação  
emergencial. 4. Segurança digital. 5. Aplicação móvel. I. Título.

CDU 004.4(0.067)



MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA  
CAMPUS GUARABIRA

ATA 7/2025 - CCSTSI/DDE/DG/GB/REITORIA/IFPB

ATA DE APRESENTAÇÃO E DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO  
CCS de Tecnologia em Sistemas para Internet

Aos 13 de março de 2025, às 13:30, no Laboratório de Informática, reuniram-se os membros da banca avaliadora, Rodrigo Leone Alves (Orientador), Nasson Paulo Sales Neves (Coorientador), Rhavy Maia Guedes (Examinador Interno), Lucas Vieira de Souza (Examinador Interno), para avaliarem a apresentação do *Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet* (Relatório Final do Projeto Integrador em Sistemas para Internet - PISI) do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), campus Guarabira, desenvolvido pela aluno(a) **Natália dos Santos Gomes**, matrícula de nº **202023810004**, intitulado "*Sistema de Informações Vitais e Assistência Médica*", protocolado para apresentação de acordo com os requisitos expostos no Projeto Pedagógico de Curso de Tecnologia em Sistemas para Internet. Após a apresentação, a banca apresentou, por unanimidade, pareceres a favor da aprovação do trabalho. Desta forma, o Trabalho de Conclusão de Curso foi **aprovado** e definiu-se a **nota final 95 (Noventa e Cinco)**.

Nada mais havendo a tratar, às 16:45, encerraram-se os trabalhos, determinando a lavratura desta ata, que, após lida e considerada conforme, será assinada pelos presentes. Eu, Rodrigo Leone Alves, lavrei a presente ata.

Guarabira/PB, em 13 de março de 2025.

Documento assinado eletronicamente por:

- **Rodrigo Leone Alves**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 20/03/2025 12:11:52.
- **Lucas Vieira de Souza**, COORDENADOR(A) DE CURSO - FUC1 - CCSTSI-GB em 20/03/2025 16:57:07.
- **Rhavy Maia Guedes**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 20/03/2025 19:22:45.



Documento assinado digitalmente

**NASSON PAULO SALES NEVES**  
Data: 22/03/2025 16:26:54-0300  
Verifique em <https://validar.iti.gov.br>

Este documento foi emitido pelo SUAP em 20/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 685193  
Verificador: e0ec1e4fec  
Código de Autenticação:





MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA  
CAMPUS GUARABIRA

ATA 8/2025 - CCSTSI/DDE/DG/GB/REITORIA/IFPB

ATA DE APRESENTAÇÃO E DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO  
CCS de Tecnologia em Sistemas para Internet

Aos 13 de março de 2025, às 13:30, no Laboratório de Informática, reuniram-se os membros da banca avaliadora, Rodrigo Leone Alves (Orientador), Nasson Paulo Sales Neves (Coorientador), Rhavy Maia Guedes (Examinador Interno), Lucas Vieira de Souza (Examinador Interno), para avaliarem a apresentação do *Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet* (Relatório Final do Projeto Integrador em Sistemas para Internet - PISI) do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), campus Guarabira, desenvolvido pela aluno(a) **Joao Igor dos Santos Barbosa**, matrícula de nº 202213810041, intitulado "*Sistema de Informações Vitais e Assistência Médica*", protocolado para apresentação de acordo com os requisitos expostos no Projeto Pedagógico de Curso de Tecnologia em Sistemas para Internet. Após a apresentação, a banca apresentou, por unanimidade, pareceres a favor da aprovação do trabalho. Desta forma, o Trabalho de Conclusão de Curso foi **aprovado** e definiu-se a **nota final 95 (Noventa e Cinco)**.

Nada mais havendo a tratar, às 16:45, encerraram-se os trabalhos, determinando a lavratura desta ata, que, após lida e considerada conforme, será assinada pelos presentes. Eu, Rodrigo Leone Alves, lavrei a presente ata.

Guarabira/PB, em 13 de março de 2025.

Documento assinado eletronicamente por:

- **Rodrigo Leone Alves**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 20/03/2025 12:14:16.
- **Lucas Vieira de Souza**, COORDENADOR(A) DE CURSO - FUC1 - CCSTSI-GB em 20/03/2025 16:56:59.
- **Rhavy Maia Guedes**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 20/03/2025 19:22:09.



Documento assinado digitalmente

**NASSON PAULO SALES NEVES**  
Data: 22/03/2025 16:23:38-0300  
Verifique em <https://validar.iti.gov.br>

Este documento foi emitido pelo SUAP em 20/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 685197  
Verificador: ff21c1a8be  
Código de Autenticação:



## RESUMO

O aumento do número de pessoas que vivem sozinhas reforça a necessidade de soluções tecnológicas que facilitem a comunicação em situações emergenciais. O MAVIS – Sistema de Informações Vitais e Assistência Médica busca mitigar essa dificuldade ao permitir o cadastramento de contatos de emergência e informações médicas, viabilizando o envio automatizado de alertas via SMS e e-mail. O desenvolvimento seguiu princípios de engenharia de software e segurança da informação, empregando Spring Boot no backend e React Native no frontend. A abordagem metodológica baseou-se no framework Scrum, garantindo uma evolução contínua do sistema. Os resultados indicam que o MAVIS oferece uma solução eficiente para comunicação emergencial. Futuras melhorias incluem integração com dispositivos IoT e otimizações de acessibilidade para ampliar seu alcance.

**Palavras-chave:** Tecnologia assistiva; Comunicação emergencial; Notificações automatizadas; Segurança digital; Aplicação móvel.

## ABSTRACT

The increasing number of people living alone highlights the need for technological solutions that facilitate communication in emergency situations. MAVIS – Vital Information and Medical Assistance System aims to address this issue by enabling the registration of emergency contacts and essential medical information, allowing automated alerts to be sent via SMS and email. The development followed software engineering and information security principles, employing Spring Boot for the backend and React Native for the frontend. The methodological approach was based on the Scrum framework, ensuring continuous system improvement. The results indicate that MAVIS provides an efficient solution for emergency communication. Future enhancements include integration with IoT devices and accessibility optimizations to expand its reach.

**Keywords:** Assistive technology; Emergency communication; Automated notifications; Digital security; Mobile application.

## SUMÁRIO

<b>DADOS BÁSICOS.....</b>	<b>7</b>
<b>1. INTRODUÇÃO.....</b>	<b>8</b>
1.2. Tema.....	8
1.3. Objetivo.....	8
1.3.1. Objetivo geral.....	8
1.3.2. Objetivo Específico.....	8
1.4. Delimitação do problema.....	9
1.5. Justificativa.....	10
1.6. Método de Trabalho.....	10
1.7. Sobre os Objetivos de Desenvolvimento Sustentável.....	11
1.8. Sobre a Lei Geral de Proteção de Dados.....	11
1.9. Organização do Trabalho.....	11
<b>2. DESCRIÇÃO GERAL DO SISTEMA.....</b>	<b>13</b>
2.1. Descrição do Problema.....	13
2.2. Perfis de Usuários.....	13
2.3. Requisitos.....	14
2.3.1. Requisitos Funcionais.....	15
2.3.2. Requisitos Não-Funcionais.....	17
<b>3. ANÁLISE E DESIGN.....</b>	<b>19</b>
3.1. Diagrama de Classes.....	19
3.2. Diagrama de Atividades.....	20
3.3. Modelo de dados.....	20
3.3.1. Modelo lógico da base de dados.....	21
3.3.2. Criação física do modelo de dados.....	22
3.3.3. Dicionário de Dados.....	23
3.4. Ambiente de Desenvolvimento.....	28
<b>4. IMPLEMENTAÇÃO.....</b>	<b>31</b>
<b>5. CONCLUSÃO.....</b>	<b>35</b>
<b>REFERÊNCIAS.....</b>	<b>36</b>



## **DADOS BÁSICOS**

### **Natureza do Projeto Integrador em Sistemas para Internet (PISI)<sup>1</sup>**

- ( ) Produção Acadêmica  
(X) Projeto de Implementação

## **IDENTIFICAÇÃO**

Produto de Software: MAVIS – Sistema de Informações Vitais e Assistência Médica

### **Integrantes**

Natália dos Santos Gomes  
Joao Igor dos Santos Barbosa

### **Orientador**

Prof. Dr. Rodrigo Leone Alves

### **Co-orientador**

Prof. Dr. Nasson Paulo Sales Neves

### **Disciplina Vinculada**

Projeto Integrador em Sistemas para Internet (PISI) - 67h

### **Período de Construção**

Calendário Acadêmico: 2024.2  
Calendário Civil: outubro/2024 a março/2025

### **Data da Apresentação**

13 de Março de 2025

### **Banca de Avaliação**

Prof. Rodrigo Leone Alves  
Prof. Nasson Paulo Sales Neves  
Prof. Rhavy Maia Guedes  
Prof. Lucas Vieira de Souza

---

<sup>1</sup>

Segundo o Projeto Pedagógico de Curso - Tecnologia em Sistemas para Internet (IFPB, 2019), o PISI pode ser originado por:

I. **Produção Acadêmica**, resultante de pesquisa científica sobre um determinado objeto, ato, fato ou fenômeno da realidade ou da produção técnica ou tecnológica, visando a aplicabilidade nos diversos campos do saber, com atendimento aos padrões técnicos de intervenção. A produção acadêmica expressa as competências e as habilidades desenvolvidas (ou os conhecimentos adquiridos) pelos estudantes durante o período de formação.

II. **Projeto de Implementação**, em sentido lato, no qual se busca encontrar uma resposta prática para um problema técnico-profissional, tecnológico ou técnico-científico.

## **1. INTRODUÇÃO**

Este documento apresenta a concepção, implementação e funcionamento de um produto de software, abordando a contextualização do problema, a justificativa do tema e a metodologia utilizada. Além disso, descreve a estrutura do sistema, as conclusões obtidas, fornecendo uma visão geral do processo de desenvolvimento e das possíveis melhorias.

### **1.2. Tema**

O MAVIS – Sistema de Informações Vitais e Assistência Médica é uma solução desenvolvida para auxiliar pessoas que vivem sozinhas a se comunicarem com seus contatos de emergência em situações críticas. O sistema permite o cadastramento de contatos de confiança e informações médicas essenciais, possibilitando o envio automatizado de alertas por SMS e e-mail em casos de emergência. Sendo assim, o MAVIS contribui para a segurança dos usuários, agilizando a comunicação sem substituir serviços médicos ou de resgate.

### **1.3. Objetivo**

#### **1.3.1. Objetivo geral**

O objetivo deste projeto é desenvolver um sistema para gerenciar o perfil de saúde dos usuários, permitindo o cadastro seguro de informações pessoais, contatos de emergência e histórico médico, com foco na comunicação eficiente em situações críticas.

#### **1.3.2. Objetivo Específico**

- **Cadastro de Usuário:** Permitir o cadastro de usuários com informações pessoais básicas, como nome, idade, telefone e endereço, além da configuração de contatos de emergência. O histórico médico poderá ser preenchido posteriormente.
- **Contatos de Emergência:** Possibilitar o cadastro de um a três contatos de emergência, que receberão alertas em situações críticas. Os alertas serão enviados por SMS e

e-mail, cabendo aos contatos cadastrados a responsabilidade por acionar ajuda ou serviços médicos.

- **Histórico Médico:** Permitir o registro e a manutenção de informações médicas essenciais, como medicações em uso, alergias e condições preexistentes. O sistema deverá suportar até cinco versões do histórico médico, removendo automaticamente as mais antigas conforme novas informações forem adicionadas.
- **Alerta de Emergência:** Oferecer um mecanismo para o envio rápido e eficiente de alertas de emergência aos contatos cadastrados por SMS e e-mail.
- **Segurança e Privacidade:** Implementar boas práticas de segurança, incluindo criptografia de dados sensíveis, autenticação segura e controle de acesso adequado. Aplicar os princípios SOLID e boas práticas de engenharia de software para garantir uma arquitetura modular e segura.

#### **1.4. Delimitação do problema**

Atualmente, muitas pessoas vivem sozinhas e, em caso de emergências, podem enfrentar dificuldades para obter ajuda de forma rápida e eficiente. Em situações críticas, a comunicação com contatos de confiança pode ser um desafio, especialmente para aqueles que não têm um suporte imediato ao seu redor. Apesar da existência de diversos sistemas voltados para monitoramento de saúde, muitos desses exigem equipamentos especializados ou integração com serviços médicos, o que pode não ser acessível para todos os usuários.

A dificuldade em estabelecer um canal de comunicação emergencial acessível e intuitivo representa um problema relevante, uma vez que a demora no acionamento de contatos de emergência pode comprometer a segurança e o bem-estar de quem precisa de ajuda. Além disso, a ausência de um meio estruturado para armazenar e compartilhar informações médicas essenciais pode dificultar a tomada de decisões rápidas por parte dos contatos responsáveis.

Dessa forma, observa-se a necessidade de um sistema que possibilite o gerenciamento de informações pessoais e médicas, garantindo que, em casos de emergência, os contatos cadastrados sejam notificados de maneira eficaz e segura.

### **1.5. Justificativa**

A escolha do tema "MAVIS – Sistema de Informações Vitais e Assistência Médica" justifica-se pela necessidade de oferecer uma solução tecnológica que promova segurança e agilidade na comunicação em situações de emergência, especialmente para pessoas que vivem sozinhas. Esse cenário tem se tornado cada vez mais comum, refletindo o crescimento do movimento "single", que, no Brasil, já representa mais de 11 milhões de domicílios ocupados por apenas uma pessoa (BASSOLI, 2023). A ausência de uma rede de suporte imediato pode dificultar o acesso a ajuda em momentos críticos, aumentando os riscos à segurança e ao bem-estar desses indivíduos.

O MAVIS surge como uma solução inovadora para esse problema, permitindo o cadastramento de contatos de emergência e informações médicas essenciais, além do envio automático de alertas por SMS e e-mail em situações de risco. Dessa forma, o sistema viabiliza uma resposta rápida e eficaz, aumentando as chances de uma intervenção imediata por parte dos contatos cadastrados, o que pode ser determinante para a preservação da vida e a mitigação de riscos.

Em suma, a escolha do MAVIS como tema se fundamenta na crescente demanda por ferramentas acessíveis e intuitivas que facilitem a comunicação emergencial, proporcionando maior segurança e suporte para aqueles que enfrentam situações de vulnerabilidade.

### **1.6. Método de Trabalho**

Para o desenvolvimento da aplicação, adotamos o SCRUM, um framework ágil que nos permitiu avançar por meio de entregas incrementais durante cada sprint. Essa abordagem possibilitou que o software fosse aprimorado continuamente, entregando valor a cada iteração e permitindo ajustes rápidos conforme surgiam novas necessidades. No gerenciamento das tarefas, utilizamos o Trello, uma ferramenta colaborativa que tornou o processo de organização e priorização mais claro e eficiente, garantindo que todas as atividades estivessem sempre sob controle e alinhadas com os objetivos do projeto.

Para a modelagem dos dados, optamos pela Programação Orientada a Objetos, o que nos ofereceu a flexibilidade de criar um sistema modular, fácil de entender e manter. Essa abordagem proporcionou uma estrutura sólida e também garantiu que o software fosse escalável, permitindo evoluções futuras sem comprometer sua integridade.

### **1.7. Sobre os Objetivos de Desenvolvimento Sustentável**

O aplicativo MAVIS está alinhado com os Objetivos de Desenvolvimento Sustentável (ODS) estabelecidos pela ONU, especialmente no que diz respeito à promoção da saúde e bem-estar (ODS 3) e à redução das desigualdades (ODS 10). Ao oferecer um serviço de gestão de perfis de usuários com informações de saúde e contatos de emergência, o MAVIS contribui para a melhoria da qualidade de vida e o acesso rápido à assistência em situações críticas. Além disso, o aplicativo visa a inclusão de pessoas em situação de vulnerabilidade, como idosos e aqueles que moram sozinhos, garantindo que tenham acesso à ajuda necessária em momentos de emergência, promovendo a equidade no acesso à saúde e segurança.

### **1.8. Sobre a Lei Geral de Proteção de Dados**

Em conformidade com a LGPD, o aplicativo MAVIS adota medidas para proteger os dados pessoais dos usuários. A senha dos usuários é armazenada de forma criptografada, garantindo sua segurança. O acesso ao sistema é restrito a usuários autenticados por login e senha, ou JWT, permitindo que apenas usuários autorizados acessem suas informações pessoais. O MAVIS assegura a transparência no uso dos dados e respeita os direitos dos usuários, conforme previsto pela LGPD, como o direito de acessar, corrigir ou excluir seus dados a qualquer momento.

### **1.9. Organização do Trabalho**

Este trabalho está estruturado de forma a apresentar, de maneira clara e sequencial, os aspectos fundamentais do desenvolvimento do sistema. Inicialmente, será realizada uma

descrição do problema, contextualizando os dados gerais e os perfis de usuário envolvidos. Em seguida, abordaremos os requisitos do sistema, tanto funcionais quanto não funcionais, essenciais para a definição dos objetivos e das limitações do projeto.

Na sequência, apresentaremos a análise e o design do sistema, com ênfase nos diagramas que ilustram a estrutura do software, como o diagrama de classes e o diagrama de atividades, além do modelo lógico e físico da base de dados, acompanhado pelo dicionário de dados. Este conjunto de elementos representa o planejamento e a arquitetura que sustentam a implementação do sistema.

O próximo tópico se dedica à implementação, detalhando as etapas práticas do desenvolvimento e as ferramentas utilizadas ao longo do processo. Por fim, a conclusão sintetiza os resultados obtidos, destacando as principais realizações e considerações finais sobre o trabalho executado, apontando também possíveis melhorias e desdobramentos futuros.

## **2. DESCRIÇÃO GERAL DO SISTEMA**

Este capítulo tem como objetivo descrever de forma geral o sistema, seu escopo e as principais funcionalidades.

### **2.1. Descrição do Problema**

Muitas pessoas, especialmente aquelas que vivem sozinhas, enfrentam dificuldades em situações de emergência, quando precisam de ajuda imediata, mas não têm acesso rápido a um suporte. A ausência de uma rede de apoio imediato pode resultar em sérios riscos à segurança e ao bem-estar dessas pessoas. O problema se intensifica na falta de soluções tecnológicas que garantam comunicação rápida e eficiente com contatos de emergência, o que pode afetar a agilidade na obtenção de assistência em momentos críticos. Esse cenário destaca a necessidade de um sistema que possibilite o envio de alertas automáticos aos contatos de emergência, oferecendo uma resposta mais rápida e eficaz em situações de risco, sem substituir serviços médicos ou de resgate, mas complementando-os de forma ágil.

### **2.2. Perfis de Usuários**

Os perfis de usuários do aplicativo MAVIS são descrições dos grupos de pessoas que interagem com o sistema, definindo suas necessidades, responsabilidades e permissões. A seguir, são apresentados os principais perfis de usuários do sistema, com ênfase nas interações que cada um terá com a plataforma.

- **Perfil 01 – Usuário Final:** O usuário final é o principal responsável pelo uso do MAVIS para gerenciar suas informações pessoais e de saúde. Esse perfil inclui pessoas com doenças crônicas, pessoas com necessidades especiais e indivíduos que residem sozinhos, sem uma rede de apoio próxima. Suas responsabilidades no sistema incluem:

- Cadastro e gerenciamento de dados pessoais e de saúde (como nome, idade, condições médicas e contatos de emergência).
  - Envio de alertas de emergência: O usuário pode acionar um alerta de emergência que será enviado aos seus contatos de emergência registrados.
  - Permissões: O usuário final tem permissão para editar suas próprias informações, adicionar contatos de emergência e acionar alertas de emergência.
- 
- **Perfil 02 – Contato de Emergência:** Os contatos de emergência são pessoas cadastradas pelo usuário final para serem alertadas em caso de emergência. Esses contatos podem ser familiares, amigos próximos ou cuidadores, e suas responsabilidades incluem:
    - Receber notificações de emergência: Quando o usuário final envia um alerta de emergência, os contatos de emergência recebem uma notificação obrigatória por SMS e/ou e-mail, com a localização atual do usuário e um link para o Google Maps. Devido às limitações de caracteres do SMS, a mensagem inclui apenas um aviso e a localização, enquanto o e-mail contém mais detalhes, incluindo dados pessoais e histórico médico.
    - Ações após a notificação: Os contatos de emergência devem entrar em contato com o usuário imediatamente ou, caso não consigam, acionar os serviços de emergência.
    - Permissões: O contato de emergência não pode editar dados do usuário, eles têm acesso apenas às notificações de emergência para facilitar uma resposta adequada.

## 2.3. Requisitos

Neste tópico, serão apresentados os requisitos que norteiam o desenvolvimento do sistema MAVIS, divididos em Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF). A fim de organizar e priorizar a implementação, os requisitos são classificados com base em sua importância para o funcionamento do sistema, conforme as seguintes categorias:



- **Essencial:** Requisitos imprescindíveis para o funcionamento do sistema. Esses requisitos devem ser implementados nas primeiras iterações do desenvolvimento, pois são fundamentais para a entrega de uma versão funcional do sistema.
- **Importante:** Requisitos que, embora não sejam essenciais, são necessários para garantir uma experiência satisfatória para o usuário. Sua implementação deve ocorrer logo após os requisitos essenciais, mas sua ausência não impede a implementação parcial do sistema.
- **Desejável:** Requisitos que não comprometem as funcionalidades básicas do sistema. O sistema pode operar de forma satisfatória sem sua implementação, e esses requisitos podem ser implementados por último, sem impacto no funcionamento principal.

#### 2.3.1. Requisitos Funcionais

**RF\_01: Cadastro de usuário no sistema** - O sistema deve permitir que os usuários se cadastrem informando nome de usuário, e-mail e senha.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_02: Autenticação de usuário** - O sistema deve permitir a autenticação de usuários através de login básico (usuário e senha) e geração de token de acesso para URLs privadas.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_03: Refresh Token** - O sistema deve permitir a geração de refresh token para renovação de tokens de acesso, proporcionando maior segurança e usabilidade.

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

**RF\_04: Exclusão de conta de usuário** - O sistema deve permitir que o usuário exclua sua conta, apagando todos os dados relacionados ao seu perfil e histórico no sistema.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_05: Criação de perfil de saúde** - O sistema deve permitir que o usuário crie seu perfil

com informações como nome completo, data de nascimento, telefone, endereço, localização e ao menos um contato de emergência.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_06: Atualização de perfil de saúde** - O sistema deve permitir que o usuário atualize seu perfil de maneira parcial ou total, incluindo informações como nome completo, data de nascimento, telefone ou endereço.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_07: Visualização do perfil de saúde** - O sistema deve permitir que o usuário visualize seu perfil completo, incluindo nome completo, data de nascimento, telefone, endereço, lista de contatos de emergência e histórico médico (incluindo alergias, medicações, doenças pré-existentes e data de criação).

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_08: Cadastro de contato de emergência** - O sistema deve permitir que o usuário adicione contatos de emergência, incluindo nome, e-mail, telefone e relacionamento. O usuário poderá adicionar de 1 a 3 contatos.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_09: Listagem de contatos de emergência** - O sistema deve permitir que o usuário final visualize uma lista com os seus contatos de emergência cadastrados. A listagem deverá exibir as informações essenciais de cada contato, como nome, número de telefone e relação com o usuário.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_10: Edição de contato de emergência** - O sistema deve permitir que o usuário edite individualmente seus contatos de emergência, podendo alterar nome, e-mail, telefone e relacionamento.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_11: Exclusão de contato de emergência** - O sistema deve permitir que o usuário exclua

um contato de emergência, desde que haja pelo menos 1 contato registrado no sistema.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_12: Adicionar informações ao histórico médico** - O sistema deve permitir que o usuário adicione informações ao seu histórico médico, como medicamentos, alergias e condições pré-existentes. O histórico médico será versionado e o sistema manterá no máximo 5 versões, apagando automaticamente as versões mais antigas.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_13: Listagem de histórico médico** - O sistema deve permitir que o usuário visualize todas as versões do seu histórico médico, incluindo medicação, alergias e condições pré-existentes, com a data de criação de cada versão.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RF\_14: Envio de alerta de emergência para contatos de emergência** - O sistema deve coletar as informações do perfil de saúde do usuário e gerar um alerta de emergência a ser enviado aos contatos de emergência cadastrados. O alerta será enviado por SMS e, quando disponível, também por e-mail. A mensagem de SMS conterá uma breve notificação com o link para a localização atual do usuário no Google Maps. Caso o alerta seja enviado por e-mail, ele incluirá informações adicionais, como dados pessoais, histórico médico e localização.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

### 2.3.2. Requisitos Não-Funcionais

**RNF\_01: Disponibilidade** - O sistema deve garantir uma alta disponibilidade, com no mínimo 97,5% de uptime mensal, monitorado por logs e ferramentas de observabilidade. Períodos de manutenção programada devem ser comunicados aos usuários com antecedência mínima de 24 horas.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RNF\_02: Eficiência** - O envio de notificações de emergência para os contatos cadastrados deve ser concluído em até 5 segundos, considerando um ambiente de rede 4G estável ou Wi-Fi com latência máxima de 100ms.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RNF\_03: Confiabilidade** - O sistema não deve apresentar falhas críticas em mais de 1% das tentativas de envio de alerta de emergência dentro de um período de 24 horas. Todas as falhas devem ser registradas e analisadas para manutenção preventiva.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RNF\_04: Usabilidade** - O aplicativo deve oferecer uma interface intuitiva e acessível, garantindo que todas as funcionalidades principais (perfil de saúde, histórico médico, contatos de emergência e envio de alerta) estejam acessíveis em no máximo três toques dentro do aplicativo, promovendo uma navegação eficiente.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RNF\_05: Manutenibilidade** - O backend deve seguir uma arquitetura modular baseada em camadas de serviço e princípios SOLID, garantindo a separação clara de responsabilidades. O código deve ser acompanhado de documentação clara para facilitar manutenções futuras.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

**RNF\_06: Performance** - O tempo de resposta do sistema para acessar e atualizar o perfil do usuário, assim como consultar o histórico médico, não deve ultrapassar 2 segundos em conexões 4G ou Wi-Fi estável, considerando uma carga normal do sistema.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

**RNF\_07: Segurança** - O sistema deve adotar práticas de segurança recomendadas. A autenticação deve ser baseada em JWT (JSON Web Token), utilizando chaves RSA para assinatura e validação dos tokens. O sistema deve estar em conformidade com a LGPD (Lei Geral de Proteção de Dados Pessoais), garantindo a proteção dos dados do usuário.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

### 3. ANÁLISE E DESIGN

Neste capítulo, são abordados os processos de análise e design do sistema MAVIS, que envolvem a definição e o planejamento das estruturas essenciais para o funcionamento da aplicação. Esta fase é fundamental para planejar a solução de forma eficiente, atendendo aos requisitos do sistema e assegurando uma implementação coesa e integrada.

#### 3.1. Diagrama de Classes

O diagrama de classes foi elaborado para representar as principais entidades do sistema e suas interações. Ele descreve a estrutura das classes e seus atributos, formando a base para o desenvolvimento orientado a objetos do software.

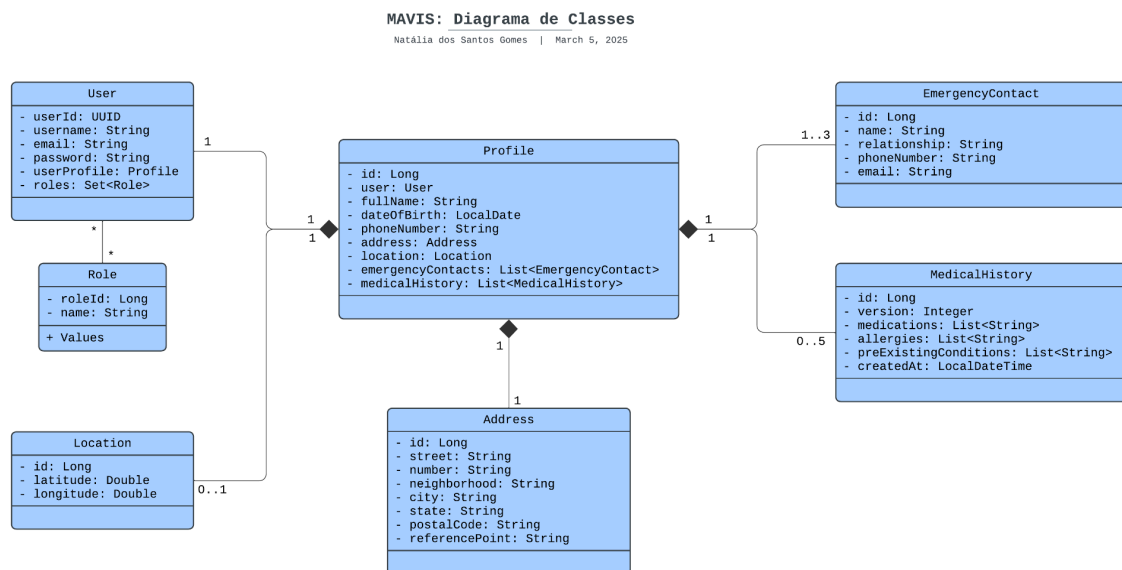


Figura 3.1 – Diagrama de Classes.<sup>2</sup>

<sup>2</sup> Elaborado por Natália S. Gomes utilizando a ferramenta Lucidchart.

### 3.2. Diagrama de Atividades

O Diagrama de Atividades tem como objetivo ilustrar o fluxo das principais funcionalidades do sistema MAVIS, conforme os requisitos funcionais estabelecidos. As atividades do sistema são organizadas de maneira sequencial, refletindo as ações que os usuários e o sistema precisam executar para garantir o bom funcionamento da aplicação.

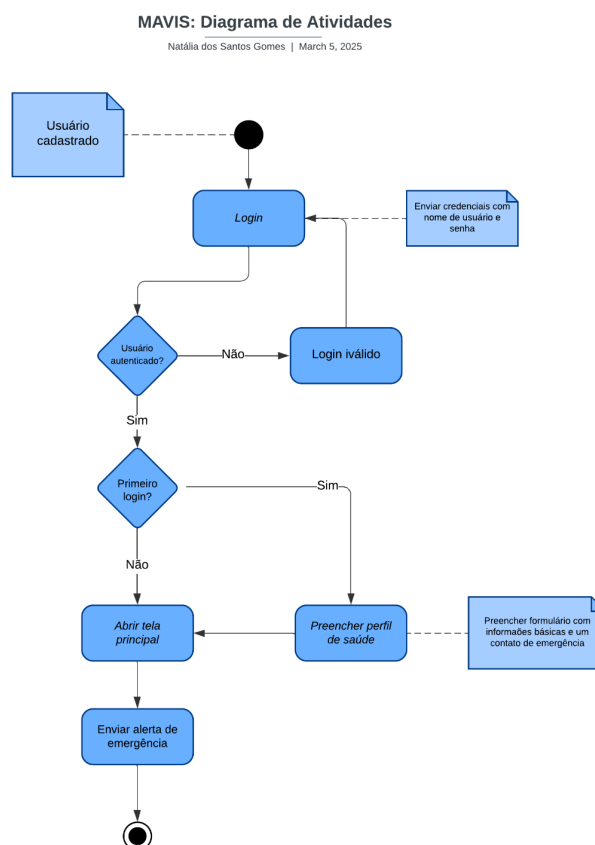


Figura 3.2 – Diagrama de Atividades<sup>3</sup>

### 3.3. Modelo de dados

O modelo de dados descreve a estrutura e a organização das informações armazenadas no sistema, incluindo as entidades, seus atributos e os relacionamentos entre elas. Ele serve

<sup>3</sup> Elaborado por Natália S. Gomes utilizando a ferramenta Lucidchart.

como base para o desenvolvimento do banco de dados, garantindo que as informações sejam armazenadas de forma eficiente e consistente.

### 3.3.1. Modelo lógico da base de dados

O modelo lógico da base de dados é uma representação abstrata e estruturada de como as informações serão organizadas no banco de dados. Ele descreve as tabelas que irão armazenar os dados e os relacionamentos entre essas tabelas, ou seja, como elas se conectam e interagem umas com as outras.

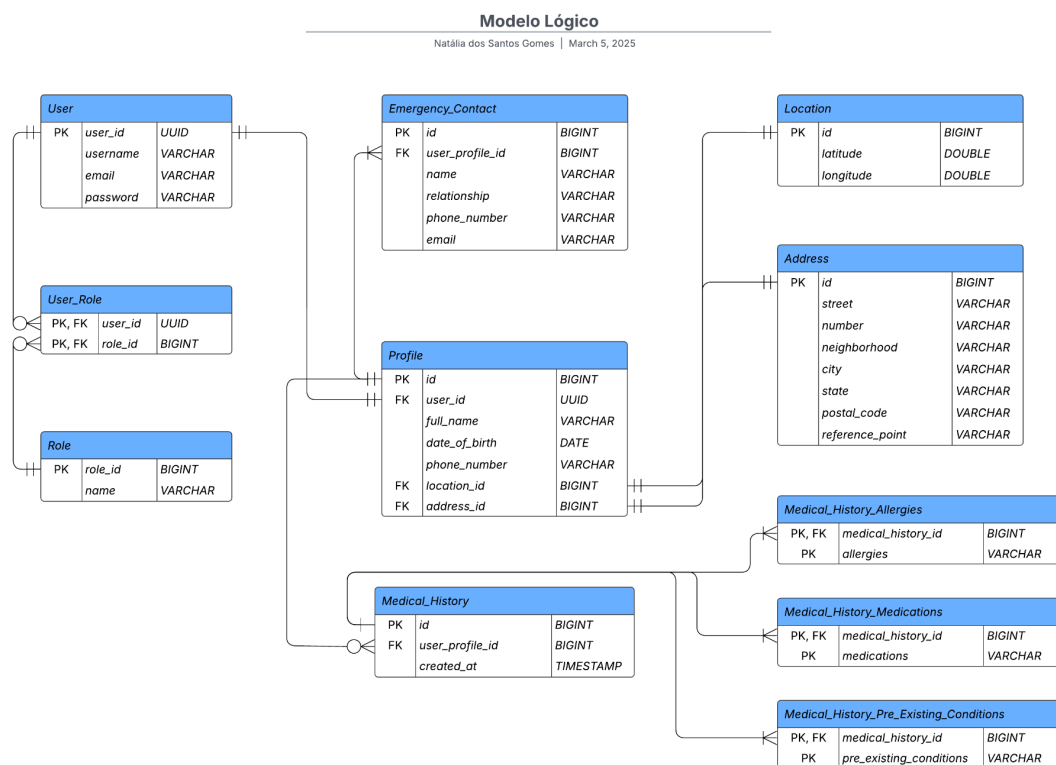


Figura 3.3 – Modelo Lógico da Base de Dados<sup>4</sup>

<sup>4</sup> Elaborado por Natália S. Gomes utilizando a ferramenta Lucidchart.

### 3.3.2. Criação física do modelo de dados

A criação física do modelo de dados refere-se à implementação concreta do banco de dados, incluindo a criação e organização das tabelas, índices e relacionamentos. Para isso, utilizamos o Flyway, uma ferramenta que automatiza e controla as migrações do banco de dados. Com o Flyway, desenvolvemos scripts SQL que são executados automaticamente, construindo as tabelas e relações conforme o modelo lógico, garantindo que a configuração do banco de dados seja consistente em diferentes ambientes de desenvolvimento.

Além disso, utilizamos o Docker Compose para orquestrar o serviço do PostgreSQL, facilitando a configuração e o gerenciamento do banco de dados em containers. Para visualizar a estrutura física do banco de dados, empregamos o pgAdmin, que permite gerar um diagrama visual das tabelas e seus relacionamentos, oferecendo uma visão clara da implementação atual.

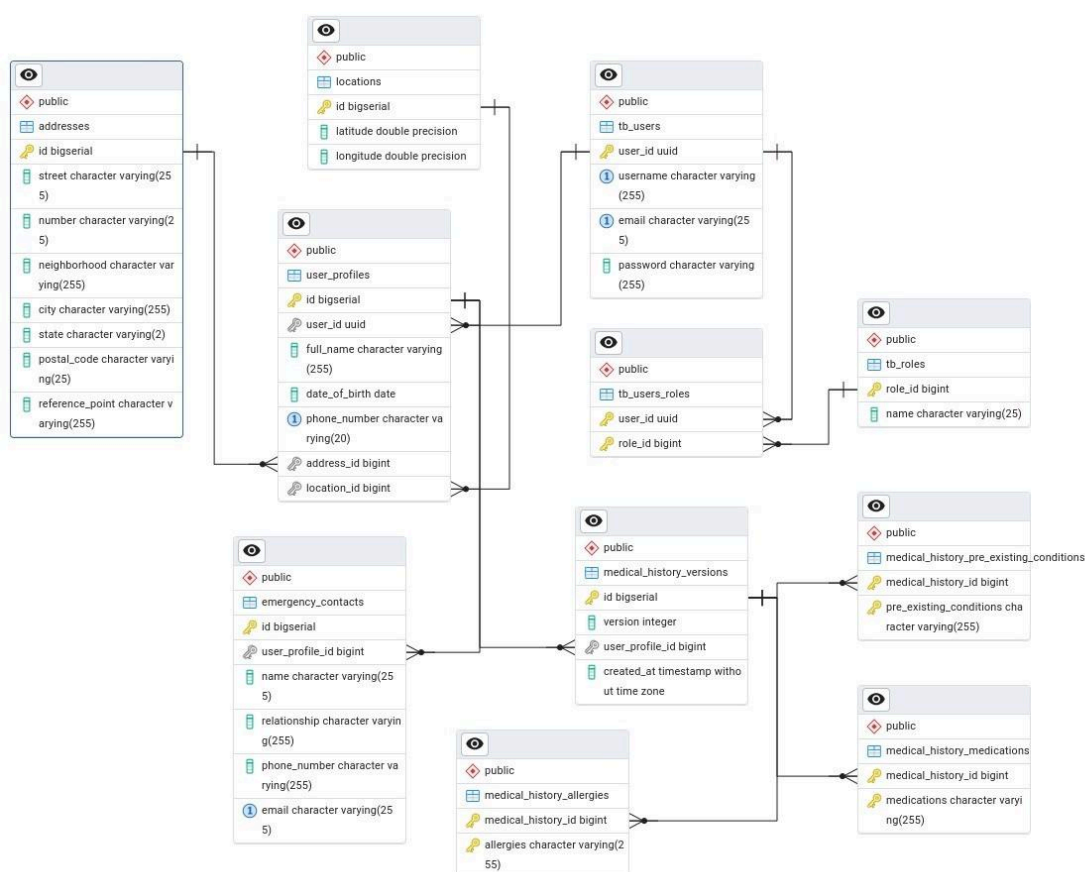


Figura 3.4 – Modelo Físico da Base de Dados<sup>5</sup>

<sup>5</sup> Gerado automaticamente pela ferramenta ERD do PgAdmin.



### 3.3.3. Dicionário de Dados

O Dicionário de Dados apresenta uma descrição detalhada das tabelas e seus campos no banco de dados, incluindo tipos de dados, regras de validação e observações importantes. Ele serve como uma referência para entender a estrutura de dados do sistema, as relações entre as tabelas e as restrições aplicadas a cada campo. O dicionário também inclui informações sobre chaves primárias, chaves estrangeiras e campos gerados automaticamente, para facilitar o entendimento e a manutenção do banco de dados.

tb_roles				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
rode_id	Identificador único da role	BIGINT	Deve ser único, chave primária	
name	Nome da role	VARCHAR (25)	Não pode ser nulo	

tb_users				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
user_id	Identificador único do usuário	UUID	Deve ser único, chave primária	Gerado automaticamente pelo sistema
username	Nome de usuário	VARCHAR (255)	Deve ser único, não pode ser nulo	
email	Endereço de e-mail do usuário	VARCHAR (255)	Deve ser único, não pode ser nulo	
password	Senha do usuário	VARCHAR (255)	Deve ser único, não pode ser nulo	

tb_users_roles				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
user_id	Identificador do usuário associado	UUID	Deve existir na tabela <i>tb_users</i>	Chave estrangeira para <i>tb_users</i>
role_id	Identificador da role associada	BIGINT	Deve existir na tabela <i>tb_roles</i>	Chave estrangeira para <i>tb_roles</i>

addresses				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
id	Identificador único do endereço	BIGINT	Chave primária	Gerado automaticamente pelo sistema
street	Rua do endereço	VARCHAR (255)	Não pode ser nulo	
number	Número do endereço	VARCHAR (25)	Não pode ser nulo	
neighborhood	Bairro do endereço	VARCHAR (255)	Não pode ser nulo	
city	Cidade do endereço	VARCHAR (255)	Não pode ser nulo	
state	Estado do endereço	VARCHAR (2)	Não pode ser nulo	
postal_code	Código postal do endereço	VARCHAR (25)	Não pode ser nulo	
reference_point	Ponto de referência do endereço	VARCHAR (255)	Opcional	

locations				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
id	Identificador único da localização	BIGINT	Chave primária	Gerado automaticamente pelo sistema
latitude	Latitude da localização	DOUBLE PRECISION	Não pode ser nulo	
longitude	Longitude da localização	DOUBLE PRECISION	Não pode ser nulo	

user_profiles				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
id	Identificador único do perfil	BIGINT		Gerado automaticamente pelo sistema
user_id	Identificador do usuário associado	UUID	Deve existir na tabela <i>tb_users</i>	Chave estrangeira para <i>tb_users</i>
full_name	Nome completo do usuário	VARCHAR (255)	Não pode ser nulo	
date_of_birth	Data de nascimento do usuário	DATE	Não pode ser nulo	
phone_number	Número de telefone do usuário	VARCHAR (20)	Não pode ser nulo, único	
address_id	Identificador do endereço do usuário	BIGINT		Chave estrangeira para <i>addresses</i>

location_id	Identificador da localização do usuário	BIGINT		Chave estrangeira para <i>locations</i>
-------------	---	--------	--	---

emergency_contacts				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
id	Identificador único do contato	BIGINT	Chave primária	Gerado automaticamente pelo sistema
user_profile_id	Identificador do perfil de saúde associado	BIGINT	Deve existir na tabela <i>user_profiles</i>	Chave estrangeira para <i>user_profiles</i>
name	Nome do contato	VARCHAR (255)	Não pode ser nulo	
relationship	Relacionamento com o usuário	VARCHAR (255)	Opcional	
phone_number	Número de telefone do contato	VARCHAR (255)	Não pode ser nulo	
email	E-mail do contato	VARCHAR (255)	Deve ser único, opcional	

medical_history_versions				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
id	Identificador único da versão	BIGINT	Chave primária	Gerado automaticamente e pelo sistema
version	Número da versão	INT	Não pode ser nulo	

user_profile_id	Identificador do perfil de saúde associado	BIGINT	Deve existir na tabela <i>users_profiles</i>	Chave estrangeira para <i>user_profiles</i>
created_at	Data de criação da versão	TIMESTAMP	Não pode ser nulo	Valor padrão: CURRENT_TIMESTAMP

medical_history_allergies				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
medical_history_id	Identificador do histórico médico	BIGINT	Deve existir na tabela <i>medical_history_versions</i>	Chave estrangeira para <i>medical_history_versions</i>
allergies	Alergia associada ao histórico médico	VARCHAR (255)	Pode ser nulo	

medical_history_medications				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
medical_history_id	Identificador do histórico médico	BIGINT	Deve existir na tabela <i>medical_history_versions</i>	Chave estrangeira para <i>medical_history_versions</i>
madications	Medicamentos associados ao histórico médico	VARCHAR (255)	Pode ser nulo	

medical_history_pre_existing_conditions				
Nome do campo	Descrição	Tipo	Regras de validação	Observações
medical_history_id	Identificador do histórico médico	BIGINT	Deve existir na tabela <i>medical_history_versions</i>	Chave estrangeira para <i>medical_history_versions</i>
pre_existing_conditions	Condições preexistentes associadas ao histórico médico	VARCHAR (255)	Pode ser nulo	

### 3.4. Ambiente de Desenvolvimento

O ambiente de desenvolvimento é o conjunto de ferramentas, tecnologias e práticas utilizadas para criar, testar e implementar um sistema de software. Em um contexto de desenvolvimento de software, é crucial que o ambiente esteja bem estruturado e configurado adequadamente, para que a colaboração entre as equipes seja eficaz e o desenvolvimento ocorra de forma fluida e sem contratempos.

No desenvolvimento web, é muito comum ouvir falar em dois termos importantes: backend e frontend. Para simplificar, imagine que o ambiente de desenvolvimento é como um estúdio de gravação. Nesse espaço, todos os equipamentos e ferramentas são usados para criar uma música de alta qualidade. O backend seria como a parte técnica do estúdio, onde ferramentas como Spring Boot e PostgreSQL processam e armazenam os dados, enquanto o frontend seria como a performance ao vivo, onde o React Native garante que a experiência do usuário seja clara e envolvente. O Trello funciona como partitura, organizando as tarefas e garantindo que cada etapa do processo seja seguida corretamente, resultando em um sistema bem executado e de alta qualidade.

## *Backend*

Ambiente de Desenvolvimento Integrado (IDE): IntelliJ IDEA - Ferramenta utilizada para facilitar a escrita e o gerenciamento do código. Oferece uma ampla gama de recursos que tornam o desenvolvimento mais eficiente e produtivo.

Linguagem de Programação: Java - É uma linguagem de programação orientada a objetos, amplamente utilizada para desenvolver aplicações que podem ser executadas em diversas plataformas, graças à Java Virtual Machine (JVM).

Framework: Spring Boot - Utilizado para simplificar o desenvolvimento de aplicações Java. Proporciona uma configuração rápida e fácil, com funcionalidades como injeção de dependência e gerenciamento de transações, permitindo o desenvolvimento ágil e escalável de serviços web e APIs (Interfaces de Programação de Aplicações).

Sistema de Gerenciamento de Banco de Dados Relacional: PostgreSQL - Usado para armazenar as informações do sistema de forma eficiente e estruturada, o PostgreSQL é conhecido pela sua alta confiabilidade, robustez e capacidade de suportar grandes volumes de dados.

Sistema de Controle de Versão: Git - Ferramenta essencial para gerenciar e versionar o código-fonte. Permite que múltiplos desenvolvedores trabalhem simultaneamente no projeto, garantindo que todas as alterações sejam registradas e que o código seja mantido em versões estáveis.

Plataforma de Containers: Docker - Utilizada para automatizar a implantação da aplicação em containers. Foi criada uma imagem Docker para o Spring Boot, permitindo que o serviço backend seja executado de maneira isolada e consistente. O Docker Compose foi utilizado para orquestrar a execução tanto do backend quanto do PostgreSQL em containers, garantindo o funcionamento integrado e eficiente de ambos.

## *Frontend*

Ambiente de Desenvolvimento Integrado (IDE): Visual Studio Code - Ferramenta utilizada para escrever e gerenciar o código do frontend.

Linguagem de Programação: JavaScript - É uma linguagem de programação amplamente utilizada para criar interatividade em páginas web. Ela permite que desenvolvedores adicionem comportamentos dinâmicos, como animações, manipulação de dados em tempo real e interação com o usuário, tornando as páginas mais responsivas e dinâmicas.

Framework: React Native - Utilizado para construir aplicativos móveis nativos utilizando JavaScript e React. Permite o desenvolvimento de aplicações para iOS e Android com uma única base de código, oferecendo alto desempenho e experiência de usuário consistente.

Aplicativo de Emulação: Expo Go - Usado como emulador para testar a aplicação móvel durante o desenvolvimento. O Expo Go permite que o código seja executado em dispositivos móveis de maneira rápida e eficiente, facilitando a visualização do progresso da aplicação.

Framework CSS: Tailwind CSS - Framework de CSS utilizado para criar interfaces responsivas e estilizadas de maneira rápida e flexível. O Tailwind permite um controle granular sobre a aparência da aplicação, proporcionando um design limpo e eficiente com classes utilitárias.

Sistema de Controle de Versão: Git - Ferramenta usada para gerenciar e versionar o código-fonte do frontend. Com o Git, é possível trabalhar em equipe de forma colaborativa, mantendo o controle de todas as mudanças e garantindo versões estáveis do código.



## 4. IMPLEMENTAÇÃO

A implementação foi conduzida com duas estruturas de projeto complementares: o backend, desenvolvido com Spring Boot, e o frontend, construído com React Native.

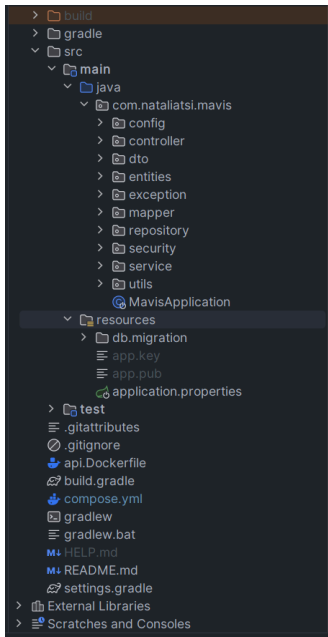
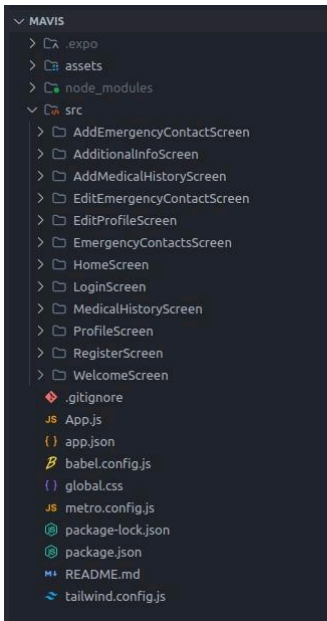
Backend	Frontend
	

Figura 4.1 – Captura de Tela dos Diretórios e Arquivos do Projeto<sup>6</sup>

### Backend

- Diretórios e Estrutura do Projeto

*build/*: Contém as classes compiladas do projeto após o processo de build.

*gradle/*: Diretório que contém arquivos e configurações relacionados ao Gradle, ferramenta de automação de build usada para compilar, testar e empacotar o projeto Java.

- Diretório de Código-fonte (src)

*src/main/java/*: Contém o código-fonte principal da aplicação Java.

*com.nataliatsi.mavis*: Pacote base da aplicação, organizando a estrutura de pacotes conforme os componentes do projeto.

<sup>6</sup> Elaborado por Natália S. Gomes e João Igor.

*config/*: Arquivos de configuração da aplicação.

*controller/*: Classes que definem os endpoints da API e processam as requisições HTTP.

*dto/*: Data Transfer Objects (DTOs), usados para transferir dados entre as camadas da aplicação.

*entities/*: Classes que representam as entidades do banco de dados (modelos de dados).

*exception/*: Classes que definem exceções personalizadas da aplicação.

*mapper/*: Interfaces que mapeiam objetos entre as camadas utilizando o MapStruct.

*repository/*: Interfaces que definem métodos para interagir com o banco de dados.

*security/*: Classes relacionadas à segurança da aplicação, como autenticação e autorização.

*service/*: Implementações da lógica de negócios da aplicação.

*utils/*: Classes utilitárias com funcionalidades genéricas utilizadas em toda a aplicação.

*MavisApplication*: Classe principal da aplicação, com o método main.

*src/main/resources/*: Contém os recursos da aplicação, incluindo configurações e arquivos estáticos.

*db/migration/*: Scripts de migração para controlar as mudanças no banco de dados ao longo do tempo.

*app.key*: Chave privada usada para criptografia.

*app.pub*: Chave pública correspondente à chave privada.

*application.properties*: Arquivo de configuração da aplicação, com dados como informações de conexão com o banco de dados e outras propriedades da aplicação.

*src/test/java/*: Contém os testes unitários e de integração da aplicação.

- Arquivos de Configuração e Build

*.gitattributes*: Define atributos específicos para arquivos e diretórios no Git.

*.gitignore*: Lista arquivos e pastas que devem ser ignorados pelo Git.

*api.Dockerfile*: Define as instruções para construir uma imagem Docker da aplicação.

*build.gradle*: Arquivo de configuração do Gradle, onde são definidas as dependências, plugins e tarefas de build.

*compose.yml*: Arquivo de configuração do Docker Compose, utilizado para gerenciar múltiplos containers Docker.

*gradlew* e *gradlew.bat*: Scripts do Gradle Wrapper, permitindo a execução do Gradle sem precisar instalá-lo globalmente.

*HELP.md*: Documentação padrão gerada pelo Spring Boot.

*README.md*: Arquivo principal de documentação do projeto, com instruções sobre como construir, executar e utilizar a aplicação.

*settings.gradle*: Arquivo de configuração do Gradle, onde são definidos subprojetos (caso existam).

## Frontend

- Diretórios Principais

*.expo/*: Diretório gerenciado pelo Expo, contendo configurações, cache e metadados do projeto.

*assets/*: Armazena recursos estáticos, como imagens, ícones e fontes utilizadas no aplicativo.

*node\_modules/*: Contém todas as dependências e bibliotecas instaladas via npm/yarn.

- Diretório de Código-fonte (src)

*src/*: Diretório principal onde estão os arquivos e componentes da aplicação.

*AdditionalInfoScreen/*: Tela para adicionar informações complementares do usuário.

*EmergencyContactsScreen/*: Tela de gerenciamento de contatos de emergência.

*HomeScreen/*: Tela inicial do aplicativo.

*LoginScreen/*: Tela de login dos usuários.

*MedicalHistoryScreen/*: Tela para visualizar e gerenciar histórico médico.

*ProfileScreen/*: Tela de perfil do usuário.

*RegisterScreen/*: Tela de cadastro de novos usuários.

*WelcomeScreen/*: Tela de boas-vindas para novos usuários.

- Arquivos de Configuração e Build

*.gitignore*: Lista arquivos e diretórios que devem ser ignorados pelo Git.

*App.js*: Arquivo principal do React Native, ponto de entrada do aplicativo.

*app.json*: Configurações do aplicativo, como nome, ícone, splash screen e definições para Android, iOS e Web.

*babel.config.js*: Configuração do Babel para transformar código moderno, incluindo suporte para o NativeWind e JSX.

*global.css*: Arquivo de estilos globais, utilizado pelo NativeWind para processar classes do Tailwind CSS.

*metro.config.js*: Configuração do Metro Bundler, o empacotador de módulos do React Native.

*package-lock.json*: Registra as versões exatas das dependências instaladas.

*package.json*: Define as dependências do projeto, scripts de build e metadados da aplicação.

*tailwind.config.js*: Configuração do Tailwind CSS para definir temas, extensões e comportamento no NativeWind.

## 5. CONCLUSÃO

O presente relatório apresentou o desenvolvimento do projeto MAVIS, cujo objetivo foi criar um serviço que auxilia usuários em situações de emergência por meio de notificações automáticas a contatos cadastrados. O projeto foi implementado utilizando Spring Boot para o backend e React Native para o frontend, garantindo escalabilidade e compatibilidade com dispositivos móveis.

Durante o desenvolvimento, diversos desafios foram enfrentados. No frontend, a principal dificuldade foi a curva de aprendizado do React Native, já que essa foi a primeira experiência da equipe com o framework. No backend, os desafios envolveram a implementação de mecanismos de segurança, como JWT e OAuth2 Resource Server, além da definição de uma arquitetura modular que permitisse a adição de novas funcionalidades sem comprometer a organização do código.

O sistema desenvolvido demonstrou potencial para ser aplicado no auxílio a pessoas que moram sozinhas, fornecendo uma ferramenta simples e eficaz para comunicação em emergências. Futuramente, melhorias podem incluir a integração com IoTs para detecção de eventos críticos, chatbots para auxiliar na interação, integração com o WhatsApp, além da otimização da interface para garantir maior acessibilidade para idosos, tornando-a mais fácil e intuitiva.

Dessa forma, o projeto atendeu aos seus objetivos e estabeleceu bases sólidas para aprimoramentos contínuos.

## REFERÊNCIAS

ARLOW, Jim; NEUSTADT, Ila. UML and the Unified Process: Practical Object Oriented Analysis & Design. Great Britain: Addison-Wesley, 2002.

BRASIL. Governo Federal. Lei Geral de Proteção de Dados Pessoais (LGPD). Disponível em: <https://www.gov.br/esporte/pt-br/acesso-a-informacao/lgpd>. Acesso em: 26 fev. 2025.

COCKBURN, Alistair. Writing Effective Use Cases. New York: Addison-Wesley, 2001.

GITHUB. *Código fonte do back-end do sistema MAVIS*. 2025. Disponível em: <https://github.com/nataliatsi/mavis-backend>. Acesso em: 20 mar. 2025.

GITHUB. *Código fonte do front-end do sistema MAVIS*. 2025. Disponível em: <https://github.com/ignizxl/mavis-app-frontend>. Acesso em: 20 mar. 2025.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA – IBGE. Pesquisa Nacional por Amostra de Domicílios Contínua. Rio de Janeiro: IBGE, 2024. Disponível em: [https://biblioteca.ibge.gov.br/visualizacao/livros/liv102004\\_informativo.pdf](https://biblioteca.ibge.gov.br/visualizacao/livros/liv102004_informativo.pdf). Acesso em: 1 fev. 2025.

JACOBSEN, Ivar; BOOCH, G.; RUMBAUGH, J. The Unified Software Development Process. New York: Addison-Wesley, 1999.

NAÇÕES UNIDAS. Sobre o nosso trabalho para alcançar os Objetivos de Desenvolvimento Sustentável no Brasil. Disponível em: <https://brasil.un.org/pt-br/sdgs>. Acesso em: 26 fev. 2025.

SHNIEDER, Geri; WINTERS, Jason P. Applying Use-Case, A practical Guide. Second Edition. New Jersey: Addison-Wesley, 2001.