UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE
DEPARTAMENTO DE COMPUTAÇÃO

Rafael de Andrade Prenholato

Aplicação Móvel Para Gerenciamento de Dados da Saúde Individual da População

Alegre, ES

2024

Rafael de Andrade Prenholato

Aplicação Móvel Para Gerenciamento de Dados da Saúde Individual da População

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como parte das exigências para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Giuliano Prado de Morais Giglio

Alegre, ES

2024

Rafael de Andrade Prenholato

Aplicação Móvel Para Gerenciamento de Dados da Saúde Individual da População

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em 30 de setembro de 2024.

COMISSÃO EXAMINADORA

Prof. MSc. Giuliano Prado de Morais Giglio
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Clayton Vieira Fraga Filho
Universidade Federal do Espírito Santo

Prof. Dr. Antonio Almeida de Barros Junior
Universidade Federal do Espírito Santo

Resumo

A crescente utilização de dispositivos móveis como ferramentas essenciais no dia a dia tem impulsionado a demanda por soluções de saúde mais acessíveis e eficientes. O presente trabalho descreve o desenvolvimento completo de uma versão móvel do sistema *Web Health Care (WHC)*, projetado para gerenciar o histórico médico da população. A aplicação visa fornecer acesso rápido e seguro às informações dos pacientes, documentando de forma detalhada seus dados médicos ao longo de suas vidas. A aplicação prioriza a legibilidade, uma interface intuitiva e otimizada, melhorando a acessibilidade e eficiência para médicos e pacientes. O projeto compreendeu todas as etapas do desenvolvimento, desde a modelagem de requisitos até a construção de diagramas UML, passando pela prototipagem de alta fidelidade e implementação do código. Além disso, o processo de garantia da qualidade foi explorado, com a realização de testes para avaliar a usabilidade da aplicação e seu nível de aceitação pelos usuários, validando a eficácia da solução desenvolvida. Em suma, este projeto representa um avanço significativo no campo da tecnologia aplicada à saúde, proporcionando um sistema acessível, eficiente e seguro para a gestão de informações médicas. Acredita-se que a aplicação desse sistema em larga escala pode gerar um impacto positivo na qualidade do atendimento médico e na continuidade do cuidado aos pacientes.

Palavras-chave: Aplicativo móvel. Histórico médico. Usabilidade. Dispositivos móveis. React Native

Lista de ilustrações

Figura 1 – Tela do Aplicativo Conecte SUS 17
Figura 2 – Tela do Aplicativo Doctoralia 18
Figura 3 – Exemplo de código com JSX no React. 21
Figura 4 – Funcionamento do DOM Virtual. 22
Figura 5 – Fluxograma de atividades 25
Figura 6 – Gerenciamento de alergias 33
Figura 7 – Diagrama de classes de projeto 36
Figura 8 – Modelo conceitual do banco de dados 37
Figura 9 – Modelo físico do banco de dados 39
Figura 10 – Quadro Kanban 43
Figura 11 – Navegação baseada em arquivos 46
Figura 12 – Splash screen e Onboarding 48
Figura 13 – Login 49
Figura 14 – Páginas de cadastro comuns 50
Figura 15 – Cadastro para Médicos 51
Figura 16 – Cadastro para Pacientes 51
Figura 17 – Home do médico 52
Figura 18 – Home do paciente 52
Figura 19 – Buscar médicos 52
Figura 20 – Buscar pacientes 53
Figura 21 – Listagem de consultas 54
Figura 22 – Cadastro de consultas 54
Figura 23 – Visualizar consultas - Médico 55
Figura 24 – Visualizar diagnóstico - Paciente 55
Figura 25 – Listagem de exames 56
Figura 26 – Cadastro e edição de exames 56
Figura 27 – Listagem de alergias 57
Figura 28 – Cadastro de alergias 57
Figura 29 – Visualizar alergia - Médico 57
Figura 30 – Visualizar alergia - Paciente 57
Figura 31 – Listagem de diagnósticos 58
Figura 32 – Cadastro de diagnósticos 58
Figura 33 – Visualizar diagnóstico - Médico 58
Figura 34 – Visualizar diagnóstico - Paciente 58
Figura 35 – Listagem de cirurgias 59
Figura 36 – Cadastro de cirurgias 59

Figura 37 – Visualizar cirurgia - Médico 60
Figura 38 – Visualizar cirurgia - Paciente 60
Figura 39 – Listagem de prescrições 61
Figura 40 – Cadastro de prescrições 61
Figura 41 – Editar prescrição - Médico 61
Figura 42 – Visualizar prescrição - Paciente 61
Figura 43 – Listagem de histórico familiar 62
Figura 44 – Visualizar histórico - Médico 62
Figura 45 – Visualizar histórico - Paciente 62
Figura 46 – Listagem de dietas. 63
Figura 47 – Visualizar dieta - Médico. 63
Figura 48 – Visualizar dieta - Paciente. 63
Figura 49 – Listagem de biodados. 64
Figura 50 – Visualizar biodados - Médico. 64
Figura 51 – Visualizar biodados - Paciente. 64
Figura 52 – Permissões - Paciente 65
Figura 53 – Histórico de Acessos 65
Figura 54 – Permissões - Médico 66
Figura 55 – Anexar arquivo 67
Figura 56 – Digitalizar documento 67
Figura 57 – Editar Digitalização 67
Figura 58 – Gerenciamento de consultas 88
Figura 59 – Gerenciamento de exames 88
Figura 60 – Gerenciamento de alergias 89
Figura 61 – Gerenciamento de prescrições 89
Figura 62 – Gerenciamento de cirurgias 90
Figura 63 – Gerenciamento de diagnósticos 90
Figura 64 – Gerenciamento de biodados 91
Figura 65 – Gerenciamento de dietas 91
Figura 66 – Gerenciamento de histórico familiar 92
Figura 67 – Gerenciamento de contas 92

Lista de tabelas

Tabela 1 – Diferenças entre React e React Native. 23
Tabela 2 – Requisitos Funcionais 26
Tabela 3 – Requisitos Não Funcionais 32
Tabela 4 – Entidade User - Modelo conceitual 38
Tabela 5 – Tabela Users - Modelo físico 38
Tabela 6 – Questionário SUS traduzido 40
Tabela 7 – Escala de Classificação Curva para o SUS 41
Tabela 8 – Ocorrências de Defeitos e Ações Corretivas 69
Tabela 9 – Roteiro de Teste de Usabilidade - Fluxo do Médico 73
Tabela 10 – Roteiro de Teste de Usabilidade - Fluxo do Paciente 73
Tabela 11 – Pontuações SUS dos Usuários 74
Tabela 12 – *Feedback* dos Usuários sobre a Usabilidade 75
Tabela 13 – Entidade Patient - Modelo conceitual 113
Tabela 14 – Entidade Doctor - Modelo conceitual 113
Tabela 15 – Entidade Prescription - Modelo conceitual 113
Tabela 16 – Entidade Exam - Modelo conceitual 114
Tabela 17 – Entidade Surgery - Modelo conceitual 114
Tabela 18 – Entidade Appointment - Modelo conceitual 114
Tabela 19 – Entidade Allergy - Modelo conceitual 115
Tabela 20 – Entidade MedicationPrescription - Modelo conceitual 115
Tabela 21 – Entidade AccessControl - Modelo conceitual 115
Tabela 22 – Entidade AccessHistory - Modelo conceitual 115
Tabela 23 – Entidade Diets - Modelo conceitual 116
Tabela 24 – Entidade FamilyHistories - Modelo conceitual 116
Tabela 25 – Entidade Biodata - Modelo conceitual 116
Tabela 26 – Tabela Surgeries - Modelo físico 117
Tabela 27 – Tabela Exams - Modelo físico 117
Tabela 28 – Tabela Prescriptions - Modelo físico 118
Tabela 29 – Tabela Patients - Modelo físico 118
Tabela 30 – Tabela Doctors - Modelo físico 118
Tabela 31 – Tabela Appointments - Modelo físico 119
Tabela 32 – Tabela Allergies - Modelo físico 119
Tabela 33 – Tabela MedicationPrescription - Modelo físico 119
Tabela 34 – Tabela AccessControl - Modelo físico 120
Tabela 35 – Tabela AccessHistories - Modelo físico 120
Tabela 36 – Tabela Diets - Modelo físico 120

Tabela 37 – Tabela FamilyHistories - Modelo físico 121
Tabela 38 – Tabela Biodata - Modelo físico 121
Tabela 39 – Questionário respondido - Usuário 01 122
Tabela 40 – Questionário respondido - Usuário 02 123
Tabela 41 – Questionário respondido - Usuário 03 124
Tabela 42 – Questionário respondido - Usuário 04 125
Tabela 43 – Questionário respondido - Usuário 05 126

Lista de abreviaturas e siglas

API	*Application Programming Interface*
CSS	*Cascading Style Sheets*
DOM	*Document Object Model*
GPS	*Global Positioning System*
HTML	*Hyper Text Markup Language*
IHC	Interface Humano-Computador
JSX	JavaScript XML
JSON	*JavaScript Object Notation*
PWA	*Progressive Web Apps*
SGBD	*Sistema de Gerenciamento de Banco de Dados*
SUS	*System Usability Scale*
UML	*Unified Modeling Language*
UI	*User Interface*
WHC	*Web Healthy Care*

Sumário

1 INTRODUÇÃO . . . 13
1.1 O Problema e sua importância . . . 14
1.2 Justificativas . . . 14
1.3 Objetivos . . . 14
1.3.1 Objetivo geral . . . 15
1.3.2 Objetivos específicos . . . 15
1.3.2.1 Desenvolvimento de uma API REST para integração com o aplicativo . . . 15
1.3.2.2 Desenvolver Protótipos das Principais Telas do Aplicativo . . . 15
1.3.2.3 Desenvolvimento de funcionalidades adicionais . . . 15
1.3.2.4 Realizar Testes de Usabilidade com Usuários Reais . . . 15

2 REFERENCIAL TEÓRICO . . . 16
2.1 Aplicativos Semelhantes . . . 16
2.1.1 Conecte SUS . . . 16
2.1.2 Doctoralia . . . 17
2.2 Aplicativos móveis . . . 18
2.2.1 Web Apps . . . 19
2.2.2 Aplicativos Híbridos . . . 19
2.2.3 Aplicativos Nativos . . . 19
2.3 Tecnologias e ferramentas utilizadas . . . 20
2.3.1 Prototipagem . . . 20
2.3.2 Teste de Usabilidade . . . 20
2.3.3 Kanban . . . 20
2.3.4 Typescript . . . 21
2.3.5 React . . . 21
2.3.6 React Native . . . 22

3 METODOLOGIA . . . 24
3.1 Requisitos de sistema . . . 25
3.1.1 Requisitos funcionais . . . 26
3.1.2 Requisitos não funcionais . . . 31
3.2 Diagrama de casos de uso . . . 32
3.2.1 Narrativas de caso de uso . . . 33
3.3 Diagrama de classe . . . 35
3.4 Modelo de banco de dados . . . 35
3.4.1 Modelo Conceitual . . . 37

3.4.2 Modelo físico . . . 38
3.5 Avaliação . . . 39
3.5.1 Testes locais . . . 39
3.5.2 Testes com usuários . . . 40

4 DESENVOLVIMENTO DA APLICAÇÃO . . . 42
4.1 Metodologia ágil . . . 42
4.2 Desenvolvimento da API . . . 43
4.2.1 Escolha das tecnologias . . . 43
4.2.2 Arquitetura da API . . . 44
4.2.2.1 Principais camadas . . . 44
4.2.3 Autenticação e Segurança . . . 45
4.3 Desenvolvimento do aplicativo . . . 45
4.3.1 Navegação . . . 46
4.3.2 Prototipagem . . . 47
4.3.3 Fluxo padrão de CRUDs no aplicativo . . . 47
4.3.4 Principais telas desenvolvidas . . . 47
4.3.4.1 *Splash* e *Onboarding* . . . 48
4.3.4.2 Login e cadastro . . . 49
4.3.4.3 Telas iniciais dos usuários . . . 51
4.3.4.4 Busca de médicos . . . 52
4.3.4.5 Busca de pacientes e solicitação de acesso . . . 53
4.3.4.6 Consultas . . . 53
4.3.4.7 Exames . . . 55
4.3.4.8 Alergias . . . 56
4.3.4.9 Diagnósticos . . . 57
4.3.4.10 Cirurgias . . . 59
4.3.4.11 Prescrições . . . 60
4.3.4.12 Histórico familiar . . . 61
4.3.4.13 Dietas . . . 63
4.3.4.14 Biodados . . . 63
4.3.4.15 Fluxo de Permissões para Pacientes . . . 64
4.3.4.16 Fluxo de Permissões para Médicos . . . 65
4.3.4.17 Anexar arquivos . . . 66

5 GARANTIA DA QUALIDADE E PROCESSO DE TESTES . . . 68
5.1 Testes Manuais . . . 68
5.2 Testes Automatizados . . . 70
5.2.1 Testes Automatizados na API . . . 70
5.2.2 Testes Automatizados no Aplicativo . . . 71

5.3 Testes de Usabilidade . . . 71
5.3.1 Escolha dos Participantes . . . 72
5.3.2 Orientações Fornecidas aos Participantes . . . 72
5.3.3 Roteiro de Teste de Usabilidade . . . 72
5.3.4 Análise dos Resultados dos Testes de Usabilidade . . . 74
5.3.5 Análise Individual dos Usuários . . . 74
5.3.6 Interpretação dos Resultados . . . 75
5.3.7 *Feedbacks* Coletados . . . 75

6 RESULTADOS ALCANÇADOS E CONSIDERAÇÕES FINAIS . . . 77
6.1 Considerações sobre os limites do sistema . . . 78
6.2 Trabalhos Futuros . . . 79
6.3 Conclusão . . . 80

7 CRONOGRAMA . . . 82

REFERÊNCIAS . . . 83

APÊNDICES 87

APÊNDICE A – DIAGRAMAS DE CASO DE USO . . . 88

APÊNDICE B – NARRATIVAS DE CASO DE USO . . . 93
B.1 UC002 – Cadastrar Paciente . . . 93
B.2 UC003 - Listar Histórico de Acessos . . . 94
B.3 UC004 - Listar solicitações de acesso . . . 94
B.4 UC005 - Solicitar permissão . . . 95
B.5 UC006 - Cadastrar Consulta . . . 95
B.6 UC007 - Editar Consulta . . . 96
B.7 UC008 - Excluir Consulta . . . 96
B.8 UC009 - Visualizar Consulta . . . 97
B.9 UC010 - Cadastrar Exame . . . 97
B.10 UC011 - Editar Exame . . . 98
B.11 UC012 - Excluir Exame . . . 98
B.12 UC013 - Visualizar Exame . . . 99
B.13 UC014 - Cadastrar Alergia . . . 99
B.14 UC015 - Editar Alergia . . . 99
B.15 UC016 - Excluir Alergia . . . 100
B.16 UC017 - Visualizar Alergia . . . 100

B.17 UC018 - Cadastrar Prescrição 101
B.18 UC019 - Editar Prescrição 101
B.19 UC020 - Excluir Prescrição 102
B.20 UC021 - Visualizar Prescrição 102
B.21 UC022 - Cadastrar Diagnóstico 103
B.22 UC023 - Editar Diagnóstico 103
B.23 UC024 - Excluir Diagnóstico 104
B.24 UC025 - Visualizar Diagnóstico 104
B.25 UC026 - Cadastrar Cirurgia 105
B.26 UC027 - Editar Cirurgia 105
B.27 UC028 - Excluir Cirurgia 106
B.28 UC029 - Visualizar Cirurgia 106
B.29 UC030 - Cadastrar Histórico Familiar 107
B.30 UC031 - Editar Histórico Familiar 107
B.31 UC032 - Excluir Histórico Familiar 108
B.32 UC033 - Visualizar Histórico Familiar 108
B.33 UC034 - Cadastrar Dieta 108
B.34 UC035 - Editar Dieta 109
B.35 UC036 - Excluir Dieta 109
B.36 UC037 - Visualizar Dieta 110
B.37 UC038 - Cadastrar Biodado 110
B.38 UC039 - Editar Biodado 111
B.39 UC040 - Excluir Biodado 111
B.40 UC041 - Visualizar Biodado 112

APÊNDICE C – DESCRIÇÃO DAS TABELAS DO MODELO CONCEITUAL DO BANCO DE DADOS 113

APÊNDICE D – DESCRIÇÃO DAS TABELAS DO MODELO FÍSICO DO BANCO DE DADOS 117

APÊNDICE E – QUESTIONÁRIOS RESPONDIDOS 122

1 Introdução

A transição para o armazenamento digital do histórico médico tem revolucionado a forma como acessamos informações de saúde, tornando-as disponíveis de forma instantânea e em qualquer lugar. Conforme ressaltado por Patricio et al. (2011), o prontuário eletrônico, ao otimizar o trabalho médico, proporciona acesso ágil às informações do paciente, documentando de maneira abrangente os atendimentos prestados ao longo de sua vida. Além disso, essa transição promove a legibilidade dos dados e a integração eficiente com outros sistemas de informação.

Entretanto, o verdadeiro potencial dessa mudança se manifesta quando essas informações são centralizadas e acessíveis a todos os profissionais que delas necessitam. A centralização propicia uma melhoria substancial na qualidade dos cuidados de saúde, resultando em diagnósticos mais precisos. Adicionalmente, ela reduz a necessidade de repetição de exames, poupando tempo e recursos. A comunicação interprofissional é aprimorada, permitindo uma coordenação de cuidados com maior precisão e consistência nas decisões clínicas, o que, por sua vez, diminui a ocorrência de erros médicos.

O estudo de Depolli (2022) apresenta o *Web Health Care* (WHC), um sistema web projetado para gerenciar os registros de saúde individuais da população. O principal objetivo do WHC é centralizar o histórico médico de cada paciente, garantindo acesso fácil para o próprio paciente e profissionais de saúde que o atendam. Isso tem o propósito de simplificar e agilizar o atendimento médico, minimizando a redundância de informações, permitindo diagnósticos mais rápidos e precisos, além de contribuir para a prevenção de doenças.

No contexto atual de transformação digital e a disseminação de dispositivos móveis, como *smartphones*, que desempenham um papel fundamental na vida do ser humano, oferecendo conveniência, flexibilidade e acesso imediato a informações em tempo real de qualquer lugar, é imperativo que o WHC se adapte a essa realidade. Portanto, este trabalho propõe o desenvolvimento de uma versão móvel para o WHC, aprimorando a usabilidade e acessibilidade. Além disso, foram integradas melhorias no sistema existente e introduzidos recursos inovadores para atender às demandas emergentes na área da saúde.

A estratégia de desenvolvimento de aplicativos nativos tem a vantagem de permitir o pleno aproveitamento dos recursos dos dispositivos, uma vez que as funcionalidades dos sistemas operacionais são acessadas diretamente. Isso não apenas resulta em melhor desempenho das aplicações, mas também elimina a necessidade de intermediários entre as instruções e o sistema operacional, conforme mencionado por Tostes e Costa (2020). No entanto, a escolha de desenvolvimento nativo geralmente implica na criação de

aplicativos específicos para cada sistema operacional (Android e iOS). Neste cenário, o uso de *frameworks* de desenvolvimento móvel híbrido apresenta uma solução alternativa, permitindo a abstração das diferenças entre as plataformas e a utilização de recursos nativos no desenvolvimento, conforme destacado por Cunha (2023). Assim, este trabalho visa o desenvolvimento de um aplicativo móvel para o WHC, utilizando um *framework* de desenvolvimento híbrido, unificando a experiência do usuário em ambos os sistemas operacionais e desfrutando das funcionalidades nativas disponíveis.

1.1 O Problema e sua importância

A crescente adoção de dispositivos móveis como parte integrante da vida cotidiana impulsiona a demanda por soluções móveis de gerenciamento de saúde. Essas soluções proporcionam acesso ágil às informações do paciente, documentando de maneira abrangente os atendimentos prestados ao longo de sua vida, promovendo a legibilidade dos dados e uma integração eficiente com outros sistemas. Dessa forma, são capazes de facilitar o acesso do paciente e do profissional às informações de saúde, o que tem implicações diretas na qualidade dos diagnósticos e tratamentos. Além disso, ao manter as informações precisas do paciente reunidas, organizadas e facilmente acessíveis, elas reduzem a necessidade de repetição de exames desnecessários, economizando tempo, recursos e contribuindo para a redução de erros em diagnósticos médicos.

1.2 Justificativas

Espera-se que a criação do aplicativo móvel para estender o WHC, juntamente com a introdução de novas funcionalidades e a adaptação dos requisitos e casos de uso, possa melhorar significativamente o acesso, a usabilidade e a eficiência da plataforma, resultando em uma experiência mais satisfatória para os usuários. A escolha de um *framework* de desenvolvimento híbrido permite economizar tempo e recursos, uma vez que um único código base pode ser utilizado para criar versões para múltiplas plataformas, proporcionando uma experiência consistente aos usuários em diferentes dispositivos e sistemas operacionais. Acredita-se que este trabalho tem o potencial de melhorar significativamente a qualidade da assistência médica, oferecer maior praticidade aos pacientes e profissionais de saúde, e promover a eficiência na gestão das informações médicas, impactando positivamente o setor de saúde como um todo.

1.3 Objetivos

Os objetivos esperados com a conclusão do trabalho são apresentados nas seções abaixo, bem como os conceitos que motivaram a elaboração do mesmo.

1.3.1 Objetivo geral

O principal objetivo deste trabalho é desenvolver uma versão móvel do WHC (*Web Healthy Care*), um sistema web de gerenciamento de dados da saúde individual da população, conforme previamente construído por Depolli (2022). Essa versão tem como objetivo fundamental aprimorar o sistema, tornando-o ainda mais acessível e eficiente para os usuários.

1.3.2 Objetivos específicos

No contexto do desenvolvimento do aplicativo móvel para o WHC, aliado à incorporação de novas funcionalidades ao sistema, torna-se essencial abordar diversos aspectos que visam aprimorar tanto a experiência do usuário quanto as funcionalidades do sistema. Neste sentido, os objetivos específicos a serem alcançados estão detalhados a seguir.

1.3.2.1 Desenvolvimento de uma API REST para integração com o aplicativo

Será desenvolvida uma API REST utilizando a tecnologia *Node.js*, visando a integração com o aplicativo móvel. Esta API será responsável por fornecer serviços e dados do sistema web originalmente construído em *Ruby on Rails*, garantindo comunicação eficiente, segura e escalável entre o *backend* e o *frontend* móvel. A escolha por *Node.js* se deve à sua capacidade de lidar com operações assíncronas de forma eficiente, além de proporcionar alta performance e facilidade de integração com outras tecnologias.

1.3.2.2 Desenvolver Protótipos das Principais Telas do Aplicativo

A criação de protótipos detalhados das principais telas do aplicativo permitirá uma visualização clara da versão final, facilitando o processo de desenvolvimento e a comunicação com os usuários.

1.3.2.3 Desenvolvimento de funcionalidades adicionais

Serão implementadas funcionalidades adicionais no aplicativo móvel, além do reajuste de funcionalidades existentes no sistema *web*. Entre as novas funcionalidades, está a capacidade de digitalização de documentos a partir do dispositivo móvel. Isso simplificará o processo de atualização de registros médicos e permitirá que os usuários armazenem documentos importantes, como exames e receitas, de forma segura e acessível.

1.3.2.4 Realizar Testes de Usabilidade com Usuários Reais

A realização de testes de usabilidade com usuários reais é essencial para avaliar a eficácia do aplicativo, assegurando que ele seja intuitivo e atenda efetivamente às necessidades dos usuários.

2 Referencial Teórico

O propósito do referencial teórico é proporcionar uma compreensão mais aprofundada deste projeto, ao abordar o conteúdo teórico subjacente. Este capítulo contemplará a exposição de softwares semelhantes que já existem em relação ao proposto por este trabalho, bem como uma análise abrangente das aplicações móveis. Além disso, será discutido o conjunto das tecnologias e ferramentas que serão empregadas no desenvolvimento da aplicação proposta.

2.1 Aplicativos Semelhantes

Em seu estudo, Depolli (2022) apresentou uma análise abrangente de softwares que se assemelham à proposta do WHC, com ênfase em sistemas baseados na web. Além disso, foram mencionados alguns aplicativos, como o Histórico Médico da MedClin (MEDCLIN, 2023). Além dos aplicativos móveis citados por Depolli (2022), é importante destacar mais dois aplicativos que foram identificados durante a pesquisa conduzida neste estudo: O Conecte SUS e o Doctoralia.

2.1.1 Conecte SUS

Conecte SUS (GOV, 2023) é um aplicativo, do Ministério da Saúde, que registra toda a trajetória de quem busca atendimento no Sistema Único de Saúde (SUS) (BRASIL, 2021). O cadastro é feito com o número do CPF ou da Carteira Nacional de Saúde do paciente, também sendo possível a conexão pelo Portal Gov.br. A Figura 1 ilustra uma tela do aplicativo.

Dentre as funcionalidades presentes no Conecte SUS estão:

- Visualizar histórico de saúde;
- Visualizar medicamentos, consultas e exames;
- Verificar quais hospitais, postos de saúde e serviços de urgência estão mais próximos;
- Marcar consultas de rotina;
- Ter acesso a informações importantes sobre saúde;
- Ter acesso à Carteira Nacional Digital de Vacinação; e
- Tirar dúvidas sobre o SUS.

Uma desvantagem do Conecte SUS é a sua limitação em acomodar exclusivamente registros relacionados ao Sistema Único de Saúde (SUS). Isso implica que informações de saúde provenientes de outras instituições não são incorporadas, o que pode criar desafios significativos no processo de revisão do histórico médico e diagnóstico por parte dos médicos.

Figura 1 – Tela do Aplicativo Conecte SUS

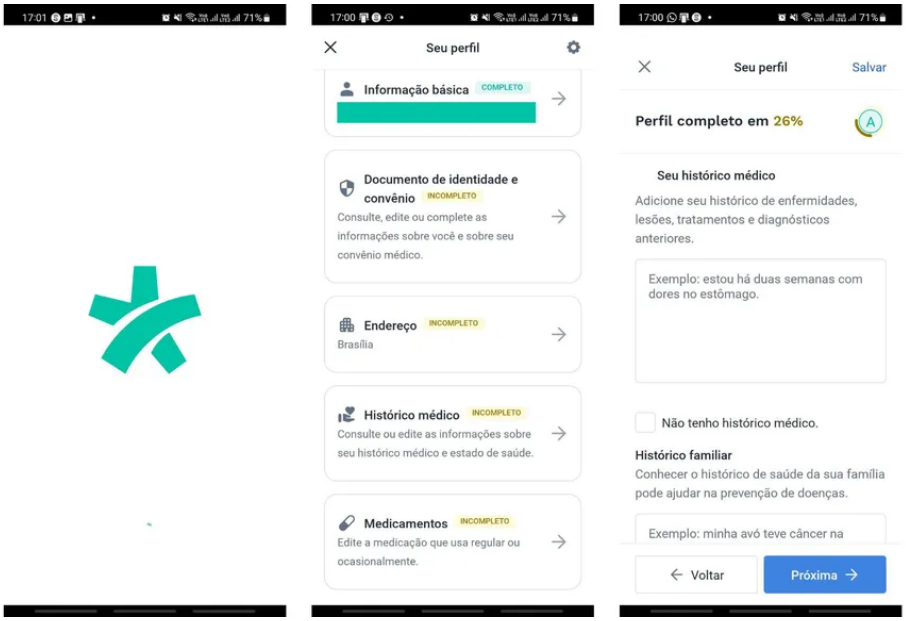
Fonte: (GOV, 2023).

2.1.2 Doctoralia

O Doctoralia (DOCTORALIA, 2023) é uma plataforma que conecta e aproxima pacientes dos profissionais de saúde. A plataforma, além de outros recursos, possui uma funcionalidade que permite registrar e monitorar o histórico de saúde por meio do aplicativo. Essa funcionalidade permite compartilhar facilmente dados entre médicos e, ao mesmo tempo, oferece aos usuários um maior controle sobre suas informações pessoais (LISBOA, 2023). Vale ressaltar que apenas os profissionais e organizações que fazem parte dos planos pagos têm acesso ao histórico de pacientes. A Figura 2 ilustra algumas telas do aplicativo.

Diferentemente do WHC, o Doctoralia permite que os pacientes escolham quais exames, consultas e informações desejam compartilhar com seus médicos, dificultando o acesso completo às informações médicas. Além disso, por ser uma ferramenta de natureza paga e privada, pode haver lacunas nas informações de organizações ou profissionais de saúde que ainda não adotaram a plataforma, como é o caso de hospitais públicos.

Figura 2 – Tela do Aplicativo Doctoralia



Fonte: (LISBOA, 2023).

2.2 Aplicativos móveis

O número de usuários de dispositivos móveis vem crescendo a cada dia. O telefone deixou de ser apenas um acessório útil e passou a fazer parte da vida do ser humano, indispensável aos negócios. Tanto que marcas e empresas ligadas a telefonia não param de lançar atualizações e criar novas versões, movimentando a economia (URMOBO, 2023). De acordo com o portal Teleco (Inteligência em Telecomunicações) e dados da ANATEL (Agência Nacional de Telecomunicações), o número de celulares ativos no Brasil em Agosto de 2023 era de 252,4 milhões, totalizando cerca de 11,7 celulares a cada 10 habitantes (TELECO, 2023).

Com a evolução da tecnologia de redes e serviços, assim como de novos aplicativos, a possibilidade de trazer, para um celular, tarefas como conectar-se à internet, controlar contas bancárias ou, até mesmo, divertir-se, relacionar-se, procurar um bom lugar para jantar, ouvir música, jogar e gravar vídeos e fotos ficou muito mais acessível (ROMAN; GONZALEZ-MESONES; MARINAS, 2007).

Aplicativos móveis são softwares instalados em dispositivos móveis, como *smartphones* e *tablets* que podem funcionar *online* e/ou *offline*. O desenvolvimento de aplicativos móveis diferencia-se do desenvolvimento de outros tipos de software por possuir particularidades e restrições. Os desenvolvedores devem ter em mente aspectos como as capacidades do dispositivo móvel, a mobilidade, as especificações dos dispositivos, o design e navegabilidade de interface do usuário, segurança e privacidade do usuário (EL-KASSAS et al., 2015).

2.2.1 Web Apps

Web Apps são aplicações que são executadas através do próprio navegador. Oferecem, portanto, uma experiência muito parecida com a do app, porém funciona a partir dos recursos do navegador (MAGALHÃES, 2021). Estes aplicativos são desenvolvidos com tecnologias voltadas para a web (HTML, CSS e Java Script) e aperfeiçoados para funcionar como um aplicativo. Existem também os *Progressive Web Apps* (PWA), estes são web apps que se aproximam ainda mais de um aplicativo nativo, possibilitando até mesmo a instalação na tela inicial.

2.2.2 Aplicativos Híbridos

Uma aplicação híbrida é capaz de combinar os elementos dos aplicativos nativos com o web. Como uma versão nativa, ele pode incorporar recursos do sistema operacional e, como um web app, pode usar tecnologias web compatíveis (MONITORA, 2017). Dessa forma, a criação de aplicativos híbridos tem por finalidade funcionar em qualquer tipo de dispositivo e para as diferentes plataformas como iPhone, Android, Windows Phone etc., utilizando o mesmo código fonte (SILVA; PRADO, 2019).

Os aplicativos híbridos geralmente são mais fáceis e rápidos de serem desenvolvidos do que os nativos, além de exigirem menos manutenções. Por outro lado, a velocidade do aplicativo híbrido depende completamente da velocidade do navegador do usuário. Isso significa que eles quase nunca serão executados na mesma velocidade que um aplicativo nativo, sendo mais lentos (MONITORA, 2017).

2.2.3 Aplicativos Nativos

Aplicações desenvolvidas nativamente desfrutam de uma integração visual mais profunda com a plataforma do dispositivo. Isso ocorre porque os componentes visuais, como botões, menus e listas, são fornecidos e estilizados pela própria plataforma, resultando em uma harmonia estética e de interação (SILVA; PRADO, 2019).

Aplicações nativas tem fácil acesso, por meio de APIs fornecidas pelas plataformas, a recursos dos dispositivos móveis, como sensores, câmera, GPS, contatos e e-mail (EL-KASSAS et al., 2015). Isso possibilita a criação de aplicativos significativamente mais sofisticados em termos de funcionalidade e integração com o dispositivo móvel (SILVA; PRADO, 2019).

A grande vantagem de optar por uma aplicação nativa é por ela ser mais rápida e confiável quando se trata de estratégias para trabalhar a experiência do usuário (MONITORA, 2017).

2.3 Tecnologias e ferramentas utilizadas

Na concepção de um novo software, a seleção das tecnologias e ferramentas desempenha um papel crucial. Essas escolhas têm um impacto direto na eficiência do desenvolvimento e na qualidade dos resultados entregues. Nesta seção serão apresentadas as tecnologias e ferramentas adotadas para este projeto, cuja seleção é baseada em pesquisas e conhecimento técnico do autor.

2.3.1 Prototipagem

A prototipagem é um processo experimental em que as equipes de design implementam ideias em formas tangíveis, do papel ao digital (INTERACTION DESIGN FOUNDATION, 2019). Esta técnica apresenta-se como uma valiosa ferramenta para comunicação, verificação do *design* e *feedback* dos usuários. A ausência da prototipagem pode levar a entrega de um produto que não corresponde às expectativas dos usuários, ocasionando prejuízos em termos de tempo e recursos (FERNANDES, 2023).

2.3.2 Teste de Usabilidade

A usabilidade em aplicações de dispositivos móveis é de grande importância, pois minimiza a dificuldade de interação dos usuários(BONIFÁCIO et al., 2010). O teste de usabilidade é a prática de testar a facilidade com que um design é usado com um grupo de usuários representativos (INTERACTION DESIGN FOUNDATION, 2019), definindo o nível de sucesso da aplicação, apontando possíveis falhas e pontos de melhoria.

2.3.3 Kanban

A metodologia ágil é um conjunto de ideais e princípios para o gerenciamento e desenvolvimento de software (MESH, 2020). O método Kanban é um conjunto de princípios e práticas que tem o objetivo de proporcionar uma evolução na forma que sua equipe (ou empresa) entrega valor nos serviços prestados (SABINO, 2023). O Kanban nasceu em uma fábrica da Toyota e se popularizou por times ágeis de desenvolvimento de software graças a sua flexibilidade, velocidade e eficiência (MESH, 2020). Como o Kanban é um fluxo de trabalho visual, ele usa um quadro (físico ou digital) para planejar e acompanhar as tarefas. Um quadro Kanban usa cartões, colunas e o conceito de melhoria contínua para controlar e engajar a equipe no trabalho que precisa ser finalizado regularmente.

Um dos fluxos de trabalho Kanban mais simples e amplamente utilizado consiste em três colunas: "A fazer", "Fazendo"e "Feito". Dependendo da complexidade do projeto, podem ser adicionadas outras colunas para melhor adequar o sistema às necessidades específicas da equipe. Este método ajuda as equipes a gerenciar suas tarefas com clareza e eficiência, promovendo uma abordagem ágil e orientada para resultados.

2.3.4 Typescript

TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft. Adiciona tipagem estática ao JavaScript, linguagem que, por padrão, possui tipagem dinâmica, ou seja, as variáveis e funções podem assumir tipos distintos durante o tempo de execução (FERNANDES, 2018). É considerado por muitos como um superconjunto de JavaScript, pois o código é "transformado" (no termo técnico: transcompilado) em JavaScript antes de ser executado (MELO, 2020). Além de ajudar no ambiente de desenvolvimento, o TypeScript ainda permite que utilizemos funcionalidades da linguagem que ainda não estão disponíveis de forma nativa (FERNANDES, 2018).

2.3.5 React

O React (também conhecido como React.js ou ReactJS) é uma biblioteca JavaScript criada pelo Facebook (atual Meta). Possui código aberto e é destinada à criação de interfaces de usuário (FACEBOOK, 2023a).

Por ser uma biblioteca modular, os componentes podem ser facilmente reutilizados e compartilhados entre diferentes partes da aplicação. Para criar interfaces de usuário, o React usa uma biblioteca declarativa baseada em XML, chamada JSX (JavaScript XML). O JSX permite a integração de código JavaScript com a sintaxe do HTML, o que torna mais fácil escrever componentes de interface de usuário em um único arquivo de código(NEVES, 2023). A Figura 3 demonstra um componente básico de um botão em React usando JSX.

Figura 3 – Exemplo de código com JSX no React.

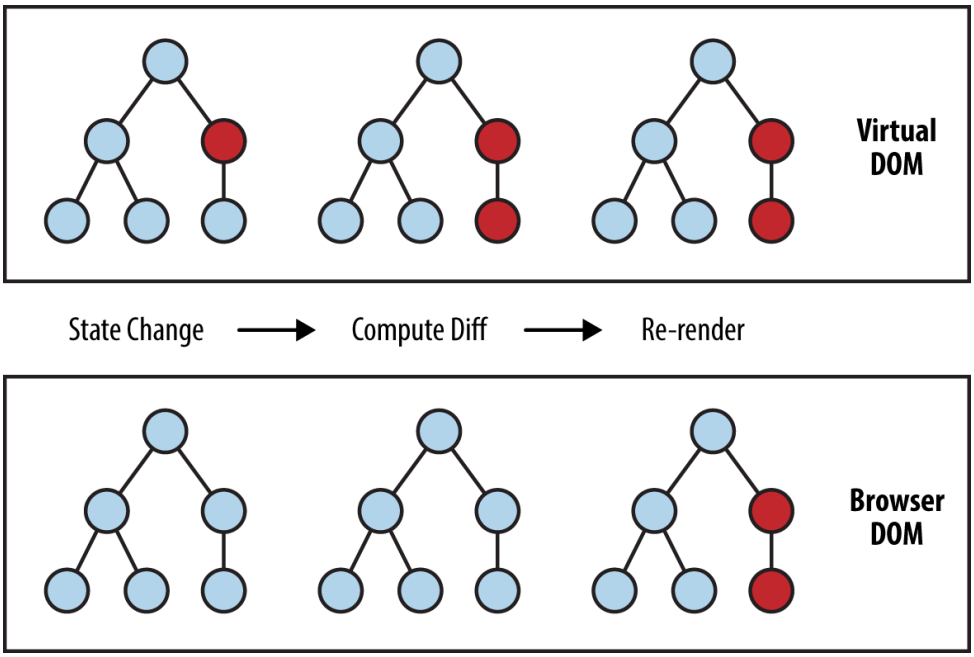
```
1 function Botao(props) {
2     return (
3       <button onClick={props.
  onClick}>
4         {props.texto}
5       </button>
6     );
7 }
```

Fonte: O autor.

Para resolver o problema de lentidão na atualização do DOM (WHATWG COMMUNITY, 2023), comum em muitos *frameworks*, o React adota a abordagem do DOM

Virtual. O DOM Virtual é uma representação em memória do DOM real, mantida em sincronia com este último. Quando um elemento JSX é renderizado, o React atualiza o DOM Virtual de forma eficiente, resultando em um custo insignificante. Após essa atualização, o React compara o DOM Virtual com seu estado anterior para identificar quais elementos foram efetivamente modificados. Em seguida, o React atualiza apenas os elementos que sofreram alterações no DOM real. Isso otimiza o desempenho e garante que apenas as partes necessárias da interface do usuário sejam atualizadas, minimizando a sobrecarga no navegador(CODECADEMY, 2017). A Figura 4 ilustra o procedimento de atualização tanto do DOM Virtual quanto do DOM Real executado pelo React. Os círculos vermelhos representam os nós que mudaram. Esses nós representam os elementos da UI que tiveram seu estado alterado. A diferença entre a versão anterior da árvore DOM virtual e a árvore DOM virtual atual é então calculada. A subárvore pai inteira é então re-renderizada. Esta nova árvore é então atualizada em lote para o DOM real. Dessa forma, o React provê um maior controle nas alterações visuais e, eventualmente, pode melhorar o tempo de renderização das aplicações (BANKS; PORCELLO, 2017)

Figura 4 – Funcionamento do DOM Virtual.



Fonte: (EISENMAN, 2016)

2.3.6 React Native

React Native é um *framework* que usa a estrutura de UI declarativa do React no desenvolvimento de aplicações multiplataforma (FACEBOOK, 2023b). Permite criar aplicações para Android, IOS, Windows, MacOS, Web e outras plataformas[1] de forma nativa.

[1] Disponível em: <https://reactnative.dev/docs/out-of-tree-platforms>

Os princípios de funcionamento do React Native são praticamente idênticos aos do React, exceto que o React Native não manipula o DOM através do DOM Virtual (EISENMAN, 2016), utilizando ferramentas nativas para renderizar os componentes e proporcionando uma experiência de usuário mais autêntica (CUNHA, 2023). O DOM virtual é representado diretamente nos componentes de apresentação nativos do sistema operacional onde a aplicação está sendo executada (TOSTES; COSTA, 2020). React Native não usa HTML ou CSS (EISENMAN, 2016). A estilização da aplicação é feita com código Javascript, com uma sintaxe semelhante ao CSS tradicional.

Apesar de haverem semelhanças, o React Native é diferente do ReactJS. A Tabela 1 apresenta as principais diferenças entre o React e o React Native

Tabela 1 – Diferenças entre React e React Native.

React	**React Native**
Uma biblioteca JavaScript.	Um framework JavaScript.
Ideal para construir aplicações web dinâmicas.	Dá uma sensação nativa à interface de usuário de aplicativos móveis.
Usa o Virtual DOM para renderizar o código do navegador.	Usa APIs nativas para renderizar código em dispositivos móveis.
Suporta CSS para criar animações.	Requer a API Animated para animar componentes.
Usa tags HTML.	Não usa tags HTML devido à falta de suporte a DOM.
Usa CSS para estilizar.	Usa a folha de estilo JS para estilizar.

Fonte: (LONGEN, 2023).

O desenvolvimento de aplicações nativas com React Native é consideravelmente mais rápido, possuindo o reaproveitamento do código gigantesco que não existiria se fosse desenvolvido em Objective-C para iOS e Java para Android (KUPKA, 2017).

3 Metodologia

Esta seção descreve a metodologia adotada para o desenvolvimento do aplicativo móvel que estende o WHC, originalmente desenvolvido por Depolli (2022). O objetivo principal deste projeto é fornecer uma solução móvel que aprimore a acessibilidade e a experiência do usuário, estendendo as funcionalidades do sistema *web* para dispositivos móveis. A metodologia adotada reflete a abordagem para alcançar esse objetivo.

O desenvolvimento do aplicativo móvel foi realizado utilizando o *framework React Native*. Essa escolha se baseia na vantagem de criar aplicativos híbridos para dispositivos móveis, tirando o máximo proveito dos recursos do dispositivo e garantindo desempenho otimizado. Além disso, a utilização de *React Native* simplifica o desenvolvimento multiplataforma, permitindo que o aplicativo seja executado em sistemas Android e iOS utilizando o mesmo código-fonte.

Uma etapa fundamental do desenvolvimento envolve a criação de uma nova API REST utilizando *Node.js* para atender ao aplicativo móvel. A aplicação *web* original, construída com o *framework Ruby on Rails*, foi projetada para fornecer páginas HTML em resposta às solicitações. Entretanto, para integrar o novo aplicativo móvel, foi necessário desenvolver uma API que forneça respostas em um formato adequado para a comunicação com o aplicativo móvel, como JSON. Esta nova API, que segue o padrão REST, permite a comunicação entre o aplicativo móvel e o banco de dados, proporcionando eficiência e escalabilidade. A escolha do *Node.js* se justifica por sua alta performance em operações assíncronas e pela facilidade de integração com outras tecnologias.

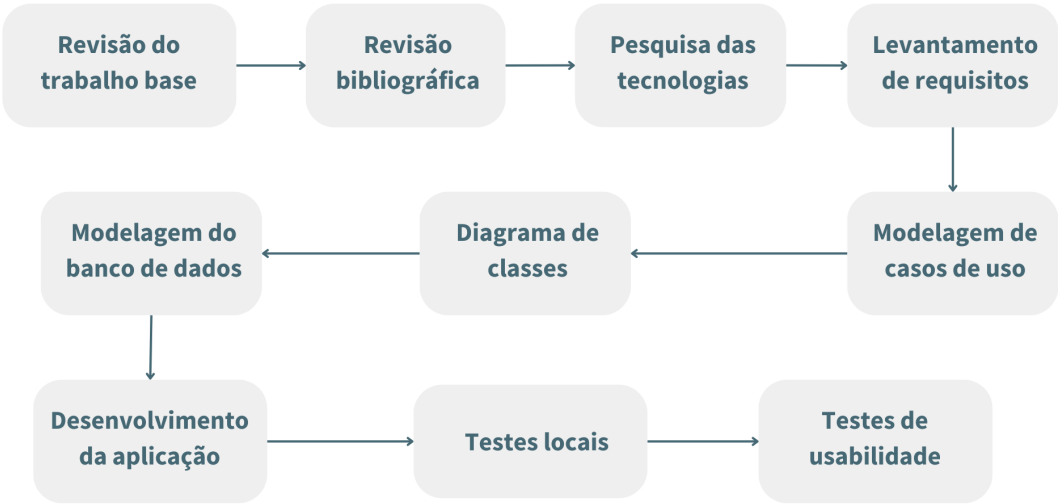
A escolha da metodologia ágil Kanban é fundamentada nas características únicas deste projeto. Como o desenvolvimento será conduzido de forma independente, o Kanban oferece uma abordagem flexível e adaptável para gerenciar as tarefas e os fluxos de trabalho. O Trello (Trello, 2023) foi empregado como a ferramenta de gerenciamento de cartões Kanban, permitindo a criação, atribuição, acompanhamento e conclusão de tarefas de desenvolvimento de maneira organizada e eficiente.

Para a fase de prototipagem, a ferramenta Figma (Figma, 2023) foi utilizada para criar protótipos interativos de alta fidelidade do aplicativo móvel. A prototipagem desempenha um papel crucial na definição da aparência e da usabilidade do aplicativo, permitindo a validação de conceitos e fluxos antes da implementação. O Figma proporciona colaboração eficiente, permitindo a revisão e o *feedback* de partes interessadas, o que é fundamental para o refinamento contínuo do design do aplicativo.

A Figura 5 apresenta o fluxograma das atividades executadas durante o trabalho, detalhando as principais etapas seguidas na metodologia aplicada. O processo inicia com a

revisão do trabalho base e a revisão bibliográfica, essenciais para fornecer embasamento teórico e identificar soluções e abordagens existentes. Em seguida, realiza-se a pesquisa das tecnologias, para escolher as ferramentas mais adequadas para o desenvolvimento. A etapa de levantamento de requisitos foca na definição das necessidades do sistema, sendo seguida pela modelagem de casos de uso, que descreve as interações do usuário com o sistema. A modelagem de classes e a modelagem do banco de dados estruturam a arquitetura do sistema, permitindo o início do desenvolvimento da aplicação. Após o desenvolvimento, são realizados os testes locais para verificar o funcionamento da aplicação, e, por fim, os testes de usabilidade para avaliar se o sistema atende às expectativas dos usuários. Esse fluxo organizado de atividades permite um desenvolvimento controlado, visando proporcionar maior qualidade das funcionalidades do sistema.

Figura 5 – Fluxograma de atividades



Fonte: O autor.

3.1 Requisitos de sistema

O processo de levantamento de requisitos desempenha um papel crítico no sucesso de qualquer projeto, assegurando que as necessidades do público-alvo sejam adequadamente compreendidas e que o produto final satisfaça os requisitos e expectativas dos usuários. A ausência de um levantamento completo pode resultar em falhas de entrega, custos adicionais e insatisfação por parte dos *stakeholders*.

No contexto deste trabalho, o sistema desenvolvido tem como base o sistema previamente concebido por Depolli (2022). A maior parte dos requisitos já estava delineada em seu trabalho anterior. No entanto, considerando as evoluções e necessidades específicas deste novo contexto, foram identificadas e detalhadas quaisquer novas exigências ou

modificações necessárias.

Esta seção abordará detalhadamente os requisitos de sistema. Isso garantirá que o sistema proposto seja robusto e atenda às demandas da pesquisa em questão.

3.1.1 Requisitos funcionais

Requisitos funcionais são declarações de serviços ou funcionalidades que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações (SOMMERVILLE, 2011). À medida que um sistema evolui, é essencial reconhecer que a melhoria contínua é fundamental para atender às necessidades em constante mudança dos usuários, sejam eles médicos, pacientes ou outros profissionais de saúde. Os requisitos funcionais do aplicativo estão especificados na Tabela 2.

Tabela 2 – Requisitos Funcionais

Código	**Nome**	**Descrição**
RF01	Cadastrar médico	O sistema permitirá o cadastro de médicos, exigindo informações como Nome, E-mail, Senha, Gênero, CPF, Telefone, Foto de perfil, CRM e Especialidade.
RF02	Cadastrar paciente	O sistema permitirá o cadastro de pacientes, exigindo dados como Nome, E-mail, Senha, Gênero, CPF, Telefone, Foto de perfil, Tipo sanguíneo, Cidade e Estado.
RF03	Editar dados de cadastro	Médicos e pacientes poderão atualizar seus dados de cadastro acessando a seção "Perfil"do aplicativo.
RF04	Autenticar usuários	O sistema possibilitará que médicos e pacientes realizem login utilizando suas credenciais, como e-mail e senha, para autenticação segura.
RF05	Manter consultas	O sistema permitirá que médicos realizem o cadastro, leitura, edição e exclusão de consultas, enquanto pacientes poderão apenas visualizar suas próprias consultas. Uma consulta conterá informações como Médico, Data, Sintomas, Queixa principal, Anamnese e Atestado médico.

Continua na próxima página

Código	**Nome**	**Descrição**
RF06	Manter prescrições	O sistema possibilitará que médicos façam o cadastro, leitura, edição e exclusão de prescrições de medicamentos. Pacientes poderão visualizar suas próprias prescrições, que incluem Data, Medicamentos e Observações.
RF07	Manter medicamentos da prescrição	Médicos poderão adicionar, visualizar, editar e remover medicamentos de uma prescrição. Pacientes terão acesso apenas à visualização dos medicamentos prescritos, que incluem Nome e Posologia.
RF08	Manter cirurgias	Médicos poderão cadastrar, visualizar, editar e excluir registros de cirurgias, enquanto pacientes poderão apenas visualizar seus próprios registros. Uma cirurgia inclui Data, Nome, Hospital, Data da alta e uma descrição detalhada do procedimento.
RF09	Manter exames	Médicos poderão cadastrar, visualizar, editar e excluir exames, enquanto pacientes terão acesso somente à visualização de seus próprios exames. Cada exame incluirá Data da solicitação, Nome, Data de realização, Resultado e Anexos.
RF10	Manter anexos de exames	Médicos poderão anexar, visualizar, editar e remover arquivos relacionados a exames. Pacientes poderão apenas visualizar seus anexos. O sistema permitirá anexar arquivos da memória do dispositivo ou digitalizá-los utilizando a câmera.
RF11	Manter diagnósticos	O sistema permitirá que médicos façam o cadastro, leitura, edição e exclusão de diagnósticos. Pacientes poderão apenas visualizar seus diagnósticos, que incluem Data, Nome, Informações adicionais e Situação (Ativo/Inativo).

Continua na próxima página

Código	**Nome**	**Descrição**
RF12	Manter alergias	Médicos poderão registrar, visualizar, editar e excluir alergias de pacientes. Pacientes poderão visualizar suas próprias alergias, que incluem Nome e Descrição.
RF13	Manter histórico familiar	Médicos poderão registrar, visualizar, editar e excluir dados do histórico familiar de pacientes. Pacientes poderão visualizar seus próprios históricos familiares, que incluem Grau de parentesco, Nome do parente e Descrição.
RF14	Manter dietas	Médicos poderão registrar, visualizar, editar e excluir dietas de pacientes. Pacientes poderão visualizar suas próprias dietas, que incluem campos para cada refeição do dia, como café da manhã, lanche da manhã, almoço, lanche da tarde e jantar.
RF15	Manter biodados	Médicos poderão registrar, visualizar, editar e excluir biodados de pacientes. Os biodados incluem registros como pressão sistólica, pressão diastólica, glicemia, frequência cardíaca, colesterol, triglicerídeos e creatina. Os pacientes poderão visualizar os seus registros de biodados.
RF16	Buscar médicos	Pacientes poderão buscar perfis de médicos, visualizar informações e conceder permissões de acesso aos seus dados.
RF17	Buscar pacientes	Médicos poderão buscar perfis de pacientes e, caso tenham permissão, acessar, modificar ou adicionar informações ao histórico médico desses pacientes.
RF18	Solicitar controle de acesso	Médicos poderão solicitar permissão para acessar dados médicos de pacientes, incluindo leitura, edição e adição de novas informações. A permissão não terá prazo de validade.

Continua na próxima página

Código	**Nome**	**Descrição**
RF19	Autorizar ou recusar acesso às informações	Pacientes poderão conceder ou recusar permissões de acesso aos médicos para seus dados. Essa decisão poderá ser tomada em resposta a uma solicitação ou de maneira proativa.
RF20	Gerenciar autorizações de acesso	Pacientes poderão visualizar, adicionar e revogar permissões de acesso concedidas aos médicos por meio de uma interface dedicada.
RF21	Digitalizar documentos	O aplicativo permitirá a digitalização de documentos utilizando a câmera do dispositivo móvel, que poderão ser anexados a exames ou consultas.
RF22	Histórico de acessos	Pacientes poderão visualizar um histórico detalhado de acessos aos seus dados médicos, contendo a Data e Hora do acesso, bem como o Médico responsável, com listagem ordenada cronologicamente.
RF23	Buscar consultas	O sistema permitirá que os usuários busquem consultas pelo nome do médico responsável.
RF24	Buscar exames	O sistema permitirá que os usuários busquem exames pelo nome do exame realizado.
RF25	Buscar alergias	O sistema permitirá que os usuários busquem alergias pelo nome da alergia registrada.
RF26	Buscar prescrições	O sistema permitirá que os usuários busquem prescrições pelo nome dos medicamentos inseridos.
RF27	Buscar cirurgias	O sistema permitirá que os usuários busquem cirurgias pelo nome do procedimento realizado.
RF28	Buscar diagnósticos	O sistema permitirá que os usuários busquem diagnósticos pelo nome da enfermidade diagnosticada.

Continua na próxima página

Código	**Nome**	**Descrição**
RF29	Buscar histórico familiar	O sistema permitirá que os usuários busquem registros do histórico familiar pelo nome do grau de parentesco.
RF30	Buscar dieta	O sistema permitirá que os usuários busquem dietas pelo nome do médico.

Fonte: O autor.

A partir do sistema proposto por Depolli (2022), alguns requisitos funcionais foram ajustados, com o objetivo de aprimorar a usabilidade e a eficiência do *WHC*. Essas modificações foram realizadas com o compromisso de proporcionar uma experiência do usuário mais intuitiva e prática, mantendo a coerência do sistema e ampliando suas funcionalidades.

Uma das principais modificações foi a unificação dos dados de perfil, que anteriormente separava as informações de "médico"e "paciente"dos dados de "usuário". Agora, esses registros foram integrados, permitindo que tanto médicos quanto pacientes editem todas as suas informações de forma simplificada. Essa unificação visa melhorar o processo de edição e proporcionar uma experiência de usuário mais uniforme.

Além disso, novas informações foram incluídas no cadastro de usuários, aumentando a completude dos dados. Entre essas informações, destaca-se a inclusão do campo Telefone, fundamental para uma comunicação mais direta, e do campo Gênero, importante para adequação a diferentes contextos médicos. Especificamente para pacientes, foi incluída a informação sobre o Tipo Sanguíneo, essencial para emergências médicas e tratamentos personalizados, ampliando a segurança dos atendimentos.

Outro aprimoramento relevante foi a adição da funcionalidade de busca em todos os tipos de registros médicos, exceto para biodados. Esse recurso possibilita que os usuários filtrem os dados, promovendo uma navegação mais ágil e facilitando o acesso a informações específicas dentro do sistema.

O gerenciamento de alergias foi inserido como uma nova categoria de registro médico. Esse ajuste atende a uma demanda importante no histórico do paciente, uma vez que a identificação de alergias é crucial para a definição de tratamentos e a prescrição de medicamentos. Com esse recurso, os médicos podem registrar e consultar fatores alérgenos que podem afetar diretamente a saúde dos pacientes, melhorando a segurança e a personalização do atendimento.

A separação das funcionalidades de prescrições de medicamentos e dietas dos registros de cirurgias e diagnósticos foi realizada, criando novos tipos de registros médicos

independentes. Além disso, os biodados foram separados dos exames, permitindo que médicos registrem biodados sem a necessidade de um exame associado.

Foi também incorporado um registro de histórico familiar para que médicos possam acessar rapidamente informações sobre a família do paciente e seu histórico de saúde.

Em relação ao controle de acesso, modificações importantes foram realizadas para simplificar e tornar o sistema mais prático. Anteriormente, médicos autorizados tinham acesso temporário aos registros dos pacientes. No entanto, o conceito de acesso temporário foi removido, e agora o acesso é contínuo, persistindo até que o paciente decida revogá-lo. Essa mudança proporciona uma maior fluidez e continuidade no acompanhamento médico, eliminando a necessidade de constantes renovações de permissões. Essa modificação é especialmente útil em situações em que o paciente está temporariamente incapacitado de acessar o sistema, como em casos de hospitalização ou indisponibilidade.

Outra melhoria significativa foi a ampliação do controle de acesso por parte dos pacientes. Agora, os pacientes têm a capacidade de conceder ou revogar o acesso dos médicos às suas informações de forma direta e imediata, sem depender de uma solicitação explícita por parte do médico. Isso reforça a autonomia do paciente sobre seus dados e agiliza o processo de gerenciamento das autorizações de acesso, proporcionando maior agilidade e controle sobre suas informações médicas.

Além disso, a funcionalidade de digitalização de arquivos foi adicionada, permitindo que os usuários anexem documentos de forma ágil diretamente pelo aplicativo, utilizando a câmera do dispositivo. Essa funcionalidade oferece uma maneira rápida e prática de digitalizar e armazenar documentos médicos, como resultados de exames e atestados, facilitando o acesso e a organização dos dados diretamente no prontuário eletrônico. Com essa melhoria, o fluxo de trabalho médico se torna mais eficiente, eliminando a necessidade de uso de equipamentos externos para a digitalização de documentos.

Essas alterações visam a proporcionar uma experiência mais eficiente, segura e autônoma para os usuários do *WHC*, refletindo um avanço no gerenciamento de dados de saúde e na relação entre médicos e pacientes.

3.1.2 Requisitos não funcionais

Requisitos não funcionais são restrições sobre os serviços ou funcionalidades providos pelo sistema. Essas restrições podem abranger considerações de tempo, limitações no processo de desenvolvimento e conformidade com normas e regulamentos. Diferentemente das características individuais ou serviços do sistema, os requisitos não funcionais frequentemente se aplicam ao sistema como um todo (SOMMERVILLE, 2011).

A tabela 3 mostra os requisitos não funcionais do aplicativo móvel.

Tabela 3 – Requisitos Não Funcionais

Código	**Nome**	**Descrição**
RNF01	Compatibilidade com Múltiplas Plataformas Móveis	O aplicativo móvel deve ser compatível com as principais plataformas móveis, como iOS e Android
RNF02	Privacidade de dados	O sistema deverá garantir a proteção dos dados contra acesso não autorizado.
RNF03	Desempenho e Tempo de Resposta	O aplicativo móvel deve oferecer desempenho rápido e tempo de resposta ágil para garantir uma experiência de usuário satisfatória.
RNF04	Interface Intuitiva	O sistema deverá possuir uma interface intuitiva e de fácil compreensão para o usuário
RNF05	Integração com Recursos do Dispositivo	O aplicativo móvel deve ser capaz de aproveitar os recursos do dispositivo, como a câmera.

Fonte: O autor.

3.2 Diagrama de casos de uso

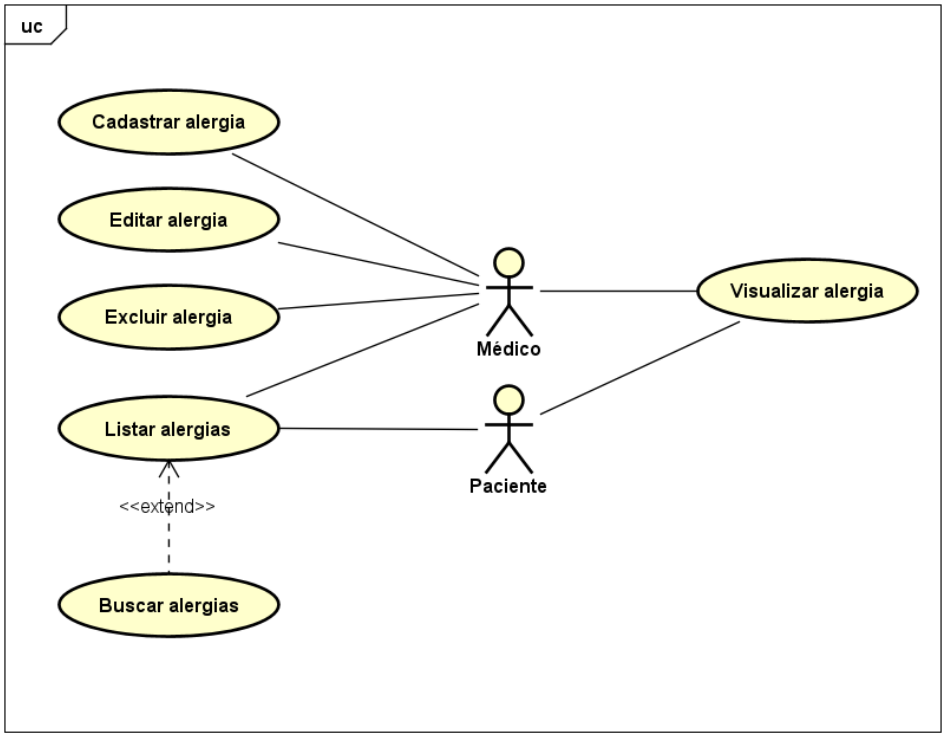
Em sua forma mais simples, um caso de uso identifica os atores envolvidos em uma interação e dá nome ao tipo de interação. Essa é, então, suplementada por informações adicionais que descrevem a interação com o sistema (SOMMERVILLE, 2011).

De acordo com Lima (2011) e Booch, Rumbaugh e Jacobson (2005), o diagrama de casos de uso desempenha um papel central na modelagem do comportamento de um sistema. Ele não apenas identifica os atores e as interações, mas também evidencia as conexões e relações externas com outras entidades que interagem com o sistema. Essa representação gráfica oferece uma visão abrangente de toda a funcionalidade do sistema, tornando-se uma ferramenta valiosa para os desenvolvedores e partes interessadas no processo de desenvolvimento de software.

É importante ressaltar que, à medida que os requisitos do sistema evoluem e se transformam, os casos de uso originais também precisam ser adaptados para acompanhar essas mudanças. Conforme discutido na Seção 3.1, as exigências do sistema podem ser alteradas e expandidas. Nesse contexto, torna-se essencial atualizar e modificar os casos de uso para refletir essas novas demandas. Os diagramas de caso de uso da aplicação foram reformulados de modo a acomodar os novos requisitos e garantir que o sistema continue

atendendo às necessidades dos usuários. A Figura 6 apresenta o diagrama de caso de uso para o gerenciamento de alergias, que também exemplifica as ações comuns realizadas por pacientes e médicos em outros tipos de dados do histórico clínico. O paciente tem a capacidade de visualizar e listar suas próprias alergias, sendo possível realizar buscas durante a listagem. Já o médico, além de possuir as mesmas permissões que o paciente, pode criar, editar e excluir alergias registradas no histórico. Os demais casos podem ser encontradas no Apêndice A.

Figura 6 – Gerenciamento de alergias



Fonte: O autor.

3.2.1 Narrativas de caso de uso

Segundo Melo (2010), um caso de uso é descrito, de forma compreensível tanto para a equipe de desenvolvimento como para os clientes, como sendo uma sequência de ações as quais representam um cenário principal "perfeito" e cenários alternativos, os quais tem como objetivo demonstrar o comportamento de um sistema ou parte dele através de interações com os atores.

Com as alterações realizadas no diagrama de casos de uso, as narrativas de casos de uso também foram ajustadas. A seguir, apresenta-se a narrativa do caso de uso UC001, enquanto as demais narrativas podem ser encontradas no Apêndice B.

UC001 – Cadastrar Médico

Descrição: Este caso de uso detalha o processo de cadastro de um médico no sistema, permitindo que ele crie uma conta e posteriormente acesse suas funcionalidades específicas.

Atores: Médico

Pré-condições: O usuário não deve estar autenticado no sistema.

Pós-condições: Uma conta será criada para o médico, permitindo que ele se autentique no sistema e acesse as funcionalidades destinadas aos médicos.

Fluxo Normal:

1. Na tela de *onboarding*, o usuário seleciona a opção "Criar conta".
2. O sistema redireciona o usuário para o formulário de cadastro de dados básicos, onde ele preenche os seguintes campos:
   - Foto de perfil;
   - Nome;
   - Sobrenome;
   - Gênero;
   - CPF;
   - Telefone.
3. Após preencher os dados, o usuário clica em "Próximo"e é redirecionado para a tela de seleção do tipo de usuário.
4. Na tela de seleção, o usuário escolhe a opção "Médico"e o sistema avança para o próximo formulário de cadastro.
5. O sistema exibe o formulário específico para médicos, onde o usuário deve preencher os seguintes campos:
   - CRM;
   - Especialidade.
6. Após preencher os dados, o usuário clica novamente em "Próximo"e o sistema avança para a etapa final.
7. O usuário define uma senha e a confirma no campo de confirmação de senha. Em seguida, clica no botão "Criar conta".
8. O sistema confirma o sucesso do cadastro exibindo uma mensagem de confirmação e redireciona o usuário para a página de login.

Fluxo de Exceções:

- Se o CPF informado já estiver cadastrado no sistema, o sistema exibe uma mensagem de erro e solicita que o usuário insira um CPF diferente.
- Se as senhas não coincidirem, o sistema exibe uma mensagem de erro e solicita que o usuário insira e confirme a senha corretamente.
- Se o sistema encontrar algum erro de validação nos campos obrigatórios, como falta de preenchimento ou dados inválidos, ele exibe mensagens de erro correspondentes e impede o avanço até que as correções sejam feitas.
- Se o sistema falhar ao criar a conta, o usuário é notificado com uma mensagem de erro genérica.

3.3 Diagrama de classe

Os diagramas de classe são usados no desenvolvimento de um modelo de sistema orientado a objetos para mostrar as classes de um sistema e as associações entre essas classes. De forma simplificada, uma classe de objeto é como um modelo que define as características e comportamentos comuns a todos os objetos de um determinado tipo no sistema. Uma associação é um link entre classes que indica algum relacionamento entre essas classes. Consequentemente, cada classe pode precisar de algum conhecimento sobre sua classe associada. (SOMMERVILLE, 2011)

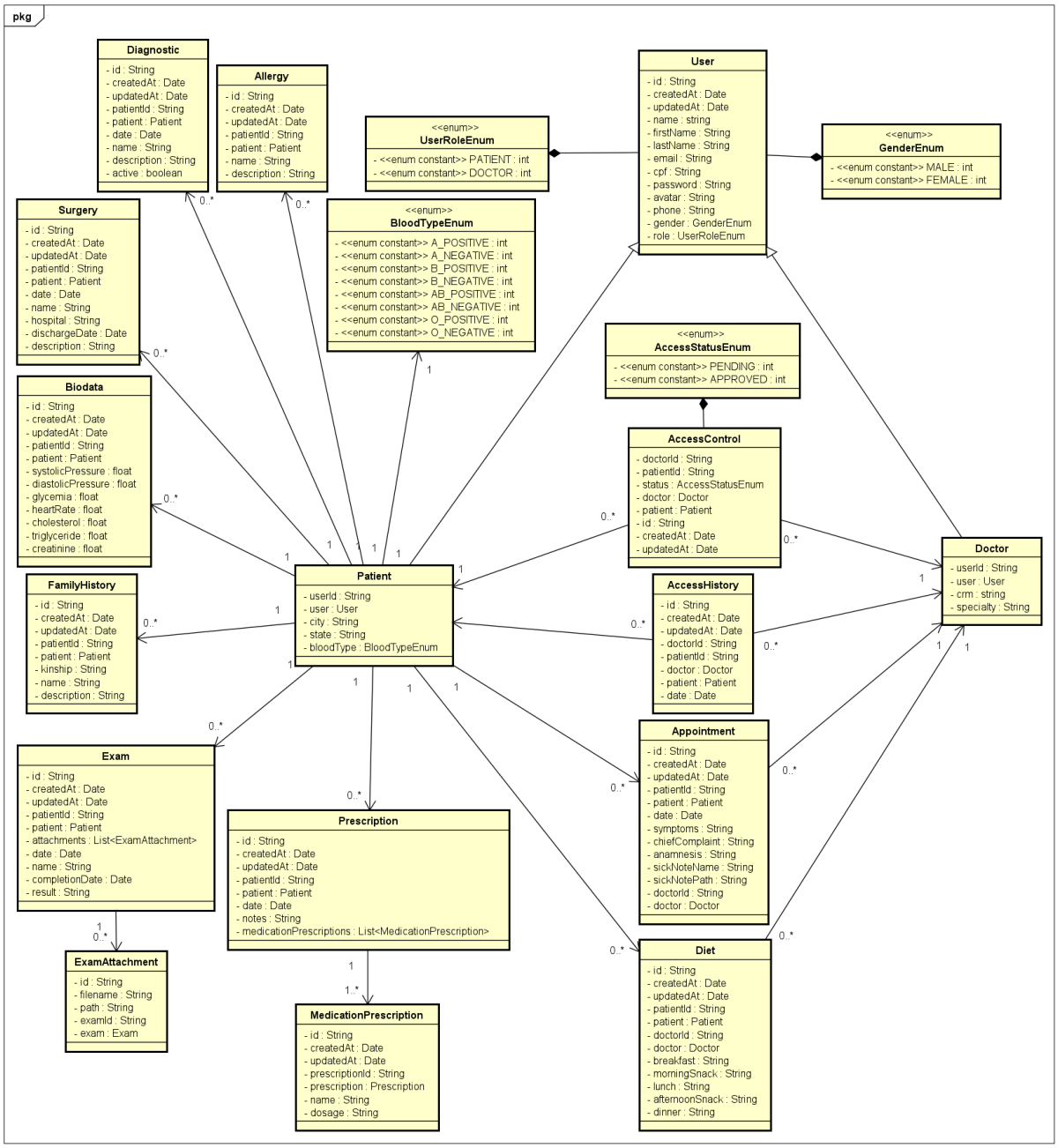
Os diagramas de classe elaborados por Depolli (2022) no trabalho original foram reformulados em virtude da reestruturação dos requisitos do sistema. A Figura 7 apresenta o diagrama de classes atualizado, refletindo as mudanças necessárias para atender às novas especificações e garantir a consistência entre o modelo de dados e as funcionalidades implementadas.

3.4 Modelo de banco de dados

Um modelo de dados é uma coleção de conceitos que podem ser usados para descrever um conjunto de dados e as operações para manipular esses dados (BATINI; CERI; NAVATHE, 1992).

A tarefa de projetar um banco de dados está intrinsecamente relacionada ao ciclo de vida de desenvolvimento de software. A cada fase desse ciclo, novas informações e detalhes são incorporados ao projeto do software (PRESSMAN, 1987). Portanto, os modelos conceituais e lógicos inicialmente propostos por Depolli (2022) para o desenvolvimento do WHC sofreram pequenas alterações.

Figura 7 – Diagrama de classes de projeto



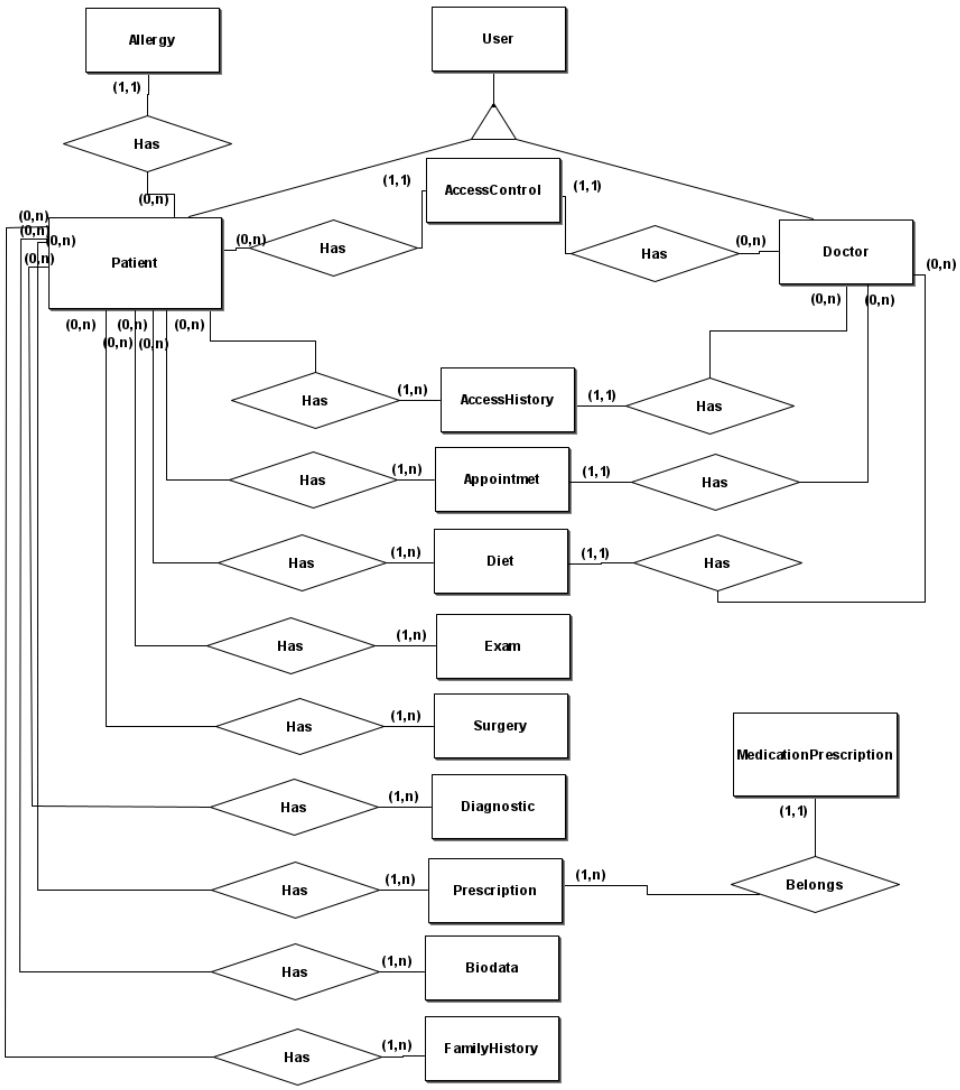
Fonte: O autor.

3.4.1 Modelo Conceitual

A característica fundamental de um modelo de dados, como o próprio termo sugere, é sua capacidade de representar uma abstração da realidade. Um modelo conceitual de dados fornece uma base formal, tanto notacional quanto semântica, que serve como alicerce para as ferramentas e técnicas empregadas na modelagem de dados (FILHO; IOCHPE, 1999).

A Figura 8 ilustra o modelo conceitual do banco de dados reconstruído. As alterações foram implementadas em resposta às mudanças nos requisitos do sistema.

Figura 8 – Modelo conceitual do banco de dados



Fonte: O autor.

Na Tabela 4, apresenta-se uma descrição da entidade *User* do modelo conceitual do banco de dados, os demais serão encontrados no Apêndice C.

Tabela 4 – Entidade User - Modelo conceitual

Coluna	**Tipo**	**Descrição**
id	Texto	Chave primária da tabela
name	Texto	Nome completo do usuário
firstName	Texto	Primeiro nome do usuário
lastName	Texto	Último nome do usuário
email	Texto	E-mail do usuário
cpf	Texto	CPF do usuário
password	Texto	Senha do usuário
avatar	Texto	URL da imagem de avatar (opcional)
role	Enum	Enum com os papéis 'PATIENT' e 'DOCTOR'
phone	Texto	Número de telefone
gender	Enum	Enum com os valores 'MALE' e 'FEMALE'
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

3.4.2 Modelo físico

O modelo físico introduz o contexto específico do banco de dados ausente em modelos conceituais e lógicos de dados. Ele representa as tabelas, colunas, tipos de dados, visualizações, restrições, índices e procedimentos dentro do banco de dados e/ou as informações comunicadas durante os processos de computador. (ERWIN, 2021) A Figura 9 ilustra o modelo físico do banco de dados construído para a aplicação.

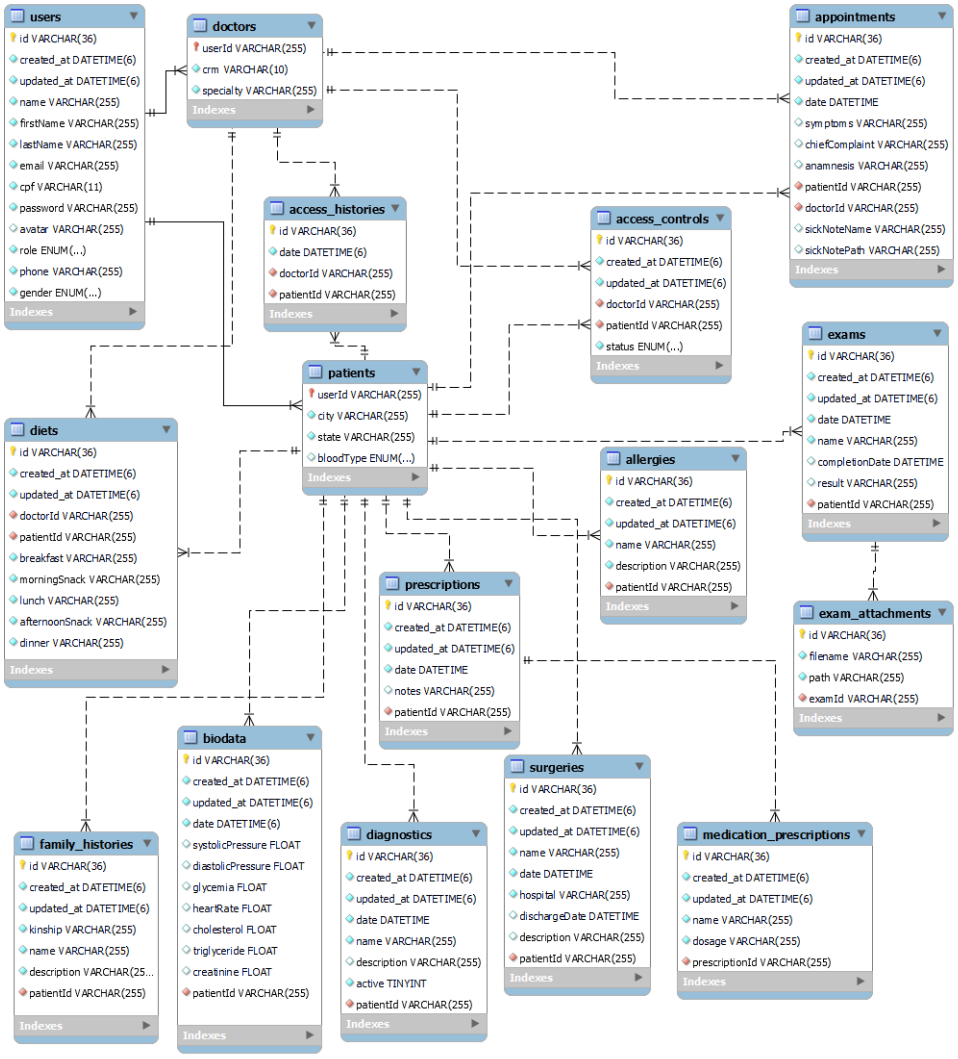
Na Tabela 5, apresenta-se uma descrição da tabela *Users* do modelo físico do banco de dados, os demais serão encontrados no Apêndice D.

Tabela 5 – Tabela Users - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
username	VARCHAR(255)	Nome de usuário para autenticação
password	VARCHAR(255)	Senha criptografada do usuário
userType	ENUM	Tipo de usuário (Médico, Paciente, Administrador)
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Figura 9 – Modelo físico do banco de dados



Fonte: O autor.

3.5 Avaliação

A fase de avaliação visa assegurar a qualidade e eficácia do aplicativo móvel, abrangendo diversos aspectos essenciais, como funcionalidades, usabilidade, integração e componentes individuais. Essa abordagem contribui significativamente para o desenvolvimento bem-sucedido do projeto.

3.5.1 Testes locais

A avaliação das funcionalidades do aplicativo foi realizada por meio de testes específicos, com o objetivo de verificar cada uma das características implementadas. Para isso, foram executados testes manuais pelo desenvolvedor a cada nova funcionalidade ou melhoria introduzida. Adicionalmente, foram criados testes automatizados para garantir que as alterações não impactassem negativamente as funcionalidades previamente existen-

tes. Os detalhes desses testes serão apresentados no Capítulo 4

3.5.2 Testes com usuários

Para avaliar a eficácia, a usabilidade e a satisfação do aplicativo proposto, foi adotado o método Escala de Usabilidade do Sistema (SUS, sigla do termo em inglês *System Usability Scale*) (BROOKE, 1995) O método foi criado por John Brooke em 1986, e pode ser usado para avaliar produtos, serviços, *hardware*, *software*, *websites*, aplicações — e qualquer outro tipo de interface (TEIXEIRA, 2015). A escala consiste em 10 perguntas, que são respondidas por usuários após interagirem com o sistema em questão. Como essas perguntas foram criadas originalmente em inglês, as perguntas utilizadas foram da versão adaptada transculturalmente para o Português do Brasil proposta por Lourenço, Carmona e Lopes (2022). Dessa forma, o questionário apresentado tem a forma apresentada pela Tabela 6, os números representam a pontuação e vão de 1 (discordo totalmente) a 5 (concordo totalmente).

Tabela 6 – Questionário SUS traduzido

Questão	1	2	3	4	5
01 Eu acho que gostaria de usar esse sistema frequentemente.					
02 Eu achei esse sistema desnecessariamente complexo.					
03 Eu achei esse sistema fácil de usar.					
04 Eu achei que precisaria de ajuda de uma pessoa técnica para ser capaz de usar esse sistema.					
05 Eu achei que as várias funções desse sistema foram bem integradas.					
06 Eu acho que o sistema apresenta muita inconsistência.					
07 Eu imagino que a maioria das pessoas pode aprender a usar esse sistema rapidamente.					
08 Eu achei esse sistema muito pesado para usar.					
09 Eu me senti muito seguro usando o sistema.					
10 Eu precisei aprender muitas coisas antes que pudesse utilizar esse sistema.					

Fonte: Adaptado de (LOURENÇO; CARMONA; LOPES, 2022; SAURO; LEWIS, 2016).

A pontuação final do SUS fornece uma medida quantitativa da usabilidade percebida e seu cálculo pode ser representado da seguinte forma:

$$SUS = 2.5 \cdot \sum_{i=1}^{10} \text{Contribuição}(P_i)$$

Onde:

- ***SUS*** é a pontuação final do *System Usability Scale*;
- $\boldsymbol{P_i}$ representa as respostas do usuário a cada uma das 10 perguntas da escala.
- **Contribuição($\boldsymbol{P_i}$)** é uma função que determina como as respostas individuais afetam a pontuação, dependendo se a pergunta é ímpar ou par.

A função de contribuição é definida da seguinte forma:

$$\text{Contribuição}(P_i) = \begin{cases} 5 - P_i, & \text{se } i \text{ é par,} \\ P_i - 1, & \text{se } i \text{ é ímpar.} \end{cases}$$

Para classificar a pontuação do SUS foi utilizada a abordagem proposta por Sauro e Lewis (2016), onde foram utilizados dados de 241 estudos e inquéritos de usabilidade industrial para criar uma escala de classificação curva em que uma pontuação do SUS. A Tabela 7 mostra a escala de classificação curva completa, com o intervalo de pontuações do SUS para cada nota e o intervalo de percentil correspondente.

Tabela 7 – Escala de Classificação Curva para o SUS

Classificação	SUS	Faixa Percentil
A+	84,1 - 100	96 - 100
A	80,8 - 84	90 - 95
A-	78,9 - 80,7	85 - 89
B+	77,2 - 78,8	80 - 84
B	74,1 - 77,1	70 - 79
B-	72,6 - 74	65 - 69
C+	71.1 - 72,5	60 - 64
C	65 - 71	41 - 59
C-	62,7 - 64,9	35 - 40
D	51,7 - 62,6	15 - 34
F	0 - 51,6	0 - 14

Fonte: (SAURO; LEWIS, 2016).

4 Desenvolvimento da aplicação

O desenvolvimento da aplicação seguiu uma abordagem incremental, visando criar um produto funcional e eficiente. Este capítulo detalha as etapas e processos envolvidos no desenvolvimento do aplicativo, desde a definição das tecnologias utilizadas até a implementação das funcionalidades e a criação da interface de usuário. Cada fase do desenvolvimento foi planejada e executada com o objetivo de permitir a entrega de um aplicativo de alta qualidade.

4.1 Metodologia ágil

Durante o desenvolvimento do projeto, foi adotada a metodologia ágil *Kanban* com o uso da ferramenta *Trello* para o gerenciamento das tarefas. O *Kanban* oferece uma abordagem visual e flexível, que se adapta bem a equipes pequenas e projetos individuais, permitindo um controle melhor do progresso e da carga de trabalho (BUENO, 2023).

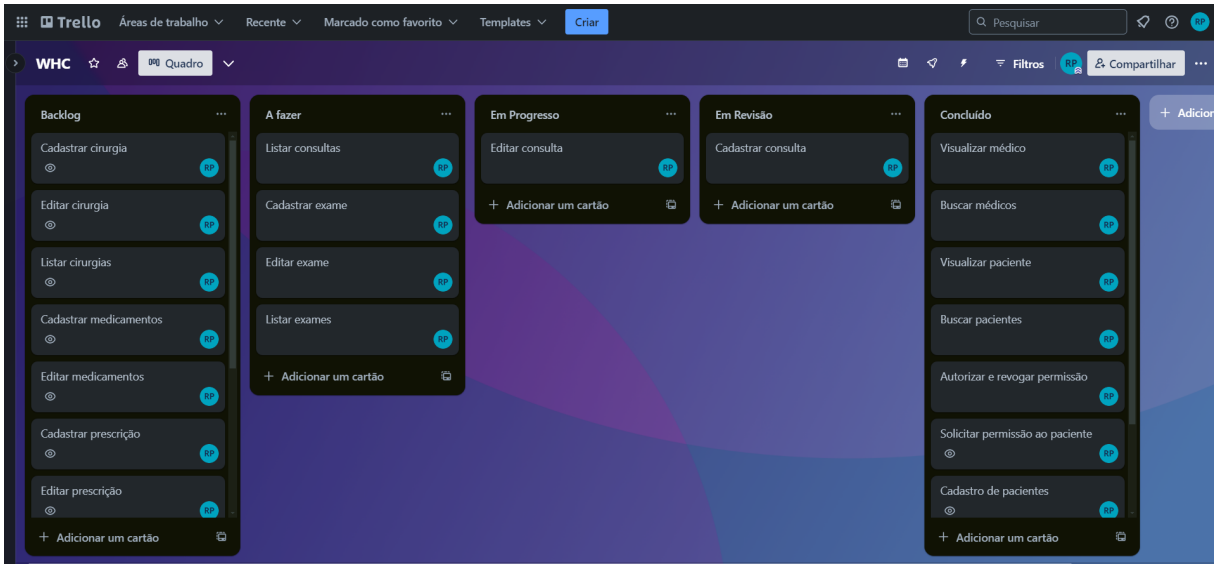
A implementação do *Kanban* no *Trello* começou com a criação de um quadro dedicado ao projeto. Este quadro foi dividido em colunas que representam os diferentes estados das tarefas: *Backlog*, A Fazer, Em Progresso, Em Revisão e Concluído. Cada tarefa ou funcionalidade do projeto foi representada por um cartão dentro dessas colunas, permitindo uma visualização clara do status e das prioridades.

- *Backlog*: Todas as ideias e funcionalidades planejadas foram inicialmente listadas no backlog. Isso garantiu que todas as tarefas fossem documentadas e pudessem ser facilmente acessadas e priorizadas quando necessário.
- A Fazer: As tarefas movidas para esta coluna eram aquelas prontas para serem iniciadas, com prioridades definidas com base nas necessidades do projeto.
- Em Progresso: Esta coluna continha as tarefas atualmente em desenvolvimento. Mover uma tarefa para esta coluna indicava o início do trabalho ativo nela.
- Em Revisão: Após a conclusão do desenvolvimento inicial, as tarefas eram movidas para "Em Revisão", onde passavam por uma análise final, ajustes e testes.
- Concluído: Uma vez aprovadas e testadas, as tarefas eram movidas para a coluna "Concluído", sinalizando sua finalização.

A escolha do *Kanban* para o desenvolvimento deste projeto se mostrou adequada pois a metodologia possibilitou um gerenciamento ágil, visual e adaptável. A Figura 10

mostra o quadro do *Trello* utilizado no desenvolvimento do sistema e a distribuição das tarefas em suas respectivas colunas.

Figura 10 – Quadro Kanban



Fonte: Do autor.

4.2 Desenvolvimento da API

O sistema *web* do WHC foi desenvolvido utilizando *Ruby on Rails*, o que dificultava a integração com um aplicativo móvel. Para contornar esse problema, foi desenvolvida uma API REST, facilitando a comunicação com o aplicativo.

4.2.1 Escolha das tecnologias

A escolha de *Node.js* como tecnologia base para a API foi motivada principalmente pela familiaridade do autor com a plataforma, além de vários benefícios inerentes. *Node.js* é amplamente reconhecido por sua eficiência em lidar com operações de *I/O* não-bloqueantes, tornando-o ideal para aplicações em tempo real e de alta demanda. Além disso, *Node.js* possui uma vasta comunidade e um ecossistema robusto de bibliotecas, o que facilita o desenvolvimento rápido e eficiente.

O *framework Express* foi selecionado devido à sua simplicidade e flexibilidade. *Express* é um *framework* minimalista para *Node.js* que oferece uma estrutura para a construção de APIs RESTful, permitindo um desenvolvimento mais organizado e modular. Sua simplicidade na configuração e uso possibilita uma curva de aprendizado suave e uma implementação eficiente de rotas e *middlewares*.

Para a camada de persistência de dados, foi escolhido o *TypeORM*, um ORM (*Object-Relational Mapper*) que suporta *TypeScript* (KHUDOIBERDIEV, 2024), proporcionando

um desenvolvimento mais seguro e eficiente. O *TypeORM* facilita o mapeamento entre objetos do sistema e tabelas do banco de dados, promovendo uma integração mais fluida e menos propensa a erros.

4.2.2 Arquitetura da API

A arquitetura da API segue os princípios da *Clean Architecture*, proposta por Robert C. Martin (*Uncle Bob*). A Clean Architecture é projetada para criar sistemas modulares, escaláveis e de fácil manutenção, separando as responsabilidades e promovendo a independência de frameworks e tecnologias específicas (MARTIN, 2017). A seguir, são descritas as principais camadas do projeto:

4.2.2.1 Principais camadas

A API é organizada em diversos diretórios, cada um com uma função específica dentro do sistema:

- ***Repositories*:** Esta camada é responsável pela interação com o banco de dados. Os repositórios contêm a lógica para acessar, armazenar e gerenciar dados, utilizando o *TypeORM* para facilitar o mapeamento entre objetos do sistema e tabelas do banco de dados. Essa separação isola o acesso ao banco de dados das demais camadas da aplicação, promovendo uma melhor organização e manutenção do código.
- ***Entities*:** As entidades representam as tabelas do banco de dados dentro do código da aplicação. Elas são mapeadas diretamente para as tabelas do banco de dados e são gerenciadas pelo *TypeORM*. As entidades definem a estrutura dos dados e as relações entre diferentes tabelas, facilitando a manipulação e a validação dos dados.
- ***Controllers*:** Os controladores gerenciam as requisições HTTP recebidas e retornam as respostas apropriadas. Eles são responsáveis por coordenar as interações entre as diferentes partes do sistema, chamando os serviços necessários para processar as requisições. Os controladores não devem conter lógica de negócio complexa, mantendo essa responsabilidade na camada de serviços.
- ***Services*:** Esta camada contém a lógica de negócio da aplicação. Os serviços processam os dados e realizam operações específicas de negócio, sendo chamados pelos controladores. A separação da lógica de negócio em serviços promove a reutilização de código e facilita a implementação de testes unitários.
- ***Routes*:** As rotas definem os *endpoints* da API e mapeiam as requisições para os controladores apropriados. Elas simplificam a configuração das rotas e ajudam a manter o código organizado, permitindo uma fácil adição e modificação de *endpoints* conforme necessário.

- ***Middlewares*:** Os *middlewares* são funções intermediárias que processam as requisições antes de chegarem aos controladores ou as respostas antes de serem enviadas ao cliente. Eles são utilizados para tarefas como autenticação, autorização, manipulação de erros, entre outras, proporcionando uma maior modularidade e reutilização de código.

4.2.3 Autenticação e Segurança

A autenticação e segurança desempenham um papel essencial no desenvolvimento da API, assegurando que somente usuários autorizados possam acessar e manipular os recursos do sistema de maneira segura.

A API utiliza *tokens* JWT (*JSON Web Tokens*), uma solução robusta e amplamente adotada para autenticação em aplicações Web e definido pela definido pela RFC7519 (JONES; BRADLEY; SAKIMURA, 2015). Durante o cadastro de usuários, o algoritmo BCrypt (RÊGO, 2020) é aplicado para gerar um hash seguro da senha do usuário, garantindo armazenamento seguro no banco de dados. No momento do *login*, o *hash* da senha fornecida é comparado com o *hash* armazenado para validar as credenciais. Uma vez autenticado com sucesso, um *token* JWT é gerado e assinado utilizando uma chave secreta, proporcionando um meio seguro para autenticação subsequente.

Além da autenticação, a API implementa middlewares dedicados para fortalecer a segurança das requisições posteriores. Esses *middlewares* são responsáveis por verificar a presença e validade dos *tokens* JWT em cada requisição, assegurando que apenas usuários autenticados e autorizados possam acessar os recursos da API. Eles também controlam as permissões de acesso, garantindo que cada usuário possa realizar apenas as operações permitidas conforme sua função e necessidade dentro do sistema.

Para gerenciar e controlar as permissões de acesso de forma eficiente, a API adota um sistema baseado em funções (*role-based system*)(Wikipedia, 2024). Esse sistema atribui funções específicas, como "Médico"e "Paciente", a cada usuário com base em suas responsabilidades e privilégios dentro da aplicação. Isso permite uma gestão granular de acesso, onde cada role define o escopo de ações que um usuário pode realizar.

4.3 Desenvolvimento do aplicativo

O processo de desenvolvimento do aplicativo iniciou-se com a prototipagem das telas, que possibilitou a criação de interfaces detalhadas e interativas. A partir dos protótipos, foi realizada a implementação das funcionalidades no ambiente de desenvolvimento do *React Native*, utilizando *Expo* para simplificar a configuração e o gerenciamento do projeto.

A implementação das funcionalidades seguiu uma abordagem incremental e iterativa,

permitindo ajustes e melhorias contínuas.

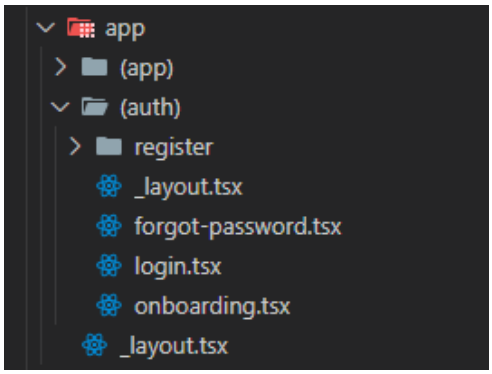
A arquitetura do aplicativo segue o padrão de componentes reutilizáveis e gerenciadores de estado. Os componentes são organizados de forma modular, facilitando a manutenção e a evolução do código. Para o gerenciamento de estado, foi utilizado *Zustand*, uma biblioteca que oferece uma API simples para a gestão de estados globais (MARQUES JANNIK WEMPE, 2024).

No que diz respeito ao design e estilização das interfaces, foi adotada uma abordagem moderna com o uso do *NativeWind*(MADUABUCHI, 2024), uma adaptação do *Tailwind CSS*(Tailwind Labs, 2024) para *React Native*. Essa escolha permitiu a aplicação de estilos de forma rápida e eficiente, mantendo a consistência visual e facilitando a customização de componentes. O *NativeWind* oferece uma sintaxe intuitiva para a definição de estilos, além de ser compatível com a estrutura modular do projeto, contribuindo para a produtividade e qualidade do desenvolvimento.

4.3.1 Navegação

A navegação do aplicativo é estruturada utilizando o *Expo Router*, um gerenciador de navegação baseado em arquivos, onde cada arquivo adicionado ao projeto se transforma automaticamente em uma rota. A Figura 11 ilustra parte da navegação baseada em arquivos implementada no aplicativo.

Figura 11 – Navegação baseada em arquivos



Fonte: Do autor.

Arquivos nomeados como *_layout* definem uma estrutura base para encapsular um conjunto de componentes. Diretórios entre parênteses são chamados de grupos e não são considerados na URL, mas servem para organizar as seções do aplicativo. Por exemplo, o diretório *(app)* contém o aplicativo principal, enquanto o diretório *(auth)* agrupa as páginas de apresentação, cadastro, login e recuperação de senha.

Em contraste, diretórios que não estão entre parênteses são considerados na URL. Por exemplo, o diretório *register* resultará na URL */register*. Da mesma forma, um arquivo

individual se torna uma rota de navegação; por exemplo, o arquivo *login* resulta na rota */login.*

4.3.2 Prototipagem

A fase inicial do desenvolvimento consistiu na prototipagem das telas utilizando o *Figma*. A prototipagem é crucial para definir previamente as interfaces do sistema, validar a experiência de usabilidade do usuário e testar critérios como facilidade de uso e intuitividade. Esse processo permite identificar e eliminar possíveis obstáculos, excessos, redundâncias e erros, garantindo que a ferramenta seja intuitiva e fluida (BORDIGNON, 2023).

Foi decidido manter a paleta de cores e os logotipos do sistema web existente, adaptando os fluxos de navegação para otimizar o uso em dispositivos móveis.

4.3.3 Fluxo padrão de CRUDs no aplicativo

No aplicativo, a interface de gerenciamento de dados médicos segue um fluxo padronizado para a maioria das funcionalidades, como cirurgias, exames, prescrições, diagnósticos, alergias, consultas e outros registros médicos. A listagem de itens é apresentada em formato de *cards*, onde cada *card* exibe informações essenciais sobre o registro. Para facilitar a busca, a interface também inclui um campo de pesquisa, caso aplicável, permitindo que o usuário localize registros específicos de maneira rápida e eficiente.

Ao selecionar um item da listagem, os usuários têm acesso aos detalhes completos do registro. A visualização detalhada permite que médicos e pacientes consultem informações adicionais, como descrições e outros dados relevantes. No entanto, as permissões variam entre médicos e pacientes: enquanto os médicos podem editar ou excluir os registros, os pacientes têm acesso apenas à visualização de suas próprias informações, sem a capacidade de realizar alterações.

Para funcionalidades de criação, as telas de listagem dos médicos incluem um botão de "Adicionar", que redireciona para uma página específica de cadastro. Essa abordagem garante que os médicos possam facilmente inserir novos registros e manter o histórico médico atualizado.

Todos esses detalhes mencionados poderão ser visualizados posteriormente na descrição de cada fluxo, onde as particularidades de cada funcionalidade serão destacadas.

4.3.4 Principais telas desenvolvidas

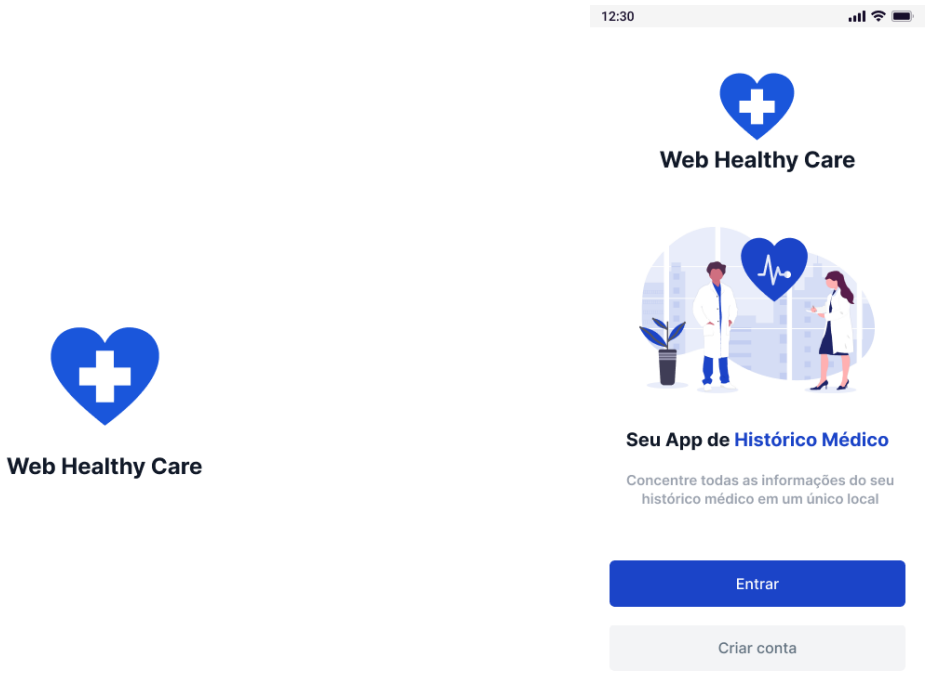
Nesta seção, são apresentadas as principais telas desenvolvidas para o aplicativo, cada uma desempenhando um papel na experiência do usuário e na funcionalidade do

sistema. As telas foram projetadas para serem intuitivas e acessíveis, facilitando o uso por médicos e pacientes. A seguir, são descritas as telas iniciais do aplicativo, seguidas pelas interfaces específicas para cada fluxo de trabalho essencial, como gerenciamento de consultas, exames, alergias, diagnósticos, cirurgias, prescrições, histórico familiar, dietas, biodados e permissões de acesso.

4.3.4.1 *Splash* e *Onboarding*

O fluxo do aplicativo se inicia na *splash screen*, a tela inicial exibida durante o carregamento do aplicativo. Esta tela apresenta apenas o logotipo do sistema sobre um fundo branco. Um usuário não autenticado é direcionado para uma tela de *onboarding*, projetada para apresentar o aplicativo. Esta tela contém dois botões principais: um para redirecionar à tela de *login* e outro para iniciar o fluxo de cadastro. Adicionalmente, a tela inclui elementos visuais e textuais que têm como objetivo dar as boas-vindas aos novos usuários, proporcionando uma introdução amigável e informativa ao aplicativo. A Figura 12 mostra as telas de *splash* e *onboarding*.

Figura 12 – Splash screen e Onboarding



Fonte: Do autor.

4.3.4.2 Login e cadastro

A tela de *login* foi projetada para oferecer uma experiência de autenticação simples e segura, contendo campos para o endereço de e-mail e senha. O campo de senha possui um botão adicional que permite ao usuário esconder ou mostrar os caracteres digitados, proporcionando maior flexibilidade e segurança durante a inserção da senha. Quando o usuário insere suas credenciais e clica no botão "Entrar", o sistema verifica a validade das informações fornecidas. Se as credenciais forem válidas, o usuário será redirecionado para a área apropriada, seja como médico ou paciente, conforme seu perfil no sistema. A Figura 13 apresenta o protótipo da tela de *login* desenvolvida.

Figura 13 – Login

Fonte: Do autor.

O fluxo de cadastro foi redesenhado para facilitar o uso em dispositivos móveis, garantindo uma experiência intuitiva e eficiente. O processo de cadastro é dividido em quatro etapas, cada uma exibida em uma tela distinta.

1. **Dados Básicos do Perfil:** Na primeira etapa, o usuário insere informações básicas do perfil, que são comuns tanto para médicos quanto para pacientes.

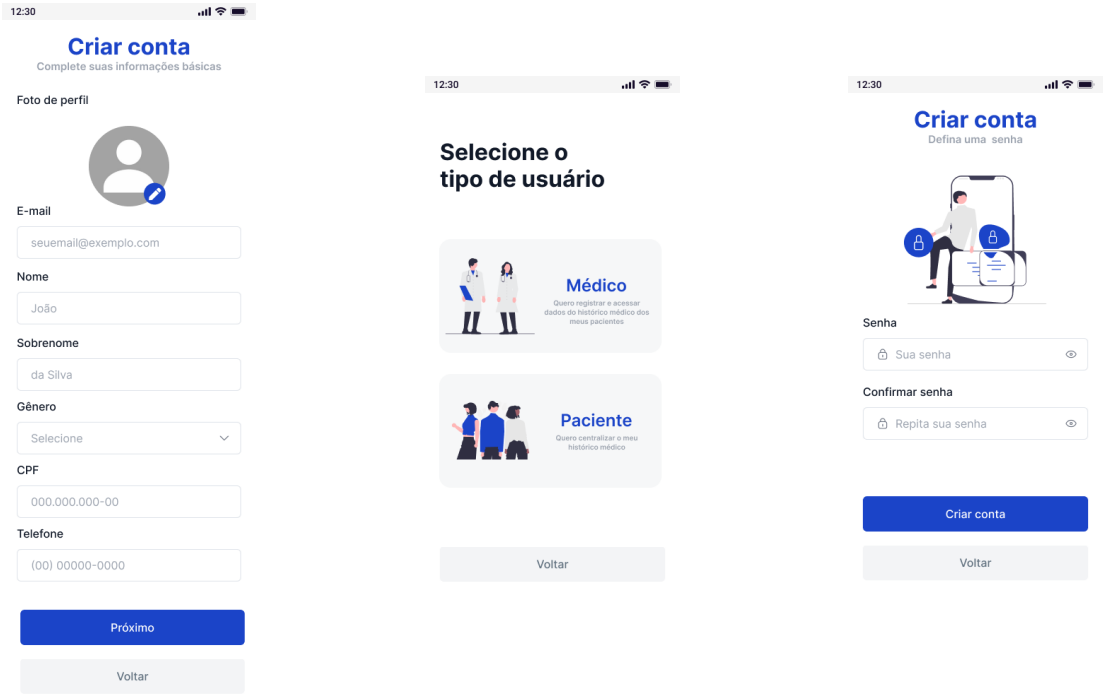
2. **Seleção do Tipo de Usuário:** Na segunda etapa, o usuário escolhe o tipo de usuário que será (médico ou paciente) e é redirecionado para a próxima etapa com base na seleção.

3. **Formulário Específico ao Tipo de Usuário:** Na terceira etapa, o usuário preenche um formulário específico ao tipo de usuário escolhido. Médicos devem informar seu CRM e especialidade, enquanto pacientes devem fornecer informações como tipo sanguíneo e endereço.

4. **Criação de Senha e Confirmação:** Na quarta e última etapa, o usuário cria uma senha de acesso e conclui o cadastro clicando no botão "Criar Conta". Se todos os dados forem validados, o usuário é redirecionado para a tela de login.

A Figura 14 mostra as páginas do cadastro que são comuns aos dois tipos de usuário e correspondem às etapas 1, 2 e 4 do cadastro. As Figuras 15 e 16 exibem a etapa 3 do cadastro para médicos e pacientes, respectivamente.

Figura 14 – Páginas de cadastro comuns



Fonte: Do autor.

Figura 15 – Cadastro para Médicos

Figura 16 – Cadastro para Pacientes

Fonte: Do autor.

4.3.4.3 Telas iniciais dos usuários

Após efetuar login, a primeira tela que os usuários visualizam é a tela *home*. Esta tela apresenta o layout base do aplicativo, que é constituído por um menu inferior com as seguintes opções: *Home* (fluxo principal do aplicativo), Permissões (gerenciador de permissões de acesso) e Perfil (configurações de perfil).

A tela inicial do médico dá as boas-vindas ao usuário e oferece um campo de busca que redireciona para a pesquisa de pacientes. A Figura 17 ilustra a página inicial do médico.

A tela inicial do paciente dá as boas-vindas ao usuário, oferece um campo de busca que redireciona para a pesquisa de médicos e apresenta botões para Consultas, Exames, Alergias, Diagnósticos, Cirurgias, Prescrições, Histórico familiar, Dietas e Biodados. Esses botões redirecionam o paciente para visualizar suas próprias informações de maneira organizada e acessível. A Figura 18 ilustra a página inicial do paciente.

Figura 17 – Home do médico

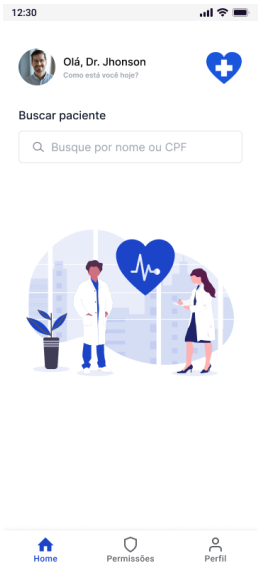
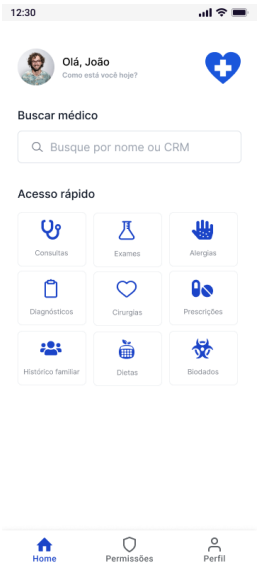


Figura 18 – Home do paciente

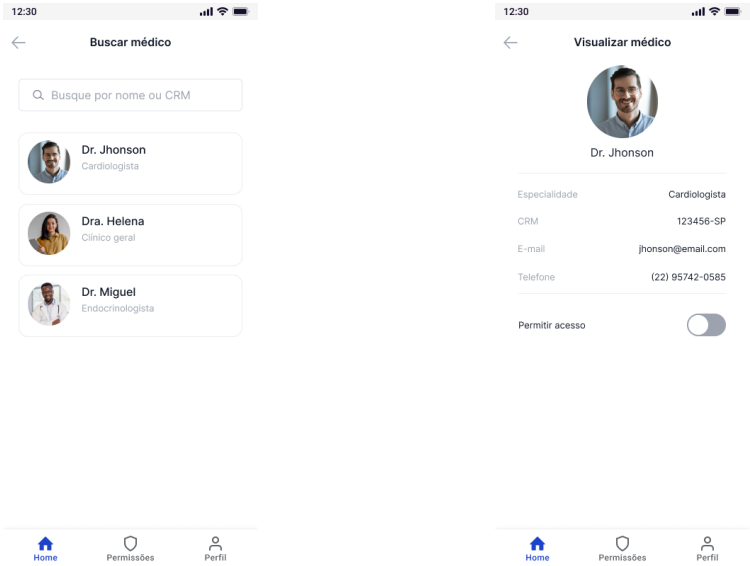


Fonte: Do autor.

4.3.4.4 Busca de médicos

A página de busca de médicos é uma funcionalidade da área do paciente. Nesta página, o paciente pode pesquisar pelo nome ou CRM do médico. Uma lista de médicos correspondentes aparecerá, permitindo que o paciente escolha um médico da lista. Ao selecionar um médico, o paciente pode visualizar informações detalhadas sobre ele e decidir se permite ou não o acesso do médico aos seus dados, utilizando um *switch*. A Figura 19 ilustra as telas da busca de médicos.

Figura 19 – Buscar médicos



Fonte: Do autor.

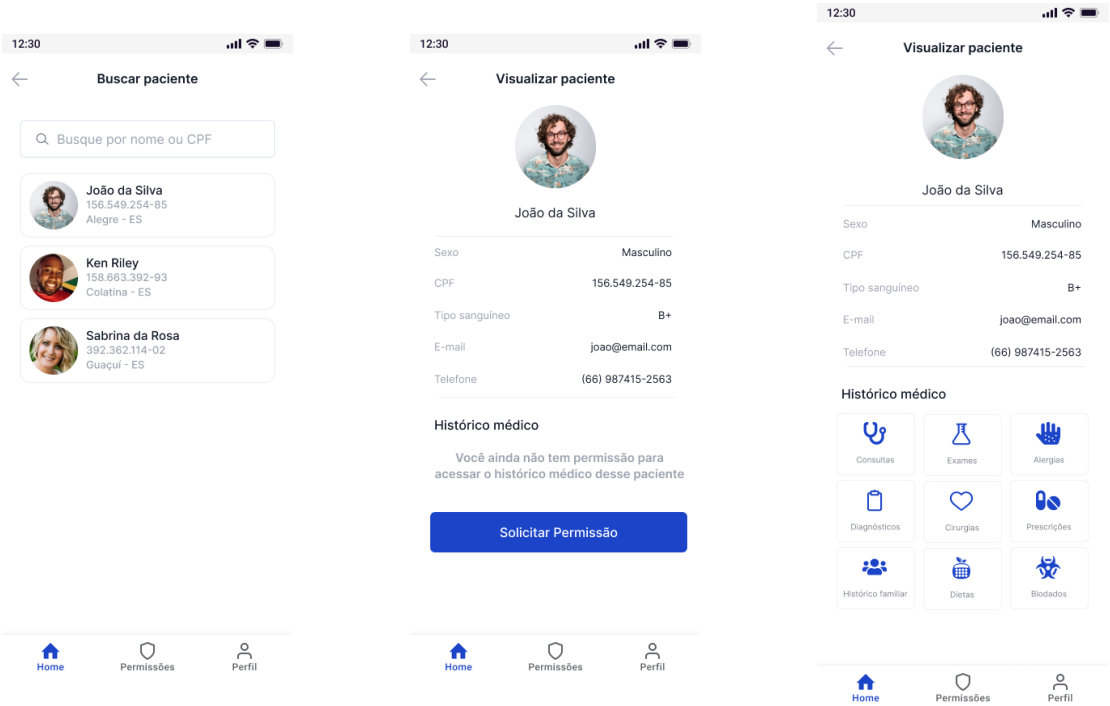
4.3.4.5 Busca de pacientes e solicitação de acesso

A página de busca de pacientes é uma funcionalidade exclusiva da área do médico. Nesta página, o médico pode pesquisar pelo nome ou CPF do paciente. Uma lista de pacientes correspondentes será exibida, permitindo que o médico selecione um paciente da lista e visualize suas informações básicas.

No entanto, em conformidade com a Lei Geral de Proteção de Dados Pessoais (LGPD)(BRASIL, 2018), o médico só terá acesso aos dados detalhados do paciente se houver autorização prévia. Caso o médico não possua essa permissão, será exibida a opção para solicitar o acesso. Somente após o consentimento explícito do paciente, o médico poderá visualizar as informações completas.

Se a permissão já tiver sido concedida, o médico verá os botões Consultas, Exames, Alergias, Diagnósticos, Cirurgias, Prescrições, Histórico Familiar, Dietas e Biodados, que o direcionam para os registros correspondentes. Esses botões permitem ao médico visualizar, editar ou adicionar informações ao histórico do paciente. A Figura 20 ilustra as telas da busca de pacientes.

Figura 20 – Buscar pacientes



Fonte: Do autor.

4.3.4.6 Consultas

As consultas representam os atendimentos realizados por médicos ao paciente. A listagem de consultas exibe informações essenciais, como a data do atendimento, os sintomas relatados e o nome do médico responsável, conforme ilustrado na Figura 21. O

processo de cadastro de consultas envolve a inserção de dados como a data da consulta, sintomas, queixa principal, além de um campo de texto onde o médico pode registrar informações detalhadas sobre a anamnese. Também há um seletor de arquivo que permite anexar o atestados médico, esse seletor também permite digitalizar documentos diretamente pela câmera do dispositivo. A interface de cadastro de consultas é apresentada na Figura 22. A Figura 23 ilustra a visualização e edição de uma consulta pelo médico, enquanto a Figura 24 demonstra a visualização da consulta sob a perspectiva do paciente.

Figura 21 – Listagem de consultas

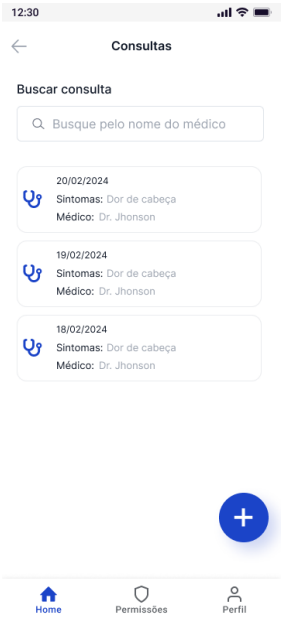
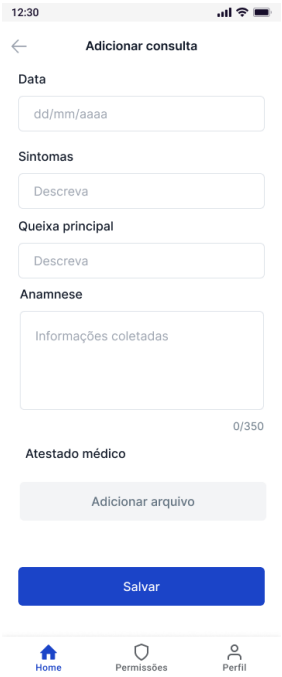


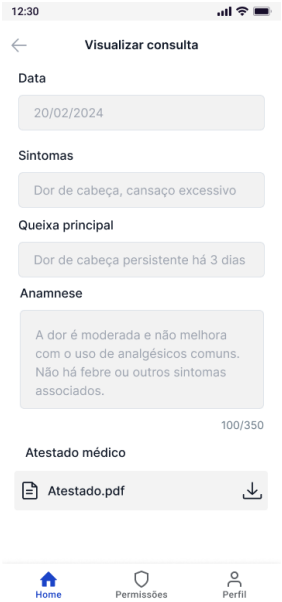
Figura 22 – Cadastro de consultas



Fonte: Do autor.

Figura 23 – Visualizar consultas - Médico

Figura 24 – Visualizar diagnóstico - Paciente



Fonte: Do autor.

4.3.4.7 Exames

Na listagem de exames, cada item apresenta o nome do exame e a data da solicitação, oferecendo uma visão geral clara e organizada dos exames realizados. A busca por exame pode ser feita pelo nome, facilitando a localização de registros específicos em meio a uma grande quantidade de dados, conforme ilustrado na Figura 25.

Um exame é composto por diversos campos essenciais para o acompanhamento do paciente. Inicialmente, ao solicitar um exame, são registrados a data da solicitação e o nome do exame. Após a realização do procedimento, são adicionadas a data da realização, os resultados, que são inseridos em um campo de texto.

A interface de cadastro e edição de exames, exibida na Figura 26, permite que o médico adicione arquivos em anexo, como laudos e imagens dos exames. Além disso, o sistema oferece a funcionalidade de digitalização de documentos, possibilitando que o médico utilize a câmera do dispositivo para capturar e anexar documentos diretamente, tornando o processo mais ágil e prático.

Essas funcionalidades proporcionam um fluxo de trabalho eficiente tanto para médicos quanto para pacientes, garantindo que todas as informações sobre os exames estejam disponíveis de forma clara e detalhada, facilitando o acompanhamento e a gestão do histórico médico.

Figura 25 – Listagem de exames

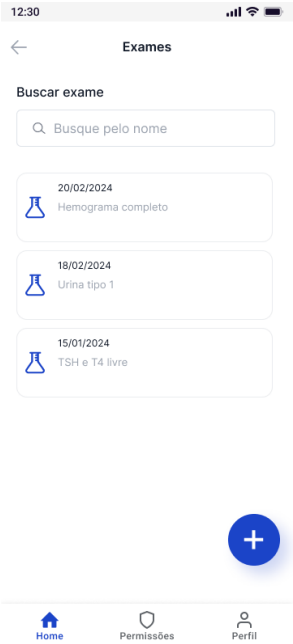
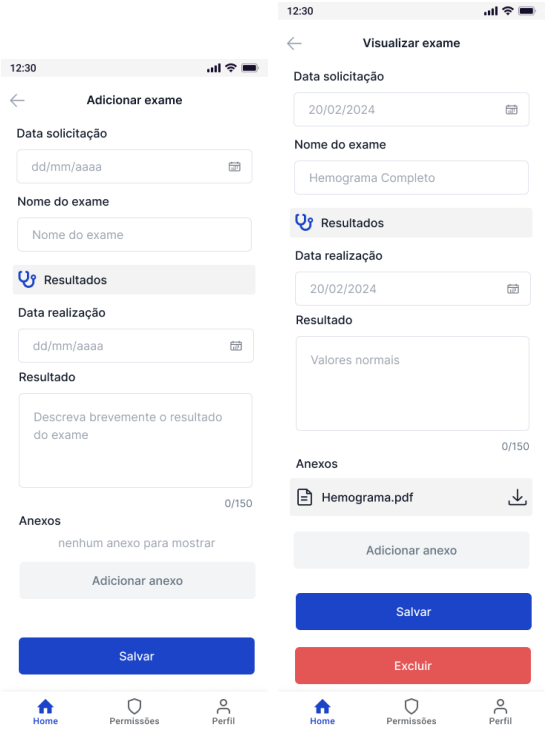


Figura 26 – Cadastro e edição de exames



Fonte: Do autor.

4.3.4.8 Alergias

A listagem de alergias exibe tanto o nome quanto uma parte da descrição de cada alergia. A Figura 27 ilustra essa listagem. A Figura 28 apresenta a interface de criação de uma nova alergia, destinada aos médicos. Os campos necessários para o cadastro de uma alergia são o nome e a descrição da alergia. A visualização de uma alergia permite que os usuários acessem informações detalhadas sobre uma alergia específica, como o nome e a descrição completa. A Figura 29 ilustra a visualização e edição de uma alergia pelo médico, enquanto a Figura 30 mostra a visualização da alergia sob a perspectiva do paciente.

Figura 27 – Listagem de alergias

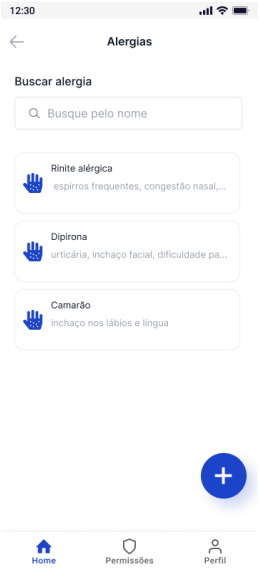


Figura 28 – Cadastro de alergias

Fonte: Do autor.

Figura 29 – Visualizar alergia - Médico

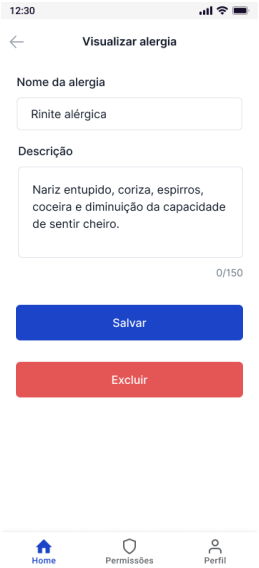
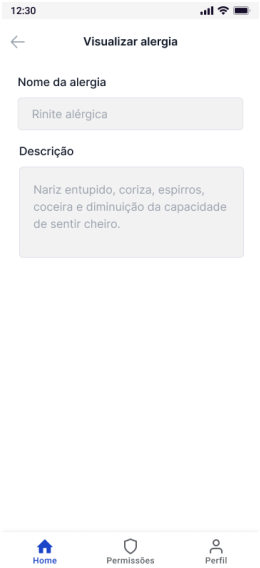


Figura 30 – Visualizar alergia - Paciente



Fonte: Do autor.

4.3.4.9 Diagnósticos

Os diagnósticos correspondem a uma doença ou condição médica identificada no paciente. A listagem de diagnósticos apresenta, para cada item, o nome da enfermidade, a data de registro e a situação (ativa ou inativa), como ilustrado na Figura 31. A interface de cadastro de novos diagnósticos, disponível para os médicos, é exibida na Figura 32. Ao acessar a visualização detalhada de um diagnóstico, além dos campos de data, nome da enfermidade e situação, há um campo adicional destinado ao registro de outras informações relevantes sobre o diagnóstico. Esse campo é especialmente útil para anotações

complementares que o médico julgue necessárias. O componente de situação é implementado como um *switch*, proporcionando uma interface intuitiva para os médicos e garantindo clareza na leitura por parte dos usuários. A Figura 33 ilustra a visualização e edição de um diagnóstico pelo médico, enquanto a Figura 34 mostra a visualização do diagnóstico sob a perspectiva do paciente.

Figura 31 – Listagem de diagnósticos

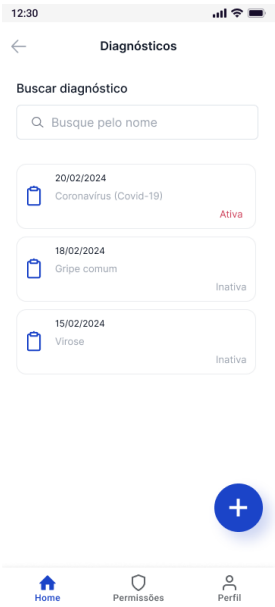


Figura 32 – Cadastro de diagnósticos

Fonte: Do autor.

Figura 33 – Visualizar diagnóstico - Médico

Figura 34 – Visualizar diagnóstico - Paciente

Fonte: Do autor.

4.3.4.10 Cirurgias

A listagem de cirurgias apresenta informações essenciais, como a data do procedimento, o nome da cirurgia e o hospital onde foi realizada, conforme ilustrado na Figura 35. A Figura 36 exibe a interface de criação de uma nova cirurgia, disponível para os médicos. Ao acessar a visualização detalhada de uma cirurgia, são apresentados dados complementares, incluindo a data de alta e uma descrição completa do procedimento, oferecendo uma visão abrangente do histórico cirúrgico do paciente. Cabe destacar que tanto a data de alta quanto a descrição são campos opcionais, que podem ser adicionados ou atualizados posteriormente pelo médico, permitindo um registro mais detalhado e personalizado do tratamento. A Figura 37 ilustra a visualização e edição de uma cirurgia pelo médico, enquanto a Figura 38 demonstra a visualização da cirurgia sob a perspectiva do paciente.

Figura 35 – Listagem de cirurgias

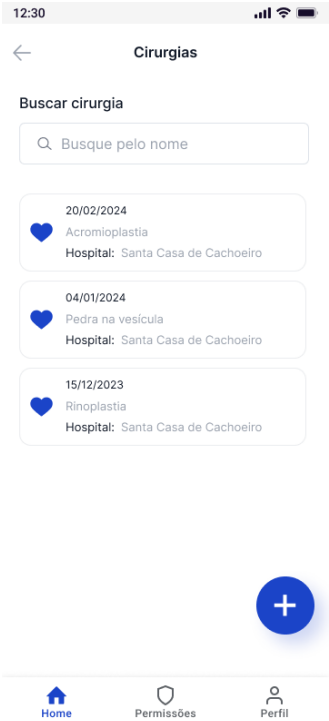


Figura 36 – Cadastro de cirurgias

Fonte: Do autor.

Figura 37 – Visualizar cirurgia - Médico

Figura 38 – Visualizar cirurgia - Paciente

Fonte: Do autor.

4.3.4.11 Prescrições

As prescrições são conjuntos de medicamentos que os médicos prescrevem aos pacientes, contendo informações essenciais para o tratamento. Cada prescrição inclui a data de emissão, uma lista de medicamentos e um campo de observações, onde o médico pode adicionar orientações ou informações complementares importantes. A Figura 39 ilustra a listagem de prescrições, que exibe a data da prescrição e os nomes dos medicamentos, permitindo a busca por nome para facilitar o acesso rápido às informações necessárias.

O processo de cadastro de prescrições se destaca por uma funcionalidade específica: um botão que permite adicionar medicamentos, direcionando o médico a uma tela dedicada onde ele pode preencher o nome e a posologia do medicamento a ser incluído na prescrição. A Figura 40 mostra essa interface de cadastro, evidenciando como o médico pode adicionar novos medicamentos à prescrição. Após a inclusão de um medicamento, o médico tem a opção de alterá-lo ou excluí-lo, como ilustrado na Figura 41, que apresenta a interface de edição disponível apenas para o médico.

Por outro lado, a visualização da prescrição pelo paciente é limitada, permitindo apenas que ele consulte as informações prescritas sem possibilidade de edição. A Figura 42 demonstra essa visualização sob a perspectiva do paciente, garantindo que ele tenha acesso às informações necessárias para seguir o tratamento prescrito.

Figura 39 – Listagem de prescrições

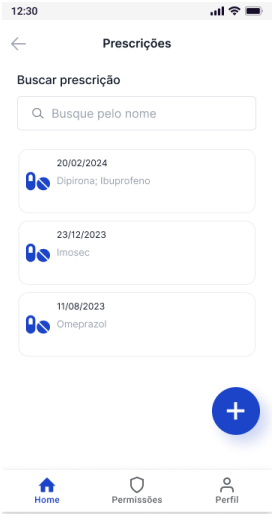
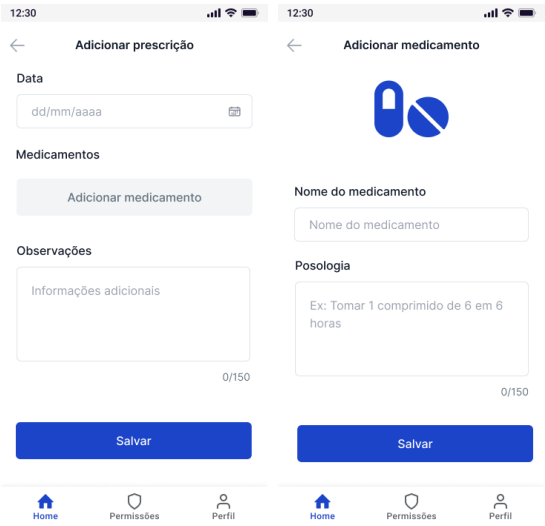


Figura 40 – Cadastro de prescrições



Fonte: Do autor.

Figura 41 – Editar prescrição - Médico

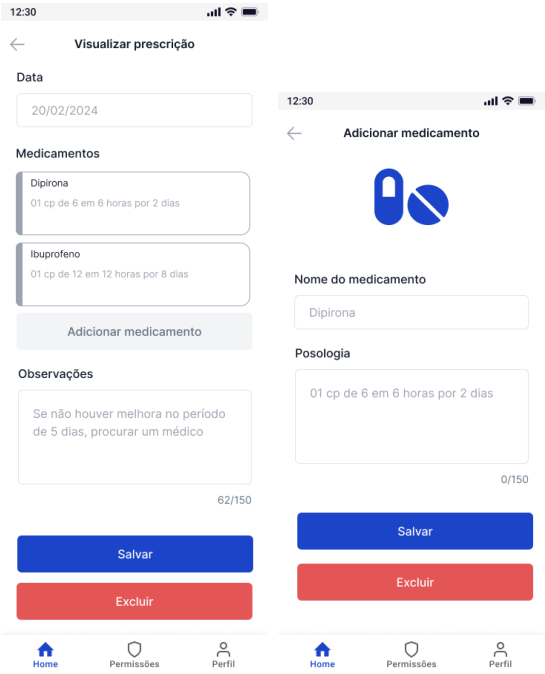
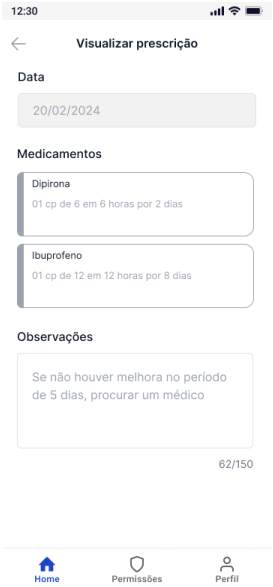


Figura 42 – Visualizar prescrição - Paciente



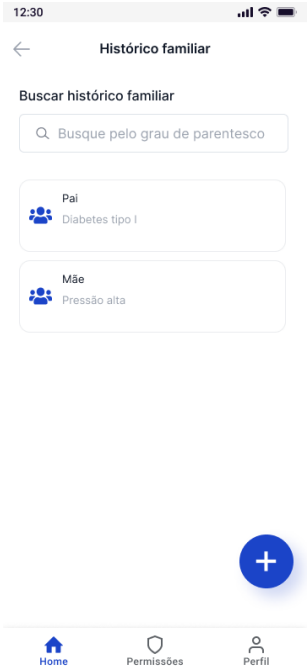
Fonte: Do autor.

4.3.4.12 Histórico familiar

A funcionalidade de listagem do histórico familiar permite que os usuários visualizem registros resumidos, apresentando o grau de parentesco e um trecho da descrição. Esta interface é exemplificada na Figura 43. Além disso, é possível realizar buscas filtrando os registros pelo grau de parentesco, facilitando a navegação.

Os campos necessários para a inserção de um novo registro de histórico familiar incluem: grau de parentesco, nome do familiar e uma descrição que destaque os pontos relevantes para a saúde do paciente. A Figura 44 ilustra como o médico pode visualizar e editar esses registros, enquanto a Figura 45 mostra a interface de visualização sob a perspectiva do paciente.

Figura 43 – Listagem de histórico familiar



Fonte: Do autor.

Figura 44 – Visualizar histórico - Médico

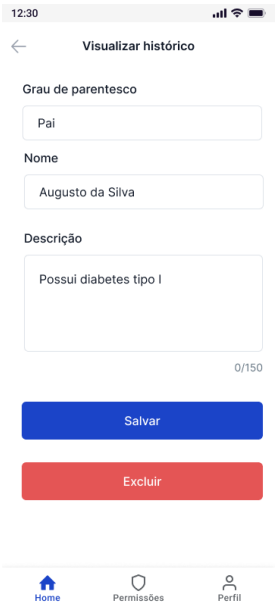
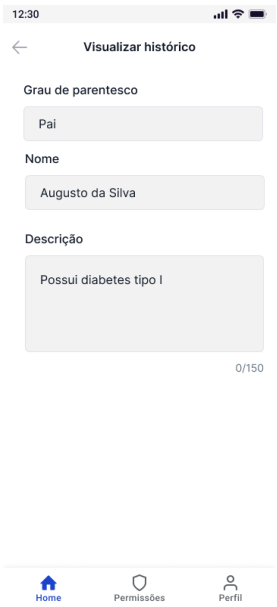


Figura 45 – Visualizar histórico - Paciente



Fonte: Do autor.

4.3.4.13 Dietas

A listagem de dietas exibe informações como a data de criação e o nome do médico responsável, organizadas em *cards* conforme ilustrado na Figura 46. Cada dieta é composta por campos de texto para detalhar as refeições do dia, incluindo café da manhã, lanche da manhã, almoço, lanche da tarde e jantar. A Figura 47 demonstra como o médico pode visualizar e editar os registros das dietas, enquanto a Figura 48 apresenta a interface de visualização sob a perspectiva do paciente.

Figura 46 – Listagem de dietas.

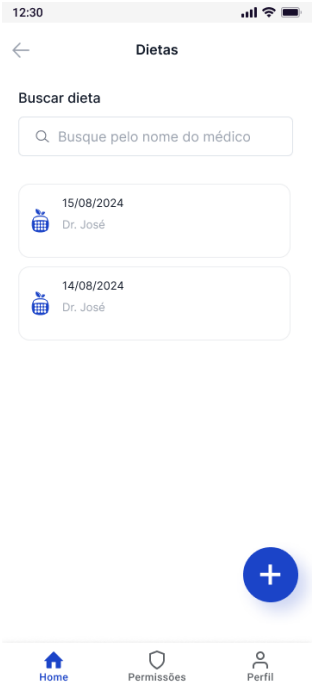


Figura 47 – Visualizar dieta - Médico.

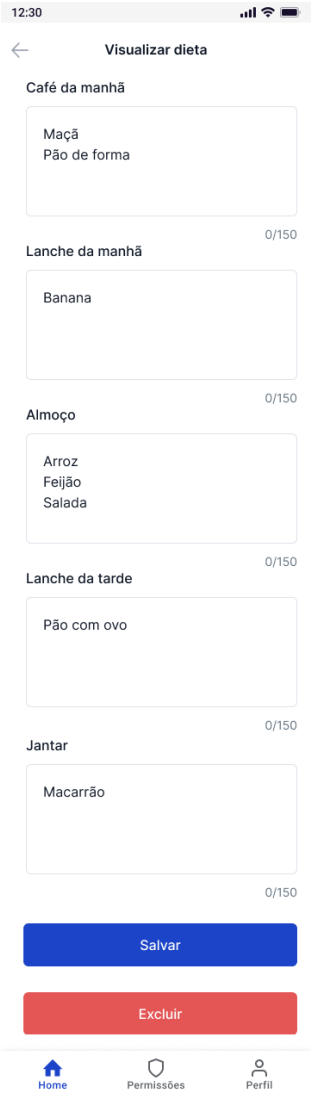
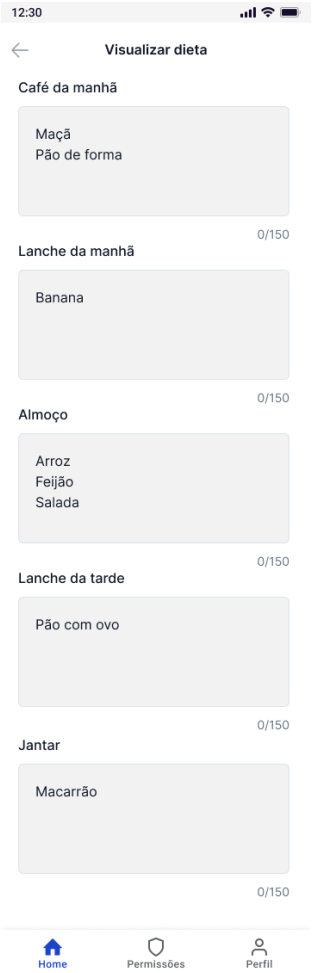


Figura 48 – Visualizar dieta - Paciente.



Fonte: Do autor.

4.3.4.14 Biodados

O cadastro de biodados permite ao médico registrar valores e taxas relevantes do paciente, contribuindo para o histórico médico. Os tipos de dados que podem ser registrados incluem pressão arterial, glicemia, frequência cardíaca, colesterol, triglicerídeos

e creatina. A listagem dos biodados apresenta a data dos registros mais recentes, oferecendo acesso fácil tanto para médicos quanto para pacientes.

A Figura 49 ilustra a tela de listagem dos biodados. A Figura 50 mostra a interface para edição e visualização dos biodados pelo médico, enquanto a Figura 51 exibe a visualização dos biodados sob a perspectiva do paciente.

Figura 49 – Listagem de biodados.

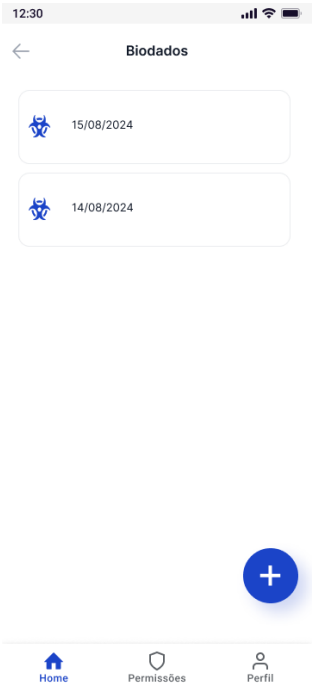
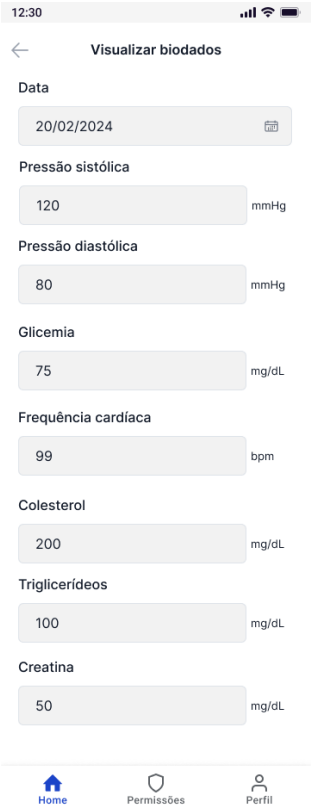


Figura 50 – Visualizar biodados - Médico.

Figura 51 – Visualizar biodados - Paciente.



Fonte: Do autor.

4.3.4.15 Fluxo de Permissões para Pacientes

Ao clicar no menu "Permissões", o paciente é redirecionado para a interface de gerenciamento de permissões de acesso aos seus dados, conforme ilustrado na Figura 52. Nesta tela, as solicitações de acesso pendentes são listadas no topo, permitindo que o paciente analise e tome uma decisão. O paciente pode aprovar ou recusar o pedido de acesso de um médico específico. Uma vez aprovada a solicitação, o médico é movido para a lista de solicitações aprovadas, que aparece logo abaixo.

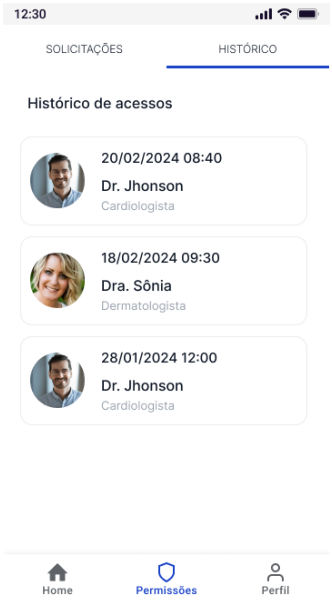
A permissão concedida ao médico é válida até que o paciente a revogue. Para isso, o paciente pode visualizar os detalhes do médico na lista de permissões aprovadas e, em

seguida, alternar o toggle "Permitir acesso"para desativar a permissão, conforme ilustrado na Figura 19. Esta interface oferece uma maneira clara e intuitiva para o paciente gerenciar quem tem acesso aos seus dados médicos.

Além disso, há uma aba "Histórico"na mesma tela de permissões, onde o paciente pode ver uma lista detalhada de médicos que acessaram seus dados recentemente, juntamente com as respectivas datas de acesso. Essa funcionalidade garante maior transparência e controle sobre o uso das informações pessoais, conforme mostrado anteriormente na Figura 53.

Figura 52 – Permissões - Paciente

Figura 53 – Histórico de Acessos



Fonte: Do autor.

4.3.4.16 Fluxo de Permissões para Médicos

Para os médicos, o menu "Permissões"também está disponível, mas com um foco diferente. Ao acessar este menu, o médico é direcionado para uma tela que exibe as solicitações de acesso pendentes, permitindo que ele saiba quais pacientes ainda precisam aprovar seu pedido de acesso, como mostrado na Figura 54. Além disso, é exibida uma lista de pacientes aos quais o médico já tem acesso, possibilitando um gerenciamento eficiente das informações de saúde dos pacientes sob sua responsabilidade.

Figura 54 – Permissões - Médico

Fonte: Do autor.

Essa interface garante que os médicos possam monitorar e gerenciar de forma simples suas permissões de acesso, enquanto os pacientes têm total controle sobre quem pode visualizar seus dados médicos.

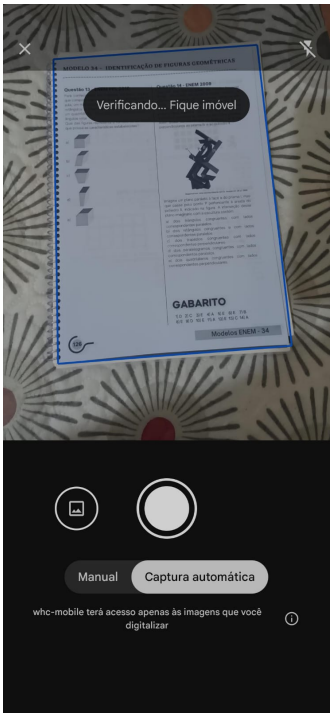
O fluxo de gerenciamento de permissões proporciona uma experiência segura e transparente tanto para pacientes quanto para médicos, garantindo que o acesso às informações sensíveis seja controlado de maneira intuitiva.

4.3.4.17 Anexar arquivos

No gerenciamento de consultas e exames, o sistema permite que o médico anexe arquivos relevantes. Ao selecionar a opção de anexar, como ilustrado na figura 26, o usuário será redirecionado para a tela mostrada na figura 55, onde terá a possibilidade de escolher entre buscar um arquivo armazenado no dispositivo ou utilizar a câmera para digitalizar um documento. Optando pela digitalização, a interface exibida na figura 56 será apresentada, a qual possui a funcionalidade de detectar automaticamente o documento, garantindo uma captura de maior qualidade.

Figura 55 – Anexar arquivo

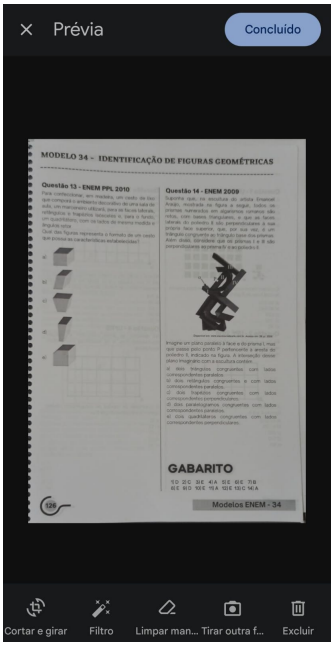
Figura 56 – Digitalizar documento



Fonte: Do autor.

Após a digitalização, o sistema oferece ao usuário a possibilidade de editar a imagem, utilizando a interface ilustrada na figura 57, proporcionando um ajuste refinado do documento digitalizado.

Figura 57 – Editar Digitalização



Fonte: Do autor.

5 Garantia da Qualidade e Processo de Testes

A garantia da qualidade em um projeto de desenvolvimento de software é um processo fundamental para assegurar que o produto final atenda aos requisitos de qualidade, com foco na entrega de um aplicativo funcional, eficiente e com o mínimo de falhas. Esse processo envolve uma abordagem sistemática para estruturar, planejar e executar atividades que garantam a conformidade do produto com os padrões de qualidade estabelecidos ao longo do desenvolvimento (BARTIÈ, 2002). Além disso, o controle contínuo da qualidade ajuda a identificar e corrigir erros precocemente, preservando a integridade do software. As práticas de verificação e validação desempenharam um papel essencial na garantia de que o aplicativo atenda às expectativas dos usuários, mantenha um alto nível de desempenho e seja capaz de lidar com possíveis falhas de maneira eficiente.

De acordo com Bartiè (2002), a garantia da qualidade de software abrange a estruturação, sistematização e execução das atividades necessárias para garantir o desempenho adequado de cada fase do desenvolvimento, de modo a satisfazer os padrões de qualidade previamente definidos. No contexto deste projeto, a garantia da qualidade foi assegurada por meio de três frentes principais: testes manuais realizados pelo desenvolvedor, testes automatizados e testes de usabilidade conduzidos por um grupo de usuários comuns.

5.1 Testes Manuais

No processo de desenvolvimento de software, é comum que a própria equipe responsável pela criação do sistema também atue como testadores do produto. Essa prática envolve a realização de testes manuais, nos quais o desenvolvedor, ao implementar novas funcionalidades ou realizar ajustes, testa diretamente o sistema para identificar possíveis erros, bugs ou inconsistências no comportamento do software. Esses testes são essenciais para garantir que o código atenda aos requisitos funcionais e técnicos especificados e que o sistema opere de maneira robusta e eficiente (SOMMERVILLE, 2011).

No decorrer do desenvolvimento deste projeto, o autor adotou uma metodologia de verificação contínua, na qual cada nova funcionalidade ou ajuste foi imediatamente testado experimentalmente de forma manual. Por exemplo, ao implementar a funcionalidade de cadastro de consultas, o fluxo completo de preenchimento dos dados foi verificado para garantir que o sistema permitisse a inserção correta das informações do paciente, a vinculação com o médico responsável e a adição de anexos.

Os defeitos encontrados foram registrados e solucionados em tempo hábil, conforme a Tabela 8.

Tabela 8 – Ocorrências de Defeitos e Ações Corretivas

Defeito	**Ações Corretivas**
Ordenação de dados médicos incorreta	Ajustada a lógica de ordenação para garantir que os dados médicos fossem listados corretamente por data ou outro critério relevante.
Pacientes viam dados de outros pacientes	Corrigida a lógica de permissões para garantir que cada paciente veja apenas seus próprios dados.
Médico via registros de todos os pacientes ao visualizar registros médicos de um paciente específico	Corrigida a requisição enviada, limitando o escopo da visualização apenas ao paciente selecionado.
Foto de perfil para usuários sem foto ficava com erro em vez de mostrar uma foto padrão	Implementado um *placeholder* de foto padrão para usuários sem imagem de perfil.
Campo de CPF não validava o CPF	Adicionada uma função de validação de CPF para garantir que apenas números válidos fossem aceitos.
Campo de CPF sem máscara	Adicionada uma máscara de entrada ao campo CPF, formatando automaticamente o número inserido.
Listagem de registros médicos não tinha texto indicativo quando não haviam registros para listar	Inserida uma mensagem padrão informando que não há registros médicos disponíveis.
A barra de rolagem era mostrada na listagem de registros médicos, atrapalhando a visualização	Ocultada a barra de rolagem da lista.
Ao abrir o perfil de um paciente, um médico sem permissão visualizava rapidamente os botões de acesso a registros médicos	Mostrar os botões somente após a confirmação das permissões de acesso.
Mensagem de sucesso para cadastro não era exibida	Adicionada uma mensagem de sucesso ao fim do processo de cadastro, garantindo *feedback* visual ao usuário.

Esses testes manuais foram fundamentais para identificar inconsistências no comportamento esperado e garantir que o sistema operasse corretamente em diferentes cenários.

5.2 Testes Automatizados

Testes automatizados são programas ou *scripts* que exercitam as funcionalidades do sistema e realizam verificações automáticas nos resultados obtidos(BERNARDO; KON, 2008). A implementação de testes automatizados é fundamental para assegurar o bom funcionamento do sistema de forma contínua, especialmente em um ambiente de desenvolvimento ágil e iterativo. Alguns testes automatizados foram implementados no sistema como exemplos.

5.2.1 Testes Automatizados na API

Para a API, foram utilizados os *frameworks Jest* e *Supertest*, ambos amplamente usados no ecossistema JavaScript. O *Jest* foi responsável pela execução de testes unitários, enquanto o *Supertest* foi empregado para simular requisições HTTP e validar o comportamento das rotas da API.

Um exemplo prático de teste automatizado foi a validação da criação de alergias por meio da API. A Listagem 5.1, apresenta parte do código do teste automatizado implementado para verificar o processo de criação de alergias.

Listagem 5.1 – Trecho do código de teste para criação de alergias

```
1  describe("POST /allergy", () => {
2    it("should create a new allergy with valid data", async () => {
3      const response = await request(app).post("/allergy").send({
4        patientId: "a03b4ab1-55a7-47a3-ac02-d490eb32980f",
5        name: "Rinite alérgica",
6        description:
7          "Muita tosse e coriza, principalmente ao ter contato com corantes",
8      });
9      expect(response.status).toBe(201);
10     expect(response.body).toHaveProperty("id");
11     expect(response.body.name).toBe("Rinite alérgica");
12     expect(response.body.description).toBe(
13       "Muita tosse e coriza, principalmente ao ter contato com corantes"
14     );
15   });
16
17   it("should return error for missing required fields", async () => {
18     const response = await request(app).post("/allergy").send({
19       patientId: "a03b4ab1-55a7-47a3-ac02-d490eb32980f",
20     });
21     expect(response.status).toBe(400);
22   });
23 });
```

5.2.2 Testes Automatizados no Aplicativo

No aplicativo, testes automatizados foram implementados usando o *Jest* e a *React Testing Library*. Esses testes foram úteis para verificar o comportamento dos componentes e assegurar que as interações de usuário, como cliques em botões e preenchimento de formulários, ocorressem corretamente.

Um exemplo prático de teste focou em garantir que todos os botões de navegação para o registro médico estivessem presentes na tela inicial do paciente. A Listagem 5.2 demonstra o trecho correspondente a esse teste.

Listagem 5.2 – Trecho do código de teste para verificação de botões na home do paciente

```
1 import React from 'react';
2 import { render, screen } from '@testing-library/react';
3 import '@testing-library/jest-dom/extend-expect';
4 import PatientHome from '@/app/(app)/patient/home';
5
6 describe('PatientHome Component', () => {
7   it('deve renderizar todos os botões de navegação para registros médicos', () =>
        {
8     render(<PatientHome />);
9
10    expect(screen.getByText('Consultas')).toBeInTheDocument();
11    expect(screen.getByText('Exames')).toBeInTheDocument();
12    expect(screen.getByText('Alergias')).toBeInTheDocument();
13    expect(screen.getByText('Diagnósticos')).toBeInTheDocument();
14    expect(screen.getByText('Cirurgias')).toBeInTheDocument();
15    expect(screen.getByText('Prescrições')).toBeInTheDocument();
16    expect(screen.getByText('Histórico')).toBeInTheDocument();
17    expect(screen.getByText('Dietas')).toBeInTheDocument();
18    expect(screen.getByText('Biodados')).toBeInTheDocument();
19  });
20 });
```

Embora a cobertura de testes automatizados ainda não seja completa, o esforço inicial já permitiu a validação de algumas partes essenciais do sistema. No entanto, para garantir uma maior robustez e confiabilidade do software, é fundamental que o desenvolvimento de testes automatizados seja ampliado no futuro, visando aumentar a cobertura e a abrangência desses testes. Sem uma cobertura adequada, a equipe de desenvolvimento pode depender excessivamente de testes manuais, que são mais suscetíveis a falhas humanas, além de serem mais demorados na verificação de todas as interações entre os componentes do sistema.

5.3 Testes de Usabilidade

A fim de validar a usabilidade do sistema, foi aplicado o método *System Usability Scale* (SUS) (BROOKE, 1995). O teste foi conduzido com cinco usuários, sem experiência

prévia na área médica, simulando interações comuns dentro do sistema.

5.3.1 Escolha dos Participantes

Os testadores selecionados foram pessoas comuns, sem experiência prévia na área médica ou com sistemas especializados em saúde. A intenção foi avaliar como um usuário leigo interage com a interface do aplicativo, observando como um público sem formação médica navegaria e realizaria tarefas simples e complexas dentro do sistema, assegurando que a usabilidade fosse acessível a qualquer tipo de usuário.

No entanto, reconhece-se que o uso de testadores sem formação médica para avaliar funcionalidades específicas do papel de médico pode ser uma limitação técnica. Embora a intenção fosse avaliar a facilidade de uso e a clareza da interface, é possível que profissionais médicos identificassem necessidades ou expectativas que não foram consideradas durante este teste. Portanto, para validar completamente a usabilidade das funcionalidades voltadas para médicos, seriam recomendáveis testes adicionais com profissionais da saúde em futuras iterações.

5.3.2 Orientações Fornecidas aos Participantes

Antes de iniciar os testes, os participantes foram devidamente orientados sobre as funcionalidades do sistema e receberam um roteiro claro para simular dois papéis: o de paciente e o de médico. A orientação foi projetada para que, apesar de os usuários não terem formação na área da saúde, eles pudessem seguir o fluxo de uso como um médico ou paciente realizaria. Não se esperava que os testadores avaliassem a exatidão clínica das funcionalidades, mas sim a facilidade de navegação, a clareza das informações e a eficiência do sistema ao desempenhar as tarefas propostas.

Foi ressaltado aos participantes que o foco principal do teste era avaliar o quão intuitivo e acessível o sistema seria para um usuário comum, sem conhecimentos técnicos ou clínicos avançados. O objetivo era identificar possíveis barreiras de usabilidade que poderiam impactar negativamente tanto médicos quanto pacientes ao utilizar o aplicativo.

5.3.3 Roteiro de Teste de Usabilidade

Os testes de usabilidade seguiram um roteiro detalhado, no qual os participantes simularam o uso do sistema tanto como médicos quanto como pacientes. As atividades incluíram o cadastro de novas contas, navegação pelas diferentes telas, gerenciamento de informações médicas e o uso das permissões de acesso, além de funcionalidades específicas como o anexo de documentos digitalizados.

Para organizar o teste, dois fluxos distintos foram definidos: um para o papel do

médico e outro para o paciente, conforme descrito nas Tabelas 9 e 10. Cada fluxo cobriu as principais funcionalidades oferecidas pelo aplicativo para ambos os perfis de usuários.

Tabela 9 – Roteiro de Teste de Usabilidade - Fluxo do Médico

Passo	**Descrição da Tarefa**
1	Efetuar o cadastro de usuário completo como médico.
2	Realizar login como médico.
3	Buscar um paciente pré-cadastrado no sistema
4	Solicitar permissão de acesso para um paciente
5	Selecionar um paciente e acessar seu histórico médico.
6	Adicionar uma novo registro médico ao histórico do paciente.
7	Visualizar um registro médico do paciente.
8	Editar um registro médico do paciente.
9	Excluir um registro médico do paciente.
10	Inserir um anexo digitalizado em um exame.
11	Alterar informações de perfil.
12	Sair do sistema.

No fluxo do médico, as principais tarefas envolvem o gerenciamento dos registros médicos dos pacientes, como a criação, edição, visualização e exclusão, além da inserção de documentos digitalizados. Esses passos simulam o cotidiano de um médico utilizando o sistema para atualizar o histórico de seus pacientes.

Tabela 10 – Roteiro de Teste de Usabilidade - Fluxo do Paciente

Passo	**Descrição da Tarefa**
1	Efetuar o cadastro de usuário completo como paciente.
2	Realizar login como paciente.
3	Buscar um médico pré-cadastrado no sistema
4	Selecionar um médico e visualizar seus dados.
5	Permitir acesso do médico aos seus dados.
6	Visualizar um registro médico.
7	Tentar fazer download de um anexo de exame.
8	Visualizar o histórico de acessos ao seu perfil.
9	Alterar informações de perfil.
10	Sair do sistema.

No fluxo do paciente, os participantes testaram funcionalidades relacionadas à visualização dos próprios dados médicos, permissões de acesso para médicos e download de anexos, além de verificarem o histórico de acessos ao seu perfil. Esse fluxo simula o uso por pacientes monitorando seu histórico médico e gerenciando as permissões concedidas a seus médicos.

5.3.4 Análise dos Resultados dos Testes de Usabilidade

Após os testes, foi realizada uma análise dos resultados obtidos através do questionário SUS aplicado aos cinco participantes dos testes de usabilidade. As respostas de cada questionário podem ser visualizadas no Apêndice E. A pontuação final de cada usuário foi calculada com base em suas respostas às 10 questões do questionário, variando de 0 a 100, onde valores mais altos indicam melhor usabilidade percebida. A tabela abaixo resume as pontuações:

Tabela 11 – Pontuações SUS dos Usuários

Usuário	**Pontuação SUS**
Usuário 1	80
Usuário 2	77,5
Usuário 3	87,5
Usuário 4	82,5
Usuário 5	87,5

Fonte: O autor.

A pontuação média SUS foi de 83,0, um valor que reflete uma percepção bastante positiva da usabilidade do sistema. De acordo com escala proposta por Sauro e Lewis (2016), o sistema receberia classificação A, indicando uma boa usabilidade.

5.3.5 Análise Individual dos Usuários

Cada usuário respondeu ao questionário com base em suas experiências ao realizar as tarefas propostas, tanto no fluxo de uso como médico quanto como paciente.

- **Usuário 1**: Obteve uma pontuação de 80, indicando uma percepção bastante positiva. As respostas sugerem que, embora o usuário tenha encontrado algumas dificuldades no sistema, ele se sentiu seguro ao utilizá-lo e considerou o sistema fácil de aprender e usar.
- **Usuário 2**: A pontuação de 77,5 mostra uma experiência semelhante ao usuário 1, mas com uma percepção um pouco mais negativa em relação à complexidade do sistema.
- **Usuário 3**: Com uma das pontuações mais altas, 87,5, o usuário considerou o sistema fácil de usar e integrado, sem grandes problemas de inconsistência. Este resultado sugere que, para este usuário, a curva de aprendizado foi baixa, indicando uma boa adaptação às funcionalidades do sistema.
- **Usuário 4**: Obteve uma pontuação de 82,5, também muito próxima da média geral.

As respostas indicam uma experiência positiva no geral, embora este usuário tenha relatado a necessidade de suporte técnico em algumas situações.

- **Usuário 5**: Com uma pontuação de 87,5, este usuário compartilhou uma experiência bastante positiva, semelhante ao usuário 3.

5.3.6 Interpretação dos Resultados

Os resultados sugerem que o sistema apresentou um bom nível de usabilidade, com pontuações acima de 77 em todos os casos, o que é significativamente superior ao valor médio esperado para sistemas comuns (68). A variação nas respostas reflete a diversidade nas percepções dos usuários, alguns necessitaram de suporte técnico, enquanto outros enfatizaram a facilidade de uso e a segurança do sistema.

A pontuação geral indica que o sistema foi bem recebido pelos participantes, com destaque para sua interface intuitiva e a boa integração das funcionalidades.

Esses resultados, portanto, indicam que o sistema está no caminho certo em termos de usabilidade, mas ainda pode ser aprimorado para atingir um público mais amplo, especialmente usuários com menos experiência tecnológica.

5.3.7 *Feedbacks* Coletados

Durante os testes de usabilidade, foi solicitado que os participantes fornecessem *feedbacks* espontâneos sobre o uso do sistema, destacando pontos positivos e possíveis áreas de melhoria. A Tabela 12 resume os comentários obtidos.

Tabela 12 – *Feedback* dos Usuários sobre a Usabilidade

Usuário	**Comentário**
Usuário 1	"Gostei muito da fluidez ao navegar entre as telas. A funcionalidade de digitalização foi muito rápida, sem dificuldades."
Usuário 2	"A funcionalidade de buscar pacientes e médicos é ótima, mas senti falta de uma opção de filtro mais avançada para facilitar a pesquisa."
Usuário 3	"A interface é clara e fácil de entender, mas alguns campos de texto poderiam ser trocados por listas de seleção, como estado e cidade."
Usuário 4	"O aplicativo é intuitivo, mas percebi que ao abrir a galeria do celular para anexar uma foto de perfil, houve um pequeno atraso."
Usuário 5	"Os campos só são validados ao tentar salvar, seria mais útil se houvesse validação em tempo real enquanto preencho o formulário."

Neste estágio de desenvolvimento, o foco principal foi a validação das funcionalidades essenciais e da usabilidade geral do sistema, a fim de garantir que o aplicativo atenda aos requisitos básicos. As sugestões fornecidas pelos usuários durante os testes de usabilidade foram cuidadosamente registradas e poderão ser avaliadas e priorizadas para iterações futuras. O objetivo é garantir que o aplicativo evolua continuamente, com base no *feedback* dos usuários, e se mantenha alinhado às expectativas e necessidades à medida que novas versões sejam lançadas.

6 Resultados alcançados e considerações finais

O processo de desenvolvimento deste aplicativo foi estruturado para a entrega de uma solução minimamente robusta e eficiente, dentre os parâmetros de testes realizados. O uso de metodologias de engenharia de software, combinando modelagem de dados, padrões de projeto e práticas modernas de desenvolvimento, desempenhou um papel fundamental no desenvolvimento e na qualidade do projeto.

A modelagem de dados, através da criação de diagramas de classe e de banco de dados, proporcionou uma base sólida para a estruturação das informações. Essa etapa foi fundamental para mapear as entidades envolvidas no sistema. Adicionalmente, a definição detalhada dos casos de uso e requisitos funcionais facilitou a compreensão das interações entre os diferentes usuários e as funcionalidades do sistema. A prototipagem, realizada com o uso do *Figma*, também desempenhou um papel crucial ao permitir que as interfaces fossem testadas e validadas antes do desenvolvimento propriamente dito. Essas abordagens colaborativas possibilitaram identificar melhorias e ajustes no fluxo do sistema, assegurando uma maior assertividade no processo de desenvolvimento.

A aplicação de padrões de projeto e boas práticas de desenvolvimento contribuiu para um código mais organizado, flexível e reutilizável, o que favorece a escalabilidade e a manutenção do sistema a longo prazo. A utilização de tecnologias modernas, como o *React Native* com *Expo*, para o desenvolvimento do aplicativo, e a implementação da API utilizando *Node.js* possibilitaram uma comunicação eficiente entre o cliente e o servidor. Esses recursos permitiram que as funcionalidades fossem integradas com sucesso, preservando a integridade dos dados e assegurando que a segurança e acessibilidade fossem geridas de maneira controlada. Médicos e pacientes possuem acessos diferenciados, com permissões adequadas ao seu papel, conforme estabelecido nos requisitos.

A API construída foi hospedada em um ambiente público e, para facilitar o acesso dos usuários, foi gerado um arquivo APK do aplicativo, permitindo a instalação em dispositivos móveis. Essa etapa foi essencial para a realização dos testes, onde as principais funcionalidades do sistema puderam ser experimentadas e avaliadas.

Os testes de usabilidade foram conduzidos com um grupo de pessoas comuns, que simularam o uso das funcionalidades do sistema. Embora os testes não tenham envolvido médicos e pacientes reais, os resultados indicaram um bom nível de aceitação, especialmente no que diz respeito à acessibilidade e à facilidade de uso das interfaces, o que reforça o potencial do sistema para o público-alvo final.

Este trabalho, portanto, apresenta um potencial significativo para melhorar a qualidade da assistência médica, oferecendo maior praticidade tanto para pacientes quanto para profissionais de saúde. Ao centralizar o histórico médico do paciente de maneira eficiente e acessível, o sistema contribui para a continuidade do cuidado, promovendo uma gestão mais organizada das informações médicas e impactando positivamente o setor de saúde.

6.1 Considerações sobre os limites do sistema

Embora o aplicativo tenha sido idealizado de forma que apenas os médicos possam inserir dados no histórico médico dos pacientes, deixando os pacientes com a permissão de apenas visualizar suas informações, essa abordagem apresenta alguns desafios. Um dos principais pontos fracos é a possibilidade de perda de informações devido à falta de preenchimento por parte do médico, seja por sobrecarga de trabalho, descuido ou mesmo falta de comprometimento no uso do aplicativo e sua proposta. Além disso, essa restrição pode causar lentidão no processo de inserção de dados, especialmente em momentos em que o tempo é um recurso crítico.

Para mitigar esses problemas, uma possível solução seria a liberação de certas funcionalidades para que os próprios pacientes possam preencher alguns dados de saúde de forma de forma independente. Informações como alergias, cirurgias, dietas, histórico familiar e diagnósticos poderiam ser fornecidas diretamente pelos pacientes, melhorando a completude dos registros. Da mesma forma, permitir que os pacientes anexem documentos importantes ao seu histórico facilitaria a obtenção de diagnósticos mais precisos no futuro.

A conformidade com a Lei Geral de Proteção de Dados (LGPD)(BRASIL, 2018) e o respeito ao Código de Ética Médica são essenciais para garantir a legitimidade e a integridade dos dados tratados pelo aplicativo. Para assegurar que as informações coletadas e geridas estejam em conformidade com essas legislações, será necessário realizar uma investigação com médicos e órgãos de saúde competentes. Essa investigação terá como objetivo validar os dados inseridos no sistema, assegurando que os processos de coleta, armazenamento e compartilhamento de informações respeitem os princípios da LGPD, como a transparência, a finalidade e a segurança dos dados. Além disso, a colaboração com profissionais da saúde permitirá alinhar as funcionalidades do aplicativo com as diretrizes éticas, garantindo que as práticas de atendimento respeitem a privacidade dos pacientes e a confidencialidade das informações médicas. Esse compromisso com a conformidade legal e ética não apenas fortalecerá a confiança dos usuários no sistema, mas também promoverá um ambiente mais seguro e responsável para o gerenciamento de dados de saúde.

Outro ponto a ser considerado é a falta de uma validação automática do CRM dos médicos que realizam o cadastro no sistema, ou da própria identidade deles. A segurança

atual do sistema baseia-se na autorização concedida pelos pacientes, o que pode não ser suficiente para prevenir fraudes, visto que o aplicativo, em sua forma atual, não garante a veracidade das credenciais médicas. Para melhorar a segurança, seria necessário integrar uma solução que permita verificar a identidade dos médicos, minimizando o risco de uso indevido do sistema.

Por fim, para garantir que o sistema atenda de forma mais adequada às necessidades dos profissionais de saúde, recomenda-se um estudo mais aprofundado do ponto de vista médico. Esse estudo, idealmente conduzido com a colaboração de médicos experientes, poderá trazer sugestões técnicas valiosas para o aprimoramento das funcionalidades e assegurar que o sistema se alinha às práticas do dia a dia no ambiente clínico.

6.2 Trabalhos Futuros

Apesar dos resultados positivos, há uma série de melhorias e funcionalidades que podem ser implementadas em versões futuras do aplicativo e que foram idealizadas durante o processo de desenvolvimento e testes. Entre as principais sugestões para trabalhos futuros estão:

- **Visualização gráfica**: A inclusão de gráficos para a exibição de dados históricos, como a evolução de peso e pressão arterial, pode oferecer *insights* mais profundos tanto para os médicos quanto para os pacientes, auxiliando na tomada de decisões.
- **Agendamento de consultas**: Uma funcionalidade de agendamento permitirá que os pacientes marquem consultas diretamente pelo aplicativo, com integração ao sistema de calendários dos médicos.
- **Permissão de preenchimento de dados pelos pacientes**: Para evitar possíveis perdas de dados pela ausência de preenchimento por parte dos médicos, algumas funcionalidades podem ser liberadas para que os pacientes insiram informações importantes, como alergias, cirurgias, dietas, histórico familiar e anexação de documentos. Isso garantiria maior completude dos dados médicos.
- **Validação do CRM e identidade dos médicos**: Implementar um sistema de validação do CRM e da identidade dos médicos que acessam o sistema, a fim de evitar fraudes e garantir a legitimidade dos profissionais que fazem uso da plataforma.
- **Melhorias na interação com o usuário**: A adição de elementos de interface mais intuitivos, como calendários embutidos nos campos de data e sugestões automáticas de preenchimento, pode facilitar a usabilidade.
- **Notificações personalizadas**: A inclusão de notificações *push*, alertando pacientes

sobre a necessidade de atualizar seus registros, realizar consultas periódicas ou dicas de bem-estar, pode aumentar a proatividade no cuidado à saúde.

- **Expansão da acessibilidade**: Expandir acessibilidade do aplicativo, visando aumentar ainda mais a inclusão de usuários com necessidades especiais. Isso pode ser feito com suporte para leitores de tela, oferecer um modo de alto contraste para facilitar a leitura, adicionar navegação por voz e vários outros recursos de acessibilidade

- **Estudo com médicos experientes**: Realizar estudos futuros com médicos experientes para avaliar a efetividade e usabilidade do sistema sob a perspectiva dos profissionais da saúde. Isso permitirá ajustes que alinhem ainda mais as funcionalidades com as necessidades práticas do ambiente clínico.

- **Base de alergias**: Atualmente, o médico preenche manualmente o nome e a descrição da alergia. Seria vantajoso implementar uma base de dados com alergias previamente cadastradas, permitindo que o médico selecione a alergia desejada, facilitando assim o preenchimento e garantindo maior precisão nas informações.

- **Revogação de acesso**: Implementar uma nova abordagem para a revogação do acesso de um médico aos dados do paciente. O médico deve ser notificado sobre a revogação, permitindo que ele confirme a ação, reforçando a importância da relação médico-paciente.

- **Configuração de permissões de acesso**: Implementar uma funcionalidade que permita ao paciente configurar quais informações cada médico poderá acessar. Isso daria ao paciente maior controle sobre seus dados, permitindo definir restrições de acesso para categorias específicas. Dessa forma, o paciente teria mais autonomia e privacidade na gestão de suas informações de saúde, em conformidade com os princípios da LGPD.

Essas melhorias visam tornar o aplicativo ainda mais completo, adaptando-se a um número maior de cenários e ampliando suas funcionalidades para oferecer um acompanhamento contínuo e mais eficiente da saúde dos pacientes.

6.3 Conclusão

A conclusão deste trabalho evidencia o êxito no desenvolvimento de uma versão móvel do WHC (*Web Health Care*), além da criação de uma API REST em *Node.js* que permitiu a integração entre o aplicativo e o servidor. O processo de desenvolvimento focou em atender de forma mais eficaz as necessidades dos usuários, reformulando a engenharia

de requisitos e os diagramas para adaptar o sistema ao ambiente móvel, sempre com o objetivo de aprimorar a usabilidade e a eficiência.

Diversas funcionalidades adicionais foram implementadas, superando o escopo inicial do trabalho base de Depolli (2022). Entre as inovações, destacam-se o gerenciamento de alergias e o registro de histórico familiar, funcionalidades que agregam valor ao monitoramento da saúde dos pacientes. A digitalização de documentos, que permite o *upload* direto de arquivos via câmera, também tornou o sistema mais ágil e moderno, facilitando o armazenamento de informações relevantes, como exames e atestados.

Outras melhorias incluem o desacoplamento de dietas, biodados e prescrições dos demais registros médicos, tornando-os registros médicos independentes no sistema e acessíveis diretamente. Esse ajuste foi feito para garantir maior flexibilidade no gerenciamento e consulta desses dados. Adicionalmente, foi incorporada a funcionalidade de busca em listagens de registros médicos, permitindo uma navegação mais rápida e precisa por parte dos usuários. O controle de acesso aos dados dos pacientes foi aprimorado, com a inclusão do histórico de acessos, a possibilidade do paciente conceder ou revogar acessos sem necessidade de solicitações, a remoção da validade de acesso e uma interface mais intuitiva para que os pacientes possam conceder ou revogar permissões de forma imediata.

Modificações no cadastro de usuários também foram realizadas, como a inclusão dos campos de gênero e tipo sanguíneo, reforçando a personalização do sistema para contextos médicos. Para consultas, foram adicionados campos de anamnese, queixa principal e sintomas, além da possibilidade de anexar múltiplos arquivos a exames, facilitando o acompanhamento completo das condições de saúde do paciente.

Embora os testes de usabilidade não tenham sido realizados com usuários reais, o sistema alcançou uma pontuação satisfatória na escala SUS, indicando que o aplicativo é acessível e fácil de usar. Esse resultado reforça a viabilidade do projeto e o seu potencial para ser implementado em ambientes reais.

Em relação ao trabalho de Depolli (2022), este serviu como uma base sólida para entender os principais objetivos e funcionalidades do sistema. No entanto, as adaptações necessárias para um ambiente móvel foram amplas, levando a melhorias significativas em termos de usabilidade e flexibilidade. A nova versão do WHC é, portanto, mais aderente às necessidades atuais dos usuários, apresentando uma interface unificada e um fluxo de navegação simplificado, além de contar com funcionalidades inovadoras que ampliam o escopo do sistema original.

Por fim, este trabalho oferece uma contribuição importante ao aprimorar o sistema já existente e torná-lo mais acessível e eficiente em plataformas móveis. Acredita-se que a aplicação desse sistema em larga escala pode gerar um impacto positivo na qualidade do atendimento médico e na continuidade do cuidado aos pacientes.

7 Cronograma

O cronograma foi proposto de acordo com as fases de construção do sistema, conjugadas com as etapas de elaboração do próprio TCC, sofrendo possíveis modificações caso haja necessidade de adequações de acordo com o andamento incremental do projeto.

	Set 2023	Out 2023	Nov 2023	Dez 2023	Jan 2024	Fev 2024	Mar 2024	Abr 2024	Mai 2024	Jun 2024	Jul 2024	Ago 2024	Set 2024
ETAPA 1	X	X											
ETAPA 2	X	X											
ETAPA 3		X											
ETAPA 4		X	X										
ETAPA 5			X										
ETAPA 6			X	X	X								
ETAPA 7					X	X	X	X	X	X	X		
ETAPA 8								X	X	X	X		
ETAPA 9											X	X	
ETAPA 10												X	X

Descrição das etapas:

ETAPA 1	Revisão da Literatura
ETAPA 2	Identificar as necessidades e levantar os requisitos
ETAPA 3	Definir as características demonstrando em casos de uso
ETAPA 4	Documentação de especificação
ETAPA 5	Estimativas (Tamanho, esforço e prazo)
ETAPA 6	Modelagem do projeto
ETAPA 7	Codificação do sistema
ETAPA 8	Testes locais (pelo desenvolvedor)
ETAPA 9	Testes de usabilidade
ETAPA 10	Entrega final e Apresentação

Referências

BANKS, A.; PORCELLO, E. *Learning React: Functional Web Developmen With React and Redux*. 1. ed. [S.l.]: O'Reilly Media, 2017. ISBN 9781491954621. 22

BARTIÈ, L. *Garantia da Qualidade de Software: Teoria e Prática*. São Paulo: Editora Nova Era, 2002. 68

BATINI, C.; CERI, S.; NAVATHE, S. B. *Conceptual Database Design: An Entity-Relationship Approach*. Redwood: Benjamin/Cummings, 1992. 35

BERNARDO, P. C.; KON, F. A importância dos testes automatizados. *Engenharia de Software Magazine*, v. 1, n. 3, p. 54–57, 2008. 70

BONIFÁCIO, B. et al. Aplicando técnicas de inspeção de usabilidade para avaliar aplicações móveis. Universidade Federal do Amazonas, Amazonas, 2010. 20

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. 6th reimpressão. ed. Rio de Janeiro: Elsevier, 2005. 32

BORDIGNON, R. *Prototipações: para que servem e quais os benefícios?* i9 Consultoria, 2023. Disponível em: <https://eji9consultoria.com.br/prototipacoes-para-que-servem-e-quais-os-seus-beneficios/>. Acesso em: 28 jul. 2024. 47

BRASIL. *Lei n.º 13.709, de 14 de agosto de 2018*. 2018. Acesso em: 1 out. 2024. Dispõe sobre a proteção de dados pessoais e altera a Lei n.º 12.965, de 23 de abril de 2014 (Marco Civil da Internet). Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm>. 53, 78

BRASIL. Você conhece o conecte sus?: Aplicativo mostra toda a sua trajetória de atendimentos no sistema Único de saúde. *Serviços e Informações do Brasil*, 4 2021. Disponível em: <https://www.gov.br/pt-br/noticias/saude-e-vigilancia-sanitaria/2021/04/voce-conhece-o-conecte-sus>. Acesso em: 2 out. 2023. 16

BROOKE, J. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, v. 189, 11 1995. 40, 71

BUENO, L. M. *Kanban: conheça mais sobre essa metodologia e saiba como aplicá-la!* 2023. Disponível em: <https://blogdaqualidade.com.br/kanban>. Acesso em: 28 jul. 2024. 42

CODECADEMY. *React: The Virtual DOM*. Codecademy, 2017. Disponível em: <https://www.codecademy.com/article/react-virtual-dom>. Acesso em: 29 set. 2023. 22

CUNHA, A. React native: o que é e tudo sobre o framework. Alura, 9 2023. Disponível em: <https://www.alura.com.br/artigos/react-native>. Acesso em: 30 set. 2023. 14, 23

DEPOLLI, I. C. *Web Healthy Care: Sistema web de gerenciamento de dados da saúde individual da população*. 61 f. TCC (Bacharelado) — Universidade Federal do Espírito Santo, Espírito Santo, 2022. 13, 15, 16, 24, 25, 30, 35, 81

DOCTORALIA. *Doctoralia*. 2023. Disponível em: <https://pro.doctoralia.com.br/>.

Acesso em: 2 out. 2023. 17

EISENMAN, B. *Learning React Native: Building Native Mobile Apps with JavaScript*. 1. ed. [S.l.]: O'Reilly Media, 2016. ISBN 1491929006. 22, 23

EL-KASSAS, W. S. et al. Taxonomy of cross-platform mobile applications development approaches. *Ain Shams Engineering Journal*, 2015. ISSN 2090-4479. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2090447915001276>. Acesso em: 31 set. 2023. 18, 19

ERWIN. *Modelagem Física de Dados*. 2021. Disponível em: <https://www.erwin.com/br-pt/solutions/data-modeling/physical.aspx>. Acesso em: 13 dez. 2023. 38

FACEBOOK. *React*. GitHub, 2023. Disponível em: <https://github.com/facebook/react>. Acesso em: 29 set. 2023. 21

FACEBOOK. *React Native*. GitHub, 2023. Disponível em: <https://github.com/facebook/react-native>. Acesso em: 30 set. 2023. 22

FERNANDES, D. *TypeScript: vantagens, mitos, dicas e conceitos fundamentais*. Rocketseat, 2018. Disponível em: <https://blog.rocketseat.com.br/typescript-vantagens-mitos-conceitos/>. Acesso em: 30 set. 2023. 21

FERNANDES, M. V. S. *Aplicativo Móvel para Aprendizado sobre Educação Financeira*. 40 f. TCC (Bacharelado) — Universidade Federal de Ouro Preto, Minas Gerais, 2023. 20

Figma. *Figma - The Collaborative Interface Design Tool*. 2023. Disponível em: <https://www.figma.com/>. 24

FILHO, J. L.; IOCHPE, C. Um estudo sobre modelos conceituais de dados para projeto de bancos de dados geográficos. *Revista IP-Informática Pública*, v. 1, n. 2, p. 37–90, 1999. 37

GOV. *Conecte SUS*. 2023. Disponível em: <https://play.google.com/store/apps/details?id=br.gov.datasus.cnsdigital&hl=pt_BR&gl=US>. Acesso em: 2 out. 2023. 16, 17

INTERACTION DESIGN FOUNDATION. What is usability testing? The Interaction Design Foundation, 2019. Disponível em: <https://www.interaction-design.org/literature/topics/prototyping>. Acesso em: 29 set. 2023. 20

JONES, M.; BRADLEY, J.; SAKIMURA, N. *JSON Web Token (JWT)*. [S.l.], 2015. <https://www.rfc-editor.org/rfc/rfc7519>. 45

KHUDOIBERDIEV, A. O. U. *TypeORM Documentation*. 2024. Disponível em: <https://typeorm.io/>. Acesso em: 28 jul. 2024. 43

KUPKA, F. *O que é React Native?* 2017. Disponível em: <https://www.organicadigital.com/seeds/o-que-e-react-native/>. Acesso em: 31 set. 2023. 23

LIMA, A. da S. *UML 2.3 do requisito à solução*. 1st. ed. São Paulo: Érica, 2011. 32

LISBOA, A. *Doctoralia lança ferramenta para você registrar o seu histórico médico*. Canaltech, 2023. Disponível em: <https://canaltech.com.br/apps/doctoralia-lanca-ferramenta-para-voce-registrar-o-seu-historico-medico-236064/>. Acesso em: 2 out. 2023. 17, 18

LONGEN, A. S. *React Descomplicado: O Que é e Como Começar Já!* Hostinger, 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript>. Acesso em: 29 set. 2023. 23

LOURENÇO, D. F.; CARMONA, E. V.; LOPES, M. H. B. de M. Tradução e adaptação transcultural da system usability scale para o português do brasil. *Aquichan*, 2022. 40

MADUABUCHI, C. *Getting started with NativeWind: Tailwind for React Native*. LogRocket Blog, 2024. Disponível em: <https://blog.logrocket.com/getting-started-nativewind-tailwind-react-native/>. 46

MAGALHÃES, A. L. *O que são Web Apps?* Canaltech, 2021. Disponível em: <https://canaltech.com.br/navegadores/o-que-sao-web-apps/>. Acesso em: 31 set. 2023. 19

MARQUES JANNIK WEMPE, C. D. *Zustand: Bear necessities for state management in React*. 2024. <https://github.com/pmndrs/zustand>. Acesso em: 28 jul. 2024. 46

MARTIN, R. C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. [S.l.]: Prentice Hall, 2017. 44

MEDCLIN. *Histórico médico, versão 1.11.1.44*: project planning software. 2023. Disponível em: <https://play.google.com/store/apps/details?id=com.cliniconline&hl=pt_BR&gl=US>. Acesso em: 2 out. 2023. 16

MELO, A. C. *Desenvolvendo Aplicações com UML 2.2 – do Conceitual à Aplicação*. São Paulo: Editora Brasport, 2010. 317 p. 33

MELO, D. *O que é TypeScript? [Guia para iniciantes]*. Tecnoblog, 2020. Disponível em: <https://tecnoblog.net/responde/o-que-e-typescript-guia-para-iniciantes/>. Acesso em: 30 set. 2023. 21

MESH, J. Método kanban: Guia detalhado e 5 modelos prontos para usar. Blog Trello, 3 2020. Disponível em: <https://blog.trello.com/br/metodo-kanban>. Acesso em: 5 out. 2023. 20

MONITORA. *Compreenda as diferenças entre aplicações nativa, web e híbrida*. 2017. Disponível em: <https://www.monitoratec.com.br/blog/aplicacoes-nativa-web-e-hibrida/>. Acesso em: 31 set. 2023. 19

NEVES, V. React: o que é, como funciona e um guia dessa popular ferramenta js. Alura, 1 2023. Disponível em: <https://www.alura.com.br/artigos/react-js>. Acesso em: 30 set. 2023. 21

PATRiCIO, M. et al. O prontuario eletronico do paciente no sistema de saude brasileiro: uma realidade para os medicos? *Scientia Medica*, v. 21, n. 3, p. 121–31, 2011. 13

PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach*. 2. ed. New York: McGraw-Hill, 1987. 35

RÊGO, J. V. V. *BCrypt — Algoritmo de Hash*. Medium, 2020. Disponível em: <https://medium.com/bcrypt-algoritmo-de-hashing/bcrypt-algoritmo-de-hash-44f7a604d1f6>. Acesso em: 28 jul. 2024. 45

ROMAN, F.; GONZALEZ-MESONES, F.; MARINAS, I. *Mobile Marketing: A revolução multimídia*. 1. ed. [S.l.]: Cengage, 2007. ISBN 8522105693. 18

SABINO, R. Kanban: o que é, o método kanban, principais conceitos e como funciona no dia a dia. Alura, 9 2023. Disponível em: <https://www.alura.com.br/artigos/metodo-kanban#o-que-e-kanban?-e-para-que-serve-esse-metodo?> Acesso em: 5 out. 2023. 20

SAURO, J.; LEWIS, J. R. *Quantifying the user experience: Practical statistics for user research*. 2. ed. [S.l.]: Morgan Kaufmann, 2016. 40, 41, 74

SILVA, F. A. de S.; PRADO, E. F. do. Análise teórica sobre o desenvolvimento de aplicativos nativos, híbridos e web aps. *Revista EduFatec: educação, tecnologia e gestão*, v. 2, n. 1, 2019. 19

SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson Education, 2011. ISBN 978-85-7936-108-1. 26, 31, 32, 35, 68

Tailwind Labs. *Tailwind CSS: A utility-first CSS framework for rapid UI development*. [S.l.], 2024. Disponível em: <https://tailwindcss.com/>. 46

TEIXEIRA, F. *O que é o SUS (System Usability Scale) e como usá-lo em seu site*. UX Collective, 2015. Disponível em: <https://brasil.uxdesign.cc/o-que-%C3%A9-o-sus-system-usability-scale-e-como-us%C3%A1-lo-em-seu-site-6d63224481c8>. Acesso em: 29 set. 2023. 40

TELECO. *Estatísticas de Celulares no Brasil*. 2023. Disponível em: <https://www.teleco.com.br/ncel.asp>. Acesso em: 2 out. 2023. 18

TOSTES, R. N.; COSTA, R. M. de R. Aplicações multiplataforma com react: Desenvolvimento de uma aplicação educacional. *ANALECTA-Centro de Ensino Superior de Juiz de Fora*, 2020. Disponível em: <https://seer.cesjf.br/index.php/ANL/article/download/2360/1579>. Acesso em: 30 set. 2023. 13, 23

Trello. *Trello*. 2023. Disponível em: <https://trello.com/>. 24

URMOBO. *Brasil possui 251,6 milhões de celulares ativos. Veja infográfico*. Urmobo, 2023. Disponível em: <https://urmobo.com.br/blog/brasil-possui-2516-milhoes-de-celulares-ativos/>. Acesso em: 1 out. 2023. 18

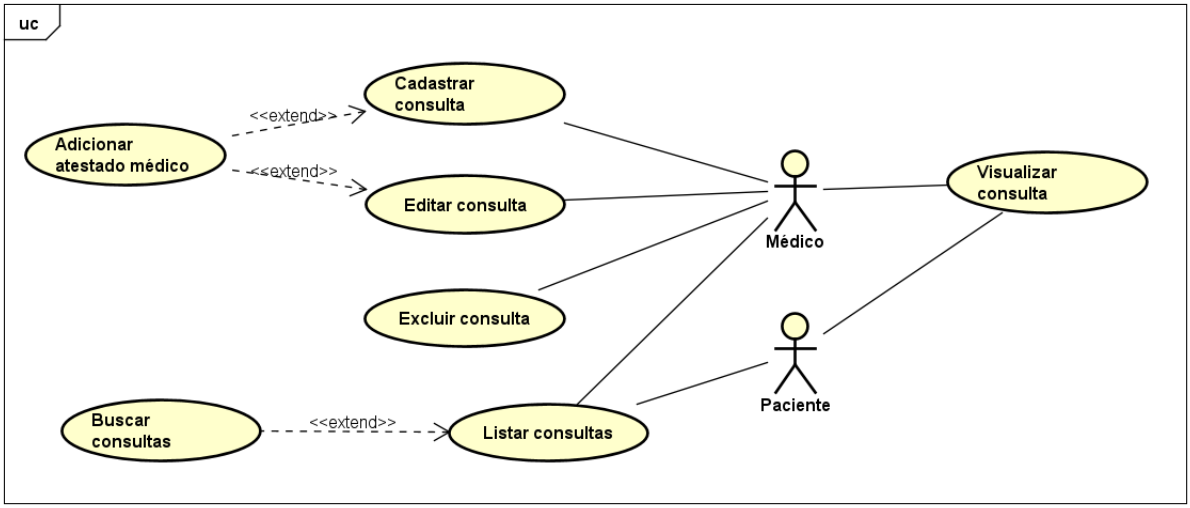
WHATWG COMMUNITY. *DOM Living Standard*. 2023. Disponível em: <https://dom.spec.whatwg.org/>. Acesso em: 5 out. 2023. 21

Wikipedia. *Role-based access control*. 2024. <https://en.wikipedia.org/w/index.php?title=Role-based_access_control&oldid=1226281024>. Acesso em: 28 jul. 2024. 45

Apêndices

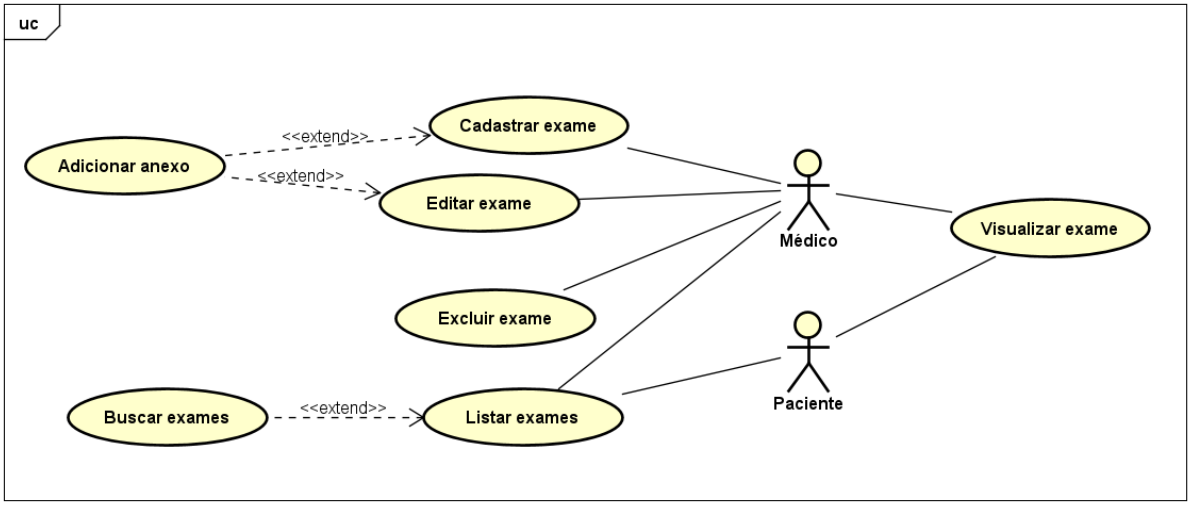
APÊNDICE A – Diagramas de caso de uso

Figura 58 – Gerenciamento de consultas



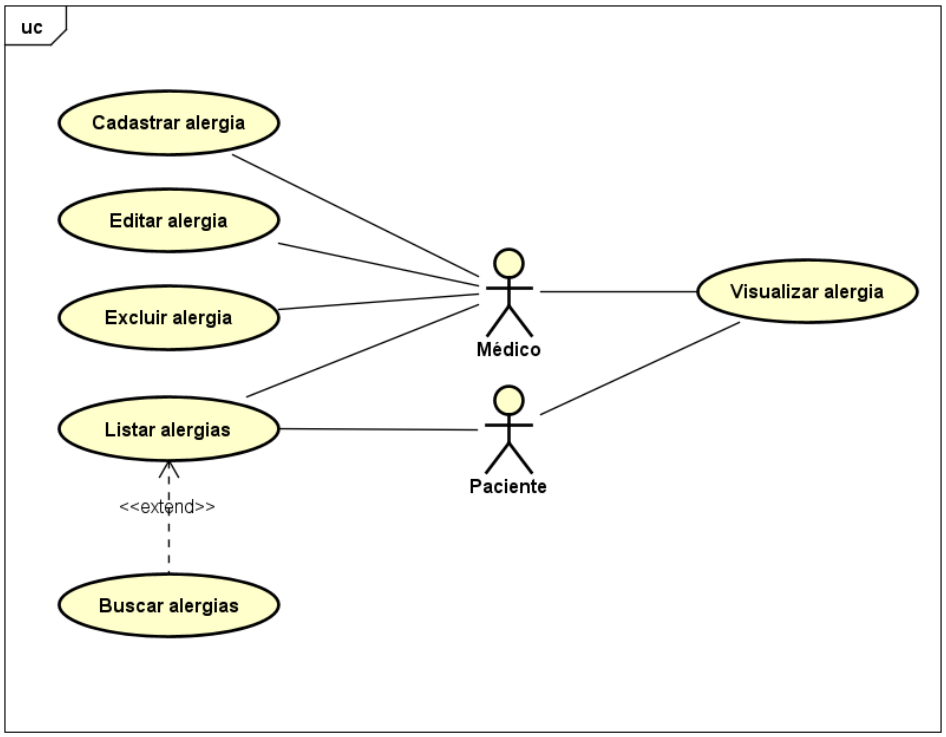
Fonte: O autor.

Figura 59 – Gerenciamento de exames



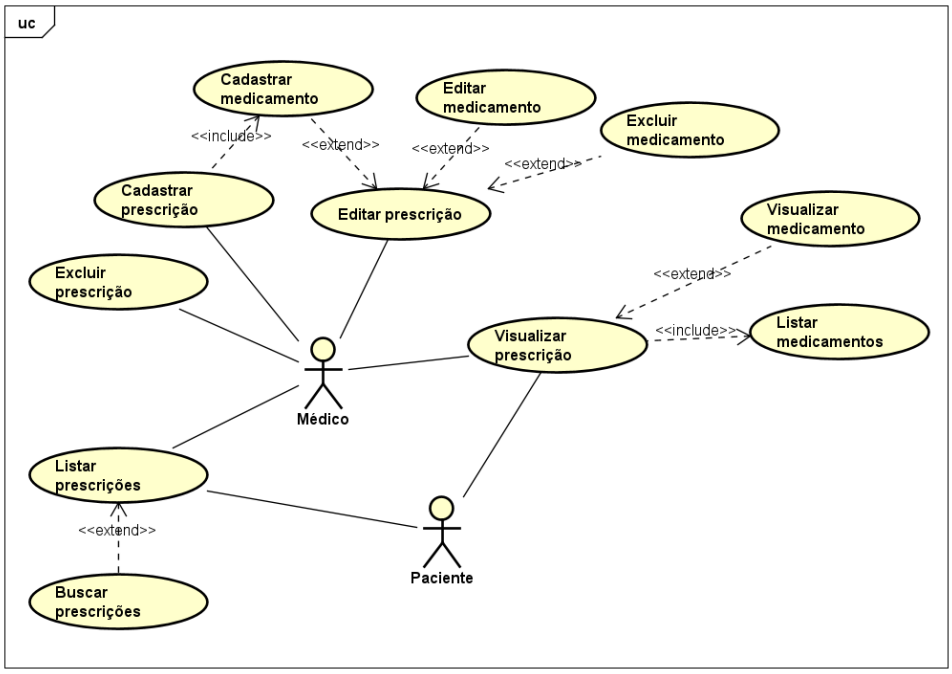
Fonte: O autor.

Figura 60 – Gerenciamento de alergias



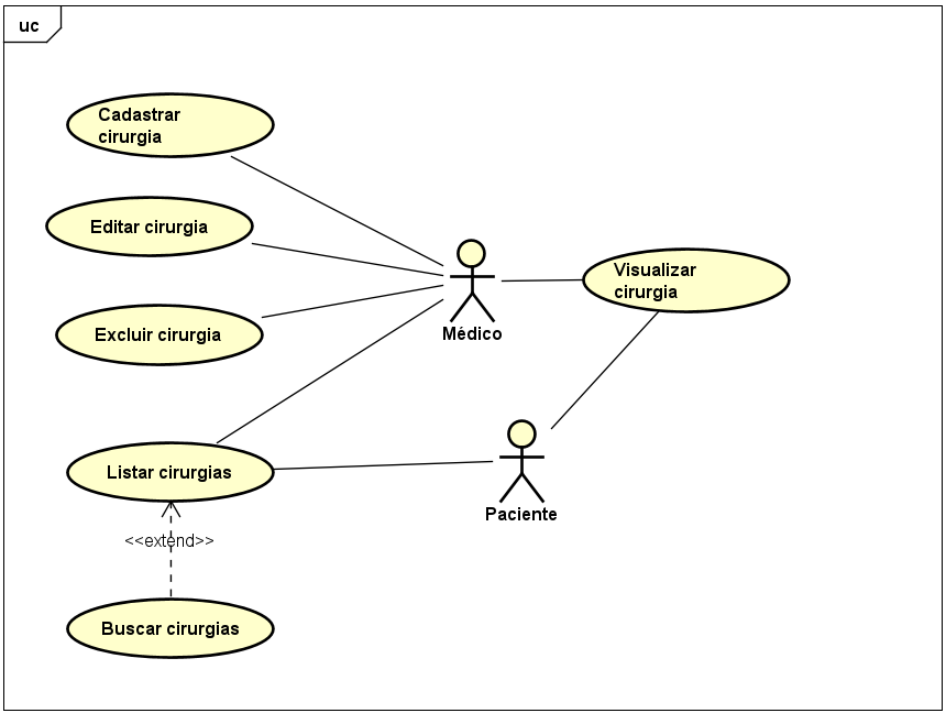
Fonte: O autor.

Figura 61 – Gerenciamento de prescrições



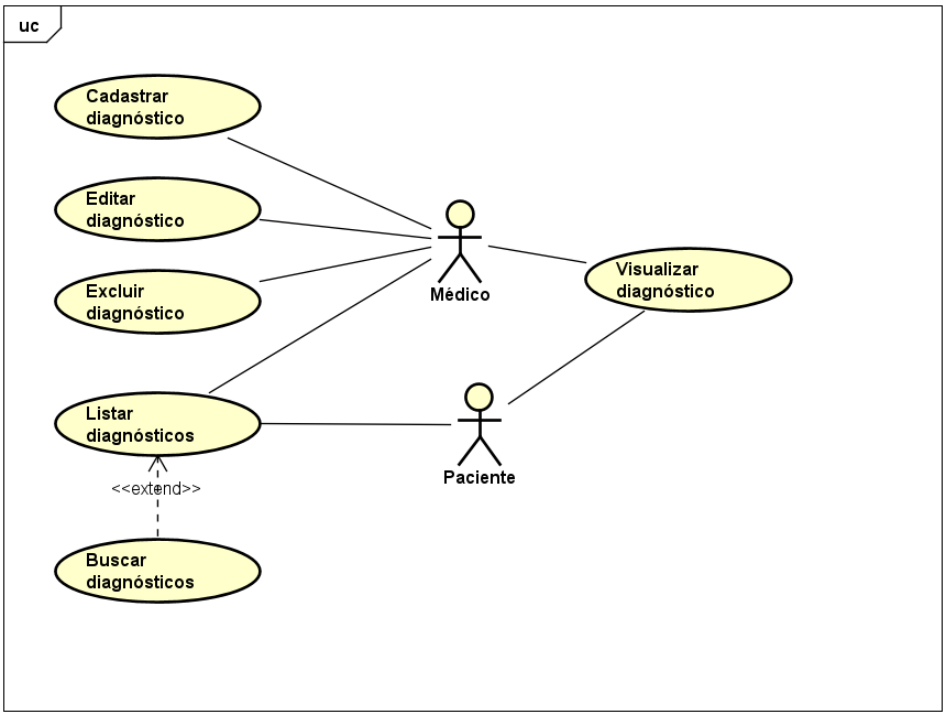
Fonte: O autor.

Figura 62 – Gerenciamento de cirurgias



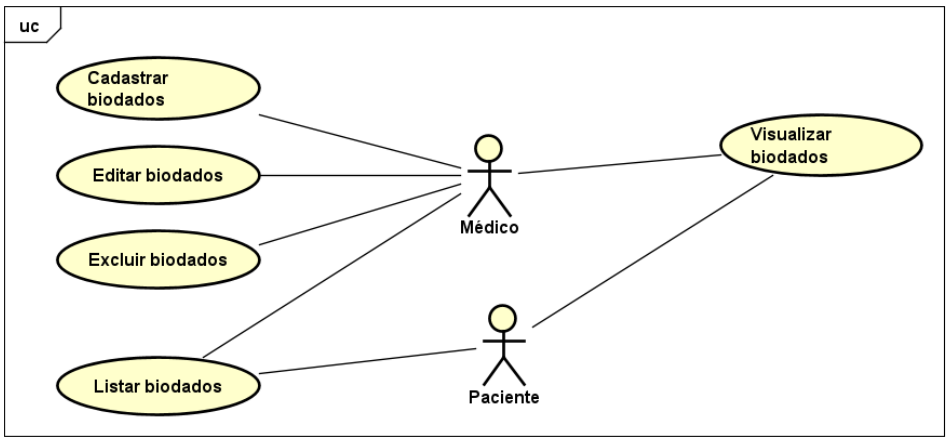
Fonte: O autor.

Figura 63 – Gerenciamento de diagnósticos



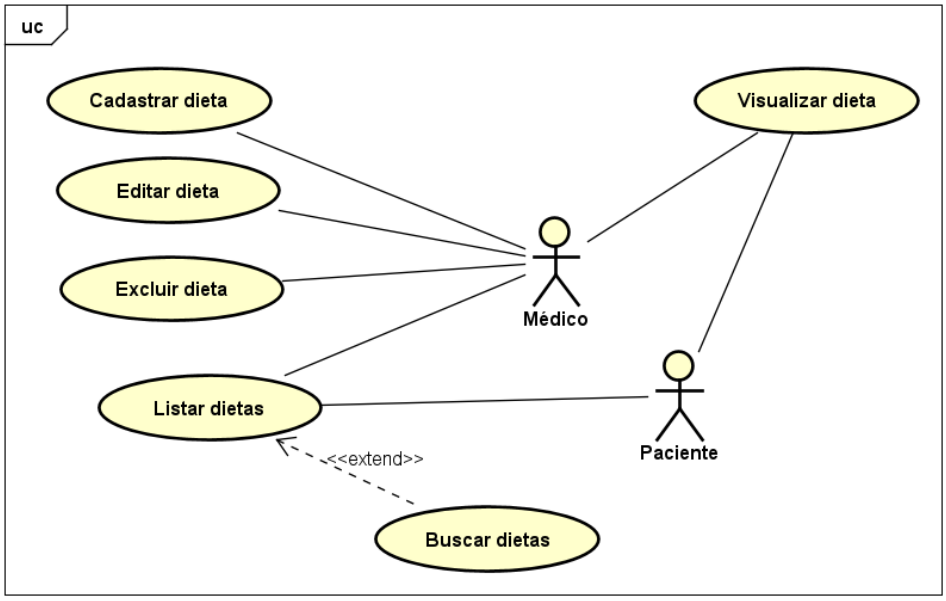
Fonte: O autor.

Figura 64 – Gerenciamento de biodados



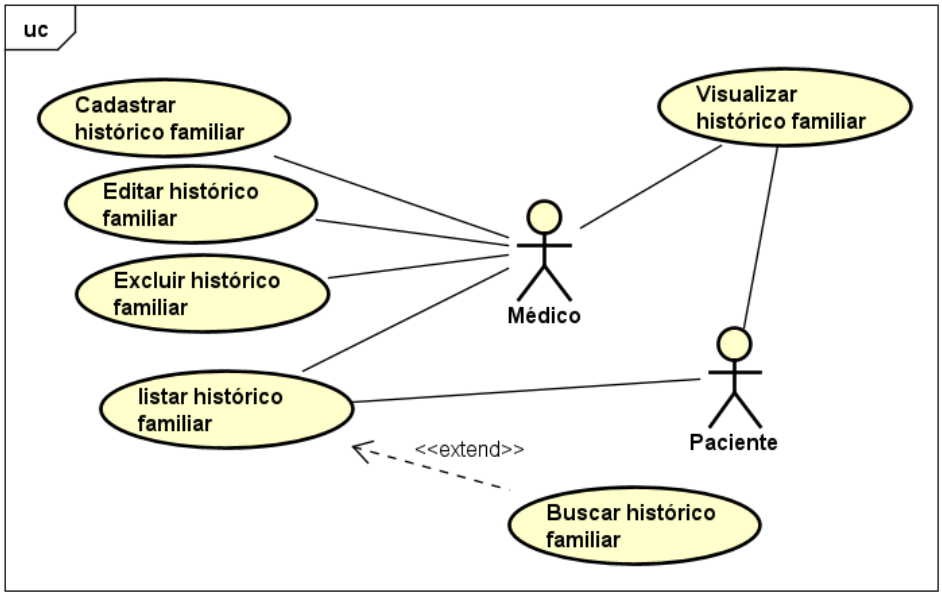
Fonte: O autor.

Figura 65 – Gerenciamento de dietas



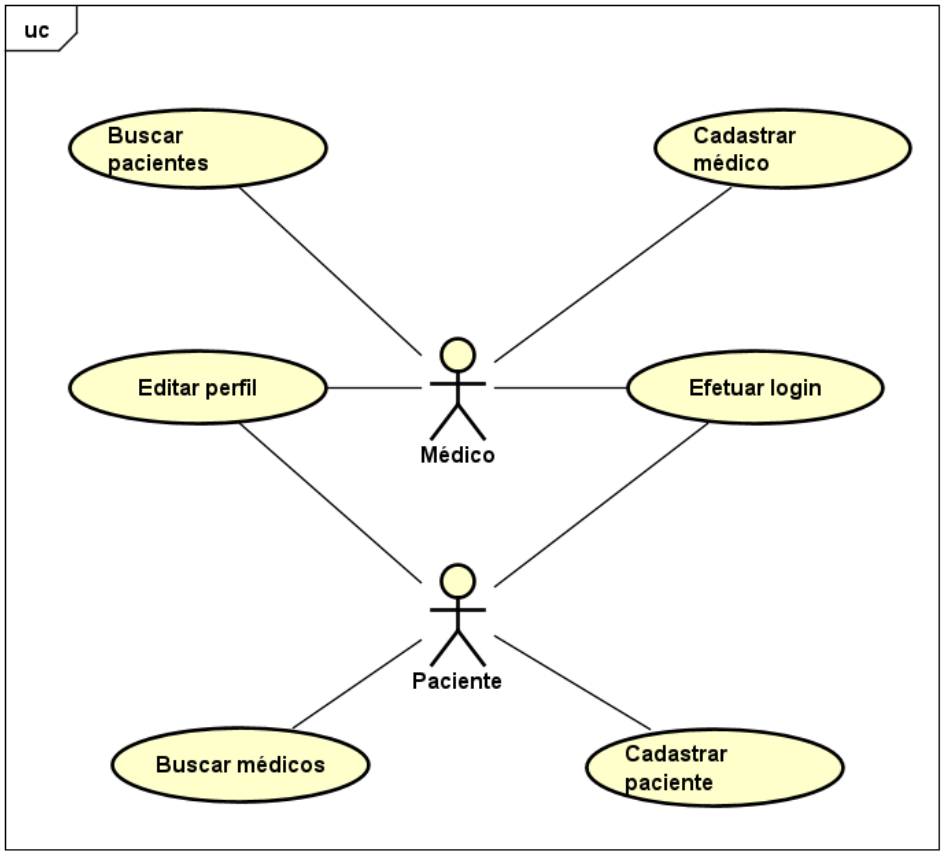
Fonte: O autor.

Figura 66 – Gerenciamento de histórico familiar



Fonte: O autor.

Figura 67 – Gerenciamento de contas



Fonte: O autor.

APÊNDICE B – Narrativas de caso de uso

B.1 UC002 – Cadastrar Paciente

Descrição: Este caso de uso detalha o processo de cadastro de um paciente no sistema, permitindo que ele crie uma conta e posteriormente acesse suas funcionalidades específicas.

Atores: Paciente

Pré-condições: O usuário não deve estar autenticado no sistema.

Pós-condições: Uma conta será criada para o paciente, permitindo que ele se autentique no sistema e acesse as funcionalidades destinadas aos médicos.

Fluxo Normal:

1. Na tela de *onboarding*, o usuário seleciona a opção "Criar conta".
2. O sistema redireciona o usuário para o formulário de cadastro de dados básicos, onde ele preenche os seguintes campos:
   - Foto de perfil;
   - Nome;
   - Sobrenome;
   - Gênero;
   - CPF;
   - Telefone.
3. Após preencher os dados, o usuário clica em "Próximo"e é redirecionado para a tela de seleção do tipo de usuário.
4. Na tela de seleção, o usuário escolhe a opção "Paciente"e o sistema avança para o próximo formulário de cadastro.
5. O sistema exibe o formulário específico para pacientes, onde o usuário deve preencher os seguintes campos:
   - Tipo sanguíneo;
   - Estado;

- Cidade;

6. Após preencher os dados, o usuário clica novamente em "Próximo"e o sistema avança para a etapa final.

7. O usuário define uma senha e a confirma no campo de confirmação de senha. Em seguida, clica no botão "Criar conta".

8. O sistema confirma o sucesso do cadastro exibindo uma mensagem de confirmação e redireciona o usuário para a página de login.

B.2 UC003 - Listar Histórico de Acessos

Descrição: O paciente pode visualizar um histórico dos médicos que acessaram seus dados. O histórico incluirá o nome do médico, data e hora de acesso, ordenado por data.

Atores: Paciente

Pré-condições: Paciente deve estar logado no sistema.

Pós-condições: O histórico de acessos é exibido na tela do paciente.

Fluxo normal:

1. O paciente acessa o menu "Permissões"no sistema.

2. Em permissões, o paciente acessa a aba "Histórico".

3. O sistema lista o histórico de acessos, exibindo o nome do médico, data e hora de acesso, ordenado por data.

4. O sistema encerra a operação.

B.3 UC004 - Listar solicitações de acesso

Descrição: O paciente pode visualizar e gerenciar as solicitações de acesso. Médicos apenas podem visualizar.

Atores: Paciente, Médico

Pré-condições: Usuário deve estar logado no sistema.

Pós-condições: As autorizações de acesso são gerenciadas conforme as ações do paciente.

Fluxo normal:

1. O usuário acessa o menu "Permissões"no sistema.
2. São listadas solicitações pendentes e aprovadas
3. Pacientes podem aprovar ou recusar solicitações pendentes, ou verificar quais foram aprovadas. Médicos podem somente acompanhar as solicitações.
4. O sistema encerra a operação.

B.4 UC005 - Solicitar permissão

Descrição: O médico pode solicitar acesso aos dados de um paciente.

Atores: Médico

Pré-condições: Médico deve estar logado no sistema.

Pós-condições: A solicitação de acesso é criada e exibida no menu "Permissões".

Fluxo normal:

1. O médico busca um paciente e o seleciona
2. Ao visualizar o paciente, o médico pressiona o botão "Solicitar acesso"
3. Um texto de aprovação pendente é exibido no lugar do botão.
4. O sistema encerra a operação.

B.5 UC006 - Cadastrar Consulta

Descrição: Permite ao médico cadastrar uma nova consulta no sistema, incluindo informações detalhadas sobre a consulta.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema e acessando o perfil de um paciente cujo acesso foi autorizado.

Pós-condições: Uma nova consulta é criada e registrada no sistema.

Fluxo normal:

1. O médico acessa o menu "Consultas"no sistema.
2. O médico seleciona o botão de adicionar na listagem de consultas.
3. O sistema exibe um formulário para preenchimento dos detalhes da consulta.

4. O médico preenche os campos obrigatórios: data, sintomas, queixa principal, e anamnese.
5. O médico pode anexar arquivos relevantes à consulta.
6. Após preencher todos os dados, o médico clica no botão "Salvar".
7. O sistema confirma o cadastro da consulta e retorna para a listagem.

B.6 UC007 - Editar Consulta

Descrição: Permite ao médico editar uma consulta existente para atualizar suas informações.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema e acessando o perfil de um paciente cujo acesso foi autorizado.

Pós-condições: As informações da consulta são atualizadas no sistema.

Fluxo normal:

1. O médico acessa o menu "Consultas"no sistema.
2. O médico seleciona a consulta que deseja editar.
3. O sistema exibe os detalhes da consulta em um formulário editável.
4. O médico realiza as alterações necessárias e pode adicionar ou remover o atestado.
5. O médico clica no botão "Salvar".
6. O sistema confirma a atualização e retorna para a listagem.

B.7 UC008 - Excluir Consulta

Descrição: Permite ao médico excluir uma consulta do sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema e acessando o perfil de um paciente cujo acesso foi autorizado.

Pós-condições: A consulta é removida do sistema.

Fluxo normal:

1. O médico acessa o menu "Consultas"no sistema.

2. O médico seleciona a consulta que deseja excluir.
3. O médico seleciona a opção "Excluir"
4. O sistema remove a consulta e retorna para a listagem.

B.8 UC009 - Visualizar Consulta

Descrição: Permite ao médico ou paciente visualizar os detalhes de uma consulta existente.

Atores: Médico, Paciente

Pré-condições: O usuário deve estar autenticado.

Pós-condições: Os detalhes da consulta são exibidos na tela.

Fluxo normal:

1. O médico acessa o menu "Consultas"a partir do perfil de um paciente que lhe concedeu acesso. O paciente acessa o menu "Consultas em sua tela inicial".
2. O usuário seleciona a consulta que deseja visualizar.
3. O sistema exibe os detalhes da consulta, incluindo dados preenchidos e anexos.

B.9 UC010 - Cadastrar Exame

Descrição: Permite ao médico cadastrar um novo exame no sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema e acessando o perfil de um paciente cujo acesso foi autorizado.

Pós-condições: Um novo exame é criado e registrado no sistema.

Fluxo normal:

1. O médico acessa o menu "Exames"no sistema.
2. O médico seleciona o botão de adicionar na listagem de exames.
3. O sistema exibe um formulário para preenchimento dos detalhes do exame.
4. O médico preenche os campos necessários.
5. O médico pode anexar arquivos relacionados ao exame.

6. Após preencher todos os dados, o médico clica no botão "Salvar".
7. O sistema confirma o cadastro do exame e retorna para a listagem.

B.10 UC011 - Editar Exame

Descrição: Permite ao médico editar um exame existente para atualizar suas informações.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema e acessando o perfil de um paciente cujo acesso foi autorizado.

Pós-condições: As informações do exame são atualizadas no sistema.

Fluxo normal:

1. O médico acessa o menu "Exames"no sistema.
2. O médico seleciona o exame que deseja editar.
3. O sistema exibe os detalhes do exame em um formulário editável.
4. O médico realiza as alterações necessárias e pode adicionar ou remover anexos.
5. O médico clica no botão "Salvar".
6. O sistema confirma a atualização e retorna para a listagem.

B.11 UC012 - Excluir Exame

Descrição: Permite ao médico excluir um exame do sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema e acessando o perfil de um paciente cujo acesso foi autorizado.

Pós-condições: O exame é removido do sistema.

Fluxo normal:

1. O médico acessa o menu "Exames"no sistema.
2. O médico seleciona o exame que deseja excluir.
3. O médico seleciona a opção "Excluir"
4. O sistema remove o exame e retorna para a listagem.

B.12 UC013 - Visualizar Exame

Descrição: Permite ao médico ou paciente visualizar os detalhes de um exame existente.

Atores: Médico, Paciente

Pré-condições: O usuário deve estar autenticado.

Pós-condições: Os detalhes do exame são exibidos na tela.

Fluxo normal:

1. O médico acessa o menu "Exames"a partir do perfil de um paciente que lhe concedeu acesso. O paciente acessa o menu "Exames"em sua tela inicial.
2. O usuário seleciona o exame que deseja visualizar.
3. O sistema exibe os detalhes, incluindo dados preenchidos e anexos.

B.13 UC014 - Cadastrar Alergia

Descrição: Permite ao médico cadastrar uma nova alergia para um paciente no sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema.

Pós-condições: Uma nova alergia é criada e registrada no sistema.

Fluxo normal:

1. O médico acessa o menu "Alergias"no sistema.
2. O médico seleciona a opção "Cadastrar nova alergia".
3. O sistema exibe um formulário para preenchimento dos detalhes da alergia.
4. O médico preenche os campos obrigatórios: nome da alergia, descrição.
5. O médico clica no botão "Salvar".
6. O sistema confirma o cadastro da alergia e exibe uma mensagem de sucesso.

B.14 UC015 - Editar Alergia

Descrição: Permite ao médico editar uma alergia existente para atualizar suas informações.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso à alergia que deseja editar.

Pós-condições: As informações da alergia são atualizadas no sistema.

Fluxo normal:

1. O médico acessa o menu "Alergias"no sistema.
2. O médico seleciona a alergia que deseja editar.
3. O sistema exibe os detalhes da alergia em um formulário editável.
4. O médico realiza as alterações necessárias.
5. O médico clica no botão "Atualizar".
6. O sistema confirma a atualização e exibe uma mensagem de sucesso.

B.15 UC016 - Excluir Alergia

Descrição: Permite ao médico excluir uma alergia do sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso à alergia que deseja excluir.

Pós-condições: A alergia é removida do sistema.

Fluxo normal:

1. O médico acessa o menu "Alergias"no sistema.
2. O médico seleciona a alergia que deseja excluir.
3. O sistema solicita uma confirmação para a exclusão.
4. O médico confirma a exclusão.
5. O sistema remove a alergia e exibe uma mensagem de sucesso.

B.16 UC017 - Visualizar Alergia

Descrição: Permite ao médico ou paciente visualizar os detalhes de uma alergia existente.

Atores: Médico, Paciente

Pré-condições: O usuário deve estar autenticado.

Pós-condições: Os detalhes da alergia são exibidos na tela.

Fluxo normal:

1. O médico acessa o menu "Alergias"a partir do perfil de um paciente que lhe concedeu acesso. O paciente acessa o menu "Alergias"em sua tela inicial.
2. O usuário seleciona a alergia que deseja visualizar.
3. O sistema exibe os detalhes, incluindo dados preenchidos.

B.17 UC018 - Cadastrar Prescrição

Descrição: Permite ao médico cadastrar uma nova prescrição para um paciente no sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema.

Pós-condições: Uma nova prescrição é criada e registrada no sistema.

Fluxo normal:

1. O médico acessa o menu "Prescrições"no sistema.
2. O médico seleciona a opção "Cadastrar nova prescrição".
3. O sistema exibe um formulário para preenchimento dos detalhes da prescrição.
4. O médico preenche os campos obrigatórios: nome do medicamento, dosagem, frequência e duração.
5. O médico clica no botão "Salvar".
6. O sistema confirma o cadastro da prescrição e exibe uma mensagem de sucesso.

B.18 UC019 - Editar Prescrição

Descrição: Permite ao médico editar uma prescrição existente para atualizar suas informações.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso à prescrição que

deseja editar.

Pós-condições: As informações da prescrição são atualizadas no sistema.

Fluxo normal:

1. O médico acessa o menu "Prescrições"no sistema.
2. O médico seleciona a prescrição que deseja editar.
3. O sistema exibe os detalhes da prescrição em um formulário editável.
4. O médico realiza as alterações necessárias.
5. O médico clica no botão "Atualizar".
6. O sistema confirma a atualização e exibe uma mensagem de sucesso.

B.19 UC020 - Excluir Prescrição

Descrição: Permite ao médico excluir uma prescrição do sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso à prescrição que deseja excluir.

Pós-condições: A prescrição é removida do sistema.

Fluxo normal:

1. O médico acessa o menu "Prescrições"no sistema.
2. O médico seleciona a prescrição que deseja excluir.
3. O sistema solicita uma confirmação para a exclusão.
4. O médico confirma a exclusão.
5. O sistema remove a prescrição e exibe uma mensagem de sucesso.

B.20 UC021 - Visualizar Prescrição

Descrição: Permite ao médico ou paciente visualizar os detalhes de uma prescrição existente.

Atores: Médico, Paciente

Pré-condições: O usuário deve estar autenticado.

Pós-condições: Os detalhes da prescrição são exibidos na tela.

Fluxo normal:

1. O médico acessa o menu "Prescrições"a partir do perfil de um paciente que lhe concedeu acesso. O paciente acessa o menu "Prescrições"em sua tela inicial.
2. O usuário seleciona a prescrição que deseja visualizar.
3. O sistema exibe os detalhes, incluindo dados preenchidos.

B.21 UC022 - Cadastrar Diagnóstico

Descrição: Permite ao médico cadastrar um novo diagnóstico para um paciente no sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema.

Pós-condições: Um novo diagnóstico é criado e registrado no sistema.

Fluxo normal:

1. O médico acessa o menu "Diagnósticos"no sistema.
2. O médico seleciona a opção "Cadastrar novo diagnóstico".
3. O sistema exibe um formulário para preenchimento dos detalhes do diagnóstico.
4. O médico preenche os campos obrigatórios: nome do diagnóstico, descrição.
5. O médico clica no botão "Salvar".
6. O sistema confirma o cadastro do diagnóstico e exibe uma mensagem de sucesso.

B.22 UC023 - Editar Diagnóstico

Descrição: Permite ao médico editar um diagnóstico existente para atualizar suas informações.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso ao diagnóstico que deseja editar.

Pós-condições: As informações do diagnóstico são atualizadas no sistema.

Fluxo normal:

1. O médico acessa o menu "Diagnósticos"no sistema.
2. O médico seleciona o diagnóstico que deseja editar.
3. O sistema exibe os detalhes do diagnóstico em um formulário editável.
4. O médico realiza as alterações necessárias.
5. O médico clica no botão "Atualizar".
6. O sistema confirma a atualização e exibe uma mensagem de sucesso.

B.23 UC024 - Excluir Diagnóstico

Descrição: Permite ao médico excluir um diagnóstico do sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso ao diagnóstico que deseja excluir.

Pós-condições: O diagnóstico é removido do sistema.

Fluxo normal:

1. O médico acessa o menu "Diagnósticos"no sistema.
2. O médico seleciona o diagnóstico que deseja excluir.
3. O sistema solicita uma confirmação para a exclusão.
4. O médico confirma a exclusão.
5. O sistema remove o diagnóstico e exibe uma mensagem de sucesso.

B.24 UC025 - Visualizar Diagnóstico

Descrição: Permite ao médico ou paciente visualizar os detalhes de um diagnóstico existente.

Atores: Médico, Paciente

Pré-condições: O usuário deve estar autenticado.

Pós-condições: Os detalhes do diagnóstico são exibidos na tela.

Fluxo normal:

1. O médico acessa o menu "Diagnósticos"a partir do perfil de um paciente que lhe concedeu acesso. O paciente acessa o menu "Diagnósticos"em sua tela inicial.

2. O usuário seleciona o diagnóstico que deseja visualizar.
3. O sistema exibe os detalhes, incluindo dados preenchidos.

B.25 UC026 - Cadastrar Cirurgia

Descrição: Permite ao médico cadastrar uma nova cirurgia para um paciente no sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema.

Pós-condições: Uma nova cirurgia é criada e registrada no sistema.

Fluxo normal:

1. O médico acessa o menu "Cirurgias"no sistema.
2. O médico seleciona a opção "Cadastrar nova cirurgia".
3. O sistema exibe um formulário para preenchimento dos detalhes da cirurgia.
4. O médico preenche os campos obrigatórios: nome da cirurgia, data, hospital.
5. O médico pode adicionar detalhes adicionais, como descrição da cirurgia e data da alta.
6. O médico clica no botão "Salvar".
7. O sistema confirma o cadastro da cirurgia e exibe uma mensagem de sucesso.

B.26 UC027 - Editar Cirurgia

Descrição: Permite ao médico editar uma cirurgia existente para atualizar suas informações.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso à cirurgia que deseja editar.

Pós-condições: As informações da cirurgia são atualizadas no sistema.

Fluxo normal:

1. O médico acessa o menu "Cirurgias"no sistema.
2. O médico seleciona a cirurgia que deseja editar.

3. O sistema exibe os detalhes da cirurgia em um formulário editável.
4. O médico realiza as alterações necessárias.
5. O médico clica no botão "Atualizar".
6. O sistema confirma a atualização e exibe uma mensagem de sucesso.

B.27 UC028 - Excluir Cirurgia

Descrição: Permite ao médico excluir uma cirurgia do sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso à cirurgia que deseja excluir.

Pós-condições: A cirurgia é removida do sistema.

Fluxo normal:

1. O médico acessa o menu "Cirurgias"no sistema.
2. O médico seleciona a cirurgia que deseja excluir.
3. O sistema solicita uma confirmação para a exclusão.
4. O médico confirma a exclusão.
5. O sistema remove a cirurgia e exibe uma mensagem de sucesso.

B.28 UC029 - Visualizar Cirurgia

Descrição: Permite ao médico ou paciente visualizar os detalhes de uma cirurgia existente.

Atores: Médico, Paciente

Pré-condições: O usuário deve estar autenticado.

Pós-condições: Os detalhes da cirurgia são exibidos na tela.

Fluxo normal:

1. O médico acessa o menu "Cirurgias"a partir do perfil de um paciente que lhe concedeu acesso. O paciente acessa o menu "Cirurgias"em sua tela inicial.
2. O usuário seleciona a cirurgia que deseja visualizar.
3. O sistema exibe os detalhes, incluindo dados preenchidos.

B.29 UC030 - Cadastrar Histórico Familiar

Descrição: Permite ao médico cadastrar informações sobre o histórico familiar de um paciente.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema.

Pós-condições: O histórico familiar é registrado no sistema.

Fluxo normal:

1. O médico acessa o menu "Histórico Familiar"no sistema.
2. O médico seleciona a opção "Cadastrar novo histórico familiar".
3. O sistema exibe um formulário para preenchimento dos detalhes do histórico familiar.
4. O médico preenche os campos obrigatórios: parentesco, doenças hereditárias.
5. O médico clica no botão "Salvar".
6. O sistema confirma o cadastro do histórico familiar e exibe uma mensagem de sucesso.

B.30 UC031 - Editar Histórico Familiar

Descrição: Permite ao médico editar informações do histórico familiar de um paciente.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso ao histórico familiar que deseja editar.

Pós-condições: As informações do histórico familiar são atualizadas no sistema.

Fluxo normal:

1. O médico acessa o menu "Histórico Familiar"no sistema.
2. O médico seleciona o histórico familiar que deseja editar.
3. O sistema exibe os detalhes do histórico familiar em um formulário editável.
4. O médico realiza as alterações necessárias.
5. O médico clica no botão "Atualizar".
6. O sistema confirma a atualização e exibe uma mensagem de sucesso.

B.31 UC032 - Excluir Histórico Familiar

Descrição: Permite ao médico excluir informações do histórico familiar de um paciente.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso ao histórico familiar que deseja excluir.

Pós-condições: O histórico familiar é removido do sistema.

Fluxo normal:

1. O médico acessa o menu "Histórico Familiar"no sistema.
2. O médico seleciona o histórico familiar que deseja excluir.
3. O sistema solicita uma confirmação para a exclusão.
4. O médico confirma a exclusão.
5. O sistema remove o histórico familiar e exibe uma mensagem de sucesso.

B.32 UC033 - Visualizar Histórico Familiar

Descrição: Permite ao médico ou paciente visualizar os detalhes de um histórico familiar existente.

Atores: Médico, Paciente

Pré-condições: O usuário deve estar autenticado.

Pós-condições: Os detalhes do histórico familiar são exibidos na tela.

Fluxo normal:

1. O médico acessa o menu "Histórico familiar"a partir do perfil de um paciente que lhe concedeu acesso. O paciente acessa o menu "Histórico familiar"em sua tela inicial.
2. O usuário seleciona o histórico familiar que deseja visualizar.
3. O sistema exibe os detalhes, incluindo dados preenchidos.

B.33 UC034 - Cadastrar Dieta

Descrição: Permite ao médico cadastrar uma nova dieta para um paciente no sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema.

Pós-condições: Uma nova dieta é criada e registrada no sistema.

Fluxo normal:

1. O médico acessa o menu "Dietas"no sistema.
2. O médico seleciona a opção "Cadastrar nova dieta".
3. O sistema exibe um formulário para preenchimento dos detalhes da dieta.
4. O médico preenche os campos obrigatórios: nome da dieta, descrição, duração.
5. O médico clica no botão "Salvar".
6. O sistema confirma o cadastro da dieta e exibe uma mensagem de sucesso.

B.34 UC035 - Editar Dieta

Descrição: Permite ao médico editar uma dieta existente para atualizar suas informações.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso à dieta que deseja editar.

Pós-condições: As informações da dieta são atualizadas no sistema.

Fluxo normal:

1. O médico acessa o menu "Dietas"no sistema.
2. O médico seleciona a dieta que deseja editar.
3. O sistema exibe os detalhes da dieta em um formulário editável.
4. O médico realiza as alterações necessárias.
5. O médico clica no botão "Atualizar".
6. O sistema confirma a atualização e exibe uma mensagem de sucesso.

B.35 UC036 - Excluir Dieta

Descrição: Permite ao médico excluir uma dieta do sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso à dieta que deseja excluir.

Pós-condições: A dieta é removida do sistema.

Fluxo normal:

1. O médico acessa o menu "Dietas"no sistema.
2. O médico seleciona a dieta que deseja excluir.
3. O sistema solicita uma confirmação para a exclusão.
4. O médico confirma a exclusão.
5. O sistema remove a dieta e exibe uma mensagem de sucesso.

B.36 UC037 - Visualizar Dieta

Descrição: Permite ao médico ou paciente visualizar os detalhes de uma dieta existente.

Atores: Médico, Paciente

Pré-condições: O usuário deve estar autenticado.

Pós-condições: Os detalhes da dieta são exibidos na tela.

Fluxo normal:

1. O médico acessa o menu "Dietas"a partir do perfil de um paciente que lhe concedeu acesso. O paciente acessa o menu "Dietas"em sua tela inicial.
2. O usuário seleciona a dieta que deseja visualizar.
3. O sistema exibe os detalhes, incluindo dados preenchidos.

B.37 UC038 - Cadastrar Biodado

Descrição: Permite ao médico cadastrar novos biodados para um paciente no sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado no sistema.

Pós-condições: Novos biodados são criados e registrados no sistema.

Fluxo normal:

1. O médico acessa o menu "Biodados"no sistema.
2. O médico seleciona a opção "Cadastrar novo biodado".
3. O sistema exibe um formulário para preenchimento dos detalhes dos biodados.
4. O médico preenche os campos obrigatórios: pressão arterial, glicemia, frequência cardíaca, colesterol, triglicerídeos, creatina.
5. O médico clica no botão "Salvar".
6. O sistema confirma o cadastro dos biodados e exibe uma mensagem de sucesso.

B.38 UC039 - Editar Biodado

Descrição: Permite ao médico editar um biodado existente para atualizar suas informações.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso ao biodado que deseja editar.

Pós-condições: As informações do biodado são atualizadas no sistema.

Fluxo normal:

1. O médico acessa o menu "Biodados"no sistema.
2. O médico seleciona o biodado que deseja editar.
3. O sistema exibe os detalhes do biodado em um formulário editável.
4. O médico realiza as alterações necessárias.
5. O médico clica no botão "Atualizar".
6. O sistema confirma a atualização e exibe uma mensagem de sucesso.

B.39 UC040 - Excluir Biodado

Descrição: Permite ao médico excluir um biodado do sistema.

Atores: Médico

Pré-condições: O médico deve estar autenticado e ter acesso ao biodado que

deseja excluir.

Pós-condições: O biodado é removido do sistema.

Fluxo normal:

1. O médico acessa o menu "Biodados"no sistema.
2. O médico seleciona o biodado que deseja excluir.
3. O sistema solicita uma confirmação para a exclusão.
4. O médico confirma a exclusão.
5. O sistema remove o biodado e exibe uma mensagem de sucesso.

B.40 UC041 - Visualizar Biodado

Descrição: Permite ao médico ou paciente visualizar os detalhes de um biodado existente.

Atores: Médico, Paciente

Pré-condições: O usuário deve estar autenticado.

Pós-condições: Os detalhes do biodado são exibidos na tela.

Fluxo normal:

1. O médico acessa o menu "Biodados"a partir do perfil de um paciente que lhe concedeu acesso. O paciente acessa o menu "Biodados"em sua tela inicial.
2. O usuário seleciona o biodado que deseja visualizar.
3. O sistema exibe os detalhes, incluindo dados preenchidos.

APÊNDICE C – Descrição das tabelas do modelo conceitual do banco de dados

Tabela 13 – Entidade Patient - Modelo conceitual

Coluna	**Tipo**	**Descrição**
userId	Texto	Chave estrangeira para a tabela Users
city	Texto	Cidade do paciente
state	Texto	Estado do paciente
bloodType	Enum	Enum com os valores dos tipos sanguíneos
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 14 – Entidade Doctor - Modelo conceitual

Coluna	**Tipo**	**Descrição**
userId	Texto	Chave estrangeira para a tabela Users
crm	Texto	Número de CRM do médico
specialty	Texto	Especialidade médica
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 15 – Entidade Prescription - Modelo conceitual

Coluna	**Tipo**	**Descrição**
id	Texto	Chave primária da tabela
date	Data	Data da prescrição
notes	Texto	Notas adicionais (opcional)
patientId	Texto	Chave estrangeira para a tabela Patients
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 16 – Entidade Exam - Modelo conceitual

Coluna	**Tipo**	**Descrição**
id	Texto	Chave primária da tabela
name	Texto	Nome do exame
date	Data	Data do exame
completionDate	Data	Data de conclusão (opcional)
result	Texto	Resultado do exame (opcional)
patientId	Texto	Chave estrangeira para a tabela Patients
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 17 – Entidade Surgery - Modelo conceitual

Coluna	**Tipo**	**Descrição**
id	Texto	Chave primária da tabela
name	Texto	Nome da cirurgia
date	Data	Data da cirurgia
hospital	Texto	Nome do hospital
dischargeDate	Data	Data de alta (opcional)
description	Texto	Descrição detalhada (opcional)
patientId	Texto	Chave estrangeira para a tabela Patients
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 18 – Entidade Appointment - Modelo conceitual

Coluna	**Tipo**	**Descrição**
id	Texto	Chave primária da tabela
patientId	Texto	Chave estrangeira da tabela Patient
doctorId	Texto	Chave estrangeira da tabela Doctor
date	Data	Data da consulta
symptoms	Texto	Descrição dos sintomas relatados
chiefComplaint	Texto	Queixa principal do paciente
anamnesis	Texto	Anamnese realizada na consulta
sickNoteName	Texto	Nome do atestado médico emitido (opcional)
sickNotePath	Texto	Caminho do atestado médico emitido (opcional)
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 19 – Entidade Allergy - Modelo conceitual

Coluna	**Tipo**	**Descrição**
id	Texto	Chave primária da tabela
patientId	Texto	Chave estrangeira da tabela Patient
name	Texto	Nome da alergia
description	Texto	Descrição detalhada da alergia
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 20 – Entidade MedicationPrescription - Modelo conceitual

Coluna	**Tipo**	**Descrição**
id	Texto	Chave primária da tabela
prescriptionId	Texto	Chave estrangeira da tabela Prescription
name	Texto	Nome do medicamento prescrito
dosage	Texto	Dosagem do medicamento
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 21 – Entidade AccessControl - Modelo conceitual

Coluna	**Tipo**	**Descrição**
id	Texto	Chave primária da tabela
patientId	Texto	Chave estrangeira da tabela Patient
doctorId	Texto	Chave estrangeira da tabela Doctor
status	Numérico	Enum AccessStatusEnum
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 22 – Entidade AccessHistory - Modelo conceitual

Coluna	**Tipo**	**Descrição**
id	Texto	Chave primária da tabela
patientId	Texto	Chave estrangeira da tabela Patient
doctorId	Texto	Chave estrangeira da tabela Doctor
date	Data	Data de acesso
createdAt	Data	Data da criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 23 – Entidade Diets - Modelo conceitual

Coluna	Tipo	Descrição
id	Texto	Chave primária da tabela
breakfast	Texto	Descrição do café da manhã
morningSnack	Texto	Descrição do lanche da manhã
lunch	Texto	Descrição do almoço
afternoonSnack	Texto	Descrição do lanche da tarde
dinner	Texto	Descrição do jantar
patientId	Texto	Chave estrangeira da tabela Patient
doctorId	Texto	Chave estrangeira da tabela Doctor
createdAt	Data	Data de criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 24 – Entidade FamilyHistories - Modelo conceitual

Coluna	Tipo	Descrição
id	Texto	Chave primária da tabela
kinship	Texto	Grau de parentesco
name	Texto	Nome do familiar
description	Texto	Descrição do histórico familiar
patientId	Texto	Chave estrangeira da tabela Patient
createdAt	Data	Data de criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

Tabela 25 – Entidade Biodata - Modelo conceitual

Coluna	Tipo	Descrição
id	Texto	Chave primária da tabela
date	Data	Data do registro de biodados
systolicPressure	Numérico	Pressão arterial sistólica
diastolicPressure	Numérico	Pressão arterial diastólica
glycemia	Numérico	Nível de glicemia
heartRate	Numérico	Frequência cardíaca
cholesterol	Numérico	Nível de colesterol
triglyceride	Numérico	Nível de triglicerídeos
creatinine	Numérico	Nível de creatinina
patientId	Texto	Chave estrangeira da tabela Patient
createdAt	Data	Data de criação
updatedAt	Data	Data da última atualização

Fonte: O autor.

APÊNDICE D – Descrição das tabelas do modelo físico do banco de dados

Tabela 26 – Tabela Surgeries - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
patientId	VARCHAR(36)	Chave estrangeira da tabela Patient
doctorId	VARCHAR(36)	Chave estrangeira da tabela Doctor
date	DATETIME(6)	Data da cirurgia
procedureName	VARCHAR(255)	Nome do procedimento cirúrgico
hospital	VARCHAR(255)	Nome do hospital
dischargeDate	DATETIME(6)	Data da alta hospitalar (opcional)
description	TEXT	Descrição detalhada do procedimento cirúrgico (opcional)
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 27 – Tabela Exams - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
patientId	VARCHAR(36)	Chave estrangeira da tabela Patient
doctorId	VARCHAR(36)	Chave estrangeira da tabela Doctor
name	VARCHAR(255)	Nome do exame
date	DATETIME(6)	Data da realização do exame
result	TEXT	Resultado do exame (opcional)
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 28 – Tabela Prescriptions - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
patientId	VARCHAR(36)	Chave estrangeira da tabela Patient
doctorId	VARCHAR(36)	Chave estrangeira da tabela Doctor
prescriptionDate	DATETIME(6)	Data da prescrição
notes	TEXT	Notas adicionais (opcional)
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 29 – Tabela Patients - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
name	VARCHAR(255)	Nome completo do paciente
birthDate	DATETIME(6)	Data de nascimento
address	TEXT	Endereço completo
phoneNumber	VARCHAR(20)	Número de telefone
email	VARCHAR(255)	E-mail do paciente
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 30 – Tabela Doctors - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
name	VARCHAR(255)	Nome completo do médico
crm	VARCHAR(20)	Registro no Conselho Regional de Medicina (CRM)
specialty	VARCHAR(255)	Especialidade médica
phoneNumber	VARCHAR(20)	Número de telefone
email	VARCHAR(255)	E-mail do médico
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 31 – Tabela Appointments - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
patientId	VARCHAR(36)	Chave estrangeira da tabela Patient
doctorId	VARCHAR(36)	Chave estrangeira da tabela Doctor
date	DATETIME(6)	Data da consulta
symptoms	TEXT	Descrição dos sintomas relatados
chiefComplaint	TEXT	Queixa principal do paciente
anamnesis	TEXT	Anamnese realizada na consulta
sickNoteName	VARCHAR(255)	Nome do atestado médico emitido (opcional)
sickNotePath	TEXT	Caminho do atestado médico emitido (opcional)
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 32 – Tabela Allergies - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
patientId	VARCHAR(36)	Chave estrangeira da tabela Patient
name	VARCHAR(255)	Nome da alergia
description	TEXT	Descrição detalhada da alergia
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 33 – Tabela MedicationPrescription - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
prescriptionId	VARCHAR(36)	Chave estrangeira da tabela Prescription
name	VARCHAR(255)	Nome do medicamento prescrito
dosage	TEXT	Dosagem do medicamento
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 34 – Tabela AccessControl - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
patientId	VARCHAR(36)	Chave estrangeira da tabela Patient
doctorId	VARCHAR(36)	Chave estrangeira da tabela Doctor
status	ENUM	Enum AccessControlStatus
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 35 – Tabela AccessHistories - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
patientId	VARCHAR(36)	Chave estrangeira da tabela Patient
doctorId	VARCHAR(36)	Chave estrangeira da tabela Doctor
accessDate	DATETIME(6)	Data e hora do acesso
accessType	ENUM	Tipo de acesso (Visualização, Edição, Exclusão)
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 36 – Tabela Diets - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
breakfast	VARCHAR(255)	Descrição do café da manhã
morningSnack	VARCHAR(255)	Descrição do lanche da manhã
lunch	VARCHAR(255)	Descrição do almoço
afternoonSnack	VARCHAR(255)	Descrição do lanche da tarde
dinner	VARCHAR(255)	Descrição do jantar
patientId	VARCHAR(36)	Chave estrangeira da tabela Patient
doctorId	VARCHAR(36)	Chave estrangeira da tabela Doctor
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 37 – Tabela FamilyHistories - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
kinship	VARCHAR(255)	Grau de parentesco
name	VARCHAR(255)	Nome do familiar
description	VARCHAR(255)	Descrição do histórico familiar
patientId	VARCHAR(36)	Chave estrangeira da tabela Patient
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

Tabela 38 – Tabela Biodata - Modelo físico

Coluna	**Tipo**	**Descrição**
id	VARCHAR(36)	Chave primária da tabela
date	DATETIME	Data do registro de biodados
systolicPressure	FLOAT	Pressão arterial sistólica
diastolicPressure	FLOAT	Pressão arterial diastólica
glycemia	FLOAT	Nível de glicemia
heartRate	FLOAT	Frequência cardíaca
cholesterol	FLOAT	Nível de colesterol
triglyceride	FLOAT	Nível de triglicerídeos
creatinine	FLOAT	Nível de creatinina
patientId	VARCHAR(36)	Chave estrangeira da tabela Patient
createdAt	DATETIME(6)	Data de criação
updatedAt	DATETIME(6)	Data da última atualização

Fonte: O autor.

APÊNDICE E – Questionários respondidos

Tabela 39 – Questionário respondido - Usuário 01

Questão	1	2	3	4	5	Contribuição
01 Eu acho que gostaria de usar esse sistema frequentemente.		X				1
02 Eu achei esse sistema desnecessariamente complexo.			X			2
03 Eu achei esse sistema fácil de usar.				X		3
04 Eu achei que precisaria de ajuda de uma pessoa técnica para ser capaz de usar esse sistema.	X					4
05 Eu achei que as várias funções desse sistema foram bem integradas.				X		3
06 Eu acho que o sistema apresenta muita inconsistência.	X					4
07 Eu imagino que a maioria das pessoas pode aprender a usar esse sistema rapidamente.					X	4
08 Eu achei esse sistema muito pesado para usar.		X				3
09 Eu me senti muito seguro usando o sistema.					X	4
10 Eu precisei aprender muitas coisas antes que pudesse utilizar esse sistema.	X					4

Fonte: O autor.

A Pontuação final do SUS para o usuário 1 foi 80.

Tabela 40 – Questionário respondido - Usuário 02

Questão	1	2	3	4	5	Contribuição
01 Eu acho que gostaria de usar esse sistema frequentemente.		X				1
02 Eu achei esse sistema desnecessariamente complexo.		X				3
03 Eu achei esse sistema fácil de usar.				X		3
04 Eu achei que precisaria de ajuda de uma pessoa técnica para ser capaz de usar esse sistema.	X					4
05 Eu achei que as várias funções desse sistema foram bem integradas.					X	4
06 Eu acho que o sistema apresenta muita inconsistência.		X				3
07 Eu imagino que a maioria das pessoas pode aprender a usar esse sistema rapidamente.				X		3
08 Eu achei esse sistema muito pesado para usar.		X				3
09 Eu me senti muito seguro usando o sistema.				X		3
10 Eu precisei aprender muitas coisas antes que pudesse utilizar esse sistema.	X					4

Fonte: O autor.

A Pontuação final do SUS para o usuário 2 foi 77,5.

Tabela 41 – Questionário respondido - Usuário 03

Questão	1	2	3	4	5	Contribuição
01 Eu acho que gostaria de usar esse sistema frequentemente.			X			2
02 Eu achei esse sistema desnecessariamente complexo.	X					4
03 Eu achei esse sistema fácil de usar.					X	4
04 Eu achei que precisaria de ajuda de uma pessoa técnica para ser capaz de usar esse sistema.	X					4
05 Eu achei que as várias funções desse sistema foram bem integradas.				X		3
06 Eu acho que o sistema apresenta muita inconsistência.		X				3
07 Eu imagino que a maioria das pessoas pode aprender a usar esse sistema rapidamente.				X		3
08 Eu achei esse sistema muito pesado para usar.	X					4
09 Eu me senti muito seguro usando o sistema.					X	4
10 Eu precisei aprender muitas coisas antes que pudesse utilizar esse sistema.	X					4

Fonte: O autor.

A Pontuação final do SUS para o usuário 3 foi 87,5.

Tabela 42 – Questionário respondido - Usuário 04

Questão	1	2	3	4	5	Contribuição
01 Eu acho que gostaria de usar esse sistema frequentemente.		X				1
02 Eu achei esse sistema desnecessariamente complexo.		X				3
03 Eu achei esse sistema fácil de usar.				X		3
04 Eu achei que precisaria de ajuda de uma pessoa técnica para ser capaz de usar esse sistema.	X					4
05 Eu achei que as várias funções desse sistema foram bem integradas.				X		3
06 Eu acho que o sistema apresenta muita inconsistência.	X					4
07 Eu imagino que a maioria das pessoas pode aprender a usar esse sistema rapidamente.					X	4
08 Eu achei esse sistema muito pesado para usar.		X				3
09 Eu me senti muito seguro usando o sistema.					X	4
10 Eu precisei aprender muitas coisas antes que pudesse utilizar esse sistema.	X					4

Fonte: O autor.

A Pontuação final do SUS para o usuário 4 foi 82,5.

Tabela 43 – Questionário respondido - Usuário 05

Questão	1	2	3	4	5	Contribuição
01 Eu acho que gostaria de usar esse sistema frequentemente.				X		3
02 Eu achei esse sistema desnecessariamente complexo.	X					4
03 Eu achei esse sistema fácil de usar.				X		3
04 Eu achei que precisaria de ajuda de uma pessoa técnica para ser capaz de usar esse sistema.	X					4
05 Eu achei que as várias funções desse sistema foram bem integradas.			X			2
06 Eu acho que o sistema apresenta muita inconsistência.	X					4
07 Eu imagino que a maioria das pessoas pode aprender a usar esse sistema rapidamente.					X	4
08 Eu achei esse sistema muito pesado para usar.	X					4
09 Eu me senti muito seguro usando o sistema.				X		3
10 Eu precisei aprender muitas coisas antes que pudesse utilizar esse sistema.	X					4

Fonte: O autor.

A Pontuação final do SUS para o usuário 5 foi 87,5.