



Faculdade de Tecnologia de Americana “Ministro Ralph Biasi”
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Daniel Santos Veloso
Gustavo Vieira Sousa

SISTEMA DE AGENDAMENTO PARA A CLÍNICA FISIOTERAPÊUTICA SANGE

Americana, SP
2025

Daniel Santos Veloso
Gustavo Vieira Sousa

SISTEMA DE AGENDAMENTO PARA A CLÍNICA FISIOTERAPÊUTICA SANGE

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas na área de concentração em Laboratório de Engenharia de Software.

Orientador: Prof. Me. Jonas Bodê.

Este trabalho corresponde à versão final do Trabalho de Conclusão de Curso apresentado por Daniel Santos Veloso e Gustavo Vieira Sousa e orientado pelo Prof. Me. Jonas Bodê.

Americana, SP
2025

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana
Ministro Ralph Biasi- CEETEPS Dados Internacionais de
Catalogação-na-fonte

VELOSO, Daniel Santos

Sistema de agendamento para a clínica fisioterapêutica
Sange. / Daniel Santos Veloso, Gustavo Vieira Sousa – Americana,
2025.

79f.

Monografia (Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de
Americana Ministro Ralph Biasi – Centro Estadual de Educação
Tecnológica Paula Souza

Orientador: Prof. Ms. Jonas Bodê

1. Desenvolvimento de software. I. VELOSO, Daniel Santos, II.
SOUSA, Gustavo Vieira III. BODÊ, Jonas IV. Centro Estadual de
Educação Tecnológica Paula Souza – Faculdade de Tecnologia de
Americana Ministro Ralph Biasi

CDU: 681.3.05

Elaborada pelo autor por meio de sistema automático gerador de
ficha catalográfica da Fatec de Americana Ministro Ralph Biasi.

Daniel Santos Veloso
Gustavo Vieira Sousa

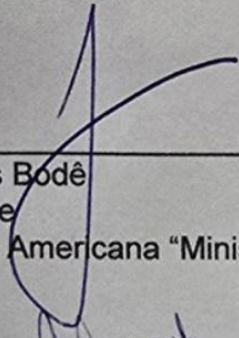
Sistema de Agendamento para a Clínica Fisioterapêutica Sange

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana Ministro Ralph Biasi.

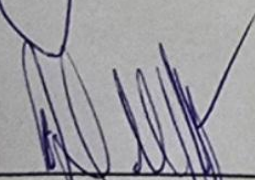
Área de concentração: Análise e Desenvolvimento de Sistemas.

Americana, 2 de dezembro de 2025.

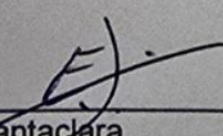
Banca Examinadora:



Jonas Bodê
Mestre
Fatec Americana "Ministro Ralph Biasi"



Renato Cividini Matthesen
Mestre
Fatec Americana "Ministro Ralph Biasi"



Evandro Santaclara
Especialista
Fatec Americana "Ministro Ralph Biasi"

“Dans la vie, rien n’est à craindre, tout est à comprendre.”

Marie Curie

(Na vida, nada deve ser temido, tudo deve ser compreendido.)

RESUMO

O presente trabalho conceitua o desenvolvimento de uma plataforma web para o gerenciamento de dados e agendamentos da Clínica Fisioterapêutica Sange, visando otimizar a organização dos dados de pacientes e automatizar as fichas para armazenamento em formato físico ou digital, agilizando-se desta forma os atendimentos do estabelecimento. Atualmente, a clínica utiliza ferramentas genéricas, como o Google Agenda, que não atendem de forma satisfatória às necessidades específicas do negócio, resultando em limitações operacionais e riscos de inconsistência nas informações. Adotou-se o estudo de caso como metodologia para o desenvolvimento, de forma a compreender as necessidades específicas dos funcionários da empresa, garantindo que o sistema seja desenvolvido de forma a atender os requisitos propostos. Com as informações obtidas através de entrevistas com o funcionário da clínica, Luiz Fernando Selles de Almeida, foram definidos os requisitos funcionais e não funcionais, além da modelagem do sistema por meio de diagramas UML, como Caso de Uso, Atividade, Sequência e Entidade-Relacionamento, tendo-se a solução desenvolvida utilizando tecnologias para programação web como PHP, React, MySQL e TypeScript. Para garantir que o desenvolvimento tenha sucesso, foi adotado o Scrum, uma metodologia ágil que permite a flexibilidade de alteração do sistema caso seja necessário. Em termos de resultados, o sistema foi projetado para agilizar e rastrear o agendamento e controle de sessões dos pacientes com base nas dificuldades encontradas dos funcionários, deste modo, o sistema atende às necessidades operacionais da clínica, facilitando o fluxo de trabalho, reduzindo erros manuais e proporcionando maior agilidade no atendimento. Desta forma, conclui-se que a aplicação contribui significativamente para a modernização da gestão da clínica fisioterapêutica Sange, trazendo maior organização, eficiência e confiabilidade às informações.

Palavras-Chave: Gerenciamento de dados; Clínica fisioterapêutica; Plataforma Web.

ABSTRACT

This paper conceptualizes the development of a web platform for data management and scheduling at the Sange Physiotherapy Clinic, aiming to optimize the organization of patient data and automate records for storage in physical or digital format, thereby streamlining the establishment's services. Currently, the clinic uses generic tools, such as Google Calendar, which do not satisfactorily meet the specific needs of the business, resulting in operational limitations and risks of inconsistency in information. The case study was adopted as the methodology for development in order to understand the specific needs of the company's employees, ensuring that the system is developed to meet the proposed requirements. With the information obtained through interviews with the clinic employee, Luiz Fernando Selles de Almeida, the functional and non-functional requirements were defined, in addition to the system modeling using UML diagrams, such as Use Case, Activity, Sequence, and Entity-Relationship, with the solution developed using web programming technologies such as PHP, React, MySQL, and TypeScript. To ensure successful development, Scrum was adopted, an agile methodology that allows for flexibility in changing the system if necessary. In terms of results, the system was designed to streamline and track the scheduling and control of patient sessions based on the difficulties encountered by employees. In this way, the system meets the operational needs of the clinic, facilitating workflow, reducing manual errors, and providing greater agility in care. Thus, it can be concluded that the application contributes significantly to the modernization of the management of the Sange physical therapy clinic, bringing greater organization, efficiency, and reliability to the information.

Keywords: *Data management; Physical therapy clinic; Web platform.*

LISTA DE ILUSTRAÇÕES

Figura 1 - Gráfico de custo de mudança no projeto ao utilizar o modelo em cascata.	18
Figura 2 - Gráfico de custo de mudanças no projeto ao aplicar metodologias ágeis.	18
Figura 3 - Exemplo de um diagrama de atividade.	24
Figura 4 - Exemplo de modelagem de caso de uso.	25
Figura 5 - Exemplo de diagrama de sequência.	26
Figura 6 - Exemplo simples de Modelo Semântico de Dados.	27
Figura 7 - Exemplo complexo de Modelo Semântico de Dados.	27
Figura 8 - Diagrama de Caso de Uso.	39
Figura 9 - Diagrama de atividade	41
Figura 10 - Diagrama de Sequência: Adicionar paciente.	42
Figura 11 - Diagrama de Sequência: Editar paciente.	43
Figura 12 - Diagrama de Sequência: Confirmar ou ausentar paciente.	44
Figura 13 - Diagrama de Sequência Impressão das informações.	45
Figura 15 - Primeiro protótipo.	46
Figura 16 - Segundo protótipo: parte 1.	47
Figura 17 - Segundo protótipo: parte 2.	48
Figura 18 - Gráfico de Burndown primeira sprint.	57
Figura 19 - Gráfico de Burndown segunda sprint.	59
Figura 20 - Gráfico de Burndown quarta sprint.	60
Figura 21 - Gráfico de Burndown terceira sprint.	62
Figura 14 - Diagrama de Entidade-Relacionamento (Apêndice A).	68
Figura 22 - Tela de agendamento do sistema.	73
Figura 23 - Tela de agendamento do sistema (tema claro).	73
Figura 24 - Tela de editar ficha do paciente.	74
Figura 25 - Comando para adicionar o usuário no sistema.	75
Figura 27 - Tela de pacientes.	76
Figura 28 - Tela do Calendário.	77

Figura 29 - PDF da ficha do paciente.....	77
Figura 30 - Filtro.	77

LISTA DE TABELAS

Tabela 1 - Requisitos Funcionais.	36
Tabela 1 - Conclusão.	37
Tabela 2 - Requisitos Não Funcionais.....	38
Tabela 3 - Dicionário de dados: Avulso.	49
Tabela 4 - Dicionário de dados: Consulta.....	49
Tabela 5 - Dicionário de dados: DiaHoraAgendado.	50
Tabela 6 - Dicionário de dados: Endereco.	51
Tabela 7 - Dicionário de dados: ExameFisico.	51
Tabela 7 - Conclusão.	52
Tabela 8 - Dicionário de dados: Paciente.....	53
Tabela 9 - Dicionário de dados: Patologia.....	54
Tabela 9 - Conclusão.	55
Tabela 10 - Dicionário de dados: Sessão.....	55
Tabela 11 - Dicionário de dados: TratamentoFisioterapico.	56
Tabela 12 - Planejamento da primeira sprint.....	57
Tabela 13 - Planejamento da segunda sprint.....	58
Tabela 14 - Planejamento da terceira sprint.....	60
Tabela 15 - Planejamento da quarta sprint.....	61

LISTA DE ABREVIATURAS E SIGLAS

UML – *Unified Modeling Language*

DOM – *Document Object Model*

CSS – *Cascading Style Sheets*

HTML – *HyperText Markup Language*

IDE – *Integrated Development Environment*

RF – Requisitos Funcionais

RNF – Requisitos Não Funcionais

PHP – *Hypertext Preprocessor*

TI – Tecnologia da Informação

MySQL – *My Structured Query Language*

PDF – *Portable Document Format*

SUMÁRIO

1 INTRODUÇÃO	13
2 REFERENCIAL TEÓRICO E TECNOLÓGICO	15
2.1 IMPACTO DOS SISTEMAS EM PEQUENAS EMPRESAS	15
2.2 IMPORTÂNCIA DAS APLICAÇÕES NA ÁREA DA SAÚDE	16
2.3 METODOLOGIAS ÁGEIS	17
2.3.1 SCRUM	19
2.4 LEVANTAMENTO DE REQUISITOS	20
2.5.1 DIAGRAMA DE ATIVIDADE	23
2.5.2 DIAGRAMA DE CASO DE USO	24
2.5.3 DIAGRAMA DE SEQUÊNCIA	25
2.5.4 DIAGRAMA ENTIDADE-RELACIONAMENTO	26
2.6 FERRAMENTAS	28
2.6.1 VISUAL STUDIO CODE	28
2.6.2 PHP	28
2.6.3 REACT	29
2.6.4 MYSQL	30
2.6.5 JAVASCRIPT	30
2.6.6 TYPESCRIPT	31
2.6.7 GITHUB	32
3 DESENVOLVIMENTO DO PROJETO	34
3.1 ESTUDO DE CASO	34
3.2 LEVANTAMENTO DE REQUISITOS	36
3.2.2 REQUISITOS NÃO FUNCIONAIS (RNF)	38
3.3 DIAGRAMAS	39
3.3.1 DIAGRAMA DE CASO DE USO	39
3.3.2 DIAGRAMA DE ATIVIDADE	40
3.3.3 DIAGRAMA DE SEQUÊNCIA	42
3.3.4 DIAGRAMA DE ENTIDADE-RELACIONAMENTO	45

3.4.1 PRIMEIRO PROTÓTIPO DE DESIGN	46
3.5 DICIONÁRIO DE DADOS	49
3.6 SPRINTS	56
3.6.1 PRIMEIRA SPRINT	56
3.6.2 SEGUNDA SPRINT	58
3.6.3 TERCEIRA SPRINT	59
3.6.4 QUARTA SPRINT	60
5 CONSIDERAÇÕES FINAIS	64
REFERÊNCIAS	65
APÊNDICE A - Imagem do Diagrama de Entidade-Relacionamento.	68
APÊNDICE B - Respostas das perguntas realizadas nas entrevistas.....	70
APÊNDICE C - Imagens do sistema.	73
APÊNDICE D - Link para github do projeto.....	78

1 INTRODUÇÃO

O progresso na tecnologia da informação tem alterado de maneira expressiva a maneira como organizações e empresas administram seus procedimentos internos, favorecendo uma maior eficácia, rapidez e segurança no manuseio de dados. No campo da saúde, essa mudança se torna ainda mais crucial, uma vez que a boa organização das informações influencia diretamente a qualidade do atendimento aos pacientes, as decisões dos profissionais e a gestão administrativa das instituições.

A clínica fisioterapêutica Sange, foco deste trabalho, utilizava métodos genéricos de organização, como o Google Agenda, para gerenciar seus atendimentos, no qual não satisfaz completamente suas necessidades. Essa limitação gerava complicações no monitoramento das sessões diárias, na visualização da agenda para os próximos dias, na categorização de pacientes e na organização dos dados dos pacientes, prejudicando a eficiência das operações na clínica.

Diante da presente situação, este projeto tem como objetivo principal a criação de uma plataforma online voltada para a administração de informações e agendamentos da clínica fisioterapêutica Sange, com o intuito de aprimorar os procedimentos administrativos e operacionais associados ao atendimento dos pacientes. Os objetivos específicos do presente trabalho são: desenvolver a funcionalidade de agendamento de consultas, permitir o gerenciamento de horários fixos e avulsos, implementar filtros de pesquisa para facilitar a localização de pacientes e possibilitar a geração de fichas de avaliação em formato PDF.

Para a execução deste projeto, adotou-se o método científico estudo de caso, fundamentado em entrevistas com o funcionário Luiz Fernando Selles de Almeida, o que possibilitou a compreensão das suas necessidades e regras de negócio. Com as informações estabelecidas, foram definidos os requisitos do sistema, além de sua modelagem e desenvolvimento por meio de tecnologias como PHP, React, MySQL e Typescript, respaldadas pela metodologia ágil Scrum.

Desta forma, este trabalho visa não só satisfazer as exigências da clínica Sange, mas também demonstrar a importância da implementação de sistemas informatizados específicos no setor da saúde, ajudando na modernização dos processos e no aprimoramento da gestão dos atendimentos fisioterapêuticos.

Este trabalho apresenta como objetivo geral a criação de uma plataforma web para a realização do gerenciamento dos dados da clínica fisioterapêutica Sange, a fim de substituir o atual sistema utilizado por ele.

Como objetivos específicos, destacam-se:

- Desenvolver a funcionalidade de agendamento para a clínica.
- Criar a função de gerenciamento de horários dos clientes para os profissionais.
- Implementar filtros de pesquisa para facilitar a visualização e organização dos dados apresentados aos usuários do sistema.
- Gerar fichas de avaliação em formato PDF, possibilitando a impressão e arquivamento das informações dos clientes de forma física.

O estudo de caso foi escolhido como método científico, tendo como foco a clínica fisioterapêutica Sange, a fim de embasar a análise da situação e desenvolvimento do projeto.

O trabalho foi estruturado em quatro capítulos, o primeiro conceituando as principais referências usadas para a construção do projeto, o segundo discute sobre as metodologias utilizadas para o desenvolvimento do sistema, o terceiro apresenta os resultados obtidos durante o desenvolvimento e, com base nas informações adquiridas a partir dos estudos realizados no capítulo anterior, o capítulo quatro se reserva às considerações finais.

2 REFERENCIAL TEÓRICO E TECNOLÓGICO

Este capítulo apresenta os principais fundamentos teóricos e tecnológicos que sustentam o desenvolvimento do sistema proposto. São abordados aspectos com relação à importância dos sistemas em empresas, tanto de forma geral quanto na área da saúde. Também são explorados os métodos utilizados para o desenvolvimento da aplicação como os diagramas UML e as ferramentas empregadas para a concepção do projeto.

2.1 IMPACTO DOS SISTEMAS EM PEQUENAS EMPRESAS

Segundo Beraldi e Escrivão Filho (2000, p. 46), “as pequenas empresas no geral não possuem sistemas informatizados, ou seja, seus controles são feitos quase que exclusivamente por meio de papeladas intermináveis.”. Mas com a queda dos valores de investimento para hardwares e o grande aumento na busca por softwares de gestão, o interesse de empresários de micro e pequenos negócios em adquirir tais tecnologias aumenta, visando a competitividade no mercado, mas se não planejado corretamente, pode acarretar em prejuízos para o negócio. (Beraldi; Escrivão Filho, 2000, p. 46-47; Lunardi; Dolci; Maçada, 2010, p. 5-6).

Assim como no homem, também na empresa o sistema de informações constitui seu sistema nervoso, responsável básico pelo bom funcionamento e desempenho. Assim, qualquer "cirurgia" realizada nessa rede nervosa provoca profundas alterações no comportamento da organização como um todo.
E é precisamente isso que ocorre quando o computador é introduzido. [...] (Mattos, 1978, p. 1).

Como descrevem Beraldi e Escrivão Filho (2000, p. 47): “Uma empresa com um sistema totalmente informatizado, funcionando eficiente e eficazmente, proporcionará grandes vantagens”. Ainda complementam os benefícios como: facilidade de obter informações, tempo otimizado, organização, previsão de dados e outros aspectos que contribuem para o sucesso da pequena empresa.

A tecnologia tem se mostrado um fator decisivo na competitividade e eficiência das indústrias. A implementação de sistemas informatizados permite automatizar os processos internos e acelerar o acesso às informações. Assim como a adoção de sistemas traz mudanças às empresas, especialmente em setores como o da saúde,

sistemas gerenciadores podem facilitar a organização de dados, o que traz diferenças na velocidade e precisão na busca de informações.

2.2 IMPORTÂNCIA DAS APLICAÇÕES NA ÁREA DA SAÚDE

Como apontam Azevedo-Marques e Salomão (2009, p. 136):

Em geral, quando se analisa o impacto da inserção da tecnologia da informação (TI) na área da saúde, observam-se três aspectos básicos: benefícios relacionados ao paciente, benefícios relacionados ao diagnóstico e benefícios relacionados ao serviço.

Complementam ainda sobre a visão dos pacientes, de benefícios ao diagnóstico e relacionados ao serviço, respectivamente:

Do ponto de vista do paciente, os ganhos possíveis estão fundamentalmente relacionados à agilização do atendimento, com uma melhora no fluxo de trabalho, maior segurança nos procedimentos e a possibilidade de humanização do atendimento, com redução de filas e tempo de espera. (Azevedo-Marques; Salomão, 2009, p. 136).

Os ganhos relacionados ao diagnóstico refletem, principalmente, as melhorias possíveis em relação à atividade dos profissionais envolvidos na cadeia de atenção à saúde e podem ser resumidos como facilidade de acesso a informações diversas e, conseqüentemente, maior suporte à tomada de decisão, e possibilidade de acesso à segunda opinião clínica e formativa, por meio de protocolos de telemedicina e telerradiologia. (Azevedo-Marques; Salomão, 2009, p. 136).

Os benefícios relacionados ao serviço (ou à instituição) estão focados principalmente em aspectos financeiros (pay-back) e englobam a possibilidade de um maior controle de procedimentos; diminuição de perdas e re-petições; diferenciação do serviço prestado; facilidade para auditoria de processos e suporte à tomada de decisão em nível de gestão. (Azevedo-Marques; Salomão, 2009, p. 136-137).

Ainda demonstra-se, na pesquisa de Godoi, Gonçalves, Peres e Wolff (2012, p. 6), a importância de um bom sistema ser instalado na área da saúde: “O que pode acontecer é que o sistema, em vez de contribuir com os processos de trabalho, toma tempo dos profissionais e desvirtua a essência do trabalho⁽⁵⁾.”.

Como é citado pelo artigo, nos discursos que demonstram as dificuldades da lentidão do sistema:

“O sistema lento acarreta muitos malefícios pro dia-a-dia do próprio prontuário”. (Enf.6). “Sabemos que a dificuldade não é só do sistema, mas também do equipamento que não tem uma capacidade de suportar, por isso é que ele fica lento. Vale ressaltar que muitas vezes a lentidão do sistema também é pelo equipamento, que já tem um certo uso. Temos uma

informática que evolui rapidamente e muda os nossos programas também”. (Enf. 3).
(Godoi; Gonçalves; Peres; Wolff, 2012, p. 7).

Como consequência da lentidão do sistema, os enfermeiros relatam que: ela ...

“faz com que a pessoa opte em examinar o paciente, orientar o paciente ou escrever. Então, se a gente for fazer uma consulta como deve ser feita, ao pensar na lentidão do sistema, a gente passa uma meia hora ou mais com cada paciente. E tem alguns pacientes, por exemplo, gestantes, que têm uma porção de coisas pra preencher pra anotar, acho que a gente demora mais ou menos isso ou uns 40 minutos. Então isso muitas vezes leva a pessoa a não fazer, a não anotar”. (Enf. 3).

(Godoi; Gonçalves; Peres; Wolff, 2012, p. 7).

Completam, Godoi, Gonçalves, Peres e Wolff (2012, p. 7): “A falta de agilidade do sistema pode comprometer o bom atendimento ao paciente, levando à insatisfação do profissional e de quem esta sendo atendido por ele.”.

Os sistemas contribuem significativamente para a otimização dos processos na área da saúde quando implementados corretamente, tendo em vista a maior eficiência nos procedimentos, segurança e facilidade de acesso às informações. Destaca-se o uso da aplicação para a clínica fisioterapêutica Sange, no qual facilita-se a disposição dos dados e a organização das informações dos clientes para.

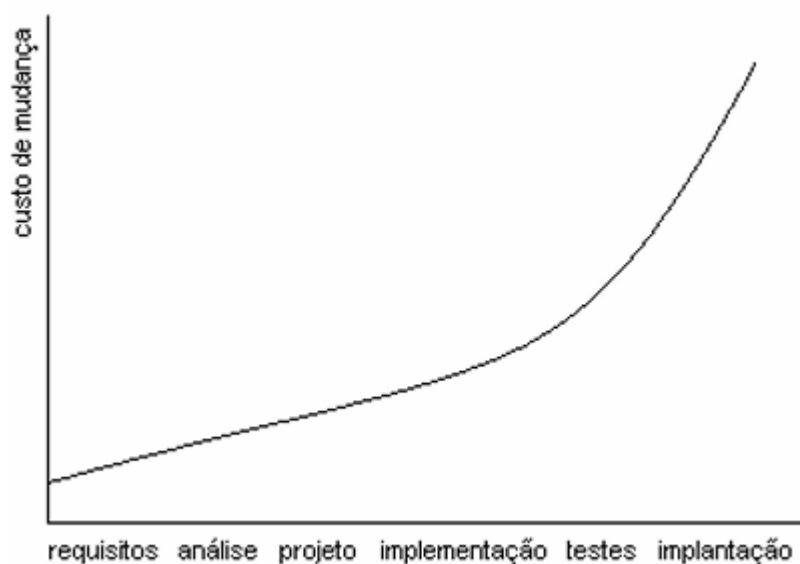
2.3 METODOLOGIAS ÁGEIS

Conforme destacado por Soares (2004, p. 1): “As metodologias ágeis para desenvolvimento de software são uma resposta às chamadas metodologias pesadas ou tradicionais.”.

Teve-se uma melhor definição das tecnologias ágeis em 2001, quando dezessete representantes das mais diversas metodologias (Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming) reuniram-se para discutir e encontrar padrões nas metodologias ágeis, o que transformou-se em o que é conhecido como *Agile Manifesto* (Manifesto Ágil), definindo assim os conceitos comuns de todas as metodologias apresentadas. (Agile Manifesto, 2001, tradução nossa).

Estas metodologias se provam efetivas quando há necessidade de mudanças no escopo do projeto que não foram previstas por metodologias tradicionais.

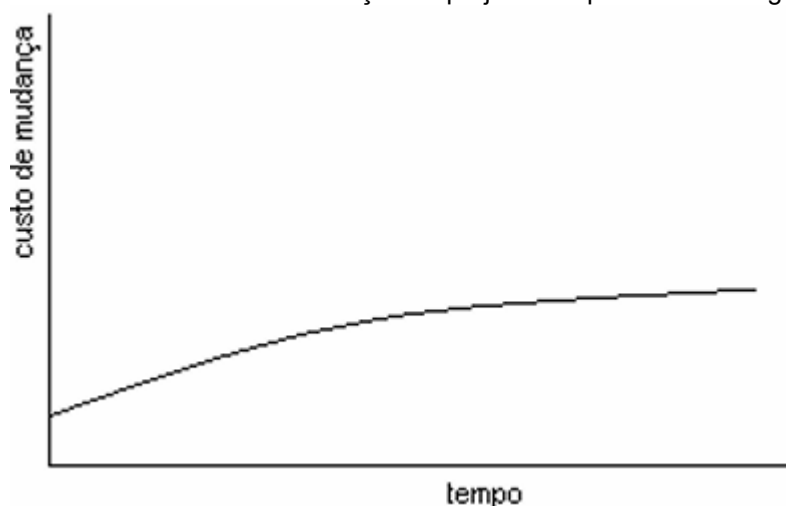
Figura 1 - Gráfico de custo de mudança no projeto ao utilizar o modelo em cascata.



Autor: Soares (2004, p. 2).

Soares (2004, p. 2) demonstra a representação dos custos da alteração dos requisitos em um software durante diversas fases do desenvolvimento ao utilizar o modelo em Cascata. Observa-se o alto impacto de mudanças durante todas as fases do projeto, tendo-se a curva de custo em seu ápice no final do projeto.

Figura 2 - Gráfico de custo de mudanças no projeto ao aplicar metodologias ágeis.



Autor: Soares (2004, p. 3).

Representa o custo de mudanças no projeto ao aplicar metodologias ágeis no desenvolvimento. Mudanças em requisitos ocorrem com frequência, de forma ideal,

os custos não devem exceder muito com alterações realizadas em fases avançadas do projeto. (Soares, 2004, p. 3).

A aplicação de metodologias ágeis demonstra-se de grande importância para as etapas de desenvolvimento dos projetos, garantindo-se a eficiência da construção do projeto em caso de alterações futuras. Ao compreender a importância da utilização de metodologias para o desenvolvimento de aplicações, destaca-se o Scrum, metodologia ágil a ser utilizada para o desenvolvimento da aplicação para a clínica fisioterapêutica Sange.

2.3.1 SCRUM

“O Scrum não define uma técnica específica para o desenvolvimento de software durante a etapa de implementação, ele se concentra em descrever como os membros da equipe devem trabalhar para produzir um sistema flexível, num ambiente de mudanças constantes.” (Franco, 2007, p. 19).

A idéia central do Scrum é que o desenvolvimento de sistemas envolve diversas variáveis (ambientais e técnicas) e elas possuem grande probabilidade de mudar durante a execução do projeto (por exemplo: requisitos, prazos, recursos, tecnologias etc.). Estas características tornam o desenvolvimento do sistema de software uma tarefa complexa e imprevisível, necessitando de um processo flexível e capaz de responder às mudanças. (Franco, 2007, p. 19).

Andrade (2019, p. 38) citando Schwaber e Sutherland (2013) ressalta sobre o Scrum:

A principal característica do Scrum é a constante agregação de valor no produto final durante a fase de desenvolvimento do projeto. É voltado para a resolução de problemas adaptativos complexos. Apesar de possuir metodologia leve e de fácil compreensão, sua implementação é complexa.

Ainda citando Schwaber e Sutherland (2013), Andrade (2019, p. 39) afirma:

Seu modelo de desenvolvimento de projeto é basicamente composto por ciclos, denominados sprints. Sprint representa um conjunto de atividades que devem ser executadas. As funcionalidades a serem implementadas em cada sprint são denominadas backlog.

De acordo com Andrade (2019, p. 40): “Em uma equipe Scrum não existe a ideia de “meu trabalho” e “seu trabalho”, a equipe obtém sucesso em conjunto e falham em conjunto. Para uma equipe Scrum de alto nível, é necessário o esforço contínuo de todos os membros.”.

Andrade (2019, p. 40) ainda complementa: “É importante que cada membro da equipe com-preenda que, independentemente da situação, seja ela de sua área de conhecimento ou não, a responsa-bilidade é de todos.”.

[...] as tomadas de decisões por parte do líder da equipe são de extrema importância e influenciam diretamen-te nos métodos a serem utilizados para que os mem-bros da equipe se comprometam e “vistam a camisa do time”. A geração de compromisso não é algo que possui prazo de início ou fim. Ela deve ser constan-te, em todo momento o compromisso do trabalho em equipe deve ser reforçado. (Andrade, 2019, p. 43).

Ao implementar o sistema Scrum em projetos, a equipe mantém-se flexível às mudanças que ocorrerem durante o desenvolvimento sem comprometer o prazo e organização do projeto. Em suma, a metodologia ágil Scrum mostra-se crucial para o desenvolvimento da aplicação para a clínica fisioterapêutica Sange por permitir a flexibilidade de alterar o sistema caso seja necessário.

2.4 LEVANTAMENTO DE REQUISITOS

De acordo com Pressman (2011, p.133-145) o levantamento de requisitos é:

O levantamento de requisitos (também chamado elicitación de requisitos) combina elementos de resolução de problemas, elaboração, negociação e especificação. Para encorajar uma abordagem colaborativa e orientada às equipes em relação ao levantamento de requisitos, os interessados trabalham juntos para identificar o problema, propor elementos da solução, negociar diferentes abordagens e especificar um conjunto preliminar de requisitos da solução(Pressman, 2011, p.133).

Como explica Sommerville (2011, p. 73), essa etapa do desenvolvimento requer um contato próximo com o usuário por meio de entrevistas que incentivem sua colaboração e a expressão de opiniões. Nessas reuniões, os entrevistadores devem se livrar de qualquer pré-concepção e ouvir atentamente o entrevistado. Após essa coleta inicial, segundo Sommerville (2011, p. 59), é necessário diferenciar os requisitos obtidos entre funcionais e não funcionais. Os requisitos funcionais expressam ações que o sistema deve realizar e são normalmente apresentados pelos usuários de forma abstrata, por meio de verbos no infinitivo, como “cadastrar” ou “realizar determinada tarefa”. Já os requisitos não funcionais, conforme Sommerville (2011, p. 60-61), referem-se a demandas que não alteram a funcionalidade do sistema, mas representam preferências dos usuários, como escolhas estéticas ou relacionadas a ferramentas.

Esse levantamento é essencial para os projetos, pois, como afirma Sommerville

(2011, p. 57), ele ajuda a definir os objetivos de forma abstrata e garante que as soluções para esses requisitos não sejam pré-definidas, eliminando a possibilidade de construir aplicações que não correspondam às necessidades e expectativas dos usuários.

2.5 DIAGRAMAS UML

Como afirma Sommerville (2011, p. 82): “Modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema.”. Representa-se o sistema em formato de notações gráficas, baseando-se atualmente em notações de UML (linguagem de modelagem unificada, do inglês Unified Modeling Language). (Sommerville, 2011, p. 82).

Os modelos são usados durante o processo de engenharia de requisitos para ajudar a extrair os requisitos do sistema; durante o processo de projeto, são usados para descrever o sistema para os engenheiros que o implementam; e, após isso, são usados para documentar a estrutura e a operação do sistema. (Sommerville, 2011, p. 82).

Segundo Esmín (2004, p. 1): “A UML (Unified Modeling Language) é o sucessor de um conjunto de métodos de análise e projeto orientados a objeto.”.

A UML é um modelo de linguagem, não um método. Um método pressupõe um modelo de linguagem e um processo. O modelo de linguagem é a notação que o método usa para descrever o projeto. O processo são os passos que devem ser seguidos para se construir o projeto. (Esmín, 2004, p. 1).

De acordo com Esmín (2004, p. 2):

Para cada uma das fases do desenvolvimento utilizam-se em seu desenvolvimento cinco tipos de visões, nove tipos de diagramas e vários modelos de elementos que serão utilizados na criação dos diagramas e mecanismos gerais que todos em conjunto especificam e exemplificam a definição do sistema, tanto a definição no que diz respeito à funcionalidade estática dinâmica do desenvolvimento de um sistema.

Esmín (2004, p. 2), complementa detalhando as visões, mecanismos gerais e diagramas:

- Visões: As Visões mostram diferentes aspectos do sistema que está sendo modelado. A visão não é um gráfico, mas uma abstração consistindo em uma série de diagramas.
- Modelos de Elementos: Os conceitos usados nos diagramas são modelos de elementos que representam definições comuns da orientação a objetos como as classes, objetos, mensagem, relacionamentos entre classes incluindo associações, dependências e heranças.
- Mecanismos Gerais: Os mecanismos gerais provêm comentários suplementares,

informações, ou semântica sobre os elementos que compõem os modelos.

- Diagramas: Os diagramas são os gráficos que descrevem o conteúdo em uma visão. UML possui nove tipos de diagramas que são usados em combinação para prover todas as visões do sistema.

As visões, como descritas por Esmín (2004, p. 2-3), que são compostas por um sistema são:

- Visão “use-case”: Descreve a funcionalidade do sistema desempenhada pelos atores externos do sistema (usuários).
- Visão Lógica: Descreve como a funcionalidade do sistema será implementada. É feita principalmente pelos analistas e desenvolvedores. Em contraste com a visão use-case, a visão lógica observa e estuda o sistema internamente. Ela descreve e especifica a estrutura estática do sistema (classes, objetos, e relacionamentos) e as colaborações dinâmicas quando os objetos enviarem mensagens uns para os outros para realizarem as funções do sistema.
- Visão de Componentes: É uma descrição da implementação dos módulos e suas dependências. É principalmente executado por desenvolvedores, e consiste nos componentes dos diagramas.
- Visão de concorrência: Trata a divisão do sistema em processos e processadores. Este aspecto, que é uma propriedade não funcional do sistema, permite uma melhor utilização do ambiente onde o sistema se encontrará, se o mesmo possui execuções paralelas, e se existe dentro do sistema um gerenciamento de eventos assíncronos.
- Visão de Organização: Finalmente, a visão de organização mostra a organização física do sistema, os computadores, os periféricos e como eles se conectam entre si.

Sommerville (2011, p. 83), comenta sobre cinco tipos de diagramas principais, com base em uma pesquisa de 2007 de Erickson e Siau, que podem construir a natureza de um software:

1. Diagramas de atividades, que mostram as atividades envolvidas em um processo ou no processamento de dados.
2. Diagramas de casos de uso, que mostram as interações entre um sistema e seu ambiente.
3. Diagramas de sequência, que mostram as interações entre os atores e o sistema, e entre os componentes do sistema.
4. Diagramas de classe, que mostram as classes de objeto no sistema e as associações entre elas.
5. Diagramas de estado, que mostram como o sistema reage aos eventos internos e externos.

Ao definir os principais diagramas UML na construção de sistemas, deve-se ter clareza sobre como as notações gráficas serão utilizadas, pois mostram-se flexíveis para modelamentos de projetos, tendo-se diferentes modos de aplicá-los. (Sommerville, 2011, p. 83).

Os principais diagramas selecionados para o desenvolvimento do software da clínica fisioterapêutica Sange são importantes para a compreensão e construção do

sistema, tendo-se o Diagrama de Atividade, o Diagrama de Caso de Uso, o Diagrama de Sequência e o Diagrama de Entidade-Relacionamento.

2.5.1 DIAGRAMA DE ATIVIDADE

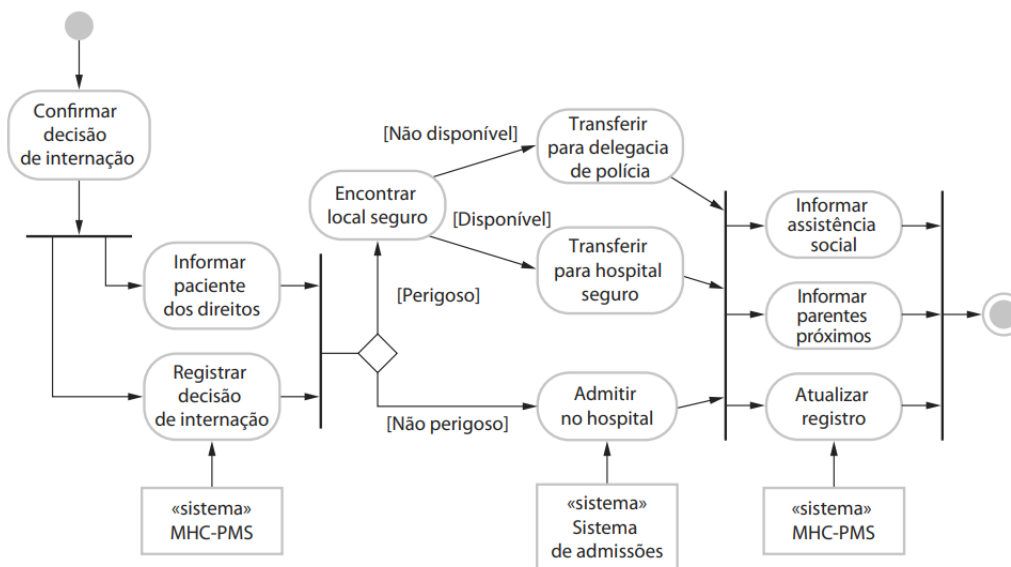
Ao iniciar o processo de especificação do sistema, deve-se impor limites no software. De acordo com Sommerville (2011, p. 84): “Isso envolve trabalhar com os stakeholders do sistema para decidir qual funcionalidade deve ser incluída no sistema e o que é fornecido pelo ambiente do sistema.”. Para evitar maiores custos e tempo para entender os requisitos, deve-se acertar as decisões de limite logo no começo do processo.

Depois de tomadas algumas decisões a respeito dos limites do sistema, parte da atividade de análise é a definição desse contexto e das dependências que o sistema tem em seu ambiente. Normalmente, a produção de um modelo de arquitetura simples é o primeiro passo para essa atividade. (Sommerville, 2011, p. 84).

Sommerville (2011, p. 85) comenta: “Os modelos de contexto simples são usados com outros modelos, como modelos de processos de negócio. Estes descrevem os processos humanos e automatizados em que os sistemas de software específicos são usados.”.

Segundo Sommerville (2011, p. 85): “Os diagramas de atividades são destinados a mostrar as atividades que compõem um processo de sistema e o fluxo de controle de uma atividade para a outra.”

Figura 3 - Exemplo de um diagrama de atividade.

Figura 5.2 Modelo de processos de internação involuntária

Autor: Sommerville (2011, p. 85).

O diagrama de atividade, segundo Sommerville (2011, p. 85), se constrói de tal forma:

O início de um processo é indicado por um círculo preenchido; o fim, por um círculo preenchido dentro de outro círculo. Os retângulos com cantos arredondados representam atividades, ou seja, os subprocessos específicos que devem ser realizados. Você pode incluir objetos em diagramas de atividades.

Sommerville (2011, p. 85) complementa sua visão:

[...] as setas representam o fluxo de trabalho de uma atividade para outra. Uma barra sólida é usada para indicar coordenação de atividades. Quando o fluxo de mais de uma atividade leva a uma barra sólida, todas essas atividades devem ser concluídas antes de o progresso ser possível. Quando o fluxo de uma barra sólida leva a uma série de atividades, elas podem ser executadas em paralelo.

2.5.2 DIAGRAMA DE CASO DE USO

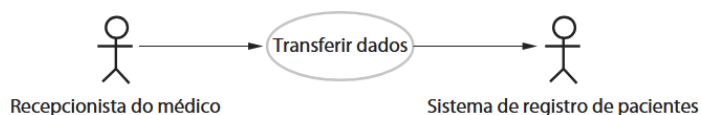
Sommerville (2011, p. 86) afirma que todos os sistemas possuem interações: “Pode-se ter interação do usuário, que envolve entradas e saídas, interação entre o sistema que está em desenvolvimento e outros sistemas, ou interação entre os componentes do sistema.”. Complementa sua afirmação ao dizer que a modelagem da interação do usuário possui grande importância para a entender os requisitos de usuários.

Para modelar as interações dos atores (usuários e outros sistemas) e o sistema principal, utiliza-se a modelagem de caso de uso. “Um caso de uso pode ser tomado como um cenário simples que descreve o que o usuário espera de um sistema.” (Sommerville, 2011, p. 86).

Figura 4 - Exemplo de modelagem de caso de uso.

Figura 5.3

Caso de uso de transferência de dados.



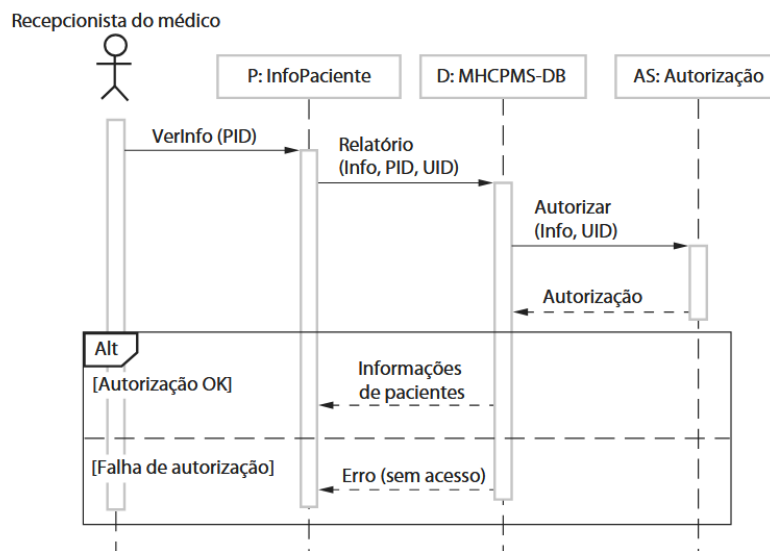
Autor: Sommerville (2011, p. 86).

Demonstra-se o caso de uso com uma forma elíptica, ao qual representa uma atividade que abrange as interações no sistema. Dois atores estão presentes na Figura 3, estes, representados pela notação figura-palito, demonstram a interação humana (ou de outros sistemas externos) no software. As setas representam a direção do fluxo de mensagens, indicando-se como as interações devem se comportar no sistema. (Sommerville, 2011, p. 86).

2.5.3 DIAGRAMA DE SEQUÊNCIA

Como comenta SommerVille (2011, p. 87): “Os diagramas de sequência em UML são usados, principalmente, para modelar as interações entre os atores e os objetos em um sistema e as interações entre os próprios objetos.”. O diagrama de sequência mostra a ordem das interações que decorrem durante instâncias do caso de uso ou um em particular.

Figura 5 - Exemplo de diagrama de sequência.

Figura 5.5 Diagrama de sequência para 'Ver informações de pacientes'

Autor: Sommerville (2011, p. 88).

Explica-se o diagrama de sequência, segundo Sommerville (2011, p. 87):

Os objetos e atores envolvidos estão listados na parte superior do diagrama, com uma linha tracejada verticalmente a partir deles. Interações entre objetos são indicadas por setas anotadas. O retângulo na linha tracejada indica a linha da vida do objeto em questão (ou seja, o tempo em que a instância do objeto está envolvida no processamento). Deve-se ler a sequência de interações de cima para baixo. As anotações sobre as setas indicam as chamadas para os objetos, seus parâmetros e os valores de retorno. (Sommerville, 2011, p. 87).

O diagrama de sequência não requer todas que as interações do sistema apareçam a menos que seja utilizado para geração de códigos. Algumas interações não possuem importância para a decisão do sistema, sendo-se desnecessária a inclusão da ação no diagrama. (Sommerville, 2011, p. 89)

2.5.4 DIAGRAMA ENTIDADE-RELACIONAMENTO

Diagramas de entidade-relacionamento são mencionados por Sommerville (2011) em seu livro "Engenharia de Software" como modelos semânticos de dados. Este comenta que usa-se os modelos semânticos para o desenvolvimento de projetos de banco de dados, mostrando-se informações como a entidade, os atributos de cada uma e as relações entre estas entidades.

A UML não inclui uma notação específica para essa modelagem de banco de dados, pois supõe um processo de desenvolvimento orientado a objetos e modela dados usando objetos e seus relacionamentos. No entanto, pode-se usar a UML para representar um modelo semântico de dados. Sob um modelo semântico de dados, podemos pensar em entidades como classes de objetos simplificados (não possuem operações), em atributos como atributos da classe de objetos e em relações como as associações nomeadas entre classes de objetos. (Sommerville, 2011, p. 90).

Figura 6 - Exemplo simples de Modelo Semântico de Dados.

Figura 5.7 Classes e associação em UML

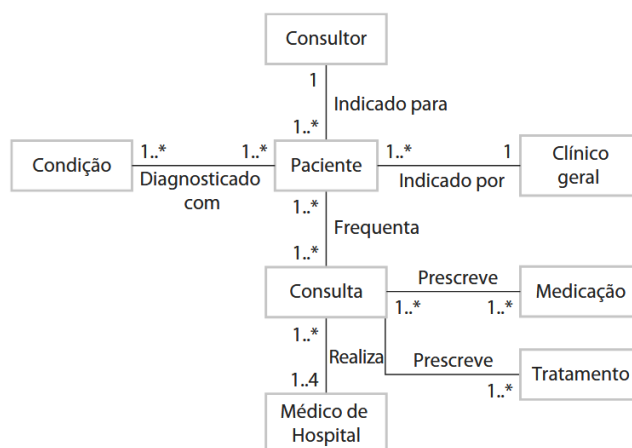


Autor: Sommerville (2011, p. 90).

Exemplo simplificado de um modelo semântico de dados mostrando a relação 1:1 (um para um) entre a entidade Paciente e Registro de paciente.

Figura 7 - Exemplo complexo de Modelo Semântico de Dados.

Figura 5.8 Classes e associações no MHC-PMS



Autor: Sommerville (2011, p. 90).

Segundo Sommerville (2011, p. 91): “Ao mostrar as associações entre classes, é conveniente representar essas classes da maneira mais simples possível.”. Definindo-as com detalhes, adiciona-se informações de atributos (as características do objeto) e operações (o que o objeto pode fazer). “Por exemplo, um objeto Paciente terá o atributo Endereço, e você pode incluir uma operação chamada MudarEndereço, que é chamada quando um paciente indica ter mudado de endereço.” (Sommerville, 2011, p. 91).

2.6 FERRAMENTAS

O presente tópico discute sobre as ferramentas a serem utilizadas para o desenvolvimento do sistema, adotando-se critérios como disponibilidade de recursos, facilidade de manuseio e eficiência das plataformas escolhidas.

2.6.1 VISUAL STUDIO CODE

Optou-se o Visual Studio Code como IDE para este trabalho pois como pontua Sole (2023) essa ferramenta proporciona um ambiente de desenvolvimento completo cheio de possibilidades e extensões para suprir demandas do projeto como análise de códigos em tempo real, integração fácil com Git e compiladores de várias linguagens.

Outras vantagens de se utilizar o Visual Studio Code são:

O Visual Studio Code é baseado em arquivos e pastas, permitindo trabalhar tanto com arquivos individuais quanto com pastas que contêm arquivos de código-fonte. Ele trata todos de forma estruturada e organizada. O Visual Studio Code também oferece suporte a diversos sistemas de projeto, como .NET, TypeScript e JavaScript, além de permitir a criação e o gerenciamento de *workspaces* com múltiplas raízes. *Workspaces* são contêineres lógicos de pastas que facilitam ter vários projetos e pastas sob a mesma raiz visual. (Del Sole, 2023, p.101, tradução nossa).

Em conclusão, Del Sole (2023) ressalta que essa ferramenta proporciona ampla liberdade ao desenvolvedor, uma vez que, por ser totalmente personalizável, pode se adaptar a diferentes tipos de projeto e escopos, integrando-se facilmente à maioria dos serviços web contemporâneos.

2.6.2 PHP

O php foi a linguagem de programação principal escolhida para o projeto pois, de acordo com Park (2003, p. 27): “[...] tem o código-fonte aberto, é repleto de recursos, serve para diversas plataformas, é estável, rápido, claramente projetado, fácil de aprender e funciona bem com outros produtos.”.

Essa versatilidade e facilidade pode ser de grande conveniência ao desenvolvimento de páginas web já que “O PHP é uma linguagem totalmente voltada à internet que possibilita o desenvolvimento de sites realmente dinâmicos. Dominando essa linguagem, pode-se transformar aqueles sites estáticos, feitos de HTML puro,

em sites interativos, utilizando todas as técnicas de programação que essa linguagem oferece” (Niederauer, 2017, p. 12).

2.6.3 REACT

Chen, Thaduri e Ballamudi (2019, p. 119) explicitam que o React é a tecnologia dominante no cenário das tecnologias front-end, especialmente no desenvolvimento web, devido à sua estrutura que integra HTML, CSS e JavaScript, possibilitando o trabalho com componentes reutilizáveis em vez de arquivos dispersos pela aplicação.

O foco principal do React, segundo Komperia et al. (2022), está na renderização de componentes e interfaces de usuário não apenas no cliente do usuário, mas também no servidor ou DOM, característica que está associada ao conceito de Single-Page Application. Essa abordagem permite a construção de aplicações de forma dinâmica e ágil, reduzindo significativamente o atraso no carregamento de páginas ou na execução de funções. Além disso, possibilita a atualização em tempo real da interface do usuário, uma vez que as informações são carregadas e enviadas pela DOM conforme necessário.

De acordo com Komperia et al. (2022, p. 1712), o que torna o React uma biblioteca realmente notável é sua capacidade de combinar código HTML com objetos JavaScript, o que permite escrever estruturas HTML diretamente dentro do JavaScript. Essa integração facilita a implementação de mudanças na interface, como o surgimento de novos campos na tela ou imagens dinâmicas, e ainda contribui para a segurança da aplicação, prevenindo certos tipos de ataques, como o cross-site scripting.

Outra ferramenta essencial do React, conforme destacam Komperia et al. (2022, p. 1713), é a Virtual DOM, que funciona como uma cópia da DOM original.

Quando uma alteração é realizada, o React cria uma representação da DOM com as mudanças e a compara com a Virtual DOM para identificar o que realmente foi modificado. Em seguida, apenas essas alterações necessárias são aplicadas à DOM real, evitando a recriação completa da página a cada atualização.

Por fim, como dito por Chen, Thaduri e Ballamudi (2019, p.117, tradução nossa): "Em desenvolvimento front-end, a função que o react.js desempenha está se

tornando cada vez mais importante, fornecendo aos desenvolvedores novas opções para criar aplicações.”

2.6.4 MYSQL

Como diz Bento (2013) o Mysql é um software livre para gerenciamento banco de dados, que de forma conceitual organiza os dados em tabelas, sendo assim um banco de dados relacional pois depende da relação destas tabelas para ter uma completude de sentido.

O uso deste banco de dados demonstra-se muito oportuno para este projeto pois:

PHP e MySQL já são velhos amigos. É bem comum encontrar aplicações que fazem uso destas tecnologias em conjunto. Desde pequenos sites pessoais, até grandes sistemas de gestão e lojas virtuais, a web está repleta de casos de sucesso desta parceria.(Bento, Evaldo Junior, 2013, p.65).

Porém, além da parceria entre PHP e MySQL, existem outras vantagens em utilizar o MySQL, como elucida Bento (2013). Entre elas, destacam-se a performance pois o sistema é consideravelmente leve e suporta grandes volumes de dados, a simplicidade de uso e a fácil integração com servidores web. Além disso, como comprovado por Taipalus (2023), o MySQL é notável por ter implementado apenas uma operação de consulta física para a junção de tabelas, o que otimiza a quantidade de querrys necessárias para seu uso, porém também o limita em achar caminhos mais rápidos de consultas.

Em conclusão seguindo as ideias de Bento (2013) o Mysql é um gerenciador de banco de dados leve, de fácil uso e com uma grande parceria com o PHP, uma das maiores linguagens da web, sendo assim prático e extremamente funcional para aplicações web.

2.6.5 JAVASCRIPT

Mariano (2017), o javascript é uma linguagem de scripting e programação que existe há muito tempo, ele é uma das linguagens mais conhecidas e utilizadas no contexto de desenvolvimento web.

Sendo uma das bases do desenvolvimento web (Biffi, 2018), essa linguagem está presente na grande maioria dos sites que acessamos diariamente. Isso ocorre porque ela pode ser utilizada em diversos contextos dentro de uma aplicação. Entre suas principais vantagens estão a alta performance já que não consome memória do computador de forma excessiva, a característica multiplataforma, que possibilita seu uso em diferentes vertentes da programação, e o fato de ser uma linguagem multiparadigma, o que significa que pode ser empregada em vários modelos de programação, como a programação imperativa e a orientada a objetos, entre outras. Por fim, como afirmam Jang e Choe (2009), destaca-se também a possibilidade de modificar objetos em tempo de execução por meio de propriedades ou pela atualização de métodos.

Em conclusão, seguindo as ideias de Biffi (2018), essa linguagem dispõe de diversas ferramentas aplicáveis ao contexto da web e apresenta um desempenho notável. Além disso, é bastante versátil, podendo ser utilizada em diferentes áreas do desenvolvimento de uma aplicação, como o back-end e o front-end.

2.6.6 TYPESCRIPT

Bogner e Merkel (2022, p. 1) esclarecem que o TypeScript é uma linguagem amplamente utilizada e relevante para o desenvolvimento de software moderno, sendo um superconjunto do JavaScript que oferece segurança em termos de tipos. Além disso, Cherny (2019, p. 1) sugere que o TypeScript vem se consolidando como a base das novas gerações de aplicações web, soluções mobile, projetos em Node.js e até dispositivos da Internet das Coisas. A linguagem contribui para tornar o código mais seguro ao detectar erros comuns, funciona como uma forma de documentação para o desenvolvedor e para equipes futuras, facilita processos de refatoração e reduz significativamente a necessidade de determinados testes.

Uma de suas vantagens, como destaca Cherny (2019, p. 7), é sua tipagem gradual, que permite adicionar anotações de tipos explicitamente ou deixá-las a cargo da inferência do próprio TypeScript. Outras vantagens, como indicado no estudo de Bogner e Merkel (2022), incluem o fato de que aplicações em TypeScript apresentam qualidade de código e compreensão significativamente maiores que aplicações escritas apenas em JavaScript. Além disso, sua capacidade de detectar erros

precocemente contribui para a prevenção de falhas. Os autores também apontam que o TypeScript possui alta manutenibilidade, pois sua tipagem funciona como uma forma de autodocumentação do código, facilitando que programadores futuros compreendam seu fluxo e solucionem problemas.

Em suma, segundo Bogner e Merkel (2022), o TypeScript é um superconjunto que torna programas mais seguros e mais fáceis de compreender, sendo uma linguagem ideal para projetos que precisam escalar em complexidade e tamanho.

2.6.7 GITHUB

Como afirmam Zagalsky et al. (2015), o GitHub é uma plataforma amplamente reconhecida pela comunidade de desenvolvedores de software como uma ferramenta fundamental para qualquer projeto. Essa plataforma funciona como um repositório de arquivos e um versionador de código, permitindo que os usuários tenham acesso a todas as versões da aplicação. Além disso, o GitHub organiza o projeto em branches, criando um ambiente seguro para testes, uma vez que essas ramificações podem ser utilizadas para experimentar novas funcionalidades antes que sejam incorporadas ao programa principal.

O GitHub também traz consigo uma transparência muito maior aos projetos (Zagalsky et al., 2015), pois oferece diversos recursos que permitem que os colaboradores acompanhem as atividades de todos os membros. Um desses recursos é o pull request, que inicia uma discussão entre os participantes do projeto para avaliar a integração de novas funcionalidades ao código principal, dependendo da aprovação da maioria dos colaboradores ou da autorização do líder do projeto previamente designado.

Em adição como afirma Munaiah et al. (2017) estes repositórios de software contém uma riqueza de informações, pois há uma proliferação de repositórios de código-fonte aberto fornece para outros usuários do Github uma abundância de dados tanto para serem utilizados para pesquisas quanto como inspiração ou até objeto de estudo.

Em consonância com as ideias de Zagalsky et al. (2015), o GitHub configura-se como uma plataforma altamente eficiente no gerenciamento e versionamento de códigos, sendo praticamente indispensável tanto para desenvolvedores quanto para

estudantes da área de tecnologia. Ademais, a plataforma atua como uma vitrine de projetos, promovendo o compartilhamento e ampliando a visibilidade das produções desenvolvidas pelos usuários.

3 DESENVOLVIMENTO DO PROJETO

O tópico a seguir retrata sobre as metodologias aplicadas para o desenvolvimento do projeto como a abordagem do método de coleta de dados, o levantamento de requisitos, a criação dos diagramas do sistema, o dicionário de dados e o progresso das Sprints da metodologia ágil Scrum.

3.1 ESTUDO DE CASO

Foi desenvolvido o estudo de caso para compreender as principais necessidades da clínica fisioterapêutica Sange com relação ao atual sistema utilizado (Google Agenda) e entender como o desenvolvimento do projeto deve prosseguir para ajudá-los.

A coleta de dados foi realizada através de 2 (duas) entrevistas com o funcionário da clínica Luiz Fernando Selles de Almeida via comunicação à distância, tendo-se a conversa gravada com a permissão do funcionário para análise. Montou-se as perguntas a seguir de forma personalizada para este documento de acordo com os dados mais relevantes retirados das entrevistas de forma a sintetizar as informações para melhor compreensão.

As respostas para cada pergunta encontram-se registradas no Apêndice B.

Pergunta 1: Quais são as funções primárias necessárias para o agendamento e controle de sessões dos pacientes?

Pergunta essencial para entender o caminho que o projeto deve tomar, garantindo-se que o sistema atenda às necessidades dos funcionários da clínica.

Pergunta 2: Como o sistema deve lidar com a flexibilidade de horários e pacientes avulsos (que vem em horários fora do que foi)?

Pergunta que surgiu durante a conversa de como funcionava para a clínica cuidar dos dados dos pacientes. Foi discutida uma regra de funcionalidade do sistema.

Pergunta 3: Quais são as regras de disponibilidade para agendamentos?

Pergunta com intuito de responder sobre regras de funcionamento da agenda, reforçando como funciona na clínica e o que deve ser realizado no sistema.

Pergunta 4: Que tipos de pacientes o sistema deve diferenciar?

Na clínica, existem 3 (três) tipos de pacientes: normais, especiais e avaliação. Foi especificado como o sistema deveria tratar estes pacientes.

Pergunta 5: Quais informações devem ser registradas no cadastro e na avaliação do cliente?

Foi especificado como o sistema deve se comportar e quais informações deve armazenar no registro de novos pacientes.

Pergunta 6: Como deve ser a visualização da agenda diária e do calendário?

Foi discutido como a parte visual do sistema deveria se comportar na agenda e no calendário.

Pergunta 7: Quais filtros e acessos são necessários para a gestão de pacientes?

Foi discutida a possibilidade de adicionar filtros de pesquisa para encontrar clientes de forma a facilitar a localização de pacientes, além do que é necessário ter ao pesquisar o cliente.

Pergunta 8: Como deve funcionar o registro de comparecimento diário?

A pergunta teve como foco compreender as exigências para um bom uso da funcionalidade principal do sistema.

Pergunta 9: Qual é o procedimento para exclusão e arquivamento de pacientes inativos?

A pergunta esclarece a regra de negócio da empresa e define como deve ser aplicado ao sistema.

3.2 LEVANTAMENTO DE REQUISITOS

Este capítulo trará o levantamento de requisitos retirados da entrevista com o funcionário da clínica Sange Luiz Fernando Selles de Almeida.

3.2.1 REQUISITOS FUNCIONAIS (RF)

Os Requisitos Funcionais descrevem as funções e serviços que o sistema deve fornecer.

Tabela 1 - Requisitos Funcionais.

RF	Descrição	Classificação
RF01	Deve realizar backups diários para garantir a integridade dos dados	Importante
RF02	O cadastro inicial é feito no momento do agendamento da avaliação, coletando nome, convênio e horário. Os dados da ficha de avaliação (histórico, objetivo do tratamento, patologias) devem ser registrados e editáveis.	Essencial
RF03	Deve fornecer uma visualização ampla dos pacientes do dia. Deve ser separada por períodos: Manhã, Tarde e Noite. Deve mostrar a quantidade de clientes em cada período. Deve mostrar o nome, horário e o problema/patologia do paciente.	Essencial
RF04	Deve ser possível clicar no dia do calendário e visualizar os nomes e horários dos clientes que virão. Deve distinguir visualmente (por cores) quais pacientes estão em avaliação.	Desejável
RF05	Deve ter filtros de busca por nome e por convênio para facilitar a localização dos pacientes.	Essencial

Autor: Autoria própria (2025).

Tabela 1 - Conclusão.

RF	Descrição	Classificação
RF06	Deve haver a possibilidade de gerar e imprimir a ficha completa em PDF.	Desejável
RF07	O sistema deve guardar o usuário por até 6 meses após o descadastro, para permitir a reabilitação rápida caso o cliente retorne.	Importante
RF08	Deve-se separar e identificar pacientes através de convênio: "normal" e "especial".	Importante
RF09	Deve permitir registrar o comparecimento de pacientes que vierem em dias avulsos (fora de suas datas padrões) e descontar de sua sessão.	Importante
RF10	Deve ser possível registrar o comparecimento ("cheque"). Se o paciente veio, desconta uma sessão. Se o paciente faltou, a sessão não é descontada.	Essencial
RF11	Deve registrar quantas sessões foram liberadas (ex: 10 sessões iniciais) e ter uma opção para renovar as sessões.	Importante
RF12	O sistema deve manter horários fixos para as sessões agendadas (ex: segunda, quarta e sexta, 14:30), mas deve permitir edições caso o paciente precise alterar o dia da semana que irá.	Essencial
RF13	Na agenda, deve ser possível ver o problema do cliente na tela.	Essencial
RF14	O sistema deve registrar e manter um histórico das últimas edições feitas na ficha do paciente, indicando quem editou.	Desejável
RF15	Deve haver um botão que abra todos os dados cadastrados, incluindo convênio, horários e telefone para contato.	Importante
RF16	A função de registro de comparecimento deve ser rápida, com opções "Confirmar" e "Ausentar"	Essencial
RF17	Deve impedir o agendamento de duas avaliações no mesmo horário. Agendamento de sessões podem ser no mesmo horário das avaliações e outras sessões.	Essencial

Autor: Autoria própria (2025).

3.2.2 REQUISITOS NÃO FUNCIONAIS (RNF)

Os Requisitos Não Funcionais descrevem critérios de qualidade, restrições e atributos do sistema.

Tabela 2 - Requisitos Não Funcionais

RNF	Descrição	Classificação
RNF01	As cores do sistema: Verde, Roxo/Lilás, Preto/Cinza e Azul. Cores com funções específicas: Vermelho (pacientes normais/que já vêm), Cinza (pacientes novos/avaliação), e Azul (pacientes especiais/academia).	Importante
RNF02	A interface deve ser muito intuitiva, porque nem todos na clínica têm familiaridade com PCs.	Importante
RNF03	Deve haver várias opções para fazer a mesma coisa (Redundância).	Desejável
RNF04	O sistema deve ter a funcionalidade de criar a ficha de avaliação digitalmente, baseada no PDF que a clínica já usa.	Desejável
RNF05	A interação principal do sistema (registrar comparecimento) não deve passar de dois cliques. O processo deve ser muito rápido.	Essencial

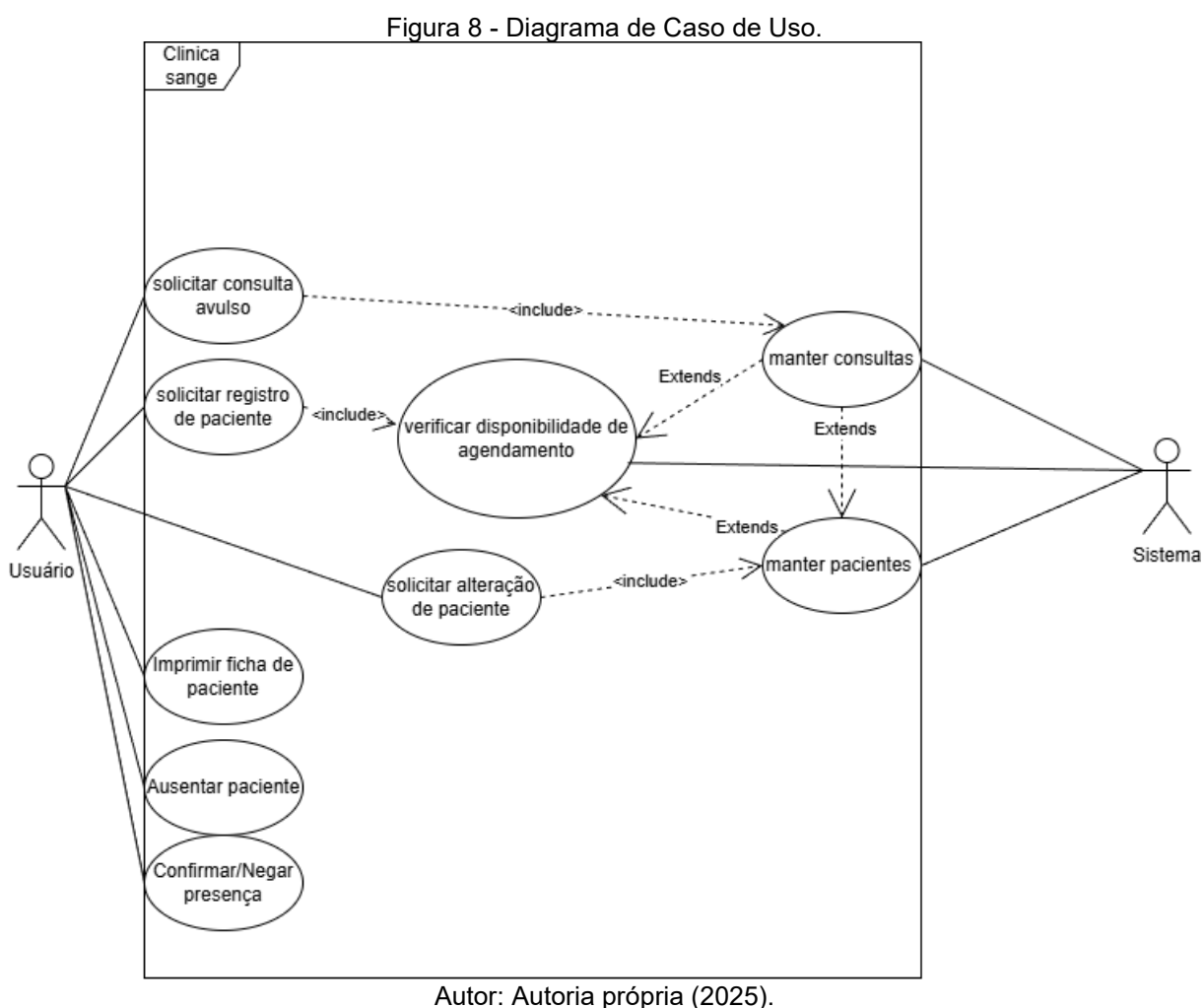
Autor: Autoria própria (2025).

3.3 DIAGRAMAS

Neste tópico demonstra-se os diagramas que foram desenvolvidos para a concepção do sistema em sua complexidade. Apresenta-se os diagramas: Caso de Uso, Atividade, Sequência, Entidade-Relacionamento.

3.3.1 DIAGRAMA DE CASO DE USO

Diagramas de caso de uso mostram as interações principais do sistema para com o usuário de forma simplificada. Diagrama de Caso de Uso desenvolvido para o sistema:



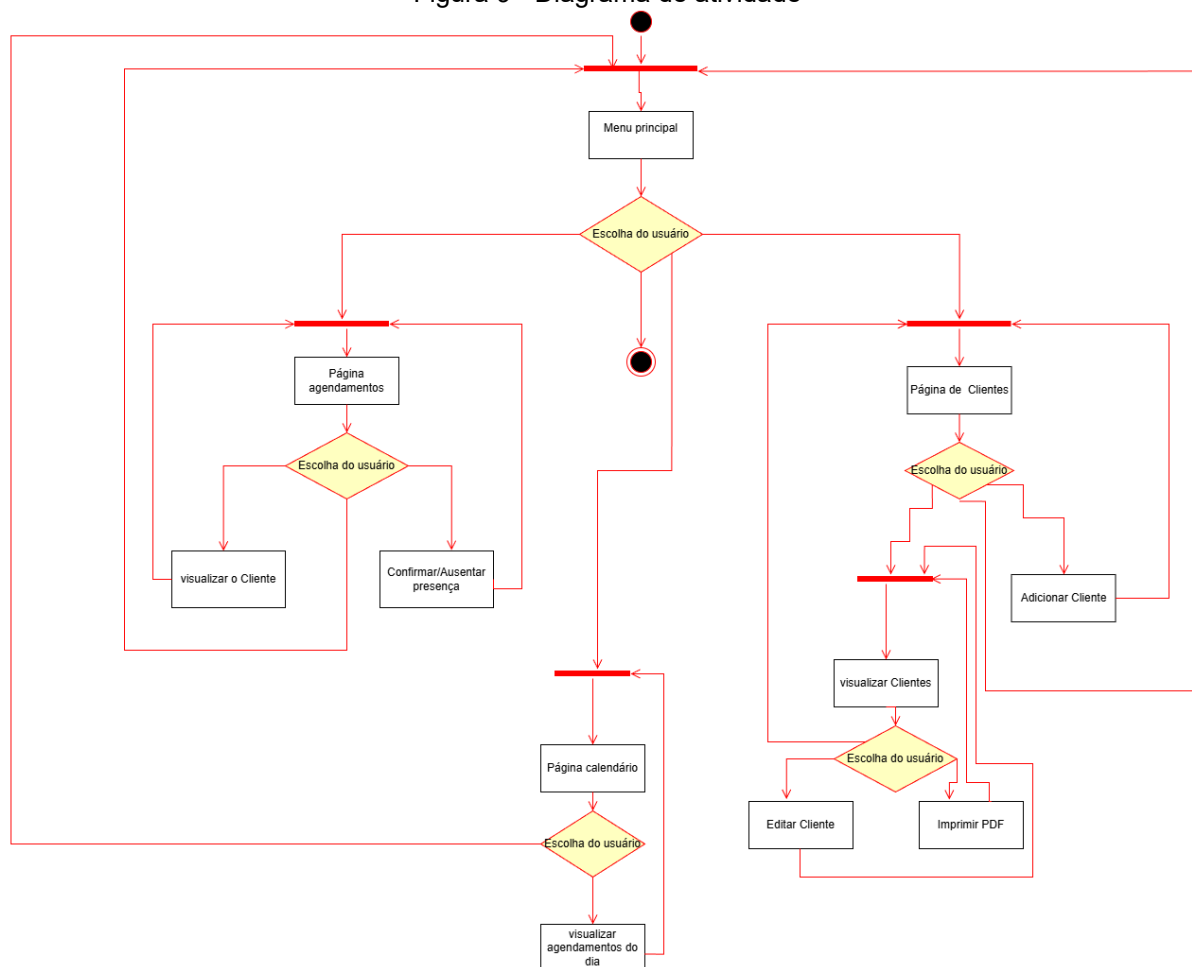
O diagrama de Caso de Uso representa, em sua forma mais básica, o comportamento do sistema com o usuário e suas interações. Interpreta-se o diagrama: Solicitar consulta avulso, Solicitar registro de paciente, Solicitar alteração de paciente,

Imprimir ficha de paciente, Ausentar paciente, Confirmar/Negar presença são ações do usuário. Verificar disponibilidade de agendamento, Manter consultas e Manter pacientes são ações do Sistema. Manter consultas ocorre obrigatoriamente se o usuário solicitar uma consulta avulso, se o usuário solicitar o registro de um novo paciente o sistema irá verificar se o horário está disponível pois todo primeiro registro é uma avaliação que por sua vez não pode ter mais de uma por horário, caso tenha horário disponível o sistema manterá o paciente e sua consulta, caso o usuário solicite uma alteração no paciente estas alterações devem ser feitas pelo sistema e se algum dado relacionado às consultas mudar o próprio sistema deve alterar as consultas deste usuário, por fim o usuário pode imprimir a ficha de pacientes, confirmar ou negar sua presença em uma consulta e por fim ausentamos.

3.3.2 DIAGRAMA DE ATIVIDADE

Diagrama de Atividade mostra como seria a navegação do usuário em uma aplicação, tendo um fluxo e um fim. o diagrama pode ser visto no Apêndice B juntamente de sua descrição.

Figura 9 - Diagrama de atividade



Autor - Autoria própria (2025).

Este diagrama descreve o fluxo de atividades dentro da aplicação, iniciando pelo Menu principal, onde o usuário pode escolher para qual página deseja navegar. Caso selecione a página de Clientes, ele terá a opção de adicionar um novo cliente ou visualizar um já existente. Se desejar, o usuário pode repetir essas ações, retornando à página de Clientes e reiniciando o ciclo, ou voltar ao menu principal.

Ao visualizar um cliente, o usuário também pode gerar um PDF com suas informações ou editar seus dados. Após essas ações, ele retorna à visualização normal, podendo novamente voltar ao menu inicial.

No menu principal, o usuário pode acessar a página de Agendamentos, onde é possível visualizar clientes, confirmar presenças ou registrar ausências. Essas ações também podem ser repetidas quantas vezes necessário, ou o usuário pode retornar ao menu principal.

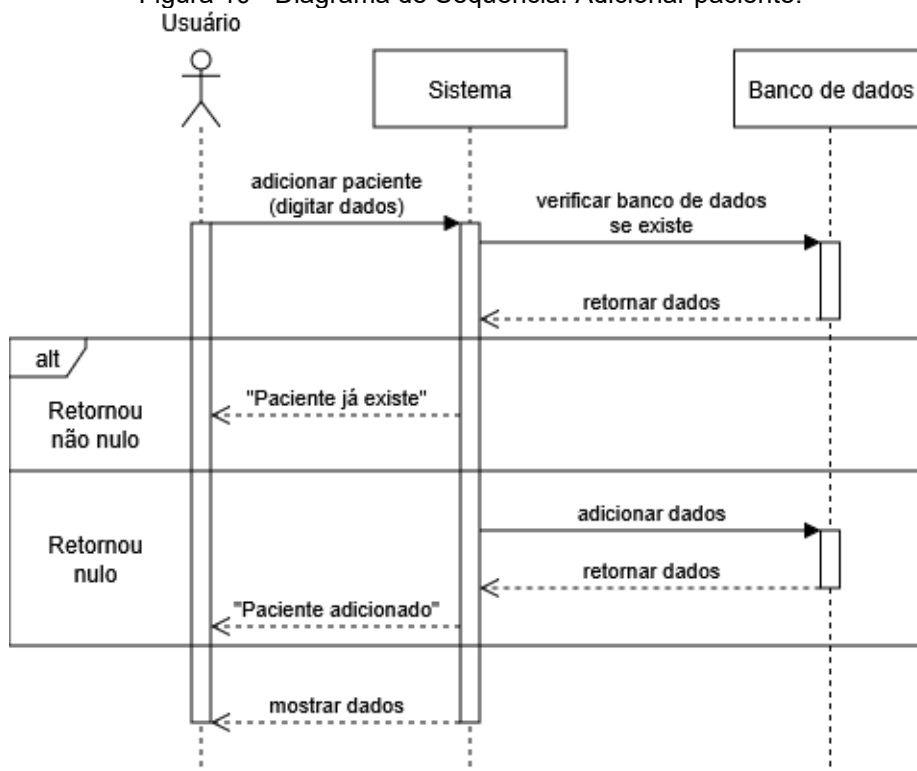
Se o usuário acessar a página de Calendário, ele poderá visualizar os agendamentos de um dia específico ou voltar ao menu principal. Após consultar o dia desejado, pode retornar à página de Calendário.

Por fim, a partir do menu principal, o usuário pode optar por encerrar a aplicação.

3.3.3 DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência representa como o sistema se comporta em resposta a certas ações. Diagramas de Sequência desenvolvidos para o sistema:

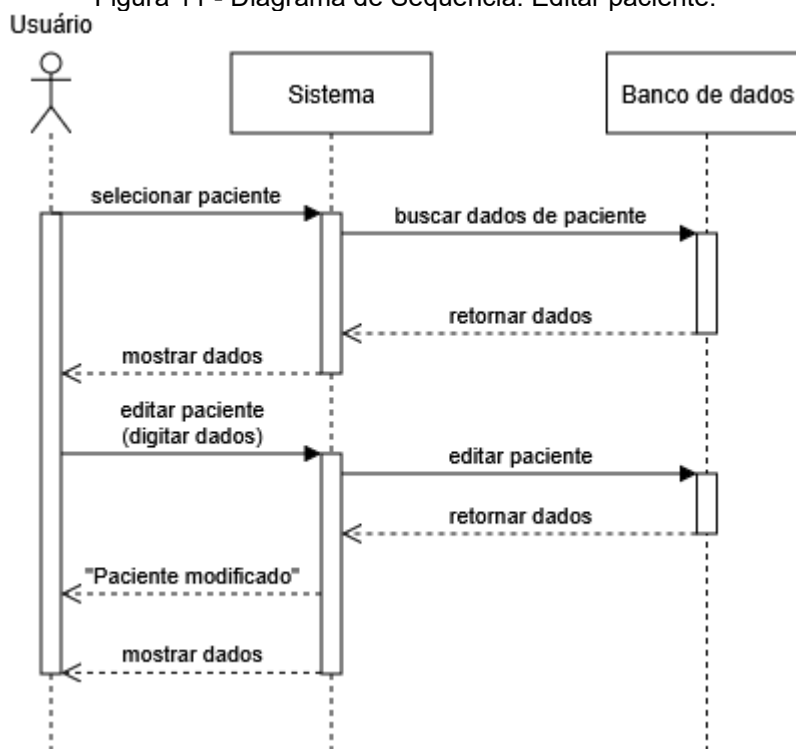
Figura 10 - Diagrama de Sequência: Adicionar paciente.



Autor: Aatoria própria (2025).

Interpreta-se o presente diagrama: Ao usuário realizar a ação adicionar paciente, o sistema deve verificar se o paciente não existe no banco de dados para, em seguida, realizar a ação de adição e retornar a resposta para o usuário caso foi adicionado com sucesso ou não.

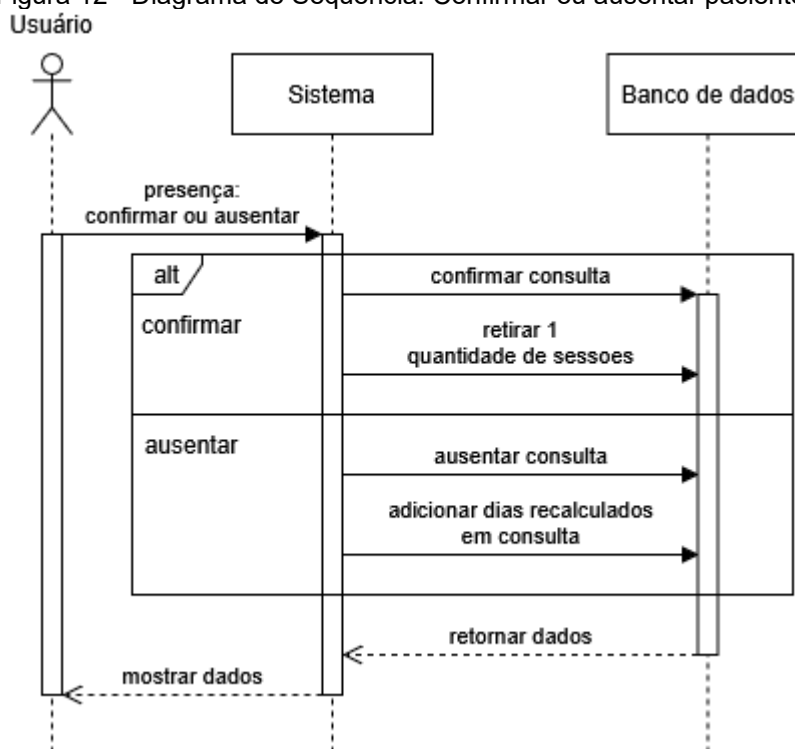
Figura 11 - Diagrama de Sequência: Editar paciente.



Autor: Autoria própria (2025).

Interpreta-se o presente diagrama: O usuário seleciona as informações do paciente a serem editadas, o sistema coleta todas as informações do paciente e retorna para a tela do usuário, ao usuário digitar os dados alterados do paciente, o sistema envia a requisição de alteração para o banco de dados, retornando a resposta para o usuário.

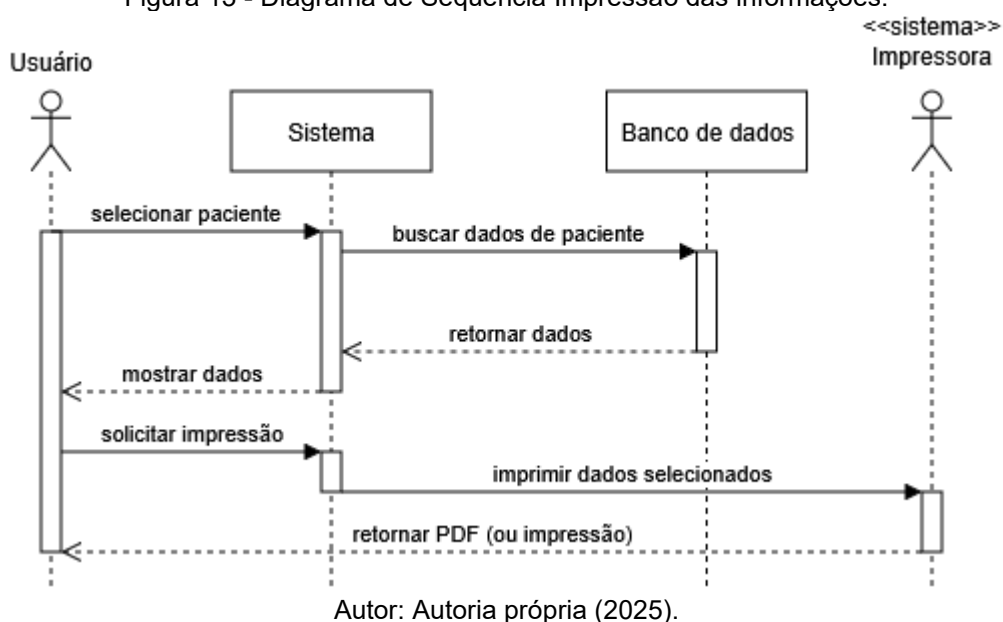
Figura 12 - Diagrama de Sequência: Confirmar ou ausentar paciente.



Autor: Autoria própria (2025).

Interpreta-se o presente diagrama: O usuário pode confirmar ou ausentar pacientes, o sistema interpreta a ação escolhida e responde de acordo, ao confirmar, retira-se 1 (um) da quantidade de sessões restantes do paciente do sistema, ao ausentar, recalcula os dias para comportar a sessão que foi perdida pelo paciente, no final, retorna os dados atualizados para o usuário.

Figura 13 - Diagrama de Sequência Impressão das informações.



Interpreta-se o presente diagrama: O usuário seleciona as informações do paciente a serem impressas, o sistema coleta todas as informações do paciente e retorna para a tela do usuário, no qual solicita a impressão e recebe a resposta diretamente do sistema de impressão.

3.3.4 DIAGRAMA DE ENTIDADE-RELACIONAMENTO

Este diagrama representa a estrutura do banco de dados do sistema e as relações entre cada entidade: Imagem e descrição do diagrama adicionado ao Apêndice A para melhor visualização.

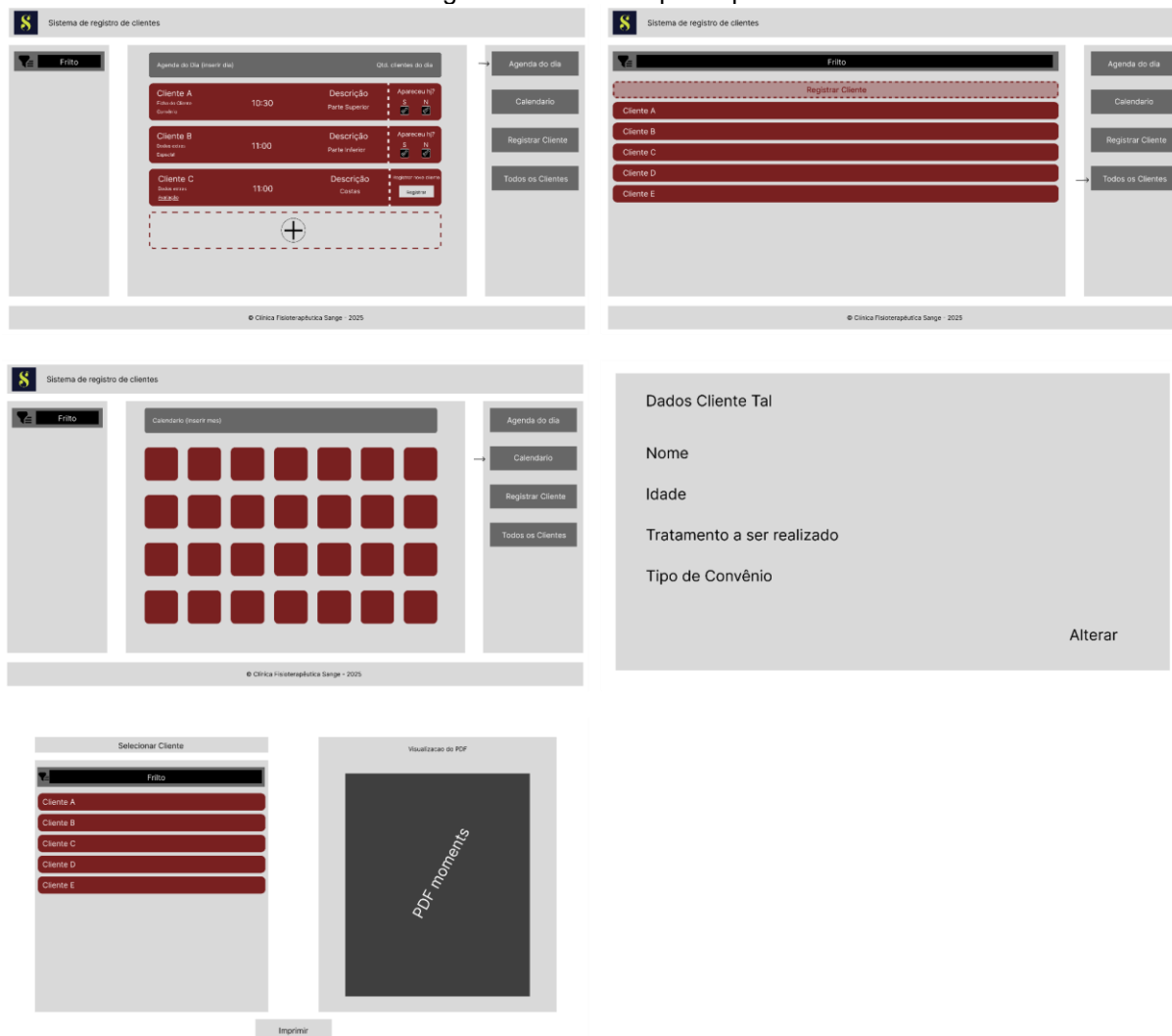
3.4 PROTÓTIPOS DE DESIGN

Como afirma Cross (2011), um projeto precisa contar com protótipos que reflitam seu valor real, buscando ao máximo antecipar o que a aplicação final pretende oferecer. Esses protótipos devem ser analisados e apresentados constantemente aos clientes, de modo a obter feedbacks e estimular novas ideias criativas por parte dos designers responsáveis por sua produção.

3.4.1 PRIMEIRO PROTÓTIPO DE DESIGN

O primeiro protótipo foi desenvolvido com o objetivo de demonstrar como todas as funções seriam apresentadas na tela do computador. Isto é, ele ainda não seguia o padrão de cores da clínica. Abaixo segue o resultado:

Figura 15 - Primeiro protótipo.



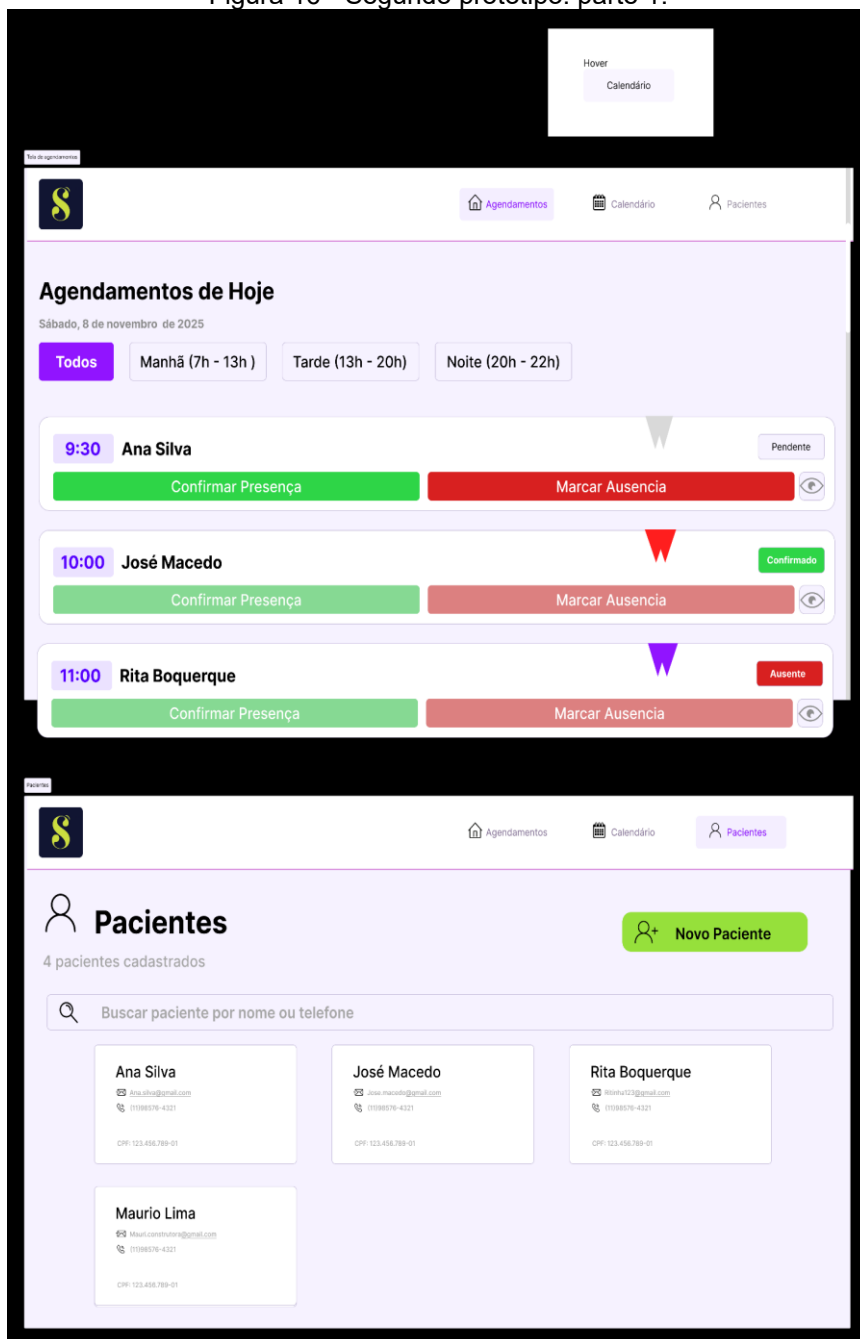
Autor: Autoria própria (2025).

Como a imagem mostra, nesse conceito haveria um *header* em todas as páginas, juntamente com componentes que apareceriam ocasionalmente, como o filtro ou o menu lateral de navegação. Nesse protótipo também são apresentadas as representações de cada uma das funções e telas. No entanto, os usuários da clínica relataram que esse modelo parecia muito antiquado e solicitaram algo mais moderno, além de apontarem que as cores ainda não seguiam o padrão visual da clínica.

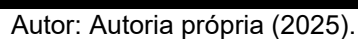
3.4.2 SEGUNDO PROTÓTIPO DE DESIGN

Este design foi feito visando atender os pedidos dos usuários acerca do visual da aplicação e melhorar sua usabilidade e praticidade, retirando materiais desnecessários da tela e aplicando melhorias a eles, os resultados estão abaixo:

Figura 16 - Segundo protótipo: parte 1.



Autor: Autoria própria (2025).



O design foi modernizado e personalizado para os padrões de cores seguidos nas redes sociais da clínica Sange, os elementos foram melhor destacados pelo

contraste das cores, este design foi muito bem aceito pelos usuários e foi tomado como o padrão final que seria seguido na aplicação final.

3.5 DICIONÁRIO DE DADOS

O capítulo de dicionário de dados apresenta tabelas descritivas de todos os atributos de cada entidade do banco de dados, organizando-os de forma a melhorar o entendimento da modelagem do banco de dados do sistema. Apresenta-se as tabelas: Avulso, Consulta, DiaHoraAgendado, Endereco, ExameFisico, Paciente, Patologia, Sessao, TratamentoFisioterapico.

Tabela 3 - Dicionário de dados: Avulso.

Coluna	Tipo	Nulo	Detalhes / Chave	Descrição
avuCodigo	int(11)	Não	Chave Primária (PRIMARY)	Código identificador da tabela Avulso
avuSesCodigo	int(11)	Não	Chave Estrangeira (fk_avulso_sessao)	Chave estrangeira identificadora da tabela Sessão
avuDtConsulta	date	Sim	Padrão: <i>NULL</i>	Data da sessão para avulsos
avuHorario	time	Sim	Padrão: <i>NULL</i>	Horário da sessão avulsa

Autor: Autoria própria (2025).

Tabela 4 - Dicionário de dados: Consulta.

Coluna	Tipo	Nulo	Detalhes / Chave	Descrição
conCodigo	int(11)	Não	Chave Primária (PRIMARY)	Código identificador da tabela Consulta
conDiaCodigo	int(11)	Não	Padrão: <i>NULL</i> . Chave Estrangeira (fk_consulta_diahoraagendado)	Chave estrangeira identificadora da tabela DiaHoraAgendado
conDiaAgendado	date	Sim	Padrão: <i>NULL</i>	Data do dia da sessão
conStatusDiaAgendado	int(1)	Sim	Padrão: <i>NULL</i>	Confirmado ou Ausente no dia

Autor: Autoria própria (2025).

Tabela 5 - Dicionário de dados: DiaHoraAgendado.

Coluna	Tipo	Nulo	Detalhes / Chave	Descrição
diaCodigo	int(11)	Não	Chave Primária (PRIMARY)	Código identificador da tabela DiahoraAgendado
diaSesCodigo	int(11)	Não	Chave Estrangeira (fk_diaHoraAgendado_sessao)	Chave estrangeira identificadora da tabela Sessão
diaSegunda	tinyint(1)	Sim	Padrão: <i>NULL</i>	Os dias da semana que a pessoa fará as sessões
diaTerca	tinyint(1)	Sim	Padrão: <i>NULL</i>	Os dias da semana que a pessoa fará as sessões
diaQuarta	tinyint(1)	Sim	Padrão: <i>NULL</i>	Os dias da semana que a pessoa fará as sessões
diaQuinta	tinyint(1)	Sim	Padrão: <i>NULL</i>	Os dias da semana que a pessoa fará as sessões
diaSexta	tinyint(1)	Sim	Padrão: <i>NULL</i>	Os dias da semana que a pessoa fará as sessões
diaQtdSessao	int(11)	Sim	Padrão: <i>NULL</i>	Quantidade de sessões restantes
diaTotalSessao	int(11)	Sim	Padrão: <i>NULL</i>	Quantidade de sessões totais
diaHorario	time	Sim	Padrão: <i>NULL</i>	Horário das sessões
diaDtInicioSessao	date	Sim	Padrão: <i>NULL</i>	Dia que a sessão começou

Autor: Autoria própria (2025).

Tabela 6 - Dicionário de dados: Endereco.

Coluna	Tipo	Nulo	Detalhes / Chave	Descrição
endCodigo	int(11)	Não	Chave Primária (PRIMARY)	Código identificador da tabela Endereco
endPacCodigo	int(11)	Não	Chave Estrangeira (fk_endereco_paciente)	Chave estrangeira identificadora da tabela Paciente
endCEP	int(11)	Sim	Padrão: <i>NULL</i>	CEP do paciente
endBairro	varchar(255)	Sim	Padrão: <i>NULL</i>	Bairro do paciente
endRua	varchar(255)	Sim	Padrão: <i>NULL</i>	Rua do paciente
endCidade	varchar(255)	Sim	Padrão: <i>NULL</i>	Cidade do paciente
endNumero	int(11)	Sim	Padrão: <i>NULL</i>	Número do paciente
endComplemento	varchar(255)	Sim	Padrão: <i>NULL</i>	Complemento do paciente

Autor: Autoria própria (2025).

Tabela 7 - Dicionário de dados: ExameFisico.

Coluna	Tipo	Nulo	Detalhes / Chave	Descrição
exaCodigo	int(11)	Não	Chave Primária (PRIMARY)	Código identificador da tabela ExameFisico
exaSesCodigo	int(11)	Não	Chave Estrangeira (fk_exameFisico_sessao)	Chave estrangeira identificadora da tabela Sessao
exaPA	varchar(255)	Sim	Padrão: <i>NULL</i>	PA do paciente
exaFR	varchar(255)	Sim	Padrão: <i>NULL</i>	FR do paciente
exaFC	varchar(255)	Sim	Padrão: <i>NULL</i>	FC do paciente
exaInspecao	varchar(255)	Sim	Padrão: <i>NULL</i>	Anotação de inspeção
exaPalpacao	varchar(255)	Sim	Padrão: <i>NULL</i>	Anotação de palpação
exaDorPalpacao	tinyint(1)	Sim	Padrão: <i>NULL</i>	Sim ou não dor a palpação

Autor: Autoria própria (2025).

Tabela 7 - Conclusão.

exaDorPalpac aoDesc	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição de dor à palpação
exaEdema	tinyint(1)	Sim	Padrão: <i>NULL</i>	Sim ou não a Edema
exaEdemaDe sc	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição de Edema
exaTestesEsp ecificos	varchar(255)	Sim	Padrão: <i>NULL</i>	Anotação de Testes Específicos
exaADM	tinyint(1)	Sim	Padrão: <i>NULL</i>	Sim ou não a ADM
exaADMDesc	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição de ADM
exaFM	tinyint(1)	Sim	Padrão: <i>NULL</i>	Sim ou não a FM
exaFMDesc	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição de FM
exaTonusMus cular	int(1)	Sim	Padrão: <i>NULL</i>	Sim ou não a Tonus Muscular
exaTonusMus cularDesc	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição de Tonus Muscular
exaMoviment o	int(1)	Sim	Padrão: <i>NULL</i>	Sim ou não a Movimento
exaFazUsoOrt ese	tinyint(1)	Sim	Padrão: <i>NULL</i>	Sim ou não a Ortese
exaFazUsoOrt eseDesc	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição de Ortese
exaDesviosP osturais	tinyint(1)	Sim	Padrão: <i>NULL</i>	Sim ou não a Desvios Posturais
exaDesviosP osturaisDesc	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição de Desvios Posturais

Autor: Autoria própria (2025).

Tabela 8 - Dicionário de dados: Paciente.

Coluna	Tipo	Nulo	Detalhes / Chave	Descrição
pacCodigo	int(11)	Não	Chave Primária (PRIMARY)	Código identificador da tabela Paciente
pacNumCarteirinha	varchar(255)	Sim	Padrão: <i>NULL</i>	Número da Carteirinha do paciente
pacTipoConvênio	varchar(255)	Não		Tipo de convênio do paciente
pacDesativado	tinyint(1)	Sim	Padrão: <i>NULL</i>	Desabilitar paciente para apagar os dados do mesmo após certo tempo
pacDtDesativado	date	Sim	Padrão: <i>NULL</i>	Data de Desabilitar paciente
pacNome	varchar(255)	Não		Nome do paciente
pacDtNascimento	date	Sim	Padrão: <i>NULL</i>	Data de nascimento do paciente
pacSexo	char(1)	Sim	Padrão: <i>NULL</i>	Sexo do paciente
pacEstadoCivil	varchar(63)	Sim	Padrão: <i>NULL</i>	Estado Civil do paciente
pacPeso	float	Sim	Padrão: <i>NULL</i>	Peso do paciente
pacAltura	float	Sim	Padrão: <i>NULL</i>	Altura do paciente
pacProfissao	varchar(255)	Sim	Padrão: <i>NULL</i>	Profissao do paciente
pacFumante	tinyint(1)	Sim	Padrão: <i>NULL</i>	Se paciente é fumante
pacNivelImportancia	int(1)	Sim	Padrão: <i>NULL</i>	Se o paciente é normal, especial ou avaliacao
pacEmail	varchar(255)	Sim	Padrão: <i>NULL</i>	Email do paciente
pacTelefone	varchar(23)	Sim	Padrão: <i>NULL</i>	Telefone do paciente
pacCpf	varchar(14)	Sim	Padrão: <i>NULL</i>	CPF do paciente

Autor: Autoria própria (2025).

Tabela 9 - Dicionário de dados: Patologia.

Coluna	Tipo	Nulo	Detalhes / Chave	Descrição
patCodigo	int(11)	Não	Chave Primária (PRIMARY)	Código identificador da tabela Patologia
patSesCodigo	int(11)	Não	Chave Estrangeira (fk_patologia_sessao)	Chave estrangeira identificadora da tabela Sessao
patDiagnosticoClinico	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Diagnóstico Clínico
pathMA	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição HMA
patAntecedentesPessoais	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Antecedentes Pessoais
patPatologiaAssociada	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Patologia Associada
patTomaMedicamento	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição de medicamentos
patQuandoDorComeçou	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Quando a dor começou
patQualPosicaoDorMaisIntensa	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Qual posição a dor é mais intensa
patQualPosicaoTrabalho	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Posição de trabalho
patFezCirurgia	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Nome da cirurgia realizada
patExamesComplementares	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Exames Complementares
patHaComprometimentoAVS	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Se há comprometimento AVS
patLimitacaoFuncional	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Limitação Funcional

Autor: Autoria própria (2025).

Tabela 9 - Conclusão.

patComprometimentoMarcha	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição Comprometimento de Marcha
patDataCirurgia	varchar(63)	Sim	Padrão: <i>NULL</i>	Data da Cirurgia realizada

Autor: Autoria própria (2025).

Tabela 10 - Dicionário de dados: Sessão.

Coluna	Tipo	Nulo	Detalhes / Chave	Descrição
sesCodigo	int(11)	Não	Chave Primária (PRIMARY)	Código identificador da tabela Sessao
sesPacCodigo	int(11)	Não	Chave Estrangeira (fk_sessao_paciente)	Chave estrangeira identificadora da tabela Paciente
sesParteSuperior	tinyint(1)	Sim	Padrão: <i>NULL</i>	Problema na parte superior
sesParteInferior	tinyint(1)	Sim	Padrão: <i>NULL</i>	Problema na parte inferior
sesColuna	tinyint(1)	Sim	Padrão: <i>NULL</i>	Problema na coluna
sesDtAvaliacao	datetime	Sim	Padrão: <i>NULL</i>	Dia da avaliação
sesDescricao	varchar(8191)	Sim	Padrão: <i>NULL</i>	Observações extras
sesUltimaEdicao	varchar(255)	Sim	Padrão: <i>NULL</i>	Última pessoa a editar os dados do paciente
sesGravidadePSuperior	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição detalhada do problema parte superior
sesGravidadePInferior	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição detalhada do problema parte inferior
sesGravidadeColuna	varchar(255)	Sim	Padrão: <i>NULL</i>	Descrição detalhada do problema coluna

Autor: Autoria própria (2025).

Tabela 11 - Dicionário de dados: TratamentoFisioterapico.

Coluna	Tipo	Nulo	Detalhes / Chave	Descrição
traCodigo	int(11)	Não	Chave Primária (PRIMARY)	Código identificador da tabela TratamentoFisioterapico
traSesCodigo	int(11)	Não	Chave Estrangeira (fk_tratamentoFisioterapico_sessao)	Chave estrangeira identificadora da tabela Sessão
traObjetivoTratamento	varchar(255)	Sim	Padrão: <i>NULL</i>	Objetivo do tratamento
traTratamentoProposto	varchar(255)	Sim	Padrão: <i>NULL</i>	Qual o tratamento proposto

Autor: Autoria própria (2025).

3.6 SPRINTS

O projeto foi desenvolvido utilizando a metodologia de sprints do Scrum. Como explica Asproni (2006), em cada período são estabelecidas metas a serem atingidas em relação ao projeto. Para isso, contam-se com ferramentas como o planejamento da sprint e o gráfico de burndown, que permitem à equipe e aos interessados obter uma visão clara do andamento do desenvolvimento de todas as tarefas e garantem a qualidade do processo. Neste projeto, optamos por realizar quatro sprints, distribuídas entre março e novembro, organizadas conforme a necessidade das atividades.

3.6.1 PRIMEIRA SPRINT

Com a primeira sprint, desejava-se obter uma compreensão mais aprofundada das necessidades dos usuários em relação à aplicação e definir os tipos de requisitos, classificando-os entre funcionais e não funcionais, além de definir quais ferramentas e tecnologias seriam utilizadas. Isso deixaria o caminho livre para desenvolver utilizando as ferramentas previstas na sprint seguinte. O planejamento desta sprint ficou conforme apresentado abaixo:

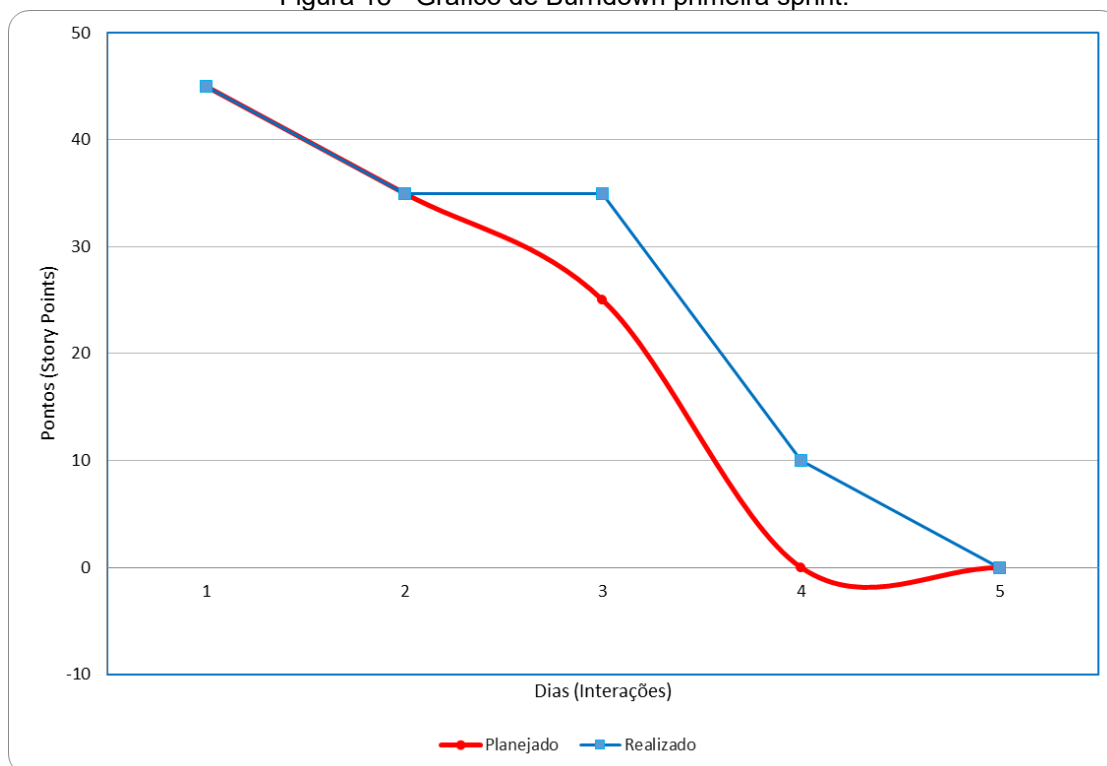
Tabela 12 - Planejamento da primeira sprint.

Atividade	Dias	Pontos
Levantamento de requisitos(primeira entrevista)	1	10
Classificando requisitos funcionais e não funcionais	10	10
Definição de ferramentas	20	25
Total	30	45

Autor: Autoria própria (2025).

Esta sprint ocorreu ao longo de todo o mês de março. Embora algumas tarefas tenham sido concluídas com certo atraso, o resultado esperado foi alcançado ao final do período. As tecnologias foram completamente definidas, sendo elas React, Typescript, JavaScript, GitHub, MySQL, Visual Studio Code e, por fim, PHP. Nosso desempenho foi melhor do que o esperado ao longo da sprint, conforme demonstrado no gráfico abaixo:

Figura 18 - Gráfico de Burndown primeira sprint.



Autor: Autoria própria (2025).

3.6.2 SEGUNDA SPRINT

Para a segunda sprint optou-se por documentar e fundamentar o que foi planejado na primeira sprint, além de iniciar a prototipagem da aparência do sistema, além de definir como seria sua funcionalidade através dos diagramas de caso de uso, atividade, sequência e entidade-relacionamento, assim tendo grande parte das bases para começar a desenvolver o software posteriormente, sendo assim a tabela de planejamento ficou assim:

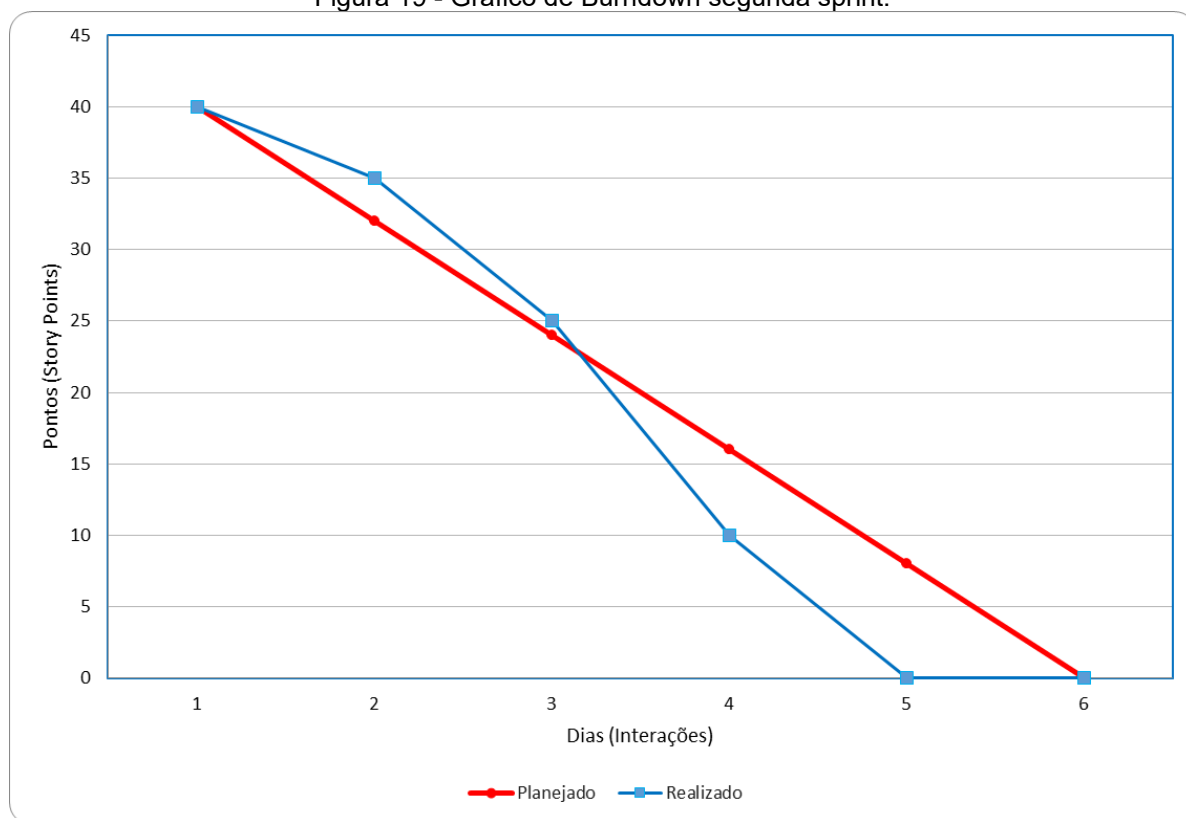
Tabela 13 - Planejamento da segunda sprint.

Atividade	Dias	Pontos
Referencial teórico	10	5
Primeiro protótipo de design	20	10
Desenvolvimento do diagrama de caso de uso	10	5
Desenvolvimento do diagrama de sequência	10	5
Desenvolvimento do diagrama de Atividade	10	10
Desenvolvimento do diagrama Entidade-relacionamento	10	10
Total	40	45

Autor: Autoria própria (2025).

Esta sprint foi realizada ao longo dos meses de abril, maio e junho. De modo geral, a maior parte das tarefas foi concluída rapidamente após sua proposição. No entanto, os diagramas foram distribuídos ao longo de toda a sprint devido a mudanças de visão e ajustes necessários nos mesmos. O resultado é apresentado abaixo:

Figura 19 - Gráfico de Burndown segunda sprint.



Autor: Autoria própria (2025).

3.6.3 TERCEIRA SPRINT

Nesta sprint, foi proposta a melhoria do design da aplicação, buscando torná-lo mais moderno e eficiente. Além disso, foram criadas a base de dados do projeto e seu respectivo dicionário de dados. Também foi planejado o registro de tudo o que havia sido desenvolvido neste documento até o momento. Por fim, deu-se início ao desenvolvimento da aplicação, contemplando inicialmente duas telas: a Home e a tela de Clientes. A tabela que descreve essas atividades é apresentada abaixo.

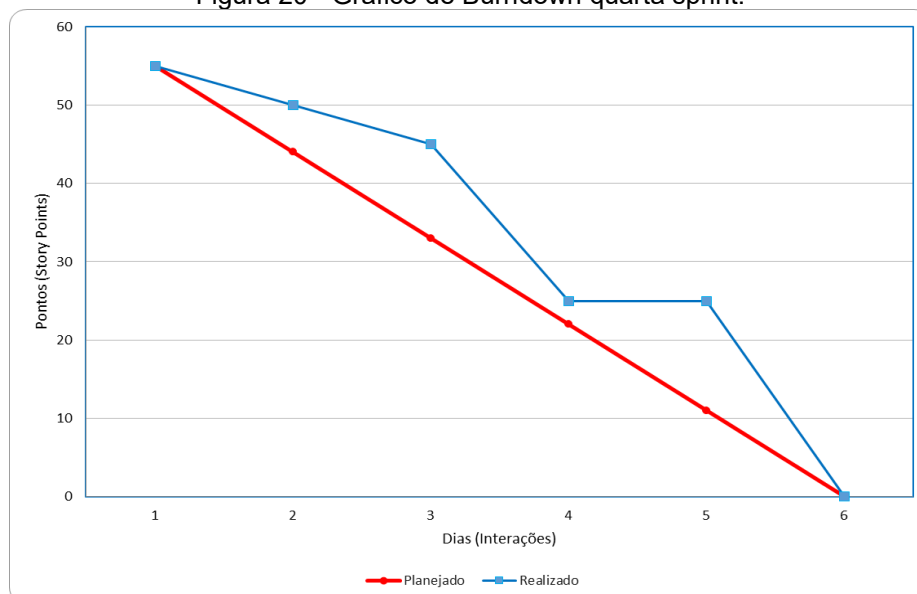
Tabela 14 - Planejamento da terceira sprint.

Atividade	Dias	Pontos
Segundo protótipo de design	10	10
Criação do banco de dados	15	10
Dicionário de dados	20	15
Desenvolvimento do documento TCC	15	10
Desenvolvimento da Tela Home	5	5
Desenvolvimento da Tela Clientes	5	5
Total	60	55

Autor: Autoria própria (2025).

A terceira sprint durou um período de 2 meses cobrindo de agosto à setembro, as atividades foram realizadas com certa dificuldade o que culminou para as atividades mais serem entregues perto do final da sprint, o gráfico abaixo demonstra essa situação:

Figura 20 - Gráfico de Burndown quarta sprint.



Autor: Autoria própria (2025).

3.6.4 QUARTA SPRINT

A quarta sprint foi planejada com foco total no desenvolvimento da aplicação, de modo a cumprir seu propósito, mesmo que ainda não estivesse completamente finalizada. Dessa forma, iniciou-se o desenvolvimento do back-end de todas as funções previstas. Assim, a tabela de planejamento ficou da seguinte forma:

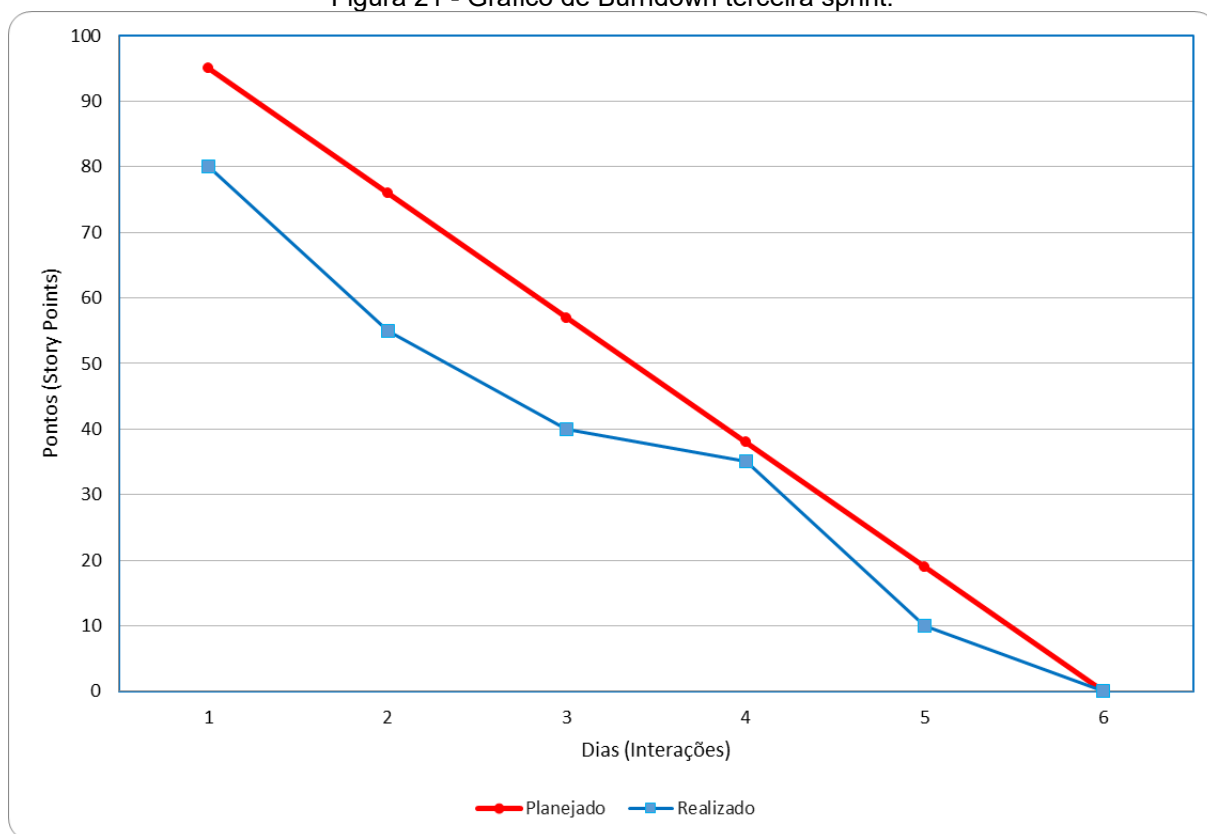
Tabela 15 - Planejamento da quarta sprint.

Atividade	Dias	Pontos
Desenvolvimento da tela de calendário	5	5
Conexão com banco de dados	5	10
Aplicação dos agendamentos(Com confirmação e ausência)	10	25
Criação e aplicação do formulário de cadastro	15	30
Aplicação da visualização dos pacientes	5	10
Aplicar o sistema de Impressão de ficha de paciente	15	10
Aplicar filtros de pesquisa e de períodos	5	5
Total	60	95

Autor: Autoria própria (2025).

A última sprint apresentou o melhor desempenho dentro do esperado, pois seu desenvolvimento transcorreu sem grandes problemas ou obstáculos. As funcionalidades da aplicação, de modo geral, foram implementadas de forma tranquila. No entanto, o formulário de cadastro representou um pequeno desafio devido à grande quantidade de dados e requisitos envolvidos. O gráfico que descreve seu desenvolvimento está apresentado abaixo:

Figura 21 - Gráfico de Burndown terceira sprint.



Autor: Autoria própria (2025).

4 RESULTADOS

O presente tópico descreve os resultados obtidos do trabalho e o que foi atingido com o desenvolvimento do software.

O software conseguiu implementar o design proposto pelos usuários, além de expandir esse conceito ao adicionar um sistema de modo noturno e diurno, de modo a atender às preferências de todos os usuários, estes modos são retratados no Apêndice C, figura 22 e 23.

Outros êxitos foram tanto o sistema de agendamentos quanto o formulário para adicionar ou editar os pacientes que de forma eficiente cumprem sua função, suas imagens seguem no Apêndice C, figuras 24, 25 e 26.

A tela de calendário foi aplicada visualmente, porém se mantém sem funções extras, enquanto isso a tela de pacientes já recebe os pacientes e os apresenta registrados no sistema permitindo que o usuário os selecione e, por fim, possa imprimir um PDF com os dados do paciente selecionado, as imagens que ilustram esses resultados estão no Apêndice C, figuras 27,28 e 29.

E por fim foi implementado a função de filtro no sistema com intuito de facilitar a busca por informações específicas dentro da aplicação, este que está à mostra no Apêndice C, figura 30.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema de agendamento para a clínica fisioterapêutica Sange demonstrou-se necessário para a automatização de processos que antes eram realizados por meio de ferramentas como o Google Agenda, desta forma, proporcionou-se maior controle sobre os atendimentos, organização das informações e redução de erros operacionais, contribuindo diretamente para a melhoria da rotina dos profissionais da clínica.

Desta forma, o sistema contribui significativamente para a clínica de forma a facilitar a disposição e organização das informações dos pacientes conforme necessidade. Para atualizações futuras do sistema, considera-se a expansão de funcionalidades, como o aprimoramento da visualização dos dados, adição de possíveis funcionalidades como o calendário interativo, ou outras funções que a clínica venha a solicitar garantindo-se, desta forma, a melhoria e eficiência dos atendimentos da clínica fisioterapêutica Sange.

REFERÊNCIAS

- AGILE MANIFESTO. *Manifesto for Agile Software Development*. 2001. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>
- ANDRADE, Vinicius Santos. Obtendo sucesso em métodos ágeis com Scrum: estudo sobre fatores relacionados ao trabalho em equipe. *Revista Computação Aplicada - UNG-Ser*, [S. l.], v. 7, n. 1, p. 37–44, 2019. DOI: 10.33947/2316-7394-v7n1-3067. Disponível em: <https://revistas.ung.br/index.php/computacaoaplicada/article/view/3067>
- ASPRONI, G. An introduction to Scrum. *Software Developer's Journal*, [S. l.], p. 1–10, June 2006. Disponível em: https://asprotunity.com/resources/articles/AnIntroductionToScrum_SDJ_06-2006_EN.pdf
- AZEVEDO-MARQUES, Paulo Mazzoncini de; SALOMÃO, Samuel Covas. PACS: sistemas de arquivamento e distribuição de imagens. *Revista Brasileira de Física Médica*, [S. l.], v. 3, n. 1, p. 131–139, 2015. Disponível em: <https://www.rbmf.org.br/rbmf/article/view/39/v3n1p131>
- BENTO, Evaldo Junior. *Desenvolvimento web com PHP e MySQL*. São Paulo: Casa do Código, 2013.
- BERALDI, Lairce Castanhera; ESCRIVÃO FILHO, Edmundo. Impacto da tecnologia de informação na gestão de pequenas empresas. *Ciência da Informação*, Brasília, v. 29, n. 1, p. 46–50, jan./abr. 2000. Disponível em: <https://doi.org/10.1590/S0100-19652000000100005>
- BIFFI, Kelvin Baumhardt. *JavaScript básico ao avançado*. [S. l.]: [s. n.], 2018. e-book. Disponível em: [https://www.kufunda.net/publicdocs/Javascript%20B%C3%A1sico%20ao%20Avan%C3%A7ado%20\(Kelvin%20Baumhardt%20Biffi\).pdf](https://www.kufunda.net/publicdocs/Javascript%20B%C3%A1sico%20ao%20Avan%C3%A7ado%20(Kelvin%20Baumhardt%20Biffi).pdf)
- BOGNER, Justus; MERKEL, Manuel. To type or not to type? A systematic comparison of the software quality of JavaScript and TypeScript applications on GitHub. In: INTERNATIONAL CONFERENCE ON MINING SOFTWARE REPOSITORIES, 19., 2022, Honolulu, Hawaii. *Proceedings...* New York: ACM, 2022. p. 1–12. Disponível em: <https://arxiv.org/abs/2203.11115>
- CAMARGO, Beatriz Santos et al. *Desenvolvimento de um aplicativo para clínicas e profissionais de fisioterapia*. 2021. 58 f. Trabalho de Graduação (Tecnologia em Análise e Desenvolvimento de Sistemas) – Faculdade de Tecnologia de Americana – Ministro Ralph Biasi, Americana, 2021. Disponível em: https://ric.cps.sp.gov.br/bitstream/123456789/15224/1/20211S_Beatriz%20Santos%20Camargo_OD1479.pdf
- CHEN, S.; THADURI, U. R.; BALLAMUDI, V. K. R. Front-end development in React: an overview. *Engineering International*, [S. l.], v. 7, n. 2, p. 117–126, dez. 2019. Disponível em: <https://abc.us.org/ojs/index.php/ei/article/view/662>
- CHERNY, Boris. *Programming TypeScript: Making Your JavaScript Applications Scale*. Sebastopol: O'Reilly Media, 2019. Disponível em: <https://books-library.website/files/books-library.net-10132058Ts3U9.pdf>

CROSS, Nigel. *Design thinking: understanding how designers think and work*. Oxford: Berg, 2011.

DEL SOLE, Alessandro. *Visual Studio Code Distilled: Evolved Code Editing for Windows, macOS, and Linux*. 3. ed. New York: Apress, 2023. Disponível em: <https://doi.org/10.1007/978-1-4842-9484-0>

ESMIN, Ahmed Ali Abdalla. Modelando com UML – Unified Modeling Language. *INFOCOMP Journal of Computer Science*, [S. l.], v. 1, n. 1, p. 48–58, 2004. Disponível em: <https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/36>

FRANCO, Eduardo Ferreira. Um modelo de gerenciamento de projetos baseado nas metodologias ágeis de desenvolvimento de software e nos princípios da produção enxuta. 2007. Dissertação (Mestrado em Sistemas Digitais) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2007. Disponível em: <https://doi.org/10.11606/D.3.2007.tde-09012008-155823>

GODOI, Jady Sobjak de Mello et al. O uso do prontuário eletrônico por enfermeiros em Unidades Básicas de Saúde brasileiras. *Journal of Health Informatics*, v. 4, n. 1, 2012. Disponível em: <https://jhi.sbis.org.br/index.php/jhi-sbis/article/view/146>

JANG, Dongseok; CHOE, Kwang-Moo. Points-to analysis for JavaScript. In: ACM SIGPLAN-SIGACT SYMPOSIUM ON PRINCIPLES OF PROGRAMMING LANGUAGES, 5., 2009, Honolulu. *Proceedings...* Honolulu: ACM, 2009. p. 1930–1937. Disponível em: https://www.researchgate.net/publication/221000891_Points-to_analysis_for_Javascript

KOMPERLA, Varun et al. React: a detailed survey. *Indonesian Journal of Electrical Engineering and Computer Science*, Bengaluru, v. 26, n. 3, p. 1710–1717, jun. 2022. Disponível em: <http://doi.org/10.11591/ijeecs.v26.i3.pp1710-1717>

LUNARDI, Guilherme Lerch; DOLCI, Pietro Cunha; MAÇADA, Antônio Carlos Gastaud. Adoção de tecnologia de informação e seu impacto no desempenho organizacional. *Revista de Administração*, São Paulo, v. 45, n. 1, p. 5–17, jan./mar. 2010. Disponível em: [https://doi.org/10.1016/S0080-2107\(16\)30505-2](https://doi.org/10.1016/S0080-2107(16)30505-2)

MARIANO, Carl Lawrence. Benchmarking JavaScript Frameworks. 2017. Dissertação (Mestrado em Advanced Software Development) – Technological University Dublin, Dublin, 2017. Disponível em: <https://arrow.tudublin.ie/scschcomdis/94/>

MATTOS, Antonio Carlos M. O impacto do computador na empresa. *Revista de Administração de Empresas*, São Paulo, v. 18, 1978. Disponível em: <https://www.scielo.br/j/rae/a/fVyx3bwFc9RPGRXp6thMp5x/>

MUNAIHAH, Nuthan et al. Curating GitHub for engineered software projects. *Empirical Software Engineering*, v. 22, n. 6, p. 3219–3253, dez. 2017. Disponível em: <https://doi.org/10.1007/s10664-017-9512-6>

NIEDERAUER, Juliano. *Desenvolvendo Websites com PHP: aprenda a criar Websites dinâmicos e interativos com PHP e bancos de dados*. 3. ed. São Paulo: Novatec, 2017. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=ODM5DwAAQBAJ>

PARK, Joyce. *PHP: a Bíblia*. 2. ed. Rio de Janeiro: Campus, 2003.

PRESSMAN, Roger S. *Engenharia de software: uma abordagem profissional*. 7. ed. Porto Alegre: AMGH, 2011. Disponível em:

[https://www.kufunda.net/publicdocs/Engenharia%20de%20Software%20-%207.ed.%20\(Roger%20S.%20Pressman\).pdf](https://www.kufunda.net/publicdocs/Engenharia%20de%20Software%20-%207.ed.%20(Roger%20S.%20Pressman).pdf)

SOARES, Michel dos Santos. Metodologias ágeis Extreme Programming e Scrum para o desenvolvimento de software. *Revista Eletrônica de Sistemas de Informação*, v. 3, n. 1, jun. 2004. Disponível em:

<https://periodicosibepes.org.br/index.php/reinfo/article/view/146>

SOMMERVILLE, Ian. *Engenharia de software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011. Disponível em: <https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>

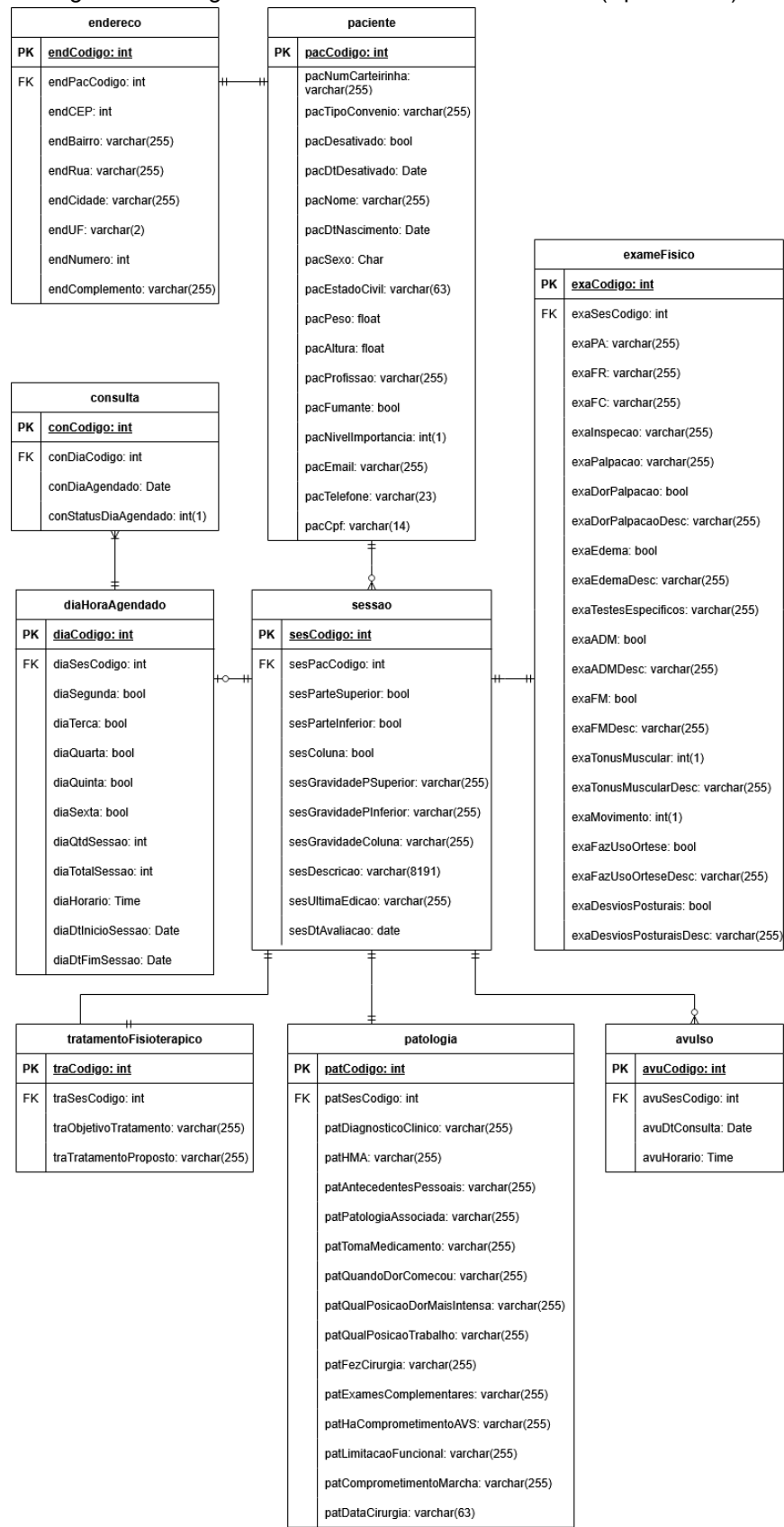
SOUZA, Marcelo Rodrigues de. *Sistema de controle financeiro*. 2017. 43 f. Trabalho de Graduação (Tecnologia em Análise e Desenvolvimento de Sistemas) – Faculdade de Tecnologia Americana – Ministro Ralph Biasi, Americana, 2017. Disponível em: https://ric.cps.sp.gov.br/bitstream/123456789/1906/1/20172S_SOUZAMarceloRodriguesde_OD0236.pdf

TAIPALUS, Toni. Database management system performance comparisons: a systematic literature review. *The Journal of Systems and Software*, v. 208, e111872, jun. 2024. Disponível em: <https://doi.org/10.1016/j.jss.2023.111872>

ZAGALSKY, Alexey et al. The emergence of GitHub as a collaborative platform for education. In: ACM CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 2015, Vancouver. *Proceedings...* New York: ACM, 2015. p. 1906–1917. Disponível em: <http://dx.doi.org/10.1145/2675133.2675284>

APÊNDICE A - Imagem do Diagrama de Entidade-Relacionamento.

Figura 14 - Diagrama de Entidade-Relacionamento (Apêndice A).



Autor: Autoria própria (2025).

O diagrama de entidade-relacionamento representa o banco de dados do sistema. O funcionamento principal do banco de dados se encontra da seguinte forma: A entidade paciente registra os dados dos pacientes na clínica, sessões registra qual a parte afetada do paciente, diaHoraAgendado registra os dias da semana que o paciente pode comparecer para realizar as sessões e consulta representa todas as sessões do paciente organizados em dias. Representa-se as entidades restantes: Avulso registra os dias que um paciente aparece na clínica além do que foi programado em consulta. TratamentoFisioterapico, exameFisico, patologia e endereco são dados extras requeridos do paciente pela clínica.

APÊNDICE B - Respostas das perguntas realizadas nas entrevistas.

Pergunta 1: Quais são as funções primárias necessárias para o agendamento e controle de sessões dos pacientes?

R: O sistema deve ter a capacidade de agendar e rastrear os horários principais dos clientes. Para pacientes com sessões registradas, deve ser possível registrar o comparecimento (se a pessoa foi no dia ou não) e descontar uma sessão se o paciente veio, ou não descontar se ele faltou.

Pergunta 2: Como o sistema deve lidar com a flexibilidade de horários e pacientes avulsos (que vem em horários fora do que foi)?

R: Os horários fixos de um paciente (por exemplo, três vezes por semana) devem ser mantidos no mesmo horário todos os dias em que ele vier, até o final das sessões. No entanto, o sistema precisa permitir edições e flexibilidade, pois alguns pacientes podem precisar alterar os dias de comparecimento semanalmente. Pacientes que não vierem em horários convencionais de suas consultas (avulsos) devem ser registrados para marcar que compareceram em um dia extra, assim descontando de suas sessões.

Pergunta 3: Quais são as regras de disponibilidade para agendamentos?

R: O sistema deve impedir que duas **avaliações** sejam marcadas no mesmo horário. Contudo, é permitido agendar uma avaliação e uma consulta regular simultaneamente. Em relação às consultas (atendimentos), é possível ter múltiplas consultas no mesmo horário (o número pode ser alto, como até oito, mas preferencialmente sem limite definido).

Pergunta 4: Que tipos de pacientes o sistema deve diferenciar?

R: O sistema deve separar e identificar pacientes por convênio (os "normais"), pacientes "especiais" (como os de academia), e avaliação que sempre serão aqueles que chegarem pela primeira vez na clínica. Para quem for "avaliação", deixe de uma forma diferente no sistema para que a gente identifique mais facilmente.

Pergunta 5: Quais informações devem ser registradas no cadastro e na avaliação do cliente?

R: O registro começa quando o cliente liga para marcar a avaliação, capturando nome, convênio e o horário/período desejado. O sistema deve incluir a ficha de avaliação digitalmente (que pode ser baseada no PDF que a clínica já usa), eliminando a necessidade de impressão. Essa ficha deve registrar o problema do cliente (membro superior, inferior ou coluna), o motivo e o objetivo do tratamento, e o histórico de saúde (como diabetes ou outros problemas). Deve ser possível registrar múltiplas patologias e detalhar o local do problema (e.g., punho). Dados adicionais obrigatórios são a data de nascimento e o número da carteirinha do convênio.

Pergunta 6: Como deve ser a visualização da agenda diária e do calendário?

R: A agenda diária deve fornecer uma visualização ampla dos pacientes que virão no dia. A tela deve ser separada por períodos: Manhã, Tarde e Noite, mostrando a quantidade de clientes em cada período. A agenda não precisa separar os clientes por patologia, mas deve mostrar a informação do problema de cada um junto ao agendamento. No calendário geral, deve ser possível clicar no dia e visualizar os nomes, horários e distinguir visualmente (por cores) quais pacientes estão em avaliação.

Pergunta 7: Quais filtros e acessos são necessários para a gestão de pacientes?

R: O sistema deve ter filtros de busca (por nome e por convênio) para facilitar a localização dos pacientes. Deve haver um botão proeminente (sugerido como "Ficha do Paciente") que abre o acesso a todos os dados cadastrados, incluindo convênio, horários e telefone para contato.

Pergunta 8: Como deve funcionar o registro de comparecimento diário?

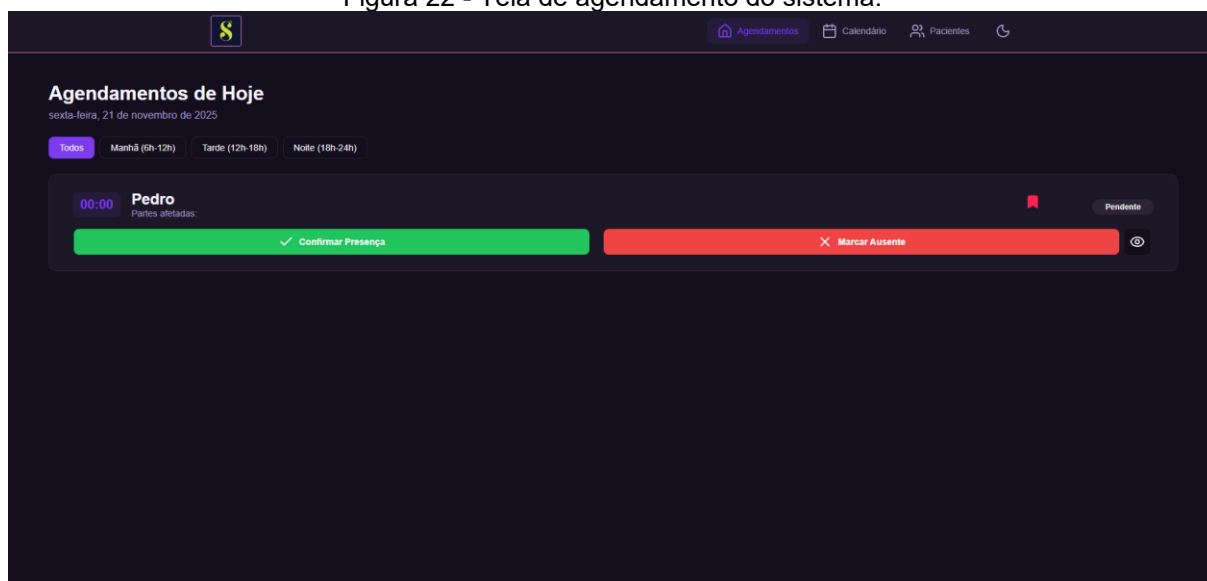
R: O registro de comparecimento deve ser rápido, não exigindo mais de dois cliques. A interação ideal seria com opções de "Confirmar" e "Ausentar" o paciente.

Pergunta 9: Qual é o procedimento para exclusão e arquivamento de pacientes inativos?

R: Quando um cliente não for mais relevante, o sistema deve gerar um PDF da ficha completa para que a clínica possa armazená-lo fisicamente, cumprindo a lei que exige a guarda dos arquivos por até 3 anos. Após o descadastro, o usuário deve ser mantido no sistema por até 6 meses para permitir a reabilitação rápida caso ele retorne.

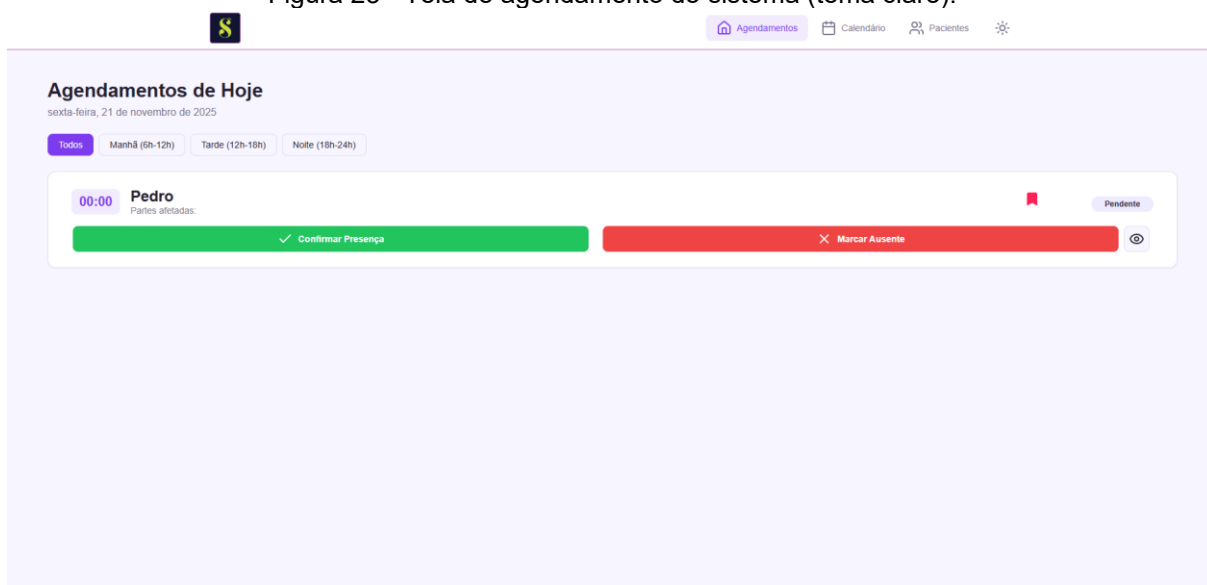
APÊNDICE C - Imagens do sistema.

Figura 22 - Tela de agendamento do sistema.



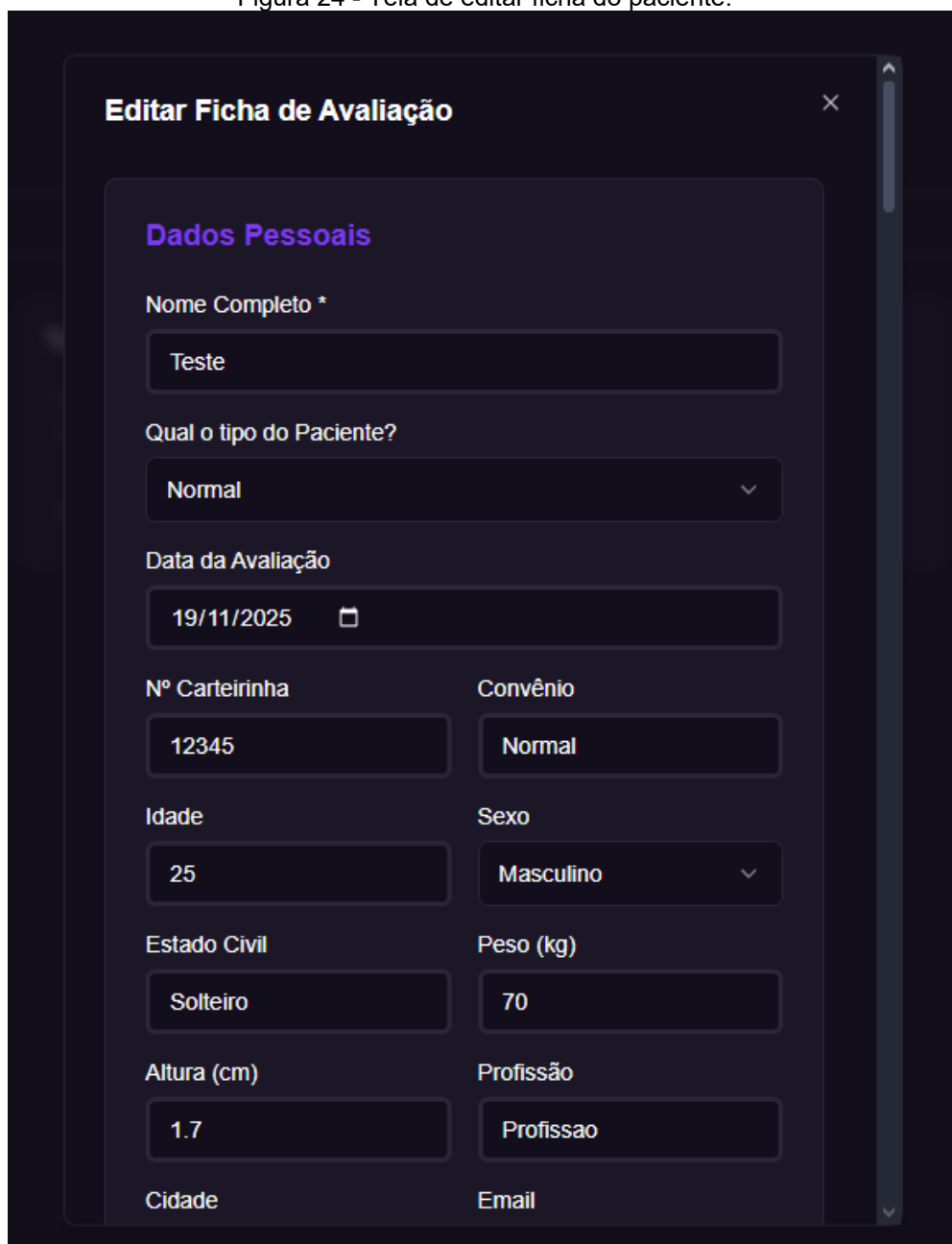
Autor: Autoria própria (2025).

Figura 23 - Tela de agendamento do sistema (tema claro).



Autor: Autoria própria (2025).

Figura 24 - Tela de editar ficha do paciente.



Editar Ficha de Avaliação ✕

Dados Pessoais

Nome Completo *

Teste

Qual o tipo do Paciente?

Normal ▾

Data da Avaliação

19/11/2025 📅

Nº Carteirinha

12345

Convênio

Normal

Idade

25

Sexo

Masculino ▾

Estado Civil

Solteiro

Peso (kg)

70

Altura (cm)

1.7

Profissão

Profissao

Cidade

Email

Autor: Autoria própria (2025).

Figura 25 - Comando para adicionar o usuário no sistema.

```

try{
    // ----- PACIENTE -----
    $addPacienteStmt = $conn->prepare(query: '
        INSERT INTO paciente (
            pacNumCarteirinha, pacTipoConvenio, pacDesativado, pacDtDesativado,
            pacNome, pacDtNascimento, pacSexo, pacEstadoCivil, pacPeso, pacAltura,
            pacFumante, pacNivelImportancia, pacEmail, pacTelefone, pacCpf
        ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
    ');

    $weight = normalizeFloat(value: $allInfo['weight']);
    $height = normalizeFloat(value: $allInfo['height']);

    $addPacienteStmt->execute(params: [
        $allInfo['cardNumber'],
        $allInfo['agreement'],
        0, // pacDesativado default
        null, // pacDtDesativado
        $allInfo['name'],
        $allInfo['birthDate'],
        $allInfo['gender'],
        $allInfo['maritalStatus'],
        $weight,
        $height,
        $allInfo['smoker'] ? 1 : 0,
        2, // sempre criar usuario como Avaliacao
        $allInfo['email'],
        $allInfo['phone'],
        $allInfo['cpf']
    ]);
}

```

Autor: Autoria própria (2025).

Figura 26 - Tela de informações do paciente.

Teste

Informações de Contato

EMAIL
teste@teste.com

TELEFONE
(00)0000-0000

Dados Pessoais

CPF
123.456.789-01

DATA DE NASCIMENTO
01/01/2000

ENDEREÇO
Rua Nome Rua, Nome do
Bairro - nº 2 (Nome do
Complemento (se houver)) |
CEP: 11111

Dados da consulta

HORÁRIO DAS SESSÕES
00:00

DATA DE INICIO
2025-11-19

NÚMERO DE SESSÕES
13

DATA DA AVALIAÇÃO
2025-11-19

SEGUNDA-FEIRA
0

TERÇA-FEIRA
1

QUARTA-FEIRA
1

QUINTA-FEIRA
0

SEXTA-FEIRA
1


Informações Adicionais

NÚMERO DA CARTEIRINHA

CONVÊNIO

Autor: Autoria própria (2025).

Figura 27 - Tela de pacientes.



Agendamentos

Calendário

Pacientes

Pacientes

2 pacientes cadastrados

Novo Paciente

Buscar paciente por Nome, Convênio ou CPF

Paciente A

pacienteA@gmail.com

(11)11111-1111

CPF: 999.999.999-99

Teste

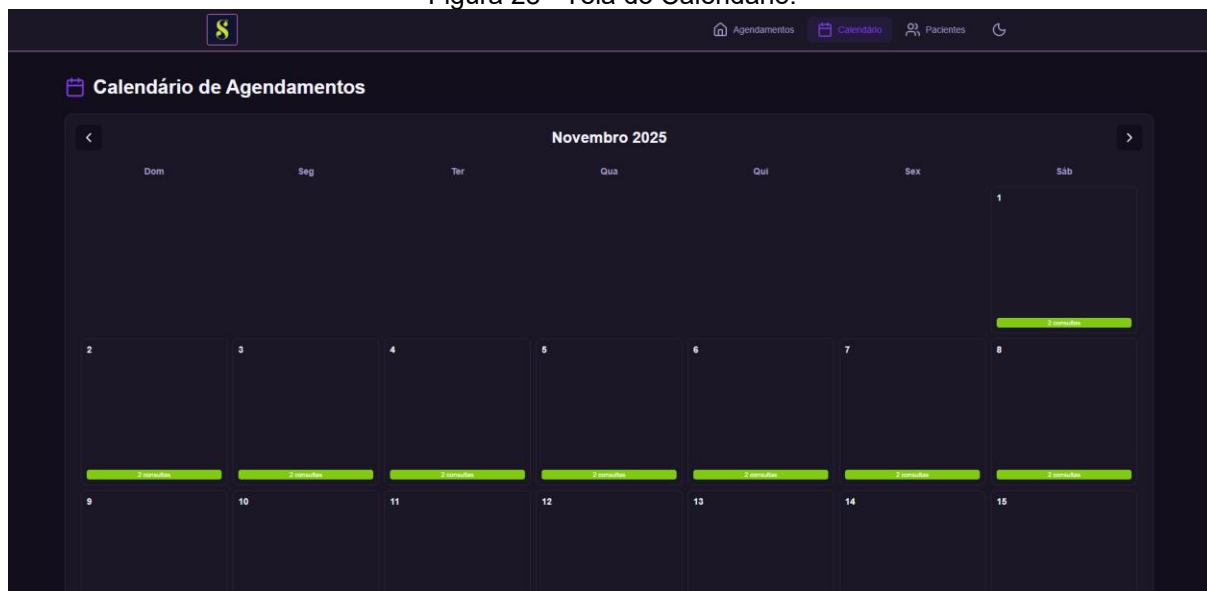
teste@teste.com

(00)0000-0000

CPF: 123.456.789-01

Autor: Autoria própria (2025).

Figura 28 - Tela do Calendário.



Autor: Autoria própria (2025).

Figura 29 - PDF da ficha do paciente

FICHA DE AVALIAÇÃO DE FISIOTERAPIA

Autorização nº: _____ Data avaliação: 19/11/2025

Dados pessoais:

Nome: _____ Sobrenome: _____ Carteira: 12345
 Idade: 25 Sexo: Outro Estado civil: Solteiro Peso: 70
 Altura: 1,7
 Endereço: Rua _____ Nº: 2 Nome do Bairro: _____ Nome do Complemento (se houver): _____
 Cidade: _____ Telefone: (00) 0000-0000
 Fumante: Sim

Dados da patologia:

Diagnóstico clínico: _____
 ICD-10: _____
 Antecedentes pessoais? _____
 Patologia associada? _____
 Tem algum medicamento? _____
 Quando começou a dor? _____
 Em qual posição a dor é mais intensa? _____
 Qual a sua profissão de trabalho? _____
 Faz alguma cirurgia? _____ Data: 00/00/0000

Exames complementares:

Há comprometimento das AVS's? _____
 Limitação funcional: _____
 Comprometimento da marcha? _____

Exame físico:

PA: _____ FR: _____ FC: _____
 Inspeção: _____
 Palpação: _____
 Dor a palpção: ☐ Sim ☐ Não
 Edema: ☐ Sim ☐ Não
 Testes específicos: _____

ADM: ☐ Normal ☐ Diminuída
FM: ☐ Normal ☐ Diminuída
Tônus Muscular: ☐ Normal ☐ Hipotônico ☐ Hipertônico
Movimento: ☐ Abduz ☐ Passivo ☐ Abduz-Assistido
Faixa de dor: ☐ Sim ☐ Não Qual? _____
Devia Postural: ☐ Sim ☐ Não

Tratamento fisioterápico

Objetivos do tratamento: _____
 Tratamento proposto: _____

Assinatura fisioterapeuta: _____ Assinatura beneficiário: _____

Imprimir 1 página

Destino: Salvar como PDF

Páginas: Todos

Layout: Retrato

Mais opções

Salvar Cancelar

Autor: Autoria própria (2025).

Figura 30 - Filtro.

Buscar paciente por Nome, Convênio ou CPF

Autor: Autoria própria (2025).

APÊNDICE D - Link para github do projeto.

Link para github: <https://github.com/GustavoVieiraSousa/Projeto-Clinica-Sange.git>