

**INSTITUTO FEDERAL GOIANO – CAMPUS CERES  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**IGOR GABRIEL SILVA BATISTA**

**Aplicação Web para gestão de prontuários eletrônicos de  
atendimento aos pacientes da Equoterapia do IF Goiano –  
Campus Ceres.**

IGOR GABRIEL SILVA BATISTA

**Aplicação Web para gestão de prontuários eletrônicos de atendimento aos pacientes da Equoterapia do IF Goiano – Campus Ceres.**

Trabalho de curso apresentado ao curso de Sistemas de Informação do Instituto Federal Goiano – Campus Ceres, como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Rafael Divino Ferreira Feitosa.

Sistema desenvolvido pelo ICMC/USP  
Dados Internacionais de Catalogação na Publicação (CIP)  
**Sistema Integrado de Bibliotecas - Instituto Federal Goiano**

BB333a Batista, igor Gabriel Silva  
Aplicação Web para gestão de prontuários eletrônicos de atendimento aos pacientes da Equoterapia do IF Goiano - Campus Ceres. / igor Gabriel Silva Batista; orientador Rafael Divino Ferreira Feitosa. -- Ceres, 2023.  
70 p.

TCC (Graduação em Bacharelado em Sistemas de Informação) -- Instituto Federal Goiano, Campus Ceres, 2023.

1. Equoterapia. 2. gerenciamento de informações.  
3. plataforma web. I. Feitosa, Rafael Divino Ferreira , orient. II. Título.

# TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano a disponibilizar gratuitamente o documento em formato digital no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

## IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

- |  |   |
|--|---|
| <input type="checkbox"/> Tese (doutorado)            | <input type="checkbox"/> Artigo científico              |
| <input type="checkbox"/> Dissertação (mestrado)      | <input type="checkbox"/> Capítulo de livro              |
| <input type="checkbox"/> Monografia (especialização) | <input type="checkbox"/> Livro                          |
| <input checked="" type="checkbox"/> TCC (graduação)  | <input type="checkbox"/> Trabalho apresentado em evento |

Produto técnico e educacional - Tipo: Aplicativos de aquisição e análise de dados

Nome completo do autor:

Igor Gabriel Silva Batista

Matrícula:

2019103202030083

Título do trabalho:

Aplicação Web para gestão de prontuários eletrônicos de atendimento aos pacientes da Equoterapia do IF Goiano – Campus Ceres.

## RESTRIÇÕES DE ACESSO AO DOCUMENTO

Documento confidencial:  Não  Sim, justifique:

Informe a data que poderá ser disponibilizado no RIIF Goiano: 01 / 08 / 2023

O documento está sujeito a registro de patente?  Sim  Não

O documento pode vir a ser publicado como livro?  Sim  Não

## DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O(a) referido(a) autor(a) declara:

- Que o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- Que obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autoria, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- Que cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Ceres

Local

01 / 08 / 2023

Data

Assinatura do autor e/ou detentor dos direitos autorais



Documento assinado digitalmente  
IGOR GABRIEL SILVA BATISTA  
Data: 02/08/2023 17:41:15-0300  
Verifique em <https://validar.ifi.gov.br>

Ciente e de acordo:

Assinatura do(a) orientador(a)



Documento assinado digitalmente  
RAFAEL DIVINO FERREIRA FEITOSA  
Data: 02/08/2023 12:40:17-0300  
Verifique em <https://validar.ifi.gov.br>



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

### ATA DE DEFESA DE TRABALHO DE CURSO

Ao(s) 15 dia(s) do mês de junho do ano de dois mil e vinte e três, realizou-se a defesa de Trabalho de Curso do(a) acadêmico(a) Igor Gabriel Silva Batista, do Curso de Bacharelado em Sistemas de Informação, matrícula 2019103202030083, cujo título é "Aplicação Web para gestão de prontuários eletrônicos de atendimento aos pacientes da Equoterapia do IF Goiano - Campus Ceres". A defesa iniciou-se às 19 horas e 35 minutos, finalizando-se às 21 horas e 00 minutos. A banca examinadora considerou o trabalho **APROVADO** com média 8,8 no trabalho escrito, média 9,5 no trabalho oral, apresentando assim média aritmética final de **9,2** pontos, estando o(a) estudante **APTO** para fins de conclusão do Trabalho de Curso.

Após atender às considerações da banca e respeitando o prazo disposto em calendário acadêmico, o(a) estudante deverá fazer a submissão da versão corrigida em formato digital (.pdf) no Repositório Institucional do IF Goiano - RIIF, acompanhado do Termo Ciência e Autorização Eletrônico (TCAE), devidamente assinado pelo autor e orientador.

Os integrantes da banca examinadora assinam a presente.

*(Assinado Eletronicamente)*

Prof. Dr. Rafael Divino Ferreira Feitosa

*(Assinado Eletronicamente)*

Prof. Me. Rangel Rigo

*(Assinado Eletronicamente)*

Profª. Me. Thalia Santos de Santana

Documento assinado eletronicamente por:

- Rafael Divino Ferreira Feitosa, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 15/06/2023 21:10:40.
- Rangel Rigo, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 15/06/2023 21:10:57.
- Thalia Santos de Santana, Thalia Santos de Santana - Conselho Superior/IF Goiano - Instituto Federal Goiano (1), em 15/06/2023 21:10:58.

Este documento foi emitido pelo SUAP em 15/06/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 504213  
Código de Autenticação: 79e269dfbf



## **Agradecimentos**

Agradeço a todos os colegas, amigos e professores que contribuíram positivamente durante minha trajetória, em especial aos meus amigos, Gabriel Baldoni e Iasmim Coutinho por todo auxílio fornecido, ao meu orientador, Rafael Feitosa por todo o comprometimento, paciência e apoio, e por fim, agradeço imensamente a Maria Alice Nunes Silva que esteve ao meu lado durante todo o período da graduação me apoiando na realização das minhas metas acadêmicas, pessoais e profissionais.

*“Somos o que pensamos. Tudo o que somos surge com nossos pensamentos. Com nossos pensamentos, fazemos o nosso mundo.”*

*Buda*

## RESUMO

A equoterapia é um procedimento terapêutico que integra e unifica diversos conhecimentos e estratégias das áreas de saúde e educação aplicadas à prática da equitação como procedimento terapêutico focando o desenvolvimento corporal e psicológico do praticante. Portanto, amplamente utilizada em casos de paralisia cerebral, síndrome de Down, distúrbios de comportamentos, autismo e diversos outros casos de comprometimento motor, sociais e emocionais. Reconhecendo a importância e a necessidade social por um atendimento de hipoterapia gratuito e de qualidade, o Instituto Federal Goiano – Campus Ceres mantém um centro especializado em equoterapia em suas instalações que oferece atendimentos para toda a comunidade interna e externa ao campus. Uma das ferramentas utilizadas para mensurar e avaliar as patologias dos praticantes da hipoterapia é o prontuário que é responsável por guardar as informações de identificação do praticante e diversos dados médicos necessários para o processo de cuidado do paciente. Conforme a crescente evolução das tecnologias de informação a sua utilização como ferramenta de inovação em diversas áreas vem se tornando cada vez maior fornecendo técnicas e recursos para auxílio em processamento, armazenamento e segurança de dados. A utilização de recursos de tecnologia de informações aplicadas ao setor de equoterapia fornecendo uma plataforma web visa facilitar o processo de gerenciamento de todas as informações voltadas aos atendimentos garantindo maior eficiência e praticidade nos procedimentos de cadastro, acesso de informações, alterações de dados e desligamento de novos praticantes, prontuários, profissionais e animais utilizados para realização dos atendimentos. Considerando a utilização das ferramentas de tecnologia da informação esta pesquisa teve como objetivo o desenvolvimento de uma aplicação web capaz de gerenciar prontuários eletrônicos dos praticantes de equoterapia facilitando o controle e tratamento de informações utilizadas durante todas as etapas do processo de triagem e tratamento dos praticantes.

**Palavras-chave:** Equoterapia; gerenciamento de informações; plataforma web

## **ABSTRACT**

Equine therapy is a therapeutic procedure that integrates and unifies various knowledge and strategies from the fields of health and education applied to the practice of horseback riding as a therapeutic procedure, focusing on the physical and psychological development of the participant. Therefore, it is widely used in cases of cerebral palsy, Down syndrome, behavioral disorders, autism, and various other cases of motor, social, and emotional impairment. Recognizing the importance and social need for free and quality hippotherapy services, the Federal Institute of Goiano - Ceres Campus maintains a specialized equine therapy center on its premises, offering services to the entire internal and external community of the campus. One of the tools used to measure and assess the pathologies of hippotherapy participants is the medical record, which is responsible for storing the participant's identification information and various medical data necessary for the patient's care process. As information technology advances, its use as an innovative tool in various fields is increasingly growing, providing techniques and resources to aid in data processing, storage, and security. The use of information technology resources applied to the field of equine therapy, through a web platform, aims to facilitate the management process of all information related to the services, ensuring greater efficiency and convenience in procedures such as registration, access to information, data changes, and enrollment of new participants, medical records, professionals, and animals used for the treatments. Considering the use of information technology tools, the objective of this work was to develop a web application capable of managing electronic medical records of equine therapy participants, facilitating the control and treatment of information used throughout all stages of the screening and treatment process for the participants.

**Keywords:** Equine therapy; information management; web platform.

## **LISTA DE ABREVIATURAS E SIGLAS**

HTML: Hypertext Markup Language.

CSS: Cascading Style Sheets.

PHP: Hypertext Preprocessor.

MVC: Model-View-Controller.

PEP: Prontuário eletrônico do paciente.

WEB: World Wide Web.

PWA: Progressive Web Apps.

IF: Instituto Federal.

SMTP: Simple Mail Transfer Protocol.

API: Application Programming Interface.

SQL: Structured Query Language.

SGBD: Sistema de Gerenciamento de Bancos de Dados.

REST: Representational State Transfer RESTful - RESTful Web Services.

HMP: História médica pregressa.

HMA: História da moléstia atual.

URLs: Uniform Resource Locators.

HTTP: Hypertext Transfer Protocol.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo de desenvolvimento da aplicação.....	24
Figura 2 – Modelo MVC.....	24
Figura 3: API REST:.....	26
Figura 4: Caso de uso de administrador do sistema.....	28
Figura 5: Caso de uso de usuário padrão do sistema.....	29
Figura 6: Fluxo geral para histórias de usuários.....	30
Figura 7: Camada de modelo da aplicação.....	35
Figura 8: Classes de controle da aplicação proposta.....	40
Figura 9: Camada de visão.....	43
Figura 10: Rotas das camadas de visão e controle.....	46
Figura 11: Rotas de autenticação de usuário.....	47
Figura 12: Rota de recuperação de senha por e-mail.....	48
Figura 13: Rota para alteração de senha.....	49
Figura 14: Rota de login na aplicação.....	49
Figura 15: Rota de logout da aplicação.....	50
Figura 16: Rota de registro(guest).....	51
Figura 17: Rota de login(guest).....	52
Figura 18: Diagrama físico da base de dados.....	56
Figura 19: Fluxo de navegação.....	60
Figura 20: Tela de login.....	62
Figura 21: Tela de início.....	63
Figura 22: Tela de registro de praticantes.....	63
Figura 23: Tela de visualização e alteração de plano terapêutico.....	64
Figura 24: Tela de cadastro de avaliação fisioterápica.....	65
Figura 25: Tela de controle de usuário para alteração de nome, e-mail e senha.....	66
Figura 26: Tela de controle de usuário para acompanhamento e segurança.....	66

## LISTA DE TABELAS

Tabela 1: Caso de uso manter praticante.....	30
Tabela 2: Caso de uso manter profissionais.....	31
Tabela 3: Caso de uso manter acompanhantes.....	31
Tabela 4: Caso de uso gerenciar avaliação psicológica.....	32
Tabela 5: Caso de uso gerenciar registro diário.....	32
Tabela 6: Caso de uso gerenciar plano terapêutico.....	33
Tabela 7: Caso de uso gerenciar plano terapêutico.....	33
Tabela 8: Casos de usos de usuários do sistema.....	34
Tabela 9: URLs para operações de instâncias da classe praticante.....	53
Tabela 10: URLs para operações de instâncias da classe responsáveis.....	53
Tabela 11: URLs para operações de instâncias da classe profissionais.....	53
Tabela 12: URLs para operações de instâncias da classe registro diário.....	54
Tabela 13: URLs para operações de instâncias da classe plano terapêutico.....	54
Tabela 14: URLs para operações de instâncias da classe avaliação psicológica.....	54
Tabela 15: URLs para operações de objetos da classe avaliação fisioterápica.....	55

# SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>12</b>
1.1 MOTIVAÇÃO.....	13
<b>2. OBJETIVOS.....</b>	<b>15</b>
2.1 OBJETIVO GERAL.....	15
2.2 OBJETIVOS ESPECÍFICOS.....	15
<b>3. MATERIAIS E MÉTODOS.....</b>	<b>16</b>
<b>4. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>18</b>
4.1 DESENVOLVIMENTO WEB.....	19
4.1.1 TECNOLOGIAS.....	20
4.1.1.1 FRAMEWORK LARAVEL 9.....	20
4.1.1.2 PHP.....	21
4.1.1.3 JAVASCRIPT.....	21
4.1.1.4 HTML.....	21
4.1.1.5 CSS.....	22
4.1.1.6 BOOTSTRAP.....	22
4.1.1.7 MAILHOG.....	22
4.1.1.8 MySQL.....	23
4.1.1.9 phpMyAdmin.....	23
4.1.2 CONCEITOS.....	23
<b>5. PROJETO DA APLICAÇÃO.....</b>	<b>27</b>
5.1 CASOS DE USO.....	27
5.1.1 CASO DE USO PARA ADMINISTRADOR DO SISTEMA.....	27
5.1.2 CASO DE USO PARA USUÁRIO PADRÃO DO SISTEMA.....	27
5.1.3 ESTÓRIAS DE USUÁRIO.....	29
5.2 CAMADA DE MODELO.....	34
5.3 CAMADA DE CONTROLE.....	39
5.4 CAMADA DE VISÃO.....	42
<b>6. IMPLEMENTAÇÃO DA APLICAÇÃO EM QUESTÃO.....</b>	<b>45</b>
6.1 SISTEMA DE ROTAS.....	45
6.2 SISTEMA DE ROTAS DE CONVIDADOS(GUEST).....	51
6.3 API RESTful.....	52
6.4 BASE DE DADOS.....	55
6.4.1 MIGRATIONS.....	57
6.5 SISTEMA DE LOGIN.....	57
6.5.1 CADASTRO DE USUÁRIOS.....	58
6.5.2 GERENCIAMENTO DE CREDENCIAIS.....	58
6.5.3 RECUPERAÇÃO DE SENHAS.....	59
<b>7. PRONTUÁRIO VIRTUAL.....</b>	<b>60</b>

7.1 INTERFACES DE REGISTRO.....	60
7.2 INTERFACES DE VISUALIZAÇÃO.....	61
<b>8. TELAS DA APLICAÇÃO.....</b>	<b>62</b>
8.1 TELA DE LOGIN.....	62
8.2 TELA DE INÍCIO.....	62
8.3 TELA DE REGISTRO DE PRATICANTES.....	63
8.4 TELA DE VISUALIZAÇÃO E ALTERAÇÃO DE DADOS DE PLANO TERAPÊUTICO.....	64
8.5 TELA DE CADASTRO DE AVALIAÇÃO FISIOTERÁPICA.....	64
8.6 TELA DE GERENCIAMENTO DE CREDENCIAIS DE USUÁRIO.....	65
<b>9. CONCLUSÃO.....</b>	<b>67</b>
<b>10. REFERÊNCIAS.....</b>	<b>69</b>

## 1. INTRODUÇÃO

A equoterapia é considerada um método de tratamento fisioterapêutico que pode ser utilizado como uma terapia ocupacional sendo uma alternativa de intervenção médica em diversas patologias que acometem seus pacientes nos âmbitos motores, psicológicos ou cognitivos. No contexto dos profissionais da saúde a Equoterapia atua como um método eficaz de tratamento das funções motoras utilizando de um cavalo como facilitador e mediador dos processos terapêuticos e educacionais de reabilitação e socialização (Zamo & Trentini, 2016).

Portanto, a equoterapia assume um papel importante para a sociedade em que é ofertada atuando como um modelo terapêutico que pode ser empregado no tratamento de quadros neuromotores. Reconhecendo a necessidade da equoterapia para a qualidade de vida da população, o IF Goiano-Campus Ceres fornece em suas instalações um centro de equoterapia que conta com uma equipe multidisciplinar que abrange fisioterapeutas, psicólogos, psicopedagogos e profissionais especializados em atuar juntamente ao equino. O centro de equoterapia do IF Goiano-Campus Ceres oferece atendimento a todas as cidades que integram o Vale do São Patrício, onde se localiza o campus em questão.

O Centro de Equoterapia do IF Goiano Campus Ceres foi desenvolvido em parceria com a Prefeitura Municipal de Ceres no ano de 2005 com o objetivo de fornecer atendimento de equoterapia para pessoas com deficiências e necessidades especiais que residem dentro do Vale de São Patrício. Seu foco principal é o atendimento de pessoas com deficiências cognitivas, psicológicas e motoras, tais como síndrome de Down, transtorno do espectro autista, transtorno do déficit de atenção e hiperatividade, dentre outras. O Centro de Equoterapia conta com uma estrutura completa para realização de diversas modalidades de atendimentos contendo picadeiro aberto, picadeiro fechado, recepção, sala de reuniões, selaria e uma trilha onde são realizados atendimentos com pessoas que possuem necessidades específicas.

Além da estrutura física, o centro em questão ainda conta com diversos profissionais de diversos segmentos da área da saúde do município de Ceres e de instituições de ensino parceiras que atuam em colaboração com o IF Goiano - Campus Ceres para que seja possível oferecer um atendimento gratuito e de

qualidade a todos os praticantes.

Considerando a importância da equoterapia no tratamento de patologias neuromotoras é de vital importância que o centro de equoterapia do IF Goiano consiga oferecer um atendimento eficiente em todas as etapas que englobam o processo de tratamento. Um destes processos é caracterizado pelo gerenciamento das informações necessárias para serem realizadas as avaliações de cada quadro clínico e seu processo de triagem. Portanto, compreende-se que nesta etapa se faz necessário o maior nível de eficiência e segurança destas informações a fim de manter a qualidade dos atendimentos e a integridade dos dados.

Levando em conta que o processo de tratamento e gerenciamento de dados no setor de equoterapia do IF Goiano utiliza de meios físicos para realizar todo o armazenamento das informações pessoais de cada praticante e todos os prontuários médicos dos mesmos. Percebe-se que este processo pode apresentar e, de fato, apresenta um alto grau de morosidade considerando que o armazenamento em meios físicos necessita da realização do processo manual de preenchimento de fichas, pesquisa de dados, alteração de informações pertinentes e desligamento dos praticantes e seus dados pessoais.

Segundo Kleemann (2016), as aplicações web podem ser construídas para atender a uma infinidade de demandas, podendo ser aplicações robustas para armazenamento e processamento massivo de dados até aplicações de página estática para visualização de dados.

Podemos considerar ainda que, por meio de Silva (2020) que um sistema WEB possui características próprias como a gerência de grandes volumes de dados, interação e navegação controlada pelo usuário da aplicação, atualizações tecnológicas constantes, que buscam proporcionar a estes mesmos usuários uma experiência confortável para continuar a usar a aplicação ou Website.

Sendo assim, o presente artigo visa o planejamento e o desenvolvimento de uma aplicação web que forneça os recursos necessários para o armazenamento e gerenciamento dos dados dos praticantes, fichas de avaliação, profissionais e outros dados necessários para o processo de triagem e planejamento dos atendimentos.

## 1.1 MOTIVAÇÃO

O centro de equoterapia gerencia diversas informações a respeito de praticantes, profissionais, animais, e prontuários médicos de praticantes. Estes conjuntos de informações são necessários durante todo o processo de atendimento do paciente, uma vez que é necessário que cada praticante seja submetido a um processo de triagem pela equipe de fisioterapia e equitação. As informações em questão são indispensáveis para serem definidas por meio do processo de triagem as melhores abordagens para cada praticante com suas necessidades, a equipe multiprofissional mais adequada e o prontuário fisioterapêutico do praticante.

O setor de equoterapia realiza o gerenciamento de diversos tipos de dados e informações armazenadas em meios impressos com a utilização de papel, o uso deste meio de armazenamento de dados gera uma série de adversidades para a equipe que gerencia o setor.

Desta forma, o processo de cadastro de praticantes, profissionais, animais e todo o processo de triagem geram um gasto elevado de material ecológico gerando danos na natureza e na biodiversidade brasileira.

Considerando o volume de dados que a equipe de gerenciamento deve registrar, analisar e armazenar utilizando impressões em papel, fica claro que estes são processos morosos e demandam muita energia de todos os profissionais envolvidos em todas as etapas do gerenciamento de informações.

Sendo assim, se faz necessário a utilização das tecnologias de informação e comunicação como uma poderosa aliada para buscar agilidade, praticidade e segurança no processo de cadastro e manipulação de informações vitais para o funcionamento do setor fornecendo uma ferramenta capaz de registrar e processar estes conjuntos de informações para garantir um maior nível de eficiência em todas as fases do gerenciamento de dados em comparação com a utilização de papel impresso para a mesma finalidade.

## **2. OBJETIVOS**

As seções 2.1 e 2.2 a seguir se referem aos objetivos da aplicação proposta.

### **2.1 OBJETIVO GERAL**

Este trabalho visa analisar e compreender a rotina dos profissionais do setor de equoterapia do Instituto Federal Goiano — Campus Ceres e desenvolver uma aplicação web que se adapte às necessidades de gerenciamento de dados para aprimorar os processos de cadastro, visualização, modificação e desligamento de funcionários, praticantes, acompanhantes; fornecer recursos para organizar as informações de prontuário médico para facilitar o processo de triagem e atendimento dos praticantes.

### **2.2 OBJETIVOS ESPECÍFICOS**

1. Projetar uma aplicação web para auxiliar o gestor e toda a equipe de terapeutas na manipulação de informações de profissionais atuantes, praticantes e seus respectivos prontuários eletrônicos.
2. Projetar um aplicativo capaz de substituir completamente o processo de manipulação de prontuários realizados manualmente em papel impresso.
3. Realizar o desenvolvimento da aplicação mediante as necessidades levantadas com o setor de equoterapia por meio das fichas impressas recolhidas no local e reuniões realizadas com os gestores.

### 3. MATERIAIS E MÉTODOS

O desenvolvimento da aplicação web foi dividido em duas etapas sendo que a primeira etapa contemplará todo o processo de modelagem onde foi levantado quais são os requisitos funcionais e não funcionais, análise de viabilidade, design, regras de negócio, análise das ferramentas que melhor se adequem às demandas da aplicação, definição da fase de testes do protótipo.

As ferramentas de desenvolvimento definidas para o desenvolvimento da aplicação web foram agrupadas de acordo com suas funções, sendo elas:

Linguagens de programação: PHP e Javascript;

Frameworks: Laravel 9 e Bootstrap;

Banco de dados: MariaDB(MySql);

De acordo com Lobo(2008), a engenharia de software é uma ciência que tem em vista analisar e catalogar as metodologias e padrões para o desenvolvimento de softwares de grande importância para empresas especializadas em desenvolvimento de softwares por definirem métodos que buscam aprimorar as técnicas e ferramentas além de todas as etapas que englobam o ambiente de desenvolvimento.

Após a etapa de engenharia foi iniciada a fase de implementação onde acontecerá toda a codificação do front-end e back-end por meio das ferramentas que foram previamente definidas na etapa anterior. A primeira fase da implementação abrange a codificação do back-end utilizando padrão de projeto MVC (model, view, controller) sendo intrínseco ao framework Laravel 9. O desenvolvimento do banco de dados ocorreu concomitantemente ao desenvolvimento do back-end, pois o Laravel 9 possui recursos próprios que permite ao desenvolvedor realizar a codificação da base de dados com o próprio framework.

A segunda fase da implementação é caracterizada pela codificação do front-end que engloba as interfaces de interação com o usuário e todas as características que definem o design das páginas do software e suas funcionalidades. Em seguida, ao final da codificação do protótipo inicial foi iniciada a fase de testes que conta com a realização de diversos testes na aplicação sendo eles: testes de rotas visando analisar o fluxo de dados e consistência das camadas de modelos, visualização e os controladores, testes dos requisitos implementados

para garantir a eficiência, disponibilidade e a integralidade dos dados que são manipulados pela aplicação. Mediante aos resultados da fase de teste foi iniciada a etapa de correção de adversidades no sistema e tratamento de erros conforme o sistema apresentou inconstâncias que comprometeram o funcionamento das ferramentas. Na figura a seguir é possível visualizar as três principais etapas do desenvolvimento da aplicação proposta.



Figura 1: Fluxo de desenvolvimento da aplicação.

## 4. FUNDAMENTAÇÃO TEÓRICA

Segundo Majewski e Oliveira (2019), A equoterapia é um tratamento fisioterapêutico e educacional que foca no uso de cavalos como principal forma de aplicação, a equoterapia busca o desenvolver as capacidades físicas, psicológicas e motoras de seus praticantes, podendo trazer benefícios a todos os indivíduos portadores ou não de alguma patologia. Conforme a necessidade do praticante o cavalo pode produzir, após treinamento, passos diferentes para serem conjugados ao atendimento proposto pelos profissionais que acompanham cada sessão.

De acordo com Flexa (2019), os ganhos promovidos pela Equoterapia são mediados por profissionais que atuam em uma equipe multiprofissional em perspectiva interdisciplinar onde a equipe mínima é composta por: um adestrador de cavalo, um fisioterapeuta e um psicólogo.

Conforme Santos e Damian (2017), O prontuário é a base fundamental de informações sobre o paciente, suas patologias e condições, fornecendo auxílio aos profissionais da saúde durante os momentos em que se é necessário realizar determinadas tomadas de decisões sobre o quadro clínico dos pacientes. Sendo assim, é consolidada a relevância e a necessidade do planejamento e aplicação de um prontuário capaz de armazenar as informações mais pertinentes a respeito das patologias, condições, receitas de medicações e observações gerais sobre os pacientes que garantem a continuidade de todo o tratamento prestado ao paciente.

Segundo Santos(2017), é possível dizer que o prontuário do paciente nada mais é do que um documento que armazena um conjunto de informações e imagens que são fruto de uma análise profissional dos fatos referentes às condições de saúde do paciente e todos os procedimentos recomendados e aplicados aos mesmos.

Ainda segundo Santos(2017), todas as informações contidas no prontuário possuem natureza pessoal, sigilosa e científica cujo objetivo é auxiliar a comunicação entre toda a equipe de profissionais envolvidas em cada um dos quadros clínicos em aberto.

A aplicação de tecnologias da informação e suas ferramentas no contexto da equoterapia pode auxiliar os profissionais envolvidos em todas as etapas do atendimento ao praticante fornecendo aplicações e dispositivos para tornar mais ágil e efetivos os processos administrativos envolvendo todo o gerenciamento de

informações.

As tecnologias de informação podem ainda garantir um nível mais elevado de confidencialidade, integridade e disponibilidade de todos os dados necessários para o pleno funcionamento de todo o ciclo de atendimento aos pacientes.

Segundo Cordeiro (2018), com o crescimento acelerado das tecnologias de comunicação e informação no âmbito da saúde foi possível realizar diversos avanços em todos os aspectos do processo de atendimento do paciente, entre estes avanços é o sistema de prontuário eletrônico do paciente (PEP) que vem sendo adotado visando elevar a eficiência e eficácia durante a gestão hospitalar.

Quando as atividades de atendimento e sua gestão são armazenadas em meios digitais é garantido um nível mais elevado de segurança das informações sensíveis de pacientes e de toda a equipe médica. Sendo assim, o prontuário eletrônico traz diversas vantagens de controle, busca e armazenamento quando comparado ao prontuário impresso em papel que ainda é amplamente utilizado no contexto da saúde humana (Cordeiro, 2018).

#### 4.1 DESENVOLVIMENTO WEB

Segundo Silva (2019, p. 03), as aplicações web são soluções de software que atuam completamente online, porém, se comportam como aplicativos nativos e podem sofrer com quedas no desempenho e acesso de suas funções, todavia, com o surgimento de tecnologias como PWA (*Progressive Web Apps*) o desenvolvimento de aplicações web vem sofrendo constantes evoluções visando a resolução destes problemas.

Ainda segundo Silva (2019, p. 11), o termo “web apps” é comumente usado para se referir a aplicativos que não requerem instalação no dispositivo sendo acessados por meio de um link. Esses aplicativos são geralmente mais leves e usam a memória cache do navegador para minimizar o consumo de memória e processamento no dispositivo. Um exemplo de web app é o Facebook Lite, que oferece funcionalidades limitadas em comparação com o aplicativo nativo do Facebook, mas é uma alternativa viável para reduzir o consumo de recursos do dispositivo. Este tipo de aplicativo pode oferecer uma experiência satisfatória para usuários que desejam evitar o alto consumo de recursos do dispositivo e ainda assim acessar funcionalidades básicas do aplicativo.

De acordo com Santiago (2020, p. 05), o desenvolvimento web é caracterizado como o planejamento, construção e manutenção de sistemas que operam exclusivamente no ambiente online. O desenvolvimento web engloba a camada de front-end que atua na máquina cliente interagindo diretamente com o usuário por meio de um navegador e a camada back-end que se localiza no servidor web e é responsável pelas funcionalidades que englobam as regras de negócio do sistema

#### 4.1.1 TECNOLOGIAS

Nos tópicos a seguir são apresentadas todas as tecnologias e ferramentas utilizadas para o desenvolvimento da aplicação proposta.

##### 4.1.1.1 FRAMEWORK LARAVEL 9

A utilização de frameworks vem se tornando cada vez mais intrínseca ao processo de desenvolvimento de software devido aos benefícios que os mesmos trazem no momento de codificação das funcionalidades do sistema, pois, os frameworks fornecem uma gama de ferramentas pré-definidas para simplificar e tornar mais eficiente a etapa de programação do sistema.

Segundo Ferreira(2021), os frameworks são ferramentas que vem tomando cada vez mais espaço no processo de desenvolvimento de aplicações mesmo não sendo uma novidade. Ainda segundo Ferreira(2021), isso se deve ao fato de que os frameworks podem integrar diversas ferramentas úteis em um mesmo ambiente de programação e ainda possuem uma estrutura que facilita o processo de codificação.

O framework Laravel foi criado por Taylor B. Otwell em 2011 sendo intitulado como um framework de desenvolvimento PHP para artesãos Web dando a entender que o Laravel unia as técnicas de desenvolvimento com a arte podendo oferecer uma ferramenta com uma sintaxe expressiva, elegante e eficiente para a codificação de aplicações robustas e complexas.

Conforme afirmado por Gabardo (2017), Laravel possui uma curva de aprendizado clara e relativamente curta quando comparada a outros frameworks e afirma que após pouco tempo de prática com o Laravel, é possível notar a clareza e simplicidade em sua sintaxe e a qualidade de desenvolvimento que o mesmo proporciona, além de apresentar um código limpo e elegante. Tais características também se refletem na documentação da ferramenta, a qual é detalhada, contendo

bons exemplos de todas as funcionalidades e é bastante completa.

#### 4.1.1.2 PHP

A linguagem PHP (Hypertext Processor) é uma linguagem de programação de código aberto (open source) utilizada para desenvolvimento de sistemas Web que se integra ao HTML (HyperText Markup Language) e ao CSS (Cascading Style Sheet). O PHP atua principalmente no servidor onde a aplicação é armazenada para gerar uma gama de funções e conteúdos dinâmicos exibidos para o usuário por meio de um navegador.

#### 4.1.1.3 JAVASCRIPT

Javascript é uma linguagem de programação web utilizada para diversas finalidades, porém, é amplamente utilizada para desenvolvimento de interfaces de interação com o usuário na camada front-end da aplicação e assim como o PHP o Javascript pode ser utilizado em conjunto com HTML e CSS para a realização da codificação do corpo e do design da página Web.

A utilização do JavaScript tem como principal foco tornar as páginas muito mais dinâmicas e responsivas para o usuário final e podendo contar a utilização de diversos frameworks em conjunto com JS para a codificação de funcionalidades ou sistemas com um maior nível de eficiência e robustez.

Conforme a documentação da MDN Web Docs (2021), o JavaScript é amplamente utilizado em scripts e funções que são incorporados a uma página HTML para interagir dinamicamente com o conteúdo da página e atualizar suas informações, em tempo real.

#### 4.1.1.4 HTML

O HTML é uma linguagem de marcação desenvolvida por Tim Berners-Lee na década de 90, visando permitir a criação de páginas da web com hipertexto e compatibilidade com diversos dispositivos. O HTML 5 é a versão mais recente e atual da linguagem, que evoluiu ao longo dos anos.

A principal característica do HTML é sua capacidade de criar conteúdo navegável por meio de links. Para criar uma página HTML, é necessário conhecer as tags básicas da linguagem, que incluem a definição do início do documento, a inclusão de informações sobre a página e a inserção do conteúdo visível.

#### 4.1.1.5 CSS

Segundo Araújo (2021), o CSS (Cascading Style Sheets, folhas de estilo em cascata) é uma ferramenta utilizada para definir regras de design e aparência dos elementos HTML de uma aplicação Web utilizando um navegador como meio de visualização e renderização do front-end do sistema.

O CSS possui recursos para personalização altamente abrangente de textos, imagens, tabelas, cores por meio das tags que são criadas para compor o corpo da aplicação com o HTML.

#### 4.1.1.6 BOOTSTRAP

Bootstrap foi criado no ano de 2010 pelo designer Mark Otto e o desenvolvedor Jacob Thornton dentro do Twitter para ser utilizado para a resolução de problemas internos e inconsistências no próprio Twitter, mas rapidamente se tornou um dos mais populares frameworks front-end de código aberto no mundo (BOOTSTRAP, 2020<sup>a</sup>).

Segundo Santiago(2020), o Bootstrap é um framework de desenvolvimento front-end que disponibiliza um código-fonte para interfaces web e por conta destes códigos é considerado como um “kit de ferramentas” que engloba componentes como HTML, CSS e JavaScript.

Ainda segundo Santiago(2020), o framework Bootstrap tem como principal objetivo auxiliar na criação de sites amigáveis e responsivos, podendo ser utilizado em diversas IDE's, editores de texto e pode ser empregada com qualquer linguagem back-end.

“Projetado para qualquer um, em qualquer lugar o Bootstrap torna o desenvolvimento front-end web mais rápido e fácil. Ele é feito para pessoas de todos os níveis e dispositivos de qualquer forma ou tamanho” (BOOTSTRAP, 2015).

#### 4.1.1.7 MAILHOG

Segundo Mendes (2022, p. 72), a plataforma MailHog é uma aplicação simples que visa apoiar desenvolvedores em seus projetos fornecendo um sistema capaz de enviar e-mails utilizando o protocolo de envio de e-mails SMTP geralmente ativo na porta 8025 em um ambiente de testes de software.

#### 4.1.1.8 MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto que teve sua primeira versão lançada em 1995. É amplamente utilizado por desenvolvedores em todo o mundo para criar, gerenciar e manter bancos de dados. Este sistema oferece recursos avançados, como replicação e particionamento, que visam melhorar a disponibilidade e desempenho do sistema, além de ser altamente escalável e confiável, suportando uma abundante quantidade de transações e conexões simultâneas. Projetado para ser fácil de usar, o MySQL oferece aos desenvolvedores uma ampla gama de ferramentas e APIs para trabalhar com o banco de dados.

Segundo Souza (2019, p.40), MySQL é um SGBD desenvolvido inicialmente para pequenas e médias empresas, que naquela época possuíam bancos de dados menores em comparação aos atuais, que ultrapassam consideravelmente os anteriores.

Em seguida, Souza (2019, p.40) afirma que o MySQL é um gerenciador de banco de dados amplamente conhecido, utilizado por grandes empresas no mercado, como a HP, Bradesco, NASA e SONY. Seu crescimento é notável, tendo em vista sua grande utilização no mercado e desde sua criação, o MySQL continua em constante desenvolvimento e é um banco de dados de código aberto (open-source), distribuído sob as licenças GNU/GPL (General Public Licence), que significa Licença Pública Geral.

#### 4.1.1.9 phpMyAdmin

O phpMyAdmin é uma ferramenta de software livre criada com a linguagem PHP com o principal objetivo de lidar com a administração de bancos de dados SQL utilizando a Web. O phpMyAdmin oferece suporte a uma vasta gama de operações com MySQL e MariaDB que permitem a manipulação de tabelas, colunas índices, relações, usuários, permissões e diversos outros recursos que podem ser executados por meio de uma interface de usuário em que é possível executar diretamente qualquer declaração de SQL (phpMyAdmin, 2023).

## 4.1.2 CONCEITOS

Nas seções a seguir são apresentados os conceitos tecnológicos que fundamentaram o desenvolvimento da aplicação proposta.

### 4.1.2.1 PADRÃO MVC (model, view e controller)

Model-View-Controller (MVC) é um padrão de projeto de software criado na década de 1970 por Trygve Reenskaug, este focado no reuso de código e na separação de conceitos em três camadas interconectadas onde cada uma é responsável por uma característica específica. Uma camada responsável pela apresentação dos dados, sendo esta a que o usuário interage com o sistema, uma para a modelagem dos dados e outra para a manipulação desses dados (IVANOVICH et al., 2019).

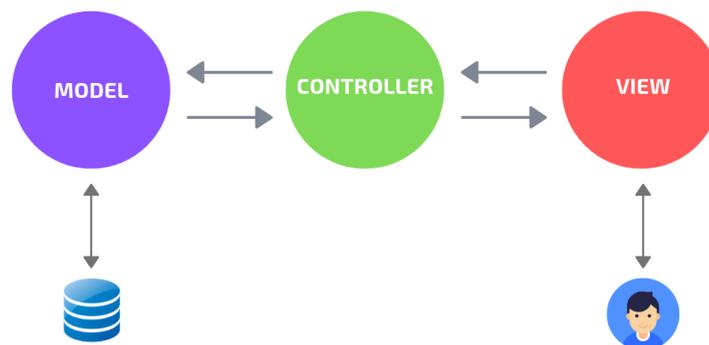


Figura 2 – Modelo MVC

Fonte: Andrade (2020)

O padrão de projetos MVC tem como foco segregar a interface de comunicação com o usuário chamada de “View” e sua camada de negócios referenciada como “Model”. Utilizando uma terceira camada chamada de Controller é possível criar comportamentos que definirão como a aplicação deverá se comportar e comunicar entre as views e models contidas na aplicação.

De acordo com Silva(2020), a camada Controller tem a função de interligar as camadas de view e model, sendo responsável pela lógica aplicada às regras de negócio e gerenciamento dos dados. Ao receber uma requisição, o Controller ativa uma função correspondente para processar e retornar os dados solicitados. A

camada Model tem em vista garantir um maior nível de encapsulamento para o sistema, limitando-se apenas às informações de operações e requisições.

Segundo Silva(2020), os dados são representados por meio da camada de view que pode interagir com o usuário e levar dados e informações do sistema ao mesmo com determinado nível de abstração por meio de inúmeras formas de representação de dados. A View recebe os dados requisitados pelo Model que são repassados para o Controller. O Controller atua como um mensageiro no processo de entrada, processamento dos dados e retorno das informações, manipulando as Views e Models da aplicação conforme a necessidade.

#### 4.1.2.2 BANCO DE DADOS RELACIONAL

Os bancos de dados são essenciais para a gestão e organização de informações em sistemas de informação e aplicativos. Eles consistem em um conjunto estruturado de informações ou dados que são organizados, armazenados e recuperados de forma eficiente e segura. Por meio de um sistema de banco de dados, as informações são armazenadas em um formato estruturado, permitindo que sejam facilmente gerenciadas, atualizadas e recuperadas quando necessário. Além disso, os bancos de dados possibilitam a tomada de decisões informadas com base nas informações armazenadas, uma vez que permitem que os dados sejam armazenados de forma consistente e segura. Assim, os bancos de dados são fundamentais para a gestão de informações em sistemas e aplicativos, contribuindo para a eficiência e segurança da informação.

Segundo Elmasri et al. (2005), a criação dos SGBDs relacionais teve como objetivo principal separar o armazenamento físico dos dados de sua representação conceitual. Essa representação permitiu que as empresas reduzissem o uso de grandes pilhas de papéis, que eram anteriormente usados para armazenar informações importantes.

### 4.1. 2.3 API RESTful

A arquitetura REST (Representational State Transfer) é um estilo de arquitetura popular para projetar sistemas de comunicação entre aplicações distribuídas. Ela é baseada em uma abordagem leve, que utiliza corretamente os protocolos HTTP e recursos para representar informações. Por conta de sua facilidade de implementação e flexibilidade, o REST tornou-se uma abordagem amplamente utilizada no desenvolvimento de APIs.

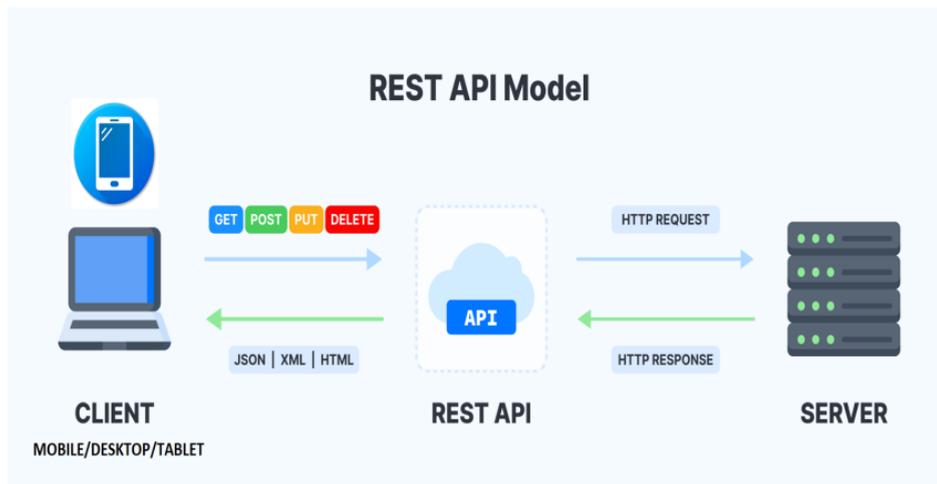


Figura 3: API REST.

Fonte: Agnihotri (2022)

Os sistemas RESTful são conhecidos por sua alta escalabilidade, capacidade de manipular grandes quantidades de dados e suporte a diferentes tipos de dispositivos e clientes. A arquitetura REST é altamente modular, permitindo a adição de novas funcionalidades ao sistema sem afetar as já existentes. Como resultado, o REST tem se tornado cada vez mais popular para soluções de integração de sistemas em diversos setores, como comércio eletrônico, mídias sociais e serviços financeiros.

## **5. PROJETO DA APLICAÇÃO**

Seção reservada para a apresentação das funcionalidades da aplicação proposta bem como sua arquitetura.

### **5.1 CASOS DE USO**

O diagrama de caso de uso (Use Case) é um elemento gráfico exclusivo por ser um diagrama usado para modelar o modo como as pessoas esperam usar um sistema. O diagrama descreve quem são os usuários relevantes, os serviços que eles exigem do sistema e os serviços que eles precisam oferecer ao sistema (PENDER, 2004).

Os diagramas de caso de uso são normalmente utilizados como parte de uma abordagem dirigida por caso de uso mais abrangente, que também inclui uma criação textual dos casos de uso individuais e a extração de cenários. A descrição textual enfatiza os requisitos detalhados para um caso de uso. Os cenários enfatizam a necessidade de explorar opções na execução do caso de uso, testar os requisitos e oferecer um plano de teste de alto nível para as fases de desenvolvimento subsequentes (PENDER, 2004).

#### **5.1.1 CASO DE USO PARA ADMINISTRADOR DO SISTEMA**

Na Figura 3 a seguir são apresentados os casos de usos arquitetados e desenvolvidos que podem ser acessados e manipulados pelo usuário padrão e o administrador do sistema.

Os utilizadores poderão gerenciar informações referentes aos casos de uso de “praticantes”, “acompanhantes”, “profissionais”, “registros diários”, “avaliações psicológicas”, “fisioterapêuticas”, “plano terapêutico”, “usuário” e, por fim, somente o usuário administrador poderá cadastrar novos usuários no sistema.

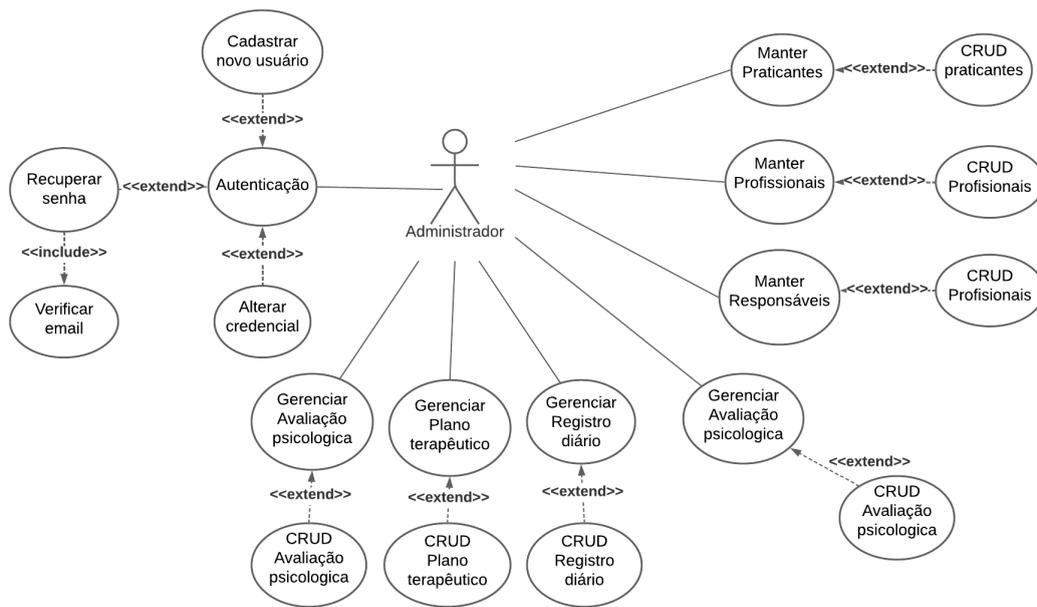


Figura 4: Caso de uso de administrador do sistema.

### 5.1.2 CASO DE USO PARA USUÁRIO PADRÃO DO SISTEMA

A Figura 4 apresenta a apresentação dos casos de usos que podem ser acessados pelo usuário padrão do sistema que poderá acessar grande parte das funcionalidades do sistema, somente sendo tendo acesso restrito a funcionalidade de “registro de novos usuários”.

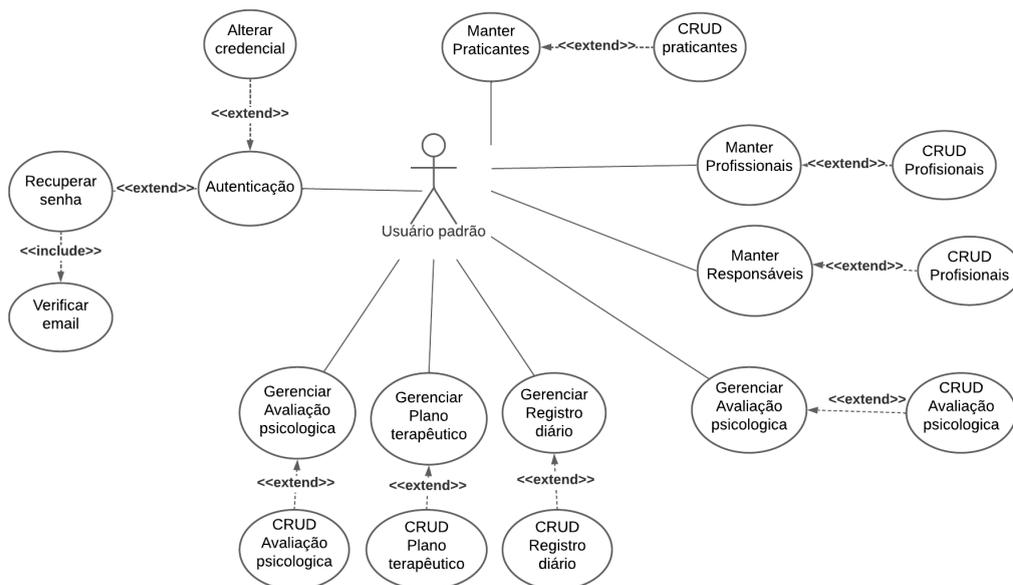


Figura 5: Caso de uso de usuário padrão do sistema.

### 5.1.3 ESTÓRIAS DE USUÁRIO

As histórias de usuário são uma técnica amplamente utilizada na metodologia ágil de desenvolvimento de software, empregada para a captura de requisitos de usuário de forma precisa e eficiente. Tais histórias são escritas na perspectiva do usuário final e descrevem funcionalidades específicas que atendem às suas necessidades e desejos, que devem ser implementadas pelo time de desenvolvimento. Cada história de usuário é composta por três elementos fundamentais, quais sejam: título, descrição e critérios de aceitação, que descrevem a funcionalidade desejada de forma detalhada e explícita.

De acordo com Andrade Filho (2019, apud CHOPA DE et al., 2017), o processo ágil de desenvolvimento de software começa com a criação de estórias de usuário, que são pequenas afirmações baseadas em solicitações dos clientes do sistema. As estórias de usuário seguem o formato “Como um <tipo de usuário> eu quero <algum objetivo> então <alguma ação>”, por exemplo: “Como um administrador eu quero entrar no sistema, então eu poderei usar a aplicação”.

Na Figura 5 pode ser observado como o modelo de metodologia ágil define o fluxo ideal para a criação de estórias de usuário para serem realizadas as definições de usuário da ação, objetivo e justificativa dos requisitos levantados e implementados no sistema.



Figura 6: Fluxo geral para estórias de usuários.

Sendo assim, é possível compreender a necessidade e justificativa para as funcionalidades do sistema e atribuir um peso para cada uma que foi utilizado para classificar o nível de complexidade e importância das funcionalidades para que a equipe de desenvolvimento defina a viabilidade dos requisitos funcionais.

Na tabela abaixo são apresentadas as estórias de usuários que justificam os casos de uso da aplicação em questão definidos nas Figuras 3 (caso de uso de administrador do sistema) e 4 (caso de uso de usuário padrão do sistema).

Tabela 1: Caso de uso “manter praticante”.

Usuário do caso de uso	Funcionalidades do caso de uso	Estória do usuário
Administrador e usuário padrão	Cadastrar praticante	Como utilizador do sistema quero cadastrar novos praticantes para gerir seus prontuários.
Administrador e usuário padrão	Visualizar praticante	Como utilizador do sistema quero visualizar os dados dos praticantes cadastrados.
Administrador e usuário padrão	Atualizar praticante	Como utilizador do sistema quero alterar os dados dos praticantes cadastrados para evitar inconsistências.
Administrador e usuário padrão	Excluir praticante	Como utilizador do sistema quero excluir dados dos praticantes do sistema.

Tabela 2: caso de uso “manter profissionais”.

Usuário do caso de uso	Funcionalidades do caso de uso	Estória do usuário
Administrador	Cadastrar profissional	Como administrador do sistema quero cadastrar novos usuários para administrar seus prontuários.
Administrador	Visualizar profissional	Como administrador do sistema quero visualizar os dados dos profissionais cadastrados.
Administrador	Atualizar profissional	Como administrador do sistema quero alterar os dados dos profissionais cadastrados para evitar inconsistências.
Administrador	Excluir profissional	Como administrador do sistema quero excluir dados dos profissionais do

		sistema para desvinculá-los.
--	--	------------------------------

Tabela 3: Caso de uso “manter acompanhantes”.

Usuário do caso de uso	Funcionalidades do caso de uso	Estória do usuário
Administrador e usuário padrão	Cadastrar acompanhante	Como utilizador do sistema quero cadastrar acompanhantes para os praticantes.
Administrador e usuário padrão	Visualizar acompanhante	Como utilizador do sistema quero visualizar os dados dos acompanhantes cadastrados.
Administrador e usuário padrão	Atualizar acompanhante	Como utilizador do sistema quero alterar os dados dos acompanhantes cadastrados para evitar inconsistências.
Administrador e usuário padrão	Excluir acompanhante	Como utilizador do sistema quero excluir dados dos acompanhantes do sistema.

Tabela 4: Caso de uso “gerenciar avaliação psicológica”.

Usuário do caso de uso	Funcionalidades do caso de uso	Estória do usuário
Administrador e usuário padrão	Cadastrar avaliação psicológica	Como utilizador do sistema quero cadastrar uma avaliação psicológica para um praticante.
Administrador e usuário padrão	Visualizar avaliação psicológica	Como utilizador do sistema quero visualizar os dados das avaliações psicológicas cadastradas.
Administrador e usuário padrão	Atualizar avaliação psicológica	Como utilizador do sistema quero alterar os dados das avaliações psicológicas cadastradas para evitar inconsistências.
Administrador e usuário padrão	Excluir avaliação psicológica	Como utilizador do sistema quero excluir dados das

		avaliações psicológicas do sistema.
--	--	-------------------------------------

Tabela 5: Caso de uso “gerenciar registro diário”.

Usuário do caso de uso	Funcionalidades do caso de uso	Estória do usuário
Administrador e usuário padrão	Cadastrar registro diário	Como utilizador do sistema quero cadastrar um registro diário para um praticante.
Administrador e usuário padrão	Visualizar registro diário	Como utilizador do sistema quero visualizar os dados dos registros diários cadastrados.
Administrador e usuário padrão	Atualizar registro diário	Como utilizador do sistema quero alterar os dados dos registros diários cadastrados para evitar inconsistências.
Administrador e usuário padrão	Excluir registro diário	Como utilizador do sistema quero excluir dados dos registros diários do sistema.

Tabela 6: Caso de uso “gerenciar plano terapêutico”.

Usuário do caso de uso	Funcionalidades do caso de uso	Estória do usuário
Administrador e usuário padrão	Cadastrar plano terapêutico	Como utilizador do sistema quero cadastrar um plano terapêutico para um praticante.
Administrador e usuário padrão	Visualizar plano terapêutico	Como utilizador do sistema quero visualizar os dados dos planos terapêuticos cadastrados.
Administrador e usuário padrão	Atualizar plano terapêutico	Como utilizador do sistema quero alterar os dados dos planos terapêuticos cadastrados para evitar inconsistências.
Administrador e	Excluir plano terapêutico	Como utilizador do sistema

usuário padrão		quero excluir dados dos planos terapêuticos do sistema.
----------------	--	---

Tabela 7: Caso de uso “gerenciar plano terapêutico”.

Usuário do caso de uso	Funcionalidades do caso de uso	Estória do usuário
Administrador e usuário padrão	Cadastrar avaliação psicológica	Como utilizador do sistema quero cadastrar uma avaliação psicológica para um praticante.
Administrador e usuário padrão	Visualizar avaliação psicológica	Como utilizador do sistema quero visualizar os dados das avaliações psicológicas cadastradas.
Administrador e usuário padrão	Atualizar avaliação psicológica	Como utilizador do sistema quero alterar os dados das avaliações psicológicas cadastradas para evitar inconsistências.
Administrador e usuário padrão	Excluir avaliação psicológica	Como utilizador do sistema quero excluir dados das avaliações psicológicas do sistema.

Tabela 8: Casos de usos de “usuários do sistema”.

Usuário do caso de uso	Funcionalidades do caso de uso	Estória do usuário
Administrador	Cadastrar usuário	Como administrador do sistema quero cadastrar novos usuários para o sistema.
Administrador e usuário padrão	Alterar credencial	Como usuário do sistema quero alterar minhas credenciais de e-mail e senha.
Administrador e usuário padrão	Recuperar senha	Como usuário do sistema quero solicitar a recuperação de senha via e-mail.

## 5.2 CAMADA DE MODELO

Segundo Silval(2023) modelo (Model) desempenha um papel fundamental na manipulação detalhada das informações, sendo altamente recomendado empregar sua utilização em consultas, cálculos e implementação das regras de negócio em um sistema ou site. O modelo pode acessar abrangentemente todas as informações, independentemente de sua fonte, seja um banco de dados ou um arquivo XML. Dessa forma, sua utilização permite uma abordagem mais precisa e refinada na manipulação e análise dos dados.

Os métodos de cada classe estão armazenados na parte inferior da estrutura e assim como os métodos no padrão UML estes também possuem um nível de acesso e o tipo de dado que este método retornará após sua execução na aplicação.

Na Figura 6 é apresentado o diagrama de classes contendo atributos e métodos dos objetos de praticantes, profissionais, acompanhantes, usuários, avaliação fisioterápica, avaliação psicológica, registro diário e plano terapêutico sendo que os atributos são responsáveis por armazenar as informações de seus respectivos objetos e os métodos consistem na caracterização das ações que os objetos das classes apresentadas poderão realizar na aplicação.

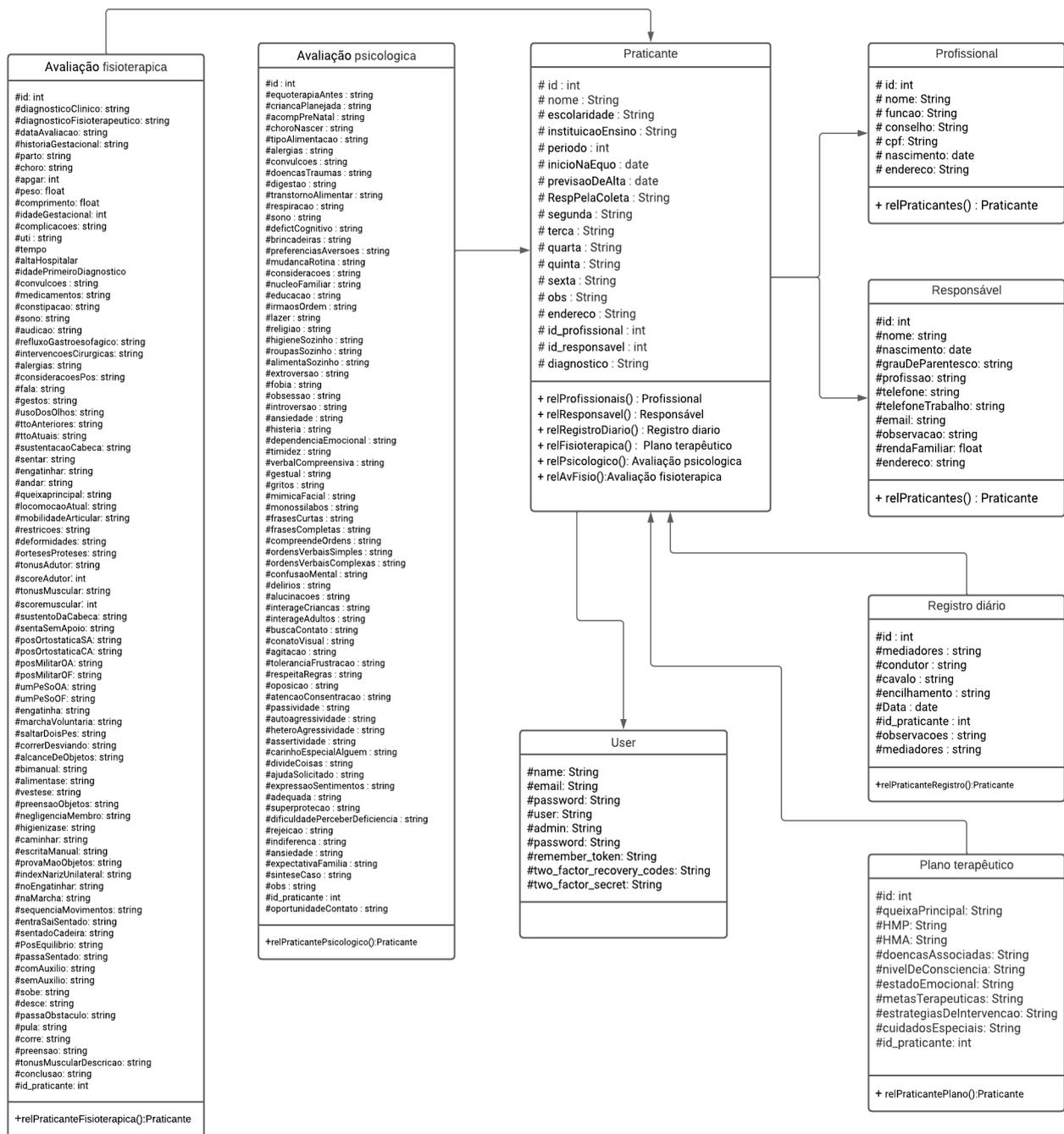


Figura 7: Camada de modelo da aplicação.

A classe de “praticante” é o centro da aplicação e é responsável por representar os praticantes de equoterapia na aplicação, a classe conta com os atributos: *id*, *nome*, *escolaridade*, *instituicaoEnsino*, *periodo*, *inicioNaEquo*, *previsaoDeAlta*, *RespPelaColeta*, *segunda*, *terça*, *quarta*, *quinta*, *sexta*, *obs*, *endereco*, *id\_profissional*, *id\_responsavel* e *diagnostico*. Os atributos *nome*, *escolaridade*, *instituicaoEnsino* e *periodo* são responsáveis por coletar informações pessoais do praticante, os atributos *inicioNaEquo*, *previsaoDeAlta*, *RespPelaColeta* tem como função receber as datas de início e término das sessões de equoterapia, por fim, os atributos *segunda*, *terça*, *quarta*, *quinta*, *sexta* e *obs* são responsáveis pelo cronograma diário de atividades do praticante e as observações acerca do praticante em específico.

A classe “responsável” tem como função representar os responsáveis legais pelos praticantes caso estes sejam incapazes de responder por si devido a implicações legais, físicas ou cognitivas. Os responsáveis geralmente são parentes próximos dos praticantes e em alguns casos são a primeira forma de contato entre o centro de equoterapia e o praticante. A classe Responsável conta com os seguintes atributos: *id*, *nome*, *nascimento*, *grauDeParentesco*, *profissão*, *telefone*, *telefoneTrabalho*, *e-mail*, *observacao*, *rendaFamiliar* e *endereco*, sendo que *nome*, *nascimento* e *grauDeParentesco* representam informações pessoais que identificam o indivíduo responsável e sua relação com o praticante e os demais atributos são encarregados de armazenar dados de contato do responsável sendo eles: *telefone*, *e-mail* e *observação* que tem como foco informações adicionais que são pertinentes ao centro de equoterapia.

“Profissional” é a classe que contém os atributos e métodos que descrevem os objetos encarregados de representar os profissionais que atuam dentro do setor de equoterapia e realizam o cadastro e o preenchimento de avaliações e planos de tratamento dos praticantes. A classe profissional possui os seguintes métodos: *id*, *nome*, *função*, *conselho*, *cpf*, *nascimento*, *endereco* e *contato* que representam características de identificação pessoal do profissional na aplicação.

“Registro diário” se trata da classe responsável pela definição de atributos e métodos para os registros diários que compõem as atividades realizadas pelos praticantes em cada uma de suas sessões de equiterapia dentro do setor de equoterapia. O registro diário conta com os atributos: *id*, *mediadores*, *condutor*, *cavalo*, *encilhamento*, *data*, *id\_praticante*, *observacoes* e *mediadores*, onde os

atributos *mediadores* e *condutor* definem quais profissionais estarão envolvidos na seção em questão, o método *cavalo* define qual animal foi utilizado como ferramenta de tratamento, *encilhamento* descreve quais materiais foram utilizados durante a sela do equino, o método de *data* informa em qual data a seção específica ocorreu e *id\_praticante* informa qual praticante está vinculado na seção.

**Avaliação psicológica:** É a classe que define os atributos e métodos utilizados para avaliar a condição psicológica do praticante que foi posteriormente utilizada como parâmetro para definir qual a melhor abordagem dentro da equoterapia para lidar com o quadro clínico geral do praticante. Esta classe possui os seguintes métodos: *equoterapiaAntes*, *criancaPlanejada*, *acompPreNatal*, *choroNascer*, *tipoAlimentacao*, *alergias*, *convulcoes*, *doencasTraumas*, *digestao*, *transtornoAlimentar*, *respiracao*, *sono*, *defictCognitivo*, *brincadeiras*, *preferenciasAversoes*, *mudancaRotina*, *consideracoes*, *nucleoFamiliar*, *educacao*, *irmaosOrdem*, *lazer*, *religiao*, *higieneSozinho*, *roupasSozinho*, *alimentaSozinho*, *extroversao*, *fobia*, *obsessao*, *introversao*, *ansiedade*, *histeria*, *dependenciaEmocional*, *timidez*, *verbalCompreensiva*, *gestual*, *gritos*, *mimicaFacial*, *monossilabos*, *frasesCurtas*, *frasesCompleatas*, *compreendeOrdens*, *ordensVerbaisSimples*, *ordensVerbaisComplexas*, *confusaoMental*, *delirios*, *alucinacoes*, *interageCrianças*, *interageAdultos*, *buscaContato*, *conatoVisual*, *agitacao*, *toleranciaFrustracao*, *respeitaRegras*, *oposicao*, *atencaoConsentracao*, *passividade*, *autoagressividade*, *heteroAgressividade*, *assertividade*, *carinhoEspecialAlguem*, *divideCoisas*, *ajudaSolicitado*, *expressaoSentimentos*, *adequada*, *superprotecao*, *dificuldadePerceberDeficiencia*, *rejeicao*, *indiferenca*, *ansiedade*, *expectativaFamilia*, *sinteseCaso*, *obs*, *id\_praticante* e *oportunidadeContato*. Esta classe representa todas as características contidas na ficha de avaliação psicológica atualmente utilizada como instrumento de coleta de dados neuropsicológicos do praticante no setor de equoterapia.

**Plano terapêutico:** A classe plano terapêutico possui métodos que definem o quadro geral do praticante, metas terapêuticas a serem atingidas, estratégias de intervenção terapêutica e diversas informações a respeito da patologia, saúde mental do praticante e um método específico que vincula o plano terapêutico ao praticante na qual pertence. O plano terapêutico tem como base os dados e informações que são coletados presencialmente no setor de equoterapia do IF Goiano — Campus Ceres para realização da avaliação e definição de abordagens

terapêuticas para o praticante na etapa de triagem do mesmo. A classe de plano terapêutico conta com os métodos: *queixaPrincipal, HMP, HMA, doençasAssociadas, nivelDeConsciencia, estadoEmocional, metasTerapeuticas, estrategiasDeIntervencao, cuidadosEspeciais, id\_praticante*.

**Avaliação fisioterápica:** A avaliação fisioterápica é a classe responsável pela definição dos métodos que vão definir o estado físico e motor do praticante considerado como critério durante o processo de triagem do praticante para compreender seu quadro geral e elaborar a melhor metodologia equiterápica para lidar com suas necessidades e restrições. A classe de avaliação fisioterápica possui os seguintes métodos: *diagnosticoClinico, diagnosticoFisioterapeutico, dataAvaliacao, historiaGestacional, parto, choro, apgar, peso, comprimento, idadeGestacional, complicacoes, uti, tempo, altaHospitalar, idadePrimeiroDiagnostico, convulcoes, medicamentos, constipacao, sono, audicao, refluxoGastroesofagico, intervencoesCirurgicas, alergias, consideracoesPos, fala, gestos, usoDosOlhos, ttoAnteriores, ttoAtuais, sustentacaoCabeca, sentar, engatinhar, andar, queixaprincipal, locomocaoAtual, mobilidadeArticular, restricoes, deformidades, ortesesProteses, tonusAdutor, scoreAdutor, tonusMuscular, scoremuscular, sustentoDaCabeca, sentaSemApoio, posOrtostaticaSA, posOrtostaticaCA, posMilitarOA, posMilitarOF, umPeSoOA, umPeSoOF, engatinha, marchaVoluntaria, saltarDoisPes, correrDesviando, alcanceDeObjetos, bimanual, alimentase, vestese, preensaoObjetos, negligenciaMembro, higienizase, caminhar, escritaManual, provaMaoObjetos, indexNarizUnilateral, noEngatinhar, naMarcha, sequenciaMovimentos, entraSaiSentado, sentadoCadeira, PosEquilibrio, passaSentado, comAuxilio, semAuxilio, sobe, desce, passaObstaculo, pula, corre, preensão, tonusMuscularDescricao, conclusão e id\_praticante*.

“User” A classe *User* define os métodos para a criação de usuários da aplicação por parte do gestor do sistema com os seguintes métodos: *name, email, password, user, admin, password, remember\_token, e two\_factor\_secret*. Os métodos *nome, e-mail, password* e *user* definem as informações de identificação e credenciais do usuário para que ele possa realizar login no sistema, por fim, os métodos *remember\_token, e two\_factor\_secret* são utilizados para gerar medidas de segurança sendo que *remember\_token* é utilizado para autenticar a seção do usuário, é o método responsável por gerar códigos de recuperação únicos e *two\_factor\_secret* armazena um código de recuperação que pode ser compartilhado.

### 5.3 CAMADA DE CONTROLE

Segundo Filho(2019, p.45), a camada de controle está especificada no padrão MVC e é ativada por meio das interações do usuário com a camada de visão que realiza a chamada da camada de controle e envia os dados necessários para sua operação por meio de parâmetros fornecidos pelo usuário na camada de visão e por meio das instâncias de classes de serviços foram feitas as operações de cadastro, leitura, alteração e desligamento de dados e informações registradas no sistema.

O framework Laravel 9 realiza a chamada da camada de controle por meio de requisições HTTP com os métodos GET, POST, PUT e DELETE para levar a informação entre a camada de visão e os métodos e funções contidos na camada de controle.

Na camada de controle define como padrão as funções “index”, “create”, “show”, “store”, “edit”, “update” e “destroy” sendo que a função index é comumente utilizada para exibir uma listagem de recursos do sistema, a função create é utilizada para apresentar um formulário que pode ser utilizado para criação de objetos ou recursos dentro do sistema.

A função “show” é encarregada de buscar e retornar diversos tipos de dados e informações para a camada de visão para que esta possa exibi-los ao usuário, a função store é responsável pela validação de dados recebidos por meio de formulários e posteriormente esta função realiza a persistência dos dados em um banco de dados.

A função “edit” consegue buscar um conjunto de informações que receberão atualizações e enviá-los para a camada de visão que exibirá os dados ao usuário que realizará as alterações dos dados e ao enviar os dados novamente a função de update receberá os dados para realizar a atualização dos dados no banco de dados.

Por meio da função “destroy” pode-se receber requisições e parâmetros da camada de visão para realizar a exclusão de dados que foram previamente persistidos na base de dados do sistema.

Conforme a figura 7, é possível observar todas as classes de controle da aplicação sendo elas: “PraticanteController”, “ProfissionalController”, “AcompanhanteController”, “RegistroDiarioController”, “PsicologicoController”,

“PlanoTerapeuticoController”, “FisioterapicaController” e notar que cada uma delas se comunica com uma entidade específica na camada de visão que é responsável por realizar as chamadas dos métodos da classe de controle em que está vinculada e fornecer os parâmetros necessários para as funções realizarem suas operações.

Segue abaixo uma breve descrição sobre cada uma das classes.

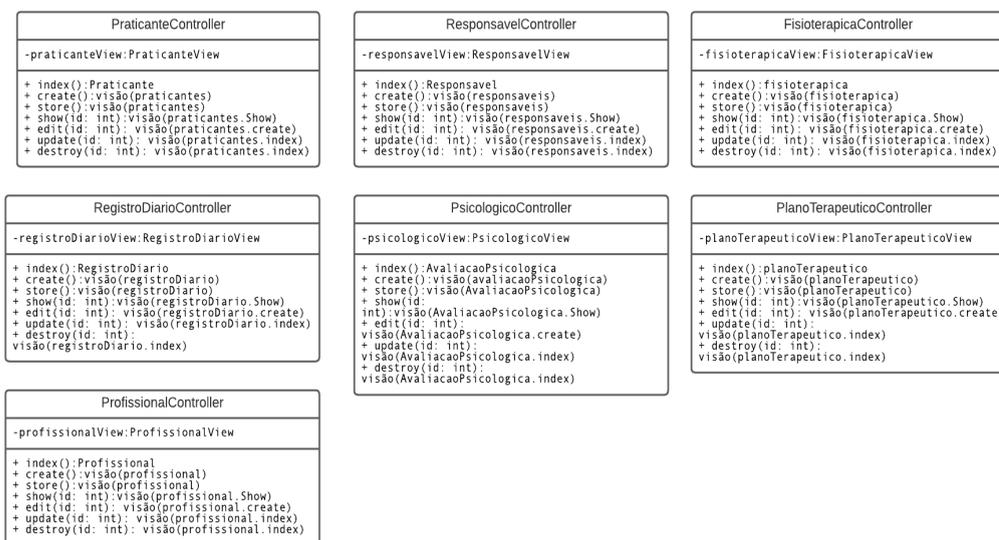


Figura 8: Classes de controle da aplicação proposta.

“PraticanteController” é a classe de controle que faz referência aos objetos estudantes da aplicação e, em conjunto com a “praticanteView” que é o elemento da camada de visão que representa os praticantes estas podem realizar as operações de CRUD dos praticantes na aplicação.

“ResponsavelController” é o controlador que realiza as operações de CRUD dos acompanhantes de praticantes. Esta classe de controle atua juntamente do elemento “responsavelView” contido na camada de visão para poderem ser realizadas as funções de gerenciamento dos responsáveis.

“ProfissionaisController” é a classe de controle que atua em conjunto com o elemento “profissionalView” na camada de visão e realizam as ações de CRUD de instâncias da classe de profissionais.

“RegistroDiarioController” é a classe de controle que armazena as funções para CRUD de instâncias da classe de registro diário. A classe de controle de registros diários se comunica e recebe os parâmetros necessários para realizar suas operações através da “registroDiarioView” que é o elemento da camada de visão

responsável pelo tráfego de informações entre o usuário e o sistema.

“PsicologicaController” é o controlador que define os métodos para realização das funções de CRUD de objetos referentes a avaliação psicológica dos praticantes. O controlador em questão se comunica com a view “psicologicaView” que realiza o chamado das funções de gerenciamento das informações e fornece os parâmetros necessários para as mesmas serem executadas corretamente.

“FisioterapicaController” é o controlador responsável pelas funções para CRUD das instâncias de avaliações fisioterápicas realizadas pelos profissionais, esta classe atua diretamente com a view “fisioterapicaView” realizando as chamadas das funções encarregadas do processo de gerenciamento das avaliações fisioterápicas na aplicação.

“PlanoTerapeuticoController” é a classe que define as funções CRUD para os objetos que representam os planos terapêuticos dos praticantes, atuando em conjunto com o “planoTerapeuticoView” são responsáveis pela interação com o usuário e a realização das tarefas de gerenciamento dos planos terapêuticos.

## 5.4 CAMADA DE VISÃO

No contexto do padrão de arquitetura de software MVC, a camada de visão (View) é responsável pela apresentação das informações ao usuário, atuando como a interface do usuário com a aplicação e representando o front-end da mesma. A camada de visualização interage com o usuário, transmite suas solicitações para o controlador (Controller) e é responsável por exibir as respostas recebidas por este da camada modelo (Model).

O framework Laravel 9 conta com a utilização do template Blade que é um de seus principais recursos para desenvolvimento de interfaces gráficas de maneira ágil e eficiente devido sua sintaxe simples e intuitiva quando se trata de codificar o layout e a estrutura de uma página da web. O template Blade permite a criação de layouts gráficos de que podem ser reutilizados pela camada de visão durante a execução da aplicação utilizando um conceito chamado de herança de layouts onde é definido um layout principal e os demais poderão reaproveitar os conteúdos deste, porém, sua maior vantagem é a possibilidade de utilizar diretivas que conseguem simplificar a codificação HTML e PHP que torna o processo de desenvolvimento mais rápido e assertivo.

Em outras palavras, a camada de visão desempenha um papel fundamental na interface com o usuário e na comunicação entre as outras camadas do sistema para atender às necessidades do usuário.

No contexto da aplicação proposta, a camada de visão engloba diversos elementos que formarão a interface gráfica que representará as diversas classes de controle separadamente. A camada de visão da aplicação web foi particionada em grupos onde cada um destes contém elementos responsáveis pela renderização das interfaces de cadastro, visualização, edição e desligamento de informações no sistema, sendo assim, os grupos que compõem a camada de visão são: “praticanteView”, “profissionalView”, “responsavelView”, “registroDiarioView”, “psicologicoView”, “fisioterapicaView”, “planoTerapeuticoView”.

Na figura 8 é possível observar a camada de visão da aplicação com suas subdivisões e os respectivos elementos que compõem cada uma delas.

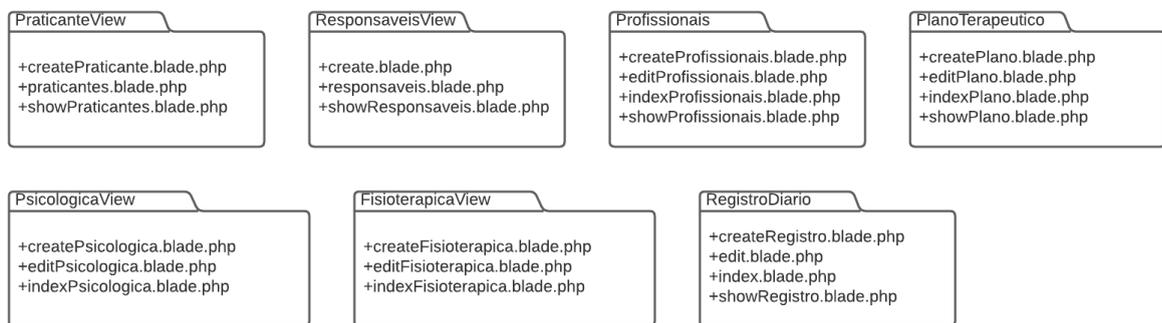


Figura 9: Camada de visão

Os grupos que compõem a camada de visão apresentada na figura 8 agrupam um conjunto de modelos Blade onde cada elemento é responsável pelo layout gráfico de uma página da aplicação para a realização das funções CRUD, logo, os elementos podem ser identificados e classificados por meio do prefixo que se encontra nos títulos dos elementos que fazem referência ao papel do layout dentro das operações básicas de manipulação de dados sendo estes:

**Index:** Este prefixo indica que a página em questão atua como uma página inicial do agrupamento e tem como função realizar uma listagem de todos os elementos de outros grupos que possuem vínculo com o grupo em que a

página em questão pertence, a página de index fornece acesso a outras páginas do mesmo grupo sendo elas responsáveis pela edição e visualização dos dados de objetos.

**Create:** É o prefixo que indica a função de cadastro dentro da camada de view, portanto, os elementos com este prefixo são utilizados para gerar uma interface gráfica contendo todas as tags HTML e comandos PHP necessários para ser renderizada uma página Web de cadastro funcional para se comunicar com um controlador e realizar a persistência dos dados.

**Show:** Indica a função de visualização completa de todos os dados pertinentes a um ou mais objetos. Todos os elementos da camada de visão com este prefixo contém o conjunto de tags e comandos para gerar as interfaces que dão ao usuário uma perspectiva completa e organizada a respeito das informações dentro da aplicação. Os elementos com este prefixo também possuem os elementos HTML e PHP que realizam a desativação dos dados de um objeto na qual ela faça referência. Sendo assim, as páginas Web com este prefixo possuem elementos das funções de visualização e exclusão de objetos e informações.

**Edit:** Indica as páginas web com a função de atualização dos dados de objetos da aplicação, onde cada um dos elementos com este prefixo pode gerar uma interface gráfica capaz de armazenar uma estrutura apropriada para apresentar e alterar os dados de objetos instanciados quando necessário.

## 6. IMPLEMENTAÇÃO DA APLICAÇÃO

Nesta seção são listadas e documentadas todas as características inerentes ao processo de implementação das funcionalidades da aplicação.

### 6.1 SISTEMA DE ROTAS

O sistema de rotas é uma funcionalidade do framework Laravel que define como as URLs do sistema estão mapeadas para que os controladores e views possam se comunicar e lidar com as requisições HTTP utilizando um conjunto de ferramentas que especificam um conjunto de regras e ações para a manipulação de URL e requisições.

Por padrão, todas as regras que definem o sistema de rotas do Laravel se encontra no arquivo “web.php” no diretório /routes e neste arquivo podem ser construídas as rotas do sistema por meio de uma sintaxe simples e versátil disponibilizada pelo próprio framework para facilitar a compreensão e codificação das rotas.

As rotas da aplicação proposta foram projetadas e construídas para atender cada um dos grupos que compõem a camada de visão e todas as classes de controle que estarão em contato direto com estes grupos com o intuito de facilitar e simplificar a troca de informações entre as camadas, sendo assim, cada grupo da camada de visão possui sua própria rota na aplicação que se vincula diretamente com o controlador com que os grupos se comunicarão.

Por meio da figura 9 é possível observar a construção das rotas que definem a comunicação entre os grupos que compõem a camada de visão e seus respectivos controladores para a realização das funções de CRUD.

```

Route::get('/', 'Auth\AuthenticatedSessionController@create');

Route::resource('/praticantes', 'praticanteController');
Route::put('/praticantes/{id}', [praticanteController::class, 'update'])->name('praticante.update');
Route::resource('/profissionais', profissionalController::class);
Route::resource('/responsaveis', responsavelController::class);
Route::resource('/registrodiario', registrodiarioController::class);
Route::resource('/planoterapeutico', PlanoTerapeuticoController::class);
Route::resource('/fisioterapica', fisioterapicaController::class);
Route::resource('/psicologico', psicologicoController::class);

Route::get('/register', 'Auth\registeredUserController@create')->name('register')->middleware('admin');

```

Figura 10: Rotas das camadas de visão e controle

Na Figura 9 é possível observar a definição das rotas utilizando o caractere “/” seguido pelo nome da rota que faz referência ao nome do controlador e do elemento da camada de visão na qual a rota pertence, portanto, a aplicação possui as seguintes rotas:

**/praticantes:** A rota de praticantes garante o tráfego de requisições HTTP entre os elementos do grupo “praticanteView” com o controlador responsável pelas ferramentas de CRUD para praticantes.

**/praticantes/{id}:** É uma rota específica para a função de atualização das informações pertinentes aos praticantes onde é utilizado o identificador do praticante(id) como requisição ao controlador que realizará a busca pelo praticante referenciado pelo id e retornará para a camada de visão os seus dados para a atualização por parte do usuário final.

**/profissionais:** É a rota que define a comunicação entre o grupo que compõe a codificação da interface gráfica de profissionais e sua respectiva classe na camada de controle para realização do gerenciamento de objetos da classe de profissionais.

**/responsaveis:** Esta rota define a comunicação entre o controlador que realiza as funções CRUD das instâncias dos responsáveis e o seu respectivo grupo na camada de visão

**/registrodiario:** A rota de registro diário que realiza a interligação entre o controlador de instâncias de registro diário com seu respectivo grupo na camada de visão.

**/planoterapeutico:** A rota responsável pela comunicação entre o controlador de instâncias de planos terapêuticos com seu respectivo grupo na camada de visão.

**/fisioterapica:** A rota de registro diário que realiza a interligação entre o controlador de instâncias de avaliações fisioterápicas com seu respectivo grupo na camada de visão para realização das funções de CRUD de objetos.

**/psicologico:** esta rota define a conexão entre o controlador de instâncias de avaliações psicológicas com seu respectivo grupo na camada de visão.

Além das rotas que permitem a troca de requisições entre os controladores e a camada de visão das funcionalidades principais de CRUD a aplicação ainda conta com outra gama de rotas utilizadas para fins de segurança e autenticação do sistema que são tratadas a seguir.

```
Route::middleware([
    'auth:sanctum',
    config('jetstream.auth_session'),
    'verified'
])->group(function () {
    Route::get('/dashboard', function () {
        return view('welcome');
    }->name('dashboard'));
});
```

Figura 11: Rotas de autenticação de usuário

Conforme a Figura 10, é apresentado um bloco de código apresentado define uma rota denominada "dashboard" que é acessível somente a usuários autenticados, com uma sessão de autenticação válida e endereço de e-mail verificado, através da utilização de três middlewares: "auth:sanctum", "config('jetstream.auth\_session')" e "verified". A rota em questão foi implementada como um método HTTP GET e, quando acessada corretamente, retornará a visualização denominada 'welcome'. Esse código é uma estratégia de proteção de recursos web que garante a segurança e privacidade dos usuários.

```
Route::post('/forgot-password',  
[App\Http\Controllers\Auth\PasswordResetLinkController::class, 'store'])  
->middleware('guest')  
->name('password.email');
```

Figura 12: Rota de recuperação de senha por e-mail.

A Figura 11 apresenta um bloco de código com uma rota para a implementação de recuperação de senha, por meio do envio de um e-mail, acessível apenas para usuários não autenticados, a fim de garantir a proteção das informações confidenciais do usuário.

Este código define uma rota para recuperação de senha, utilizando o método HTTP POST e nomeada como "password.email", para a URL "/forgot-password". Quando essa rota é acionada, o método "store" da classe "PasswordResetLinkController" é executado, sendo este localizado no diretório "App\Http\Controllers\Auth" e o middleware "guest" é utilizado nessa rota, o que significa que somente usuários não autenticados poderão acessá-la, garantindo a segurança e privacidade do processo de recuperação de senha.

```

Route::get('/reset-password/{token}',
[App\Http\Controllers\Auth\NewPasswordController::class, 'create'])
    ->middleware('guest')
    ->name('password.reset');

Route::post('/reset-password',
[App\Http\Controllers\Auth\NewPasswordController::class, 'store'])
    ->middleware('guest')
    ->name('password.update');

```

Figura 13: Rota para alteração de senha.

Na Figura 12 são apresentadas as rotas fundamentais para garantir a segurança e a privacidade do processo de redefinição de senha, permitindo que apenas usuários autorizados acessem as funcionalidades. Além disso, o framework Laravel simplifica e padroniza a implementação dessas rotas, tornando a programação mais eficiente e produtiva.

A primeira rota utiliza o método HTTP GET e tem a URL "/reset-password/{token}". Esta rota realiza a chamada do método "create" da classe "NewPasswordController" que se encontra no diretório "App\Http\Controllers\Auth". O objetivo desta rota é permitir que o usuário acesse a página de criação de nova senha. Para garantir a segurança do processo, o middleware "guest" é utilizado, permitindo o acesso apenas a usuários não autenticados.

A segunda rota utiliza o método HTTP POST e tem a URL "/reset-password". Esta rota chama o método "store" da classe "NewPasswordController". O objetivo desta rota é processar a solicitação de redefinição de senha feita pelo usuário. Assim como na primeira rota, o middleware "guest" é utilizado para assegurar que somente usuários não autenticados possam acessá-la.

```

Route::get('/login',
[App\Http\Controllers\Auth\AuthenticatedSessionController::class, 'create'])
    ->middleware('guest')
    ->name('login');

Route::post('/login',
[App\Http\Controllers\Auth\AuthenticatedSessionController::class, 'store'])
    ->middleware('guest');

```

Figura 14: Rota de login na aplicação.

As rotas apresentadas na Figura 13 permitem que um usuário se autentique no sistema e também realize o logout. O uso do middleware "guest" na rota de login garante que apenas usuários não autenticados acessem a página de login, enquanto a ausência de middleware na rota de logout garante que apenas usuários autenticados realizem o logout. O framework Laravel simplifica a implementação dessas rotas, tornando o processo de autenticação mais eficiente e produtivo.

Na primeira rota, nomeada como "login", utiliza o método HTTP GET e chama o método "showLoginForm" da classe "LoginController", localizada em "App\Http\Controllers\Auth". O objetivo desta rota é permitir que um usuário acesse a página de login do sistema. Para garantir a segurança do processo de autenticação, o middleware "guest" é utilizado, permitindo o acesso apenas a usuários não autenticados.

A segunda rota, nomeada como "logout", utiliza o método HTTP POST e chama o método "logout" da classe "LoginController". O objetivo desta rota é processar a solicitação de logout efetuada pelo usuário. É importante destacar que essa rota não é protegida por middleware, pois ela só deve ser acessada por usuários autenticados.

```
Route::post('/logout',  
    [App\Http\Controllers\Auth\AuthenticatedSessionController::class, 'destroy'])  
    ->middleware('auth')  
    ->name('logout');
```

Figura 15: Rota de logout da aplicação.

Na Figura 14 contém a rota de "logout", essa rota permite que um usuário autenticado no sistema encerre sua sessão, garantindo a segurança do sistema e evitando o acesso não autorizado a recursos protegidos. O uso do middleware "auth" assegura que somente usuários autenticados possam acessar esta rota, mantendo a integridade do processo de encerramento da sessão.

Essa rota utiliza o método HTTP POST e tem a URL "/logout". Quando um usuário acessa esta rota, ela chama o método "destroy" da classe "AuthenticatedSessionController" localizada em "App\Http\Controllers\Auth". O objetivo desta rota é encerrar a sessão de um usuário autenticado no sistema, impedindo o acesso não autorizado a recursos protegidos.

A rota é nomeada como "logout", permitindo que ela seja facilmente referenciada em outras partes do código. O framework Laravel fornece meios de

simplificar a implementação desta rota para não haver processos complexos durante a codificação do sistema de login e logout da aplicação.

Para garantir que apenas usuários autenticados realizem o logout, o middleware "auth" é utilizado nesta rota. Isso significa que, para acessar esta rota, é necessário que o usuário tenha realizado a autenticação no sistema previamente.

## 6.2 SISTEMA DE ROTAS DE CONVIDADOS(GUEST)

As rotas de acesso ao sistema sem autenticação são denominadas como rotas de "guest", seu código base foi desenvolvido utilizando um middleware que se encontra fora do arquivo padrão de rotas web.php, porém, as rotas de convidado podem ser invocadas dentro do arquivo padrão de rotas.

O uso do middleware "guest" em rotas é comumente aplicado em páginas de login, recuperação de senha e registro de usuários, onde o acesso deve ser restrito a usuários não autenticados. Ao agrupar várias rotas em um único conjunto, é possível aplicar a mesma restrição de acesso a todas elas, tornando o processo de implementação mais eficiente e produtivo.

```
Route::get('register',  
[RegisteredUserController::class, 'create'])  
->name('register');  
  
Route::post('register',  
[RegisteredUserController::class, 'store']);
```

Figura 16: Rota de registro(guest).

A Figura 15 apresenta o bloco de código que implementa a funcionalidade de registro de usuários em uma aplicação Laravel 9, definindo duas rotas que direcionam o usuário para a página de registro e registram um novo usuário no banco de dados. O uso de nomes para as rotas torna a referência a elas mais simples e organizada onde a primeira rota utiliza o método HTTP GET e direciona o usuário para a página de registro, enquanto a segunda rota utiliza o método HTTP POST e é responsável por registrar o usuário.

Ao acessar a primeira rota, o usuário é direcionado para o método "create" do controlador "RegisteredUserController", que é responsável por retornar a view de registro de usuários. A segunda rota, por sua vez, aciona o método "store" do mesmo controlador, que é responsável por receber os dados enviados pelo usuário no formulário de registro e criar um novo registro na base de dados.

```
Route::get('login',  
    [AuthenticatedSessionController::class, 'create'])  
    ->name('login');  
  
Route::post('login',  
    [AuthenticatedSessionController::class, 'store']);
```

Figura 17: Rota de login(guest).

A Figura 16 apresenta um trecho de código que implementa a funcionalidade de autenticação de usuários por meio de recursos do próprio Laravel 9, definindo duas rotas que direcionam o usuário para a página de login e autenticam o usuário no sistema. O uso de nomes para as rotas torna a referência a elas mais simples e organizada.

Ao acessar a primeira rota, o usuário é direcionado para o método "create" do controlador "AuthenticatedSessionController", que é responsável por retornar a view de login. A segunda rota, por sua vez, aciona o método "store" do mesmo controlador, que é responsável por receber os dados enviados pelo usuário no formulário de login e autenticá-lo no sistema.

### 6.3 API RESTful

API Restful é um modelo de arquitetura muito utilizado para o desenvolvimento de APIs, que segue um conjunto bem definido de práticas e padrões. Essa abordagem se baseia no protocolo HTTP, que permite a comunicação entre diferentes sistemas padronizadamente e interoperável.

A API RESTful utiliza como meio para realizar as trocas de informações entre toda a aplicação os métodos HTTP, como GET, POST, PUT e DELETE definidos e utilizados no padrão arquitetural REST.

Tabela 9: URLs para operações de instâncias da classe praticante

Requisição HTTP	Método executado	Descrição
POST /praticantes/create	create	Registra um novo usuário.
GET /praticantes	índex	Listagem de praticantes.
GET /praticantes/{id}	show	Visualiza um praticante específico.
PUT /praticantes/{id}	update	Altera os dados de um praticante.
DELETE /praticantes/{id}	delete	Exclui um praticante.

Tabela 10: URLs para operações de instâncias da classe responsáveis

Requisição HTTP	Método executado	Descrição
POST /responsaveis/create	create	Registra um novo responsável.
GET /responsaveis	índex	Listagem de responsáveis.
GET /responsaveis/{id}	show	Visualiza um responsável específico.
PUT /responsaveis/{id}	update	Altera os dados de um responsável.
DELETE /responsaveis/{id}	delete	Exclui um responsável.

Tabela 11: URLs para operações de instâncias da classe profissionais

Requisição HTTP	Método executado	Descrição
POST /profissionais/create	create	Registra um novo profissional.
GET /profissionais	índex	Listagem de profissionais.
GET /profissionais/{id}	show	Visualiza um profissional específico.
PUT /profissionais/{id}	update	Altera os dados de um profissional.

DELETE /profissionais/{id}	delete	Exclui um profissional.
----------------------------	--------	-------------------------

Tabela 12: URLs para operações de instâncias da classe registro diário

Requisição HTTP	Método executado	Descrição
POST /registrodiario/create	create	Cadastra um novo registro.
GET /registrodiario	index	Listagem de registros.
GET /registrodiario/{id}	show	Visualiza um registro específico.
PUT /registrodiario/{id}	update	Altera os dados de um registro.
DELETE /registrodiario/{id}	delete	Exclui um registro.

Tabela 13: URLs para operações de instâncias da classe plano terapêutico

Requisição HTTP	Método executado	Descrição
POST /planoterapeutico/create	create	Cadastra um novo plano.
GET /planoterapeutico	index	Listagem de planos.
GET /planoterapeutico/{id}	show	Visualiza um plano específico.
PUT /planoterapeutico/{id}	update	Altera os dados de um plano.
DELETE /planoterapeutico/{id}	delete	Exclui um plano.

Tabela 14: URLs para operações de instâncias da classe avaliação psicológica

Requisição HTTP	Método executado	Descrição
POST /psicologico/create	create	Cadastra uma avaliação.
GET /psicologico	index	Listagem de avaliações.
GET /psicologico/{id}	show	Visualiza uma avaliação específica.
PUT /psicologico/{id}	update	Altera os dados de uma avaliação.
DELETE /psicologico/{id}	delete	Exclui uma avaliação.

Tabela 15: URLs para operações de objetos da classe avaliação fisioterápica

Requisição HTTP	Método executado	Descrição
POST /fisioterapia/create	create	Cadastra uma avaliação.
GET /fisioterapia	index	Listagem de avaliações.
GET /fisioterapia/{id}	show	Visualiza uma avaliação específica.
PUT /fisioterapia/{id}	update	Altera os dados de uma avaliação.
DELETE /fisioterapia/{id}	delete	Exclui uma avaliação.

## 6.4 BASE DE DADOS

O sistema gerenciador de banco de dados(SGBD) utilizado como ferramenta para construção e manipulação dos dados da aplicação proposta foi o sistema MariaDB que utiliza linguagem SQL, oferece recursos avançados, como suporte a JSON, replicação de alta disponibilidade e sharding para distribuir dados em várias instâncias de bancos de dados.

A escolha desta ferramenta para manipulação da base de dados se deu pela alta versatilidade do SGBD quando se trata de lidar com diversos dados e conjuntos de dados mantendo a eficiência da aplicação e sua compatibilidade com diversas linguagens de programação, incluindo PHP que é a linguagem escolhida para o desenvolvimento do back-end do sistema proposto.

A base de dados da aplicação conta com as seguintes colunas: avaliacaofisioterapia, migrations, password\_resets, permissions, personal\_access\_tokens, planoterapeutico, praticantes, profissionais, psicologicos, registrodiario, responsaveis, sessions, users, failed\_jobs e roles. Na figura 17 é apresentado o diagrama físico da base de dados com todas as colunas com seus respectivos campos e detalhes de descrição de dados pertinentes a cada um destes.

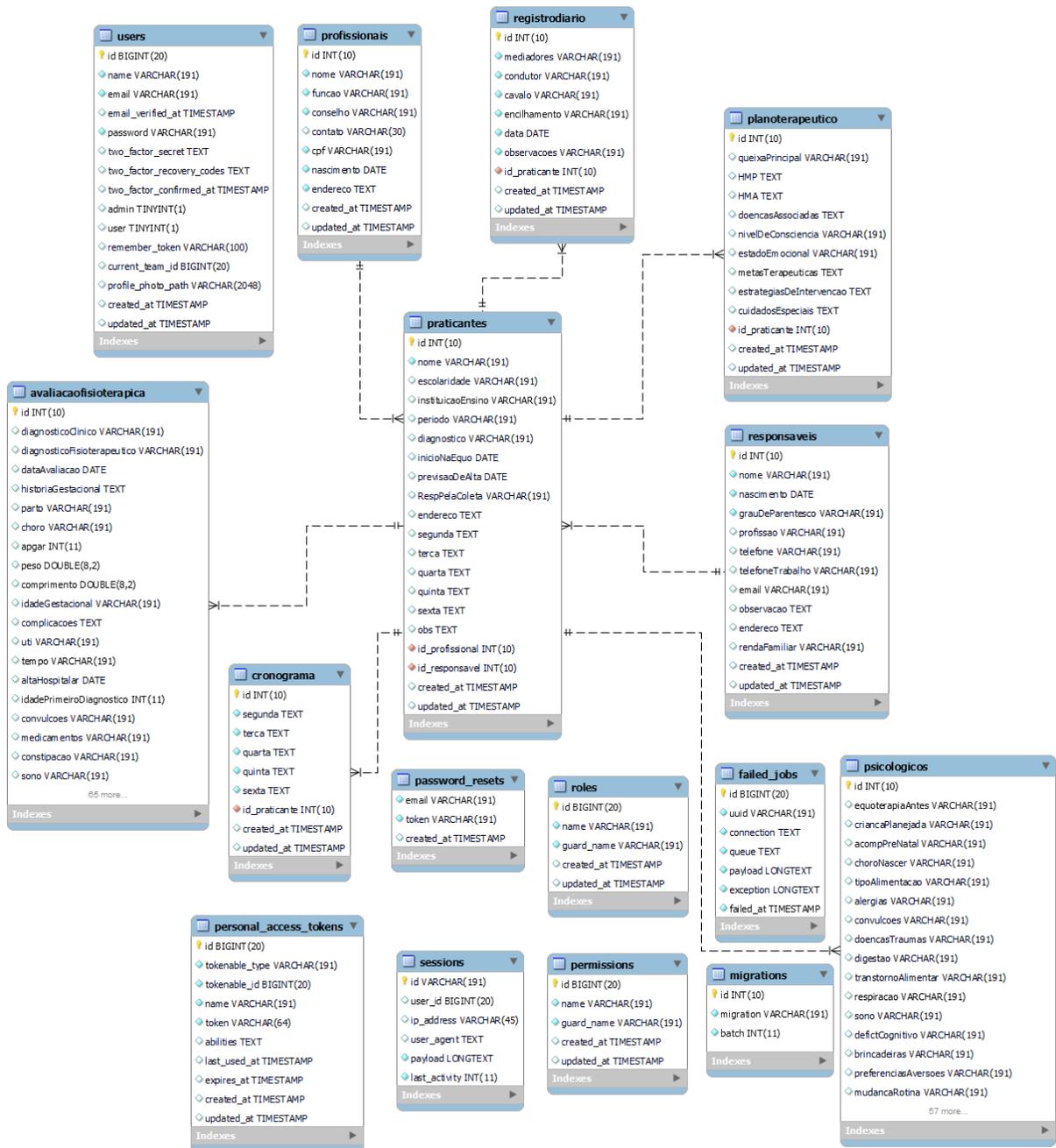


Figura 18: Diagrama físico da base de dados.

### 6.4.1 MIGRATIONS

As migrations são uma funcionalidade do Laravel 9 que possibilita o gerenciamento do esquema do banco de dados utilizando conceitos de programação e de modo versionado. Com as migrations é possível definir a estrutura do banco de dados em instruções PHP que foram executadas automaticamente no banco de dados da aplicação.

O modelo de migrations permite que as alterações no esquema do banco de dados sejam gerenciadas de forma segura e consistente, evitando conflitos e erros. Além disso, o Laravel 9 oferece comandos úteis para o gerenciamento das migrations, facilitando a criação, execução e reversão das mesmas.

## 6.5 SISTEMA DE LOGIN

O Laravel Jetstream é uma biblioteca para o framework Laravel que permite aos desenvolvedores construir aplicativos modernos e escaláveis com recursos avançados de autenticação, suporte a equipes e gerenciamento de sessão. O Jetstream é projetado para trabalhar em conjunto com tecnologias como Livewire ou Inertia, que simplificam a construção de interfaces de usuário reativas e dinâmicas.

Uma das principais vantagens do Jetstream é a CLI, que oferece uma maneira fácil e rápida de gerar recursos comuns em um aplicativo, como autenticação, gerenciamento de equipes e perfis de usuário. O Jetstream também é integrado com o Laravel Fortify, que fornece recursos avançados de autenticação, como autenticação de dois fatores, recuperação de senha e confirmação de senha.

O Jetstream inclui suporte integrado para gerenciamento de sessão, permitindo que os usuários autenticados acessem o aplicativo em várias guias e dispositivos. Essa funcionalidade torna o Jetstream uma excelente opção para aplicativos complexos com muitos usuários.

No contexto da aplicação proposta foram utilizados as tecnologias Livewire e NPM em conjunto com a biblioteca Jetstream para o desenvolvimento do sistema de login na aplicação devido à grande automação e praticidade no processo de codificação que estas tecnologias trazem quando utilizadas em conjunto com a linguagem de programação PHP.

### 6.5.1 CADASTRO DE USUÁRIOS

O cadastro de usuário é uma funcionalidade apresentada no menu lateral da aplicação e permite que sejam criados novos usuários para a aplicação fornecendo informações como: nome, e-mail e senha.

A funcionalidade de cadastro de novas contas para utilizar o sistema é restrita apenas aos usuários definidos como administradores, portanto, usuários padrões

podem manipular dados referentes aos prontuários, entretanto, não podem acessar a área de criação de novas contas no menu lateral.

Caso um usuário padrão tente acessar a funcionalidade de criação de contas o sistema realizará uma verificação daquele usuário onde o sistema por meio de uma middleware específica para verificação de níveis de acessos de contas cadastradas que verifica na base de dados o nível de permissão do usuário em questão e, então, com base no resultado desta verificação o sistema pode permitir o acesso à página de cadastro ou bloqueá-lo e redirecionar o usuário para a tela de início.

### 6.5.2 GERENCIAMENTO DE CREDENCIAIS

A aplicação proposta possui ferramentas que permitem com que usuários autenticados realizem a manutenção de suas credenciais de login em uma funcionalidade específica que requer que o usuário informe sua senha atual como medida de segurança para realizar a alteração das credenciais de login.

O sistema de login está restrito ao uso somente de endereços de e-mail como ferramenta de login, pois, os e-mails são uma forma eficiente de identificação única para cada usuário e são pré-requisitos para serem realizadas as recuperações de senha por meio da tela de login que apresenta um recurso de recuperação de senhas para usuários não autenticados.

### 6.5.3 RECUPERAÇÃO DE SENHAS

A funcionalidade de recuperação de senhas pode ser acessada por usuários não autenticados na aplicação por meio da interface de login onde o usuário terá acesso a um botão que direciona o mesmo para uma tela que solicitará ao mesmo o e-mail vinculado a sua conta. Caso o usuário forneça um e-mail que não esteja vinculado a uma conta, a aplicação retornará uma mensagem informando que nenhum usuário foi encontrado com o e-mail especificado. Caso o e-mail esteja vinculado a uma conta de usuário, então o sistema realizará o envio de uma mensagem automática para o e-mail vinculado contendo um link para o usuário conseguir alterar sua senha.

## 7. PRONTUÁRIO VIRTUAL

A aplicação proposta possui um menu de navegação posicionado à esquerda da interface que proporciona ao usuário acesso direto a todas as funcionalidades do sistema de gerenciamento de prontuários, portanto, este menu lateral é a principal ferramenta de navegação na aplicação. Conforme apresentado na figura abaixo é possível visualizar o fluxo de navegação da aplicação iniciando da tela inicial (dashboard).

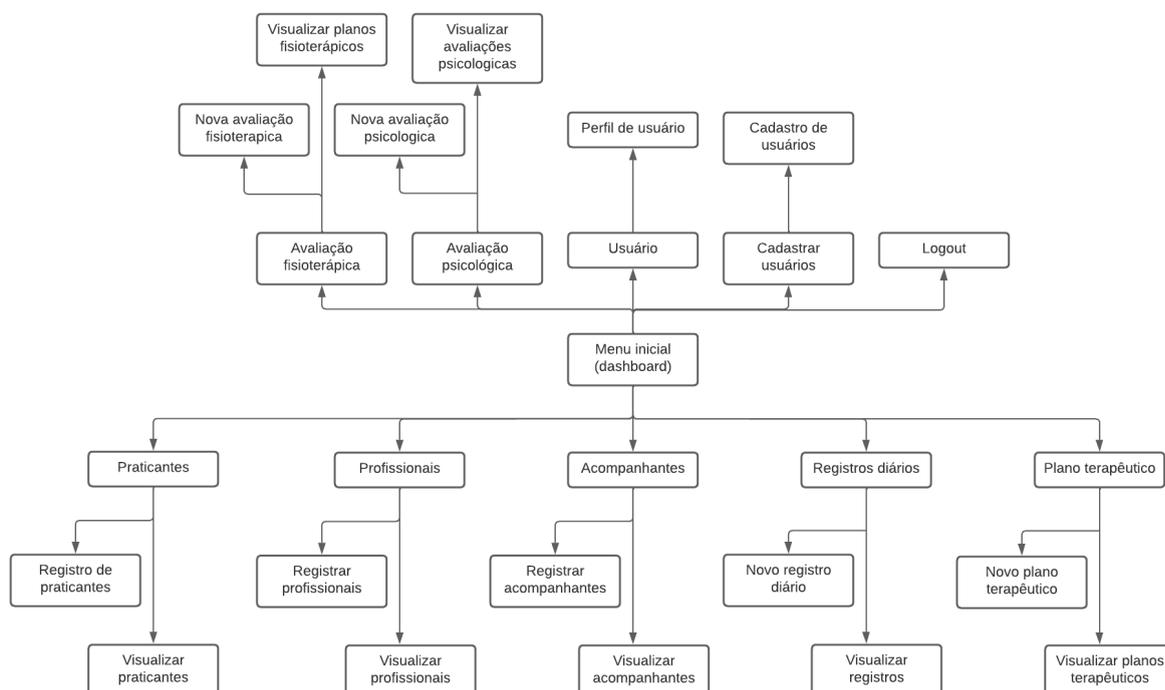


Figura 19: Fluxo de navegação

### 7.1 INTERFACES DE REGISTRO

Na figura 18 é possível observar diversos fluxos que levam a diferentes telas de registros de novas instâncias para diversas classes que compõem o prontuário eletrônico, sendo elas: registro de praticantes, profissionais, acompanhantes, registros diários, planos terapêuticos, avaliação fisioterápica e avaliação psicológica.

Cada uma destas telas apresenta as views responsáveis por realizar as requisições com os dados fornecidos pelo usuário para as suas respectivas classes de controle para que estas possam realizar a persistência dos dados que vão compor as novas instâncias das classes do prontuário eletrônico.

## 7.2 INTERFACES DE VISUALIZAÇÃO

Por meio da figura 18, os fluxos que fornecem acesso para as telas de visualização de dados apresentam suas próprias listagens de praticantes cadastrados onde cada praticante terá uma apresentação resumida de seus dados e em seguida, um menu que fornecerá ao usuário a opção de visualizar os dados dos objetos vinculados aos praticantes que são referentes a view específica que o usuário estiver acessando.

Por meio da view que fornece acesso para as instâncias vinculadas os usuários também é possível realizar a ação de atualizar os dados das instâncias vinculadas, portanto, cada instância vinculada terá seu próprio menu contendo a opção de alteração de suas informações que levará o usuário para a view de edição destas informações e realizará a comunicação com a classe de controle da instância em questão para que esta possa realizar a substituir as informações específicas na base de dados

Por fim, cada instância também terá seu próprio botão de exclusão que realizará a chamada de sua classe de controle responsável para que a mesma realize a função de excluir a instância especificada da base de dados.

## 8. TELAS DA APLICAÇÃO

Nesta seção são apresentadas as interfaces gráficas em contato direto com o usuário final para fornecer acesso às funcionalidades da aplicação.

### 8.1 TELA DE LOGIN

Conforme a figura 19 apresentada abaixo, é possível observar a interface gráfica desenvolvida para a aplicação proposta. Essa interface é encarregada de apresentar um link para recuperação de senhas para usuários não credenciados, por meio da utilização do e-mail vinculado ao usuário como meio de validação e recuperação de senhas. Além disso, nessa tela de login, são solicitadas as credenciais de acesso do usuário para realizar sua validação e garantir seu acesso às funcionalidades dispostas na aplicação.

A imagem mostra a tela de login de uma aplicação web. No topo, há um cabeçalho com o logo do "Centro de equoterapia de Ceres" (um cavalo vermelho) e o nome do instituto: "Instituto Federal Goiano - Campus Ceres". Abaixo, há dois campos de entrada: "Email" e "Senha". O campo "Email" contém um cursor de texto. Abaixo dos campos, há um link "Esqueceu sua senha?" e um botão "LOGAR" em um fundo escuro.

Figura 20: Tela de login

### 8.2 TELA DE INÍCIO

A Figura 20 apresenta a tela de início ou dashboard, sendo a interface gráfica inicial da aplicação. Nessa tela, são apresentadas informações sobre as especificações técnicas de construção da aplicação e seu contexto. Além disso, essa é a primeira tela que o usuário visualiza ao acessar a aplicação e nela é possível encontrar o menu de navegação localizado à esquerda da tela, que contém as opções de acesso a todas as funcionalidades da aplicação.

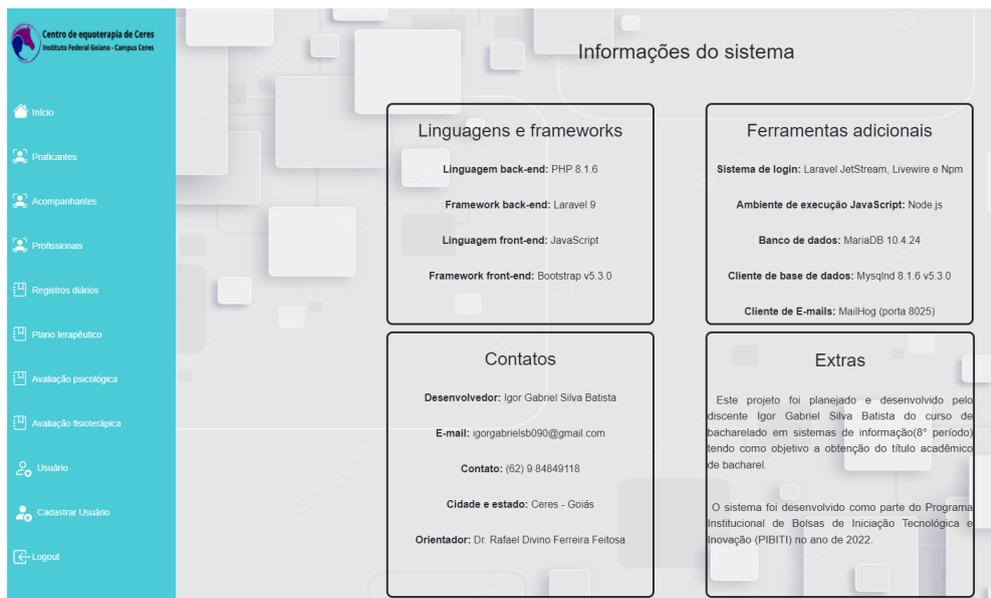


Figura 21: Tela de início

### 8.3 TELA DE REGISTRO DE PRATICANTES

A tela de registro de praticantes apresentada a seguir na Figura 21 contém a interface gráfica para registro de novos praticantes na aplicação para os profissionais poderem realizar o preenchimento de seu prontuário por meio das demais funcionalidades do software e realizar todo o gerenciamento destes dados pertinentes aos usuários devidamente cadastrados.

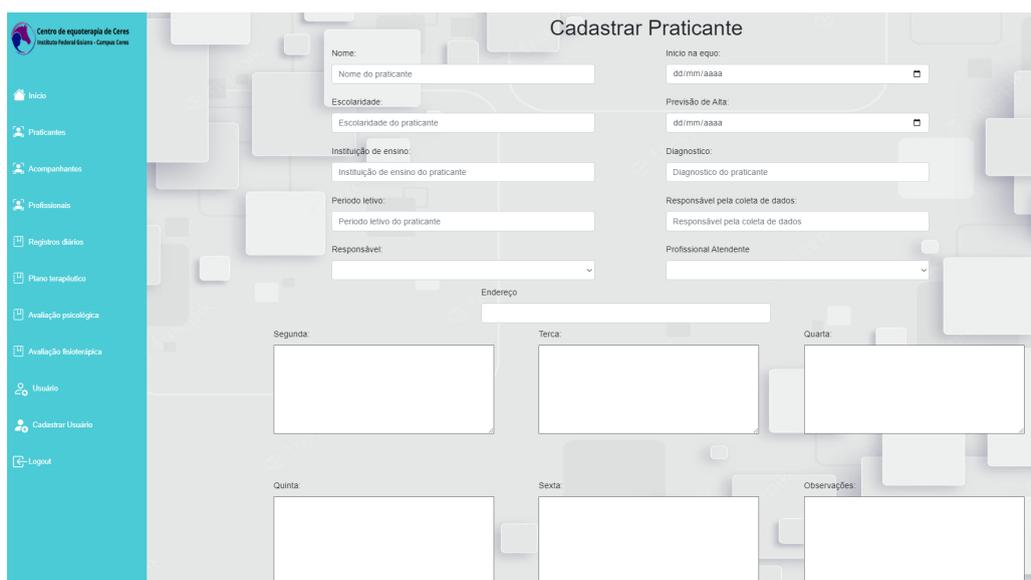


Figura 22: Tela de registro de praticantes

## 8.4 TELA DE VISUALIZAÇÃO E ALTERAÇÃO DE DADOS DE PLANO TERAPÊUTICO

A interface gráfica apresentada na Figura 22 mostra o processo de visualização e edição dos dados previamente cadastrados no plano terapêutico do praticante, logo, o usuário poderá ter uma perspectiva completa dos dados do plano terapêutico do praticante em questão e realizar alterações caso necessário.

A imagem mostra a interface de usuário para a edição de um plano terapêutico. No topo, há um cabeçalho com o logo do 'Centro de equoterapia de Ceres' e o título 'Alterar Plano terapêutico'. À esquerda, um menu lateral contém opções como 'Início', 'Praticantes', 'Acompanhantes', 'Profissionais', 'Registros diários', 'Plano terapêutico', 'Avaliação psicológica', 'Avaliação fisioterápica', 'Usuário', 'Cadastrar Usuário' e 'Logout'. O conteúdo principal é dividido em seções:

- Queixa principal:** Um campo de texto contendo 'Dificuldade de locomoção'.
- HMP:** Um campo de texto contendo 'O praticante realizou diversas sessões de fisioterapia para mobilidade, realizou intervenções cirúrgicas na região lombar e está sob acompanhamento médico'.
- Metas terapêuticas:** Um campo de texto contendo 'Intervenção e acompanhamento no quadro de mobilidade do praticante'.
- Estratégias de intervenção (Plano de ação - o que fazer? Como treinar?):** Um campo de texto contendo 'Tratamento utilizando hipnoterapia de baixa intensidade para trabalhar as respostas de movimentação do praticante e possíveis melhoras no quadro clínico geral do mesmo'.
- HMA:** Um campo de texto vazio.
- Doenças associadas:** Um campo de texto contendo 'TDAH'.
- Cuidados especiais:** Um campo de texto contendo 'Selas especiais acolchoadas para suavizar o impacto na lombar do praticante'.
- Nível de consciência atual:** Um campo de texto contendo 'Lúcido-orientado'.
- Estado atual:** Um campo de texto contendo 'Calmo'.

Na parte inferior direita, há um botão azul rotulado 'Alterar Avaliação'.

Figura 23: Tela de visualização e alteração de plano terapêutico

## 8.5 TELA DE CADASTRO DE AVALIAÇÃO FISIOTERÁPICA

Na Figura 23 a seguir é apresentada a interface de cadastro de planos terapêuticos onde são registradas as informações a respeito do histórico hospitalar do praticante, as metas que os fisioterapeutas desejam alcançar, estratégias de intervenção e outras informações pertinentes ao processo de tratamento do praticante.

Cadastrar avaliação fisioterápica

Centro de equoterapia de Ceres  
Instituto Federal Goiás - Campus Ceres

- [Início](#)
- [Praticantes](#)
- [Acompanhantes](#)
- [Profissionais](#)
- [Registros diários](#)
- [Plano terapêutico](#)
- [Avaliação psicológica](#)
- [Avaliação fisioterápica](#)
- [Usuário](#)
- [Cadastrar Usuário](#)
- [Logout](#)

**Dados do praticante**

Praticante:

**Diagnóstico clínico:**

Informe o diagnóstico clínico

**Diagnóstico terapêutico:**

Informe o diagnóstico terapêutico

**Data da avaliação:**

dd/mm/aaaa

**Queixas principais**

Insira as queixas do praticante

**História gestacional:**

Insira a história gestacional

**Período peri-natal**

**Tipo de parto:**

Informe o tipo de parto

**Tipo de choro:**

Informe o choro

**Índice APGAR:**

Informe o APGAR

**Peso:**

Informe o peso do praticante

**Período Pós-natal**

**Complicações:**

Informe a ocorrência de complicações

**UTI:**

Informe a UTI

**Tempo de UTI:**

Informe o tempo de UTI

**Data da alta hospitalar:**

dd/mm/aaaa

**Saúde geral e comunicação**

Dados	Sim	Não
Convulsões anteriores	<input type="checkbox"/>	<input type="checkbox"/>
Medicamentos	<input type="checkbox"/>	<input type="checkbox"/>
Constipação	<input type="checkbox"/>	<input type="checkbox"/>
Sono	<input type="checkbox"/>	<input type="checkbox"/>
Audição	<input type="checkbox"/>	<input type="checkbox"/>
Fluxo gastroesofágico	<input type="checkbox"/>	<input type="checkbox"/>
Intervenções cirúrgicas	<input type="checkbox"/>	<input type="checkbox"/>
Alergias	<input type="checkbox"/>	<input type="checkbox"/>

Dados	Sim	Não
Fala	<input type="checkbox"/>	<input type="checkbox"/>
Gestos	<input type="checkbox"/>	<input type="checkbox"/>
Uso dos olhos	<input type="checkbox"/>	<input type="checkbox"/>

Figura 24: Tela de cadastro de avaliação fisioterápica

## 8.6 TELA DE GERENCIAMENTO DE CREDENCIAIS DE USUÁRIO

A tela em questão apresenta a interface gráfica que fornece ao usuário diversas ferramentas para controle e gerenciamento das credenciais utilizadas para login. Nesta interface é possível alterar o nome do usuário, seu e-mail, gerar autenticação em dois fatores, desativar o usuário e visualizar as seções em andamento de seu usuário informando qual endereço IP está conectado a cada seção e as informações sobre o navegador utilizado para fins de garantir uma maior segurança ao usuário. Por meio das Figuras 24 e 25 é possível observar a interface gráfica de gerenciamento do usuário e as funcionalidades mencionadas anteriormente.



### Perfil de usuário

#### Informações de usuário

Altere suas credenciais de usuário.

Nome

admin

E-mail

adm@adm

SALVAR

#### Alterar senha

Certifique-se de escolher uma senha longa, com letras, números e caracteres maiúsculos.

Senha atual

Nova senha

Confirmar senha

SALVAR

Figura 25: Tela de controle de usuário para alteração de nome, e-mail e senha

#### Autenticação em dois fatores

Adicione um nível de segurança extra a sua conta com a autenticação em dois fatores.

Você não adicionou a autenticação em dois fatores.

Quando a autenticação de dois fatores estiver habilitada, você será solicitado a fornecer um token aleatório seguro durante a autenticação. Você pode recuperar esse token do aplicativo Google Authenticator do seu telefone.

ATIVAR

#### Sessões do navegador

Gerencie e saia de suas sessões ativas em outros navegadores e dispositivos.

Se necessário, você pode sair de todas as outras sessões do navegador em todos os seus dispositivos. Algumas de suas sessões recentes estão listadas abaixo; no entanto, esta lista pode não ser exaustiva. Se você acha que sua conta foi comprometida, você também deve atualizar sua senha.



Windows - Opera  
127.0.0.1, This device



Windows - Opera  
127.0.0.1, Last active 14 hours ago

SAIR DE OUTRAS SESSÕES DO NAVEGADOR

#### Deletar Usuário

Desativar permanentemente o usuário.

Depois que sua conta for excluída, todos os seus recursos e dados serão excluídos permanentemente. Antes de excluir sua conta, faça o download de todos os dados ou informações que deseja reter.

DELETAR USUÁRIO

Figura 26: Tela de controle de usuário para acompanhamento e segurança

## 9. CONCLUSÃO

O presente trabalho teve como objetivo projetar e desenvolver uma aplicação web, abrangendo a elaboração da interface gráfica, denominada front-end, desenvolvimento de funcionalidades que englobam as operações de cadastro, leitura, alteração e exclusão de uma ampla gama de informações, que, em conjunto, constituem um prontuário fisioterapêutico completo

A aplicação tem o intuito de suprir a demanda de um sistema que realize o gerenciamento de diversos tipos de informações sobre os praticantes que frequentam o setor de equoterapia do IF Goiano — Campus Ceres, tal qual todo o conjunto de informações sobre avaliações, registros e planos terapêuticos que formam todo o quadro clínico do praticante por meio de um prontuário eletrônico.

Para compreender e fundamentar a proposta e o desenvolvimento desta aplicação foram realizadas diversas reuniões com a gerência do setor de equoterapia, fisioterapeutas e diversos outros profissionais que atuam diretamente em todas as etapas do processo de atendimento aos praticantes de equoterapia.

Portanto, o perfil de profissionais entrevistados engloba desde profissionais especializados até estagiários e extensionistas provenientes da própria instituição e de instituições parceiras do centro de equoterapia em questão.

Avaliando as demandas espera-se que a aplicação consiga suprir a demanda por um conjunto de ferramentas tecnológicas que possam agilizar e simplificar o processo de gerenciamento de informações utilizadas no centro de equoterapia.

O gerenciamento de informações deve facilitar o atendimento de praticantes desde seu ingresso no centro de equoterapia até o momento de sua alta das atividades hipoterápicas. Tornando a atuação de todos os profissionais envolvidos mais dinâmica e eficiente.

O desenvolvimento da presente aplicação envolveu a utilização de diversas tecnologias, como Laravel 9, Bootstrap e a biblioteca JetStream. Essas tecnologias, juntamente com as linguagens PHP e JavaScript, permitiram a criação completa da aplicação. Dessa forma, este trabalho proporcionou a oportunidade de engajar em todas as fases do processo de construção de software, desde o planejamento até o desenvolvimento de um sistema plenamente funcional.

Levando em conta a premissa de desenvolvimento de uma aplicação por meio do levantamento de requisitos do setor de equoterapia do IF Goiano. O escopo inicial não engloba a implementação da aplicação em ambiente real para uso dos usuários finais. Portanto, é considerado para trabalhos futuros a aplicação em cenário de produção e posteriormente conforme resultados positivos pode-se considerar a implementação efetiva da aplicação.

É importante ressaltar que este trabalho foi realizado sem a alocação direta de recursos financeiros pela instituição e pelo setor de equoterapia, ao qual a aplicação web está vinculada. Portanto, representa uma valiosa fonte de conhecimento e experiências reais no desenvolvimento de software.

Por fim, espera-se que este trabalho se torne um legado para o IF Goiano – Campus Ceres, o setor de Equoterapia e para a comunidade em geral, proporcionando acesso gratuito e de qualidade a metodologias de tratamento alternativas.

## 10. REFERÊNCIAS

Majewski, R. L., & Oliveira, D. dos S. de. (2019). EQUOTERAPIA – A IMPORTÂNCIA DA AVALIAÇÃO DO EQUINO COMO INSTRUMENTO TERAPEUTICO. *Vivências*, 16(30),- 233–246. <https://doi.org/10.31512/vivencias.v16i30.153>

Flexa, S., Ribeiro Brandão<sup>3</sup>, P., Carmo Da Silva<sup>4</sup>, T. M., Custódio, N. M., & Sá<sup>5</sup>, M. (n.d.). *EQUOTERAPIA COMO RECURSO TERAPÊUTICO NA TERAPIA OCUPACIONAL: ELABORAÇÃO DE CARTILHA EDUCATIVA SOBRE NOÇÕES DA EQUOTERAPIA E ATUAÇÃO DO TERAPEUTA OCUPACIONAL 1.*

Cordeiro, T. L. R., Andrade, L. A. S., Santos, S. de P., & Stralhoti, K. N. de O. (2019). PRONTUÁRIO ELETRÔNICO COMO FERRAMENTA PARA A SISTEMATIZAÇÃO DA ASSISTÊNCIA DE ENFERMAGEM NO SERVIÇO DE URGÊNCIA/EMERGÊNCIA: PERCEPÇÃO DOS ENFERMEIROS. *Espaço Para a Saúde - Revista de Saúde Pública Do Paraná*, 20(2), 30–41. <https://doi.org/10.22421/15177130-2019v20n2p30>

Santos, B. R. P., & Damian, I. P. M. (n.d.). *Análise da competência em informação mediante.*

Franca, F., Aparecido, F., & Silva, S. (n.d.). *ANÁLISE TEÓRICA SOBRE O DESENVOLVIMENTO DE APLICATIVOS NATIVOS, HÍBRIDOS E WEBAPPS.* <https://www.mobiletime.com.br/previsoes/em-2022-serao-feitos-25815-bilhoes-de-downloads-de->

Pinheiro Santiago, C., de Lima Veras, N., Passos de Ara-gão, A., Albuquerque Carvalho, D., & Alves Amaral, L. (n.d.). *Capítulo 5 Desenvolvimento de sistemas Web orientado a reuso com Python, Django e Bootstrap.*

IVANOVICH, B. V. et al. Using mvc pattern in the software development to simulate production of high cylindrical steel ingots. *Journal of Crystal Growth*, v. 526, p. 125240, 2019. ISSN 0022-0248

Ferreira, B. S., & Preto, O. (2021). *UNIVERSIDADE FEDERAL DE OURO PRETO INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS DEPARTAMENTO DE COMPUTAÇÃO FRAMEWORK LARAVEL: UM ESTUDO DE CASO FULL STACK DEVELOPMENT*.

[https://sei.ufop.br/sei/controlador.php?acao=documento\\_imprimir\\_web&acao\\_origem=arvore\\_visualizar&id\\_documento=238010&infra\\_sistema=1](https://sei.ufop.br/sei/controlador.php?acao=documento_imprimir_web&acao_origem=arvore_visualizar&id_documento=238010&infra_sistema=1)

GABARDO, A. C. *Laravel para Ninjas*. 1. ed.. ed. [S.l.]: Novatec Editora Ltda, 2017. ISBN 9788575226063.

Emerson, N., Adriano, A., & Borges, M. (2021). *DESENVOLVIMENTO DE UM SISTEMA PARA CONTROLE E GERENCIAMENTO DE ESTOQUE DA EMPRESA ARTLOG SOLUTIONS LTDA* (Issue 2).

BOOTSTRAP.Sobre: visão geral. Visão geral. 2020a. Disponível em: <https://getbootstrap.com.br/docs/4.1/about/overview/>. Acesso em: 04 abr. 2020.

BOOTSTRAP.Documentação: componentes. Componentes. 2020b. Disponível em: <https://getbootstrap.com.br/docs/4.1/components/>. Acesso em: 04 abr. 2020.

Jorge, R., & Mendes, M. (n.d.). *Interoperabilidade e normalização de plataforma de serviços de big data em ambientes federados*.

SOUZA, E. C.; OLIVEIRA, M. R. de. COMPARATIVO ENTRE OS BANCOS DE DADOS MYSQL E MONGODB: quando o MongoDB é indicado para o desenvolvimento de uma aplicação. *Revista Interface Tecnológica, [S. l.]*, v. 16, n. 2, p. 38–48, 2019. DOI: 10.31510/infa.v16i2.664. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/664>. Acesso em: 12 abr. 2023.

ELMASRI, R.; NAVATHE, S. B.; PINHEIRO, M. G. et al. *Sistemas de banco de dados*. Pearson Addison Wesley São Paulo, 2005. Citado 1 vez(es) na(s) página(s) 18.

ANDRADE FILHO, Mércio Antônio Oliveira de. Projeto e desenvolvimento de uma aplicação WEB para a promoção do reuso de livros didáticos universitários. 2019. 91 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Departamento de Computação, Universidade Federal Rural de Pernambuco, Recife, 2019.

Oliveira, A. C. M. V., de Paula Silva, L. P., Gonzaga, L. P., da Cruz Carvalho, M. R., Panza, R. F., Vieira, V. G., & Tavares, J. F. (2023). Integração de Software em Ambiente Corporativo. *ANALECTA-Centro Universitário Academia*, 8(1).

FERREIRA, Brenda Sotero. Framework Laravel: um estudo de caso full stack development. 2021. 99 f. Monografia (Graduação em Ciência da Computação) - Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto, Ouro Preto, 2021.

SANTIAGO, Cynthia Pinheiro et al. Desenvolvimento de sistemas Web orientado a reuso com Python, Django e Bootstrap. **Sociedade Brasileira de Computação**, 2020.

SILVA, Ebenezer Andrade da. Catálogo de segurança para o padrão arquitetural MVC: modelagem orientada a grafos de interdependências de requisitos não funcionais. 2019. 87 f., il. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Software)—Universidade de Brasília, Brasília, 2019.

PENDER, Tom. “UML, A Bíblia”, Rio de Janeiro: Elsevier, 2004.

DA SILVA SILVA, L. A. S.; DE VARGAS FLORES, L.; VIEGAS, S. C. ASP NET MVC E SITES COMERCIAIS. REFAQI - REVISTA DE GESTÃO EDUCAÇÃO E E TECNOLOGIA, [S. l.], v. 6, n. 2, 2023. Disponível em: <https://refaqi.faqi.edu.br/index.php/refaqi/article/view/83>. Acesso em: 24 maio. 2023.

ANDRADE, Ana Paula. O que é Laravel? 2020. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-laravel>

AGNIHOTRI, Ankush. Restful API In .NET Core Using EF Core And Postgres, 2022.  
Disponível em: <https://www.c-sharpcorner.com/article/restful-api-in-net-core-using-ef-core-and-postgres/>