SPECEE - Sistema de Prontuário Eletrônico Um estudo de caso da clínica Escola de Enfermagem da Universidade Católica do Salvador

Deise M. A. Paz, Francisco Virissimo, Isabelle C. J. Lopes, Nelson A. C. Neto, Fernando C. R. Borges

[1]Escola de Tecnologias
Universidade Católica do Salvador (UCSAL)
Av. Prof. Pinto de Aguiar, 2589 Pituaçu, CEP: 41740-090
Salvador / BA, Brasil

{deise.paz, francisco.virissimo, isabelle.lopes, nelsonafonso.neto}
@ucsal.edu.br, fernando.borges@pro.ucsal.br

Abstract. *The Electronic Health Record System (SPECEE) at the Nursing School Clinic of the Catholic University of Salvador (UCSAL) advances the management and organization of patient health information. This study outlines the development, implementation, and benefits of SPEE, focusing on the enhancement of data accessibility, security, and operational efficiency. By transitioning from manual record-keeping to an electronic system, the clinic aims to improve patient care and comply with data protection regulations such as the General Data Protection Law (LGPD). The system's design incorporates web technologies and security measures to ensure the confidentiality and integrity of patient data.*

Resumo. *O Sistema de Prontuário Eletrônico de Saúde (SPECEE) da Clínica Escola de Enfermagem da Universidade Católica de Salvador (UCSAL) avança na gestão e organização das informações de saúde dos pacientes. Este estudo des-creve o desenvolvimento, implementação e benefícios do sistema de prontuário eletrônico, com foco na melhoria da acessibilidade dos dados, segurança e eficiência operacional. Ao migrar da manutenção manual de registros para um sistema eletrônico, a clínica visa melhorar o atendimento ao paciente e cumprir as regulamentações de proteção de dados, como a Lei Geral de Proteção de Dados (LGPD). O design do sistema incorpora tecnologias web e medidas de segurança para garantir a confidencialidade e integridade dos dados do paciente.*

1. Introdução

O Prontuário Eletrônico do Paciente (PEP) nasceu da necessidade de organização e padronização das informações dos pacientes, sendo um sistema que acompanha o registro de saúde do indivíduo desde o seu nascimento até a sua morte [Almeida et al. 2016].

Esse documento tem a finalidade de registrar toda e qualquer situação relacionada à saúde do paciente, com o objetivo de manter um histórico do mesmo para que seja usado na assistência prestada quando necessário e embora esse registro seja compartilhado entre os profissionais de saúde, o mesmo é protegido por leis, como a LGPD . [Thofehrn and Lima 2006].

De acordo com [Silva and Soares 2016] , a adoção do Prontuário Eletrônico do Paciente (PEP) é essencial para estabelecer um processo contínuo e consistente no registro cronológico das informações clínicas. Embora os autores destaquem a importância da implementação do PEP, também ressaltam algumas fragilidades do prontuário eletrônico, entre as quais se destaca o inadequado mapeamento das funcionalidades, que devem ser implementadas de acordo com as necessidades dos usuários.

A evolução da Tecnologia da Informação e Comunicação (TIC) tem transformado a disseminação de informações pela internet, permitindo que pacientes e médicos autorizados acessem o PEP de forma on-line. Esse sistema é hospedado em um servidor externo, conforme destacado por [Figueiredo and Oliveira 2024]. Essa abordagem simplifica significativamente o acompanhamento e a atualização das informações médicas do paciente.

Apesar de representar uma modernização na coleta e armazenamento de dados do paciente, o PEP levanta uma preocupação crucial: a segurança e privacidade dessas informações. Diante dessa necessidade, surgiram mecanismos, leis e medidas de segurança destinados a preservar a confidencialidade desses dados, tais como: a Resolução Nº 7, datada de 24 de novembro de 2016, emitida pelo Ministério da Saúde. Essa resolução estabelece que as informações relativas à atenção básica devem ser registradas por meio de prontuários eletrônicos do paciente, contribuindo para assegurar a integridade e confidencialidade desses registros. [Magnagnagno et al. sd]

[Hawryliszyn et al. 2021] destacam que a Lei Geral de Proteção de Dados (LGPD) tem como principal propósito resguardar as informações pessoais dos indivíduos. No contexto específico da relação paciente-hospital, o paciente é identificado como Titular, o hospital como Controlador, os colaboradores como Operadores, e o Encarregado de Dados é o colaborador designado pelo hospital para ser o responsável pela comunicação com a Autoridade Nacional de Proteção de Dados (ANPD), uma autarquia federal vinculada ao Ministério da Justiça e Segurança Pública que atua como órgão fiscalizador. É fundamental que todos os profissionais estejam em conformidade com a lei, desde a recepção até a alta do paciente, principalmente quando do envolvimento de dados sensíveis.

O consentimento do paciente (ou responsável legal) desempenha um papel crucial na coleta, tratamento e utilização dos dados pelo hospital. É fundamental informar claramente a necessidade e o propósito da obtenção dessas informações como algo indispensável. Porém o paciente pode solicitar a exclusão de seus dados a qualquer momento. Os titulares têm o direito de confirmar, revogar, corrigir, anonimizar, bloquear ou eliminar suas informações. [Hawryliszyn et al. 2021].

A Universidade Católica do Salvador dispõe, dentre outros serviços para a comunidade, de uma Clinica Escola de Enfermagem que atualmente atende ao seu público interno. A coleta de dados do atendimento é feito a partir do preenchimento dos diversos formulários de controle e, quando necessário, busca-se a ficha do paciente em pastas

organizadas em um armário próprio.

Diante desse cenário, o presente trabalho tem como objetivo desenvolver uma solução de prontuário eletrônico batizado como Sistema de Prontuário Eletrônico do Paciente da Clinica Escola de Enfermagem da Universidade Católica do Salvador (SPECEE).

A incorporação desse prontuário eletrônico traz um avanço significativo de modernização e eficiência dos serviços de saúde prestados pela Clinica Escola de Enfermagem da Universidade Católica do Salvador, com vistas a ampliar a agilidade no controle e coleta das informações necessárias para o atendimento, no aumento da segurança dos dados e na eficiência operacional dos processos da clínica. Sendo assim, busca-se maiores benefícios tanto para os profissionais de saúde envolvidos na clínica, quanto para os pacientes atendidos.

A divisão do artigo apresenta na 1ª seção uma introdução ao tema, descrevendo a ideia central do trabalho, na 2ª seção a fundamentação teórica que inclui ferramentas utilizadas no processo de desenvolvimento, conceitos importantes e resultados de outros autores com produção acadêmica relacionada ao tema do artigo, na seção 3 o laboratório da construção do sistema, na seção 4 a prova de conceito, na seção 5 a conclusão e trabalhos futuros, e por último as referências bibliográficas utilizadas na pesquisa.

O objetivo central deste trabalho foi desenvolver um sistema que atenda às necessidades específicas da Clínica Escola da Universidade Católica do Salvador, automatizando processos para proporcionar maior agilidade no atendimento e na coleta das informações necessárias, bem como na gestão desses dados.

Como objetivos secundários, o prontuário eletrônico visa otimizar a acessibilidade das informações para pessoas autorizadas, facilitar a gestão dos profissionais ativos por período, proporcionar maior controle da manipulação dos registros, promover agilidade no atendimento, oferecer flexibilidade no atendimento para atender às necessidades específicas da clínica escola e fornecer funcionalidades personalizadas que atendam às necessidades dos usuários.

2. Fundamentação Teórica

No decorrer do atendimento aos pacientes, surgiu a necessidade imperativa de registrar, de maneira cronológica e sistemática, todas as informações pertinentes ao estado clínico de cada indivíduo. No entanto, esse procedimento era realizado de forma manual. Foi então, diante dessa demanda premente de manter um registro consistente e acessível ao longo do tempo, que surgiu a concepção do prontuário eletrônico do paciente. O termo prontuário tem origem no latim *promptuarium*, que significa um local onde informações essenciais, prontas para serem acessadas a qualquer momento, são guardadas e organizadas [da Silva 2021].

A dinâmica do PEP traz inúmeras vantagens para a assistência médica, tais como facilidade de acesso aos dados por pessoas autorizadas, segurança das informações, aumento da produtividade e integridade dos registros. É importante ressaltar que o PEP é protegido por lei, sendo a Resolução CFM Nº 1.638, de 10 de julho de 2002, uma das legislações fundamentais nesse sentido [Bezerra 2015]

2.1. Segurança em Sistemas Web

A era da internet revolucionou a comunicação, o acesso à informação e as transações comerciais, porém, também trouxe consigo desafios significativos de segurança, demandando uma atenção especial aos sistemas web [Mozilla Developer Network]. No contexto de um prontuário eletrônico, a segurança assume um papel crucial, especialmente considerando as regulamentações como a Lei Geral de Proteção de Dados (LGPD) e as diretrizes estabelecidas pelo Conselho Federal de Medicina (CFM) [Telles et al. 2021].

A LGPD tem como objetivo proteger a liberdade e a privacidade dos indivíduos, regulamentando o tratamento de dados por organizações públicas e privadas. Ela estabelece os papéis de Controlador, Operador e Encarregado, além de abranger diversas operações com dados pessoais, exigindo transparência e justificativa para seu uso. No âmbito público, o tratamento de dados está relacionado à execução de políticas públicas, permitindo o compartilhamento transparente de informações. A legislação garante direitos aos titulares dos dados e oferece diretrizes detalhadas para sua proteção, conforme descrito no guia de boas práticas para implementação na Administração Pública Federal [Governo Brasileiro 2018].

Por sua vez, o CFM atua como órgão regulador supervisionando a prática médica no Brasil, estabelecendo diretrizes éticas e normas para a conduta dos médicos, com o intuito de garantir a qualidade e segurança no atendimento médico. Além disso, o CFM é responsável pelo registro e fiscalização do exercício da profissão médica, promovendo a atualização e o aprimoramento contínuo dos profissionais. Suas ações visam assegurar a integridade e o bem-estar dos pacientes, bem como a excelência no cuidado de saúde prestado pelos médicos em todo o país [CFM et al.].

Para auxiliar os desenvolvedores na promoção de uma cultura de segurança, existe o Projeto de Segurança de Aplicações Web de Código Aberto, conhecido como OWASP, uma entidade internacional sem fins lucrativos voltada para a proteção de aplicativos online. Um dos projetos mais reconhecidos associados ao OWASP é o OWASP Top 10, que identifica as 10 principais vulnerabilidades de segurança em aplicações web. [OWASP 2024] e desses, selecionamos as que se aplicam melhor a uma primeira versão do projeto:

- ***Cryptographic Failures*** **(Falhas Criptográficas)**: A criptografia é crucial para proteger os dados sensíveis dos pacientes em trânsito e em repouso no sistema.
- ***Injection*** **(Injeção)**: Importante para evitar a inserção de código malicioso no sistema, protegendo contra ataques que poderiam comprometer os dados do paciente.
- ***Identification and Authentication Failures*** **(Falhas de Identificação e Autenticação)**: Mecanismos robustos de autenticação são essenciais para garantir que apenas usuários autorizados possam acessar o sistema e as informações dos pacientes.

O tratamento dessas vulnerabilidades proporciona um nível de segurança para o sistema. No entanto, é importante reconhecer que a segurança de um sistema é um processo contínuo de amadurecimento. Como a tecnologia e os métodos de ataque evoluem, novas vulnerabilidades podem surgir, exigindo atualizações e aprimoramentos contínuos. Portanto, a cobertura de outras possíveis vulnerabilidades será tratada como trabalho futuro, visando garantir que o sistema se mantenha seguro e confiável ao longo do tempo.

2.2. Tecnologias Utilizadas na Pesquisa

A tecnologia Angular, utilizada na construção do *front-end* do sistema, foi escolhida por ser uma plataforma de desenvolvimento web de código aberto criada pela Google em 2016. Utilizando JavaScript e *TypeScript*, Angular permite a criação de aplicações robustas e escaláveis acessíveis via navegadores, adotando o padrão *Model-View-View-Model* (MVVM), que facilita a organização, manutenção e testes do código, sendo amplamente utilizada devido a sua comunidade ativa e uma rica coleção de ferramentas e bibliotecas disponíveis [Jr. and Mercado 2018].

Complementando o uso do Angular, o TypeScript foi integrado ao projeto por ser uma extensão do JavaScript com um sistema de módulos, classes, interfaces e um sistema de tipos estático. TypeScript visa facilitar o desenvolvimento de aplicações, oferecendo suporte leve aos programadores. Seus sistemas de módulos e tipos são flexíveis e acessíveis, adaptados para suportar práticas comuns de programação em JavaScript. Além disso, proporciona ferramentas e experiências de desenvolvimento de IDE que são familiares em linguagens como C e Java [Bierman et al. 2014]).

Para garantir a responsividade e a consistência visual do prontuário eletrônico, o Bootstrap foi empregado como *framework front-end*. Este *framework* simplifica o desenvolvimento de interfaces web, oferecendo componentes prontos como *tooltips*, menus *dropdown* e modais. Criado para padronizar interfaces no Twitter, o Bootstrap facilita a criação de sites com boa usabilidade e sua grade flexível ajuda na organização do layout para dispositivos variados, eliminando a necessidade de múltiplas versões do site [Santiago et al. 2020].

Associadas a essas tecnologias, foram feitos ajustes diretamente no HTML e SCSS para estruturar e estilizar o conteúdo visual do prontuário eletrônico. HTML, sendo uma linguagem para criar páginas web, usa tags para estruturar textos, links e imagens e o SCSS, uma extensão do CSS, organiza estilos de forma estruturada, com classes definidas no SCSS e aplicadas no HTML. Isso permite personalização visual detalhada como cores, tamanhos de fonte e alinhamentos, garantindo uma interface agradável e funcional [Bueno 2018].

Dessa forma, a combinação dessas tecnologias foi fundamental para o desenvolvimento de um prontuário eletrônico eficiente, robusto e esteticamente agradável, integrando funcionalidades avançadas com uma interface intuitiva e responsiva.

O *back end* foi estruturado com padrão de arquitetura de software MVC (*Model-View-Controller*) que divide a aplicação em três partes: o Modelo (*Model*), responsável pela lógica de negócios e pelo gerenciamento dos dados; a Visão (*View*), que é responsável pela apresentação dos dados ao usuário; e o Controlador(*Controller*), que atua como um intermediário entre o Modelo e a Visão, processando as entradas do usuário e atualizando o Modelo e a Visão conforme necessário. Essa divisão de responsabilidades oferece diversas vantagens, como a facilidade de manutenção, a reutilização de componentes, e a escalabilidade, permitindo que a aplicação cresça de forma organizada e eficiente. Além disso, o MVC torna a estrutura de código mais clara e organizada, o que facilita o desenvolvimento em conjunto e a adaptação às mudanças nos requisitos da aplicação [Luciano and Alves 2011].

O protuário eletrônico foi desenvolvido com Java que, de acordo com o estudo

de [da Silva 1996], surgiu como uma linguagem de programação completa e altamente versátil, com uma história marcada pela sua origem nos laboratórios da *Sun Microsystem*s nos anos 90. Java foi desenvolvida com o objetivo de simplificar e tornar mais eficiente a criação de software para uma grande variedade de dispositivos eletrônicos.

Claro e Sobral, em ”Programação em Java”, Florianópolis, SC: Pearson EducationA, ressaltam que a capacidade da linguagem de ser compilada e interpretada proporcionou uma portabilidade sem precedentes, permitindo que os programas escritos em Java fossem executados em diferentes plataformas sem a necessidade de recompilação. Além disso, sua arquitetura neutra e segurança integrada tornaram-na uma opção popular para o desenvolvimento de aplicativos distribuídos e de rede [Claro and Sobral 2008].

O Java Development Kit (JDK) oferece uma grande variedade de ferramentas e bibliotecas para que os desenvolvedores criem desde aplicativos simples até aplicações corporativas complexas. A abordagem orientada a objetos de Java, aliada à sintaxe simplificada e ao gerenciamento automático de memória, torna a linguagem acessível para iniciantes e poderosa o suficiente para projetos de grande escala. Dessa forma, as características de portabilidade, segurança, versatilidade e facilidade de uso tornam o Java uma ótima opção para diversas aplicações em diferentes setores da indústria de software [Rabello 2009].

Com o passar dos anos novos *frameworks* sugiram com intuito de otimizar o desenvolvimento de software. Para o desenvolvimento desse prontuário eletrônico foi utilizando o framework Spring Boot, que é destacado como um poderoso *framework* Java para o desenvolvimento de serviços *web RESTful*, sua abordagem baseada no padrão MVC simplifica significativamente o processo de criação e implantação de aplicativos, eliminando a necessidade de configurações complexas através de arquivos XML.

O Spring Boot se destaca por sua capacidade de lidar com aplicações web de grande escala, o que é essencial para administrar as operações intensivas de dados do prontuário eletrônico. O Spring Boot é particularmente eficiente para aplicações web de grande porte, o que é importante para administrar as operações de dados do prontuário eletrônico. Como discutido por [Dhalla 2021] , o Spring Boot foi a escolha coerente para complementar o Java neste projeto.

Além disso, o Spring Boot oferece uma ampla gama de recursos, incluindo suporte para servidores de aplicação, integração fácil com outros projetos Spring e forte segurança por meio de ferramentas como o *Spring Security*. Essas características tornam o Spring Boot uma escolha popular para desenvolvedores que buscam uma solução robusta e eficiente para construir aplicativos empresariais escaláveis. [Boaglio 2017]

As ferramentas descritas acima desempenham um papel relevante no presente trabalho, atuando em conjunto para o funcionamento adequado do prontuário eletrônico, com a utilização do banco de dados PostgreSQL, responsável pela persistência dos dados.

PostgreSQL é um sistema de código aberto e não há custos de licenciamento, conhecido por sua robustez e conformidade com as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), essenciais para a integridade dos dados. Oferece suporte a transações complexas, controle de concorrência avançado e uma vasta gama de tipos de dados e funções. Sua capacidade de escalar desde pequenas aplicações até grandes sistemas empresariais, juntamente com seu suporte a múltiplos sistemas o-

peracionais, incluindo Windows, Linux e macOS, torna-o extremamente versátil tornando uma escolha sólida para ser utilizado no projeto do SPEP.

A adoção do PostgreSQL pela BASF para suportar milhares de usuários simultâneos e transações rápidas é um exemplo de sua eficácia em ambientes de alta demanda. Essas características fazem do PostgreSQL uma escolha confiável e eficiente para diversas aplicações de banco de dados [Santos and Mendes s d]).

2.3. Trabalhos correlatos à pesquisa

De acordo com [da Rocha et al. 2024] , a migração para os prontuários eletrônicos representa um avanço significativo na gestão de informações médicas. O PEP, desenvolvido com software livre permite um gerenciamento eficiente dos dados dos pacientes, o que torna possível a gestão dessas informações. A adoção de ferramentas livres reduz custos, garante a atualização contínua e a flexibilidade multiplataforma, fundamentais para a eficiência operacional e a melhoria do atendimento ao paciente [da Rocha et al. 2024]

O artigo de [de Souza et al. 2020] analisa a deficiência dos sistemas de saúde no compartilhamento de informações médicas e propõe uma solução através da criação de um Sistema de Prontuário Eletrônico do Paciente (PEP). O sistema utiliza tecnologias como Java, Apache Maven, Spring Boot e PostgreSQL para simplificar o armazenamento e a recuperação de dados médicos. A consulta a profissionais da saúde durante o desenvolvimento permitiu que o sistema atendesse às necessidades práticas, ao compartilhar o histórico médico dos pacientes de forma eficiente e segura, o que contribuiu para a melhoria da qualidade do atendimento médico [de Souza et al. 2020].

O artigo de [Lourenção and Junior 2016] analisa as percepções sobre a implantação do PEP no Brasil, destacando os progressos e dificuldades enfrentados nesse processo. A revisão bibliográfica revelou que, embora o uso da informática em saúde traga melhorias significativas na qualidade e organização das informações médicas, há dificuldades relativas à adaptação dos profissionais e à infraestrutura tecnológica.

As principais vantagens apontadas são a melhoria do acesso e da qualidade das informações, a otimização dos processos assistenciais e a redução dos custos com prontuários em papel. Os resultados demonstram que, apesar das dificuldades, o PEP é uma ferramenta relevante para aperfeiçoar a assistência à saúde no Brasil [Lourenção and Junior 2016].

O artigo de [Costa and Sabbatini 2016] analisa o desenvolvimento e a avaliação de um PEP baseado na internet, denominado PEPWeb. A pesquisa revela que uma parcela significativa dos projetos de PEP ainda não segue metodologias formais de engenharia e, consequentemente, poucos são coordenados por especialistas em Informática Médica.

O PEPWeb foi desenvolvido de acordo com uma metodologia estruturada e orientada a objetivos, demonstrando a complexidade e os desafios técnicos na implementação de sistemas de PEP. O sistema oferece funcionalidades como a acessibilidade pelo paciente, a comunicação segura entre médicos e pacientes, o que representa um grande avanço na gestão de dados de saúde [Costa and Sabbatini 2016].

3. Metodologia: Estudo de caso

A metodologia utilizada neste trabalho possui uma abordagem qualitativa centrada no levantamento e validação de requisitos, através de interações diretas com os principais usuários e responsáveis pela conformidade legal de um prontuário eletrônico. Como procedimento para essa abordagem, foi utilizado um estudo de caso com o desenvolvimento de um sistema de prontuário eletrônico para a clínica escola da Universidade Católica do Salvador.

Para este trabalho, a metodologia de estudo de caso foi escolhida porque é capaz de fornecer compreensão e contextualização do problema em estudo por meio de observações diretas, entrevistas sistemáticas e levantamentos detalhados em um ambiente real. Este método aumenta o envolvimento dos participantes, permite ajustes para atender às necessidades específicas da pesquisa e contribui significativamente para o conhecimento teórico e prático da disciplina [Bressan 2000].

Inicialmente foi feito um primeiro contato com a responsável pela Clínica Escola, a coordenadora da enfermagem, pelo qual foi verificado a forma como a clínica atua na gestão dos seus pacientes. Nesse contato, foram identificados diversos pontos que poderiam ser melhorados com a implantação de um sistema de gestão de prontuário para atendimentos clínicos na área de enfermagem.

Em entrevista com a coordenadora, foram definidos alguns pontos importantes como: facilidade de uso, escalabilidade para novos recursos, acesso ao código fonte, controle total dos dados e redução do controle quando for necessário

Diante do cenário, foi feita uma pesquisa dos sistemas relacionados com prontuário eletrônico, cujos detalhes estão descritos na seção 4.1 desse artigo, e, na sequência, foi feita uma análise desses insumos em sintonia com a responsável pela Clínica Escola, onde foi decidido desenvolver um sistema personalizado.

Após a definição do desenvolvimento do sistema, foi feito o levantamento de requisitos a partir da realização de reuniões estruturadas com a enfermeira responsável pela clínica escola. Estas reuniões foram essenciais para ampliar o entendimento sobre a rotina diária da clínica e as limitações do sistema de fichas manuais utilizadas para registro.

Em seguida, com o objetivo de facilitar a visualização das melhorias e mudanças propostas, foi desenvolvido um protótipo no Figma, para facilitar os testes de usabilidade, coletar *feedback* e validar o *layout*. Essa interação contínua permitiu ajustes e refinamentos essenciais, assegurando que o design do sistema estivesse alinhado com as necessidades do usuário final.

Paralelamente ao que foi citado no parágrafo anterior, foram conduzidas entrevistas com a pessoa responsável pela conformidade com a Lei Geral de Proteção de Dados (LGPD) na UCSAL, validando assim a utilização dos dados sensíveis a serem trabalhados no sistema. Essas entrevistas buscaram garantir que todos os aspectos do sistema fossem desenvolvidos em conformidade com os padrões legais de privacidade e segurança de dados. As discussões foram cruciais para delinear os requisitos de segurança e privacidade do sistema.

Após a coleta e compilação dos dados, procedeu-se a análise detalhada dos requisitos juntamente com todos os *stakeholders* envolvidos. Esta fase foi essencial para

validação do entendimento das necessidades, priorizando a segurança e a privacidade dos dados a fim de assegurar a conformidade com a legislação vigente e proteger adequadamente as informações sensíveis dos pacientes.

O resultado alcançado é um sistema de prontuário eletrônico que não apenas atenda às necessidades operacionais da clínica escola, mas também assegure a segurança e privacidade dos dados dos pacientes.

3.1. Levantamento dos Sistemas Relacionados a Prontuário Eletrônico

Durante a elaboração do trabalho, foram pesquisados e comparados sistemas de prontuários eletrônicos disponíveis no mercado. Esta pesquisa reuniu um total de oito sistemas: iClinic, MV, ConClínica, Amplimed, AmorSaúde, GNU Health, OpenMRS, *Open Clinic* pelos quais foram selecionados quatro (4) para um comparativo, conforme pode ser observado na Tabela 1.

Table 1. Tabela comparativa entre sistemas de prontuário eletrônico. Fonte: imagem própria

Funcionalidades	SPECEE UCSAL	Open MRS	OPEN CLINIC	GNU Health	AMORSAUDE
Marcação de Consultas	✗	✓	✓	✓	✓
Cadastro de paciente	✓	✓	✓	✓	✓
Acesso Remoto	✗	✓	✗	✗	✗
Histórico do paciente	✓	✓	✓	✓	✓
Registro de procedimento	✓	✓	✓	✓	✓
Segurança e Conformidade	✓	✓	✓	✓	✓
Cadastro de profissional	✓	✓	✓	✓	✓
Cadastro de Atendimentos	✓	✓	✓	✓	✓
Registro de atendimento	✓	✓	✓	✓	✓
Suporte a protocolo de saúde pública	✗	✗	✗	✓	✗

A escolha dos quatro (4) sistemas se deu pela maior simplicidade e por possuírem suporte às dez (10) principais funcionalidades destacadas como importantes pela clínica de enfermagem da UCSAL. Nesse comparativo, foram analisadas 10 funcionalidades sendo que o ”Suporte a protocolo de saúde pública” foi identificado como disponível apenas no sistema GNU *Health*.

Uma questão importante a ser destacada nessa análise é que todas as funcionalidades identificadas como não implementadas (não disponível), identificadas na cor ver-

melha, nessa versão do SPEP-UCSAL, foram incluídas como trabalho futuro, uma vez que não houve tempo hábil para implementá-las nessa primeira versão apresentada nesse TCC.

Os sistemas utilizados na comparação possuem, além das funcionalidades descritas na Tabela 1, outras que não foram consideradas necessárias para o cenário atual trazendo opções que podem causar certa confusão no seu uso. Essa situação representou um dos motivos considerados na decisão do desenvolvimento de uma solução personalizada em conjunto com os *stakeholders*.

3.2. Definição das tecnologias

Para o desenvolvimento do *backend*, optou-se pela pela linguagem Java e o *framework Spring Boot*. A escolha do Java se deve, além das questões técnicas descritas na seção 3.2, à sua familiaridade pelos desenvolvedores, uma vez que é a linguagem de base utilizada nos cursos da Escola de Tecnologias da UCSAL. Além disso, segundo [Junior et al. 2005] , o Java é excepcionalmente confiável, com destaque ao seu sistema de tratamento de exceções, que garante uma operação segura e estável, essencial para lidar com informações médicas sensíveis e evitar interrupções inesperadas.

A portabilidade do Java é outra característica relevante, permitindo que o software desenvolvido seja executado em diversas plataformas sem alterar o código, facilitando assim a utilização de testes e a implantação em diferentes ambientes computacionais disponíveis na infraestrutura da universidade.

O desenvolvimento com a linguagem Java foi complementado com a utilização do *framework Spring Boot*, que incorpora os benefícios do Java com as qualidades destacadas do estudo de [Dhalla 2021]. O *Spring Boot* tem como característica acelerar o processo de desenvolvimento graças à sua configuração automática e ao seu grande número de funcionalidades prontas para uso. Essas características favoreceram concentrar os esforços nos detalhes do sistema de prontuário eletrônico, sem perder tempo com configurações técnicas complexas.

O processo de construção do *front end* começou com a criação de um protótipo visual, desenvolvido com a ferramenta Figma, que foi validado em reuniões com os *stakeholders*. Após a conclusão do design, foram selecionadas e definidas as tecnologias a serem empregadas, visando assegurar uma experiência de usuário fluida e uma implementação robusta.

Para a criação dos componentes, foi utilizado o Angular CLI, que conforme [Galiani et al. 2023], é uma ferramenta adequada para apoiar a construção e testes de aplicações com Angular. O desenvolvimento foi trabalhado com HTML, CSS, Angular *Bootstrap* e *TypeScript*, sendo que o Angular é uma tecnologia que possui uma boa integração com o Java.

Os fundamentos do desenvolvimento web, HTML e CSS, permitem a estruturação e o estilo visual do conteúdo, essenciais para a criação de páginas web bem estruturadas e visualmente agradáveis [Miletto and de Castro Bertagnolli 2014]. Angular *Bootstrap* e *TypeScrip*t foram selecionados para desenvolver uma interface de usuário interativa e atraente para o sistema de prontuário eletrônico.

Essas escolhas tecnológicas foram fundamentais para garantir uma aplicação efi-

ciente, fácil de manter e capaz de proporcionar uma experiência de usuário de qualidade.

As características avançadas de *TypeScript*, como tipagem estática e suporte a objetos e classes, ajudam a evitar erros comuns de *JavaScript* durante a fase de desenvolvimento. Como os erros podem ser detectados durante a compilação, antes mesmo da execução do código, isso melhora significativamente a escalabilidade e a manutenção do código [NPM 2018].

A combinação dessas tecnologias definiu uma base sólida para criar uma interface de usuário não apenas funcional e responsiva, mas também escalável e fácil de manter. Isso aumenta a eficácia geral e a qualidade do sistema criado.

4. Desenvolvimento do SPECEE

O projeto foi conduzido seguindo os princípios ágeis do Scrum e do kanban, com reuniões diárias de 15 a 20 minutos para alinhamento das entregas parciais do desenvolvimento feitas a cada semana. As entregas foram validadas junto à responsável pela Clínica Escola e ao Orientador do projeto, além da validação feita por todos da equipe.

A Figura 1 representa uma visão estática da divisão do trabalho, feita na ferramentas Trello, utilizando *cards* em colunas que representam o status dos grupos de tarefas com base no kanban. O Trello foi a ferramenta escolhida para o gerenciamento do projeto devido a facilidade de adequação ao fluxo de trabalho da equipe, cujo acompanhamento pode ser feito por dispositivos móveis.

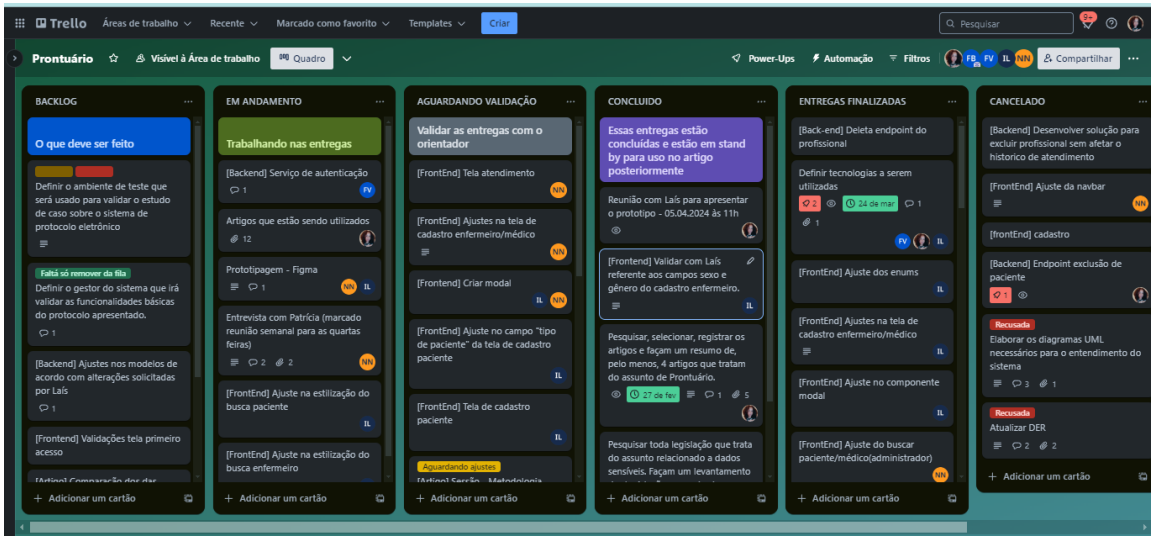

Figure 1. Tela do sistema Trello utilizando o template do kanban. Fonte: imagem própria

As colunas foram definidas para representar a lista de tarefas que deveriam ser elaboradas, com a respectiva priorização. Essas tarefas foram atribuídas a um ou mais membros da equipe, considerando a maior familiaridade na execução e balanceamento do tempo empregado na pesquisa por cada pessoa da equipe, a fim de evitar sobrecarga de trabalho.

O protótipo foi elaborado com base nos requisitos levantados, considerando a usabilidade do sistema de forma fácil e simples. O resultado pode ser observado nas Figuras

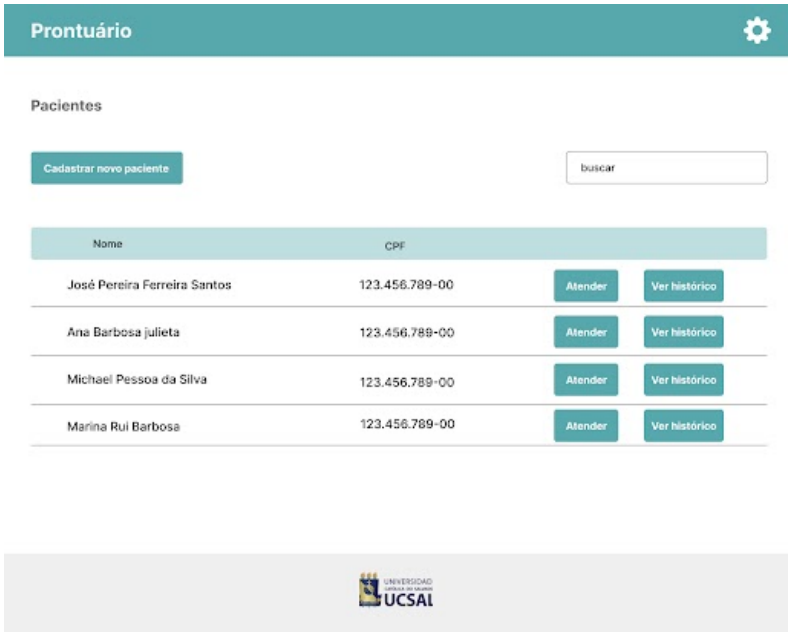

Figure 2. Tela do cadastro de paciente. Fonte: imagem própria

2 e 3.

Na tela mostrada na figura 2 o usuário terá uma lista de pacientes ordenados pela data de atendimento de forma crescente. Nesta tela pode-se efetuar uma busca por nome ou por CPF cadastrado. Essa lista é formada pelos pacientes atendidos pelo usuário do sistema, nesse caso a enfermeira responsável pela clínica.

Durante o desenvolvimento, foram concebidos componentes estáticos, tais como a barra de navegação e o rodapé. Esta abordagem nos permitiu a aplicação do conceito de *Single Page Application*, evitando o recarregamento da página inteira e, consequentemente, mitigando uma possível lentidão durante a navegação entre as páginas. Além disso, a adoção de componentes nos proporcionou a reutilização eficiente de código, minimizando a necessidade de reescrita em diferentes partes da aplicação.

Para a estilização, optamos por incorporar o Bootstrap ao projeto Angular. Essa escolha nos proporcionou acesso a uma ampla variedade de componentes de interface prontos para uso, tais como botões e formulários, contribuindo significativamente para a agilidade no desenvolvimento e garantindo uma estética consistente e profissional em toda a aplicação.

Apesar da robustez dos componentes oferecidos pelo Angular e pelo Bootstrap, a intervenção direta no HTML e SCSS foi essencial para personalizar e ajustar os componentes do Bootstrap, garantindo assim a conformidade do *layout* com o design estabelecido no Figma. Esta abordagem permitiu a adaptação precisa dos elementos visuais, garantindo uma harmonia entre a implementação técnica e as diretrizes de design.

Em resumo, a integração harmoniosa do Angular CLI, Bootstrap, HTML e SCSS proporcionou uma base sólida para o desenvolvimento de *Single Page Applications* (SPAs) de alta qualidade. Essas tecnologias simplificaram significativamente o processo de desenvolvimento. Ao aproveitar as capacidades dessas ferramentas e tecnologias, con-

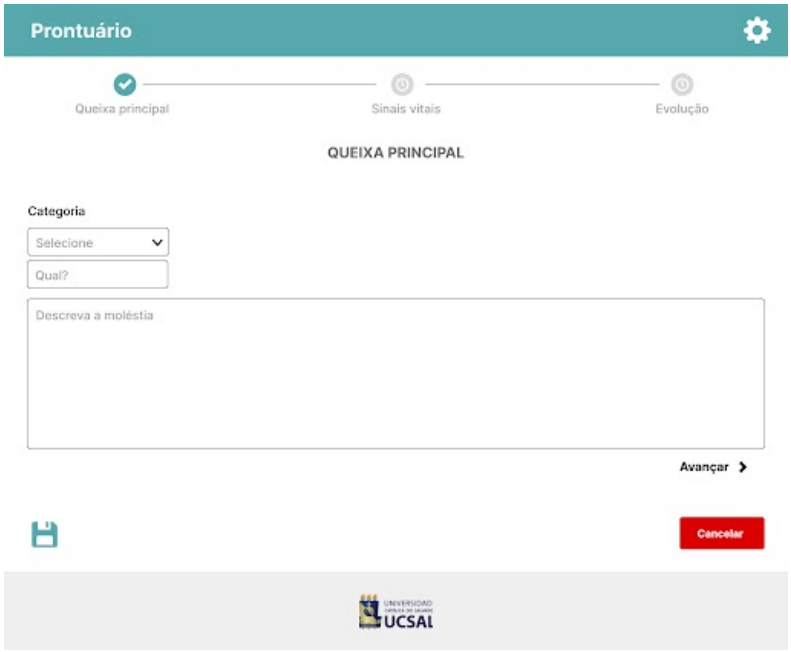

Figure 3. Tela para descrição dos sintomas paciente. Fonte: imagem própria

seguimos criar um prontuário eletrônico moderno e eficiente para atender às demandas da clínica da UCSAL atualmente.

O desenvolvimento do *back end* iniciou-se com a modelagem do banco de dados com o DER elaborado com base nos requisitos do sistema. Essa modelagem está representada na Figura 4.

O sistema concebido possui uma estrutura de camadas organizada e bem definida, no qual cada camada possui responsabilidades específicas que colaboram para a modularidade e a manutenção eficiente do código.

A camada útil abriga classes utilitárias que fornecem funcionalidades e recursos auxiliares para o restante do projeto. Essas classes desempenham um papel essencial ao oferecerem métodos e constantes reutilizáveis que facilitam diversas operações dentro do sistema.

Complementando essa estrutura, a camada model contém classes que representam as entidades do domínio e os modelos de dados do sistema. Essas classes são mapeadas para tabelas no banco de dados e são manipuladas pelo ORM (*Object-Relational Mapping*), permitindo uma interação orientada a objetos com o banco de dados.

Já a camada *service* se destaca ao conter classes que implementam a lógica de negócios da aplicação. Essas classes orquestram operações, aplicam regras de negócios e interagem com os repositórios para a realização das operações de persistência de dados. Elas atuam como intermediárias entre os controladores (*controllers*) e os repositórios (*repositories*), encapsulando a lógica de negócios e promovendo a separação de responsabilidades.

Para garantir uma comunicação fluida entre as diferentes camadas do sistema, a pasta mapper é crucial. Ela contém classes e interfaces que realizam a conversão entre

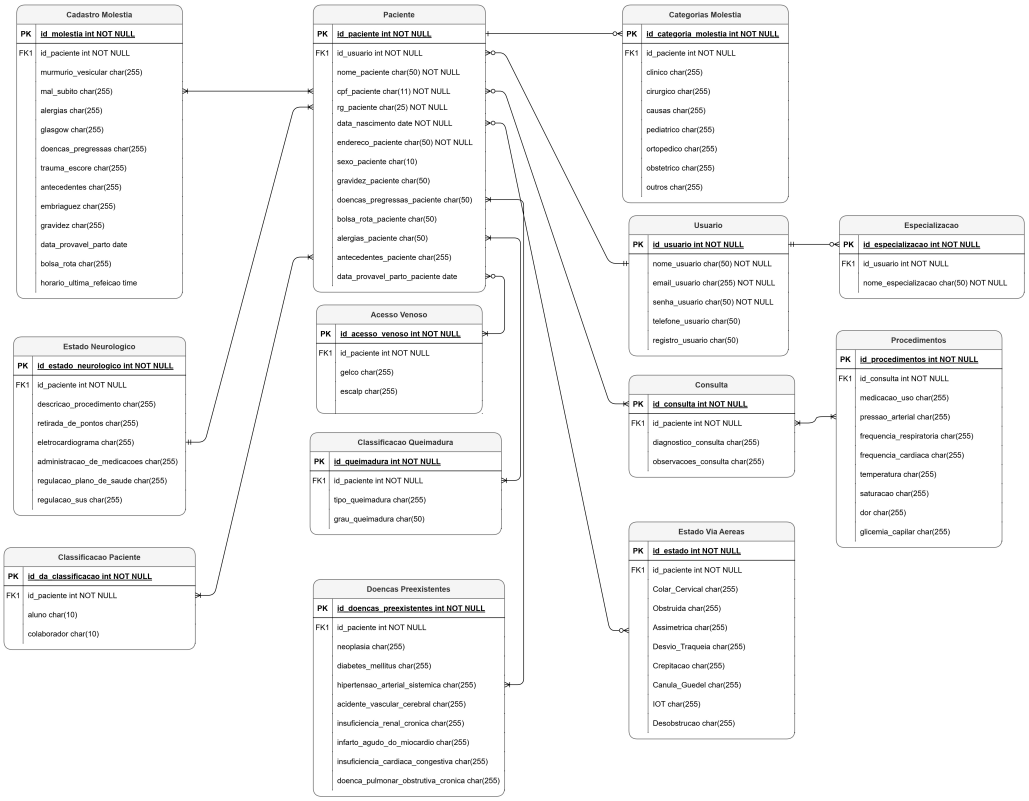

Figure 4. Tela do DER. Fonte: imagem própria

entidades do domínio e DTOs (*Data Transfer Objects*). Esse processo de mapeamento facilita a transferência de dados entre as camadas da aplicação, assegurando que a lógica de negócios e a representação dos dados sejam tratadas de forma independente.

O ponto de contato inicial com o sistema ocorre através da camada *controller*, que contém classes responsáveis por gerenciar as requisições HTTP recebidas pelo servidor. Esses controladores expõem *endpoints RESTful*, processam a entrada do usuário, chamam a camada de serviço e retornam as respostas HTTP correspondentes. Assim, eles atuam como a interface entre o *frontend* (ou outros consumidores de API) e a lógica de negócios da aplicação.

Para amarrar todas essas camadas e assegurar que o sistema funcione de maneira integrada e eficiente, a camada configuration desempenha um papel vital. Ela contém classes que definem configurações e *beans* do *Spring*, utilizados para configurar o comportamento da aplicação, registrar beans no contexto do *Spring* e habilitar ou customizar funcionalidades do *framework*. Essa configuração centralizada promove a modularidade e facilita a gestão de dependências e funcionalidades transversais.

5. Conclusões

O objetivo principal deste estudo foi desenvolver um Sistema de Prontuário Eletrônico do Paciente (SPEP) voltado para a Clínica Escola de Enfermagem da Universidade Católica do Salvador (UCSAL), com o objetivo de tornar as informações disponíveis para os profissionais de saúde, otimizar o atendimento e aumentar a segurança e privacidade dos dados dos pacientes. A implantação deste sistema teve como objetivo proporcionar uma gestão

mais eficiente dos registros médicos, de acordo com as necessidades específicas da clínica e de acordo com a Lei Geral de Proteção de Dados (LGPD).

Para atingir esses objetivos, foi realizada uma análise das necessidades da clínica, seguida pelo desenvolvimento de um protótipo com ferramentas como Figma para validação de usabilidade. Em seguida, o sistema foi desenvolvido com as tecnologias Angular para o *front-end, Spring Boo*t para o *back-end* e PostgreSQL para o banco de dados. O desenvolvimento seguiu metodologias ágeis, com reuniões para alinhamento e validação contínua das entregas junto aos públicos-alvo.

Os resultados da pesquisa demonstram que a implementação do SPECEE promoverá melhorias na eficiência operacional da clínica, permitindo um acesso mais rápido e seguro às informações dos pacientes. A integração das tecnologias escolhidas permitiu uma interface intuitiva. A conformidade com a LGPD assegura a proteção e privacidade dos dados sensíveis dos pacientes.

Os benefícios do desenvolvimento do SPECEE incluem a melhoria da acessibilidade e qualidade das informações, a otimização dos processos assistenciais e a redução dos custos associados ao uso de prontuários em papel.

A implantação do SPECEE na Clínica Escola de Enfermagem da UCSAL representa um avanço na modernização dos serviços de saúde, uma vez que aumenta a eficiência, a segurança e a qualidade do atendimento aos pacientes.

5.1. Trabalhos Futuros

O projeto do Sistema de Prontuário Eletrônico do Paciente (SPECEE) da Clínica Escola de Enfermagem da Universidade Católica do Salvador (UCSAL) prevê melhorias e expansões com o objetivo de aperfeiçoar ainda mais a eficiência e a qualidade dos serviços oferecidos. Dentre as principais ações, está a implementação da funcionalidade de agendamento de consultas. Esta funcionalidade facilitará o atendimento, reduzindo o tempo de espera e aprimorando a gestão de horários na clínica.

Além de agendar consultas, o SPECEE deve ser integrado a outros sistemas de saúde, permitindo a interoperabilidade entre as diferentes plataformas. Essa integração permitirá uma troca de dados bem mais eficiente e segura entre os diversos sistemas utilizados em diferentes instituições de saúde, contribuindo para um atendimento mais integrado e completo.

Em resumo, os projetos futuros para o SPECEE são fundamentais para que o sistema continue a evoluir e a atender às crescentes demandas da Clínica Escola de Enfermagem da UCSAL, promovendo a satisfação dos seus usuários.

Referências Bibliográficas

Almeida, M. J. G. G. et al. (2016). Discussão Ética sobre o prontuário eletrônico do paciente. *Revista Brasileira de Educação Médica*, 40(3):521–527.

Bezerra, S. M. (2015). Prontuário eletrônico do paciente: uma ferramenta para aprimorar a qualidade dos serviços de saúde. Acesso em: 14 jun. 2024.

Bierman, G., Abadi, M., and Torgersen, M. (2014). Understanding typescript. Disponível em: https://users.soe.ucsc.edu/ abadi/Papers/FTS-submitted.pdf.

Boaglio, F. (2017). *Spring Boot: Acelere o desenvolvimento de microsserviços*. Casa do Código.

Bressan, F. (2000). O método do estudo de caso. *Administração on line*, 1(1):1–13.

Bueno, J. P. R. (2018). Plataforma em nuvem para gestão de dispositivos de iot. Disponível em: https://cepein.femanet.com.br/BDigital/arqTccs/1411320144.pdf.

CFM, E. et al. Conselho federal de medicina. Disponível em: https://portal.cfm.org.br. Acesso em: 30 mar. 2024.

Claro, D. B. and Sobral, J. B. M. (2008). *Programação em JAVA*. Copyleft Pearson Education, Florianópolis.

Costa, C. and Sabbatini, R. (2016). Desenvolvimento e avaliação de prontuário eletrônico do paciente. Acesso em: 15 jun. 2024.

da Rocha, E. L., Jaraceski, J. C., Charnovscki, R., do Nascimento Filho, J. R., Nicoleit, E. R., de Azevedo Simões, P. W. T., and de Mattos, M. C. (2024). Prontuário eletrônico para acompanhamento médico dos pacientes de uma unidade de terapia intensiva. Acessado: 14-junho-2024.

da Silva, A. R. F. (1996). Introdução à programação com java. Acessado: 14-junho-2024.

da Silva, C. R. (2021). História do prontuário médico: Evolução do prontuário médico tradicional ao prontuário eletrônico do paciente – pep. *Research, Society and Development*, 10(9):e28510918031.

de Souza, A. G., Lombardi, L. V., and da Silva Santos, L. F. (2020). Sistema de prontuário eletrônico e a disponibilidade de suas informações. Acesso em: 15 jun. 2024.

Dhalla, H. K. (2021). A performance comparison of restful applications implemented in spring boot java and ms.net core. *J. Phys.: Conf. Ser.*, 1933:012041.

Figueiredo, J. and Oliveira, D. (2024). Gestão de tecnologias da informação e da comunicação na saúde: uma análise sobre o uso do prontuário eletrônico. Disponível em: https://felipe.pusanovsky.nom.br/files/GC/6M.pdf.

Galiani, D. H., Dallilo, F. D., and Florian, F. (2023). Desenvolvimento de uma aplicação pwa para adoção de animais. Disponível em: https://semanaacademica.org.br/system/files/artigos/128_desenvolvimento_de_uma_aplicacao_pwa_para_adocao_de_animais_1.pdf Acesso em: 14 de junho de 2024.

Governo Brasileiro (2018). A implementação da lei geral de proteção de dados no exercício profissional na Área da saúde. Disponível em: https://www.gov.br/esporte/pt-br/acesso-a-informacao/lgpd Acesso: 14 de junho de 2024.

Hawryliszyn, L. O., Coelho, N. G. S. C., and Barja, P. R. (2021). Lei geral de proteÇÃo de dados (lgpd): O desafio de sua implantaÇÃo para a saÚde. *Revista Univap*, 27(54).

Jr., N. N. L. and Mercado, N. B. G. (2018). Vantagens e desvantagens da utilização do ionic framework para o desenvolvimento de aplicativos móveis.

Disponível em: https://semanaacademica.com.br/system/files/artigos/artigo_74.pdf.

Junior, M. A. C., Virtuoso, G. H., and Martins, P. J. (2005). Propriedades desejáveis a uma linguagem de programação: Uma análise comparativa entre as linguagens c, c++ e java. volume 1. Anais SULCOMP.

Lourenção, L. G. and Junior, C. J. F. (2016). Implantação do prontuário eletrônico do paciente no brasil. Acesso em: 15 jun. 2024.

Luciano, J. and Alves, W. J. B. (2011). Padrão de arquitetura mvc: Model-view-controller. *Revista EPeQ Fafibe*, 1(3):102–107.

Magnagnagno, O., Luciano, E., and Lübeck, R. (sd). Como proteger informações do prontuário eletrônico do paciente: proposta de mecanismos. Disponível em: https://repositorio.pucrs.br/dspace/bitstream/10923/18380/2/Como_proteger_informaes_do_pronturio_eletrnico_do_paciente_proposta_de_mecanismos.pdf.

Miletto, E. M. and de Castro Bertagnolli, S. (2014). *Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JAVASCRIPT e PHP*. Bookman.

Mozilla Developer Network. Segurança em aplicação web. https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/First_steps/Website_security. Acesso em: 20 jun. 2024.

NPM (2018). Typescript. https://www.npmjs.com/package/typescript. Acesso em: 20 jun. 2024.

OWASP (2024). Owasp top ten. https://owasp.org/www-project-top-ten/. Acessado: 14-junho-2024.

Rabello, L. M. (2009). Programa em linguagem java para comunicação serial. Disponível em: https://www.infoteca.cnptia.embrapa.br/bitstream/doc/663550/1/CT1092009.pdf. Acesso em: 14 jun. 2024.

Santiago, C. P., de Lima Veras, N., de Aragão, A. P., Carvalho, D. A., and Amaral, L. A. (2020). Desenvolvimento de sistemas web orientado a reuso com python, django e bootstrap. PDF.

Santos, R. d. C. and Mendes, L. A. M. (s. d.). Estudo comparativo dos sistemas gerenciadores de bancos de dados: Oracle, sql server e postgresql. Disponível em: https://ri.unipac.br/repositorio/wp-content/uploads/2019/08/Rodrigo.pdf. Acesso em: 14 jun. 2024.

Silva, F. G. and Soares, M. S. (2016). Modelagem de processos de negócios usando bpmn para desenvolvimento de um prontuário eletrônico de pacientes. *Journal of Health Informatics*, 8.

Telles, E. T. G., de Oliveira Rodrigues Maruco, F., and da Silva, V. D. S. T. (2021). A implementação da lei geral de proteção de dados no exercício profissional na Área da saúde. Acessado: 14-junho-2024.

Thofehrn, C. and Lima, W. C. d. (2006). Prontuário eletrônico do paciente – a importância da clareza da informação. *Revista Eletrônica de Sistemas de Informação*, 5(1).