



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

FELIPE BOMTEMPO

**TECHMEDICINE: SISTEMA DE GERENCIAMENTO PARA CLÍNICAS
MÉDICAS**

**Assis/SP
2022**



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

FELIPE BOMTEMPO

**TECHMEDICINE: SISTEMA DE GERENCIAMENTO PARA CLÍNICAS
MÉDICAS**

Projeto de pesquisa apresentado ao curso de Análise e Desenvolvimento de Sistema do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): Felipe Bomtempo

Orientador(a): Prof. Dr. Luiz Ricardo Begosso

**Assis/SP
2022**

Dados Internacionais de Catalogação na Publicação (CIP)

B696t Bomtempo, Felipe.

Techmedicine: sistema de gerenciamento para clínicas médicas / Felipe Bomtempo – Assis, SP: FEMA, 2022.

74 f.

Trabalho de Conclusão de Curso (Graduação) – Fundação Educacional do Município de Assis – FEMA, curso de Análise e Desenvolvimento de Sistemas, Assis, 2022.

Orientador: Prof. Dr. Luiz Ricardo Begosso.

1. Sistema WEB. 2. Clínica Médica. 3. Informatização. I. Título.

CDD 006.786

Biblioteca da FEMA

Ficha catalográfica elaborada de acordo com os dados fornecidos pelo(a) autor(a).

TECHMEDICINE: SISTEMA DE GERENCIAMENTO PARA CLÍNICAS MÉDICAS

FELIPE BOMTEMPO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador:

Prof. Dr. Luiz Ricardo Begosso

Examinador:

Prof. Me. Douglas Sanches da Cunha

AGRADECIMENTOS

Agradeço primeiramente a minha família, que sempre acreditou no meu potencial e desde o começo me deu total apoio nessa jornada, sendo muito compreensiva em relação a atitudes e decisões que tomei pensando na finalização desse trabalho.

Aos amigos que estiveram ao meu lado em todos os momentos do curso, sempre me incentivando a dar meu melhor, me levantando nos momentos difíceis e se dispondo a me ajudar no que fosse preciso.

Ao professor e meu orientador Dr. Luiz Ricardo Begosso, que em todo o período de realização do trabalho se preocupou em como estava o andamento do mesmo e se dispôs a me prestar auxílio, fazendo correções e comentários relevantes que foram de grande importância.

“Os grandes sucessos vêm de ter a liberdade para falhar.”

Mark Zuckerberg

RESUMO

Ao longo das últimas décadas, a internet e os dispositivos eletrônicos, como computadores, vem experienciando contínuos avanços. Com base nisso, diversas organizações buscaram implementar tais tecnologias em seus negócios e a informatização dominou o meio comercial. Nesse contexto, o presente trabalho tem como objetivo implementar um sistema WEB voltado para clínicas médicas, com o objetivo de melhorar o serviço oferecido, proporcionando mais agilidade e qualidade nos processos realizados. Inicialmente, foi realizada toda a documentação do sistema, destacando as fases elicitação de requisitos e de detalhamento e confecção dos casos de uso, atores e diagramas. A documentação foi importante para delimitar o escopo do projeto e facilitou o processo de desenvolvimento do mesmo. Para o desenvolvimento, foram utilizados os frameworks Spring Boot e Angular, de *back-end* e *front-end*, respectivamente. Ao término do trabalho, o resultado obtido foi satisfatório e o *software* atendeu aos requisitos levantados inicialmente. As tecnologias escolhidas foram de grande importância para o sucesso do projeto e se mostraram muito eficientes em todos os aspectos.

Palavras-chave: sistema WEB; clínica médica; informatização; Spring Boot; Angular.

ABSTRACT

Over the past decades, the internet and electronic devices such as computers have experienced continuous advances. Based on this, several organizations have sought to implement such technologies in their businesses, and computerization has dominated the commercial environment. In this context, the present work aims to implement a WEB system directed to medical clinics, with the objective of improving the service offered, providing more agility and quality in the processes performed. Initially, the entire system documentation was carried out, encompassing the phases of requirements elicitation and detailing and preparation of use cases, actors, and diagrams. The documentation was important to delimit the project's scope and facilitated the project's development process. For the development, the Spring Boot and Angular frameworks were used, for back-end and front-end, respectively. At the end of the work, the result obtained was satisfactory and the software met the requirements initially raised. The chosen technologies were of great importance for the success of the project and proved to be very efficient in all aspects.

Keywords: WEB system; medical clinic; informatization; Spring Boot; Angular.

LISTA DE ILUSTRAÇÕES

Figura 1: Estrutura Analítica do Projeto	26
Figura 2: Mapa Mental dos Requisitos.....	28
Figura 3: Diagrama de Caso de Uso.....	29
Figura 4: Diagrama UC Fazer Login	30
Figura 5: Diagrama UC Manter Pacientes	31
Figura 6: Diagrama UC Manter Médicos.....	33
Figura 7: Diagrama UC Manter Funcionários	35
Figura 8: Diagrama UC Manter Cargos	37
Figura 9: Diagrama UC Manter Especialidades	39
Figura 10: Diagrama UC Manter CIDs	41
Figura 11: Diagrama UC Manter Usuários.....	43
Figura 12: Diagrama UC Manter Agendamentos	45
Figura 13: Diagrama UC Manter Atendimentos	47
Figura 14: Diagrama UC Realizar Atendimento	49
Figura 15: Diagrama UC Visualizar Prontuário Eletrônico do Paciente	50
Figura 16: Diagrama UC Emitir Relatório de Atendimentos.....	52
Figura 17: Diagrama UC Emitir Atestado Médico	53
Figura 18: Diagrama UC Emitir Receita Médica	54
Figura 19: Diagrama UC Emitir Receita Médica	55
Figura 20: Diagrama de Classes	57
Figura 21: Diagrama Entidade-Relacionamento	58
Figura 22: Página de Login.....	59
Figura 23: Página Inicial	60
Figura 24: Página de Cadastro de Especialidades	61

Figura 25: Página de Inserção de Especialidades.....	62
Figura 26: Página de Visualização de Especialidades.....	63
Figura 27: Página de Exclusão de Especialidades	64
Figura 28: Calendário de Realização de Agendamentos.....	65
Figura 29: Página de Prontuário Eletrônico do Paciente	66
Figura 30: Página do Paciente.....	67
Figura 31: Página de Relatórios	68
Figura 32: Página do Administrador.....	69

LISTA DE TABELAS

Tabela 1: Narrativa UC Fazer Login	31
Tabela 2: Narrativa UC Manter Pacientes	33
Tabela 3: Narrativa UC Manter Médicos.....	35
Tabela 4: Narrativa UC Manter Funcionários.....	37
Tabela 5: Narrativa UC Manter Cargos.....	38
Tabela 6: Narrativa UC Manter Especialidades	40
Tabela 7: Narrativa UC Manter CIDs	42
Tabela 8: Narrativa UC Manter Usuários	44
Tabela 9: Narrativa UC Manter Agendamentos	46
Tabela 10: Narrativa UC Manter Atendimentos	48
Tabela 11: Narrativa UC Realizar Atendimento	50
Tabela 12: Narrativa UC Visualizar Prontuário Eletrônico do Paciente.....	51
Tabela 13: Narrativa UC Emitir Relatório de Atendimentos	52
Tabela 14: Narrativa UC Emitir Atestado Médico	53
Tabela 15: Narrativa UC Emitir Receita Médica	55
Tabela 16: Narrativa UC Emitir Exames Solicitados	56

SUMÁRIO

1. INTRODUÇÃO	14
1.1. OBJETIVOS	15
1.2. JUSTIFICATIVA	15
1.3. MOTIVAÇÃO.....	16
1.4. PERSPECTIVAS DE CONTRIBUIÇÃO	16
1.5. METODOLOGIA.....	17
1.6. ESTRUTURA DO TRABALHO.....	17
2. METODOLOGIAS E FERRAMENTAS UTILIZADAS	18
2.1. UML.....	18
2.2. ASTAH	19
2.3. DBDESIGNER.....	19
2.4. EDRAWMAX	19
2.5. FREEMIND.....	20
2.6. JAVA.....	20
2.7. SPRING BOOT	20
2.8. SPRING TOOL SUITE	21
2.9. HTML5.....	21
2.10. CSS3	22
2.11. JAVASCRIPT	22
2.12. TYPESCRIPT	22
2.13. ANGULAR	23
2.14. VISUAL STUDIO CODE.....	23
2.15. POSTGRESQL.....	24
2.16. HEROKU	24
3. DESENVOLVIMENTO DO TRABALHO	25
3.1. ESTRUTURA ANALÍTICA DO PROJETO	25
3.2. REQUISITOS DO SISTEMA	26
3.2.1. Levantamento de Requisitos	26
3.2.2. Mapa Mental.....	27
3.3. CASOS DE USO	28

3.3.1.	Atores.....	28
3.3.2.	Diagrama de Caso de Uso	29
3.4.	NARRATIVAS DOS CASOS DE USO	30
3.4.1.	Fazer Login.....	30
3.4.2.	Manter Pacientes.....	31
3.4.3.	Manter Médicos.....	33
3.4.4.	Manter Funcionários.....	35
3.4.5.	Manter Cargos	37
3.4.6.	Manter Especialidades.....	39
3.4.7.	Manter CIDs	41
3.4.8.	Manter Usuários	43
3.4.9.	Manter Agendamentos.....	45
3.4.10.	Manter atendimentos	47
3.4.11.	Realizar Atendimento	49
3.4.12.	Visualizar Prontuário Eletrônico do Paciente.....	50
3.4.13.	Emitir Relatório de Atendimentos	52
3.4.14.	Emitir Atestado Médico	53
3.4.15.	Emitir Receita Médica.....	54
3.4.16.	Emitir Exames Solicitados	55
3.5.	DIAGRAMA DE CLASSES.....	56
3.6.	DIAGRAMA ENTIDADE-RELACIONAMENTO	57
4.	INTERFACES DO SISTEMA.....	59
4.1.	PÁGINA DE LOGIN.....	59
4.2.	PÁGINA INICIAL	60
4.3.	PÁGINAS DE CADASTRO	60
4.4.	PÁGINAS DE FORMULÁRIOS DE CADASTRO	61
4.5.	PÁGINAS DE VISUALIZAÇÃO DE DADOS	62
4.6.	PÁGINAS DE EXCLUSÃO DE DADOS.....	63
4.7.	PÁGINAS DE REALIZAÇÃO DE AGENDAMENTOS	64
4.8.	PÁGINA DE PRONTUÁRIO DOS PACIENTES.....	65
4.9.	PÁGINA DO PACIENTE	66
4.10.	PÁGINA DE RELATÓRIOS.....	67
4.11.	PÁGINA DO ADMINISTRADOR	68
5.	CONSIDERAÇÕES FINAIS	70

5.1. TRABALHOS FUTUROS	70
REFERÊNCIAS.....	72

1. INTRODUÇÃO

Com o surgimento e avanços dos computadores e da internet, um leque enorme de possibilidades foi aberto e trouxe consigo vários benefícios à sociedade, como a informatização de diversos serviços. Neste cenário, as organizações rapidamente começaram a buscar implementar tais novas tecnologias em seus ambientes, visando conseguirem alcançar um melhor desempenho e uma maior competitividade em suas respectivas áreas.

A aplicação de Tecnologia de Informação (TI) nas organizações com o intuito de melhora no funcionamento implica em um maior benefício no desempenho geral delas, proporcionando vantagens competitivas como aumento de mercado, aumento da produtividade e aumento das vendas (LUNARDI; DOLCI; MAÇADA, 2010). Portanto, a constante busca por informatização se mostra justificada e de grande impacto socioeconômico.

No âmbito da saúde, a aplicação da TI se mostra presente através dos Sistemas de Informação de Saúde (SIS). Em suma, um SIS tem por objetivo coletar, processar, armazenar e distribuir informações que auxiliem no controle da instituição de saúde onde foi implementado (MARIN, 2010). De acordo com Fornazin e Joia (2015 apud. OLIVEIRA; FLEURY, 1989; MORAES, 1994), os primeiros SIS no Brasil surgiram na década de 1970, e tinham como objetivo realizar o controle do faturamento apresentado por prestadores de saúde da época.

Entre os benefícios proporcionados pela utilização de um SIS, pode-se citar uma melhora no acesso às informações, na eficiência do atendimento, na gestão da instituição e nas cobranças, além de também prover um aumento na produtividade dos profissionais, redução de erros médicos e redução em gastos com registros médicos em papel (KHALIFA, 2017).

Fica evidente o quão valioso é a implementação da TI através dos SIS nas organizações de saúde. Entretanto, vivencia-se um cenário onde é comum encontrar clínicas e consultórios que não são informatizados, funcionando com base em registros físicos. Esse tradicional método de armazenamento de informações implica em algumas deficiências, como ocorrência de subnotificação e perda de dados (VALVERDE; IDROGO; ALVA, 2015).

Além disso, um estudo conduzido por Ayaad et al. (2017) mostra que a qualidade do serviço de hospitais que utilizam prontuário eletrônico é superior a aqueles que não o adotam.

Na maioria dos casos, essa falta de informatização pode ser atrelada às instituições de pequeno ou médio porte, que não veem necessidade na utilização de um SIS por julgarem não valer a pena financeiramente. Muitos dos sistemas existentes no mercado são voltados para hospitais de grande porte, e contém funções que acabam sendo inutilizadas em instituições menores. Portanto, vê-se apropriado o desenvolvimento de um *software* simplificado e funcional, voltado para tais clínicas supracitadas.

1.1. OBJETIVOS

Objetiva-se com este trabalho realizar a construção de um sistema WEB capaz de atender às necessidades diárias dessas instituições de saúde. Caberá ao sistema realizar a gerência de informações, permitindo o cadastro, alteração, exclusão e visualização de entidades presentes nesses ambientes e ainda possibilitando o controle de processos, como agendamentos e atendimentos dos pacientes. Por fim, busca-se implementar um prontuário eletrônico, para organizar o histórico de pacientes e eliminar a necessidade de milhares de registros físicos.

1.2. JUSTIFICATIVA

O presente trabalho é justificado pois beneficia tanto a instituição de saúde quanto a população em geral. Primeiramente, a premissa do trabalho é de desenvolver um sistema WEB funcional para clínicas médicas. Esse tipo de orientação de desenvolvimento apresenta uma maior praticidade e flexibilidade de implantação se comparada com a clássica orientação *desktop*, que é limitada pela plataforma onde o *software* foi desenvolvido e pode conflitar com outros tipos de sistemas. As aplicações WEB por sua vez, precisam apenas de conexão com a internet e um navegador para acessar o software.

Outro ponto a ser observado é o impacto para a sociedade proporcionado por este trabalho. Com a construção do sistema, as clínicas que possuem baixo índice de informatização

poderão implantá-lo em seus ambientes. Como descrito na introdução, a adoção de um sistema em um ambiente de saúde garante vários benefícios para a instituição. Dessa forma, o sistema vai contribuir para melhorar a produtividade, podendo garantir resultados positivos não só para a instituição em si, mas também para os pacientes, que teriam a suas disposições serviços médicos mais eficientes.

1.3. MOTIVAÇÃO

A motivação do trabalho surge da necessidade de implementação de sistemas de informação em instituições de saúde, especificamente clínicas médicas. Nos dias atuais ainda é comum encontrar clínicas que operam de maneira antiquada, isto é, utilizando de recursos físicos como documentos em papel, pranchetas e outros recursos semelhantes para o gerenciamento do ambiente. Tal abordagem implica em algumas desvantagens para a instituição, como o volume ocupado no armazenamento de registros físicos e a questão de inconsistência dos dados, que por muitas vezes podem apresentar erros ou mesmo serem perdidos.

1.4. PERSPECTIVAS DE CONTRIBUIÇÃO

Espera-se que o presente trabalho contribua positivamente para o bom funcionamento da área de saúde, com foco voltado para as instituições de pequeno e médio porte. A premissa do produto desenvolvido é de contornar os problemas citados em relação a utilização de recursos físicos no gerenciamento das clínicas médicas. Dessa forma, aumentando a produtividade e qualidade do serviço oferecido e beneficiando ambas as instituições quanto a população que necessita de tais serviços.

Ainda, é esperado que o trabalho contribua e possa servir como base para futuros desenvolvimentos voltados para essa área de pesquisa, visto que o sistema construído terá algumas das principais atividades presentes em ambientes de saúde, como realização de agendamentos e atendimentos e implementação do prontuário eletrônico do paciente, recurso de grande valia em SIS atuais.

1.5. METODOLOGIA

Para que o desenvolvimento do trabalho se torne possível, primeiramente serão consultadas pessoas com conhecimento na área de pesquisa, com a finalidade de levantar alguns dos requisitos mais importantes que devem fazer parte de um SIS. Após esse levantamento, será prosseguido para a parte de construção do *software*.

Na fase de modelagem do *software*, serão confeccionados alguns diagramas que serão apresentados ao longo do trabalho e auxiliarão na compreensão da funcionabilidade do sistema. São eles o digrama de casos de uso, o diagrama de classes e o diagrama entidade-relacionamento (DER). No que diz respeito ao desenvolvimento do código, será utilizada a abordagem de Orientação a Objetos (OO).

Para realizar a criação dos diagramas de casos de uso e de classes, será utilizada a linguagem de modelagem UML. De modo a trabalhar com a UML, será utilizado o *software* de modelagem Astah. Por fim, a criação do DER irá ser feita através do *software* DBDesigner.

No tocante ao desenvolvimento do sistema, a linguagem de programação a ser empregada para o *back-end* é o Java, em conjunto com o *framework* Spring Boot, utilizando a IDE Spring Tool Suite. O *front-end* do sistema ficará sob responsabilidade do *framework* Angular, baseado em TypeScript. Como plataforma de desenvolvimento do Angular, será utilizado o editor de códigos Visual Studio Code. Por último, no quesito de banco de dados, será empregado o sistema gerenciador de banco de dados (SGBD) PostgreSQL.

1.6. ESTRUTURA DO TRABALHO

O presente trabalho está estruturado da seguinte forma: o Capítulo 1 é esta introdução, que apresenta uma visão geral do tema e do produto. No Capítulo 2 serão apresentadas as metodologias e ferramentas utilizadas para o desenvolvimento do sistema. O Capítulo 3 apresentará os requisitos e a modelagem do software a ser desenvolvido. O Capítulo 4 apresentará as interfaces do software e a descrição do funcionamento dos módulos. O Capítulo 5 descreverá as conclusões obtidas com o desenvolvimento do projeto e as propostas para trabalhos futuros, seguido pelas referências bibliográficas utilizadas para o desenvolvimento deste projeto.

2. METODOLOGIAS E FERRAMENTAS UTILIZADAS

Neste capítulo serão apresentadas as metodologias e ferramentas empregadas no desenvolvimento do trabalho, sendo decorridas as linguagens, os *softwares* e conceitos utilizados durante o desenvolvimento.

Como metodologia para a codificação, optou-se por utilizar a abordagem de Orientação a Objetos (OO). Em suma, é uma forma de programação que dá foco à modelagem e aos processos e tarefas realizados pelo sistema, podendo ser caracterizada como uma maneira de representar as necessidades de um *software* de modo mais realista (CARVALHO, 2016).

Para o *back-end*, será empregada a linguagem orientada a objetos Java em conjunto com o framework Spring Boot, para a criação de uma API Rest. No tocante do *front-end*, será utilizado o *framework* Angular, que trabalha com ambas as linguagens TypeScript e HTML. O banco de dados empregado será o PostgreSQL, através de sua licença gratuita. A implantação no sistema será feita na plataforma nuvem Heroku. Por fim, será utilizada a linguagem UML com a criação de diagramas que auxiliam no entendimento do sistema.

Para melhor compreensão dos recursos utilizados, a seguir serão apresentadas as tecnologias anteriormente citadas e uma descrição detalhada sobre suas características.

2.1. UML

Para realizar a criação dos diagramas de caso de uso e de classes, será utilizada a linguagem de modelagem UML – Unified Modeling Language, ou Linguagem de Modelagem Unificada. Conforme Booch, Rumbaugh e Jacobson (2006):

A UML, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. A UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis.

2.2. ASTAH

De modo a trabalhar com a UML, será utilizado o Astah. O Astah é um *software* de modelagem utilizado para trabalhar com a criação de diagramas UML além de outros tipos de diagramas e modelos utilizados durante o projeto de um sistema. Foi criado em 2006, pela empresa japonesa Change Vision.

Uma pesquisa conduzida por Agner e Lethbridge (2017) apontou que o Astah é atualmente uma das ferramentas de modelagem de software mais populares do mercado, sendo amplamente utilizada. Além disso, os resultados desse estudo mostraram que o Astah pode ser considerado como um *software* de baixa complexidade e fácil utilização, isentando o usuário de dificuldades na hora de se trabalhar com a ferramenta e criar diagramas.

2.3. DBDESIGNER

o DBDesigner é uma ferramenta de modelagem de código aberto e com licença gratuita, desenvolvida pela fabFORCE. A aplicação apresenta diversas ferramentas de modelagem que permite a criação de diagramas entidade relacionamento de qualquer tipo de banco de dados, englobando todos os tipos de relações entre as tabelas. Além disso, possui outras funcionalidades que auxiliam na criação de banco de dados, como engenharia reversa que permite gerar diagramas a partir de bancos já existentes e uma interface onde é possível gerar instruções SQL de consultas de informações.

2.4. EDRAWMAX

De acordo com Edraw Software (2022), o EdrawMax é uma ferramenta de diagramação que permite a criação e customização de mais de 280 tipos de diagramas, que vão desde fluxogramas até plantas de construções civis. Ainda de acordo com a fonte, o software oferece uma série de funcionalidades avançadas — voltadas para o desenvolvimento de diagramas profissionais — em conjunto com interfaces amigáveis, permitindo que usuários de qualquer nível técnico façam sua utilização.

2.5. FREEMIND

Freemind é um aplicativo de código aberto voltado para a criação de mapas mentais, escrito em Java. A ferramenta combina uma fácil usabilidade e interfaces intuitivas, sendo uma excelente e amigável opção para a criação, uso e compartilhamento de mapas mentais (TSINAKOS; BALAFOUTIS, 2009).

Entre suas funcionalidades, permite ao usuário definir o tipo da fonte, cores, tamanhos, entre outras configurações. Ainda, possui as funções de importação e exportação de mapas em diferentes formatos de arquivo, que vão desde imagens à extensão HTML.

2.6. JAVA

Java é uma linguagem de programação de alto nível, orientada a objetos, sendo apresentada em 1995 pela Sun Microsystems. Originalmente utilizado para desenvolver pequenos aplicativos que rodariam na internet, a linguagem sofreu diversas evoluções ao longo do tempo e hoje é considerada uma plataforma de desenvolvimento, a criação de aplicações *desktop*, *mobile*, WEB, entre outros ambientes.

Uma característica importante dessa linguagem é a forma com que as aplicações são executadas. Os algoritmos são interpretados por uma *Java Virtual Machine* (JVM). Desse modo, um código escrito em Java pode ser portado para ambientes com diferentes sistemas operacionais, sendo necessário apenas que o sistema onde o código será executado tenha a JVM propriamente instalada (HORSTMANN, 2009).

2.7. SPRING BOOT

Spring Boot é um *framework* cujo objetivo é de facilitar e agilizar a construção de aplicações Java, no que diz respeito a configurações do projeto. O Spring Boot emergiu como sendo um projeto do Spring, um outro *framework* que tem como função facilitar o desenvolvimento de qualquer aplicação Java. Um dos pontos negativos do Spring é a complexidade em configurar um projeto, onde tarefas como ativar certas funcionalidades ou realizar o controle de dependências envolve extensas configurações feitas em XML ou Java (WALLS,

2015). Com base nisso, o Spring Boot surge como forma de suprir essa deficiência e facilita a configuração de servidores, de dependências, de bibliotecas, entre outras.

2.8. SPRING TOOL SUITE

Spring Tool Suite (STS) é uma IDE de código aberto que facilita o desenvolvimento de aplicações baseadas em Spring e Spring Boot, por possuir integração natural com componentes desses *frameworks*. O STS foi criado com base no Eclipse, uma outra IDE usada para escrever códigos em Java e, portanto, compartilha de diversas características com esse outro.

Entre suas funcionalidades, possibilita realizar a configuração inicial de aplicações Spring Boot de maneira prática e fácil, permite a navegação entre diferentes partes da aplicação, facilitando a busca por partes específicas do projeto, além de possuir um sugestor de código que também conta com a função de completamento de código (SPRING, 2021).

2.9. HTML5

O HTML é uma linguagem de marcação utilizada para realizar a criação de páginas de sites na internet, permitindo que se descreva toda a estrutura de uma página web, desde a parte textual até a parte gráfica (SHARMA; BHARDWAJ, P.; BHADRWAJ, M., 2012). Ao longo dos anos, o HTML sofreu evoluções que culminaram em novas versões a partir da versão original. Para este trabalho, será utilizado o HTML5, quinta versão da linguagem original e que é mantida e desenvolvida até os dias atuais. Essa nova especificação trouxe consigo melhorias, como a adição de novos elementos HTML, utilizados para uma melhor estruturação das páginas, e suporte nativo a áudio e vídeo, permitindo que os navegadores reproduzissem esses elementos multimídia de forma nativa (FREEMAN, 2011).

2.10. CSS3

CSS é uma linguagem utilizada na estilização de documentos web. De acordo com McFarland (2012), O CSS oferece controle criativo em relação ao *design* de páginas web, permitindo o arranjo de imagens com precisão, criação de colunas, bordas, estilização de cores de botões, entre diversas outras customizações gráficas. É uma linguagem específica para se trabalhar em conjunto com linguagens de marcação, como o HTML, permitindo a customização dos elementos definidos na estruturação da página.

Assim como outras linguagens, o CSS possui mais de uma versão. No projeto, será usado o CSS3, que apresenta uma melhor e mais completa estilização em relação as versões anteriores, além de ser suportado em um maior número de navegadores.

2.11. JAVASCRIPT

JavaScript é uma linguagem de programação de alto nível com sintaxe baseada no Java, apropriada para desenvolvimento orientado a objetos, interpretada e não tipada (FLANAGAN, 2004), isto é, não há a definição do tipo das variáveis criadas. Dentro do contexto de desenvolvimento web, o JavaScript é o responsável por dar comportamento aos elementos de uma página web. Exceto pela sintaxe derivada do Java, essa última e JavaScript não possuem mais nenhum tipo de relação, sendo linguagens completamente diferentes uma da outra.

O JavaScript é uma linguagem interpretada, mais especificamente interpretada pelos navegadores utilizados para acessar a internet e, portanto, dizemos que é uma linguagem interpretada no lado do cliente (SILVA, 2010). Hoje em dia, grande parte dos navegadores modernos presentes no mercado possuem um interpretador JavaScript implementado.

2.12. TYPESCRIPT

TypeScript é uma linguagem de programação criada e mantida pela Microsoft com o objetivo primário de permitir o desenvolvimento de aplicações de larga escala com JavaScript. O TypeScript é um superconjunto do JavaScript, que implementa módulos,

técnicas de programação orientada a objetos como classes e interfaces e outras características como tipagem explícita de variáveis, enquanto ainda oferece suporte a práticas de programação do JavaScript (BIERMAN; ABADI; TORGERSEN, 2014), o que torna essa linguagem mais adequada para se usar em projetos grandes e de maior complexidade.

Uma característica dessa linguagem é que a compilação é feita para o JavaScript e, portanto, um código em TypeScript pode ser compreendido e executado por qualquer navegador de internet.

2.13. ANGULAR

Angular é um *framework front-end* que permite o desenvolvimento de interfaces de aplicações *desktop*, *WEB* ou *mobile* utilizando HTML em conjunto com JavaScript ou TypeScript, essa última sendo uma linguagem desenvolvida pela Microsoft baseada no JavaScript. Para este projeto, foi utilizado a versão do *framework* baseada em TypeScript.

Uma das características do Angular é a possibilidade de se trabalhar com *Single Page Applications* (SPA), ou aplicações de página única, em português. SPAs se diferenciam de aplicações WEB convencionais pois nelas há apenas o carregamento de uma única página HTML inicial, que serve como fundação para receber todas as outras interfaces do site, eliminando a ocorrência de recarregamentos completos que aconteceriam ao se acessar uma nova página (FINK; FLATOW, 2014).

2.14. VISUAL STUDIO CODE

Visual Studio Code é um editor de códigos da Microsoft, gratuito e de código aberto. É um editor com muita flexibilidade pois permite trabalhar com diversas linguagens de programação. Além disso, oferece muitas opções de customização para o usuário e, entre elas, admite a adição de extensões no editor. Esse fato mais a natureza de código aberto permitiu que a comunidade realizasse a criação de extensões e, portanto, o editor

atualmente conta com uma vasta gama de extensões disponíveis, com diversas funcionalidades.

O editor ainda conta com a tecnologia IntelliSense, um assistente de código que é capaz de identificar a linguagem que está sendo trabalhada e auxiliar o trabalho do desenvolvedor, através de uma série de funcionalidades. Entre elas, pode-se citar o oferecimento de sugestões de membros para o trecho atual do código, como variáveis e funções, e capacidade de até mesmo autocompletar o restante de uma linha de código (VISUAL STUDIO CODE, 2021).

2.15. POSTGRESQL

PostgreSQL é um banco de dados objeto-relacional de código-fonte aberto e licença de uso gratuita, que teve surgimento no ano de 1986 como parte de um projeto da Universidade da Califórnia em Berkeley, intitulado de POSTGRES. Utiliza a linguagem de banco de dados padrão SQL, porém com pequenas alterações na sintaxe e em algumas funcionalidades. Embora seja gratuito, possui todas as características presentes nos principais bancos de dados que são utilizados no mercado, apresentando um alto teor de confiabilidade e potencial e sendo indicado para ambos fins estudantis ou empresariais (MILANI, 2008).

2.16. HEROKU

Heroku é uma plataforma de computação em nuvem utilizada para realizar a publicação de aplicações WEB, sendo responsável por realizar toda a parte de configuração da implantação de uma aplicação e fazendo com que o desenvolvedor possa voltar seu foco apenas para o desenvolvimento da mesma (HANJURA, 2014). A plataforma disponibiliza uma grande flexibilidade aos desenvolvedores, oferecendo suporte para a hospedagem de aplicações em diferentes linguagens de programação e para bancos de dados relacionais ou não-relacionais. Depois de implantada, todo o gerenciamento da aplicação pode ser feito através de um painel de controle oferecido pela plataforma ou por meio de linhas de comando.

3. DESENVOLVIMENTO DO TRABALHO

Nesse capítulo, serão abordadas algumas etapas-chave executadas durante o desenvolvimento do trabalho e que possibilitaram que o mesmo fosse concluído. O objetivo é de proporcionar uma visão geral dos processos envolvidos na criação do software, contemplando fases como especificação e planejamento do sistema.

3.1. ESTRUTURA ANALÍTICA DO PROJETO

Dentro de gerenciamento de projetos, a Estrutura Analítica do Projeto (EAP) é uma ferramenta que auxilia na determinação do escopo e na organização de um projeto através da divisão do mesmo em diferentes partes menores, de maneira que o gerenciamento desse se torne mais fácil.

A EAP pode ser definida como “uma decomposição hierárquica do escopo total do trabalho a ser realizado pela equipe do projeto para atingir os objetivos do projeto e criar as entregas necessárias” (PROJECT MANAGEMENT INSTITUTE, 2021, tradução do autor)¹. Segundo Heagney (2016), a divisão proposta pela EAP facilita a determinação do tempo gasto em cada tarefa e o custo necessário para realizá-las, evitando falhas no projeto causadas por estimativas errôneas.

¹ A hierarchical decomposition of the total scope of work to be carried out by the project team to accomplish the project objectives and create the required deliverables.

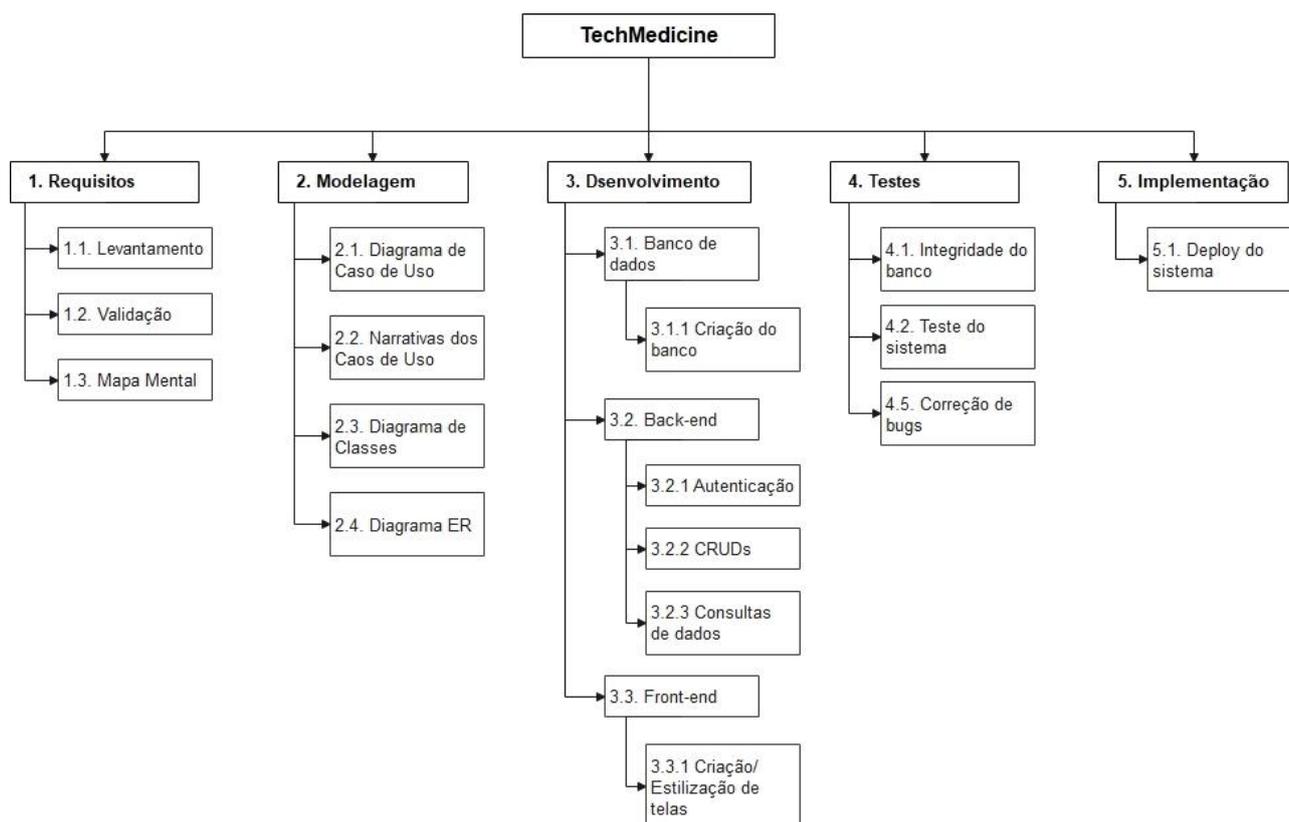


Figura 1: Estrutura Analítica do Projeto

3.2. REQUISITOS DO SISTEMA

3.2.1. Levantamento de Requisitos

A fase de levantamento de requisitos consiste em identificar as necessidades que o sistema deve alcançar para que atinja as expectativas iniciais e resolva os problemas que originaram o projeto.

É importante que nesse momento fique bem claro quais são os objetivos que o sistema deve alcançar a longo prazo. Esses objetivos servem como base para diversas atividades envolvendo o escopo de requisitos, como elaboração, validação e evolução dos mesmos, além de contribuir para a definição da arquitetura do sistema (PRESSMAN; MAXIM, 2021).

Com base nisso, de modo a garantir a eficácia do *software* proposto, foram levantados alguns requisitos que deverão ser cumpridos. São eles:

- Fazer login;

- Manter pacientes;
- Manter médicos;
- Manter funcionários;
- Manter cargos;
- Manter especialidades;
- Manter CIDs;
- Manter usuários;
- Manter agendamentos;
- Manter atendimentos;
- Realizar atendimento;
- Visualizar prontuário eletrônico do paciente;
- Emitir relatório de atendimento;
- Emitir atestado médico;
- Emitir receita médica;

3.2.2. Mapa Mental

Os mapas mentais são diagramas voltados para a organização de ideias, possibilitando a sintetização de um determinado assunto de forma visual e estruturada, a partir de palavras-chave. Entre suas utilidades estão a capacidade de facilitar a expressão de ideias, melhorar a produtividade, aumentar a concentração, entre outras, sendo uma ótima ferramenta para a revisão de conceitos e ideias (PINHEIRO, 2021).

A partir dessas informações, mostra-se pertinente a criação de um mapa mental em relação aos requisitos do sistema. A seguir, é apresentado o mapa mental dos requisitos levantados anteriormente.

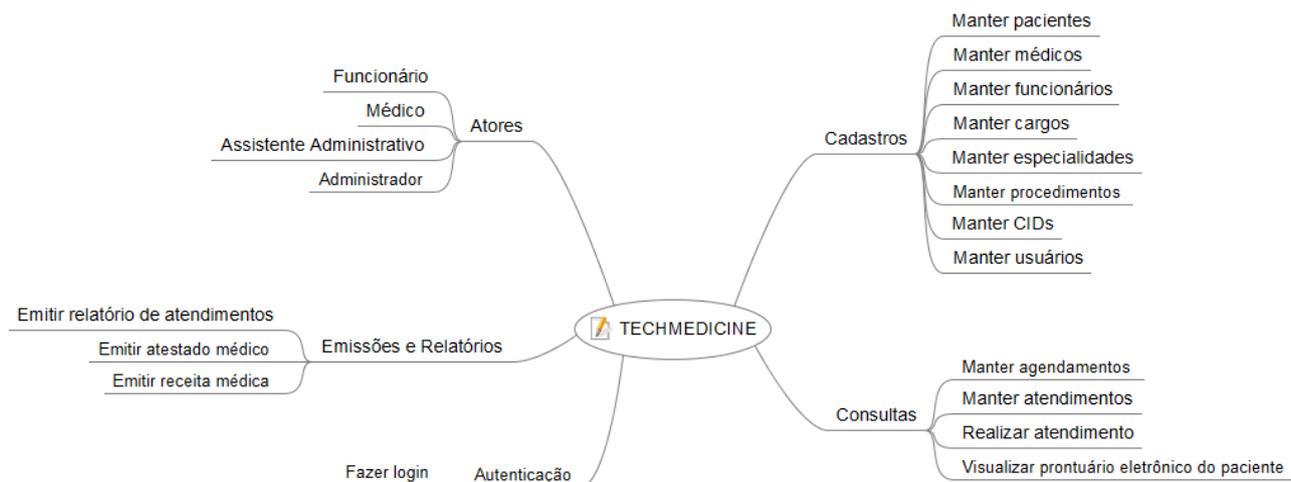


Figura 2: Mapa Mental dos Requisitos

3.3. CASOS DE USO

Os casos de uso são uma maneira de representar as interações existentes entre os usuários e o sistema, descrevendo como os utilizadores interagem com as diferentes funcionalidades do *software* e como ele será utilizado. Segundo Pressman e Maxim (2021), a maneira de se descrever essas interações é variada, podendo ser em formato de um texto narrativo, uma descrição geral das interações, uma representação esquemática, entre outras estruturas. Para esse trabalho, optou-se por explanar os casos de uso através de um diagrama e de narrativas, que serão abordados mais adiante.

3.3.1. Atores

Entende-se por atores as partes que desempenham alguma atividade e interagem diretamente com o sistema em questão. Normalmente esse termo é associado com os usuários finais que irão utilizar o produto, entretanto não se limita a apenas eles, podendo também caracterizar outros *softwares* externos que interajam com o sistema.

Em ordem para definir os casos de uso, primeiramente é necessário estabelecer quais serão os atores do sistema. Nesse projeto, foram identificados os seguintes atores, listados abaixo:

- Recepcionista.

- Médico.
- Assistente Administrativo.
- Administrador;

3.3.2. Diagrama de Caso de Uso

O diagrama de caso de uso é uma representação gráfica fornecida pela linguagem UML que ilustra todo o conjunto de casos de uso do sistema e seus relacionamentos com os atores, englobando pontos como quais atores utilizam cada caso e quais casos incluem outros casos de uso (FOWLER, 2005). A seguir, segue o diagrama de caso de uso confeccionado para esse trabalho.

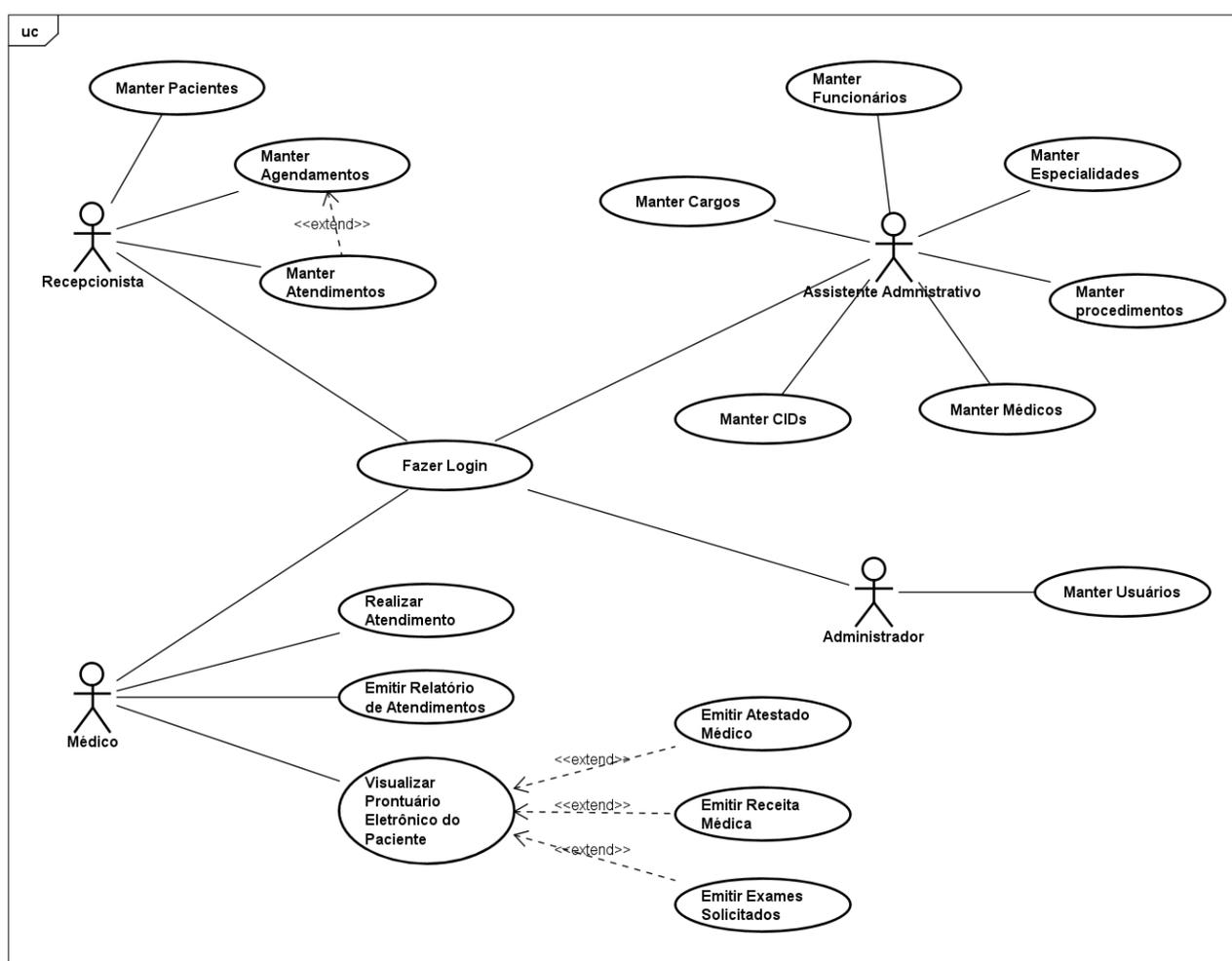


Figura 3: Diagrama de Caso de Uso

3.4. NARRATIVAS DOS CASOS DE USO

Como decorrido no início do capítulo, uma possível maneira de descrever os casos de uso é através de textos narrativos, que contam uma história envolvendo o ator e o caso de uso, descrevendo com detalhes como se dá a o processo de interação entre as partes. A seguir, serão apresentadas as narrativas dos casos de uso do sistema proposto.

3.4.1. Fazer Login

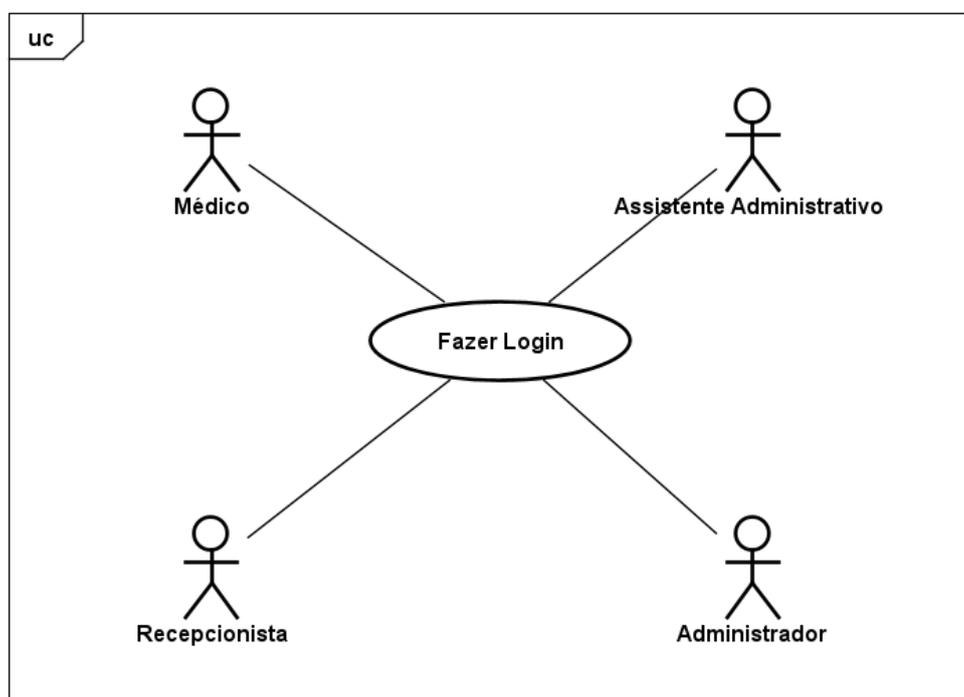


Figura 4: Diagrama UC Fazer Login

Finalidade/Objetivo	Permitir que o colaborador seja autenticado para que possa fazer uso do sistema de acordo com suas permissões.
Atores	Recepcionista, Médico, Assistente Administrativo e Administrador.
Pré-condições	
Evento Inicial	O ator inicia o UC abrindo o sistema.

Fluxo Principal	<p>a) O sistema mostra na tela campos para que o usuário preencha com seu usuário e senha.</p> <p>b) O ator preenche os campos e pressiona o botão “Logar”; [E1] [T1]</p> <p>c) O login é efetuado e o usuário tem acesso as funções correspondentes as suas permissões;</p> <p>d) O usuário clica no ícone de perfil no canto superior direito da tela e expande o <i>dropdown</i>;</p> <p>e) O usuário clica na opção “Sair” e é deslogado do sistema;</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	
Fluxo de Exceção	<p>E1 – Credenciais inválidas</p> <p>a) O sistema verifica que as credenciais informadas estão inválidas;</p> <p>b) O sistema exibe na tela uma mensagem “Usuário ou senha inválido(s)”;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Casos de Teste	<p>T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações</p>

Tabela 1: Narrativa UC Fazer Login

3.4.2. Manter Pacientes

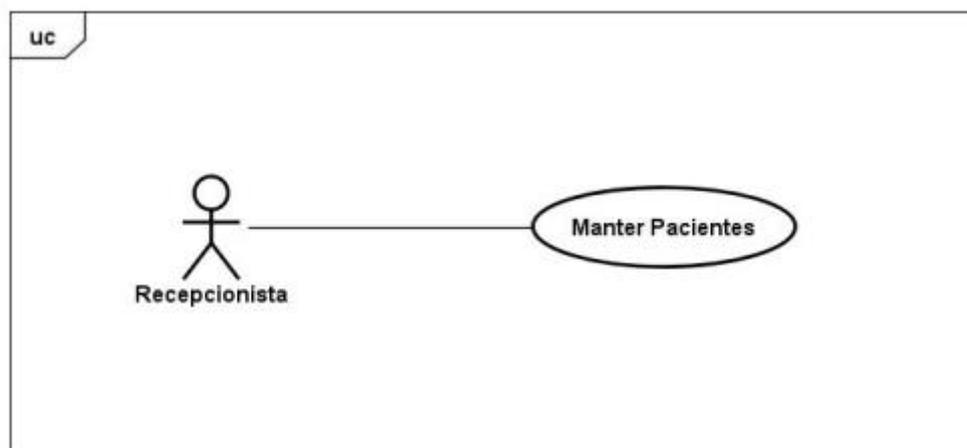


Figura 5: Diagrama UC Manter Pacientes

Finalidade/Objetivo	Permitir que o recepcionista faça a inserção, edição, exclusão e visualização de pacientes da clínica.
Atores	Recepcionista.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC expandindo o <i>dropdown</i> “Cadastros” no menu de navegação e selecionando “Cadastro de Pacientes”.
Fluxo Principal	<p>a) O sistema mostra na tela um campo de texto de pesquisa para filtragem e uma tabela com todos os pacientes já cadastrados;</p> <p>b) O ator clica no botão “Novo Paciente”; [A1] [A2] [A3]</p> <p>c) O sistema mostra uma janela com os campos para serem preenchidos;</p> <p>d) O ator preenche os campos solicitados e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>e) O sistema mostra uma mensagem na tela “Paciente cadastrado com sucesso” e a página é redirecionada;</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar paciente</p> <p>a) O ator seleciona um paciente da lista e pressiona o botão “Editar”;</p> <p>b) O sistema mostra uma janela com os campos para serem atualizados;</p> <p>c) O ator altera os campos que desejar e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>d) O sistema mostra uma mensagem na tela “Paciente alterado com sucesso” e a página é redirecionada;</p> <p>e) Volta ao passo “f)” do fluxo principal;</p> <p>A2 – Excluir paciente</p> <p>a) O ator seleciona um paciente da lista e pressiona o botão “Excluir”;</p> <p>b) O sistema exibe uma janela de confirmação da exclusão;</p> <p>c) O ator confirma pressionando o botão “Sim”; [A4]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A3 – Visualizar paciente</p> <p>a) O ator seleciona um paciente da lista e pressiona o botão “Visualizar”;</p> <p>b) O sistema mostra uma janela com os campos do paciente;</p> <p>c) Após visualizar, o ator clica no botão “Voltar a Lista”; [A1]</p>

	<p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A4 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	<p>E1 – Erro ao realizar operação</p> <p>a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada;</p> <p>b) O sistema exibe uma janela com uma mensagem de erro;</p> <p>c) O ator fecha a mensagem e a página é redirecionada;</p> <p>d) Volta ao passo “a)” do fluxo principal;</p>
Casos de Teste	<p>T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações</p>

Tabela 2: Narrativa UC Manter Pacientes

3.4.3. Manter Médicos

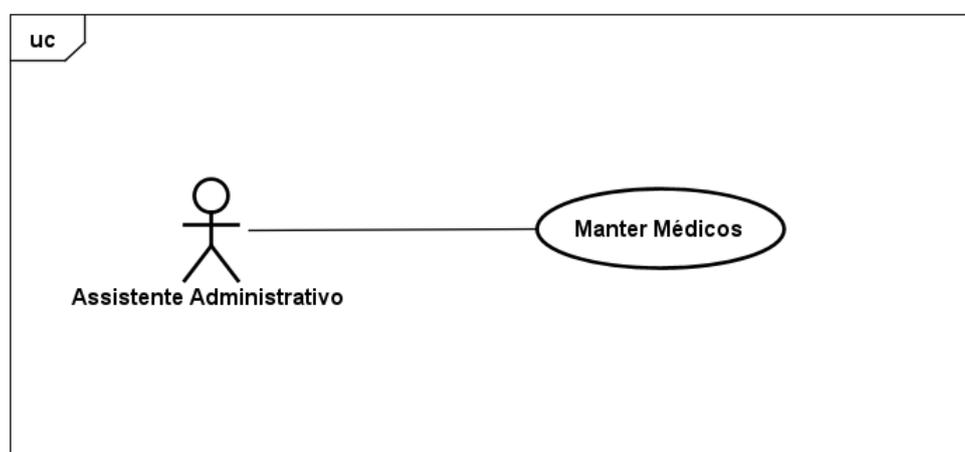


Figura 6: Diagrama UC Manter Médicos

Finalidade/Objetivo	Permitir que o assistente administrativo faça a inserção, edição, exclusão e visualização de médicos da clínica.
Atores	Assistente Administrativo.
Pré-condições	O ator precisa estar autenticado no sistema.

Evento Inicial	O ator inicia o UC expandindo o <i>dropdown</i> “Cadastros” no menu de navegação e selecionando “Cadastro de Médicos”.
Fluxo Principal	<p>a) O sistema mostra na tela um campo de texto de pesquisa para filtragem e uma tabela com todos os médicos já cadastrados;</p> <p>b) O ator clica no botão “Novo Médico”; [A1] [A2] [A3]</p> <p>c) O sistema mostra uma janela com os campos para serem preenchidos;</p> <p>d) O ator preenche os campos solicitados e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>e) O sistema mostra uma mensagem na tela “Médico cadastrado com sucesso” e a página é redirecionada;</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar médico</p> <p>a) O ator seleciona um médico da lista e pressiona o botão “Editar”;</p> <p>b) O sistema mostra uma janela com os campos para serem atualizados;</p> <p>c) O ator altera os campos que desejar e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>d) O sistema mostra uma mensagem na tela “Médico alterado com sucesso” e a página é redirecionada;</p> <p>e) Volta ao passo “f)” do fluxo principal;</p> <p>A2 – Excluir médico</p> <p>a) O ator seleciona um médico da lista e pressiona o botão “Excluir”;</p> <p>b) O sistema exibe uma janela de confirmação da exclusão;</p> <p>c) O ator confirma pressionando o botão “Sim”; [A4]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A3 – Visualizar médico</p> <p>a) O ator seleciona um médico da lista e pressiona o botão “Visualizar”;</p> <p>b) O sistema mostra uma janela com os campos da especialidade;</p> <p>c) Após visualizar, o ator clica no botão “Voltar a Lista”; [A1]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A4 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p>

	b) A operação pendente é cancelada; c) Volta ao passo “a)” do fluxo principal;
Fluxo de Exceção	E1 – Erro ao realizar operação a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada; b) O sistema exibe uma janela com uma mensagem de erro; c) O ator fecha a mensagem e a página é redirecionada; d) Volta ao passo “a)” do fluxo principal;
Casos de Teste	T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações

Tabela 3: Narrativa UC Manter Médicos

3.4.4. Manter Funcionários

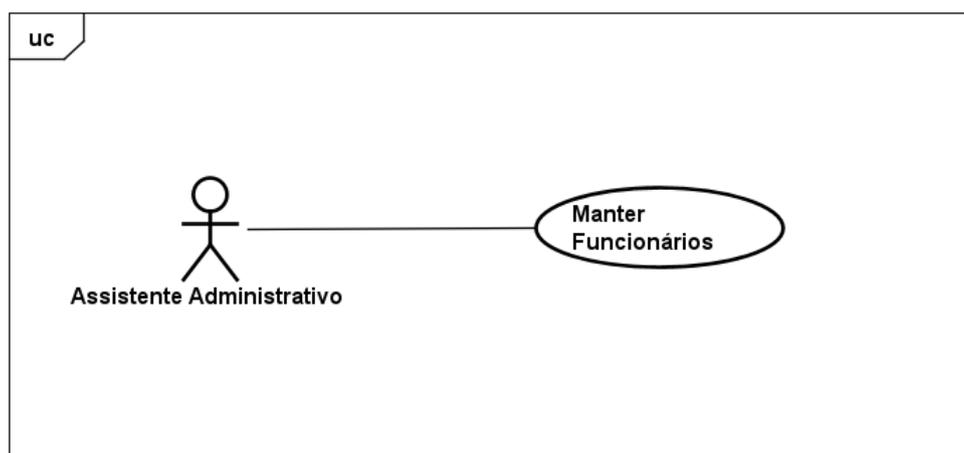


Figura 7: Diagrama UC Manter Funcionários

Finalidade/Objetivo	Permitir que o assistente administrativo faça a inserção, edição, exclusão e visualização de funcionários da clínica.
Atores	Assistente Administrativo.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC expandindo o <i>dropdown</i> “Cadastros” no menu de navegação e selecionando “Cadastro de Funcionários”.
Fluxo Principal	a) O sistema mostra na tela um campo de texto de pesquisa para filtragem e uma tabela com todos os funcionários já cadastrados; b) O ator clica no botão “Novo Funcionário”; [A1] [A2] [A3]

	<p>c) O sistema mostra uma janela com os campos para serem preenchidos;</p> <p>d) O ator preenche os campos solicitados e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>e) O sistema mostra uma mensagem na tela “Funcionário cadastrado com sucesso” e a página é redirecionada;</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar funcionário</p> <p>a) O ator seleciona um funcionário da lista e pressiona o botão “Editar”;</p> <p>b) O sistema mostra uma janela com os campos para serem atualizados;</p> <p>c) O ator altera os campos que desejar e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>d) O sistema mostra uma mensagem na tela “Funcionário alterado com sucesso” e a página é redirecionada;</p> <p>e) Volta ao passo “f)” do fluxo principal;</p> <p>A2 – Excluir funcionário</p> <p>a) O ator seleciona um funcionário da lista e pressiona o botão “Excluir”;</p> <p>b) O sistema exibe uma janela de confirmação da exclusão;</p> <p>c) O ator confirma pressionando o botão “Sim”; [A4]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A3 – Visualizar funcionário</p> <p>a) O ator seleciona um funcionário da lista e pressiona o botão “Visualizar”;</p> <p>b) O sistema mostra uma janela com os campos da especialidade;</p> <p>c) Após visualizar, o ator clica no botão “Voltar a Lista”; [A1]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A4 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	<p>E1 – Erro ao realizar operação</p> <p>a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada;</p>

	b) O sistema exibe uma janela com uma mensagem de erro; c) O ator fecha a mensagem e a página é redirecionada; d) Volta ao passo “a)” do fluxo principal;
Casos de Teste	T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações

Tabela 4: Narrativa UC Manter Funcionários

3.4.5. Manter Cargos

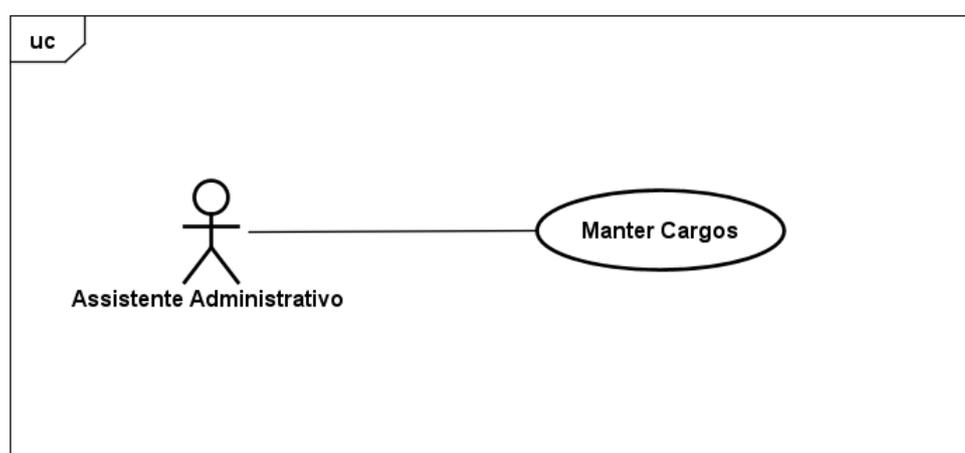


Figura 8: Diagrama UC Manter Cargos

Finalidade/Objetivo	Permitir que o assistente administrativo faça a inserção, edição, exclusão e visualização de cargos dos funcionários.
Atores	Assistente Administrativo.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC expandindo o <i>dropdown</i> “Cadastros” no menu de navegação e selecionando “Cadastro de Cargos”.
Fluxo Principal	a) O sistema mostra na tela um campo de texto de pesquisa para filtragem e uma tabela com todos os cargos já cadastrados; b) O ator clica no botão “Novo Cargo”; [A1] [A2] [A3] c) O sistema mostra uma janela com os campos para serem preenchidos; d) O ator preenche os campos solicitados e pressiona o botão “Salvar”; [A4] [E1] [T1] e) O sistema mostra uma mensagem na tela “Cargo cadastrado com sucesso” e a página é redirecionada;

	f) Caso de uso é encerrado.
Fluxo Alternativo	<p>A1 – Editar cargo</p> <p>a) O ator seleciona um cargo da lista e pressiona o botão “Editar”;</p> <p>b) O sistema mostra uma janela com os campos para serem atualizados;</p> <p>c) O ator altera os campos que desejar e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>d) O sistema mostra uma mensagem na tela “Cargo alterado com sucesso” e a página é redirecionada;</p> <p>e) Volta ao passo “f)” do fluxo principal;</p> <p>A2 – Excluir cargo</p> <p>a) O ator seleciona um cargo da lista e pressiona o botão “Excluir”;</p> <p>b) O sistema exibe uma janela de confirmação da exclusão;</p> <p>c) O ator confirma pressionando o botão “Sim”; [A4]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A3 – Visualizar cargo</p> <p>a) O ator seleciona um cargo da lista e pressiona o botão “Visualizar”;</p> <p>b) O sistema mostra uma janela com os campos da especialidade;</p> <p>c) Após visualizar, o ator clica no botão “Voltar a Lista”; [A1]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A4 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	<p>E1 – Erro ao realizar operação</p> <p>a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada;</p> <p>b) O sistema exibe uma janela com uma mensagem de erro;</p> <p>c) O ator fecha a mensagem e a página é redirecionada;</p> <p>d) Volta ao passo “a)” do fluxo principal;</p>
Casos de Teste	T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações

Tabela 5: Narrativa UC Manter Cargos

3.4.6. Manter Especialidades

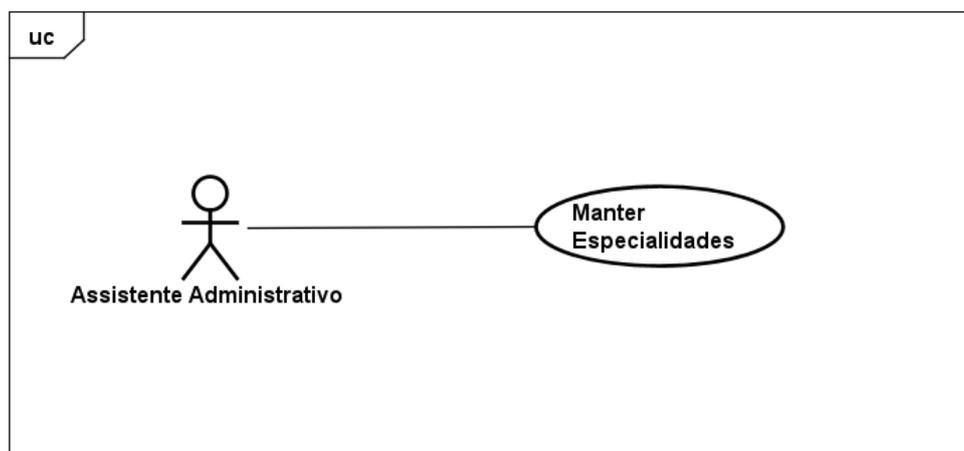


Figura 9: Diagrama UC Manter Especialidades

Finalidade/Objetivo	Permitir que o assistente administrativo faça a inserção, edição, exclusão e visualização de especialidades médicas.
Atores	Assistente Administrativo.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC expandindo o <i>dropdown</i> “Cadastros” no menu de navegação e selecionando “Cadastro de Especialidades”.
Fluxo Principal	<p>a) O sistema mostra na tela um campo de texto de pesquisa para filtragem e uma tabela com todas as especialidades já cadastradas;</p> <p>b) O ator clica no botão “Nova Especialidade”; [A1] [A2] [A3]</p> <p>c) O sistema mostra uma janela com os campos para serem preenchidos;</p> <p>d) O ator preenche os campos solicitados e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>e) O sistema mostra uma mensagem na tela “Especialidade cadastrada com sucesso” e a página é redirecionada;</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar especialidade</p> <p>a) O ator seleciona uma especialidade da lista e pressiona o botão “Editar”;</p> <p>b) O sistema mostra uma janela com os campos para serem atualizados;</p>

	<p>c) O ator altera os campos que desejar e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>d) O sistema mostra uma mensagem na tela “Especialidade alterada com sucesso” e a página é redirecionada;</p> <p>e) Volta ao passo “f)” do fluxo principal;</p> <p>A2 – Excluir especialidade</p> <p>a) O ator seleciona uma especialidade da lista e pressiona o botão “Excluir”;</p> <p>b) O sistema exibe uma janela de confirmação da exclusão;</p> <p>c) O ator confirma pressionando o botão “Sim”; [A4]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A3 – Visualizar especialidade</p> <p>a) O ator seleciona uma especialidade da lista e pressiona o botão “Visualizar”;</p> <p>b) O sistema mostra uma janela com os campos da especialidade;</p> <p>c) Após visualizar, o ator clica no botão “Voltar a Lista”; [A1]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A4 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	<p>E1 – Erro ao realizar operação</p> <p>a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada;</p> <p>b) O sistema exibe uma janela com uma mensagem de erro;</p> <p>c) O ator fecha a mensagem e a página é redirecionada;</p> <p>d) Volta ao passo “a)” do fluxo principal;</p>
Casos de Teste	<p>T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações</p>

Tabela 6: Narrativa UC Manter Especialidades

3.4.7. Manter CIDs

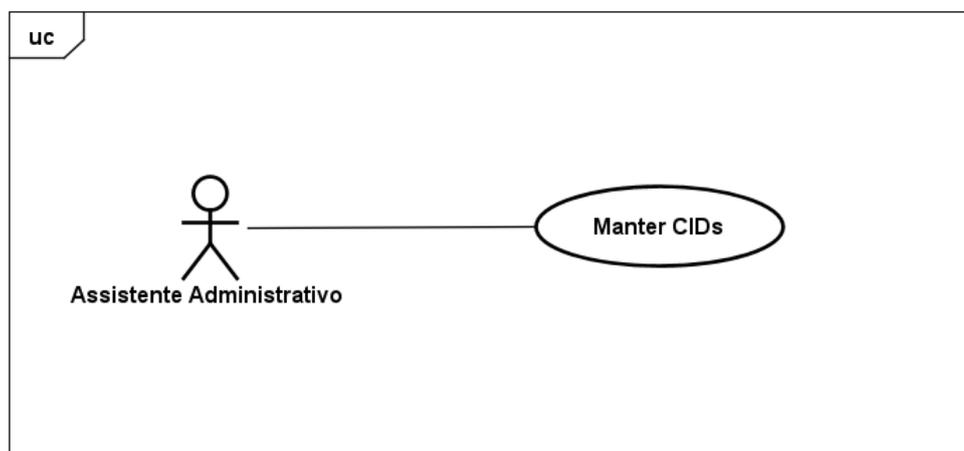


Figura 10: Diagrama UC Manter CIDs

Finalidade/Objetivo	Permitir que o assistente administrativo faça a inserção, edição, exclusão e visualização de CIDs.
Atores	Assistente Administrativo.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC expandindo o <i>dropdown</i> “Cadastros” no menu de navegação e selecionando “Cadastro de Doenças”.
Fluxo Principal	<p>a) O sistema mostra na tela um campo de texto de pesquisa para filtragem e uma tabela com todas as CIDs já cadastradas;</p> <p>b) O ator clica no botão “Nova Doença”; [A1] [A2] [A3]</p> <p>c) O sistema mostra uma janela com os campos para serem preenchidos;</p> <p>d) O ator preenche os campos solicitados e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>e) O sistema mostra uma mensagem na tela “Doença cadastrada com sucesso” e a página é redirecionada;</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar CID</p> <p>a) O ator seleciona uma CID da lista e pressiona o botão “Editar”;</p> <p>b) O sistema mostra uma janela com os campos para serem atualizados;</p> <p>c) O ator altera os campos que desejar e pressiona o botão “Salvar”; [A4] [E1] [T1]</p>

	<p>d) O sistema mostra uma mensagem na tela “Doença alterada com sucesso” e a página é redirecionada;</p> <p>e) Volta ao passo “f)” do fluxo principal;</p> <p>A2 – Excluir CID</p> <p>a) O ator seleciona uma CID da lista e pressiona o botão “Excluir”;</p> <p>b) O sistema exibe uma janela de confirmação da exclusão;</p> <p>c) O ator confirma pressionando o botão “Sim”; [A4]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A3 – Visualizar CID</p> <p>a) O ator seleciona uma CID da lista e pressiona o botão “Visualizar”;</p> <p>b) O sistema mostra uma janela com os campos da especialidade;</p> <p>c) Após visualizar, o ator clica no botão “Voltar a Lista”; [A1]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A4 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	<p>E1 – Erro ao realizar operação</p> <p>a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada;</p> <p>b) O sistema exibe uma janela com uma mensagem de erro;</p> <p>c) O ator fecha a mensagem e a página é redirecionada;</p> <p>d) Volta ao passo “a)” do fluxo principal;</p>
Casos de Teste	<p>T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações</p>

Tabela 7: Narrativa UC Manter CIDs

3.4.8. Manter Usuários

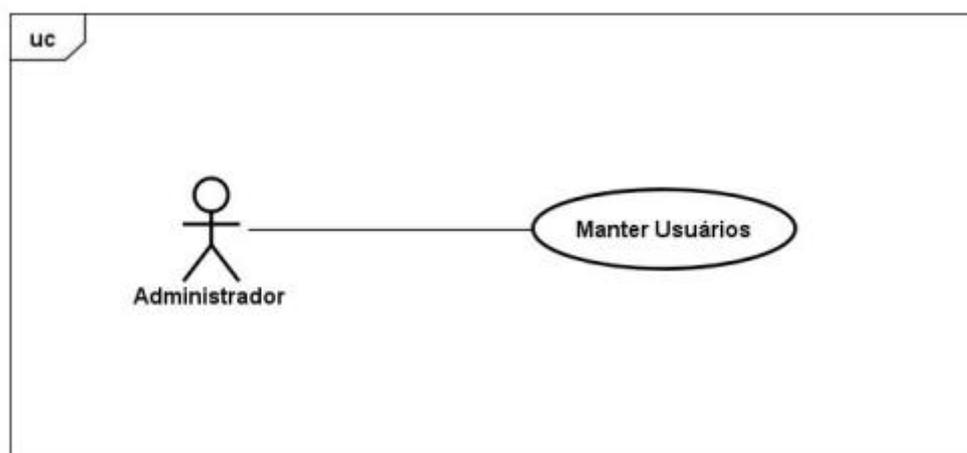


Figura 11: Diagrama UC Manter Usuários

Finalidade/Objetivo	Permitir que o administrador faça a inserção, edição, exclusão e visualização de usuários de acesso.
Atores	Administrador.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC selecionando o item “Admin” no menu de navegação.
Fluxo Principal	<p>a) O sistema mostra na tela um campo de texto de pesquisa para filtragem e uma tabela com todos os usuários já cadastrados;</p> <p>b) O ator clica no item de navegação na esquerda “Cadastrar Usuário”; [A1] [A2] [A3]</p> <p>c) O sistema mostra uma janela com os campos para serem preenchidos;</p> <p>d) O ator preenche os campos solicitados e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>e) O sistema mostra uma mensagem na tela “Usuário cadastrado com sucesso” e a página é redirecionada;</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar usuário</p> <p>a) O ator seleciona um usuário da lista e pressiona o botão “Editar”;</p> <p>b) O sistema mostra uma janela com os campos para serem atualizados;</p>

	<p>c) O ator altera os campos que desejar e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>d) O sistema mostra uma mensagem na tela “Usuário alterado com sucesso” e a página é redirecionada;</p> <p>e) Volta ao passo “f)” do fluxo principal;</p> <p>A2 – Excluir usuário</p> <p>a) O ator seleciona um usuário da lista e pressiona o botão “Excluir”;</p> <p>b) O sistema exibe uma janela de confirmação da exclusão;</p> <p>c) O ator confirma pressionando o botão “Sim”; [A4]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A3 – Visualizar usuário</p> <p>a) O ator seleciona um usuário da lista e pressiona o botão “Visualizar”;</p> <p>b) O sistema mostra uma janela com os campos da especialidade;</p> <p>c) Após visualizar, o ator clica no botão “Voltar a Lista”; [A1]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A4 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	<p>E1 – Erro ao realizar operação</p> <p>a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada;</p> <p>b) O sistema exibe uma janela com uma mensagem de erro;</p> <p>c) O ator fecha a mensagem e a página é redirecionada;</p> <p>d) Volta ao passo “a)” do fluxo principal;</p>
Casos de Teste	<p>T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações</p>

Tabela 8: Narrativa UC Manter Usuários

3.4.9. Manter Agendamentos

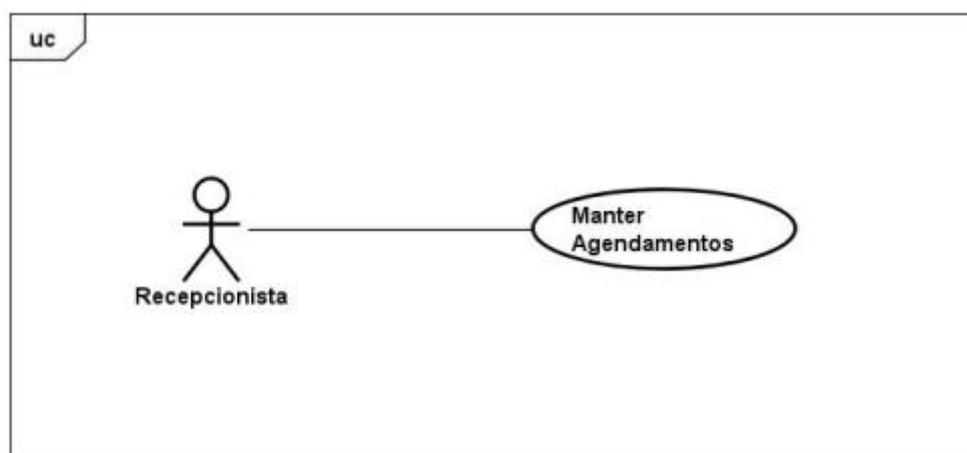


Figura 12: Diagrama UC Manter Agendamentos

Finalidade/Objetivo	Permitir que o recepcionista faça a inserção, edição, exclusão e visualização de agendamentos da clínica.
Atores	Recepcionista.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC expandindo o <i>dropdown</i> "Consultas" no menu de navegação e selecionando "Realização de Agendamentos".
Fluxo Principal	<p>a) O sistema mostra na tela um calendário da semana atual com os agendamentos realizados, dispostos em seus respectivos dias;</p> <p>b) O ator clica na fatia de horário do dia desejado; [A1] [A2] [A3]</p> <p>c) O ator preenche os campos solicitados e pressiona o botão "Salvar"; [A4] [E1] [T1]</p> <p>d) O sistema mostra uma mensagem na tela "Agendamento cadastrado com sucesso" e a página é redirecionada;</p> <p>e) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar agendamento</p> <p>a) O ator clica em cima de um agendamento disposto no calendário;</p> <p>b) O sistema mostra uma janela com os campos para serem atualizados;</p> <p>c) O ator altera os campos que desejar e pressiona o botão "Salvar"; [A4] [E1] [T1]</p>

	<p>d) O sistema mostra uma mensagem na tela “Agendamento alterado com sucesso” e a página é redirecionada;</p> <p>e) Volta ao passo “g)” do fluxo principal;</p> <p>A2 – Cancelar agendamento</p> <p>a) O ator clica em cima de um agendamento disposto no calendário;</p> <p>b) O sistema exibe um modal com as informações do agendamento;</p> <p>c) O ator clica no botão “Cancelar”;</p> <p>d) O sistema exibe uma janela de confirmação da exclusão;</p> <p>e) O ator confirma pressionando o botão “Sim”; [A4]</p> <p>f) Volta ao passo “e)” do fluxo principal;</p> <p>A3 – Visualizar agendamento</p> <p>a) O ator clica em cima de um agendamento disposto no calendário;</p> <p>b) O sistema exibe um modal com as informações do agendamento;</p> <p>c) Após visualizar, o ator clica no botão de fechar o modal;</p> <p>d) Volta ao passo “e)” do fluxo principal;</p> <p>A4 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	<p>E1 – Erro ao realizar operação</p> <p>a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada;</p> <p>b) O sistema exibe uma janela com uma mensagem de erro;</p> <p>c) O ator fecha a mensagem e a página é redirecionada;</p> <p>d) Volta ao passo “a)” do fluxo principal;</p>
Casos de Teste	<p>T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações</p>

Tabela 9: Narrativa UC Manter Agendamentos

3.4.10. Manter Atendimentos

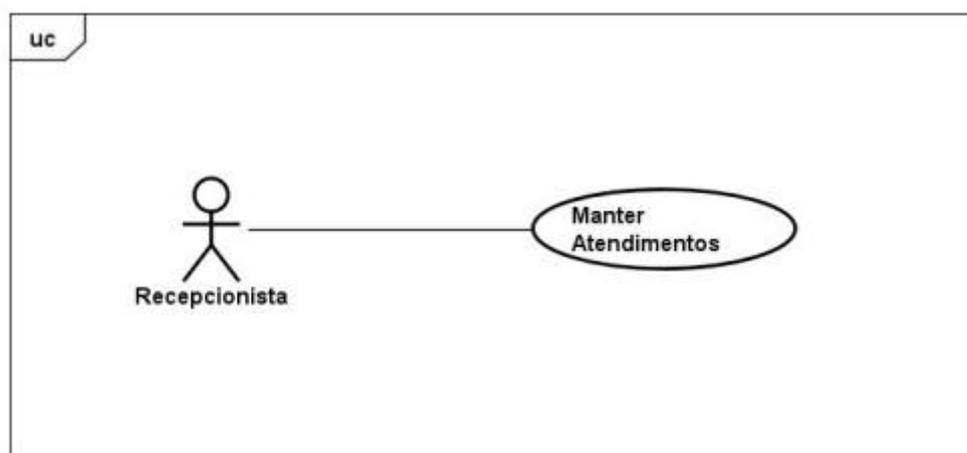


Figura 13: Diagrama UC Manter Atendimentos

Finalidade/Objetivo	Permitir que o recepcionista faça a inserção, edição, exclusão e visualização de atendimentos.
Atores	Recepcionista.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC expandindo o <i>dropdown</i> "Consultas" no menu de navegação e selecionando "Realização de Atendimentos".
Fluxo Principal	<p>a) O sistema mostra na tela uma tabela com todos os cabeçalhos de atendimentos presentes no sistema;</p> <p>b) O ator clica no botão "Iniciar Atendimento"; [A1] [A2] [A3]</p> <p>c) O sistema mostra uma janela com os campos para serem preenchidos;</p> <p>d) O ator preenche os campos solicitados e pressiona o botão "Salvar"; [A4] [E1] [T1]</p> <p>e) O sistema mostra uma mensagem na tela "Cabeçalho de atendimento cadastrado com sucesso" e a página é redirecionada;</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Editar cabeçalho de atendimento</p> <p>a) O ator seleciona um cabeçalho de atendimento da lista e pressiona o botão "Editar";</p> <p>b) O sistema mostra uma janela com os campos para serem atualizados;</p>

	<p>c) O ator altera os campos que desejar e pressiona o botão “Salvar”; [A4] [E1] [T1]</p> <p>d) O sistema mostra uma mensagem na tela “Cadastro alterado com sucesso” e a página é redirecionada;</p> <p>e) Volta ao passo “f)” do fluxo principal;</p> <p>A2 – Cancelar cabeçalho de atendimento</p> <p>a) O ator seleciona um cabeçalho de atendimento da lista e pressiona o botão “Cancelar”;</p> <p>b) O sistema exibe uma janela de confirmação da exclusão;</p> <p>c) O ator confirma pressionando o botão “Sim”; [A4]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A3 – Visualizar cabeçalho de atendimento</p> <p>a) O ator seleciona um cabeçalho de atendimento da lista e pressiona o botão “Visualizar”;</p> <p>b) O sistema mostra uma janela com os campos da especialidade;</p> <p>c) Após visualizar, o ator clica no botão “Voltar a Lista”; [A1]</p> <p>d) Volta ao passo “f)” do fluxo principal;</p> <p>A4 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	<p>E1 – Erro ao realizar operação</p> <p>a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada;</p> <p>b) O sistema exibe uma janela com uma mensagem de erro;</p> <p>c) O ator fecha a mensagem e a página é redirecionada;</p> <p>d) Volta ao passo “a)” do fluxo principal;</p>
Casos de Teste	<p>T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações</p>

Tabela 10: Narrativa UC Manter Atendimentos

3.4.11. Realizar Atendimento

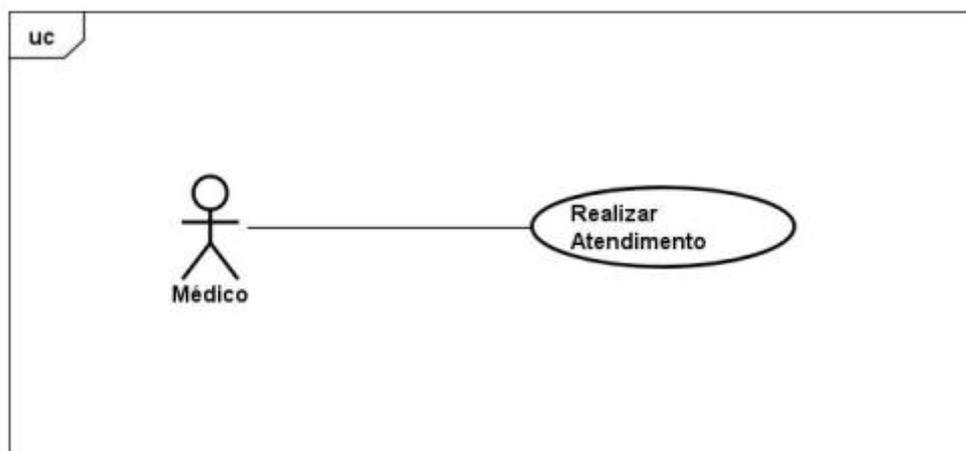


Figura 14: Diagrama UC Realizar Atendimento

Finalidade/Objetivo	Permitir que o médico realize o atendimento dos pacientes.
Atores	Médico.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC expandindo o <i>dropdown</i> "Consultas" no menu de navegação e selecionando "Realização de Atendimentos".
Fluxo Principal	<p>a) O sistema mostra na tela uma tabela com todos os cabeçalhos de atendimentos presentes no sistema;</p> <p>b) O ator seleciona um cabeçalho de atendimento da lista clica no botão "Atender"; [A1]</p> <p>c) O sistema mostra uma janela com os campos a serem preenchidos do atendimento;</p> <p>d) O ator preenche os campos solicitados e pressiona o botão "Salvar"; [A2] [E1] [T1]</p> <p>e) O sistema mostra uma mensagem na tela "Atendimento realizado com sucesso" e a página é redirecionada;</p> <p>f) O sistema salva as informações do atendimento realizado e atualiza o status do atendimento para "FINALIZADO";</p> <p>g) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Cancelar atendimento</p> <p>a) O ator seleciona um paciente na tela "Prontuário Eletrônico do Paciente";</p> <p>b) O ator seleciona um atendimento da lista e pressiona o botão "Cancelar";</p>

	<p>c) O sistema salva as informações do atendimento cancelado e atualiza o status do atendimento selecionado para “CANCELADO”;</p> <p>d) Volta ao passo “g)” do fluxo principal;</p> <p>A2 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	<p>E1 – Erro ao realizar operação</p> <p>a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada;</p> <p>b) O sistema exibe uma janela com uma mensagem de erro;</p> <p>c) O ator fecha a mensagem e a página é redirecionada;</p> <p>d) Volta ao passo “a)” do fluxo principal;</p>
Casos de Teste	<p>T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações</p>

Tabela 11: Narrativa UC Realizar Atendimento

3.4.12. Visualizar Prontuário Eletrônico do Paciente

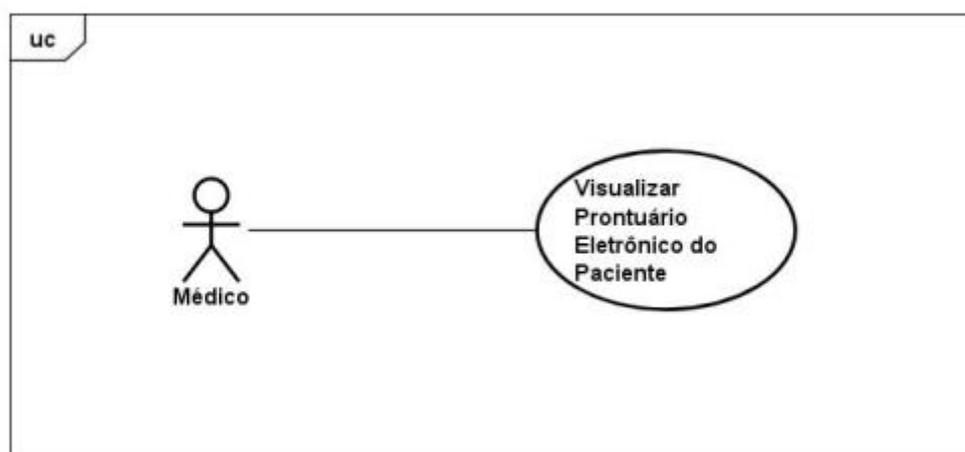


Figura 15: Diagrama UC Visualizar Prontuário Eletrônico do Paciente

Finalidade/Objetivo	Permitir que o médico realize a visualização do prontuário eletrônico do paciente.
Atores	Médico.

Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC expandindo o <i>dropdown</i> “Consultas” no menu de navegação e selecionando “Prontuário Eletrônico do Paciente”.
Fluxo Principal	<p>a) O sistema mostra na tela um campo onde o ator deve pesquisar pacientes por nome ou CPF;</p> <p>b) O ator preenche o campo; [A1] [T1]</p> <p>c) O sistema busca os pacientes e devolve uma lista na tela com os resultados;</p> <p>d) O ator seleciona um paciente da lista e clica em cima do item com os dados dele;</p> <p>e) O sistema abre a página daquele paciente, onde conterà os dados de cadastro do mesmo e uma lista com todos os atendimentos realizados até o momento na clínica.</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Pesquisar todos</p> <p>a) O ator pressiona o botão “Pesquisar Todos”;</p> <p>b) O sistema busca todos os pacientes cadastrados no banco e devolve uma lista na tela com os resultados;</p> <p>c) Volta ao passo “d)” do fluxo principal.</p>
Fluxo de Exceção	
Casos de Teste	T1 – Verificar se o campo foi preenchido corretamente, respeitando o tipo e validações

Tabela 12: Narrativa UC Visualizar Prontuário Eletrônico do Paciente

3.4.13. Emitir Relatório de Atendimentos

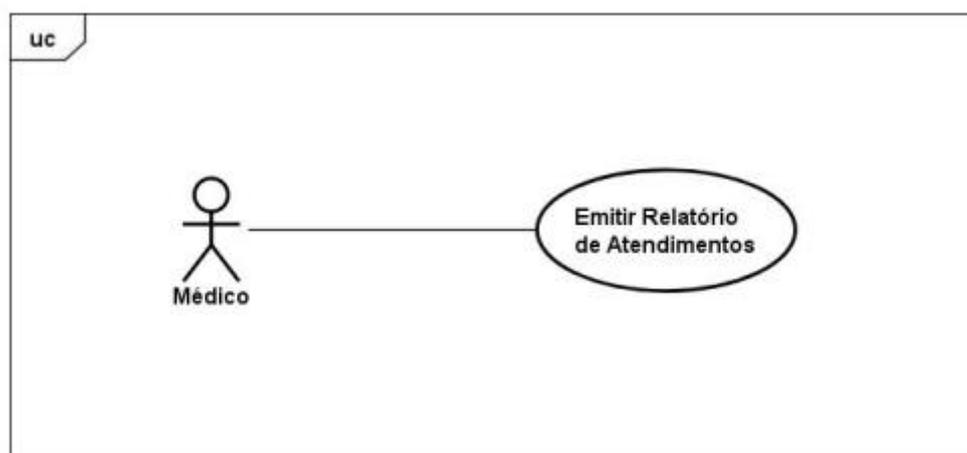


Figura 16: Diagrama UC Emitir Relatório de Atendimentos

Finalidade/Objetivo	Permitir que o médico realize a emissão de relatórios do total de atendimentos em um determinado período.
Atores	Médico.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC selecionando o item “Relatórios” no menu de navegação.
Fluxo Principal	<p>a) O sistema mostra na tela informações gerais sobre a clínica e um campo onde o ator deve selecionar um médico;</p> <p>b) O ator preenche o campo; [E1] [T1]</p> <p>c) O sistema buscará e agrupará todos os atendimentos em geral da clínica e referentes ao médico, mensalmente;</p> <p>d) O sistema irá exibir na tela um gráfico com os dados obtidos;</p> <p>e) Caso de uso é encerrado.</p>
Fluxo Alternativo	
Fluxo de Exceção	<p>E1 – Erro ao realizar operação</p> <p>a) O sistema identifica algum erro que impossibilita que a operação pendente seja finalizada;</p> <p>b) O sistema exibe uma janela com uma mensagem de erro;</p> <p>c) O ator fecha a mensagem e a página é redirecionada;</p> <p>d) Volta ao passo “a)” do fluxo principal;</p>
Casos de Teste	T1 – Verificar se os campos foram preenchidos corretamente, respeitando os tipos e as validações

Tabela 13: Narrativa UC Emitir Relatório de Atendimentos

3.4.14. Emitir Atestado Médico

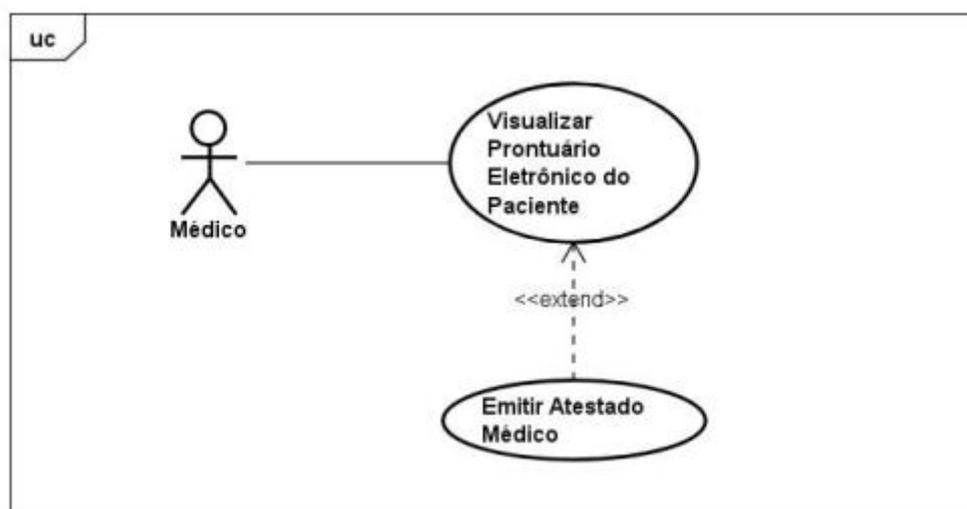


Figura 17: Diagrama UC Emitir Atestado Médico

Finalidade/Objetivo	Permitir que o médico realize a emissão de atestados médicos.
Atores	Médico.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC “Visualizar Prontuário Eletrônico do Paciente”, fazendo com que o histórico de um paciente seja exibido.
Fluxo Principal	<p>a) O sistema mostra na tela uma lista, em formato de linha do tempo, com todos os atendimentos do paciente selecionado;</p> <p>b) O ator seleciona atendimento e pressiona o botão “Emitir Atestado”;</p> <p>c) O sistema gera um modelo de atestado com os dados referentes ao paciente e ao atendimento;</p> <p>d) Uma mensagem de confirmação perguntando se o ator deseja imprimir o atestado é exibida e o ator confirma pressionando o botão “Sim”; [A1]</p> <p>e) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	
Casos de Teste	

Tabela 14: Narrativa UC Emitir Atestado Médico

3.4.15. Emitir Receita Médica

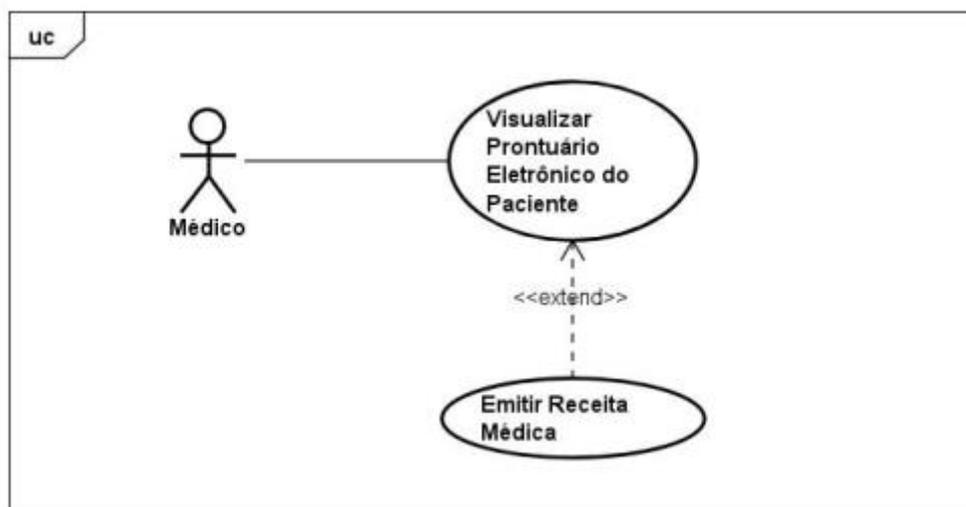


Figura 18: Diagrama UC Emitir Receita Médica

Finalidade/Objetivo	Permitir que o médico realize a emissão de receitas médicas.
Atores	Médico.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC “Visualizar Prontuário Eletrônico do Paciente”, fazendo com que o histórico de um paciente seja exibido.
Fluxo Principal	<p>a) O sistema mostra na tela uma lista, em formato de linha do tempo, com todos os atendimentos do paciente selecionado;</p> <p>b) O ator seleciona um atendimento e expande o item “Receituário”</p> <p>c) O ator pressiona o botão “Emitir Receita” dentro do receituário expandido;</p> <p>d) O sistema gera um modelo de receita com os dados referentes ao paciente e ao atendimento;</p> <p>e) Uma mensagem de confirmação perguntando se o ator deseja confirmar e imprimir a receita é exibida e o ator confirma pressionando o botão “Sim”; [A1]</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	<p>A1 – Cancelar operação</p> <p>a) O ator pressiona o botão “Cancelar”;</p> <p>b) A operação pendente é cancelada;</p> <p>c) Volta ao passo “a)” do fluxo principal;</p>
Fluxo de Exceção	

Casos de Teste

Tabela 15: Narrativa UC Emitir Receita Médica

3.4.16. Emitir Exames Solicitados

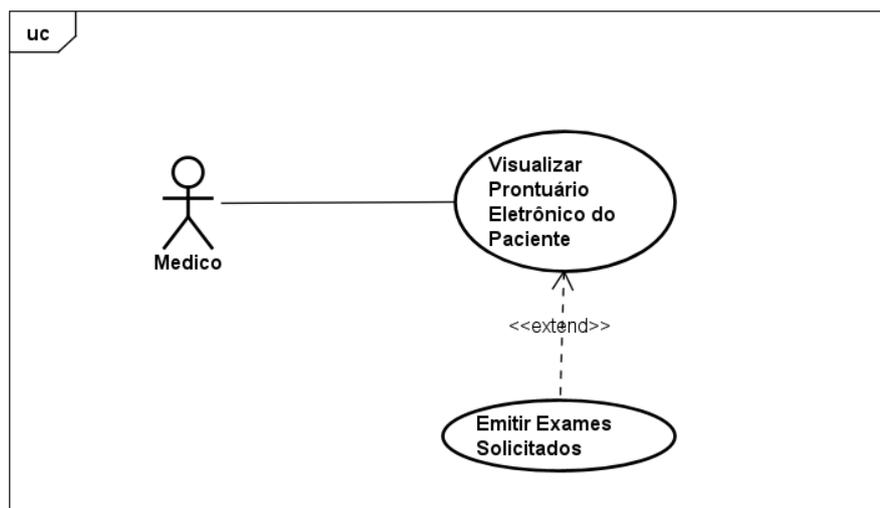


Figura 19: Diagrama UC Emitir Receita Médica

Finalidade/Objetivo	Permitir que o médico realize a emissão de receitas médicas.
Atores	Médico.
Pré-condições	O ator precisa estar autenticado no sistema.
Evento Inicial	O ator inicia o UC “Visualizar Prontuário Eletrônico do Paciente”, fazendo com que o histórico de um paciente seja exibido.
Fluxo Principal	<p>a) O sistema mostra na tela uma lista, em formato de linha do tempo, com todos os atendimentos do paciente selecionado;</p> <p>b) O ator seleciona um atendimento e expande o item “Exames”</p> <p>c) O ator pressiona o botão “Emitir Exames” dentro da seção de exames expandida;</p> <p>d) O sistema gera um modelo de exames solicitados com os dados referentes ao paciente e ao atendimento;</p> <p>e) Uma mensagem de confirmação perguntando se o ator deseja confirmar e imprimir a solicitação de exames é exibida e o ator confirma pressionando o botão “Sim”; [A1]</p> <p>f) Caso de uso é encerrado.</p>
Fluxo Alternativo	A1 – Cancelar operação

	a) O ator pressiona o botão “Cancelar”; b) A operação pendente é cancelada; c) Volta ao passo “a)” do fluxo principal;
Fluxo de Exceção	
Casos de Teste	

Tabela 16: Narrativa UC Emitir Exames Solicitados

3.5. DIAGRAMA DE CLASSES

O diagrama de classe representa uma maneira gráfica de ilustrar a estrutura de um sistema e as conexões existentes entre seus diferentes componentes, tratando-se da base de projetos de *software* que fazem uso do padrão de desenvolvimento orientado a objetos. É de responsabilidade desses diagramas descrever os objetos presentes no sistema, representados pelas classes, as propriedades e operações de cada objeto e como se darão as ligações entre cada um deles (FOWLER, 2005).

Devido a seu alto valor dentro da engenharia de *software*, esse tipo de diagrama é amplamente utilizado, sendo uma das representações UML mais comuns de ser encontrada. A seguir, a Figura 20 ilustra o diagrama de classes do sistema.

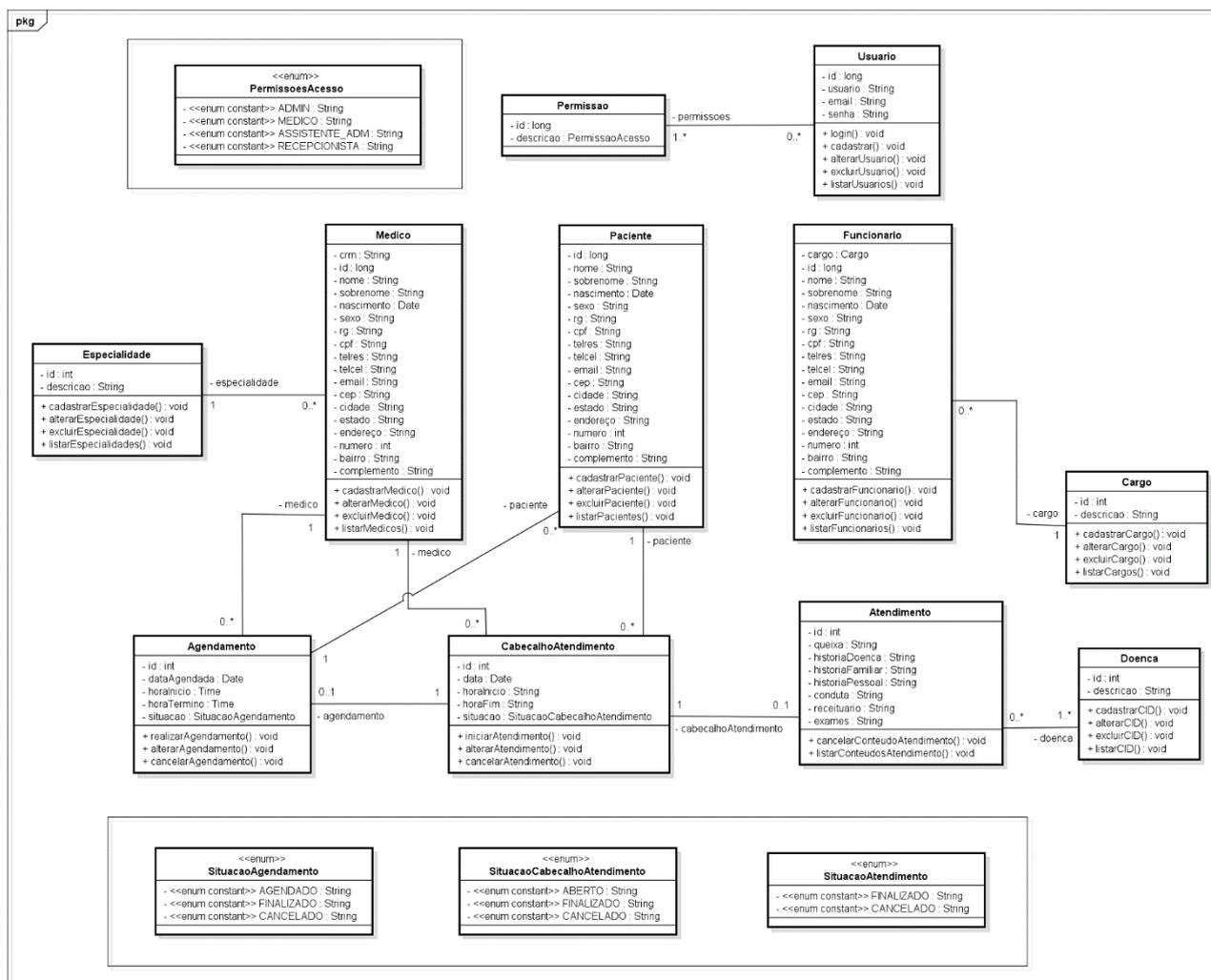


Figura 20: Diagrama de Classes

3.6. DIAGRAMA ENTIDADE-RELACIONAMENTO

O diagrama entidade-relacionamento (DER) é uma técnica aplicada no *design* de bancos de dados relacionais. Através dele, é possível definir como será a estrutura de um banco de dados, definindo as entidades ou tabelas que deverão existir, os atributos ou campos que as compõe e os relacionamentos existentes entre diferentes tabelas do banco. O DER pode ser utilizado como suporte para o planejamento e compreensão dos dados necessários em uma empresa ou outro ambiente, na construção do banco de dados físico e na integração com bancos de dados já existentes (LI; CHEN, 2009).

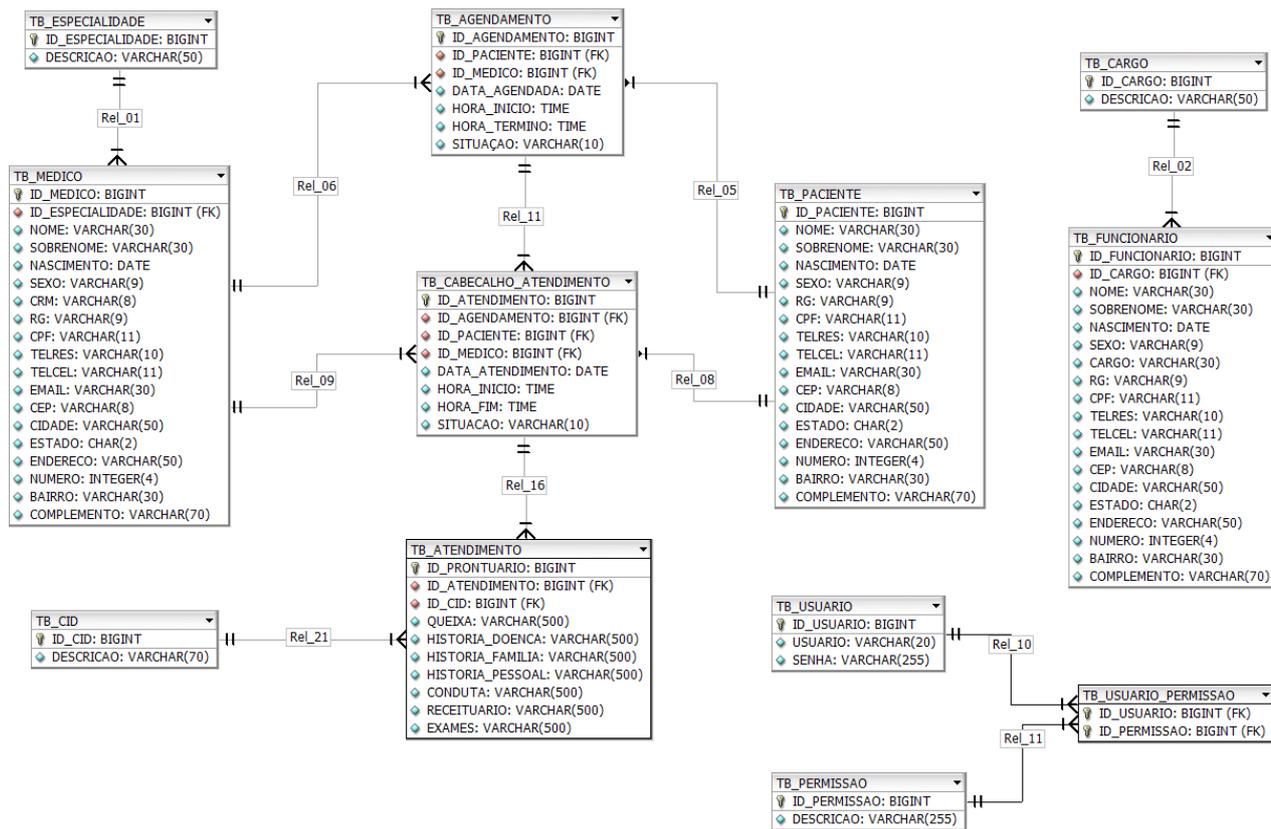


Figura 21: Diagrama Entidade-Relacionamento

4. INTERFACES DO SISTEMA

O presente capítulo tem como propósito apresentar as interfaces gráficas e as diferentes funcionalidades do sistema desenvolvido. Cada interface será abordada em uma seção, que além de expor a representação visual da interface, conterá um breve texto descrevendo a funcionalidade do módulo associado.

4.1. PÁGINA DE LOGIN

A página de login é a primeira interface exibida quando o endereço do sistema é acessado e está ilustrada na Figura 22. Nessa tela, o usuário deverá informar seus dados de autenticação previamente cadastrados no banco de dados pelo Administrador. Caso as credenciais informadas estejam corretas, a autenticação é efetuada e o usuário é redirecionado ao sistema. Caso contrário, ocorrerá um erro e o acesso ao sistema não será permitido.

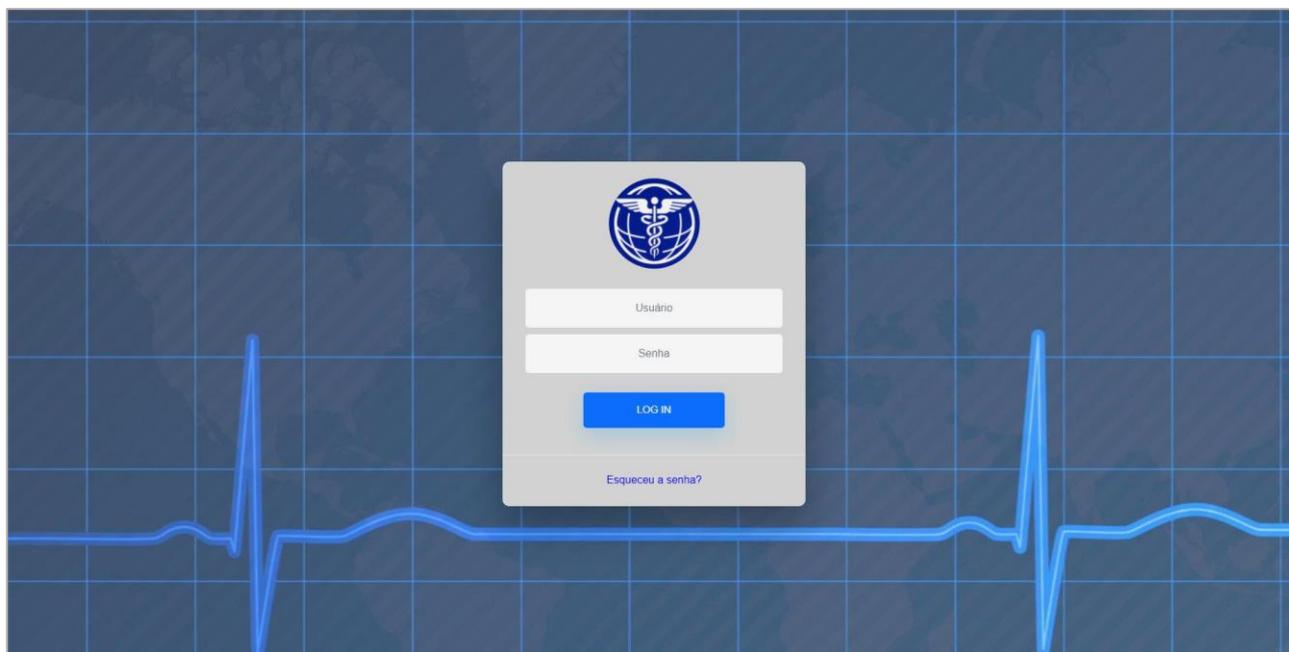


Figura 22: Página de Login

4.2. PÁGINA INICIAL

A página inicial serve como base para toda a extensão do sistema, sendo a primeira página exibida após o usuário efetuar a autenticação. Através dela torna-se possível acessar todos os módulos da aplicação, por meio do menu de navegação superior ou dos ícones dispostos na parte central, conforme ilustrado na Figura 23.



Figura 23: Página Inicial

4.3. PÁGINAS DE CADASTRO

As páginas de cadastro são responsáveis pela realização do controle dos cadastros das diferentes entidades presentes no sistema. Todas as páginas seguem um padrão, salvo as interfaces de agendamento e atendimento, que possuem uma estilização diferente e serão decorridas mais adiante.

Cada página de cadastro irá exibir uma lista, em forma de tabela, com todos os registros existentes de uma determinada, além de botões para a realização de ações. A Figura 24, a seguir, ilustra um exemplo de uma tela de cadastro.



Figura 24: Página de Cadastro de Especialidades

4.4. PÁGINAS DE FORMULÁRIOS DE CADASTRO

Os formulários de cadastro permitem ao usuário realizar a inserção de novos registros no banco de dados ou atualização de registros existentes. Todos os formulários seguem o mesmo padrão, se diferenciando apenas pelos campos a serem preenchidos.

Dentro da página de formulário, o usuário deve preencher os campos, respeitando as validações necessárias, para assim poder salvar a inserção ou edição. A Figura 25 abaixo exemplifica um formulário de inserção.

The screenshot shows a web interface for adding a new specialty. At the top, there is a blue navigation bar with the text 'TECHMEDICINE' and menu items 'Home', 'Especialidades', and 'Consultas'. A user profile icon is visible in the top right corner. The main content area is titled 'Nova Especialidade' and contains a form with the label 'Descrição da Especialidade' and a text input field. Below the input field are two buttons: 'Salvar' (blue) and 'Cancelar' (red). The footer of the page is blue and contains the copyright notice '© 2021 TechMedicine, Inc.', a heart icon, and social media links for 'Twitter', 'Instagram', and 'Facebook'.

Figura 25: Página de Inserção de Especialidades

4.5. PÁGINAS DE VISUALIZAÇÃO DE DADOS

As interfaces de visualização de dados apresentam as informações de um determinado cadastro do sistema de forma organizada. Tais telas tem o objetivo de mostrar todos os dados de um registro, visto que as tabelas das páginas de cadastro mostram apenas alguns dados específicos, conforme ilustrado na Figura 26.



Figura 26: Página de Visualização de Especialidades

4.6. PÁGINAS DE EXCLUSÃO DE DADOS

As páginas de exclusão de registros não são interfaces novas, assumindo a forma de janelas que são exibidas nas telas de cadastros, sempre no centro superior da página. Essas janelas são denominadas de modal e se comportam de maneira que não interferem no conteúdo da interface onde o modal é disposto.

O modal de exclusão apresenta um texto e botões de confirmação, onde o usuário pode cancelar ou confirmar a ação, nesse último caso fazendo com que o registro seja deletado do banco de dados. A Figura 27 ilustra um exemplo de um modal de exclusão.

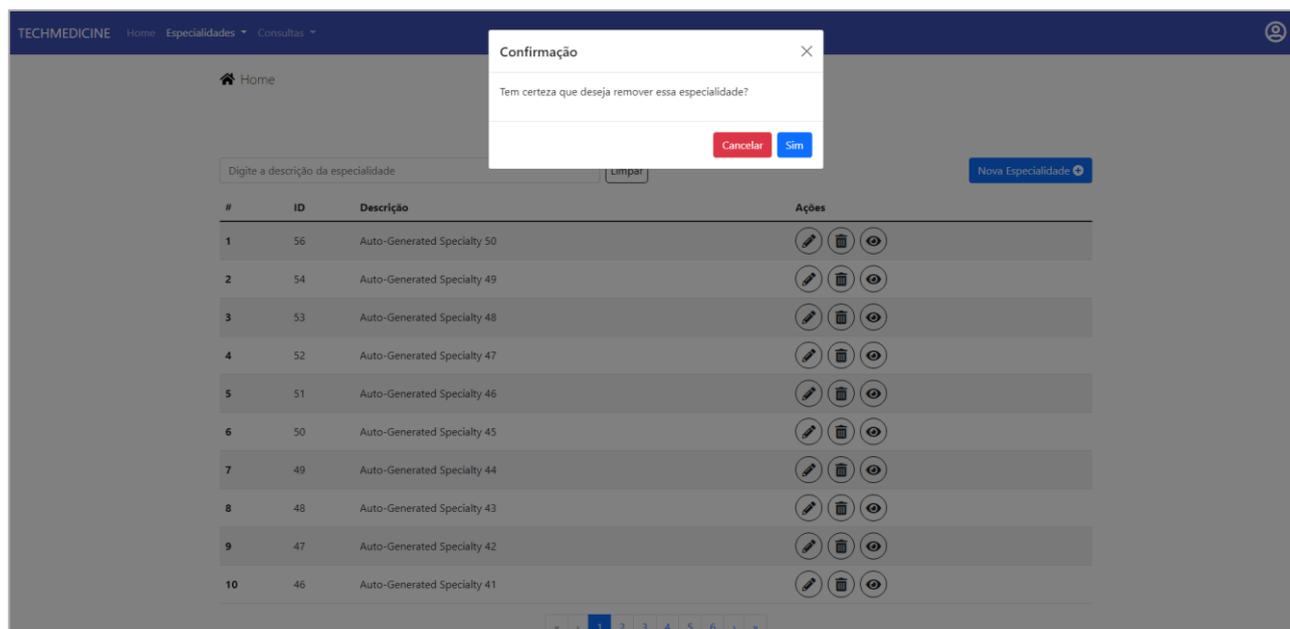


Figura 27: Página de Exclusão de Especialidades

4.7. PÁGINAS DE REALIZAÇÃO DE AGENDAMENTOS

A página de realização de agendamentos é composta por duas janelas diferentes, uma que contém uma visualização em formato de calendário e outra que segue o padrão da tabela de listagem exibida anteriormente,

A interface de calendário é composta por um calendário disposto na tela que permite a visualização semanal dos dias. As linhas do calendário abrangem os horários disponíveis para atendimento, onde cada linha representa uma fatia de tempo. Os agendamentos são dispostos em formato de pequenos cartões, posicionados na linha e coluna que representam o horário e o dia, respectivamente.

Em ambas as janelas, há a presença de opções de filtros que permitem um controle sobre os dados que serão exibidos. Essas opções permitem que os dados sejam filtrados de acordo com a situação do atendimento e o médico associado. A seguir, a Figura 28 ilustra o calendário de agendamentos.

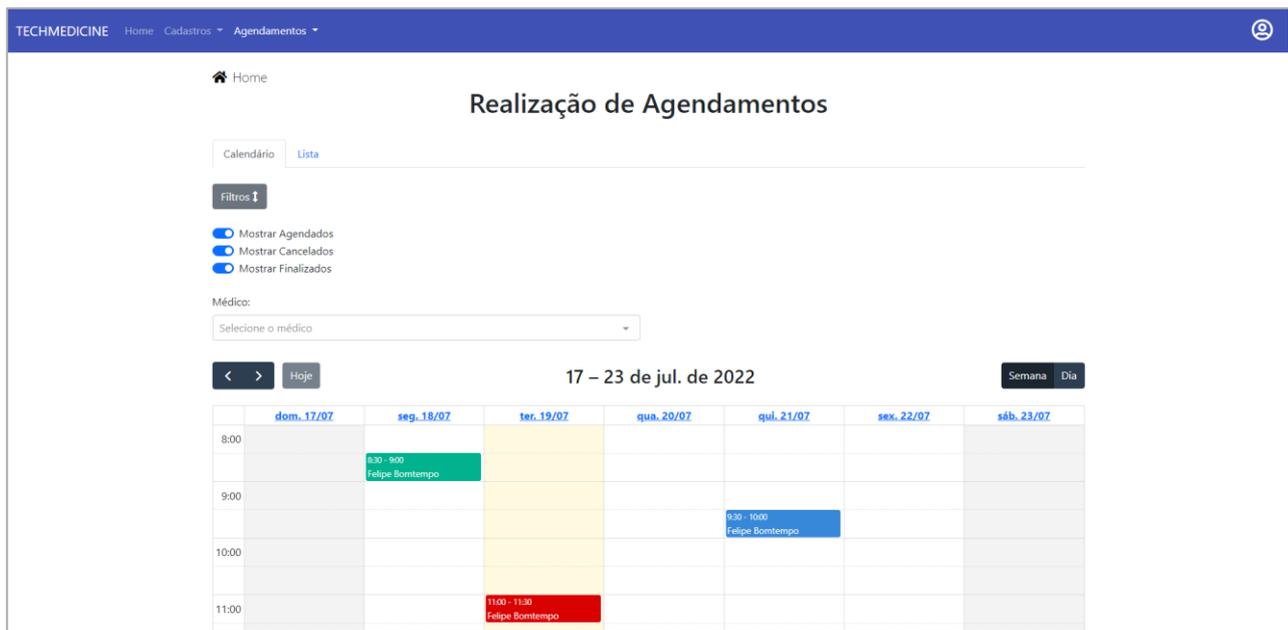


Figura 28: Calendário de Realização de Agendamentos

4.8. PÁGINA DE PRONTUÁRIO DOS PACIENTES

Para a visualização do histórico e outras informações referentes aos atendimentos de um paciente, o sistema conta com um prontuário eletrônico que permite visualizar as informações citadas anteriormente. Quando acessada a página de prontuários, o usuário se depara com um campo de texto onde poderá informar o nome do paciente desejado. A cada tecla pressionada, o sistema irá realizar a filtragem e buscar os resultados que contenham o texto informado no campo de pesquisa. Ao encontrar o paciente desejado, o usuário pode clicar no cartão para acessar a página do paciente. A Figura 29 abaixo ilustra essa interface com um paciente já filtrado.

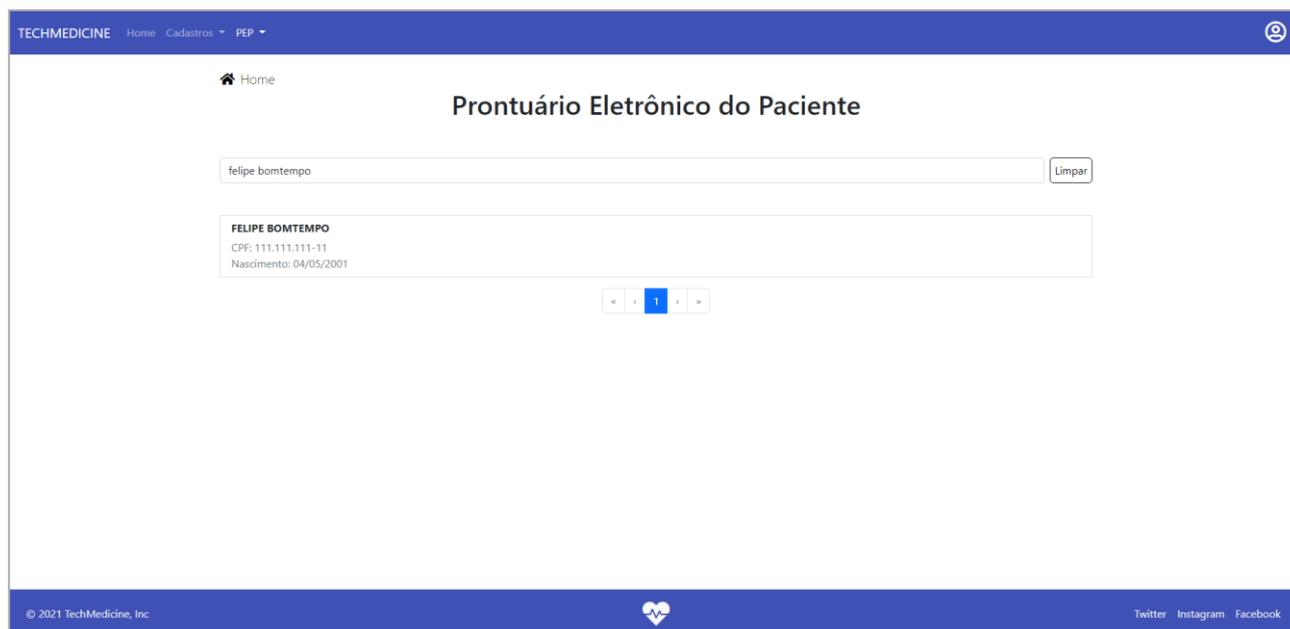


Figura 29: Página de Prontuário Eletrônico do Paciente

4.9. PÁGINA DO PACIENTE

A página do paciente é a interface que conterà todo o histórico e dados do paciente na clínica. Ela é composta por um cabeçalho que contém as informações do cadastro do paciente e por uma linha do tempo, ordenada por data de maneira decrescente, que exhibe todos os atendimentos associados ao paciente. Ainda, caso seja necessário alterar ou cancelar um atendimento, isso se mostra possível devido aos botões posicionados embaixo das informações do atendimento. A Figura 30 subsequente exemplifica essa tela.



Figura 30: Página do Paciente

4.10. PÁGINA DE RELATÓRIOS

A página de relatórios permite ao usuário ter uma breve visão geral dos números da clínica. Entre os dados gerados, estão o total de pacientes cadastrados, agendamentos realizados e atendimentos bem-sucedidos. Mais abaixo, há um campo *dropdown* que permite ao usuário escolher um médico. Ao selecionar, o sistema irá gerar um gráfico de linhas, dividido por mês, exibindo todos os atendimentos daquele médico, além de mostrar também a relação de atendimentos totais da clínica nos meses. A seguir, tem-se a Figura 31 que ilustra essa interface.



Figura 31: Página de Relatórios

4.11. PÁGINA DO ADMINISTRADOR

A página do administrador é voltada apenas para usuários especiais que possuem tal permissão no sistema, e através terão a possibilidade de gerenciar os demais usuários de acesso. A interface, quando carregada, exibe uma lista com todos os usuários registrados. O admin pode editar, excluir ou apenas visualizar qualquer usuário. Na esquerda, há um menu de navegação que permite acessar o formulário de cadastro, onde ocorrem as inserções e alterações. A Figura 32 a seguir exibe a tela descrita.

TECHMEDICINE Home Cadastros Consultas Relatórios Admin

Home

Página do Administrador

Listar Usuários

Cadastrar Usuário

#	ID	Nome de Usuário	Permissões
1	4	medico	Ver Permissões
2	3	func	Ver Permissões
3	1	admin	Ver Permissões

« 1 »

© 2021 TechMedicine, Inc. Twitter Instagram Facebook

Figura 32: Página do Administrador

5. CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo construir um sistema WEB voltado para clínicas e consultórios médicos. A justificativa para tal surgiu após análise e pesquisas acerca da informatização nesse nicho da saúde. Os resultados mostraram que através da implantação da informatização, as instituições de saúde apresentam mais qualidade no serviço oferecido.

No início, foi feito o levantamento de requisitos do *software*. Tal levantamento se mostrou essencial para o projeto, contribuindo com as decisões referentes ao andamento do mesmo. Entretanto, certos requisitos que não foram pensados corretamente tiveram de ser alterados ou mesmo cortados durante o desenvolvimento, o que causou algumas dificuldades e lentidões em certos momentos.

O desenvolvimento do sistema girou em torno de duas tecnologias principais, os *frameworks* Spring Boot e Angular. As tecnologias empregadas se mostraram muito eficientes e cumpriram com as expectativas iniciais. A utilização da linguagem Java em conjunto com o *framework* Spring Boot permitiu a construção de uma API de maneira simples e eficaz. O Angular destaca-se pela liberdade dada ao desenvolvedor em utilizar diversas bibliotecas com as mais variadas componentes visuais e funcionalidades próprias da tecnologia.

De um modo geral, não houveram muitos empecilhos ou problemas graves que impactaram severamente no desenvolvimento. Ao final do desenvolvimento, o sistema atendeu aos requisitos levantados inicialmente. Espera-se que este projeto possa atender às expectativas iniciais de melhorar o serviço oferecido pelas clínicas e contribuir com futuros trabalhos relacionados com o tema ou mesmo com desenvolvimento WEB em geral.

5.1. TRABALHOS FUTUROS

Como trabalhos futuros, pode-se pensar na inclusão de módulos de controle financeiro ao sistema. Considerando que o produto do trabalho é voltado para instituições de saúde que são privados, com fins lucrativos e que precisam gerir as movimentações financeiras, essa

funcionalidade se mostra muito pertinente. Entre as atividades que poderiam ser controladas, tem-se o cadastro de convênios, cadastro de métodos de pagamento e valores das consultas, controle de lucro e gastos, entre outras.

Outro ponto a ser pensado é a criação de relatórios mais detalhados. Os relatórios, se alinhados também com a inserção do módulo financeiro, podem ser ótimas ferramentas para os gestores das clínicas, auxiliando na geração de conhecimento sobre o andamento do negócio e na tomada de decisões.

REFERÊNCIAS

- AGNER, Luciane; LETHBRIDGE, Timothy. A survey of tool use in modeling education. In: 2017 ACM/IEEE 20TH INTERNATIONAL CONFERENCE ON MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS (MODELS), 20, 2017, Austin, USA. **MODELS '17: Proceedings of the ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems**, setembro, 2017. p. 303-11.
- AYAAD, Omar; ALLOUBANI, Aladeen; ALHAJAA, Eyad Abu; FARHAN, Mohammad; ABUSEIF, Sami; HROUB, Ahmad Al; AKHU-ZAHEYA, Laila. The role of electronic medical records in improving the quality of health care services: Comparative study. **International journal of medical informatics**, v.127, n.1, julho, 2019. p.63-7.
- BIERMAN, Gavin; ABADI, Martín; TORGERSEN, Mads. Understanding typescript. In: EUROPEAN CONFERENCE ON OBJECT-ORIENTED PROGRAMMING, 28, Uppsala, Suécia. **ECOOP 2014 – Object-Oriented Programming**. v.8586, julho/agosto, 2014. p. 257-81.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. 2. ed. Tradução de Fábio Freitas da Silva e Cristina de Amorim Machado. Rio de Janeiro: Editora Elsevier, 2006.
- CARVALHO, Thiago Leite e. **Orientação a Objetos: Aprenda seus conceitos e suas aplicabilidades de forma efetiva**. 1. ed. São Paulo: Editora Casa do Código, 2016.
- EDRAW SOFTWARE. **EdrawMax Guide**. Disponível em: <https://www.edrawsoft.com/guide/edrawmax/>. Acesso em: 14 mar. 2022.
- FLANAGAN, David. **JavaScript: o guia definitivo**. 6. ed. Tradução de João Eduardo Nóbrega Tortello. Porto Alegre: Editora Bookman, 2004.
- FINK, Gil; FLATOW, Ido. **Pro Single Page Application Development: Using Backbone. Js and ASP. Net**. 1. ed. [S.l.]: Editora Apress, 2014.
- FORNAZIN, Marcelo; JOIA, Luiz Antonio. Articulando perspectivas teóricas para analisar a informática em saúde no Brasil. **Saúde e Sociedade**, v.24, n.1, janeiro/março, 2015. p. 46-60.

FOWLER, Marin. **UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos**. 3. ed. Tradução por João Tortello. Porto Alegre: Editora Bookman, 2005.

FREEMAN, Adam. **The Definitive Guide to HTML5**. 1. ed. [S.l.]: Editora Apress, 2011.

HANJURA, Anubhav. **Heroku Cloud Application Development**. 1. ed. Birmingham: Editora Packt Publishing, 2014.

HEAGNEY, Joseph. **Fundamentals of Project Management**. 5. ed. Nova Iorque: AMACOM, 2016.

HORSTMANN, Cay. **Conceitos de Computação com Java**. 5. ed. Tradução por Edson Furmankiewicz. Porto Alegre: Editora Bookman, 2009.

KHALIFA, Mohamed. Perceived Benefits of Implementing and Using Hospital Information Systems and Electronic Medical Records. In: INTERNATIONAL CONFERENCE ON INFORMATICS, MANAGEMENT, AND TECHNOLOGY IN HEALTHCARE, 15, 2017, Atenas, Grécia. **Informatics Empowers Healthcare Transformation**, v.238, julho, 2017. p. 165-8.

LI, Qing; CHEN, Yu-Liu. **Modeling and Analysis of Enterprise and Information Systems**. 1. ed. Berlim: Editora Springer, 2009.

LUNARDI, Guilherme Lerch; DOLCI, Pietro Cunha; MAÇADA, Antônio Carlos Gastaud. Adoção de tecnologia de informação e seu impacto no desempenho organizacional: um estudo realizado com micro e pequenas empresas. **Revista de Administração**, v.45, n.1, janeiro/março, 2010. p. 5-17.

MARIN, Heimar de Fátima. Sistemas de informação em saúde: considerações gerais. **Journal of Health Informatics**, v.2, n.1, janeiro/março, 2010. p. 20-4.

MCFARLAND, David Sawyer. **CSS3: the missing manual**. 3. ed. Sebastopol: Editora O'Reilly Media, 2012.

MILANI, André. **PostgreSQL-Guia do Programador**. 1. ed. São Paulo: Editora Novatec, 2008.

MONTAÑEZ-VALVERDE, Raúl A.; MONTENEGRO-IDROGO, Juan J.; VÁSQUEZ-ALVA, Rolando. Pérdida de información en historias clínicas: Más allá de la calidad en el registro. **Revista médica de Chile**, v.143, n.6, junho, 2015. p. 812.

PINHEIRO, Antônio Carlos da Fonseca Braga. **Mapas mentais: aprenda a expressar suas ideias de forma inteligente**. 1. ed. São Paulo: Editora Saraiva, 2021.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software: uma abordagem profissional**. 9. ed. Tradução de Francisco Araújo da Costa. Porto Alegre: Editora AMGH, 2021.

PROJECT MANAGEMENT INSTITUTE. **A guide to the Project Management Body of Knowledge (PMBOK guide)**. 7. ed. Newton Square: Project Management Institute, 2021.

SHARMA, T. N.; BHARDWAJ, Priyanka; BHARDWAJ, Manish. Differences between HTML and HTML 5. **International Journal Of Computational Engineering Research**, v.2, n.5, setembro, 2012. p. 1430-37.

SILVA, Maurício Samy. **JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript**. 1. ed. São Paulo: Editora Novatec, 2010.

SPRING. **Spring Tools 4**. Disponível em: <<https://spring.io/tools>>. Acesso em: 20 dez. 2021.

TSINAKOS, Avgoustos A.; BALAFOUTIS, Thanasis. A comparative survey on mind mapping tools. **Turkish Online Journal of Distance Education**, v.10, n.3, julho, 2009. p. 55-67.

VISUAL STUDIO CODE. **IntelliSense**. Disponível em: <<https://code.visualstudio.com/docs/editor/intellisense>>. Acesso em: 20 dez. 2021

WALLS, Craig. **Spring Boot in Action**. 5. ed. Nova Iorque: Editora Manning, 2015.