

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA

BRUNO WILLIAM MIRANDA SILVA

Proposta de Prontuário Eletrônico Para Apoio às Atividades da Diretoria de Qualidade de
Vida e Saúde do Servidor da UFU

Uberlândia

2025

BRUNO WILLIAM MIRANDA SILVA

**Proposta de Prontuário Eletrônico Para Apoio às Atividades da Diretoria de Qualidade de
Vida e Saúde do Servidor da UFU**

Trabalho de Conclusão de Curso à Faculdade
de Engenharia Elétrica da Universidade Federal
de Uberlândia como requisito parcial para
obtenção do título de bacharel em Engenharia
Biomédica.

Área de concentração: Engenharia Biomédica.

Orientador: Alcimar Barbosa Soares

Uberlândia

2025

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

S586	Silva, Bruno William Miranda, 1999-
2025	Proposta de Prontuário Eletrônico Para Apoio às Atividades da Diretoria de Qualidade de Vida e Saúde do Servidor da UFU [recurso eletrônico] / Bruno William Miranda Silva. - 2025.
<p>Orientador: Alcimar Barbosa Soares. Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Uberlândia, Graduação em Engenharia Biomédica. Modo de acesso: Internet. Inclui bibliografia. Inclui ilustrações.</p>	
<p>1. Engenharia biomédica. I. Soares, Alcimar Barbosa, 1965-, (Orient.). II. Universidade Federal de Uberlândia. Graduação em Engenharia Biomédica. III. Título.</p>	
CDU: 62:61	

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

BRUNO WILLIAM MIRANDA SILVA

Proposta de Prontuário Eletrônico Para Apoio às Atividades da Diretoria de Qualidade de
Vida e Saúde do Servidor da UFU

Trabalho de Conclusão de Curso à Faculdade
de Engenharia Elétrica da Universidade Federal
de Uberlândia como requisito parcial para
obtenção do título de bacharel em Engenharia
Biomédica.

Área de concentração: Engenharia Biomédica.

Uberlândia, 14/05/2025

Banca Examinadora:

Prof. Dr. Alcimar Barbosa Soares – FEELT-UFU(Orientador)

Prof. Dr. Sérgio Ricardo de Jesus Oliveira – FEELT-UFU(Membro)

Prof. Dr. Pedro Cunha Carneiro – FEELT-UFU(Membro)

À minha família, aos que estão aqui e aos que
ainda virão.

AGRADECIMENTOS

Agradeço a Deus por tudo. Agradeço à minha família aos que estão aqui e aos que ainda virão.

Ao meu pai Gero, minha mãe Tereza, meus melhores amigos, que fizeram essa jornada ser possível, mesmo à distância, com muito apoio sempre se fizeram presentes. Sem o apoio de vocês eu não estaria onde estou.

À minha companheira Luísa, com quem tenho o prazer de dividir meus sonhos, dificuldades, alegrias e tristezas, a mais de 9 anos. Obrigado por estar comigo nessa jornada, por fazer ela ser mais leve, pelas infinitas risadas que damos juntos, pelas promessas de aprender a dançar forró, por multiplicar os momentos de felicidade e dividir os momentos difíceis.

Agradeço ao professor e amigo Alcimar, que insistentemente em suas aulas tentava nos ensinar a não nos contentar com pouco, obrigado pelo incentivo, motivação e orientação nesta caminhada acadêmica.

Agradeço ao professor e amigo Sérgio, que me ensinou o poder da comunicação e a sua importância em cultivar bons relacionamentos e obrigado por me fazer olhar para a comunicação com outras lentes.

Agradeço aos professores da Engenharia Biomédica, pelos ensinamentos e aprimoramento desde o primeiro dia em que cheguei na UFU. Agradeço a todos os amigos que pude cultivar nesses anos que multiplicaram os momentos de felicidade e dividiram os momentos difíceis em todo esse processo.

“Ninguém ignora tudo. Ninguém sabe tudo.
Todos nós sabemos alguma coisa. Todos nós
ignoramos alguma coisa.”

(Freire, 2002, p. 69)

RESUMO

Este trabalho apresenta o desenvolvimento de um protótipo de Prontuário Eletrônico do Paciente (PEP) voltado para as necessidades da Diretoria de Qualidade de Vida do Servidor (DIRQS). O projeto teve como objetivos estudar prontuários eletrônicos existentes, identificar os requisitos legais relacionados à informatização na área da saúde, levantar os requisitos funcionais e não funcionais, desenvolver o diagrama Entidade-Relacionamento e prototipar o sistema com base nas necessidades identificadas. A revisão de literatura permitiu mapear funcionalidades comuns e boas práticas, enquanto a análise da legislação brasileira garantiu a conformidade do sistema com normas como a LGPD. A partir de uma metodologia estruturada, os requisitos foram definidos e modelados. O *backend* do protótipo foi desenvolvido em Node.js e Express, o banco de dados foi desenvolvido em PostgreSQL, e o *frontend* foi desenvolvido em Angular, usando HTML e CSS. Posteriormente o protótipo foi submetido a testes e simulações, que demonstraram sua funcionalidade, usabilidade e potencial para aprimorar a gestão de dados em saúde dentro da instituição. Os resultados indicam que os objetivos propostos foram alcançados, oferecendo fundamentos técnicos para futuras implementações e expansão do sistema.

Palavras-chave: DIRQS, Informática em Saúde, Prontuário Eletrônico, Prototipação de Sistemas, Sistemas de Saúde.

ABSTRACT

This work presents the development of a prototype of an Electronic Health Record (EHR) system tailored to the needs of the Directorate of Quality of Life for Public Servants (DIRQS). The project aimed to study existing electronic health record systems, identify the legal requirements related to digitalization in the healthcare field, gather both functional and non-functional requirements, develop the Entity-Relationship diagram, and prototype the system based on the identified needs. The literature review helped map common functionalities and best practices, while the analysis of Brazilian legislation ensured the system's compliance with regulations such as the LGPD (General Data Protection Law). Using a structured methodology, the requirements were defined and modeled. The prototype's backend was developed using Node.js and Express, the database was implemented in PostgreSQL, and the frontend was developed using Angular, with HTML and CSS. The prototype was later submitted to tests and simulations, which demonstrated its functionality, usability, and potential to enhance health data management within the institution. The results indicate that the proposed objectives were achieved, providing a technical foundation for future implementations and system expansion.

Keywords: DIRQS, Electronic Health Record, Health Informatics, Healthcare Systems, System Prototyping.

LISTA DE ILUSTRAÇÕES

Figura 1 – Tela do e-SUS: Tela inicial do Prontuário Eletrônico do Cidadão (v 5.3.28)	22
Figura 2 – Fluxograma de desenvolvimento do protótipo.....	31
Figura 3 – Diagrama entidade Relacionamento do protótipo.....	36
Figura 4 – Funcionamento de aplicações fullstack.....	37
Figura 5 – Objetivos e tecnologias utilizadas na construção da API.....	39
Figura 6 – Fluxograma cadastro de Novo Usuário: Verificação, Criptografia e Armazenamento.....	42
Figura 7 – Fluxograma de Login no Sistema com Verificação e Geração de <i>Token JWT</i>	43
Figura 8 – Fluxograma de Processamento de <i>Token</i> no <i>Middleware</i> de Autenticação	45
Figura 9 – Fluxograma de criptografia AES-CBC.....	46
Figura 10 – Fluxograma de descriptografia AES-CBC.....	47
Figura 11 – Conexão com banco de dados	48
Figura 12 – Organização da interface com Framework Angular.....	49
Figura 13 – Tela de registro do sistema, necessário inserir nome, <i>e-mail</i> , cpf e senha.....	53
Figura 14 – Tabela de usuários no banco de dados, preenchida com quatro usuários aguardando acesso.....	53
Figura 15 – Tabela de tipo de acesso no banco de dados, contendo todos os tipos de acesso.....	54
Figura 16 – Mensagem que os usuários sem acesso veem ao tentar fazer login no sistema.....	54
Figura 17 – Tela de gerenciar acessos, onde o administrador e o super usuário podem dar diferentes tipos de acesso a outros usuários.....	55
Figura 18 – Tabela de usuários no banco de dados, preenchida com três usuários aguardando acesso e um administrador.....	55
Figura 19 – Tabela de usuários no banco de dados, preenchida com usuários de todos os tipos de acesso.....	56
Figura 20 – Tela de Login de acesso.....	56
Figura 21 – Tela de finalização de Cadastro para Médicos.....	57
Figura 22 – Tela de agendar nova consulta - acesso de técnico administrativo.....	58
Figura 23 – Tabela de agendamento no banco de dados, preenchida com três agendamentos de exemplo.....	58
Figura 24 – Tela de calendário, acesso de técnico administrativo.....	59

Figura 25 – Tabela de agendamento no banco de dados, preenchida com três agendamentos com status: paciente-presente, desmarcado e agendado.....	59
Figura 26 – Tela de acompanhamento de solicitações.....	60
Figura 27 – Página do paciente, detalhes da consulta com status agendado.....	61
Figura 28 – Página do paciente, detalhes da consulta com status finalizado.....	61
Figura 29 – Seção de solicitações onde o paciente pode solicitar consultas.....	62
Figura 30 – Tabela de solicitações no banco de dados, preenchida com uma solicitação.....	62
Figura 31 – Seção de Calendário - acesso de profissional da saúde.....	63
Figura 32 – Histórico do paciente no prontuário eletrônico - acesso de profissional da saúde.	64
Figura 33 – Histórico do paciente expandido no prontuário eletrônico - acesso de profissional da saúde.....	64
Figura 34 – Tela de SOAP - Subjetivo no prontuário eletrônico - acesso de profissional da saúde.....	65
Figura 35 – Tela de SOAP - Objetivo no prontuário eletrônico - acesso de profissional de saúde.....	66
Figura 36 – Tela de SOAP - Avaliação no prontuário eletrônico - acesso de profissional de saúde.....	66
Figura 37 – Tela de SOAP - Plano no prontuário eletrônico - acesso de profissional de saúde.....	67
Figura 38 – Tela de geração de atestado.....	67
Figura 39 – Atestado Gerado.....	68
Figura 40 – Tela de geração de Prescrição.....	68
Figura 41 – Prescrição gerada.....	69
Figura 42 – Tela de geração de pedidos de exames e encaminhamentos.....	69
Figura 43 – Exemplo de exame gerado.....	70
Figura 44 – Tabela de consulta no banco de dados, preenchida com uma consulta exemplo.....	70
Figura 45 – Tabela de atestado no banco de dados, preenchida com um atestado exemplo....	70
Figura 46 – Tabela de exame no banco de dados, preenchida com um exame exemplo.....	71
Figura 47 – Tabela de prescrição no banco de dados, preenchida com uma prescrição exemplo.....	71

LISTA DE TABELAS

Tabela 1 – Práticas de Segurança recomendadas	23
Tabela 2 – Requisitos funcionais e não funcionais do protótipo de prontuário	31

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CSS	Cascading Sheet Style
CTIC	Centro de Tecnologia da Informação e Comunicação
DIRQS	Diretoria de Qualidade de Vida e Saúde do Servidor
HTML	<i>Hyper Text Markup Language</i>
ICP-Brasil	Infraestrutura de Chaves Públicas Brasileira
JWT	<i>JSON Web Token</i>
LGPD	Lei Geral de Proteção de Dados
NGS1	Nível de Garantia de Segurança 1
NGS2	Nível de Garantia de Segurança 2
NPM	<i>Node Package Manager</i>
PEC	Prontuário Eletrônico do Cidadão
PEP	Prontuário Eletrônico do Paciente
RCOP	Registro Clínico Orientado a Problemas
RNDS	Rede Nacional de Dados em Saúde
SBIS	Sociedade Brasileira de Informática em Saúde
SGBD	Sistema Gerenciador de Banco de Dados
SOAP	Subjetivo, Objetivo, Avaliação e Plano
SUS	Sistema Único de Saúde
UBS	Unidade Básica de Saúde
UFU	Universidade Federal de Uberlândia

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	Motivação	19
1.2	Objetivos	19
1.2.1	<i>Objetivos Específicos</i>	19
2	REFERENCIAL TEÓRICO	20
2.1	Legislações Associada à PEPs.....	20
2.2	Prontuários na literatura.....	21
2.3	Segurança dos PEPs	23
2.4	Desenvolvimento de sistemas.....	25
2.4.1	Requisitos.....	25
2.4.2	<i>Frontend.....</i>	25
2.4.3	<i>Backend.....</i>	26
2.4.4	Bancos de Dados	27
2.4.5	Criptografia.....	28
3	METODOLOGIA.....	31
3.1	Levantamento de Requisitos	32
3.2	Prototipação do Banco de Dados.....	34
3.3	Desenvolvimento da aplicação	36
3.3.1	Desenvolvimento do <i>backend</i>	37
3.3.1.1	Criação da API e definição dos <i>endpoints</i>	37
3.3.1.2	Autenticação e Segurança	40
3.3.1.2.1	Registro no sistema	41
3.3.1.2.2	Login no sistema.....	42
3.3.1.2.3	<i>Middleware</i> de autenticação.....	44
3.3.1.2.4	Criptografia de Dados Sensíveis	45
3.3.1.3	Conexão com banco de dados	47
3.3.2	Desenvolvimento do <i>frontend</i>	48
3.4	Testes da Prova de Execução do Projeto (PEP)	49
4	RESULTADOS E DISCUSSÃO.....	51
4.1.1	Registrar	52
4.2	Super usuário	54
4.3	Perfil de Administrador	54

4.4	Autenticação	55
4.4.1	Gerenciamento de acesso	55
4.4.2	Login	56
4.5	Perfil de Técnico	57
4.5.1	Marcar novas consultas.....	57
4.5.2	Acompanhamento das consultas	58
4.5.3	Acompanhamento das solicitações.	59
4.6	Perfil de Paciente	60
4.6.1	Página do paciente	60
4.6.2	Solicitações.....	62
4.7	Perfil de Profissional da Saúde	62
4.7.1	Iniciar Atendimento.....	63
4.7.2	Histórico.....	63
4.7.3	SOAP.....	64
4.7.4	Atestado.	67
4.7.5	Prescrição	68
4.7.6	Exames e Encaminhamentos.	69
4.7.7	Finalizar Consulta	70
5	CONCLUSÃO.....	72
6	TRABALHOS FUTUROS	73

1 INTRODUÇÃO

De acordo com o Conselho Federal de Medicina (CFM), o prontuário médico é um documento elaborado pelo profissional de saúde que contém informações importantes acerca do paciente, como o seu histórico familiar, descrição e evolução clínica, resultados de exames, anamnese, além das orientações terapêuticas e prescrições realizadas. (CFM, 1999).

Um dos primeiros indícios da prática de se ter um meio onde as informações acerca dos pacientes fossem registradas remete a Hipócrates no século V a.C. Mesmo antes do surgimento da medicina como ciência formal, ele estimulava curandeiros e estudiosos a fazerem registros escritos. Para Hipócrates, esses registros tinham dois objetivos: refletir de forma exata o curso da doença e indicar as possíveis causas da doença. (MASSAD et al., 2003).

Até o começo do século XIX, os médicos baseavam suas informações no que podiam ouvir, sentir e ver. Em 1816, Laennec inventou o estetoscópio (LITTMANN, 2025), o que contribuiu consideravelmente para as técnicas de diagnóstico. À medida que novas ferramentas de diagnóstico foram sendo desenvolvidas, o prontuário se tornou cada vez mais robusto.

Em 1880, William Mayo fundou com amigos a Clínica Mayo em Minnesota, nos Estados Unidos. William observou que os médicos arquivavam as informações de todos os pacientes em forma cronológica em um documento único. Esse aglomerado de informações dificultava o acompanhamento de um mesmo paciente já que era necessário passar por todo esse documento para encontrar as informações. Assim, a partir de 1907, a Clínica de Mayo adotou um registro individual das informações de cada paciente, que passaram a ser arquivadas separadamente. Esse novo método deu origem ao prontuário médico centrado no paciente, ainda seguindo uma ordem cronológica. (MASSAD et al., 2003).

Em 1920, a Clínica Mayo iniciou um movimento de padronização dos conteúdos dos prontuários, definindo um conjunto mínimo de dados que deveriam ser registrados. Essa padronização criou uma estrutura mais sistematizada das informações do paciente e do avanço do seu tratamento, caracterizando o modelo de prontuário do paciente utilizado até os dias de hoje. Porém, apesar dos esforços de padronização o prontuário ainda contém uma mistura de informações, queixas, resultados de exames, achados clínicos, o que acaba dificultando a obtenção de informações claras sobre a evolução do paciente, especialmente dos pacientes com mais de uma enfermidade. (MASSAD et al., 2003).

Em 1969, o médico americano Lawrence Weed introduziu a ideia do prontuário orientado pelo problema (*Problem-Oriented Medical Record – POMR*). Nesse método, os

problemas de saúde do paciente são identificados e organizados seguindo a estrutura SOAP, um acrônimo em inglês que representa quatro etapas: *subjective* (subjetivo): refere-se aos sintomas descritos pelo paciente. *Objective* (objetivo): abrange os achados concretos coletados pelo profissional durante a consulta médica. *Assessment* (Avaliação): o profissional de saúde integra as informações relatadas pelo paciente e os dados objetivos para chegar a uma conclusão ou possíveis diagnósticos. *Plan* (Plano): corresponde ao plano de ação definido com base na avaliação. (WEED et al., 1968).

Segundo Albergaria & Bax (2013), Lau & Price (2014) e Souza (2016), o prontuário, que inicialmente tinha a função de registrar dados sobre a saúde do paciente, histórico de doenças e dados sociodemográficos, evoluiu e passou a desempenhar um papel essencial no sistema de saúde. Dentre os seus principais benefícios estão, a preservação do histórico clínico do paciente, facilitar o compartilhamento de informações entre múltiplos profissionais da saúde, o suporte a pesquisas e estudos, além de ser o registro legal das ações médicas.

Com este avanço dos diagnósticos clínicos, a equipe que acessa essas informações tornou-se cada vez mais interdisciplinar e organizada de forma horizontal, pois cada profissional desempenha um papel essencial na recuperação do paciente. Devido à grande multidisciplinaridade do sistema e à necessidade de um acesso simultâneo e mais prático às informações, foi desenvolvido um sistema integrado que facilita essa visualização e compartilhamento: o prontuário eletrônico do paciente (PEP). De acordo com o *Institute of Medicine* (IOM, 1997) o PEP “é um registro eletrônico armazenado em um sistema projetado para auxiliar os usuários fornecendo acesso a informações completas e precisas, alertas, sistema de apoio a decisão, links para base de conhecimento médico e outras funções”.

No Brasil o PEP foi reconhecido em 2002 pelo Conselho Federal de Medicina por meio das resoluções 1.638 definindo o Prontuário médico:

Definir prontuário médico como o documento único constituído de um conjunto de informações, sinais e imagens registradas, geradas a partir de fatos, acontecimentos e situações sobre a saúde do paciente e a assistência a ele prestada, de caráter legal, sigiloso e científico, que possibilita a comunicação entre membros da equipe multiprofissional e a continuidade da assistência prestada ao indivíduo.

A utilização do prontuário eletrônico contém diversas vantagens, a seguir serão destacadas algumas que são consideradas mais importantes sobre o tema segundo a literatura, de acordo com Powsner, Wyatt e Wright (1998):

- Redução no tempo e no custo do atendimento;
- Compartilhamento de informações por meio de acesso remoto e simultâneo;
- Facilidade na organização e no acesso às informações;
- Otimização do espaço físico destinado ao armazenamento.
- Segurança de dados, é comum os usuários preocuparem-se com a segurança de dados, devido a mau funcionamento de sistemas. No entanto em um sistema bem projetado com um bom sistema de backup e planos contra desastres o PEP é muito mais confiável;
- Confidencialidade dos dados, dado que apenas usuários autorizados poderão acessar o sistema, além disso cada usuário deve ter um nível de acesso específico;
- Integração com outros sistemas de informação do hospital, como de farmácia e de faturamento;
- Captura automática de dados, no qual os dados fisiológicos podem ser automaticamente capturados, de monitores, análises de laboratório, equipamentos de imagens localizados em qualquer parte do hospital. Evitando erros de digitação e extravio desses documentos.
- Atualização constante dos dados, pois sendo o PEP integrado a todo o hospital todos os dados estão disponíveis imediatamente a todos os médicos do sistema.
- Construção de relatórios, os dados integrados do PEP podem ser impressos em relatórios que auxiliam os médicos, com indicadores, chamando a atenção para detalhes que possam passar desapercebidos.

Segundo Panitz (2014), é perceptível que a implantação de um PEP traz diversos avanços para uma instituição, quando bem projetado, tendo em vista todos os pontos de dificuldade e desafios que podem ser encontrados no seu desenvolvimento, implementação e manutenção. Entre os principais obstáculos, destacam-se:

- Grande investimento com hardware e *software*;
- Investimento em treinamentos dos profissionais que utilizarão o sistema;
- A resistência de grande parcela dos profissionais em utilizar um sistema computacional;
- Problemas técnicos imprevistos que podem atrapalhar o uso do sistema;
- O acréscimo no tempo de trabalho e de atendimento devido à dificuldade de registro de dados nos sistemas;
- O receio dos profissionais de saúde em expor as suas práticas clínicas, para visualização pelos outros profissionais;

- Resistência no uso da tecnologia por receio de falhas de segurança e confiabilidade do sistema;
- Sistemas não intuitivos, que acabam dificultando o processo de assistência e impactando na proximidade entre paciente e médico;
- Demora em se perceber os resultados da implantação do PEP, pois ele se torna mais robusto e relevante com o tempo de utilização e quantidade de dados inseridos nele;
- Adoção de sistemas que são incompatíveis com outros sistemas já utilizados pela instituição, impedindo a interoperabilidade.

Segundo Lima, Vieira e Nunes (2018), as Tecnologias da Informação (TI) na área da saúde têm como objetivo central aplicar tecnologias no gerenciamento e assistência do cuidado. Essa abordagem busca aprimorar o acolhimento do paciente e oferecer um serviço mais humanizado. Combinando conhecimentos científicos de medicina, enfermagem, computação e informação, essa especialidade se desenvolve há mais de 35 anos. O foco está em potencializar o ensino, a pesquisa e a prática dos profissionais de saúde, promovendo um cuidado sistematizado, ágil e de alta qualidade (MELO; ENDERS, 2013).

A Pró-Reitoria de Gestão de Pessoas da Universidade Federal de Uberlândia (UFU) possui uma unidade organizacional denominada de Diretoria de Qualidade de Vida do Servidor (DIRQS), a DIRQS tem como missão formular diretrizes e coordenar a supervisão das questões relacionadas à assistência biopsicossocial, à saúde ocupacional, à vigilância e à perícia em saúde em conformidade com as normas do Governo Federal. Seu objetivo central é atender aos servidores da UFU e órgãos federais, promovendo a saúde dos trabalhadores, com ênfase na legalidade das ações e na conscientização sobre a prevenção. Seus trabalhos abrangem a atenção primária e secundária, priorizando abordagens interdisciplinares e uma visão holística de saúde e qualidade de vida. (PROGEP/UFU, 2025a).

A DIRQS contém o Ambulatório de Saúde do Servidor (Asser), que oferece consultas médicas, atendimentos de fisioterapia e de enfermagem, que é disponibilizado para servidores e seus dependentes, trabalhadores e empregados da Ebserh (PROGEP/UFU, 2025b). O prontuário utilizado no Asser é um prontuário de papel e neste trabalho será feito um estudo da implementação de um prontuário eletrônico e um protótipo que supra as necessidades da Asser, e de outras unidades de atendimento da DIRQS.

Este trabalho teve parceria com o Centro de Tecnologia da Informação e Comunicação (CTIC) da UFU com supervisão do Fabiano Silvério Ribeiro Alves e validação dos requisitos / funcionalidades pela analista de sistemas do CTIC Amanda Gaspar Pereira.

1.1 Motivação

A crescente demanda por atendimentos eficientes e bem documentados em instituições públicas, como a Diretoria de Qualidade de Vida do Servidor da UFU (DIRQS), evidencia a necessidade de soluções tecnológicas que otimizem o fluxo de informações e melhorem a experiência de profissionais e pacientes. Nesse contexto, o desenvolvimento de um protótipo de prontuário eletrônico surge como uma ferramenta estratégica para automatizar processos, reduzir o uso de documentos físicos e permitir o acesso rápido e organizado ao histórico clínico do servidor, servindo também como base para futuras implementações e aprimoramentos do sistema.

1.2 Objetivos

Este trabalho teve como objetivo desenvolver um protótipo de sistema de prontuário eletrônico para a Diretoria de Qualidade de Vida do Servidor da Universidade Federal de Uberlândia (DIRQS), visando solucionar a ausência de um sistema informatizado que dificulta o registro, acesso e organização dos dados clínicos dos servidores. A proposta busca automatizar processos para aumentar a agilidade no atendimento e proporcionar acesso facilitado ao histórico dos pacientes.

1.2.1 Objetivos Específicos

- Realizar um estudo acerca de outros prontuários eletrônicos na literatura e identificar os requisitos que se adequem ao sistema a ser desenvolvido;
- Identificar as legislações que regem a informatização de sistemas no âmbito da saúde;
- Levantamento de requisitos funcionais e não funcionais do Prontuário eletrônico;
- Desenvolvimento do diagrama de Entidade e Relacionamento;
- Prototipação do PEP;
- Testes e simulações.

2 REFERENCIAL TEÓRICO

A seguir serão abordados tópicos importantes para a elaboração deste trabalho. Serão abordadas as legislações associadas aos PEPs, levantamento de prontuários na literatura e aspectos de segurança dos prontuários. Após esse levantamento, serão abordadas questões de desenvolvimento de sistemas: o *frontend*, o *backend*, o banco de dados e as técnicas de criptografia utilizadas no desenvolvimento.

2.1 Legislações Associada à PEPs

Segundo Bezerra (2015), o prontuário eletrônico traz grandes vantagens para a assistência médica, como o aumento da qualidade no cuidado aos pacientes, a facilidade de acesso aos serviços disponíveis, o aumento da produtividade e até mesmo a redução dos custos. É de suma importância salientar que a sua implementação deve ser regida por leis, sendo elas: a resolução do CFM Nº 1821/2007, a Lei Geral de Proteção de Dados (LGPD). Além disso o sistema de prontuário eletrônico deve conter um conjunto de funcionalidades específicas definidas pelo Conselho Federal de Medicina (CFM) em conjunto com a Sociedade Brasileira de Informática em Saúde (SBIS).

A resolução do CFM Nº 1821/2007 veio para aprovar as normas técnicas para o uso de sistemas informatizados para a guarda e manuseio de prontuários dos pacientes, autorizando a eliminação do prontuário em papel e a troca de informações na saúde. No artigo 2º a resolução autoriza a digitalização dos prontuários, desde que o sistema obedeça à norma específica e “após análise obrigatória da Comissão Permanente de Avaliação de Documentação da unidade-médico hospitalar geradora do arquivo”.

A SBIS em conjunto com o CFM (2016) classificou os prontuários eletrônicos, segundo a segurança da informação em dois níveis de segurança: Nível de garantia de segurança 1 (NGS1) e Nível de Garantia de Segurança 2 (NGS2). O NGS1 é aplicável a prontuários eletrônicos que não desejam eliminar os registros em papel, já o NGS2 viabiliza a eliminação dos registros em papel, utilizando os certificados digitais da Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil) para a assinatura e identificação. É importante ressaltar que para atingir o NGS2 o sistema também atenda aos requisitos NGS1.

Segundo o Instituto Nacional de Tecnologia e Informação (ITI), A ICP-Brasil é uma cadeia hierárquica de confiança que viabiliza a emissão de certificados digitais para identificação virtual de pessoas ou empresas de forma segura e com validade jurídica. Dentre

os benefícios de se ter um certificado digital ICP-Brasil, está a segurança, pois a infraestrutura é monitorada, fiscalizada e está em constante atualização.

2.2 Prontuários na literatura.

O Ministério da Saúde disponibilizou para uso gratuito de todos os municípios o Prontuário Eletrônico do Cidadão (PEC), em 2013, com o objetivo de facilitar o processo de informatização das unidades básicas de saúde (UBS) em todo o território brasileiro. O PEC foi construído como base no Registro Clínico Orientado a Problemas (RCOP), que possui três áreas essenciais para o registro das informações clínicas, a base de dados do cidadão, a lista de problemas e as notas de evolução clínica, estruturadas no formato SOAP (CELUPPI et al., 2024). O PEC atualmente está em sua versão 5.3.28, que foi disponibilizada em março de 2025 e a interface gráfica pode ser vista na Figura 1.

O Prontuário Eletrônico do Cidadão (PEC) permite que os profissionais de saúde realizem o registro das consultas em diversas abas, as quais oferecem diferentes funcionalidades, como demonstrado na ferramenta de Atendimentos do sistema e-SUS APS (ATENDIMENTOS, Estratégia e-SUS APS. Disponível em: https://saps-ms.github.io/Manual-eSUSAPS/docs/PEC/PEC_06_atendimentos/#61-lista-de-atendimentos. Acesso em: 4 abr. 2025). Essas funcionalidades são:

- Folha de rosto: este recurso oferece uma visão geral do quadro clínico do paciente, facilitando o acesso às informações mais importantes relacionadas à saúde do paciente;
- SOAP: este recurso orienta o registro de informações subjetivas e clínicas do paciente, o que permite a definição de diagnósticos, o planejamento de condutas, e a avaliação das condições de saúde averiguadas durante o atendimento;
- Vacinação: esta ferramenta exibe uma versão digital da caderneta de vacinação do paciente;
- Histórico: esta ferramenta permite consultar de forma detalhada os atendimentos passados do paciente;
- Cadastro do Cidadão: funcionalidade que permite o acesso rápido a informações cadastrais do paciente, facilitando a conferência e atualização dos dados;
- Agendamento: é possível visualizar todos os agendamentos futuros e passados do paciente.

- Acessar RNDS: possibilita ao profissional de saúde consultar informações clínicas do cidadão registradas em outros serviços de saúde e compartilhados por meio da Rede Nacional de Dados em Saúde (RNDS).

Figura 1: Tela do e-SUS: Tela inicial do Prontuário Eletrônico do Cidadão (v 5.3.28)

Fonte: Ministério da Saúde, 2025. Disponível em: https://saps-ms.github.io/Manual-eSUS_AP/PEC/PEC_06_atendimentos/#61-lista-de-atendimentos. Acesso em: 4 abr. 2025.

Ao longo desses 12 anos de PEC, muitos módulos e funcionalidades foram lançadas em atualizações, como a declaração de comparecimento ao atendimento, sistema de agendamento online, módulos de vídeo chamada e prescrições eletrônicas. (CELUPPI et al., 2024).

Um grande problema para o SUS é o e absentismo, uma palavra de origem latina que significa estar fora ou ausente, no setor da saúde pode ser entendida como falha no atendimento ou o não atendimento do paciente. Os principais motivos para o não comparecimento dos usuários foram elencados por estudos e são: esquecimento, falta ou falha de comunicação entre paciente e unidade de saúde, agendamentos em horários inconvenientes, dificuldade do acesso ao serviço de saúde e agendamentos marcados com muito tempo de antecedência. Além disso, o cenário atual em muitas unidades de atenção primária a saúde (APS) é caracterizado pela formação de filas para conseguir o atendimento, muitas vezes o paciente chega a pernoitar para conseguir uma ficha de atendimento. Isso prejudica o acesso do usuário aos serviços de saúde. (POSTAL et al., 2021).

Pesquisas indicam que organizar o agendamento de consultas de acordo com a necessidade do usuário pode aumentar o acesso aos serviços de saúde, melhorar a qualidade do cuidado, favorecer o acompanhamento das condições clínicas, tornar o atendimento mais humano, além de contribuir para a redução dos custos ao sistema de saúde. (TURKCAN et al., 2011; BRASIL, 2007).

Diante desse cenário, tornou-se necessário que o sistema público de saúde brasileiro repensasse suas formas de organizar o trabalho, com o objetivo de ampliar o acesso dos usuários aos serviços, adotando um sistema de agendamento. Esse recurso teve como objetivo principal facilitar o processo de marcação de consultas, permitindo que o usuário realize o agendamento pelo aplicativo, sem a necessidade do paciente se deslocar até a unidade de saúde. Melhorando o acesso ao serviço de saúde. (POSTAL et al., 2021).

Purkayastha et. al. no artigo *Comparison of Open-Source Electronic Health Record Systems Based on Functional and User Performance Criteria* discute a evolução e os impactos dos sistemas de informação em saúde (SIS) na organização e prestação de serviços da área da saúde. Ele apresenta os principais conceitos relacionados à tecnologia da informação aplicada à saúde, e destaca a importância da interoperabilidade, padronização e integração dos dados para melhorar a qualidade, segurança e eficiência dos cuidados prestados.

Além disso, o artigo reforça os desafios enfrentados na implementação de sistemas de saúde eletronicamente, como por exemplo a resistência organizacional, necessidade de infraestrutura e tecnologia adequada, e questões éticas relacionadas à privacidade e segurança da informação. Contudo, apesar desses obstáculos, a pesquisa destaca que os avanços na área vêm promovendo mudanças significativas nas práticas administrativas e clínicas, o que contribui para a modernização da saúde pública e privada. (PURKAYASTHA et al., 2019).

Com base na análise comparativa de prontuários eletrônicos de código aberto usados ao redor do mundo, observou-se que existem requisitos fundamentais e que são atendidos por todos os sistemas analisados. Esses requisitos, parcialmente ou totalmente comuns aos prontuários *OpenEMR*, *OpenEHR*, *OpenMRS*, *GNU Health* e *OSHERA VistaA*, que evidenciam um padrão mínimo de funcionalidades e medidas de segurança. (PURKAYASTHA et al., 2019). São eles:

- Informações e dados de saúde;
- Gerenciamento de resultados;
- Suporte à decisão;
- Processos administrativos;
- Autenticação, controle de acesso e autorização;
- Integridade.

2.3 Segurança dos PEPs

No livro “*For the Record: Protecting Electronic Health Information* (CSTB, 1997), há um relato de como proteger a informação nos sistemas eletrônicos na área, que estão resumidos

no quadro I. Para se chegar a níveis aceitáveis de segurança e confidencialidade a maioria das recomendações devem ser implementadas.

Quadro I – Práticas de Segurança recomendadas

Práticas de Segurança recomendadas
<ul style="list-style-type: none"> • Autenticação Individual de Usuários: todos os usuários devem ter um identificador único, para uso individual em todos os sistemas da instituição; • Controle de acesso: os usuários devem ter acesso apenas a informações que pertencem ao seu perfil de acesso, bloqueando o usuário de ter acesso a conteúdos que não tenham direito; • Trilha de auditoria: deve-se manter um registro de todos os acessos a informações clínicas, ficando disponível para auditoria quando necessário. • O acesso aos dispositivos e prontuários deve ser restrito. Deve haver plano de acesso em caso de desastre, com backups seguros e criptografados. • Proteção ao Acesso Remoto: Utilizar firewalls para limitar o acesso apenas a máquinas autorizadas, além de considerar métodos adicionais para múltiplos locais ou acesso remoto. • Proteção à Comunicação Externa: Encriptar dados transmitidos por redes públicas, como a internet. • Disciplina de <i>Software</i>: Limitar o uso de <i>softwares</i> nas estações de trabalho para evitar infecção por vírus. • Avaliação do Sistema: As instituições devem avaliar formalmente a segurança e vulnerabilidades de seus sistemas de informação.

Fonte: Adaptado de "For the Record: Protecting Electronic Health Information" (CSTB, 1997).

Os principais obstáculos para o uso do PEP são relacionados à segurança e confidencialidade dos dados dos pacientes. O que gera discussões sobre o assunto, e com o avanço da tecnologia os ataques hackers ficam cada vez mais robustos, o que traz à tona essa preocupação regularmente. Por isso a segurança do PEP é um trabalho contínuo, se atentando sempre com novas tecnologias e tipos de ameaças.

Para auxiliar os desenvolvedores a atuarem com cada vez mais segurança, existe o Projeto de Segurança de Aplicações Web de Código Aberto, conhecido como OWASP, uma entidade internacional voltada à proteção de sistemas online. O projeto mais conhecido da OWASP é o OWASP Top 10, que indica as 10 principais vulnerabilidades em aplicações web (OWASP, 2025), as que mais se aplicam ao desenvolvimento do protótipo do PEP são:

- Falhas de Criptografia: a criptografia é crucial para proteger os dados sensíveis dos pacientes;

- Falhas de Identificação e Autenticação: técnicas robustas de autenticação são essenciais para garantir que apenas os usuários autorizados possam acessar o sistema;
- Desing Inseguro: é importante implementar práticas de desenvolvimento seguro, que garantam que o sistema seja revisado periodicamente para identificar e corrigir possíveis vulnerabilidades;

2.4 Desenvolvimento de sistemas.

Para desenvolver um prontuário eletrônico eficiente, é fundamental compreender as etapas essenciais do desenvolvimento web. Esse processo é dividido em três fases principais: levantamento e a análise de requisitos, desenvolvimento do *frontend* e desenvolvimento do *backend*. (GONÇALVES et al., 2005).

2.4.1 Requisitos

Os requisitos de um sistema, conforme definido por *Sommerville* (2011), consistem nas descrições detalhadas sobre o que o sistema deve realizar, os serviços que deve oferecer e as restrições ao seu funcionamento. Esses requisitos refletem as necessidades e expectativas dos stakeholders do *software*, sendo declarações precisas que especificam o comportamento esperado do sistema, bem como os critérios de desempenho e qualidade que devem ser atendidos.

Os requisitos de *software* são comumente classificados como requisitos funcionais e requisitos não funcionais. Os requisitos funcionais (RF) são informações específicas que o *software* precisa executar. Eles definem os serviços e ações que o sistema deve fornecer ao usuário. Eles descrevem o que o sistema deve fazer. Por exemplo, um sistema de pedidos de pizzaria deve permitir que o cliente escolha a pizza que deseja.

Os Requisitos Não Funcionais (RNF) focam em como o sistema será implementado. Eles estão frequentemente implícitos nas solicitações dos stakeholders, como conformidade com legislações, boa usabilidade, baixa latência e definição da linguagem de programação. No exemplo do sistema de pedidos da pizzaria, um requisito não funcional seria a linguagem de programação escolhida para o desenvolvimento do *frontend*.

2.4.2 Frontend

Para a estruturação dos sistemas web a linguagem utilizada é o *Hyper Text Markup Language* (HTML), que é traduzido para o português como linguagem de marcação de

hipertexto, e da forma ao esqueleto das páginas web. O hipertexto é todo o conteúdo inserido na página web que tem como principal característica se ligar a outros conteúdos da web. É utilizado para criar toda a estrutura das páginas por meio das *tags*, para indicar onde o elemento começa e onde termina, e cria toda a estrutura, como cabeçalho, título, parágrafo, imagens e muitos outros. (SILVA, 2014).

Para a estilização das estruturas criadas é usada a linguagem de *Cascading Sheet Style* (CSS) o que permite que o desenvolvedor altere fontes, cores, espaçamentos, e crie layouts mais interativos e com maior usabilidade e interatividade com o usuário. (MEYER, 2023).

Para adicionar o comportamento a linguagem de programação utilizada é o *JavaScript*, uma linguagem de alto nível, dinâmica e não tipada que permite tornar as páginas dinâmicas e interativas. Torna possível a captura de eventos como cliques, teclas pressionadas, manipular elementos em tempo real, validar formulários e muitas outras ações. Além disso, o *JavaScript* é responsável pela integração da API entre o *backend* e o *frontend*. (Flanagan, 2012).

Esses Três Pilares, HTML, CSS e *JavaScript*, trabalham juntos para criar aplicações web responsivas e interativas. Enquanto o HTML define a estrutura, o CSS estiliza e o *JavaScript* adiciona a interatividade, tornando possível o desenvolvimento de sites simples até sistemas mais complexos. (Flanagan, 2012).

Existem diversos frameworks para o Desenvolvimento *Frontend*. Para o desenvolvimento deste trabalho o Angular foi escolhido; este é um framework de código aberto criado pelo Google em 2016. Utilizando *TypeScript*, o Angular permite o desenvolvimento de aplicações robustas e escaláveis, que facilita a organização, testes de código e a manutenção. A popularidade do Angular vem da sua grande comunidade ativa e da ampla variedade de bibliotecas e ferramentas disponíveis. (Jr. and Mercado 2018).

2.4.3 Backend

Atualmente existem diversas tecnologias para o desenvolvimento do *backend* das aplicações.

O Node.js permite a execução do *JavaScript* no *backend*, tornando possível programar em *JavaScript* diretamente na máquina, sem depender de um navegador. A arquitetura do Node.js é construída sobre um único thread de execução, tem um modelo engenhoso de programação assíncrona e é orientada a eventos. O que permite que múltiplas operações sejam processadas simultaneamente sem bloquear a execução do código, melhorando a performance

em aplicação que fazem muitas operações de entrada e saída, como *Application Programming Interface* (API) e serviços web em tempo real. (CASCIARO; MAMMINO, 2020).

O Node.js possui diversos frameworks, um deles é o Express.js, que é baseado no módulo HTTP principal do Node.js. Os sistemas desenvolvidos em Express.js são altamente configuráveis, permitindo que o desenvolvedor escolha as bibliotecas necessárias para cada projeto. O que proporciona alta flexibilidade e alta personalização no desenvolvimento de aplicações Web. (Mardan, 2018).

As principais vantagens de se utilizar o Express.js listadas por Mardan no livro *Practical Node.js* estão listadas a seguir:

- Minimalista: o Express.js possui um design simples, ajudando a reduzir a curva de aprendizado para novos desenvolvedores, permitindo o desenvolvimento de forma mais ágil e fácil.
- Flexibilidade: É altamente personalizável e flexível, um exemplo disso é o sistema de *middleware* que permite ao desenvolvedor adicionar lógica personalizada às requisições recebidas ou às respostas enviadas;
- Escalável: suporta programação assíncrona, o que permite a criação de aplicações escaláveis que lidam com muitas requisições simultaneamente;
- Compatibilidade com Node.js: foi desenvolvido exclusivamente para funcionar com o Node.js, sendo compatível com Node.js e suas bibliotecas;
- Comunidade Colaborativa: possui uma comunidade grande e ativa e por isso está em constante evolução. Ter uma comunidade robusta ajuda os desenvolvedores com erros e dificuldades ao longo do desenvolvimento.

O Node.js é amplamente suportado pelo *Node Package Manager* (NPM), que é um gerenciador de pacotes de *software*, um dos principais facilitadores para o uso de *software* desenvolvido por terceiros. O uso de pacotes desenvolvidos por terceiros tem se tornado cada vez mais popular, permitindo que equipes reduzam o tempo e custo do desenvolvimento, além de aumentar a produtividade. (CHOWDHURY, 2021).

2.4.4 Bancos de Dados

Um banco de dados é um conjunto estruturado de dados, desenvolvido para simbolizar atributos do mundo real, também chamados de minimundo ou universo discurso. Os dados são organizados de forma lógica e coerente e tem o intuito de atender às necessidades de um grupo de usuários específicos. Além disso um banco de dados tem estrutura e finalidade bem

definidas, podendo variar em tamanho e complexidade, desde uma simples agenda até sistemas de órgãos governamentais. (Elmasri e Navathe, 2005).

Os bancos de dados podem ser mantidos de forma digital, por meio de um Sistema Gerenciador de Banco de Dados (SGBD). O SGBD é um conjunto de programas que auxilia na criação, definição, manipulação e compartilhamento dos dados armazenados. Ele permite especificar tipos e estruturas de dados, realizar buscas e atualizações, a geração de relatórios além de muitas outras ações. Portanto, o banco de dados não é apenas um repositório de informações, mas uma ferramenta dinâmica que auxilia a organização, a análise e a tomada de decisões. (Elmasri e Navathe, 2005).

O *PostgreSQL* é um SGBD gratuito e de código aberto, e é reconhecido pela sua robustez, desempenho e conformidade com padrões SQL. Ele é considerado o SGBD open source mais avançado por esses motivos: apesar de ser um SGDB de alto nível de performance, tem uma usabilidade bem amigável na utilização do dia a dia, utiliza linguagem SQL, tem alto nível de conectividade com a internet e, por isso, foi escolhido para o desenvolvimento deste trabalho. (CALDEIRA, 2015).

2.4.5 Criptografia

Criptografia, vem dos termos gregos: *kryptos*, que significa oculto, escondido e *graphien* que significa escrever. Seu objetivo não é esconder a mensagem em si, mas sim ocultar o seu significado. Deste modo ao caírem em mãos erradas, sua compreensão será difícil. Apenas o remetente e o destinatário, que compartilham um acordo pré-estabelecido de criptografia, têm acesso ao significado da mensagem (Fiarresga, 2010).

Simon Singh (1999) no livro *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*, se aprofunda na história da criptografia, no século XVI, Mary, rainha da Escócia, conspirou com o lorde Babington para assassinar a Elizabeth I, rainha da Inglaterra, e tomar o trono. Para se comunicarem, utilizaram mensagens criptografadas por um sistema frágil. Este método foi rapidamente decifrado pelo decodificador da rainha Elizabeth, revelando toda a conspiração e sentenciando os envolvidos à execução. Esse episódio ilustra o perigo de confiar em criptografia fraca.

Na implementação de sistemas é uma boa prática fazer a criptografia das senhas, para deixar o sistema seguro e de acordo com as normas de segurança. E para isso são usadas funções de *hashings*. O *hashing* elimina a necessidade de uma chave criptográfica. Além disso, o *hashing* torna os dados irreversíveis, ou seja, uma vez processados, não podem ser convertidos

de volta ao seu estado original, o que aumenta a proteção contra acessos indevidos. (Menezes, 2018).

No entanto o *hashing* ainda tem as suas falhas, se duas senhas iguais forem processadas pela mesma função de *hash*, gerarão o mesmo *hash*, o que a princípio isso parece um problema. No entanto, um indivíduo mal-intencionado que obtenha acesso a um banco de dados de senhas e *e-mail*, pode recorrer a um *rainbow table*, testando senhas comumente utilizadas e aplicando a mesma função de *hash* para gerar uma lista de *hashes* correspondentes a senhas fracas. Assim, ao comparar os *hashes* armazenados no banco de dados com os da tabela, ele pode descobrir a senha de diversos usuários. (Boneh, 2020).

Um avanço nas funções de *hashing* é o *Salting* que pode ser traduzido como salgar. Em adição à informação que será salva, adiciona-se um conjunto de caracteres aleatórios, e então a informação é processada pela função de *hashing*. Agora duas senhas iguais processadas pela mesma função *hashing* serão diferentes, pois cada uma contém um *salt*. Isso aumenta a segurança das informações, mas ainda não é muito seguro (RATHOD; SONKAR; CHANDAVARKAR, 2020).

O *bcrypt* é uma função de *hasing* que contém *Salt Rounds*, para aumentar a segurança. O processo funciona da seguinte forma: inicialmente, um *salting* é adicionado à informação e é processada pela função *hash*, isso categoriza um ciclo. Para os ciclos seguintes é adicionado o mesmo valor de *salting* ao *hash* gerado no ciclo anterior e é processada novamente pela função de *hash*, até os ciclos serem concluídos. Quanto mais salgado o *hash* salvo for, mais seguro ele será. (Provos, 1999).

Para proteger os dados sensíveis, são utilizados algoritmos de criptografia simétricos, método que usa uma chave para criptografar e descriptografar dados. Um desses algoritmos de criptografia simétrica é o *Advanced Encryption Standard* (AES). O AES trabalha com blocos de 128 bits e aceita chaves de 128, 192 ou 256 bits. O texto de entrada é transformado em uma matriz quadrada de bytes 4x4. O número de rodadas do algoritmo varia conforme o tamanho da chave: são 10 rodadas para chaves de 128 bits, 12 para 192 bits e 14 para 256 bits. Em cada rodada, são realizadas transformações como substituição de bytes, deslocamento de linhas, mistura de colunas e, por fim, a adição da chave de criptografia. Essas operações, combinadas, garantem a robustez e a segurança do algoritmo. (NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, 2001).

O modo de operação *Cypher Block Chaining* (CBC) complementa o AES. Nele, cada bloco de texto é misturado com o bloco anterior já criptografado, antes de ser criptografado. No

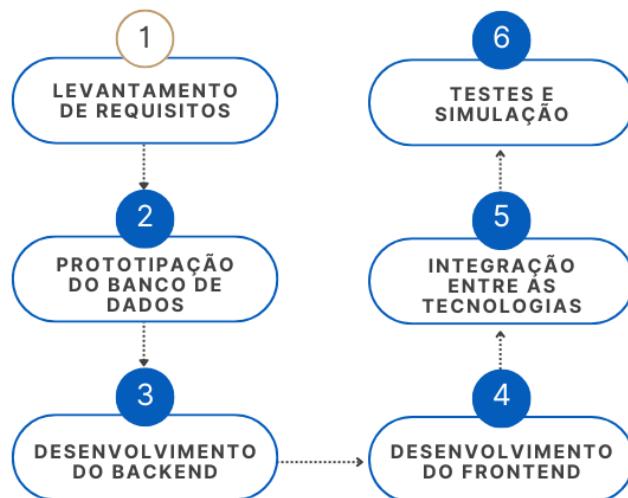
primeiro bloco, é adicionado um vetor de inicialização (VI), que garante um resultado único. Tem a mesma lógica do *salting* e impede que dados iguais gerem códigos iguais, dificultando a descoberta de padrões na mensagem criptografada. (BLAZHEVSKI, 2013).

3 METODOLOGIA

O desenvolvimento do prontuário eletrônico na web foi implementado seguindo as etapas: levantamento de requisitos, prototipação de banco de dados, desenvolvimento do *backend*, desenvolvimento do *frontend* e a integração entre as tecnologias. O fluxograma de desenvolvimento do protótipo pode ser visualizado na Figura 2. O intuito do *software* é propiciar melhoria nos atendimentos dos pacientes na área da saúde, com a centralização do acesso às informações dos pacientes, evitar extravio de informações oficiais e otimizar o tempo dos profissionais da saúde. O *software* é um protótipo que atenda às necessidades das unidades da Universidade Federal de Uberlândia que oferecem atendimento na área da saúde a pacientes, como a Diretoria de Qualidade de Vida do Servidor (DIRQS).

Figura 2: Fluxograma de desenvolvimento do protótipo.

Fluxograma do desenvolvimento do Protótipo



Fonte: Autoria própria.

3.1 Levantamento de Requisitos

O levantamento de requisitos é uma etapa fundamental no desenvolvimento de sistemas, pois permite compreender as necessidades reais dos usuários e delimitar as funcionalidades que o *software* deverá oferecer. Para isso, foram utilizados sistemas similares para coletar informações relevantes sobre o prontuário eletrônico, que foram descritos na seção de prontuários na literatura. A partir dessa coleta, foi possível elaborar a especificação do sistema, que serviu de base para o desenvolvimento do sistema e das próximas etapas. Os requisitos funcionais podem ser divididos em: acesso ao sistema, gestão de atendimentos e gestão de prontuário eletrônico. Esses requisitos podem ser visualizados na Tabela 1.

Tabela 1: Requisitos funcionais e não funcionais do protótipo de prontuário

Identificador	Requisito	Descrição
Requisitos Funcionais		
Acesso ao Sistema		
RF1.1	Autenticação	O sistema deverá exigir “login” e senha para que os usuários acessem as funcionalidades.
RF1.2	Gerenciar Nível de Acesso	O sistema deve controlar acessos por perfil: paciente, profissional, secretário e administrador, limitando as informações conforme a função.
RF1.3	Cadastro de usuários	O sistema deve permitir o cadastro de novos usuários com perfil inicial de “aguardando acesso”.
RF1.4	Conceder acessos	O sistema deve permitir a alteração do perfil de acesso por administradores.
Gestão de Atendimentos		
RF2.1	Solicitação de Atendimentos	O sistema deve permitir que pacientes solicitem atendimentos, escolham data e horário, visualizem o status das solicitações.

RF2.2	Agendamento de Consultas	O sistema deve permitir agendar e desmarcar consultas com profissionais, respeitando o horário de funcionamento e evitando conflitos de horário com outros pacientes.
RF2.3	Visualizar quadro de atendimentos	O sistema deve exibir os atendimentos do dia, com horários e presença dos pacientes, permitindo filtro por data inicial e final.
RF2.4	Visualizar histórico de agendamentos	O paciente e os profissionais devem conseguir visualizar todos os agendamentos anteriores e futuros.
RF2.6	Aprovar ou rejeitar solicitações	O sistema deve disponibilizar interface para técnicos/administradores aprovarem ou rejeitarem solicitações de atendimento.
Gestão do Prontuário Eletrônico		
RF3.1	Confirmar Presença do Paciente	O sistema deve permitir confirmar a presença do paciente. Também deve ser possível desmarcar a consulta em caso de ausência.
RF3.2	Gestão da Consulta	O sistema deve permitir que o profissional inicie a consulta, visualizando os dados de consultas anteriores do paciente e inserindo informações estruturadas no prontuário. Apenas usuários com perfil de profissional de saúde podem utilizar essa função.
RF3.3	Inserir anamnese na estrutura SOAP	O sistema deve permitir ao profissional de saúde registrar a consulta utilizando a estrutura SOAP (Subjetivo, Objetivo, Avaliação e Plano).
RF3.4	Emitir atestados, prescrições, solicitação e exame e encaminhamento	O sistema deve possibilitar a emissão de, atestados, prescrições médicas, solicitações

		de exames e encaminhamentos durante a consulta.
RF3.5	Visualizar histórico clínico do paciente	O profissional deve ter acesso às consultas anteriores do paciente para melhor continuidade do cuidado.
Requisitos não funcionais		
RNF1	Criptografia de senhas	As senhas dos usuários devem ser criptografadas utilizando <i>bcrypt</i> com salt rounds.
RNF2	Uso de <i>JWT</i>	A autenticação deve ser feita com tokens <i>JWT</i> válidos para sessões seguras.
RNF3	Controle de acesso por perfil	Cada usuário deve acessar apenas funcionalidades permitidas de acordo com seu perfil.
RNF4	Criptografia de dados Sensíveis	As informações sensíveis das consultas dos usuários devem ser criptografadas utilizando AES-CBC.
RNF5	Linguagem de Desenvolvimento do <i>Backend</i>	O <i>backend</i> será desenvolvido em Node.js.
RNF6	Linguagem de Desenvolvimento do <i>Frontend</i>	O <i>frontend</i> será desenvolvido em Angular.

Fonte: Autoria própria.

3.2 Prototipação do Banco de Dados

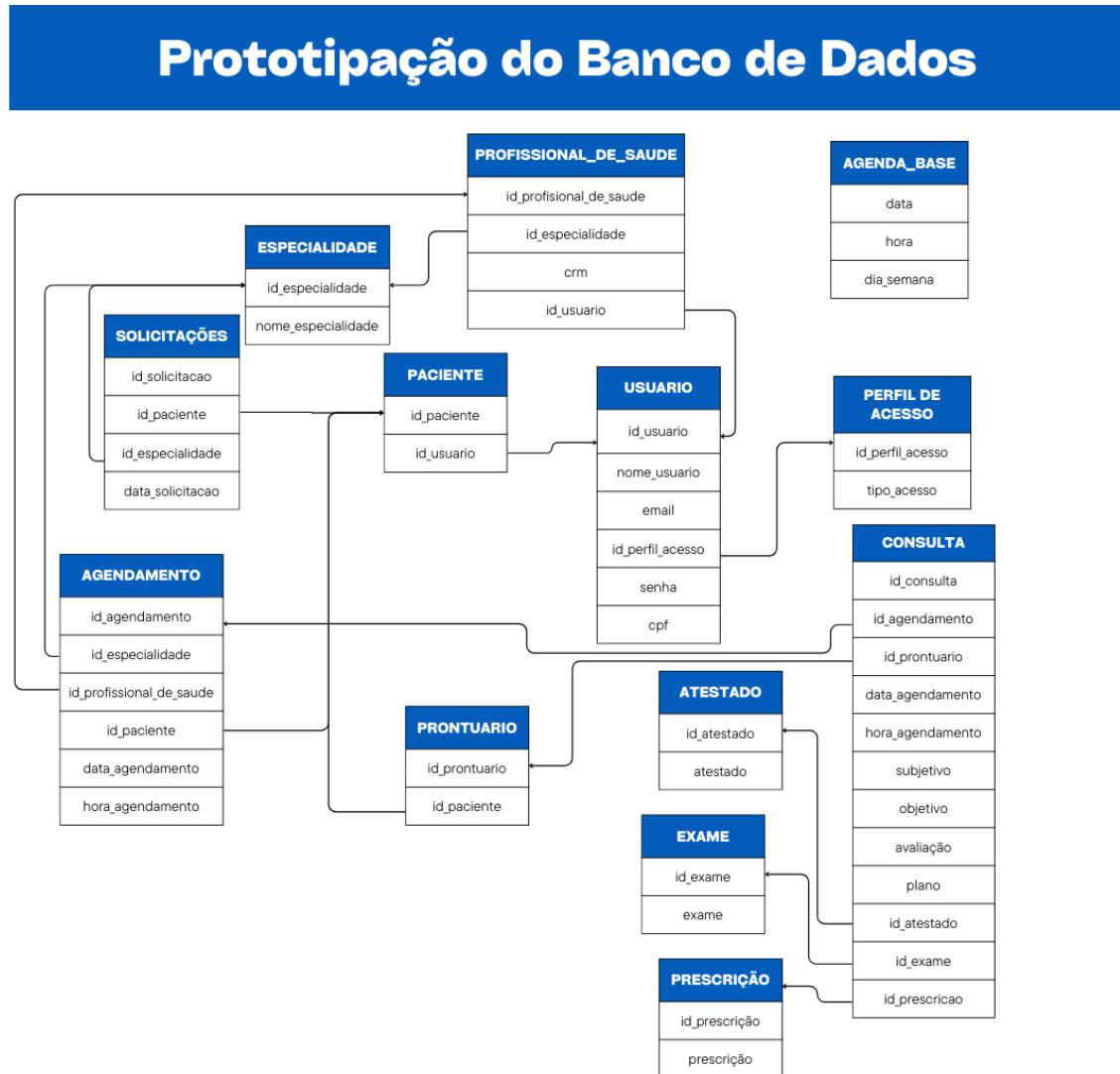
Após a análise de requisitos, foi desenvolvida a estrutura do banco de dados apresentado na Figura 3. Esse Modelo Entidade relacionamento (MER) inclui as tabelas de usuário, perfil de acesso, profissional de saúde, paciente, especialidade, solicitações, agenda, agendamento, consulta, prontuário, atestado, prescrição, exame e encaminhamentos.

- Usuário: armazena os dados de autenticação dos usuários, e será utilizada no procedimento de login, o sistema validará se o usuário está registrado na tabela,

permitindo acesso ao sistema quando o registro for confirmado. Caso o usuário não esteja na tabela o login não será efetivado;

- Perfil de acesso: define os tipos de acesso do sistema, como técnico administrativo, profissional de saúde, administrador e paciente;
- Paciente: registra os usuários com perfil de paciente, vinculando-os à tabela de usuários;
- Profissional de saúde: registra os usuários com perfil de profissional de saúde, além de conter outras informações como o CRM do profissional, e a sua especialidade;
 - Especialidade: contém os tipos de especialidades médicas disponíveis no sistema;
- Solicitações: registra os pedidos de atendimento realizados pelos pacientes, incluindo a identificação do paciente, a especialidade desejada e a data preferencial para o atendimento;
- Agendamento: registra os agendamentos confirmados, com identificação do paciente, profissional, data e horário da consulta;
- Consulta: armazena as informações referentes ao atendimento realizado, incluindo anamnese, SOAP, atestados, prescrições, exames e encaminhamentos;
- Prontuário: reúne todas as consultas registradas no sistema, e possibilita a visualização do histórico clínico do paciente;
- Atestado, prescrição e exames e encaminhamentos: para armazenar documentos;
- Agenda_base: tabela auxiliar que contém os horários padrões para marcação de consultas.

Figura 3: Diagrama entidade Relacionamento do protótipo.

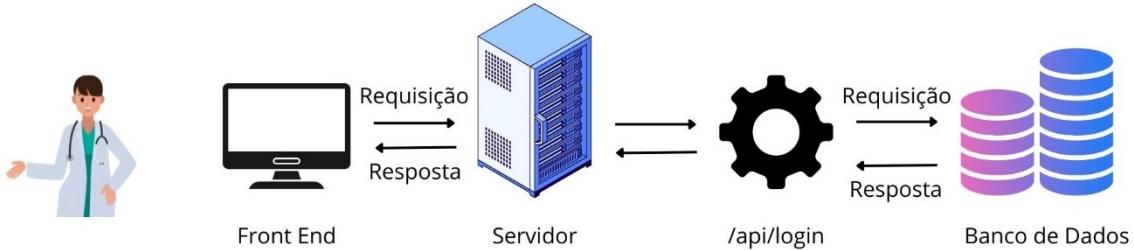


Fonte: Autoria própria.

3.3 Desenvolvimento da aplicação

Uma aplicação Web funciona da seguinte forma: o usuário interage com o *frontend* e faz requisições ao *backend*. Por exemplo, ao digitar o *e-mail* e a senha e clicar em fazer *login* o servidor recebe a requisição de *login*. Dentro do servidor, existe uma rota específica para tratativas de *login* e ela é chamada. Esta rota, por sua vez, realiza uma consulta ao banco de dados para conferir se o usuário está cadastrado. Em caso afirmativo, uma resposta positiva é enviada de volta ao *frontend*, permitindo o acesso ao sistema. Caso contrário, uma resposta negativa é retornada, impedindo o acesso. Esse fluxo pode ser visualizado na Figura 4.

Figura 4: Funcionamento de aplicações fullstack.



Fonte: Autoria própria.

A arquitetura escolhida para a implementação é conhecida como *RESTful*, uma interface que permite a dois sistemas trocar informações de forma segura pela internet. Grande parte das aplicações precisam se comunicar com outras aplicações, internas ou de terceiros, para executar tarefas. Neste trabalho a arquitetura *RESTful* será utilizada para comunicação entre o *frontend* e o *backend* (AWS Amazon, 2025). Essa arquitetura segue princípios fundamentais, listados a seguir:

- Separação cliente e servidor: O *frontend* e o *backend* operam de forma independente;
- Ausência de estado: o servidor completa cada requisição do cliente independente de todas as requisições anteriores;
- Interface Uniforme: geralmente, APIs *RESTful* são implementadas usando *Hipertext Transfer Protocol* (HTTP) para manipulação de recursos facilitando a comunicação entre os sistemas.

3.3.1 Desenvolvimento do *backend*

Para o desenvolvimento do *backend*, foi escolhido o Node.js, juntamente com o Express, um framework que facilita a criação de APIs e a execução do servidor em portas locais da máquina, oferecendo recursos já encapsulados para agilizar a implementação.

3.3.1.1 Criação da API e definição dos *endpoints*

Durante o desenvolvimento do *backend*, é imprescindível a criação de uma API robusta e segura, que funcione como intermediária entre o *frontend* e o banco de dados, garantindo a troca de informações de forma eficiente, organizada e protegida. Para isso, foi escolhido o Express como framework para a criação do servidor. Sua simplicidade no gerenciamento de

rotas, manipulação de requisições e respostas, além da ampla documentação e comunidade ativa tornou-o a opção ideal para o desenvolvimento do projeto.

Como o *frontend* e o *backend* do prontuário serão hospedados separadamente, e é necessário que a aplicação web se comunique com a API, foi escolhida a biblioteca CORS (*Cross-Origin Resource Sharing*). Essa biblioteca permite a comunicação entre diferentes origens, respeitando as boas práticas de segurança.

Outra preocupação durante o desenvolvimento foi a gestão de informações sensíveis, como senhas de criptografia e configurações de conexão com o banco de dados. Para isso, foi utilizada a biblioteca *dotenv*, que permite armazenar essas informações em arquivos *.env*, evitando que dados críticos fiquem expostos no código-fonte.

Como o prontuário precisa lidar constantemente com o envio e recebimento de dados em formato *JSON*, foi adicionada a biblioteca *body-parser*. Essa biblioteca faz com que os dados enviados pelas requisições sejam convertidos em objetos Javascript, o que facilita o tratamento interno.

Para garantir a segurança das credenciais dos usuários, foi adotada a biblioteca *bcrypt*, que permite aplicar *hashing* nas senhas antes de serem armazenadas no banco de dados. Isso assegura que, mesmo em caso de vazamento, os dados não sejam facilmente comprometidos.

Além das senhas, outras informações confidenciais também são protegidas por meio da biblioteca nativa *crypto* do *Node.js*, que oferece métodos robustos para criptografia e descriptografia de dados.

Por fim, a comunicação com o banco de dados *PostgreSQL* foi realizada com a biblioteca *pg*, que fornece os recursos necessários para realizar consultas, inserções e atualizações no banco de forma eficiente e segura.

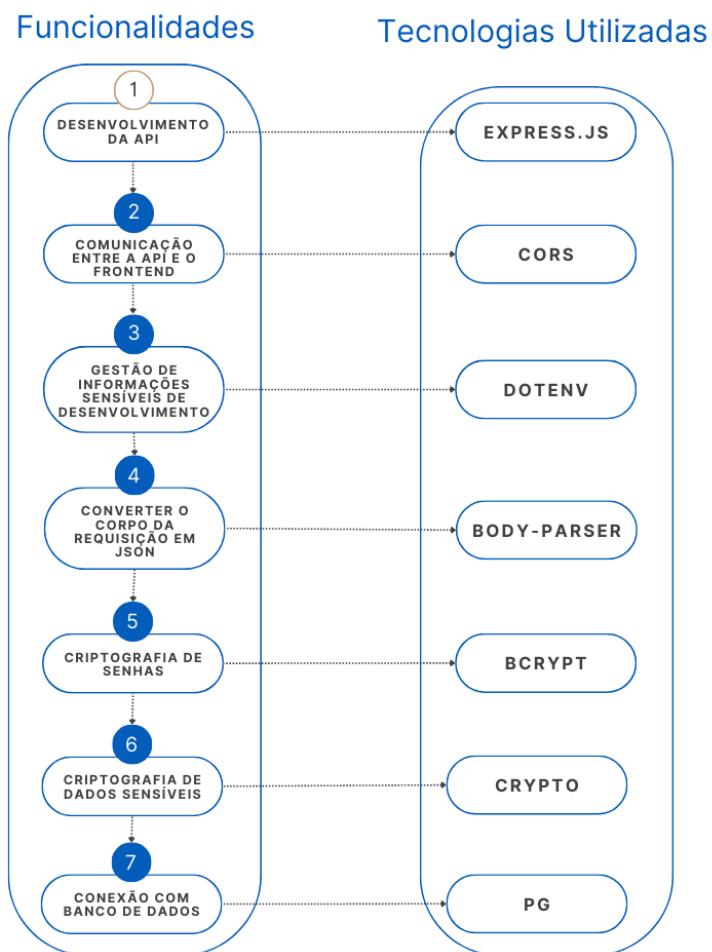
A escolhas de bibliotecas, portanto, foram guiadas tanto pela necessidade técnica do sistema quanto pela preocupação em garantir segurança, escalabilidade e facilidade de manutenção no futuro. Os objetivos e tecnologias utilizadas podem ser visualizadas na Figura 5.

As rotas definidas na API foram:

- */login*: responsável por autenticar o usuário. Ele verifica se o *e-mail* e senha enviados pelo usuário são válidos e, se estiverem corretos, retorna um *token* decesso.
- */agenda*: retorna os agendamentos cadastrados no sistema, usado para visualizar a agenda dos profissionais de saúde.

- /agenda-livre: exibe os horários livres na agenda. Permite ao usuário saber quando há horários vagos para marcar um novo atendimento.
- /desmarcar-consulta: usado para cancelar um agendamento, altera o status da consulta para desmarcada.
- /marcar-consulta: permite agendar uma nova consulta em um horário disponível.
- /confirmar-presença: confirma que o paciente compareceu e está disponível para iniciar o atendimento.
- /agendamentos-pacientes: retorna os agendamentos de um determinado paciente, útil para mostrar aos pacientes seus atendimentos futuros e passados.
- /solicitar-atendimento: permite que o usuário solicite um agendamento no sistema.

Figura 5: Funcionalidades e tecnologias utilizadas na construção da API.



Fonte: Autoria própria.

- /get-solicitacoes: retorna a lista de solicitações de atendimentos de cada paciente, para ele acompanhar qual o status das suas solicitações;
- /get-all-solicitacoes: retorna as solicitações pendentes de todos os pacientes, para que possam ser aceitas ou rejeitadas.
- /rejeitar-solicitacao: rejeita uma solicitação de atendimento feita pelo usuário;
- /aceitar-solicitacao: aceita uma solicitação de atendimento feita pelo usuário;
- /get-atendimento: retorna a lista de pacientes que estão presentes e aguardando o inicio do atendimento
- /iniciar-atendimento: inicia o atendimento, gerando uma nova consulta para o prontuário eletrônico.
- /consultas-prontuario: rota responsável por retornar o histórico de consultas do paciente;
- /salvar-atestado: rota responsável por salvar um novo atestado vinculado à consulta em andamento;
- /salvar-exames: rota responsável por salvar um novo exame vinculado à consulta em andamento;
- /salvar-prescrecao: rota responsável por salvar uma nova prescrição vinculada à consulta em andamento;
- /finalizar-consulta: marca o atendimento como encerrado, impedindo alterações futuras.
- /get-arquivos-consulta: esta rota recupera arquivos relacionados à consulta, atestados, exames e prescrição para o paciente poder visualizar.
- /registrar: cria um novo usuário no sistema que recebe o acesso de aguardando acesso
- /get-users-without-access: lista todos os usuários do sistema que tem acesso de aguardando_acesso;
- /give-access: concede um dos tipos de acesso ao usuário (paciente, técnico administrativo, médico ou administrador). Usuários que receberem o tipo de acesso "médico" ainda precisarão completar o perfil, adicionando especialidade e CRM.

3.3.1.2 Autenticação e Segurança

A segurança no sistema de prontuário eletrônico foi tratada como um requisito inegociável desde o primeiro momento, pois o sistema manipula dados sensíveis, incluindo

senhas e informações médicas. Por isso, foi fundamental garantir que essas informações fossem protegidas da melhor forma possível.

Para fortalecer a proteção das senhas, foi utilizado o *bcrypt*, uma biblioteca amplamente reconhecida pela sua eficiência na criptografia de senhas. O *bcrypt* transforma a senha em um *hash* seguro e implementa um mecanismo de *salting*, que dificulta tentativas de quebra por ataques de força bruta, já que senhas iguais geram *hashes* diferentes.

Além da proteção das senhas, outros dados sensíveis dos usuários, também necessitam de um tratamento especial, como por exemplo as informações clínicas. Para esses casos, foi escolhido o algoritmo de criptografia *AES-256-CBC*, um dos padrões mais seguros e utilizados para proteção de dados. Essa escolha garante que informações críticas sejam armazenadas de forma criptografada e só possam ser descriptografadas por quem possuir as chaves corretas, aumentando ainda mais a segurança geral do sistema.

Essas decisões foram tomadas para assegurar que, independentemente do ponto de acesso ou do tipo de ataque, as informações dos usuários permaneçam protegidas, reforçando a confiabilidade e a robustez do prontuário eletrônico.

3.3.1.2.1 Registro no sistema

O processo de registro de novos usuários no sistema de prontuário eletrônico foi estruturado para garantir segurança, organização e validação eficiente dos dados. Para isso, as senhas dos usuários devem passar por um processo de criptografia, a biblioteca escolhida para essa finalidade foi a *bcrypt*.

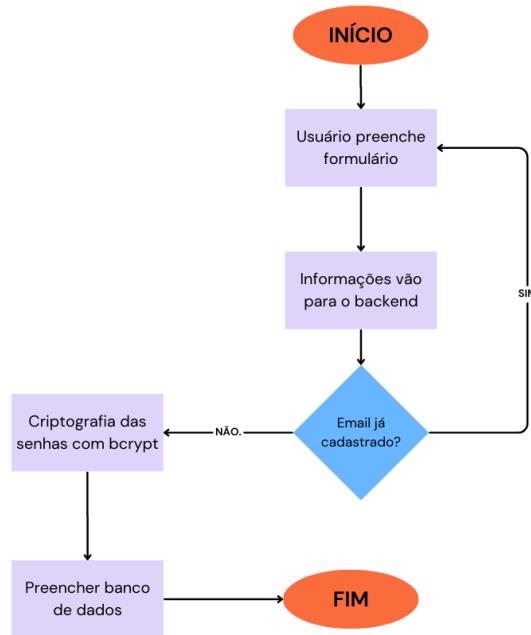
O fluxograma de registro no sistema é representado na figura 6. O fluxo começa com o preenchimento do formulário de cadastro pelo usuário, no qual são solicitadas informações essenciais, como nome, *e-mail*, senha e CPF. Após o envio, esses dados são encaminhados para o *backend* da aplicação, onde se inicia a primeira etapa de validação.

O sistema então realiza uma verificação no banco de dados para identificar se o *e-mail* informado já está cadastrado. Caso o *e-mail* já exista, o processo é imediatamente interrompido, e uma mensagem de erro é retornada ao usuário, informando que aquele *e-mail* não pode ser utilizado. Se o *e-mail* não estiver registrado, o fluxo segue para a próxima etapa.

Em seguida, os dados sensíveis passam por processos de criptografia. A senha é criptografada utilizando o *bcrypt*, o que garante a proteção contra possíveis vazamentos ou acessos não autorizados.

Após a criptografia, as informações tratadas são finalmente inseridas no banco de dados, finalizando o processo de registro de forma segura e confiável. Dessa forma, o sistema assegura não apenas a integridade das informações, mas também a proteção dos dados pessoais de seus usuários.

Figura 6: Fluxograma cadastro de Novo Usuário: Verificação, Criptografia e Armazenamento



Fonte: Autoria própria.

3.3.1.2.2 Login no sistema

O processo de login no sistema de prontuário eletrônico foi projetado para garantir a autenticação segura dos usuários. O fluxograma do login no sistema pode ser visualizado na Figura 7.

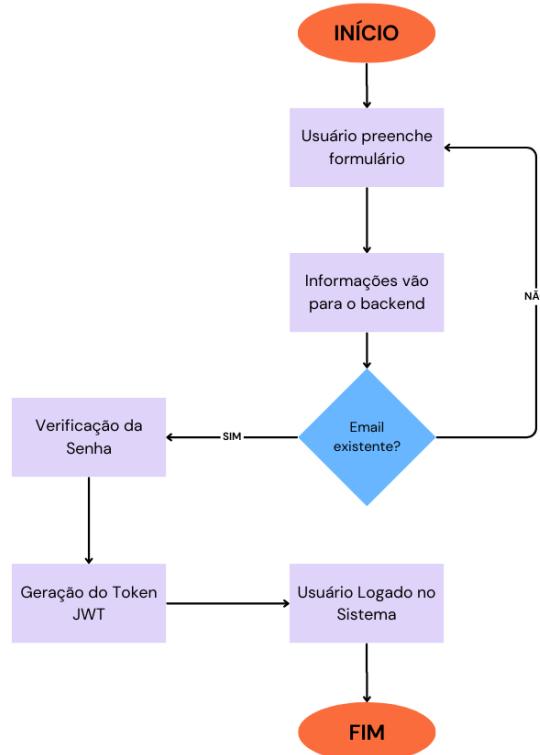
O fluxo se inicia quando o usuário insere seu *e-mail* e senha no formulário de login. Essas informações são enviadas para o *backend*, que realiza a primeira validação: uma consulta ao banco de dados para verificar se o *e-mail* informado está cadastrado na tabela de usuários.

Se o *e-mail* não for encontrado, o sistema retorna ao *frontend* um erro 401 (Não autorizado), acompanhado da mensagem “Usuário não encontrado”, interrompendo o processo de autenticação.

Caso o *e-mail* esteja registrado, o sistema avança para a verificação da senha. Para isso, é utilizada uma função do *bcrypt*, que compara o *hash* da senha armazenada no banco com o *hash* gerado a partir da senha fornecida pelo usuário no momento do login. Se a senha for inválida, o *backend* retorna para o *frontend* uma mensagem de erro indicando “Senha inválida”.

Se a senha estiver correta, o sistema gera um *JSON Web Token (JWT)*, uma solução segura e compacta para a transmissão de informações entre aplicações. O *token* gerado inclui dados relevantes como o nome do usuário, o escopo de acesso, o id e o perfil do usuário. Essas informações são utilizadas no *frontend* para personalizar a experiência do usuário e garantir o controle de acesso às funcionalidades do sistema.

Figura 7: Fluxograma de Login no Sistema com Verificação e Geração de *Token JWT*



Fonte: Autoria própria.

3.3.1.2.3 Middleware de autenticação

Para aumentar a segurança é necessário proteger as rotas da API. Isso é feito verificando se o usuário que está tentando acessar determinada rota possui um *Token JWT* válido, permitindo que apenas usuários autenticados tenham acesso a recursos sensíveis do sistema, para isso foi utilizado um *middleware* de autenticação, o fluxograma do *middleware* de autenticação no sistema é visualizado na Figura 8.

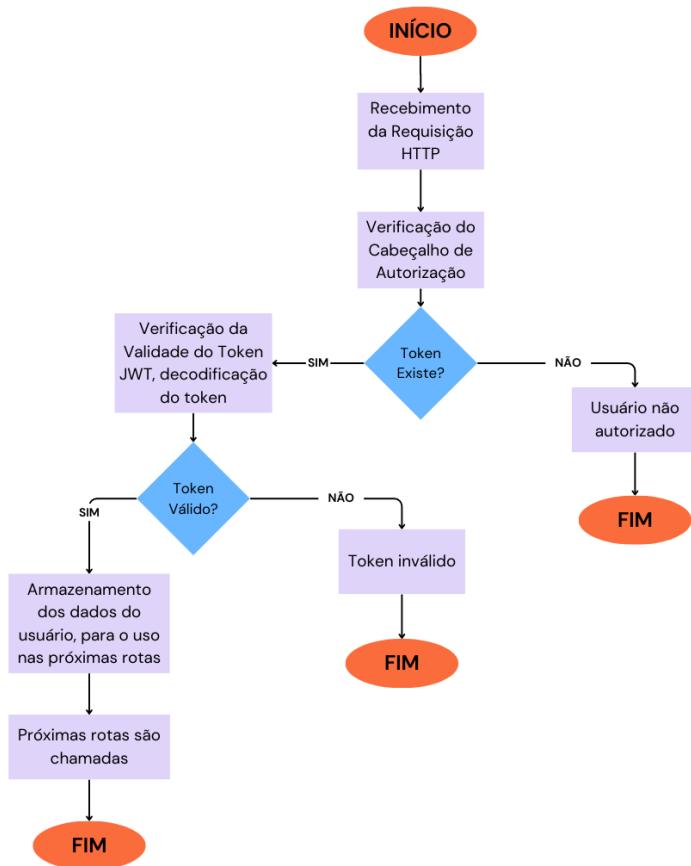
O fluxo começa quando o servidor recebe uma requisição HTTP para uma rota protegida. Esse *middleware* é executado antes de a requisição chegar ao manipulador final. Ele tem como objetivo verificar se há um *token* de autenticação válido na requisição. A primeira etapa do *middleware* é acessar o cabeçalho da requisição e extrair o *token JWT*.

Após isso, o próximo passo é verificar se o *token* foi fornecido. Caso não haja *token*, por exemplo, se o cabeçalho estiver ausente ou malformado, a função retorna imediatamente um status 401 (Não Autorizado), encerrando o fluxo para essa requisição. Isso impede o acesso de usuários não autenticados. Se o *token* existir, o *middleware* verifica se o *token* é válido e foi assinado com a chave secreta correta (SECRET_KEY). Esse processo também decodifica o conteúdo do *token*. Se a verificação falhar (por exemplo, *token* expirado ou inválido), o *middleware* retorna o status 403 (Proibido), indicando que o usuário não tem permissão para acessar, mesmo que tenha fornecido algo.

Se o *token* for válido, o conteúdo decodificado do *token*, dados do usuário, como id, tipo de acesso, é armazenado num objeto. Isso permite que os próximos *middlewares* ou controladores tenham acesso a essas informações para autorizações específicas. Após isso o fluxo segue para o próximo *middleware* ou rota.

Existem rotas protegidas que apenas usuários com perfil de médico, super usuário ou administrador podem acessar. Portanto, após o *middleware* de autenticação validar o *token* e permitir o avanço da requisição, a rota seguinte realiza uma verificação adicional, confirmando se o usuário possui o nível de acesso exigido, o que garante que apenas os perfis apropriados tenham permissão para prosseguir, aumentando a segurança e o controle de acesso do sistema.

Figura 8: Fluxograma de Processamento de *Token* no *Middleware* de Autenticação



Fonte: Autoria própria.

3.3.1.2.4 Criptografia de Dados Sensíveis

Para o protótipo de prontuário eletrônico, é essencial que as informações sensíveis dos pacientes como: diagnósticos, tratamentos e dados pessoais, sejam protegidos por uma criptografia robusta, garantindo a confidencialidade dos dados, mesmo em caso de vazamentos.

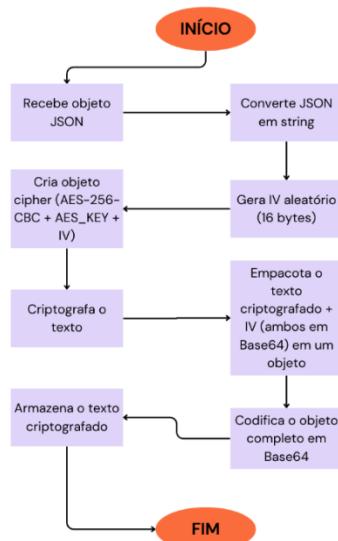
Para isso, foi utilizado o módulo *crypto* do Node.js, que implementa o algoritmo AES-256 no modo CBC. Esse algoritmo é amplamente reconhecido por sua robustez e segurança em aplicações críticas que lidam com dados sensíveis.

O fluxo de criptografia começa recebendo um objeto *JSON*, que primeiramente é convertido em uma string para que seja realizado o processo de criptografia. Em seguida, um *IV* (vetor de inicialização) de 16 *bytes* é gerado aleatoriamente. Com o *IV* e a chave secreta (*AES_KEY*), é criado o objeto *cipher*, responsável pela codificação. O texto é então criptografado e convertido para a codificação *base64*. Tanto o texto criptografado quanto o *IV* — também codificado em *base64* — são reunidos em um objeto, que é convertido em *base64*. Esse processo final facilita o armazenamento e o envio seguro das informações através da rede, pois evita que caracteres especiais sejam mal interpretados numa requisição HTTP.

Já o fluxo de descriptografia reverte o processo. Primeiramente, o texto criptografado em *base64* é decodificado, separando o *IV* e os dados criptografados. Utilizando o mesmo algoritmo **AES-256-CBC**, a mesma chave secreta e o *IV* original, é criado um objeto *decipher*. O texto criptografado é então decodificado de *base64* para sua forma original em **UTF-8**, recuperando o conteúdo inicial de maneira segura.

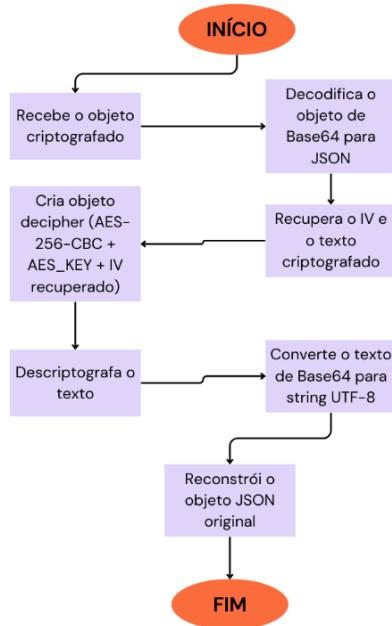
Essa simetria entre os processos de criptografia e descriptografia assegura que apenas quem possuir a chave secreta e o *IV* correspondente poderá acessar as informações confidenciais dos pacientes, mantendo a integridade e a privacidade dos dados armazenados no sistema.

Figura 9: Fluxograma de criptografia AES-CBC.



Fonte: Autoria própria.

Figura 10: Fluxograma de descriptografia AES-CBC



Fonte: Autoria própria.

3.3.1.3 Conexão com banco de dados

Para que a aplicação funcione com o banco de dados é necessário fazer a conexão com o servidor ao banco de dados. Para se conectar ao *PostgreSQL* é necessário, importar a biblioteca *pg* e criar um novo *client* com as informações *user*, *host*, nome do banco de dados, senha do banco de dados e a porta na qual o banco de dados está rodando. Todas essas variáveis são guardadas no *dotenv* para se assegurar que ao carregar este o projeto para um repositório online essas informações não vazem, evitando possíveis falhas de segurança no acesso ao banco de dados. O banco de dados é conectado e é exportado como *db*, permitindo que ele seja reutilizado em diferentes partes do projeto. A conexão com o banco de dados pode ser visualizada na figura 11.

Figura 11: Conexão com banco de dados

```
backend > config > JS database.js > ...
1 import pg from "pg";
2 import dotenv from "dotenv";
3
4 dotenv.config();
5
6 const db = new pg.Client({
7   user: process.env.DB_USER,
8   host: process.env.DB_HOST,
9   database: process.env.DB_NAME,
10  password: process.env.DB_PASSWORD,
11  port: process.env.DB_PORT,
12 });
13
14 db.connect();
15
16 export default db;
17
```

Fonte: Autoria própria.

3.3.2 Desenvolvimento do *frontend*

O angular foi escolhido para o desenvolvimento do *frontend* por ser um framework baseado em componentes, o que permite dividir a aplicação em partes reutilizáveis e organizadas. Cada componente representa uma seção da interface e pode ser reutilizado em diferentes partes do projeto. Isso diminui a repetição de código HTML e CSS, pois o componente pode ser importado onde for necessário, o que auxilia na maior produtividade do desenvolvimento e na manutenção do projeto.

Além disso, o Angular possui o *Angular Material*, uma biblioteca oficial de componentes estilizados. Essa biblioteca oferece uma variedade de elementos prontos, como botões, tabelas, campos de entrada, toolbars e muito mais, garantindo uma interface moderna, responsiva e com boa usabilidade. No desenvolvimento deste projeto, o *Angular Material* foi utilizado para compor os componentes da aplicação.

Na figura 12, é possível visualizar como o Angular organiza a interface por meio de componentes. O retângulo preto representa o componente principal de calendário, responsável por importar e estruturar todos os outros componentes da página. Em roxo a barra de toolbar, que serve como cabeçalho do sistema e ponto de navegação. O retângulo azul destaca o componente de filtro de agendamento, permitindo que o usuário refine a busca por profissional, especialidade e período. Em verde, observa-se o componente de listagem dos agendamentos, que exibe as consultas em formato de tabela. Por fim, em vermelho, está o componente de detalhes da consulta, que apresenta informações específicas da consulta selecionada e disponibiliza ações como confirmar presença ou desmarcar. Essa estrutura modular exemplifica

bem a reutilização e organização que tornam o Angular tão eficiente no desenvolvimento de interfaces ricas e dinâmicas.

Figura 12: Organização da interface com Framework Angular.

Horário	Data	Profissional	Especialidade	Paciente
08:00	31/03/2025	Felipe Costa	Cardiologista	Fernanda Costa
08:30	31/03/2025	Laura Martins	Cardiologista	Fernanda Costa
08:30	31/03/2025	Felipe Costa	Cardiologista	Lucas Oliveira
09:00	31/03/2025	Felipe Costa	Cardiologista	Ana Souza

Fonte: Autoria própria.

No *frontend*, foi implementada uma arquitetura de rotas protegidas de acordo com os diferentes níveis de acesso dos usuários. Essa estrutura garante que cada usuário só consiga visualizar e interagir com as áreas que estão dentro do seu escopo de permissão: médico, paciente, administrador, técnico administrativo e super usuário. O controle de acesso é realizado por meio de um *token JWT (JSON Web Token)*, citado anteriormente, que é decodificado no lado do cliente para identificar o tipo de usuário e, assim, permitir ou restringir o acesso às rotas. Assim, essa abordagem aumenta a segurança da aplicação, evitando acessos indevidos a funcionalidades sensíveis.

A conexão do Angular com o *backend* é feita por meio do *HttpClient*, um serviço fornecido pelo módulo `@angular/common/http`. Esse serviço permite que a aplicação envie requisições HTTP para servidores externos, como APIs REST, e receba as respostas dessas requisições.

3.4 Testes da Prova de Execução do Projeto (PEP)

O teste de prova da execução do PEP teve como objetivo verificar o funcionamento integrado entre *frontend* e *backend*, certificando que os principais fluxos da aplicação estivessem funcionando como esperado. Durante o teste, foram simuladas ações reais dos

usuários, como registro de novos usuários, login com diferentes perfis de acesso, criação e visualização de agendamentos, início das consultas e preenchimento de dados, geração de documentos, visualização de consultas passadas, visualização de documentos gerados nas consultas.

4 RESULTADOS E DISCUSSÃO

Como resultado deste trabalho, foi desenvolvido um sistema protótipo de PEP que atende aos principais objetivos da DIRQS. O sistema contempla funcionalidades essenciais, como o módulo completo de agendamentos, controle de acesso por tipo de usuário e uma interface intuitiva que facilita a visualização de informações de consultas passadas para os profissionais de saúde. A aplicação foi construída com foco na segurança, usabilidade e organização da informação clínica, proporcionando uma experiência eficiente para todos os usuários.

O sistema proposto apresenta características estruturais similares ao Prontuário Eletrônico do Cidadão (PEC), adotado amplamente no Sistema Único de Saúde (SUS). Assim como o PEC, ele adota o modelo SOAP para organização das informações clínicas, o que garante uma padronização no registro dos dados e favorece a continuidade do cuidado. Além disso, incorpora a funcionalidade de solicitação de atendimentos diretamente pelo paciente, o que representa uma tentativa de mitigar o problema recorrente do absenteísmo no SUS — um desafio amplamente documentado na literatura (POSTAL et al., 2021).

No entanto, diferentemente do PEC, que possui módulos avançados como acesso à RNDS, histórico completo de atendimentos e gerenciamento integrado de vacinação, o sistema proposto ainda se limita a funcionalidades básicas, focando em acessibilidade e agilidade no agendamento. Embora isso torne o sistema mais leve e de fácil implementação, também o restringe quanto à interoperabilidade e profundidade clínica.

Em comparação com os prontuários eletrônicos de código aberto analisados por Purkayastha et al. (2019) — como OpenEMR, OpenEHR e GNU Health — o sistema proposto atende parcialmente aos requisitos funcionais observados como essenciais em sistemas consolidados internacionalmente. Destacam-se positivamente o suporte ao processo administrativo e a autenticação de usuários, mas ainda estão ausentes recursos como o gerenciamento de resultados de exames e suporte à decisão clínica. Tais funcionalidades, embora não implementadas na versão atual, podem ser consideradas para futuras expansões, o que aumentaria significativamente a aplicabilidade e a segurança clínica do sistema.

Quanto à escalabilidade, o sistema foi projetado com uma arquitetura modular, o que permite sua expansão gradual, adaptando-se às necessidades de diferentes unidades de saúde. A interoperabilidade, por outro lado, ainda é um ponto em desenvolvimento: embora a integração com bancos de dados nacionais, como a RNDS, ainda não tenha sido implementada, o fato de o sistema já estar estruturado com base no modelo SOAP representa um passo

importante nessa direção. Esse modelo favorece a padronização das informações clínicas e permite, tecnicamente, a futura integração com repositórios nacionais, desde que haja adequação aos padrões de segurança, autenticação e compatibilidade exigidos pelas plataformas de dados do SUS.

Já no quesito aceitação pelos usuários, espera-se maior adesão devido à interface simplificada e à autonomia proporcionada ao paciente no processo de solicitação de atendimentos — o que responde diretamente aos problemas de acesso descritos por TURKCAN et al. (2011).

Portanto, embora o sistema ainda careça de funcionalidades mais robustas presentes em prontuários mais consolidados, ele representa um avanço significativo no acesso e gestão de atendimentos em unidades de saúde com infraestrutura limitada, podendo servir como base para futuras implementações mais complexas.

Para a demonstração, serão adicionados 4 usuários, João Pedro, Bruno, Luísa e Pedro um administrador, um médico, um técnico e um paciente e serão seguidos os fluxos de uso habituais do sistema.

4.1.1 Registrar

Para se registrar no sistema, o usuário deve inserir o nome, e-mail, CPF e a senha. A senha deve conter no mínimo 8 caracteres, incluindo ao menos uma letra maiúscula, uma letra minúscula, um número e caracteres especiais. Como exemplo, a adição dos usuários está ilustrada na figura 13. Ao clicar em "Registrar-se", as informações são enviadas para o backend e salvas no banco de dados na tabela de usuários, como pode ser visto na figura 14, onde as informações dos usuários adicionados são visualizadas.

Figura 13: Tela de registro do sistema, necessário inserir nome, *e-mail*, cpf e senha.

Nome
Bruno William

Email
bruno@gmail.com

Cpf
000000000000

Senha
Sa35(trfvj)S

- Mínimo 8 caracteres
- Pelo menos uma letra maiúscula
- Pelo menos uma letra minúscula
- Pelo menos um número
- Pelo menos um caractere especial

Registrar

Voltar

Fonte: Autoria própria.

Figura 14: Tabela de usuários no banco de dados, preenchida com quatro usuários aguardando acesso.

ID_USUARIO	NOME_USUARIO	EMAIL	ID_PERFIL_ACESSO	SENHA	CPF	PROFILE_COMPLETED
2	Bruno William	bruno@gmail.com	5	\$2b\$10\$o568HiUR6j6/YpxxW/WdoOkr/T.chZk.nJZabl0ItGkK4J PINNahtm	000000000000	false
3	João Pedro	jpedro@gmail.com	5	\$2b\$10\$temtrbaM1A4wzCCmp tWJ6.8kI/mXaQ.U9SJ.nl17WIb TxJfnU3AC	111111111111	false
4	Luísa	luisa@gmail.com	5	\$2b\$10\$KEf1t8tUxZSgkNBfccM zleRlecqsq4k2OM2IyTkkf8qg1.I GZziHa	222222222222	false
5	Pedro	pedro@gmail.com	5	\$2b\$10\$YJ/ASx5t9.LsGs0ON4 0YDe1SiSEg5VhLhfqXRox95Z69 aYjTq//y.	333333333333	false

Fonte: Autoria própria.

É possível verificar que todos os usuários cadastrados têm o mesmo *id_tipo_acesso* até o dado momento (5), mas o que isso significa? O *Id_perfil_acesso* faz referência à tabela de tipos de acesso, que pode ser visualizada na figura 15. De acordo com a figura o tipo de acesso que corresponde ao 5 é o *aguardando_acesso*. Portanto nenhum usuário contém acesso ao sistema ainda, e se tentarem fazer login no sistema, aparecerá a tela que indicará que ele ainda não contém acesso, visualizada na figura 16.

Figura 15: Tabela de tipo de acesso no banco de dados, contendo todos os tipos de acesso.

ID_PERFIL_ACESSO	TIPO_ACESSO
1	admin
2	paciente
3	tecnico
4	medico
5	aguardando_acesso
6	super_user

Fonte: Autoria própria.

Figura 16: Mensagem que os usuários sem acesso veem ao tentar fazer login no sistema
Aguardando acesso, espere que o administrador te aceite no sistema

Fonte: Autoria própria.

4.2 Super usuário

O super usuário é cadastrado pelo desenvolvedor no banco de dados e este usuário tem acesso a todas as funcionalidades inclusive pode adicionar novos administradores. O super usuário pode adicionar novos administradores na página de gerenciamento de acessos, que pode ser visualizada na figura 17.

Durante o teste o usuário João Pedro será definido como administrador, após o super usuário dar a João o acesso, o banco de dados será atualizado e o tipo de acesso de João será atualizado para 1, como mostrado na figura 18.

4.3 Perfil de Administrador

O acesso de administrador tem a responsabilidade de gerenciar as permissões dos outros usuários, sua importância está diretamente ligada à segurança dos dados e da informação, pois garante que cada usuário tenha o nível de acesso adequado à sua função. Ele pode acessar todos os recursos do perfil administrativo, fica restrita a ele apenas a tela de gerenciamento de acesso, que é ilustrada na figura 17.

Figura 17: Tela de gerenciar acessos, onde o administrador e o super usuário podem dar diferentes tipos de acesso a outros usuários

Nome do Usuário	Email do Usuário	Tipo de acesso
João Pedro	jpedro@gmail.com	Admin
Bruno William	bruno@gmail.com	Paciente

Items per page: 5 | < < > >>

Dar Acesso

Fonte: Autoria própria.

Figura 18: Tabela de usuários no banco de dados, preenchida com três usuários aguardando acesso e um administrador.

ID_USUARIO	NOME_USUARIO	EMAIL	ID_PERFIL_ACESSO	SENHA	CPF	PROFILE_COMPLETED
2	Bruno William	bruno@gmail.com	5	\$2b\$10\$o568HiUR6j6/YpxxWWdoOkr/T.chZk.nZabl0ItGkK4JPINahtm	000000000000	false
3	João Pedro	jpedro@gmail.com	1	\$2b\$10\$temtrbaM1A4wzCCmp twJ6.8kI/mXaQ.U9SJ.nbl17WlbtXJfnU3AC	111111111111	true
4	Luísa	luisa@gmail.com	5	\$2b\$10\$KEf1t8tUxZSgkNBfccM zleRlecqsq4k20M2IyTkkf8qg1.IGZziHa	222222222222	false
5	Pedro	pedro@gmail.com	5	\$2b\$10\$YJ/ASx5t9.LsGs0ON40YDe1SiSEg5VhLhfqXRox95Z69	333333333333	false

Fonte: Autoria própria.

4.4 Autenticação

4.4.1 Gerenciamento de acesso

O usuário que se registrar no sistema deve ser aceito pelo administrador, para garantir que apenas usuários aceitos possam acessar o sistema. O administrador João deu acesso de paciente a Bruno, de médico a Luísa e de técnico a Pedro. Agora, o quadro de usuários sofre novamente uma alteração, com Bruno tendo o tipo de acesso 2, Pedro 3 e Luísa 4. O administrador utiliza a mesma tela mostrada na figura 17 para realizar essas ações. Quando

pacientes e médicos são adicionados, eles também são registrados nas tabelas de pacientes e profissionais de saúde.

Figura 19: Tabela de usuários no banco de dados, preenchida com usuários de todos os tipos de acesso.

ID_USUARIO	NOME_USUARIO	EMAIL	ID_PERFIL_ACESSO	SENHA	CPF	PROFILE_COMPLETED
2	Bruno William	bruno@gmail.com	2	\$2b\$10\$o568HiUR6j6/YpxxW/WdoOkr/T.chZk.nJZabl0ItGkK4J PINahtm	000000000000	true
3	João Pedro	jpedro@gmail.com	1	\$2b\$10\$temtrbaM1A4wzCCmp tWJ6.8kI/mXaQ.U9SJ.nbl17WIb TxJfnU3AC	111111111111	true
4	Luísa	luisa@gmail.com	4	\$2b\$10\$KEf1t8tUxZSgkNBfccM zleRlecqsq4k2OM2IyTkkf8qg1.I GZziHa	222222222222	false
5	Pedro	pedro@gmail.com	3	\$2b\$10\$YJ/ASx5t9.LsGsOON4 OYDe1SiSEg5VhLhfqXRox95Z69 aYjTq//y.	333333333333	true

Fonte: Autoria própria.

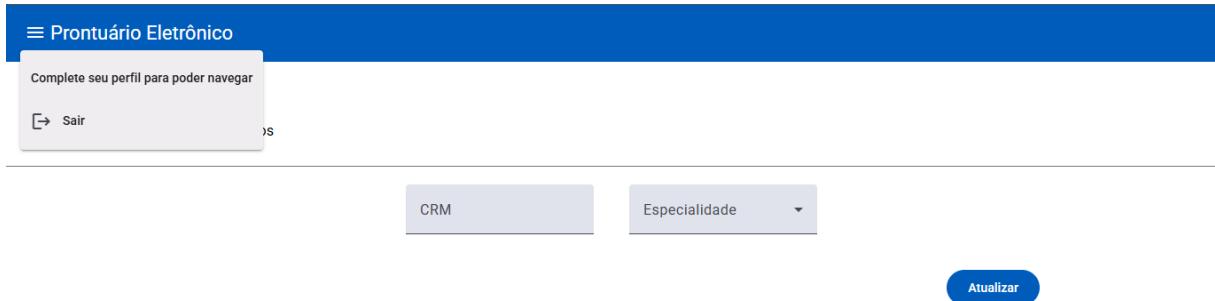
4.4.2 Login

O usuário deve fazer login no sistema informando o seu e-mail e a sua senha, como pode ser visualizado na figura 20. Observando a figura 19, todos os usuários têm o 'profile_completed' como verdadeiro, exceto Luísa. Isso ocorre porque, ao realizar o login, o usuário com perfil de médico deve completar o seu perfil, adicionando o seu CRM e a sua especialidade. Enquanto esse passo não for concluído, o médico não terá acesso às outras funcionalidades. A tela de finalização de cadastro pode ser visualizada na figura 21.

Figura 20: Tela de Login de acesso.

Fonte: Autoria própria.

Figura 21: Tela de finalização de Cadastro para Médicos.



Fonte: Autoria própria.

4.5 Perfil de Técnico

O acesso administrativo no PEP permite a atuação estratégica e operacional sobre o funcionamento da unidade de saúde, com ponto central na gestão de agendas, que envolve a marcação e remarcação de consultas. Outro aspecto fundamental é a gestão das solicitações. Essa organização tem o objetivo de melhorar a experiência dos usuários e fortalecer os laços com os pacientes.

4.5.1 Marcar novas consultas

Na seção de agendar nova consulta, o usuário pode marcar uma nova consulta, inserindo o CPF do paciente, o CRM do profissional de saúde e a data em que deve ser feito o agendamento. Na tabela, aparecem os horários em que o médico e o paciente estão disponíveis. À direita da tela, aparecem as informações do novo agendamento e o botão de marcar consulta. A interface pode ser visualizada na figura 22.

Figura 22: Tela de agendar nova consulta - acesso de técnico administrativo.

Data	Horário	Profissional
01/05/2025	08:00	Luisa
01/05/2025	08:30	Luisa
01/05/2025	09:00	Luisa
01/05/2025	09:30	Luisa

Fonte: Autoria própria.

Ao selecionar o paciente, o profissional de saúde, o dia e o horário, e clicar em 'marcar consulta', essas informações são enviadas para o backend, e a tabela de agendamento é preenchida com esses dados. Na figura 23, é possível visualizar um exemplo do banco de dados de agendamentos.

Figura 23: Tabela de agendamento no banco de dados, preenchida com três agendamentos de exemplo.

ID_AGENDAMENTO	ID_PROFISSIONAL_DE_SAÚDE	ID_PACIENTE	DATA_AGENDAMENTO	HORA_AGENDAMENTO	STATUS
1	1	1	2025-05-01	08:00:00-03:00	agendado
2	1	1	2025-06-02	08:00:00-03:00	agendado
3	1	1	2025-07-01	08:00:00-03:00	agendado

Fonte: Autoria própria.

4.5.2 Acompanhamento das consultas

Na seção de calendário, o usuário com perfil administrativo pode visualizar e gerenciar os atendimentos agendados. Um filtro é disponibilizado para que o usuário possa refinar suas buscas pelas consultas marcadas, podendo filtrar por nome do profissional de saúde, especialidade e intervalo de datas. No centro da tela, há uma tabela com a lista de consultas agendadas, organizadas por horário, data, profissional, especialidade e nome do paciente. À direita da tela, quando o agendamento é selecionado, são exibidos detalhes da consulta, e é

possível desmarcar a consulta ou confirmar a presença do paciente. A interface pode ser visualizada na figura 24.

Figura 24: Tela de calendário, acesso de técnico administrativo.

Fonte: Autoria própria.

Há a possibilidade de desmarcar a consulta ou confirmar a presença do paciente, para que o profissional de saúde prossiga com a consulta. Ao desmarcar a consulta, o status é atualizado para 'desmarcado' na tabela de agendamentos do banco de dados. Ao clicar em 'confirmar presença', o status é atualizado para 'paciente_presente'. Um exemplo disso pode ser visualizado na figura 25.

Figura 25: Tabela de agendamento no banco de dados, preenchida com três agendamentos com status: paciente-presente, desmarcado e agendado

ID_AGENDAMENTO	ID_PROFISSIONAL_DE_SAÚDE	ID_PACIENTE	DATA_AGENDAMENTO	HORA_AGENDAMENTO	STATUS
1	1	1	2025-05-01	08:00:00-03:00	paciente-presente
2	1	1	2025-06-02	08:00:00-03:00	desmarcado
3	1	1	2025-07-01	08:00:00-03:00	agendado

Fonte: Autoria própria.

4.5.3 Acompanhamento das solicitações.

Na seção de solicitações, o usuário pode acompanhar os pedidos realizados. No centro da tela, há uma tabela com as solicitações que ainda estão em análise, organizadas da mais antiga

para a mais recente. À direita da tela, são exibidos os detalhes da solicitação, e o usuário pode aceitá-la ou rejeitá-la, além de inserir uma resposta ao paciente.

No exemplo demonstrado na Figura 26, o paciente solicitou um agendamento com um profissional cardiologista para o dia 01/05/2025. Cabe ao técnico verificar se há algum profissional disponível nessa data e aceitar ou não a solicitação do paciente. Em caso afirmativo, o técnico deve realizar o agendamento na seção 'Agendar nova consulta'.

Figura 26: Tela de acompanhamento de solicitações.

The screenshot shows a digital interface for managing patient requests. At the top, a blue header bar displays the text 'Prontuário Eletrônico'. Below it, a section titled 'Solicitações' (Requests) is shown. A message 'Seja bem vindo! Aqui você acompanha as Solicitações' (Welcome! Here you can track the requests) is displayed. The main content area contains a table with one row of data:

Data	Especialidade	Status
01/05/2025	Cardiologista	análise

Below the table, there are navigation controls: 'Items per page:' dropdown set to 5, and pagination links '1 - 1 of 1' and '< > >>'.

To the right of the table, a vertical sidebar labeled 'Detalhes da Solicitação' (Request Details) provides specific information about the request:

- Paciente : Bruno William
- Especialidade : Cardiologista
- Data : 01/05/2025
- Status : análise
- Resposta

At the bottom of the sidebar are two buttons: 'Rejeitar' (Reject) and 'Aceitar' (Accept).

Fonte: Autoria própria.

4.6 Perfil de Paciente

O acesso do paciente ao prontuário é essencial para garantir a autonomia sobre a sua própria saúde. Por meio desse acesso, o paciente pode acompanhar prescrições, atestados, encaminhamentos, solicitações de exames e o histórico de atendimentos, estimulando o autocuidado. Além disso, ele pode realizar solicitações de exames, o que fortalece o relacionamento com a equipe médica e engaja o paciente no processo de cuidado.

4.6.1 Página do paciente

Na página do paciente, ele pode acompanhar suas consultas. As que já ocorreram estão com o status 'finalizado'; ao clicar nesse item da tabela, o paciente tem acesso aos documentos gerados durante a consulta. Por outro lado, se o status for 'agendado', o usuário pode visualizar as informações referentes à consulta.

No exemplo utilizado, o paciente pode visualizar suas próximas consultas marcadas, como pode ser visto na figura 27. Caso o status da consulta seja 'finalizado' e tenham sido gerados documentos durante a consulta, estes poderão ser baixados pelo paciente na seção lateral à direita, como mostrado na figura 28.

Figura 27: Página do paciente, detalhes da consulta com status agendado.

The screenshot shows a patient dashboard titled "Página do paciente". At the top, there are filters for "Profissional da Saúde" (Luisa), "Especialidade" (Cardiologista), "Enter a date range" (May 1, 2025), and "Status" (agendado). A "Filtrar" button is present. To the right, a sidebar titled "Detalhes da consulta" displays the appointment details: Paciente (Bruno William), Profissional da Saúde (Luisa), Especialidade (Cardiologista), Data (May 1, 2025), and Horário (08:00).

Horário	Data	Profissional	Especialidade	Paciente	status
08:00	01/05/2025	Luisa	Cardiologista	Bruno William	agendado
08:00	02/06/2025	Luisa	Cardiologista	Bruno William	agendado
08:00	01/07/2025	Luisa	Cardiologista	Bruno William	agendado

Fonte: Autoria própria.

Figura 28: Página do paciente, detalhes da consulta com status finalizado.

The screenshot shows a patient dashboard titled "Página do paciente". The interface is similar to Figure 27, with filters for "Profissional da Saúde" (Luisa), "Especialidade" (Cardiologista), "Enter a date range" (May 1, 2025), and "Status" (finalizado). A "Filtrar" button is present. To the right, a sidebar titled "Detalhes da consulta finalizada" displays the appointment details: Paciente (Bruno William), Profissional da Saúde (Luisa), Especialidade (Cardiologista), Data (May 1, 2025), and Horário (08:00). Below the sidebar, a "Documentos disponíveis" section includes links to download "Atestado", "Encaminhamento", and "Prescrição".

Horário	Data	Profissional	Especialidade	Paciente	status
08:00	01/05/2025	Luisa	Cardiologista	Bruno William	finalizado
08:00	01/07/2025	Luisa	Cardiologista	Bruno William	agendado

Fonte: Autoria própria.

4.6.2 Solicitações

Na seção de solicitações, o paciente pode realizar pedidos de consultas, escolhendo a especialidade e a data de interesse, além de acompanhar as solicitações já feitas. As solicitações possuem os seguintes status: 'análise', 'aceita' e 'rejeitada'. No exemplo exibido na imagem 29, o paciente solicitou uma consulta com um cardiologista para o dia 01/05/2025, e é possível verificar que a solicitação foi aceita. O banco de dados das solicitações pode ser visualizado na figura 30. Em casos de recusa, o status será 'rejeitada', e enquanto não houver resposta, o status será 'em análise'.

Figura 29: Seção de solicitações onde o paciente pode solicitar consultas.

The screenshot shows a blue header bar with the text '≡ Prontuário Eletrônico'. Below it, a white page has a dark blue header with the title 'Solicitações' and a subtitle 'Aqui você pode solicitar consultas e acompanhá-las.' A search bar at the top right contains 'Enter a date range' and a date picker icon. To the right is a blue button labeled 'Solicitar'. Below the search bar is a table with three columns: 'Data', 'Especialidade', and 'Status'. The table contains one row of data: '01/05/2025', 'Cardiologista', and 'aceito'. At the bottom of the table are pagination controls: 'Items per page: 5', '1 - 1 of 1', and navigation icons.

Data	Especialidade	Status
01/05/2025	Cardiologista	aceito

Fonte: Autoria própria.

Figura 30: Tabela de solicitações no banco de dados, preenchida com uma solicitação.

ID_SOLICITAÇÃO	ID_PACIENTE	ID_ESPECIALIDADE	DATA_SOLICITAÇÃO	STATUS
1	1	1	2025-05-01	aceito

Fonte: Autoria própria.

4.7 Perfil de Profissional da Saúde

O profissional de saúde é o único tipo de usuário com acesso completo e detalhado ao prontuário, o que possibilita a realização de atendimentos seguros, eficazes e personalizados. Com informações atualizadas sobre o histórico clínico e as medicações em uso, é possível tomar decisões mais assertivas, evitar erros médicos e garantir a continuidade do cuidado.

4.7.1 Iniciar Atendimento

Na seção de calendário, o profissional de saúde acompanha a tabela com os pacientes que já tiveram a presença confirmada. Ao clicar em uma linha da lista de atendimentos, os detalhes aparecem à direita e a consulta pode ser iniciada. Quando a consulta é iniciada, o status na tabela de agendamentos é atualizado para 'consulta_em_andamento'.

Figura 31: Seção de Calendário - acesso de profissional da saúde.

Horário	Data	Profissional	Especialidade	Paciente
08:00	01/05/2025	Luisa	Cardiologista	Bruno William

Fonte: Autoria própria.

4.7.2 Histórico.

Ao iniciar a consulta, o profissional de saúde é redirecionado para o prontuário eletrônico, diretamente para o histórico do paciente. No histórico, ele pode acompanhar as consultas passadas do paciente, com o título da Classificação Internacional de Atenção Primária Segunda Edição (CIAP2). Essa ferramenta permite a codificação dos motivos da consulta e dos problemas de saúde. Além disso, no título, também aparecem o profissional que atendeu o paciente e a data da consulta. Na Imagem 32, podemos ver todas as consultas passadas do paciente, e na Imagem 33, podemos visualizar as informações de uma consulta específica.

Figura 32: Histórico do paciente no prontuário eletrônico - acesso de profissional da saúde.

Fonte: Autoria própria.

Figura 33: Histórico do paciente expandido no prontuário eletrônico - acesso de profissional da saúde.

Fonte: Autoria própria.

4.7.3 SOAP.

Na seção SOAP, o profissional de saúde registra as informações da consulta de forma estruturada. Há um campo onde podem ser inseridas as informações subjetivas, que dizem respeito às queixas do paciente, ou seja, aquilo que ele relata espontaneamente, como dores, sintomas, histórico recente ou qualquer percepção sobre sua saúde. Além disso, há um campo para seleção da CIAP2 no item 'Motivo da consulta'.

Em seguida, o profissional registra os dados objetivos, como sinais observáveis, resultados de exames e tudo o que pode ser mensurado clinicamente, como peso, altura, IMC,

pressão arterial, frequência respiratória, frequência cardíaca, entre outros. Na etapa de avaliação, há um campo onde o profissional pode inserir suas hipóteses diagnósticas. Por fim, no plano, há um campo para que o profissional insira as condutas a serem tomadas. Esses fluxos podem ser visualizados nas imagens a seguir. Ao clicar em 'finalizar', o profissional retorna à tela de início de atendimento e pode seguir para outros atendimentos. Esse modelo padronizado facilita a comunicação entre as equipes, organiza o raciocínio clínico e contribui para um prontuário mais claro e eficiente.

Figura 34: Tela de SOAP - Subjetivo no prontuário eletrônico - acesso de profissional da saúde.

The screenshot shows a digital interface for a medical record. At the top, a blue header bar reads "Prontuário Eletrônico". Below it, a white header section titled "Prontuário" with the subtitle "Acompanhe as consultas do paciente". A navigation menu on the left includes "Histórico", "SOAP" (which is highlighted in a yellow box), "Atestado", "Prescrição", "Exames e Encaminhamentos", and "Finalizar". A "Salvar" button is located below the menu. The main content area contains a "Subjetivo" section with the following text:

Paciente masculino, 52 anos, comparece à consulta referindo episódios recorrentes de cefaleia occipital ao final do dia, sensação de cansaço aos mínimos esforços e palpitações intermitentes. Refere histórico familiar positivo para hipertensão e infarto do miocárdio (pai faleceu aos 58 anos por IAM). Nega tabagismo atual, etilismo ocasional. Alimentação desregulada, sedentarismo. Uso esporádico de losartana 50 mg sem orientação médica. Nega dor torácica típica, dispneia paroxística noturna ou ortopneia.

Below this text is a "Motivo da consulta" field containing "A05 - Fadiga" and a "Salvar" button.

Fonte: Autoria própria.

Figura 35: Tela de SOAP - Objetivo no prontuário eletrônico - acesso de profissional de saúde.

Perímetro Cefálico (cm) 57 cm	Peso (kg) 94 kg	Altura (cm) 174 cm	Índice de massa corpórea (IMC) 31,0 kg/m ² (obesidade §)
Perímetro de Panturrilha 36 cm	Pressão arterial(mmHg) 158/98 mmHg	Frequência Respiratória(ppm) 18 rpm	Frequência Cardíaca(bpm) 88 bpm
Temperatura °C 36,6 °C	Saturação O2 97% em ar ambiente	Glicemia 109 mg/dL	

Fonte: Autoria própria.

Figura 36: Tela de SOAP - Avaliação no prontuário eletrônico - acesso de profissional de saúde.

Fonte: Autoria própria.

Figura 37: Tela de SOAP - Plano no prontuário eletrônico - acesso de profissional de saúde.

Fonte: Autoria própria.

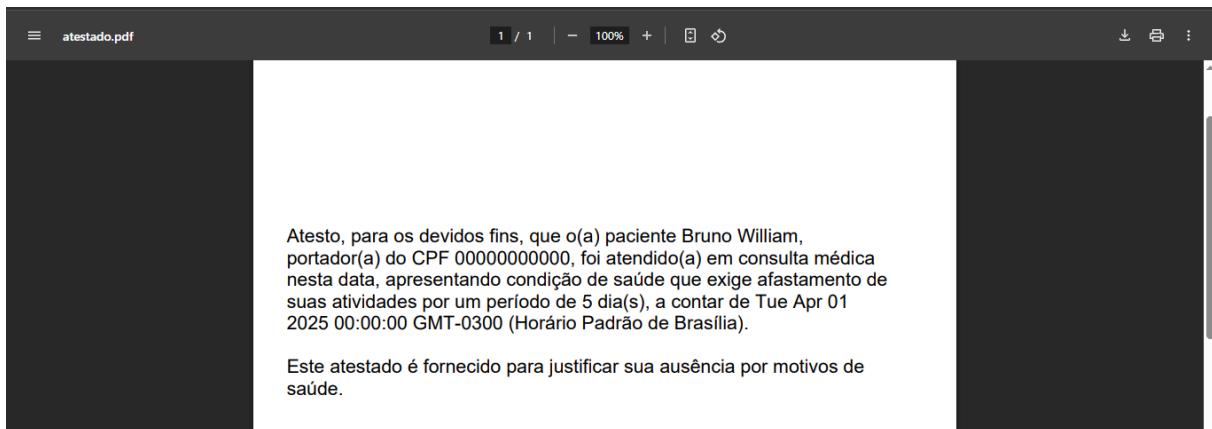
4.7.4 Atestado.

Se necessário, na seção de atestado, o profissional gera um atestado, adicionando o nome do paciente, o CPF, a data da consulta, a hora e a quantidade de dias de atestado que o paciente necessita. Ao clicar em ‘Gerar PDF’, o documento é gerado. Se for necessário imprimir no local, o sistema realiza automaticamente o download do PDF e o salva no backend, para que o paciente possa acessá-lo quando necessário.

Figura 38: Tela de geração de atestado.

Fonte: Autoria própria.

Figura 39: Atestado Gerado



Fonte: Autoria própria.

4.7.5 Prescrição

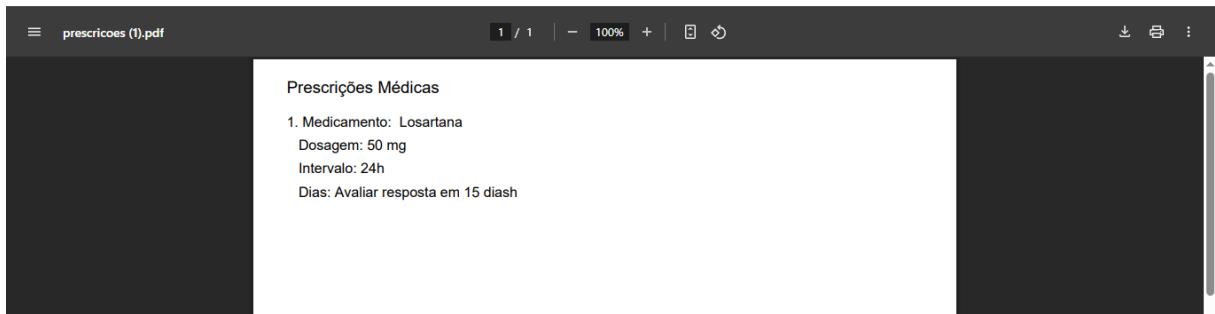
Se necessário, na seção de prescrição, o profissional pode adicionar uma nova prescrição, inserindo o nome da medicação, a dosagem, o intervalo de horas e o intervalo de dias. Ao clicar em ‘Gerar PDF’, o documento é gerado. Caso seja necessário realizar a impressão no local, o sistema baixa automaticamente o PDF e o salva no backend, permitindo que o paciente o acesse quando necessário.

Figura 40: Tela de geração de Prescrição.

Nome do Remédio	Dosagem	Intervalo (horas)	Dias	Adicionar
Losartana - 50 mg	- a cada 24h	0		Adicionar

Fonte: Autoria própria.

Figura 41:Prescrição gerada.



Fonte: Autoria própria.

4.7.6 Exames e Encaminhamentos.

Na seção de exames e encaminhamentos, o profissional de saúde pode solicitar exames e encaminhamentos para o paciente, adicionando o nome do exame, o CID10, a justificativa para o procedimento ou encaminhamento, e observações, se houver.

Figura 42: Tela de geração de pedidos de exames e encaminhamento.

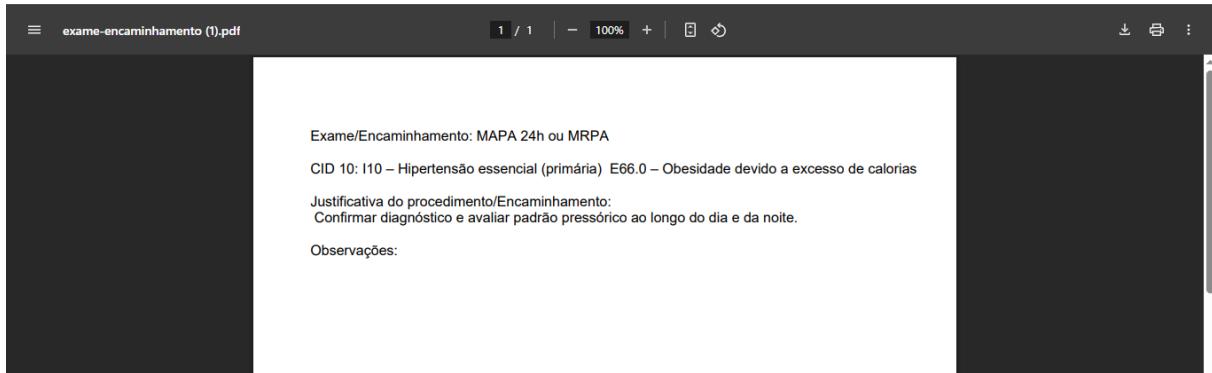
Prontuário Eletrônico

Prontuário
Acompanhe as consultas do paciente

Histórico	SOAP	Atestado	Prescrição	Exames e Encaminhamentos	Finalizar
Exame ou Encaminhamento ECG de repouso Ecocardiograma transtorácico MAPA 24h ou MRPA Exa					
CID 10 I10 – Hipertensão essencial (primária) E66.0 – Obesidade devido a exce					
Justificativa do Procedimento Avaliar sinais de hipertrofia ventricular esquerda (complicação comum d...					
Observações Paciente referiu poliúria, fadiga, cefaleia, ECG aponta alterações no...					

Fonte: Autoria própria.

Figura 43: Exemplo de exame gerado.



Fonte: Autoria própria.

4.7.7 Finalizar Consulta

Após finalizar a consulta, os dados são armazenados nas tabelas de consulta, atestado, prescrição e exames. A tabela de consulta pode ser visualizada na Figura 44. Nesse momento, o status do agendamento é alterado para ‘finalizado’, permitindo que a consulta seja registrada no histórico do paciente. A partir daí, o paciente pode acessar os detalhes da consulta em sua página inicial, incluindo os documentos gerados.

Os arquivos de atestado, prescrição, solicitação de exames e encaminhamentos são armazenados de forma criptografada em tabelas específicas. A tabela de atestado pode ser visualizada na Figura 45, a tabela de exames na Figura 46 e a de prescrição na Figura 47.

Figura 44: Tabela de consulta no banco de dados, preenchida com uma consulta exemplo

ID_CONSULTA	ID_PRONTUARIO	ID_PROFISSIONAL	DATA_CONSULTA	SUBJETIVO	OBJETIVO	AVALIAÇÃO	PLANO	ID_AGENDAMENTO	ID_ATESTADO	ID_EXAME	ID_PRESCRICAO
1	1	1	2025-05-01	eyJpdii6IjFZNHIBewJPDNORnm5iyrmfZ02dhpHL3i6l3RPaytHFNV1E9PSIsImRhdGcE9PSIsImRhdGEIOiIySjU2YnFnUEFvUUd...	eyJpdii6InlBeDFjnQeyJpdii6Imc3MEZCL3V4VFCUldFcTgvNVYxZ0lZTYsOxjSxpJEyeF9PSIsImRhdGERGc9PSIsImRhdGEIOiIvtTU9rZTYvVDlqQW11...ORhaEl...	eyJpdii6Ikg1TFIYeU34VIFCUldFcTgvNVYxZ0lZTYsOxjSxpJEyeF9PSIsImRhdGERGc9PSIsImRhdGEIOiIvtTU9rZTYvVDlqQW11...ORhaEl...	eyJpdii6Ikg1TFIYeU34VIFCUldFcTgvNVYxZ0lZTYsOxjSxpJEyeF9PSIsImRhdGERGc9PSIsImRhdGEIOiIvtTU9rZTYvVDlqQW11...ORhaEl...	1	1	1	1

Fonte: Autoria própria.

Figura 45: Tabela de atestado no banco de dados, preenchida com um atestado exemplo.

ID_ATESTADO	ATESTADO
1	eyJpdii6IjNqenZKFhpZmo4UjJYQXVEK3hrZnc9PSIsImRhdGEiOjIOTThKK2JOU01GcffxN3FqSWRoZmh2ZFhYOFBLUnVnaXo1bzRHmoyVUovKOZI...

Fonte: Autoria própria.

Figura 46: Tabela de exame no banco de dados, preenchida com um exame exemplo.

ID_EXAME	EXAME
1	eyJpdii6IIRIYVNSR3hxUkxIeDhKbFE3K0dHQnc9PSIsImRhGEiOijTb2lFaERrSndjU3lFbWt6Q1l5eldLTmk3L3pZMzJyM1lZckpqUjFvSmtFM1lpUnhPVTZCQU5...

Fonte: Autoria própria.

Figura 47: Tabela de prescrição no banco de dados, preenchida com uma prescrição exemplo.

ID_PRESCRICAO	PRESRICAO
1	eyJpdii6Ilgvdm4vQjhYVHFJMzInb0dSM0c1R3c9PSIsImRhGEiOijpZnROVOIRRj9lV25NUEc5S3h1QnpzTXdJL0pkVEVueHnRDV3eEM0VTIyZTdGRm5nOGt...

Fonte: Autoria própria.

Os testes realizados no prontuário eletrônico permitiram validar o funcionamento dos principais fluxos da aplicação, garantindo a integração entre o frontend e o backend. Todas as ações simuladas, como autenticação com diferentes níveis de acesso, cadastro de usuários, agendamentos, execução e encerramento de consultas, além da geração e consulta de documentos clínicos, ocorreram de forma satisfatória, sem falhas críticas. Os resultados demonstram que o sistema é funcional, atende aos requisitos propostos e constitui uma base sólida para futuras implementações.

5 CONCLUSÃO

A realização deste trabalho teve como objetivo o desenvolvimento de um protótipo funcional de Prontuário Eletrônico do Paciente (PEP), voltado para atender às necessidades da DIRQS, buscando otimizar os processos de registro e acesso às informações de saúde, além de assegurar conformidade com as exigências legais e técnicas. Para isso, foram realizados estudos sobre os principais modelos de prontuário disponíveis na literatura, bem como sobre a legislação vigente, com destaque para a LGPD.

A etapa de levantamento de requisitos funcionais e não funcionais foi essencial para compreender as demandas específicas dos usuários, resultando na construção de um modelo de dados representado por um Diagrama Entidade-Relacionamento. Em seguida, o processo de prototipação permitiu o desenvolvimento de uma interface que simula os fluxos operacionais da DIRQS, testando a coerência entre os dados, as funcionalidades e o uso real.

Durante o desenvolvimento, algumas dificuldades técnicas foram enfrentadas, como a integração segura entre as camadas da aplicação e a definição de critérios de usabilidade acessíveis a todos os perfis de usuários. Além disso, houve a necessidade constante de equilibrar o escopo do projeto com os recursos disponíveis, o que levou à priorização de funcionalidades essenciais em detrimento de outras que poderão ser incorporadas futuramente.

O protótipo apresenta limitações importantes, como a ausência de autenticação baseada em múltiplos fatores e a falta de testes em ambiente real com usuários da DIRQS. Essas limitações, no entanto, foram mapeadas e documentadas como pontos de atenção para fases futuras de desenvolvimento.

Apesar dos desafios, o processo resultou em aprendizados significativos, como o aprofundamento nas boas práticas de segurança da informação em saúde, a importância de um design centrado no usuário e o papel crítico da documentação na escalabilidade em sistemas. Além disso, o projeto proporcionou uma vivência prática sobre o ciclo de desenvolvimento de sistemas na área da saúde, destacando o valor da interoperabilidade e do cumprimento da legislação.

Conclui-se, portanto, que os objetivos propostos foram alcançados de forma consistente, e o protótipo entregue representa uma base sólida para futuras evoluções do sistema. A viabilidade técnica do projeto foi reconhecida pelo coordenador da Divisão de Sistemas, conforme registrado no anexo, reforçando a aderência do sistema às necessidades operacionais da DIRQS e abrindo caminho para implementações futuras mais robustas.

6 TRABALHOS FUTUROS

Para trabalhos futuros, podem ser realizados testes em ambiente real com profissionais da área da saúde para avaliar a usabilidade, eficiência e confiabilidade do prontuário eletrônico desenvolvido. A aplicação prática do sistema permitirá identificar eventuais melhorias na interface, na organização dos dados e na experiência geral do usuário. A coleta da experiência direto dos profissionais será fundamental para orientar ajustes na plataforma, garantindo que ela atenda de forma eficaz às necessidades do dia a dia em ambientes clínicos e hospitalares. Além disso, será possível observar o desempenho do sistema em condições reais de operação, testando sua estabilidade e resposta em diferentes contextos de uso.

Além disso, uma possibilidade de evolução do projeto é o desenvolvimento de um módulo de geração automática de relatórios, com base nas informações armazenadas nos prontuários eletrônicos. Esse recurso deve permitir a extração de dados relevantes de forma estruturada, facilitando análises clínicas, auditorias internas, acompanhamento de pacientes e tomada de decisões por parte da gestão. Além de possibilitar a geração de relatórios estatísticos para visualização de padrões de atendimento, prevalência de doenças, evolução de tratamentos e produtividade dos profissionais de saúde.

1. ANEXO I – Declaração de Viabilidade Técnica do Protótipo

Declaro que Bruno William Miranda Silva, na posição de estagiário da Divisão de Sistemas do Centro de Tecnologia da Informação e Comunicação (CTIC) da Universidade Federal de Uberlândia (UFU), desde 01 de junho de 2023, tem demonstrado excelente competência e dedicação no exercício de suas funções.

Durante o período de estágio, desempenhou um papel fundamental no estudo de legislação relacionada e prototipação para aplicações do projeto de Prontuário Eletrônico, seguindo os padrões digitais do governo (Design System) e alinhada às metas do Plano Diretor de TIC da UFU.

Além disso, Bruno esteve diretamente envolvido em projetos importantes, como o desenvolvimento para o sistema de Manuais UFU, documentando aplicações e serviços mantidos pelo CTIC . Sua capacidade de aprendizagem de novas tecnologias foi fundamental para o desenvolvimento destes projetos.

Fabiano Silvério Ribeiro Alves
Coordenador da Divisão de Sistemas (DS)
Centro de Tecnologia da Informação e Comunicação (CTIC)
Universidade Federal de Uberlândia (UFU)

Documento assinado digitalmente
 FABIANO SILVERIO RIBEIRO ALVES
Data: 07/05/2025 16:58:56-0300
Verifique em <https://validar.iti.gov.br>

REFERÊNCIAS

Albergaria, E. T. D. & Bax, M. P. (2013). Propriedades dos documentos arquivísticos aplicados ao prontuário eletrônico do paciente. In XIV Encontro Nacional de Pesquisa e Ciência da Informação. ENANCIB, Florianópolis.

Atendimentos. Estratégia e-SUS APS. Disponível em: <https://saps-ms.github.io/Manual-eSUS_APS/docs/PEC/PEC_06_atendimentos/#61-lista-de-atendimentos>. Acesso em: 4 abr. 2025.

BEZERRA, Selene Maria. Prontuário Eletrônico do Paciente: uma ferramenta para aprimorar a qualidade dos serviços de saúde. **Revista Meta: Avaliação**, v. 1, n. 1, p. 73-82, 2009.

BLAZHEVSKI, Dobre et al. Modes of operation of the AES algorithm. 2013.

BONEH, Dan; SHOUP, Victor. A graduate course in applied cryptography. **Draft 0.5**, 2020.

Brasil. Ministério da Saúde (MS). Secretaria de Atenção à Saúde, Política Nacional de Humanização da Saúde. Documento Base. 4^a ed. Brasília: MS; 2007.

CALDEIRA, Carlos. PostgreSQL: Guia Fundamental. 2015.

CASCIARO, Mario; MAMMINO, Luciano. **Node.js Design Patterns**. Packt Publishing Ltd, 2016.

CELUPPI, Ianka Cristina et al. Dez anos do Prontuário Eletrônico do Cidadão e-SUS APS: em busca de um Sistema Único de Saúde eletrônico. **Revista de Saúde Pública**, v. 58, p. 23, 2024.

CHOWDHURY, Md Atique Reza et al. On the untriviality of trivial packages: An empirical study of npm javascript packages. **IEEE Transactions on Software Engineering**, v. 48, n. 8, p. 2695-2708, 2021.

Conheça o ASSER | PROGEP/UFU. Progep.ufu.br. Disponível em: <<https://progep.ufu.br/acontece/2024/08/conheca-o-asser>>. Acesso em: 15 abr. 2025b.

CONSELHO FEDERAL DE MEDICINA. *Prontuário Médico*. 1999. Disponível em: <https://portal.cfm.org.br/artigos/prontuario-medico>. Acesso em: 5 dez. 2024.

Costa CGA da. Desenvolvimento e avaliação tecnológica de um sistema de prontuário eletrônico do paciente, baseado nos paradigmas da World Wide Web e da engenharia de software [Dissertação]. [Campinas]: UNICAMP; 2001.

Cruz JAS. Prontuário Eletrônico de Pacientes (PEP): Políticas e Requisitos Necessários à Implantação no HUSM [Dissertação]. [Santa Maria]: Universidade de Santa Maria; 2011.

CSTB. For the Record: Protecting Electronic Health Information. Washington, DC: National Academy Press, 1997.

Diretoria de Qualidade de Vida e Saúde do Servidor | PROGEP/UFU. Progep.ufu.br. Disponível em: <<https://progep.ufu.br/unidades-organizacionais/diretoria/diretoria-de-qualidade-de-vida-e-saude-do-servidor>>. Acesso em: 30 mar. 2025a.

ELMASRI, Ramez et al. Sistemas de banco de dados. 2005.

FIARRESGA, Victor Manuel Calhabrê. **Criptografia e matemática.** 2010. Dissertação de Mestrado. Universidade de Lisboa (Portugal).

FLANAGAN, David. **JavaScript: o guia definitivo.** Bookman Editora, 2012.

GONÇALVES, Rodrigo Franco et al. Uma proposta de processo de produção de aplicações Web. **Production**, v. 15, p. 376-389, 2005.

HAWRYLISZYN, Larissa Oliveira; COELHO, Natalia Gavioli Souza Campos; BARJA, Paulo Roxo. Lei Geral De Proteção De Dados (LGPD): O Desafio De Sua Implantação Para A Saúde. **Revista Univap**, v. 27, n. 54, 2021.

História do estetoscópio. Littmann.com.br. Disponível em: <https://www.littmann.com.br/3M/pt_BR/estetoscopios-littmann-br/instituto-educacional/hist%C3%B3ria/>. Acesso em: 15 abr. 2025.

ICP-Brasil. Instituto Nacional de Tecnologia da Informação. Disponível em: <<https://www.gov.br/iti/pt-br/acesso-a-informacao/perguntas-frequentes/icp-brasil>>. Acesso em: 23 jan. 2025.

ILIC, Branimir. Best Way to Structure an Express.js App: Modular vs Layered Approach Explained | Medium. Medium. Disponível em: <<https://medium.com/%40branimir.ilic93/express-js-best-practices-modular-vs-layered-approach-for-medium-and-large-appsintroduction-626e61cc908d>>. Acesso em: 8 abr. 2025.

Lau, F & Price, M. B. J. (2014). Toward a Coordinated Electronic Health Record (EHR) Strategy for Canada. Creating Strategic Change. Canada: Canadian Healthcare, p. 35.

Lei Geral de Proteção de Dados Pessoais (LGPD). Ministério do Esporte. Disponível em: <<https://www.gov.br/esporte/pt-br/acesso-a-informacao/lgpd>>. Acesso em: 23 jan. 2025.

LIMA, José Janailton de; VIEIRA, Larissa Gabrielle Dias ; NUNES, Marília Mendes. Computerized nursing process: development of a mobile technology for use with neonates. **Revista Brasileira de Enfermagem**, v. 71, n. suppl 3, p. 1273–1280, 2018. Disponível em: <<https://www.scielo.br/j/reben/a/44PFMYy3g63CxqSLMPNLvpS/?lang=en>>.

LÚCIO, Marcelo; LUIZ, Silva ; VIRGINIO, Aparecido. **Manual de Certificação para Sistemas de Registro Eletrônico em Saúde.** [s.l.: s.n., s.d.]. Disponível em: <http://www.sbis.org.br/certificacao/Manual_Certificacao_SBIS-CFM_2016_v4-2.pdf>.

MARDAN, Azat. Using express. js to create node. js web apps. In: **Practical Node. js: Building Real-World Scalable Web Apps.** Berkeley, CA: Apress, 2018. p. 51-87.

Mardan, A. (2018). Using Express.js to Create Node.js Web Apps. In: Practical Node.js. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3039-8_2

Massad E, Marin H de F, Azevedo Neto RS de. O prontuário eletrônico do paciente na assistência, informação e conhecimento médico. São Paulo: FMUSP/UNIFESP/OPAS; 2003.

MELO, Erik Cristóvão Araújo; ENDERS, Bertha Cruz. Construção de sistemas de informação para o processo de enfermagem: uma revisão integrativa. **Journal of Health Informatics**, v. 5, n. 1, 2013.

MENEZES, Alfred J.; VAN OORSCHOT, Paul C.; VANSTONE, Scott A. **Handbook of applied cryptography**. CRC press, 2018.

MEYER, Eric; WEYL, Estelle. **CSS: The Definitive Guide**. " O'Reilly Media, Inc.", 2023.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *Advanced Encryption Standard (AES)*. FIPS PUB 197. Gaithersburg, MD: NIST, 2001. Disponível em: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf>. Acesso em: 15 abr. 2025.

NERI, Norberto; LANGE; BIBIANA, Neyza; et al. VANTAGENS E DESVANTAGENS DA UTILIZAÇÃO DO IONIC FRAMEWORK PARA O DESENVOLVIMENTO DE APLICATIVOS MÓVEIS. [s.l.: s.n., s.d.]. Disponível em: <https://semanaacademica.org.br/system/files/artigos/artigo_74.pdf>.

O que é a API RESTful? – Explicação sobre a API RESTful – AWS. Amazon Web Services, Inc. Disponível em: <<https://aws.amazon.com/pt/what-is/restful-api/>>. Acesso em: 8 abr. 2025.

OWASP Top Ten | OWASP Foundation. Owasp.org. Disponível em: <<https://owasp.org/www-project-top-ten/>>. Acesso em: 31 mar. 2025.

PANITZ, Leandro Manassi. Registro eletrônico de saúde e produção de informações da atenção à saúde no SUS. 2014. 179 f. Dissertação (Mestrado Profissional em Saúde Pública) - Escola Nacional de Saúde Pública Sergio Arouca, Fundação Oswaldo Cruz, Rio de Janeiro, 2014.

PDF.js viewer. Cfm.org.br. Disponível em: <<https://sistemas.cfm.org.br/normas/visualizar/resolucoes/BR/2007/1821>>. Acesso em: 21 jan. 2025.

PDF.js viewer. Cfm.org.br. Disponível em: <<https://sistemas.cfm.org.br/normas/visualizar/resolucoes/br/2002/1638>>. Acesso em: 22 jan. 2025.

POSTAL, Lucas et al. Sistema de agendamento online: uma ferramenta do PEC e-SUS APS para facilitar o acesso à Atenção Primária no Brasil. **Ciência & Saúde Coletiva**, v. 26, n. 6, p. 2023-2034, 2021.

POWSNER, Seth M.; WYATT, Jeremy C.; WRIGHT, Patricia. Opportunities for and challenges of computerisation. **The lancet**, v. 352, n. 9140, p. 1617-1622, 1998.

PROVOS, Niels; MAZIERES, David. A future-adaptable password scheme. In: **USENIX annual technical conference, FREENIX track**. 1999. p. 81-91.

Purkayastha S, Allam R, Maity P, Gichoya JW. **Comparison of Open-Source Electronic Health Record Systems Based on Functional and User Performance Criteria**. Healthc Inform Res. 2019 Apr;25(2):89-98. doi: 10.4258/hir.2019.25.2.89. Epub 2019 Apr 30. PMID: 31131143; PMCID: PMC6517630.

RATHOD, Urvesh; SONKAR, Meghna; CHANDAVARKAR, B. R. An experimental evaluation on the dependency between one-way *hash* functions and salt. In: **2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)**. IEEE, 2020. p. 1-7.

SILVA, Maurício Samy. **HTML5–2ª Edição: A linguagem de marcação que revolucionou a web**. Novatec Editora, 2014.

Sistemas de gerenciamento de banco de dados [recurso eletrônico] / Raghu Ramarkrishnan, Johannes Gehrke ; tradução: Célia Taniwake. – 3. ed. – Dados eletrônicos. – Porto Alegre : AMGH, 2011.

SOMMERVILLE, I. Engenharia de *software*, 9a edição. São Paulo: Pearson Prentice Hall, 2011. 529 p.

Souza, R. S. (2016). Prontuário eletrônico: ótica do profissional de saúde da atenção primária (Dissertação de Mestrado). Mestrado em Ensino em Saúde. Universidade do Estado do Rio Grande do Sul, Porto Alegre. Recuperado de:<https://www.lume.ufrgs.br/handle/10183/171394>

Turkcan A, Zeng B, Muthuraman K, Lawley M. Sequential clinical scheduling with service criteria. Eur J Oper Res 2011; 214(3):780-795.

VAN BEMMEL, J. H. Handbook of medical informatics. 1997.

Weed LL. Medical records that guide and teach. N Engl J Med. 1968 Mar 14;278(11):593-600. doi: 10.1056/NEJM196803142781105. PMID: 5637758.