



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
CURSO TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

EZEQUIEL TEIXEIRA COSTA

GESTOR PRONTUÁRIO MOBILE

TERESINA, PIAUÍ
2023

EZEQUIEL TEIXEIRA COSTA

GESTOR PRONTUÁRIO MOBILE

Trabalho de Conclusão de Curso (monografia) apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Orientador: Prof. M^e. Ely da Silva Miranda

TERESINA, PIAUÍ
2023

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Costa, Ezequiel Teixeira

C837g Gestor prontuário mobile / Ezequiel Teixeira Costa. - 2023.
75 f.: il. color.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central, 2023.

Orientador : Prof Me. Ely da Silva Miranda.

1. Prontuário eletrônico do paciente. 2. Aplicativo móvel. 3. Sinais vitais. 4. Saúde digital. I. Título.

CDD - 004.21

Elaborado por Tanize Maria Sales CRB 3/1024

EZEQUIEL TEIXEIRA COSTA

GESTOR PRONTUÁRIO MOBILE

Trabalho de Conclusão de Curso (monografia) apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Aprovada em 13 de dezembro de 2023.

BANCA EXAMINADORA:

Prof. M^e. Ely da Silva Miranda(Orientador)
Instituto Federal do Piauí (IFPI)

Prof. M^e. Rogério da Silva
Instituto Federal do Piauí (IFPI)

Prof. M^e. Fernando Castelo Branco Gonçalves Santana
Instituto Federal do Piauí (IFPI)

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me dado a vida e ter me permitido chegar até aqui. Porque Nele vivemos, e nos movemos e existimos.

Agradeço a empresa Supra Tecnologia porque ter me confiado a execução do projeto Gestor Prontuário Mobile, assim como ter me oferecido todo suporte para a execução deste trabalho, e também por toda a experiência profissional e pessoal a mim transmitida.

Agradeço aos professores Rogério da Silva e Fernando Castelo Branco Gonçalves Santana por aceitarem participar da banca examinadora.

Agradeço ao professor Ely da Silva Miranda, por ter aceitado, de prontidão, orientar-me na execução deste trabalho, dedicando do seu tempo para discussões regulares, oferecendo conselhos e *feedbacks* construtivos, *insights* valiosos e compreendendo minhas dificuldades e me ajudando a superá-las.

Agradeço aos meus amigos pelos vários momentos de descontração, longas conversas, aconselhamento e incentivos, no qual tornaram o processo de desenvolvimento desse projeto mais leve e ajudaram-me a manter a calma e seguir em frente.

Finalmente, agradeço a todos os professores do Instituto Federal do Piauí os quais tive o prazer de conhecer. Pelo partilhar de seus conhecimentos e experiências e também por impulsionar os discentes a evoluir profissionalmente.

RESUMO

O gerenciamento de prontuários médicos físicos apresenta desafios significativos relacionados ao volume de armazenamento, degradação do papel e dificuldades de acesso, especialmente em situações emergenciais. O sistema web “Gestor Saúde”, embora possua módulos de Prontuário Eletrônico do Paciente (PEP), não oferece design responsivo, comprometendo a visualização de dados por profissionais que utilizam dispositivos móveis. Além disso, o registro de Sinais Vitais (SSVV) ocorre inicialmente em fichas impressas para posterior inserção no sistema, causando perda do histórico de dados. Este trabalho teve como objetivo proporcionar aos profissionais de saúde um ambiente melhor para acesso aos dados do PEP do Gestor Saúde, tornando a visualização mais fluída e organizada, além de aperfeiçoar a assistência ao paciente. A metodologia empregada foi o desenvolvimento ágil Scrum, com sprints de uma semana, utilizando React Native para desenvolvimento multiplataforma, API REST para comunicação back-end, PostgreSQL como banco de dados e JWT para autenticação. O aplicativo “Gestor Prontuário Mobile” foi desenvolvido para médicos, enfermeiros e técnicos de enfermagem, oferecendo funcionalidades como visualização do histórico médico, registro direto de sinais vitais com visualização gráfica, solicitações de visita entre profissionais, e acesso aos históricos de enfermagem e multidisciplinar. Os principais resultados incluem a eliminação da necessidade de fichas físicas para SSVV, interface intuitiva com design responsivo, e facilidade de comunicação entre profissionais de saúde. O aplicativo resolve o problema de mobilidade dos profissionais de saúde, agilizando a visualização do histórico do paciente e o registro de dados essenciais, contribuindo para melhor análise clínica e diagnóstico médico.

Palavras-chaves: prontuário eletrônico do paciente; aplicativo móvel; sinais vitais; saúde digital.

ABSTRACT

The management of physical medical records presents significant challenges related to storage volume, paper degradation, and access difficulties, especially in emergency situations. The web system “Gestor Saúde”, although it has Electronic Health Record (EHR) modules, does not offer responsive design, compromising data visualization for professionals who use mobile devices. Additionally, Vital Signs (VS) registration occurs initially on printed forms for subsequent insertion into the system, causing loss of data history. This work aimed to provide healthcare professionals with a better environment for accessing EHR data from Gestor Saúde, making visualization more fluid and organized, while improving patient care. The methodology employed was agile Scrum development, with one-week sprints, using React Native for cross-platform development, REST API for back-end communication, PostgreSQL as database, and JWT for authentication. The “Gestor Prontuário Mobile” application was developed for physicians, nurses, and nursing technicians, offering functionalities such as medical history visualization, direct vital signs registration with graphical visualization, visit requests between professionals, and access to nursing and multidisciplinary histories. The main results include eliminating the need for physical forms for VS, intuitive interface with responsive design, and ease of communication between healthcare professionals. The application solves the mobility problem of healthcare professionals, streamlining patient history visualization and essential data recording, contributing to better clinical analysis and medical diagnosis.

Key-words: electronic health record; mobile application; vital signs; digital health.

LISTA DE ILUSTRAÇÕES

Figura 1 – Prontuários Físicos	13
Figura 2 – Arquitetura Legado do <i>React Native</i>	25
Figura 3 – Arquitetura <i>Fabric</i>	26
Figura 4 – Estrutura de um Token JWT	27
Figura 5 – Ciclo de uma <i>Sprint</i>	28
Figura 6 – Casos de Uso	31
Figura 7 – Diagrama de Classes Simplificado do Gestor Saúde	34
Figura 8 – Arquitetura do Gestor Prontuário Mobile	35
Figura 9 – <i>Splash Screen</i>	42
Figura 10 – Tela de Login	43
Figura 11 – Tela da Lista de Estabelecimentos Vinculados à Conta do Usuário	43
Figura 12 – Janela para Seleção do Posto, Setor ou Especialidade	44
Figura 13 – Tela Inicial do Aplicativo	45
Figura 14 – Menu Lateral do Aplicativo	45
Figura 15 – Listagem dos Pacientes Internados	46
Figura 16 – Tela de Listagem das Solicitações de Visita	47
Figura 17 – Tela contendo o Gráfico de Temperatura Axilar	47
Figura 18 – Tela para a Solicitação de Visitas	48
Figura 19 – Listagem de CBOs	48
Figura 20 – Formulário de Adição dos Sinais Vitais	49
Figura 21 – <i>Cards</i> representando a Solicitação de Parecer e a Evolução Médica	50
Figura 22 – <i>Card</i> representando a Solicitação de Procedimentos	50
Figura 23 – <i>Card</i> representando a Prescrição Médica	51
Figura 24 – <i>Card</i> representando a Evolução de Enfermagem	52
Figura 25 – Tela contendo a Seção Cardiovascular/Respiratório e Neurológico	52
Figura 26 – Seções Epiderme, Nutrição, GastroIntestinal/Urinarío e Outros	53
Figura 27 – Seções Diagnóstico e Sinais Vitais	53
Figura 28 – Evolução Multidisciplinar	54

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CBO	Classificação Brasileira de Ocupações
CEP	Código de Endereçamento Postal
CFM	Conselho Federal de Medicina
CRUD	Create-Read-Update-Delete
ECDSA	Elliptic Curve Digital Signature Algorithm
FGV	Fundação Getúlio Vargas
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
IA	Inteligência Artificial
IMIA	International Medical Informatics Association
IANA	Internet Assigned Numbers Authority
IOM	Institute of Medicine
JSI	JavaScript Interface
JSON	JavaScript Object Notation
JWT	JSON Web Token
LGPD	Lei Geral de Proteção de Dados Pessoais
MVC	Model-View-Controller
PCR	Parada Cardiorespiratória
PHP	Hypertext Preprocessor
PEP	Prontuário Eletrônico do Paciente
PO	Product Owner
RFC	Request for Comments
REST	Representational State Transfer

RSA	Rivest-Shamir-Adleman
SAE	Sistematização da Assistência de Enfermagem
SAME	Serviço de Arquivo Médico
SESA	Secretaria de Estado da Saúde do Espírito Santo
SUS	Sistema Único de Saúde
SSVV	Sinais Vitais
SO	Sistema Operacional
TAX	Temperatura Axilar
UML	Unified Model Language
XML	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	11
1.1	CONTEXTUALIZAÇÃO	11
1.2	JUSTIFICATIVA	14
1.3	OBJETIVOS	16
1.3.1	Objetivo Geral	16
1.3.2	Objetivos Específicos	16
1.4	ORGANIZAÇÃO DO PROJETO	17
1.5	RESUMO	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	PRONTUÁRIO MÉDICO	18
2.2	PRONTUÁRIO ELETRÔNICO DO PACIENTE	20
2.3	SEGURANÇA E PRIVACIDADE DE ACORDO COM A LGPD	21
2.4	DESENVOLVIMENTO MULTIPLATAFORMA	23
2.5	REACT NATIVE	23
2.5.1	JSON Web Token	26
3	METODOLOGIA	28
4	GESTOR PRONTUÁRIO MOBILE	30
4.1	CASOS DE USO	30
4.2	CARACTERÍSTICAS DA IMPLEMENTAÇÃO	32
4.3	ORGANIZAÇÃO DO PROJETO	32
4.4	ARQUITETURA	34
4.5	API REST	36
4.6	APRESENTAÇÃO DA APLICAÇÃO	42
5	CONCLUSÃO	55
5.1	TRABALHOS FUTUROS	55
	REFERÊNCIAS	57
	ANEXO A – ESPECIFICAÇÃO DOS CASOS DE USO	61
	ANEXO B – DOCUMENTAÇÃO DA API REST	67

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

O Prontuário do Paciente é um elemento crucial no atendimento à saúde dos indivíduos, devendo reunir as informações necessárias para garantir a continuidade dos tratamentos prestados ao paciente (Marin, Massad e Neto, 2003). Nele, deve conter todo o histórico de saúde do paciente desde o nascimento até sua morte (Silva, 2021). Dentre as informações contidas no prontuário, podem-se citar prescrições médicas, sinais vitais, resultados de exames laboratoriais, receituário, evoluções e tantos outros dados que informam o estado do paciente durante sua estadia no hospital.

O prontuário deve ser a fonte primária e genuína de informações. Para garantir isso, alguns pontos devem ser considerados, tais como ser escrito em linguagem compreensível e clara, evitar ambiguidades, ser mantido em segurança e garantir o princípio da disponibilidade, ou seja, a informação sempre deve estar disponível, independente de intempéries (Brasil, 2011).

As origens do prontuário remontam ao século V a.C, com Hipócrates, o qual instruía médicos a realizarem o registro escrito com os propósitos de refletir de forma exata o curso da doença e indicar as possíveis causas delas. Ao longo de décadas, vários profissionais contribuíram para a formação e padronização do Prontuário do Paciente. Florence Nightingale (1820-1910), precursora da Enfermagem Moderna, proferiu sua clássica frase quando tratava feridos na Guerra da Criméia (1853-1856):

Na tentativa de chegar à verdade, eu tenho buscado, em todos os locais, informações; mas, em raras ocasiões eu tenho obtido os registros hospitalares possíveis de serem usados para comparações. Estes registros poderiam nos mostrar como o dinheiro tem sido usado, o quê de bom foi realmente feito com ele (Marin, Massad e Neto, 2003).

Logo após, em 1880, William Mayo, que formou a Clínica Mayo com um grupo de colegas, observou que a maioria dos médicos tinha dificuldades em manipular os registros médicos dos pacientes, uma vez que ficavam em um só documento. Foi então que surgiu a ideia de separar o documento em registros individuais para cada paciente, dando início ao prontuário orientado ao paciente e de forma cronológica. Em 1920, a mesma clínica desejou estabelecer uma padronização ao documento, sugerindo um conjunto mínimo de informações a serem registradas (Marin, Massad e Neto, 2003).

Ao longo do século XX, inúmeras conferências e estudos foram realizados com o intuito de aperfeiçoar o Prontuário do Paciente. A primeira conferência da *International Medical Informatics Association* (IMIA), em 1979, buscou estabelecer aspectos fundamentais a serem considerados para os sistemas clínicos. Posteriormente, no início dos anos 90, um estudo realizado pelo *Institute of Medicine* (IOM) dos Estados

Unidos, buscou definir o Prontuário Eletrônico do Paciente (PEP) e também estabelecer melhorias em função das evoluções tecnológicas, embora na década de 70, o Prontuário Eletrônico já fosse uma realidade (Silva, 2021).

Como resultado de anos de aperfeiçoamento, o Prontuário do Paciente tornou-se um dos mais importantes documentos médicos, sendo também imprescindível para a sociedade. A Resolução CFM nº 1.638/2002, destaca que o prontuário é importante para o paciente, para o profissional que o assiste e para as instituições de saúde, bem como para o ensino, a pesquisa e os serviços públicos de saúde, além de ser instrumento de defesa legal (Conselho Federal de Medicina, 2002). Marin, Massad e Neto (2003) afirmam que:

O sistema de saúde de um país, é estabelecido graças ao que se tem documentado em um prontuário, uma vez que dele são extraídas as informações sobre a saúde dos indivíduos que formam uma comunidade e uma nação (Marin, Massad e Neto, 2003).

O conteúdo do Prontuário do Paciente é composto por inúmeras informações heterogêneas, mas que se complementam. Isso se dá devido ao caráter multidisciplinar que o documento possui. Profissionais como médicos, enfermeiros, fonoaudiólogos, psicólogos, nutricionistas entre outros, podem contribuir com o histórico do paciente, tornando o prontuário mais diversificado e rico em informações, beneficiando o paciente. Entretanto, o volume de informação pode tornar-se um problema para os hospitais, considerando que a vida útil de preservação dos dados é de 20 anos (Brasil, 2018b). Além disso, a preservação das informações também é um desafio, visto que o modelo tradicional utiliza papel, que se desgasta naturalmente com o tempo. A Figura 1 apresenta uma sala pertencente ao setor Serviço de Arquivo Médico (SAME) do Hospital Estadual Dr. Jayme Santos Neves, na Serra, Espírito Santo, o qual contabiliza mais de 97 mil prontuários e um valor superior a 220 mil atendimentos, todos em papel (Secretaria de Estado da Saúde do Espírito Santo, 2014). Diante disso, surge a necessidade de soluções que organizem e estruturem os dados de forma a resolver esse problema.

Felizmente, o avanço da tecnologia da informação tem representado alternativas promissoras para a redução dos custos e para a melhoria dos serviços de saúde (Negri, 2020). O prontuário impresso do paciente, vem gradualmente sendo substituído pelo Prontuário Eletrônico do Paciente (PEP), uma alternativa mais segura e que busca resolver os problemas que o prontuário físico possui, como o grande volume de papel impresso e o armazenamento em espaços físicos. Esses problemas podem comprometer a integridade e confidencialidade dos dados. Com o PEP, as informações são armazenadas em bancos de dados remotos, ocupando menos espaço e permitindo a mobilidade e *backup* dos dados.

Figura 1 – Prontuários Físicos



Fonte: Secretaria de Estado da Saúde do Espírito Santo (2014).

Diante das vantagens do PEP, o presente trabalho busca desenvolver um aplicativo para dispositivos móveis, que facilite aos profissionais de saúde visualizarem, com melhor qualidade, o Prontuário do Paciente. O aplicativo consumirá os dados vindos do software Gestor Saúde, um sistema Web, proprietário da empresa Supra Tecnologia, que oferece serviços de gestão voltados ao Sistema Único de Saúde (SUS). O Gestor Saúde possui módulos de PEP que são utilizados pelos profissionais de saúde no tratamento do paciente durante sua passagem pelo o hospital. Entretanto, o Gestor Saúde não oferece *design* responsivo, o qual consiste em adaptar o leiaute do sistema a diversos dispositivos, telas e resoluções, garantindo a boa experiência do usuário, fluidez na navegação e leitura confortável, sem comprometer o conteúdo (Silva, 2014). Devido a essa limitação, profissionais que visitam leitos diariamente e utilizam o Gestor Saúde para conferir informações do Prontuário do Paciente, possuem a visualização desses dados comprometida.

Além disso, identificou-se uma demanda em relação à anotação dos sinais vitais pelos profissionais de saúde, geralmente técnicos de enfermagem. A Sistematização da Assistência de Enfermagem (SAE) no software Gestor Saúde apresenta algumas deficiências na coleta dos sinais vitais. Devido à frequência dessa coleta e à localização fixa do computador, os sinais são inicialmente registrados em uma ficha impressa para, posteriormente, serem inseridos no sistema. No entanto, quando os profissionais adicionam os dados ao sistema, geralmente é registrado apenas um resumo representando a média dos dados coletados, a fim de evitar a inclusão de todas as informações no sistema. Isso, no entanto, pode prejudicar o tratamento do indivíduo, uma vez que o histórico dos dados é perdido, impossibilitando a análise da evolução do paciente. Diante desse cenário, a solução proposta busca também resolver essa problemática, adicionando um formulário para o registro dos sinais vitais do paciente.

Após a coleta, esses dados poderão ser visualizados em formato de gráficos, auxiliando os profissionais na percepção da evolução do quadro dos pacientes.

O aplicativo acessa os dados do Gestor Saúde através de chamadas à *API REST*¹, implementadas, pelo o autor, no projeto do software Gestor Saúde. As requisições *REST* acessam à base de dados contendo as informações do PEP dos paciente internados nos estabelecimentos onde o Gestor Saúde atual, por isso, os dados apresentados no aplicativo estão restritos à essa condição, dessa forma, os dados dos pacientes presentes em estabelecimentos privados ou alheios ao uso do software, não serão alcançados pelo aplicativo.

1.2 JUSTIFICATIVA

De acordo com StatCounter (2023), o uso de dispositivos móveis (*smartphones* e *tablets*) ultrapassou o uso de computadores *desktop* entre março de 2022 e março de 2023 em todo o mundo (StatCounter, 2023). Os benefícios que a tecnologia móvel proporciona são inúmeros, dos quais podem-se citar:

- **O acesso à informação:** Os dispositivos móveis permitem obter informação a qualquer hora e em qualquer lugar, viabilizando às pessoas o acesso imediato à informação de que precisam para tomar decisões;
- **Comunicação:** Diversas pessoas podem compartilhar e consumir informações, seja no âmbito profissional ou pessoal, através de chamadas de voz, mensagens de texto, aplicativos de mensagem instantânea, e-mail, entre outros;
- **Mobilidade:** A tecnologia móvel proporciona ao indivíduo a realização de tarefas enquanto se está em movimento, como acessar um aplicativo bancário, realizar compras online, solicitar uma viagem com motoristas de aplicativo e etc.

De fato, a demanda por serviços Web outrora disponíveis apenas para computadores com resoluções semelhantes, abrem espaço para dispositivos com uma enorme variedade de tamanhos e gêneros de telas (Prostt, 2013). Consequentemente, os aplicativos vêm se tornando mais especializados e completos, dispensando muitas vezes o uso de computadores pessoais para a realização de determinadas tarefas ou complementando tarefas realizadas por eles. Um recurso presente em um sistema Web pode ser estendido para um aplicativo, a fim de atender a um público e mercado ainda mais amplo e facilitar o acesso ao recurso. Isso ocorre porque os *smartphones* estão sempre disponíveis, enquanto os computadores podem não estar ligados o tempo todo. Como resultado, o uso de um aplicativo pode tornar o acesso mais conveniente para os usuários.

¹ Uma interface de programação de aplicações (API) em conformidade com as restrições do estilo de arquitetura *REST*, permitindo a interação com serviços web (Red Hat, 2023)

Visto que o custo de dispositivos móveis é geralmente mais baixo se comparado ao de computadores, o resultado foi a adoção e popularização do *smartphone* pela sociedade. De acordo com o Panorama do Uso de TI no Brasil, edição 2020, elaborado pela Fundação Getúlio Vargas (FGV), a população brasileira possui mais de um *smartphone* por habitante (Fundação Getúlio Vargas, 2022). De fato, o uso de dispositivos móveis tem-se intensificado nos últimos anos, e sistemas mais antigos, que foram construídos décadas atrás, enfrentam desafios para adequar-se às diversas resoluções de tela hoje existentes.

No ambiente de internação hospitalar, os sinais vitais (SSVV) desempenham um papel crucial como indicadores precoces, revelando o grau de melhora ou deterioração no estado dos pacientes hospitalizados, sendo que a análise deles deve ser correta, pois reflete a conduta decisória da equipe de enfermagem (Oliveira *et al.*, 2022). A coleta dos SSVV é uma atividade periódica no hospital e de extrema importância, visto que determina o estado de saúde do indivíduo e a evolução do quadro clínico (Souza *et al.*, 2019). Através do monitoramento diário dos sinais vitais, profissionais de saúde conseguem prever a ocorrência de eventos diversos, dentre eles a parada cardiorrespiratória (PCR). De acordo com Souza *et al.* (2019), raramente uma PCR é um evento inesperado. Geralmente, é precedida de sinais de piora clínica, como progressiva deterioração da função respiratória e cardiovascular (Souza *et al.*, 2019).

Considerando a importância de permanecer constantemente atento às alterações nos sinais vitais dos pacientes, tornar-se imprescindível que os SSVV estejam posicionados de maneira que facilite sua visualização pelo profissional e também permita uma análise rápida. Dessa forma, o uso de gráficos para representar dados tão importantes como os sinais vitais, é uma excelente maneira de informar sobre um determinado assunto de forma rápida e dinâmica, ocupando destaque nos meios de comunicação escrita e falada. Além disso, a utilização de gráficos simplifica a organização dos dados coletados, bem como permite visualizar conexões entre eles, de forma a facilitar a elaboração de conclusões a partir da análise realizada (Peça, 2008).

Considerando as informações anteriormente mencionadas sobre a facilidade e acessibilidade dos dispositivos móveis, identificou-se uma demanda quanto ao fluxo de visita dos profissionais de saúde aos leitos e enfermarias de estabelecimentos de saúde que utilizam o Gestor Saúde como ferramenta para administração de internações e atendimentos. Devido à ausência de uma experiência responsiva no sistema mencionado, este trabalho propõe o aplicativo Gestor Prontuário Mobile, no qual visa auxiliar os profissionais de saúde a visualizar as informações do prontuário contidas na base de dados do Gestor Saúde, sem a necessidade de acessar o sistema Web. Além disso, permite a comunicação dos profissionais envolvidos no tratamento do paciente através de solicitações de visitas ao leito especificado e a coleta e análise

dos sinais vitais de pacientes internados, que é uma evolução à coleta realizada no sistema legado, no qual além de dispor o histórico de SSVV em formato de tabela, pouco é utilizada de fato pelos os profissionais, antes, porém utilizam fichas impressas para preenchimento dos dados e raramente importam esses dados para o Gestor Saúde. A solução buscar solucionar esse problema permitindo a adição dos sinais vitais diretamente pelo aplicativo.

Diante do exposto, percebe-se a relevância do projeto, uma vez que atende primeiramente à sociedade, representada pelos os pacientes atendidos pelo o Sistema Único de Saúde (SUS), e também à empresa Supra Tecnologia, visto que o produto será comercializado pela companhia como extensão do sistema já existente.

1.3 OBJETIVOS

Nesta seção serão apresentados o objetivo geral e específicos do projeto.

1.3.1 Objetivo Geral

Proporcionar aos profissionais de saúde um ambiente melhor para o acesso aos dados do PEP do Gestor Saúde, tornando a visualização desses dados mais fluída e organizada. Além disso, aperfeiçoar a assistência ao paciente, proporcionando um cuidado melhor, agilizando e aprimorando o atendimento.

1.3.2 Objetivos Específicos

- Incentivar os profissionais a monitorar os paciente internados com maior avidez, favorecendo a interação indireta entre profissionais de saúde durante o tratamento dos pacientes;
- Solucionar o problema da adição dos sinais vitais no sistema. Por meio do aplicativo, os sinais vitais serão adicionados diretamente no sistema, dispensando o intermédio de uma ficha física;
- Projetar uma interface do usuário amigável e intuitiva que permita aos profissionais de saúde acessar facilmente o PEP e localizar, rapidamente, informações relevantes no mesmo;
- Migrar, gradativamente, algumas funcionalidades do sistema Web na plataforma mobile, desenvolvendo um aplicativo que seja pioneiro em implementar funcionalidades do Gestor Saúde, estabelecendo um ponto de partida para, posteriormente, expandir mais recursos para dispositivos móveis.

1.4 ORGANIZAÇÃO DO PROJETO

Este documento está dividido em cinco capítulos, no qual o primeiro foi abordado acima. Os próximos capítulos estão organizados da seguinte forma:

- **Fundamentação Teórica:** Este capítulo abordará todos os conceitos sobre ferramentas utilizadas e sobre o domínio que envolve a solução apresentada no presente trabalho;
- **Gestor Prontuário Mobile:** Neste capítulo, a solução é apresentada e descrita em aspectos técnicos através de diagramas UML, detalhes de linguagem e bibliotecas utilizadas e também organização e arquitetura da solução e outros detalhes de implementação;
- **Metodologia:** Este capítulo descreve como foi realizado o planejamento para a implementação da solução, isto é, qual metodologia, ferramentas e atores envolvidos no processo;
- **Conclusão:** Por fim, o último capítulo discorre sobre a conclusão dos objetivos propostos no projeto, bem como os resultados obtidos e os trabalhos futuros.

1.5 RESUMO

Neste capítulo foi apresentado o contexto que envolve a solução proposta, isto é, a origem do Prontuário do Paciente e seu desenvolvimento ao longo das décadas, ressaltando a importância do PEP como avanço do prontuário físico e descrevendo seus benefícios se comparado ao documento físico. Assim também, justifica a implementação da solução apontando os problemas do sistema Web legado da empresa Supra Tecnologia. Por fim, expõe o objetivo geral e objetivos específicos almejados com a solução, além de descrever a organização do documento.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, serão apresentados conceitos-chave para o entendimento do projeto, no qual auxiliarão o leitor a compreender de forma mais clara o escopo do projeto e também as tecnologias nele utilizadas.

2.1 PRONTUÁRIO MÉDICO

O prontuário médico consiste em um conjunto de documentos clínicos que contém todo o histórico de atendimentos do paciente. No prontuário são registradas informações, sinais e imagens no qual são coletadas a partir da assistência ao paciente em um estabelecimento de saúde. Isso possibilita a comunicação entre toda a equipe de multiprofissionais e a continuidade da assistência prestada ao indivíduo (Conselho Federal de Medicina, 2002).

Em linhas gerais, o objetivo do prontuário médico é centralizar as informações pertinentes acerca do paciente durante seu processo de atenção à saúde. Essas informações tais como procedimentos realizados e seus resultados, prescrições de medicamentos, terapias, dieta, entre outras, descrevem a evolução do indivíduo e ajudam os profissionais de saúde a obter agilidade e qualidade no atendimento e também um atendimento mais direcionado.

O Prontuário médico deve conter os itens listados abaixo, seja em caráter eletrônico ou físico:

- Identificação do paciente – nome completo, data de nascimento (dia, mês e ano com quatro dígitos), sexo, nome da mãe, naturalidade (indicando o município e o estado de nascimento), endereço completo (nome da via pública, número, complemento, bairro/distrito, município, estado e CEP);
- Anamnese, exame físico, exames complementares solicitados e seus respectivos resultados, hipóteses diagnósticas, diagnóstico definitivo e tratamento efetuado;
- Evolução diária do paciente, com data e hora, discriminação de todos os procedimentos aos quais o mesmo foi submetido e identificação dos profissionais que os realizaram, assinados eletronicamente quando elaborados e/ou armazenados em meio eletrônico;
- Nos prontuários em suporte de papel é obrigatória a legibilidade da letra do profissional que atendeu o paciente, bem como a identificação dos profissionais prestadores do atendimento. São também obrigatórias a assinatura e o respectivo número do CRM;

- Nos casos emergenciais, nos quais seja impossível a colheita de história clínica do paciente, deverá constar relato médico completo de todos os procedimentos realizados e que tenham possibilitado o diagnóstico e/ou a remoção para outra unidade (Conselho Federal de Medicina, 2002).

Visto que a identificação do paciente é um item obrigatório para o estabelecimento de um prontuário, é possível concluir que o prontuário médico configura-se como um documento pessoal e sensível, de acordo com Lei Geral de Proteção de Dados Pessoais (LGPD), no qual define (Brasil, 2018a):

- **Dado pessoal:** toda informação relacionada a pessoa natural identificada ou identificável;
- **Dado pessoal sensível:** dado pessoal sobre origem racial ou étnica, convicção religiosa, opinião política, filiação a sindicato ou a organização de caráter religioso, filosófico ou político, dado referente à saúde ou à vida sexual, dado genético ou biométrico, quando vinculado a uma pessoa natural.

Nesse contexto, é fundamental preservar o princípio da confidencialidade, o qual estabelece que as informações relativas ao paciente devem ser acessadas apenas pelos profissionais envolvidos em seu tratamento, ou anonimizadas para fins de pesquisa científica ou ensino. Além disso, é essencial que o local de armazenamento dos documentos seja de acesso restrito, impedindo qualquer pessoa não autorizada, visualizar os dados.

Uma vez que o prontuário exerce a função de principal fonte de informação acerca do paciente, os dados contidos nele devem estar em conformidade com a real situação do paciente, ou seja, devem ser os mais verídicos possíveis. As informações coletadas precisam ser acertadas e verdadeiras, a fim de favorecer o tratamento do paciente e não prejudicá-lo. Irregularidades contidas nos documentos podem comprometer o estado do paciente e oferecer graves riscos à saúde do mesmo, além de prejudicar o profissional e o estabelecimento de saúde. Portanto, o registro de informações deve ser um dever de toda a equipe (Tonello, Nunes e Panaro, 2013).

Em suma, para todos os envolvidos no tratamento do paciente, o prontuário mostra-se um grande aliado. Para o paciente, o documento traduz-se em toda sua história clínica: procedimentos realizados, alergias, internações, entre outros. Essas informações são úteis e favorecem a continuidade do tratamento e também podem servir como instrumento de defesa legal em casos de irregularidade ou equívocos, assim como documento comprobatório para a concessão de seguros.

Para o profissional de saúde, o prontuário desempenha um papel fundamental ao auxiliar e agilizar seu trabalho, permitindo uma visualização prévia e completa do histórico do paciente sob seus cuidados. Essa ferramenta possibilita ao profissional adquirir um conhecimento aprofundado sobre o paciente, reduzindo equívocos

e garantindo a oferta de um tratamento adequado e condizente com a realidade do paciente.

Para os estabelecimentos de saúde, o uso do prontuário permite obter um maior controle de gestão. Isso ocorre devido ao registro detalhado de todos os eventos relacionados ao indivíduo, incluindo procedimentos executados, medicamentos administrados, consultas realizadas, entre outros. Todos esses itens são essenciais para o faturamento adequado, permitindo que o hospital administre corretamente os valores repassados a ele, garantindo assim uma gestão eficiente. Ademais, a análise conjunta dos dados dos prontuários pode contribuir para melhorar a qualidade de saúde de uma comunidade ou sociedade. Através da observação do prontuário de pacientes ou grupos de pacientes que compartilham a mesma área geográfica e/ou social, é possível gerar indicadores e mensurar quais tratamentos foram realizados e o custo para realizá-los. A sistematização dessas informações pode levar ao conhecimento do nível de saúde de uma determinada população, sendo possível a criação de políticas de atendimento e gestão das organizações de saúde (Marin, Massad e Neto, 2003).

Atualmente, a assistência ao paciente tem envolvido diversos profissionais da saúde, tais quais médicos, enfermeiros, fisioterapeutas, nutricionistas, entre outros, com o objetivo de oferecer ao paciente o aparato necessário para sua melhora de quadro. Diante disso, o conteúdo presente no prontuário é bastante diversificado, visto que é alimentado por áreas multidisciplinares, no qual cada profissional individual lança uma variedade de informações, o que beneficia cada profissional envolvido, uma vez que essa variedade de informações enriquece o documento e permite um maior nível de confiabilidade.

2.2 PRONTUÁRIO ELETRÔNICO DO PACIENTE

Como explicado anteriormente, o prontuário médico é um dos documentos mais importantes para um paciente que está sob cuidados médicos, uma vez que ele define a história clínica e evolução do indivíduo, desde de seu primeiro atendimento. É imprescindível que esse documento esteja armazenado com segurança, preserve o sigilo médico exigido legalmente (Bezerra, 2009) e esteja sempre disponível. Entretanto, na prática, utilização de papel como meio para gravar informações tão importantes, mostrou-se inviável para cumprir as exigências assinaladas. Diante disso, a utilização do Prontuário Eletrônico do Paciente possui várias vantagens se comparado ao modelo tradicional, dentre elas (Wesolowski, 2016):

- **Precisão de informações médicas:** O PEP ajuda a evitar erros e reduz a duplicação de informações, visto que os profissionais acessam os dados de forma atualizada;
- **Uniformidade na apresentação dos itens obrigatórios do PEP:** Devido ao seu

formato eletrônico, o PEP estrutura os dados de forma organizada e eficiente, reduzindo possíveis erros de interpretação, omissões e ilegibilidade;

- **Acesso e arquivamento instantâneos, gerando ganho de tempo:** O tempo gasto na busca do paciente, dentre diversos outros paciente presentes no estabelecimento de saúde, a fim de associar um novo registro ao seu prontuário, é substituído por poucos cliques na tela de um computador;
- **Qualidade no atendimento:** A visualização dos dados torna-se mais ágil e compreensível, o que faz o profissional tirar melhor proveito das informações e, diante disso, oferecer melhoria no serviço de saúde prestado;
- **Integração entre os profissionais de saúde:** O acesso em tempo real, possibilita profissionais de saúde compartilharem informações entre si de forma simultânea, como também acessar novas atualizações acerca do paciente de forma quase que instantânea;
- **Segurança de dados:** O uso do papel como meio de armazenamento de dados críticos é bastante falho. As Informações nele escritas são perdidas facilmente pelo o desgaste, além de não oferecer proteção a visualização das informações por terceiros. O PEP, além de não sofrer desgaste físico, possui recursos de autenticação e autorização;
- **Atualização instantânea de dados:** Com o uso do PEP, novos registros são automaticamente inseridos ou atualizados tornando-se imediatamente disponíveis para todos os usuários com acesso aos dados. Isso garante que os usuários tenham sempre acesso à versão mais atualizada das informações;
- **Portabilidade e acesso remoto:** Por estar na Internet, o PEP pode ser acessado de qualquer lugar através de qualquer dispositivo que possua acesso à rede;
- **Redução de custos:** Para a manutenção do prontuário em papel é necessário espaços físicos com tamanho significativo para conter a quantidade de documentos. Além disso, o custo com papel e impressão, assim como o tempo gasto na busca e manuseio dos documentos é reduzido drasticamente.

2.3 SEGURANÇA E PRIVACIDADE DE ACORDO COM A LGPD

A Lei Geral de Proteção de Dados (LGPD) trouxe uma revolução na forma como os dados pessoais devem ser tratados por organizações. O intuito principal da Lei é proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural. Ou seja, a Lei diz respeito estritamente a como os dados de pessoas físicas são tratados, respeitando a individualidade. A LGPD

(Brasil, 2018a) define o que são dados pessoais e qual desses dados são considerados sensíveis, assim como os “personagens” que atuam na LGPD e lidam com esses dados, que são:

Art. 5º Para os fins desta Lei, considera-se:

I - dado pessoal: informação relacionada a pessoa natural identificada ou identificável;

II - dado pessoal sensível: dado pessoal sobre origem racial ou étnica, convicção religiosa, opinião política, filiação a sindicato ou a organização de caráter religioso, filosófico ou político, dado referente à saúde ou à vida sexual, dado genético ou biométrico, quando vinculado a uma pessoa natural;

V - titular: pessoa natural a quem se referem os dados pessoais que são objeto de tratamento;

VI - controlador: pessoa natural ou jurídica, de direito público ou privado, a quem competem as decisões referentes ao tratamento de dados pessoais;

VII - operador: pessoa natural ou jurídica, de direito público ou privado, que realiza o tratamento de dados pessoais em nome do controlador;

VIII - encarregado: pessoa indicada pelo controlador e operador para atuar como canal de comunicação entre o controlador, os titulares dos dados e a Autoridade Nacional de Proteção de Dados (ANPD); (Redação dada pela Lei nº 13.853, de 2019)

IX - agentes de tratamento: o controlador e o operador;

Em poucas palavras, dados pessoais são aquelas informações referentes a pessoas naturais (pessoas físicas), passíveis de serem identificadas, e para os profissionais de saúde, os dados pessoais sensíveis são ainda mais comuns, visto que fazem parte do cotidiano da sua profissão, o que torna a responsabilidade destes maior.

É importante destacar que o profissional de saúde não é representado por nenhum dos personagens citados na Lei, sendo ele meros prepostos dos agentes de tratamento, isto é, aplicando os personagens ao contexto da saúde, o controlador representa o estabelecimento de saúde, por sua vez, o operador deve ser uma entidade distinta do controlador, isto é, que não atua como profissional subordinado a este ou como membro de seus órgãos (Conselho Federal de Medicina, 2022), por exemplo, num estabelecimento fora da rede hospitalar do estabelecimento controlador, o encarregado é aquele que guiará os agentes de tratamento a seguir à Lei, e, por fim, o titular é o paciente, que é o dono dos dados.

Vale mencionar que o Conselho Federal de Medicina (CFM) também possui normas de segurança, descritas na sua regulamentação e também na Lei nº 3.268/1957, e que, quando considerados infrações quanto à privacidade dos dados, tanto a LGPD quanto a ambas as leis mencionadas serão utilizadas para punir o ato, ou seja, não há interseção entre as leis, cada uma, porém atua em sua respectiva competência (Conselho Federal de Medicina, 2022).

O aplicativo Gestor Prontuário Mobile busca atender às normas da LGPD, uma vez que preserva os dados do titular evitando que outros indivíduos além do profissional consultem as informações sensíveis do paciente. Para isso, o profissional deverá logar sempre que inicializar o aplicativo, garantindo, através da autenticação, que é ele próprio que está acessando a aplicação. Além disso, o cadastro do profissional é realizado via plataforma Web (Gestor Saúde) pelos gestores dos estabelecimentos ou pela a equipe de suporte do sistema.

2.4 DESENVOLVIMENTO MULTIPLATAFORMA

Os sistemas operacionais (SOs) *Android* e *IOS* estão presentes na maior parcela de *smartphones* e *tablets* da atualidade. Por possuírem naturezas diferentes, a construção de aplicativos para o ecossistema *Android* diverge da construção de aplicativos para o sistema operacional *IOS*. Entre os pontos de diferenças estão, por exemplo: a linguagem de programação adotada em cada plataforma, componentes nativos, padrão de arquitetura, entre outros.

Considerando os desafios citados anteriormente, bem como a necessidade atual de criação de aplicativos que sejam comerciais, de rápida implementação e baixo custo de desenvolvimento, surgem os *frameworks cross-platform*, isto é, ferramentas especializadas em integrar as duas bases de código (*Android* e *IOS*) de forma transparente, no qual é refletida em um base de código comum, pertencente a linguagem utilizada pela ferramenta. As vantagens de utilização de *frameworks* desse tipo são:

- **Alcance de um amplo público:** *Frameworks cross-platform* exportam para os principais ecossistemas, garantindo que o código escrito será executado de forma nativa no SO alvo;
- **Redução de custo e tempo:** Desenvolver para múltiplas plataformas exige custo com equipes especializadas em cada plataforma e leva tempo até que toda a equipe esteja alinhada.

A estratégia de implementação do *cross-plataform* está em função da tecnologia utilizada. Na seção seguinte será discutida a forma que o *React Native* constrói interfaces *cross-plataform* de forma nativa.

2.5 REACT NATIVE

React Native é uma biblioteca *open-source* criada em 2015 pelo o *Facebook* com foco na criação de aplicativos nativos utilizando *React* Componentes e permite que o desenvolvedor crie aplicativos verdadeiramente nativos, não comprometendo a experiência dos usuários. A biblioteca fornece um conjunto central de componentes

nativos agnósticos à plataforma, como *View*, *Text* e *Image*, que são diretamente mapeados para os elementos de construção de interface nativa da plataforma alvo. Vários sistemas populares fazem uso dessa ferramenta, dentre eles o Facebook, Discord, Skype, Outlook, PlayStation, entre outros (Meta Platforms, 2023c).

O principal motivo de sua popularidade está no fato de ser baseado em *JavaScript* e *React*. Uma vez que *JavaScript* está em alta adesão nos últimos anos, é comum que desenvolvedores da linguagem procurem ferramentas que sejam amigáveis e familiares à eles. Assim também, desenvolvedores *React* não enfrentam dificuldade em iniciar no *React Native*, visto que a sintaxe de construção é a mesma, todavia com componentes diferentes e recursos voltados para dispositivos móveis.

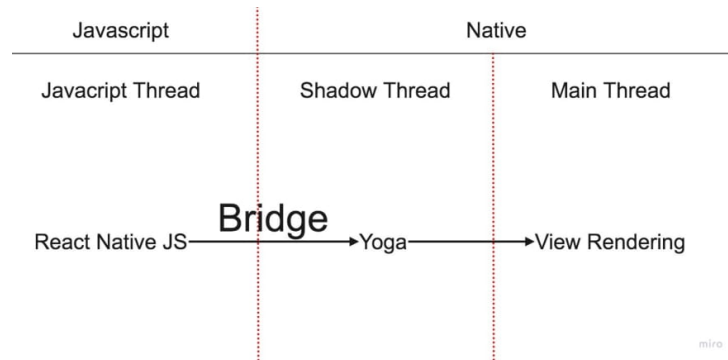
Outra vantagem do *React Native* é o recurso de *hot reloading* que permite que desenvolvedores vejam alterações realizadas no código fonte em tempo real no dispositivo ou simulador. O *hot reloading* é útil para ajustar os elementos de interface do usuário e estilos do aplicativo sem precisar reconstruí-lo toda vez. O *React Native* permite que os desenvolvedores executem os aplicativos enquanto atualizam os arquivos editados. Portanto, se ocorrer alguma alteração na interface do usuário, nenhum dado será perdido (Bigio, 2016).

A chave para o funcionamento do *React Native* é sua abordagem de “renderização nativa”. Em vez de utilizar *webviews*, como fazem muitos outros frameworks de desenvolvimento móvel multiplataforma, o *React Native* traduz os componentes de interface do usuário escritos em *JavaScript* diretamente para componentes nativos. Isso se dá graças ao novo gerenciador de interfaces de usuário (*UI Manager*), *Fabric*, uma evolução ao modelo legado chamado de *Bridge*.

Anteriormente, a comunicação entre o domínio *JavaScript* e o nativo era realizado utilizando a arquitetura *Bridge*, conforme a Figura 2. O *UI Manager* consistia de três *threads* que lidavam com as várias etapas de renderização até chegar à plataforma hospedeira. A *Javascript thread* lidava com a lógica de negócio, com o código fonte *Javascript*. É onde o código *React* é executado. A *Shadow thread* é responsável pelo cálculo dos layouts na plataforma hospedeira. Um vez que os dispositivos móveis não reconhecem o estilo *Flexbox*, como na Web, é utilizado o *Yoga*, uma ferramenta que traduz o estilo para a plataforma hospedeira. Finalmente, após a conclusão das operações de layout, a *Main thread* renderiza os componentes correspondentes na plataforma hospedeira. É importante salientar que os domínios *Javascript* e nativo, não possuem consciência da existência um do outro. A interação entre esses domínios é realizada exclusivamente pela *Bridge*. Supondo que se queira renderizar um componente em uma determinada plataforma, o *React Native* transformará o comando de criação desse componente em um *array JSON*, serializá-lo e então, enviá-lo à *Bridge* para que ela possa executá-lo na plataforma alvo. Considerando que todas as operações desse antigo modelo são assíncronas, de acordo com a documentação oficial do *React Native*

(Meta Platforms, 2023a): uma camada enviava os dados para a *Bridge* e “esperava” assincronamente que a outra camada os processasse, mesmo quando isso não era realmente necessário (tradução livre). Dessa forma, a renderização de animações e eventos, como a rolagem, pode parecer lenta devido ao tempo gasto na transferência de dados da *thread JavaScript*, no cálculo do *leiaute*, antes que ocorra qualquer renderização real. Devido à esses percalços, uma nova arquitetura foi proposta.

Figura 2 – Arquitetura Legado do *React Native*

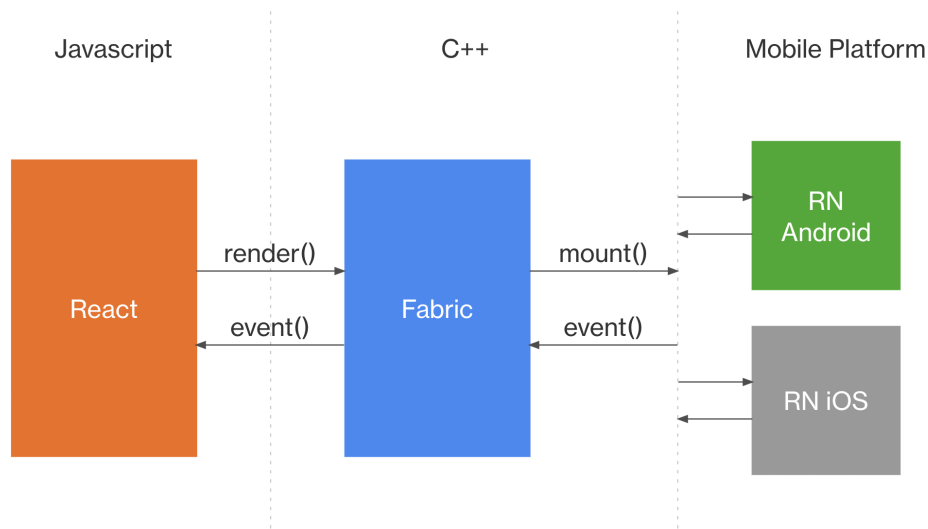


Fonte: Okafor (2022).

A arquitetura *Fabric* trouxe uma sequência de melhorias em relação ao modelo legado, dentre elas: unificar mais lógica de renderização em C++, melhorar a interoperabilidade com plataformas hospedeiras agora permitindo chamada síncronas, evitando os problemas de “salto” de *leiaute* ao incorporar uma visualização renderizada pelo *React Native* em uma visualização do *host* e desbloquear novas capacidades para o *React Native*. Em substituição à *Bridge* como comunicação entre os domínios *JavaScript* e nativo, a arquitetura *Fabric* utiliza o *JavaScript Interface(JSI)*, no qual expõe módulos nativos diretamente para o *JavaScript* via C++. Esse sistema permite chamar métodos na plataforma hospedeira usando código nativo estaticamente. Uma vantagem da implementação com C++ é o compartilhamento do mesmo código, o que melhora o desempenho, performance e manutenção para todas as plataformas. A arquitetura está representada na Figura 3.

A sequência de etapas seguidas pelo *Fabric* para a execução na plataforma hospedeira é chamada *render pipeline*, e é composta pelos estágios *render*, *commit* e *mount*. A etapa *render* começa quando o aplicativo é inicializado, então é criada uma árvore de componentes *React* (*React Element Tree*). Dessa árvore, o *Fabric* cria uma *React Shadow Tree* imutável feita em C++. Esta árvore representa o que será renderizado na plataforma hospedeira. Cada nó é criado sincronicamente, representado cada *host component* da árvore *React Element Tree*. Uma vez que a *React Shadow Tree* é imutável, qualquer alteração no estado gera um clone da árvore atual, com adição das mudanças realizadas nos componentes. Após a fase de *render*, a etapa de *commit* se inicia, utilizando o *Yoga* para calcular o *leiaute* final dos componentes de acordo com a *React Shadow Tree*, assim como promover qual das *shadow trees* será

Figura 3 – Arquitetura *Fabric*



Fonte: React Native (2023b).

renderizada. Finalmente, a etapa *mount* acontece na *thread Main* e consistem em o *React Native* saber qual será a próxima *React Shadow Tree* a ser renderizada, e uma vez descoberta, transformará esta árvore em uma *host tree* vista pelo usuário final. As alterações são percebidas através do *Tree Diffing*, no qual calcula as diferenças entre a árvore anterior e a árvore atual. Então, os componentes são renderizados na sua plataforma hospedeira correspondendo aos componentes nativos nela contidos.

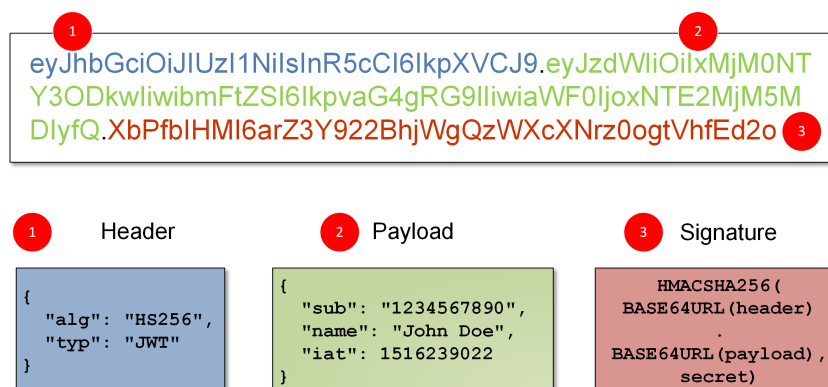
2.5.1 JSON Web Token

O JSON Web Token (JWT) é um padrão aberto (RFC 7519) que define uma forma compacta e autocontida de transmitir informações de forma segura entre as partes como um objeto JSON. Essas informações podem ser verificadas e confiadas porque são assinadas digitalmente. JWTs podem ser assinados usando um segredo (com o algoritmo HMAC) ou um par de chaves pública/privada usando RSA ou ECDSA (Auth0, 2023).

Alguns dos cenários onde o JWT é útil é na Autorização e Troca de informações. O cenário de Autorização é o mais utilizado. Seu principal objetivo é proteger as rotas de uma API através de uma camada de segurança, no qual o cliente HTTP, após autenticado, deverá informar um token nas próximas requisições, a fim de obter os recursos desejados. No segundo cenário, o token é útil para transmitir informações de forma segura entre as partes, isto porque o JWT pode ser assinado utilizando a chave pública e privada, o que garante validar a identidade dos remetentes e também a integridade da informação, uma vez que o token é assinado utilizando suas partes *header* e *payload* (Auth0, 2023). Um token JWT é formado por três estruturas que são separadas por pontos e codificadas para *base64*, são essas: *header*, *payload* e

signature como mostrado na Figura 4.

Figura 4 – Estrutura de um Token JWT



Fonte: Sajdak (2019).

A parte *header* define o tipo do token, nesse caso, JWT, e o algoritmo de assinatura utilizado e é um objeto JSON que é codificado para *base64* formando a primeira parte do token. A segunda parte armazena as *claims* (“reivindicações”, tradução livre), no qual representam as declarações sobre uma determinada entidade (geralmente o usuário) e metadados adicionais. As *claims* poderão ser reservadas, públicas ou privadas. *Claims* reservadas não são obrigatórias, mas quando usadas possuem significado, algumas delas são:

- **iss (emissor, do inglês *issuer*)**: identifica o emissor do token;
- **exp (tempo de expiração, do inglês *expiration time*)**: informa o tempo de expiração do token, em dias;
- **sub (sujeito, do inglês *subject*)**: identifica a entidade alvo. Geralmente as *claims* são sobre o sujeito.

Claims públicas são personalizáveis pelo o criador do token. Não são obrigatórias e geralmente armazenam informações sobre o usuário (entidade mais comum). *Claims* privadas são definidas entre as partes que trocam o JWT e não são registradas no IANA (Internet Assigned Numbers Authority).

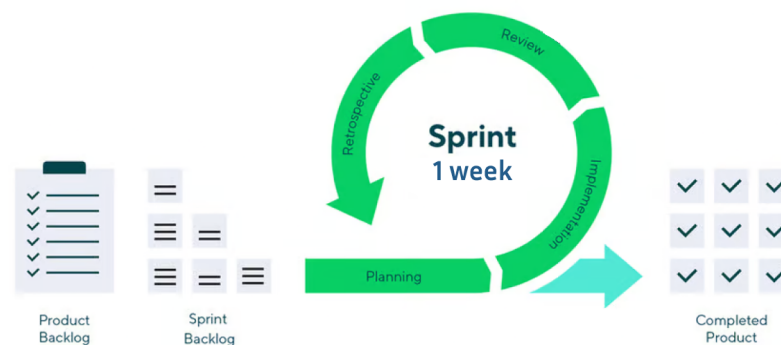
Por fim, a última parte, denominada *signature*, é formada pelas estruturas anteriores *header* e *payload*, encriptadas em *base64* e separadas por ponto, além disso, também é fornecido um segredo e então, esses dados são encriptados utilizando o algoritmo definido no *header*, como ilustra a Figura 4. A *signature* é importante pois é ela quem valida se houve alteração na mensagem ao longo do caminho e também, no caso de token assinados com uma chave privada, ele pode verificar se o emissor é mesmo quem ele diz ser (Auth0, 2023).

3 METODOLOGIA

O desenvolvimento de software envolve várias etapas que, se não forem organizadas de forma adequada, poderão comprometer a qualidade do produto, atrasar sua entrega e diminuir o valor do mesmo. Diante disso, algumas etapas da Engenharia de Software foram seguidas no projeto, tais como levantamento de requisitos, processo de software *Scrum*, modelos de software e prototipagem das telas da aplicação.

Para o planejamento das atividades foi utilizada a metodologia ágil *Scrum*, que consiste em uma estrutura de gestão de projetos com foco na abordagem de gerenciamento de desenvolvimento iterativo, ao invés das abordagens técnicas específicas da engenharia de software ágil (Sommerville, 2011). Isto permite que o produto de software seja entregue de forma mais clara e segura para o cliente, uma vez que à cada ciclo *sprint*, novos requisitos podem ser adicionados ou priorizado, estando à cargo do *Product Owner* (PO) definir quais requisitos são mais urgentes. A Figura 5 descreve o ciclo de *sprints* da metodologia *Scrum*, adaptada ao presente projeto.

Figura 5 – Ciclo de uma *Sprint*



Fonte: Wrike (2023).

No *Scrum*, o *Product Backlog* (Lista de Produtos) representa uma lista de todas as funcionalidades, tarefas e requisitos pendentes de um produto, e é usado para planejar o trabalho a ser realizado. Os itens do *Product Backlog* foram definidas pelo o autor considerando os requisitos apontados pelo PO. A etapa de *Planning* (Planejamento) consiste em discutir sobre quais objetivos deverão ser alcançados durante a próxima *sprint* e então, selecionar itens do *Product Backlog* que satisfaçam o objetivo. Os itens selecionados durante uma *sprint* são chamados *Sprint Backlog*. Esta etapa foi realizada pelo o autor.

Uma vez definido o *Sprint Backlog*, ocorre a implementação (*Implementation*) dos requisitos. A *sprint* foi definida com duração de uma semana. Após o término do ciclo, o autor apresenta os itens concluídos ao *Scrum Master*, papel atribuído ao professor orientador, em primeiro momento, e após, ao PO, que validará os requisitos. Esta fase chama-se *Review* (Revisão). Logo após, a fase de *Retrospective* (Retrospectiva) analisa quais metas definidas na *sprint* foram concluídas e o que pode ser melhorado para o próximo ciclo. Essa etapa envolve o *Scrum Master* e o autor.

O levantamento dos requisitos do aplicativo ocorreu através da interação com o PO, um dos colaboradores da empresa Supra Tecnologia responsável pelo o suporte ao sistema Gestor Saúde e, por isso, tem contato constante com os profissionais de saúde usuários do sistema logo, entende suas dificuldades e anseios. Diante da análise dos requisitos, os casos de uso foram criados e descritos, como já mencionados nas seções 4.1 e A.

Através desse processo colaborativo, o autor criou os protótipos iniciais das telas. Em seguida, foram submetidos à avaliação do PO e do professor orientador deste trabalho, que desempenhou o papel de *Scrum Master*. Os dois personagens foram fundamentais na análise crítica dos *designs* propostos e, logo depois implementados. O PO, com base em sua experiência e vivência com os profissionais, ofereceu sugestões valiosas para aprimorar tanto a aparência visual quanto a facilidade de uso do aplicativo, além disso foi responsável pelo o domínio da aplicação, ou seja, as *tasks* geradas no *Product Backlog* provém da análise e conversa direta do autor com ele. Igualmente, o *Scrum Master* contribuiu na sugestão de componentes de telas que fossem mais semânticos e coerentes, também corrigindo arestas de *designer* que vieram a surgir, e, não menos importante, gerenciando a escolha das *tasks* para cada *sprint*, analisando o resultado e dificuldades ao final de cada ciclo (através de reuniões *online* semanais com o autor) ou durante a execução deste.

O *Product Backlog* foi implementado utilizando o *Kanban*, um método visual para gerenciar e conduzir o trabalho. Este método é utilizado principalmente no desenvolvimento de software devido a essa área ser bastante volátil no sentido de exigir mudanças constantes e interações. Geralmente o *Kanban* é utilizado por equipes que utilizam *Scrum* devido aos conceitos de ambos se complementarem. O software utilizado para aplicar os conceitos do *Kanban* será o software *Trello*¹, que é referência em ferramentas do gênero.

¹ <https://trello.com/>

4 GESTOR PRONTUÁRIO MOBILE

O protótipo atual foi concebido devido à análise do uso dos profissionais dos menus¹ de evoluções do sistema Gestor Saúde, utilizados durante o período em que o paciente está internado. Considerando a limitação do Gestor Saúde quanto à visualização dos dados do prontuário, assim como a facilidade de uso que os dispositivos móveis proporcionam, nasceu o aplicativo Gestor Prontuário Mobile, que busca solucionar os problemas citados anteriormente e também adiciona outros recursos que auxiliam ainda mais o profissional de saúde.

O aplicativo será utilizado por três grupos de profissionais principais: médicos, enfermeiros e técnicos de enfermagem, correspondendo às evoluções médica, de enfermagem e multidisciplinar, respectivamente. Vale destacar que esta divisão diz respeito somente à inserção dos dados, ou seja, qualquer profissional pode visualizar as evoluções do paciente independente de seu conselho profissional², uma vez que é esse detalhe que torna o PEP tão eficiente: o compartilhamento de informações multidisciplinares.

4.1 CASOS DE USO

Nesta seção serão explorados os principais casos de uso do aplicativo Gestor Prontuário Mobile. O levantamento de requisitos foi realizado através da análise das funcionalidades do sistema já existente, o Gestor Saúde. Os casos de uso encontrados foram:

- **Caso de Uso 01 – Visualizar histórico médico:** Caso necessário, o profissional de saúde pode visualizar o histórico médico de um dado paciente, buscando através do número de prontuário ou nome;
- **Caso de Uso 02 – Visualizar solicitações de visita:** O profissional de saúde pode visualizar as solicitações de visita atrelas ao seu código de conselho;
- **Caso de Uso 03 – Registrar sinais vitais:** O profissional de saúde pode registrar sinais vitais do paciente periodicamente;
- **Caso de Uso 04 – Solicitar visita:** Caso o paciente necessite de uma visita profissional, o profissional plantonista poderá solicitar a visita de profissionais de acordo com o código de conselho;

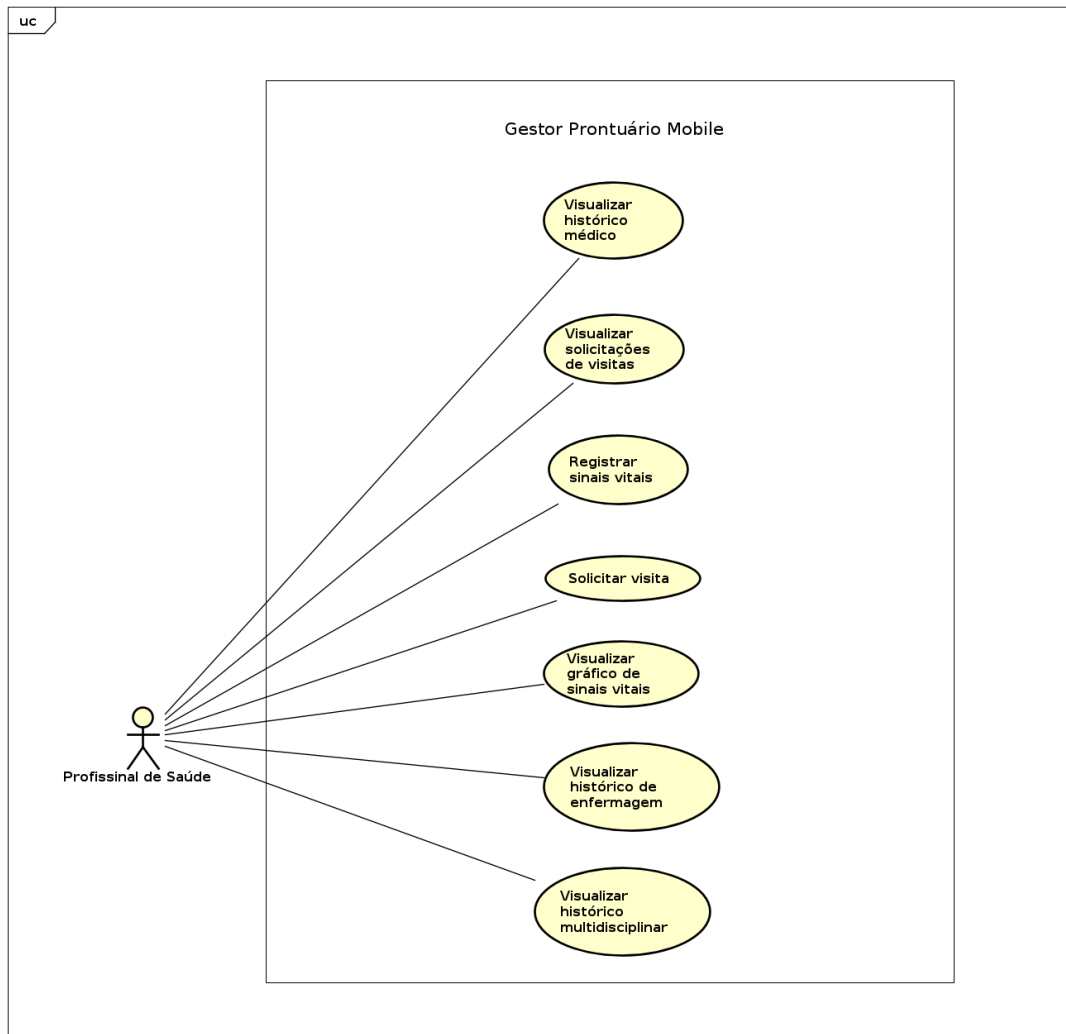
¹ Representa cada serviço do Gestor Saúde

² Organizações formadas pelos trabalhadores de uma determinada profissão e que tem a principal função de regulamentar a atividade profissional (Pontifícia Universidade Católica do Paraná, 2023)

- **Caso de Uso 05 – Visualizar gráfico de sinais vitais:** Facilitar a interação entre os profissionais de saúde e os dados dos sinais vitais coletados;
- **Caso de Uso 06 – Visualizar histórico de enfermagem:** Caso necessário, o profissional de saúde pode visualizar o histórico de enfermagem de um dado paciente, buscando-o através do seu número de prontuário ou nome;
- **Caso de Uso 07 – Visualizar histórico multidisciplinar:** Caso necessário, o profissional de saúde pode visualizar o histórico multidisciplinar de um dado paciente, buscando através do número de prontuário ou nome.

A Figura 6 demonstra os casos de usos listado acima em forma de diagrama. Cada caso de uso está devidamente descrito na seção A.

Figura 6 – Casos de Uso



Fonte: Próprio Autor.

4.2 CARACTERÍSTICAS DA IMPLEMENTAÇÃO

O projeto utiliza a linguagem de programação PHP³ como base para o *back-end*. Esta linguagem é específica para a Web do lado do servidor e possui ampla popularidade e largo uso na Web. O Gestor Saúde utiliza da estratégia *low-code* para auxiliar o desenvolvimento de novas aplicações tornando a entrega de software mais rápida, por meio da geração do *CRUD* de um novo menu para então, o desenvolvedor inserir a lógica de negócio. As rotinas *low-code* do sistema são consideradas legado, mantidas na versão 5.6 do PHP e não sofrem mais modificações significativas no seu código-fonte.

Para realizar a comunicação do Gestor Saúde com outros sistemas é disponibilizado interfaces de comunicação externas utilizando o pacote *Luracast Restler*, que implementa interfaces *Web API Server* escritas em PHP (Luracast, 2023). As rotas consumidas pelo o aplicativo foram criadas de acordo com este modelo.

Para armazenamento de dados, é utilizado o PostgreSQL, um banco de dados objeto-relacional *open-source* com mais de 35 anos de desenvolvimento ativo, o que lhe conferiu uma forte reputação em termos de confiabilidade, robustez de recursos e desempenho (PostgreSQL Global Development Group, 2023). Assim também, devido à sua característica *open-source*, torna-se altamente extensível e oferece diversos recursos como, por exemplo, possibilitar a definição de tipos próprios.

A comunicação entre o banco de dados e a aplicação é realizada por meio do pacote *Active Record* que utiliza o padrão com o mesmo nome. A arquitetura desse padrão consiste em estender as classes de domínio utilizando uma classe base, geralmente chamada *Model*, que adiciona comportamentos e atributos às subclasses, a fim de relacionar a classe de domínio a uma tabela do banco de dados e através disso, permitir operações de *CRUD* e relacionamento entre as classes de domínio.

4.3 ORGANIZAÇÃO DO PROJETO

Aqui serão descritas as camadas do projeto e o objetivo de cada uma delas. Também será descrito o padrão de arquitetura utilizado no aplicativo, assim como eventuais diagramas que representam as partes principais da aplicação.

O *back-end* utiliza a arquitetura *Model-View-Controller* (MVC), no qual define três camadas que representam o fluxo de dados na aplicação. A camada *Model*, representa o domínio da aplicação, regras de negócio e está mais próxima da realidade do cliente. É esta camada que define o escopo da aplicação e sua finalidade. Por sua vez, a camada de visualização (*View*) é responsável por apresentar os dados da

³ Uma linguagem de *script* geral e popular, especialmente adequada para o desenvolvimento Web. Atualmente, boa parte dos sites em todos mundo ainda utilizam PHP. Durante a escrita deste documento, a versão mais recente é a 8.2 (The PHP Group, 2023)

camada de *Model* ao usuário, é a camada mais próxima ao usuário e por isso define a chamada de eventos para a camada *Controller* que, por sua vez, retorna dados para ela para serem apresentados.

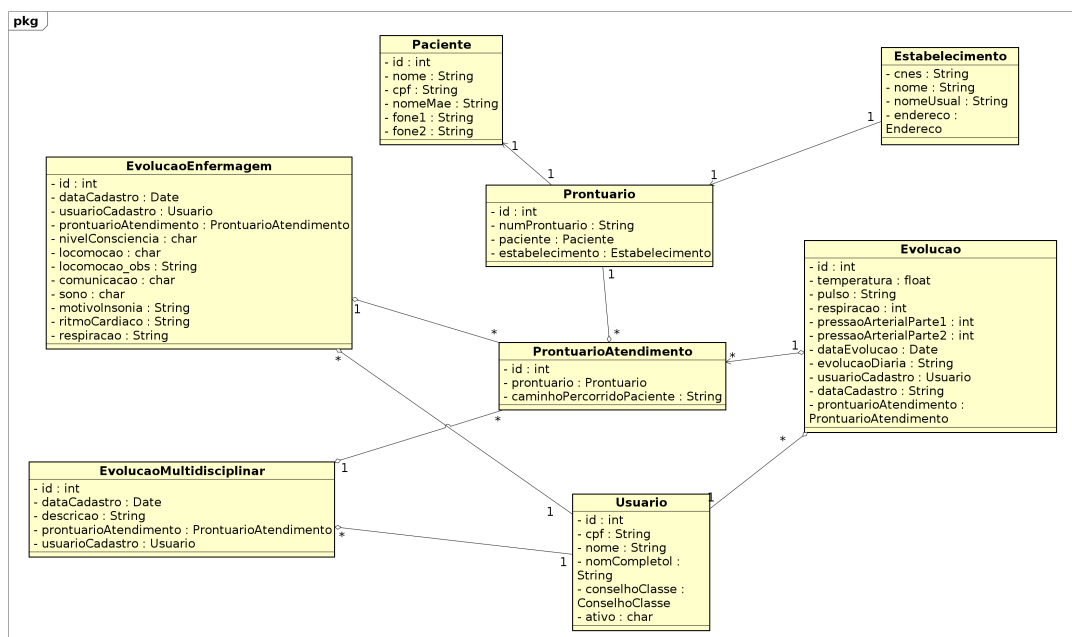
Finalmente, a camada *Controller* é responsável por gerenciar as outras duas camadas, recebendo os dados vindos da *View*, interagindo com o *Model* de acordo com os dados reportados e após, retornando novamente dados atualizado para a camada de visualização.

O objetivo dessa arquitetura é facilitar o desenvolvimento, uma vez que reduz o risco que alterações em um componente prejudiquem outro fora do escopo, a manutenção, visto que a separação de responsabilidades leva a um código mais “limpo” e organizado e a escalabilidade do sistema conquanto, um componente pode ser dimensionado independente de outros.

A Figura 7 apresenta o digrama de classe que corresponde à camada *Model* do MVC. Vale destacar que o sistema possui inúmeras outras classes que, por sua extensão, não foram mencionadas no diagrama. As principais entidades do sistemas são:

- **Usuario:** Representa o profissional que possui login no sistema. Considerando o escopo do projeto, cada evolução está atrelada a um usuário, ou seja, cada registro armazena o usuário que o criou e o alterou. É importante destacar que existe a classe Profissional, entretanto o modelo de dados do sistema não oferece nenhum relacionamento entre a classe Usuário e a classe Profissional. Para resgatar o registro de profissional através da classe Usuário, é necessário comparar o atributo CPF, comum às duas classes;
- **Prontuario:** Esta entidade guarda a identificação do paciente quando este visita o local pela primeira vez;
- **ProntuarioAtendimento:** Diferente da entidade Prontuario, esta classe identifica cada passagem do paciente no estabelecimento de saúde, bem como seu fluxo durante sua permanência no local. Ela está associada às classes de evoluções, uma vez que é necessário saber em qual visita aquela evolução pertence;
- **Evolucao:** Representa um registro da evolução médica;
- **EvolucaoEnfermagem:** Representa um registro da evolução de enfermagem;
- **EvolucaoMultidisciplinar:** Representa um registro da evolução multidisciplinar;
- **Paciente:** Representa o indivíduo que visita o estabelecimento e possui seu prontuário registrado;
- **Estabelecimento:** A entidade que representa o local onde o paciente se encontra internado.

Figura 7 – Diagrama de Classes Simplificado do Gestor Saúde



Fonte: Próprio Autor.

Para fins de simplificação, nem todos os atributos das entidades foram adicionados ao diagrama, devido ao fato de corresponderem a escopos diferentes do abordado no projeto.

4.4 ARQUITETURA

O sistema possui uma arquitetura baseada em duas camadas, isto é, uma camada para visualização dos dados e outra para a manipulação destes. A camada de visualização é chamada *front-end*, enquanto a de manipulação chama-se *back-end*.

Os modelos de arquitetura modernos visam principalmente separar as responsabilidades de uma aplicação, reduzindo-as a escopos menores, a fim de lidar com um serviço de forma individual, sem que um interfira no outro. Além disso, possibilita que vários serviços sejam implementados em tecnologias diferentes, por exemplo em linguagens de programação diferentes, desde que possuam uma interface de comunicação entre eles.

Portanto, a divisão do sistema em *front-end* e *back-end* favorece o desenvolvimento do software, uma vez que facilita a criação de times de desenvolvimento especializados na área especificada e também sua manutenção, pois correções e melhorias em um serviço não afetarão a outros.

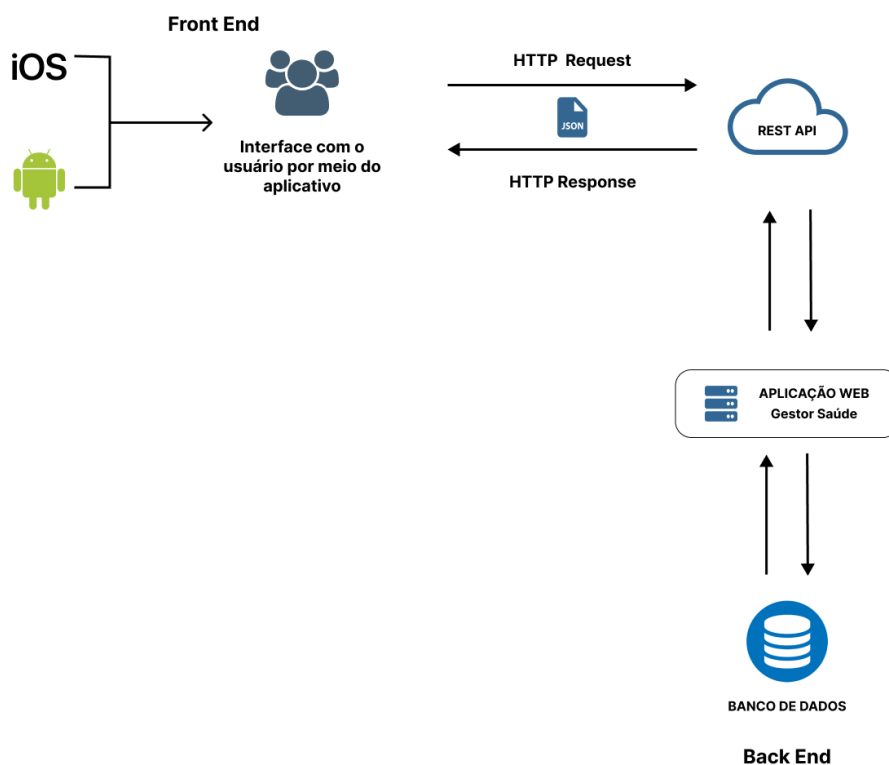
A comunicação entre o *front-end* e o *back-end* é realizada através de uma *Application Programming Interface (API)*, traduzido como Interface de Programação de Aplicações. Uma API representa um conjunto de protocolos e definições usados no desenvolvimento e integração de aplicações. A API define regras de comunicação

entre o emissor e o receptor da mensagem.

Existem diversas maneiras de implementar APIs Web, mas para a construção do sistema foi adotado o estilo arquitetural de Transferência de Estado Representacional, do inglês *Representational State Transfer* (REST), amplamente utilizado como modelo de comunicação entre sistemas distribuídas. O modelo REST utiliza o Protocolo de Transferência de Hipertexto, do inglês *Hypertext Transfer Protocol* (HTTP), como base para sua implementação, dando uma interface de abstração semântica e padronizada. A Figura 8 ilustra a arquitetura do Gestor Prontoário Mobile e o caminho que a informação percorre até chegar aos usuário do aplicativo.

O aplicativo consome os dados vindo do Gestor Saúde. O formato de dados empregado para a comunicação entre os sistemas foi o *JavaScript Object Notation* (JSON), amplamente utilizado nos modelos REST devido à sua flexibilidade, leveza e simplicidade. Se comparado à Linguagem de Marcação Extensível, do inglês *Extensible Markup Language* (XML), por exemplo, o JSON utiliza menos caracteres para representar o mesmo conteúdo, além de ser menos verboso. Atualmente, a maioria das linguagens de programação possuem suporte para lidar com o formato.

Figura 8 – Arquitetura do Gestor Prontoário Mobile



Fonte: Próprio Autor.

O Gestor Prontuário Mobile solicita ao sistema Gestor Saúde um determinado recurso através de requisição HTTP (*HTTP Request*). A aplicação, por sua vez, processa a requisição e acessa o banco de dados a fim de buscar os dados solicitados. Após obtê-los, processa-os e retorna ao cliente através de uma resposta HTTP (*HTTP Response*) a ser tratada pelo aplicativo e exibida ao usuário. É pertinente pontuar que o acesso às informações é restrito aos usuário que estejam previamente autenticados.

A autenticação é realizada por meio do padrão JWT, um padrão aberto definido pela *RFC7519*, no qual descreve uma forma compacta e autossuficiente de transmitir informações de forma segura entre as partes como um objeto *JSON*. Essa ferramenta tem se popularizado no contexto de autenticação de APIs REST devido ao formato adotado, isto é, o *JSON*, no qual é também o formato mais utilizado para intercâmbio de informações em *REST APIs* e também pela a segurança, uma vez que o *token* é assinado digitalmente.

Após o usuário validar com sucesso suas credenciais através do login no aplicativo, um *token* é gerado e retornado ao aplicativo, no qual o armazenará no dispositivo móvel e o utilizará para futuras requisições à API durante a sessão no aplicativo.

4.5 API REST

Visto que uma REST API pode ser definida como um contrato entre um cliente que solicita um recurso e um provedor, que fornece o recurso, é necessário que o conteúdo definido pelo o cliente, bem como o conteúdo definido pelo o provedor estejam bem definidos e documentados. Caso isso não seja atendido, a curva de aprendizado poder ser ingreme, dificultando a adoção da API por parte dos desenvolvedores e prejudicando a colaboração entre as equipes.

Diante disso, a fim de fornecer informações claras e concisas sobre a estrutura, funcionalidade e os parâmetros e retornos da API, bem como facilitar a integração da mesma a outro sistema, caso necessário, a REST API do *back-end* do projeto Gestor Prontuário Mobile foi documentada.

Considerando a inexistência de uma especificação oficial para documentação de APIs, neste projeto adotou-se a especificação *OpenAPI*, principalmente devido à sua popularidade, simplicidade e também integração com *frameworks* e bibliotecas mais conhecidas.

A maioria das rotas estão protegidas, sendo necessário a autenticação prévia do usuário, como já foi abordado na seção 4.4. Pode-se observar nos Quadros 1 a 5 a documentação de algumas rotas. Embora não sejam a completude da documentação, ajudam a dar uma visão geral das demais rotas, uma vez que a maioria delas são documentadas de forma semelhante. Toda a documentação segue no anexo do documento.

Tabela 1 – Descrição da rota Autenticação do Usuário

Autenticação do Usuário

O usuário informa suas credenciais de acesso que serão validadas pelo o sistema. Uma vez validadas com sucesso, o usuário recebe um *token* que serve para obter acesso às demais rotas da API, caso não, um código de erro é retornado.

- **URL**
 - /sign
- **Método**
 - POST
- **Parâmetros**
 - Nenhum parâmetro
- **Corpo**
 - urs: String
 - pwd: String
 - cliente: Integer
- **Resposta de Sucesso**
 - Código: 200 OK
 - Conteúdo: JSON com a chave *token* contendo o *token JWT* gerado.
- **Resposta de Erro**
 - Código: 200 OK
 - Conteúdo: JSON com o seguinte conteúdo:
{"status": "error", "message": "Usuário e/ou senha inválidos"}

Fonte: Próprio Autor.

Tabela 2 – Descrição da rota Estabelecimentos por Usuário

Estabelecimentos por Usuário

Lista os estabelecimentos que um usuário está associado.

- **URL**

- /usuario/{usuarioNome}/estabelecimentos/

- **Método**

- POST

- **Parâmetros**

- usuario: String - Representa o nome do usuário

- **Resposta de Sucesso**

- Código: 200 OK

- Conteúdo: Array JSON contendo o cnes e nome dos estabelecimentos.

- **Resposta de Erro**

- Código: 200 OK

- Conteúdo: JSON com o seguinte conteúdo:

- {“error”: {“message”:“Usuário inválido”}}

- Código: 403 Forbidden

- Conteúdo: JSON com a chave “message” contendo o valor “Usuário não autenticado”

Fonte: Próprio Autor.

Tabela 3 – Descrição da rota Evoluções Médicas de Pacientes Internados

Evoluções Médicas de Pacientes Internados

Retorna a listagem das evoluções médicas dos pacientes internados em um determinado estabelecimento.

- **URL**

- /evolucao-medica/pacientes-internados

- **Método**

- POST

- **Parâmetros**

- Nenhum parâmetro

- **Corpo**

- estabelecimentoCnes: *String* (Obrigatório)

- **Resposta de Sucesso**

- Código: 200 OK
- Conteúdo: Array *JSON* com as chaves: id, prontuario, paciente_nome e data_evolucao

- **Resposta de Erro**

- Código: 400 Bad Request
- Conteúdo: JSON com o seguinte conteúdo:
{"message": "Estabelecimento Cnes é obrigatório"}
- Código: 403 Forbidden
- Conteúdo: JSON com a chave "message" contendo o valor "Usuário não autenticado"

Fonte: Próprio Autor.

Tabela 4 – Descrição da rota Visualizar Evoluções Médicas

Visualizar Evoluções Médicas

Retorna as evoluções médicas de um determinado paciente.

- **URL**

- /evolucao-medica/{prontuarioid}

- **Método**

- POST

- **Parâmetros**

- prontuarioid: Integer - Número de prontuário do paciente

- **Resposta de Sucesso**

- Código: 200 OK

- Conteúdo: *Array JSON* contendo as informações das evoluções médicas, solicitações de parecer e solicitações de procedimentos.

- **Resposta de Erro**

- Código: 404 Not Found

- Conteúdo: JSON contendo o seguinte valor:

- {“message”:“Not found”}

- Código: 403 Forbidden

- Conteúdo: JSON com a chave “message” contendo o valor “Usuário não autenticado”

Fonte: Próprio Autor.

Tabela 5 – Descrição da rota Buscar Evoluções Médicas

Buscar Evoluções Médicas

Retorna as evoluções médicas correspondentes ao valor preenchidos nos campos paciente e/ou setores e estabelecimento cnes.

- **URL**

- /evolucao-medica/paciente/filtro

- **Método**

- POST

- **Parâmetros**

- Nenhum parâmetro

- **Corpo**

- paciente: String - O conteúdo pode representar um número de prontuário, trechos do nome do paciente ou laudo SIH (Opcional)
- setores: Array - Contêm um Array com os ids dos setores (Opcional)
- estabelecimentoCnes: String - Indica de qual estabelecimento os registros devem retornar (Obrigatório)

- **Resposta de Sucesso**

- Código: 200 OK
- Conteúdo: *Array JSON* contendo as informações das evoluções médicas, solicitações de parecer e solicitações de procedimentos.

- **Resposta de Erro**

- Código: 400 Bad Request
- Conteúdo: JSON com o seguinte conteúdo:
 { "message": "Estabelecimento Cnes é obrigatório" }
- Código: 403 Forbidden
- Conteúdo: JSON com a chave "message" contendo o valor "Usuário não autenticado"

Fonte: Próprio Autor.

4.6 APRESENTAÇÃO DA APLICAÇÃO

Nesta seção serão apresentadas as telas do aplicativo desenvolvido neste trabalho. O designer inicial foi prototipado utilizando a ferramenta *Figma*⁴, tomando como base um *design kit*⁵ da biblioteca *Native Base*⁶. Entretanto, foi verificado que a biblioteca em questão foi descontinuada e por isso não se achou prudente prosseguir utilizando-a. Devido a isso, a biblioteca *React Native Paper*⁷ foi escolhida em substituição à anterior, porém o protótipo foi mantido para servir de base para a implementação do *leiaute*.

Ao iniciar o aplicativo, o usuário visualiza a *splash screen*, onde é apresentada a logo da aplicação, apresentada na Figura 9. Essa tela é comum em aplicativos e possui a função de dar uma resposta ao usuário que algum recurso está em carregamento, enquanto o aplicativo termina algum processamento inicial.

Figura 9 – *Splash Screen*



Fonte: Próprio Autor.

⁴ <https://www.figma.com/>

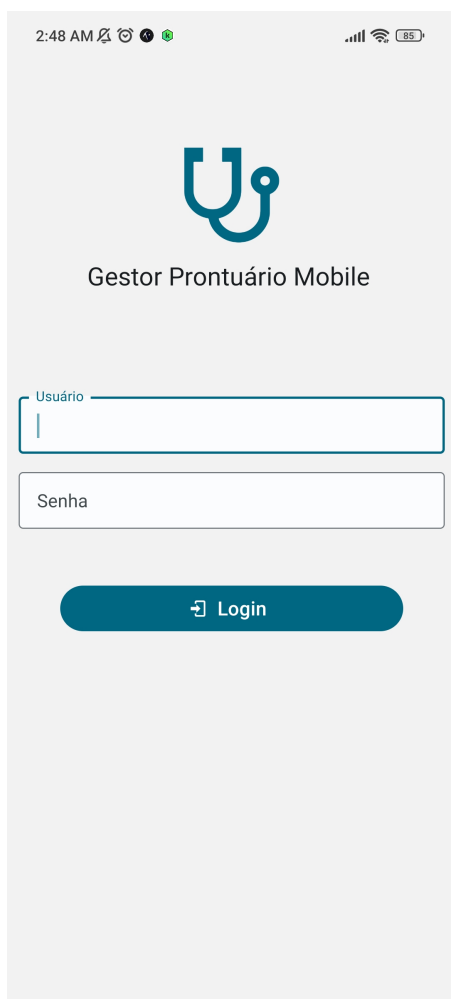
⁵ Pacote organizado de elementos visuais que serve como base para criar interfaces digitais mais rapidamente e consistentemente (Fluxo Estúdio de Design, 2023)

⁶ <https://nativebase.io/>

⁷ <https://callstack.github.io/react-native-paper/>

Após inicializar, o usuário é levado à tela de *login*, onde pode inserir seus dados de acesso ao sistema, conforme mostra a Figura 10. Ao realizar o *login*, o usuário informa o seu nome de usuário e senha e, uma vez validadas suas credenciais, é direcionado à tela de listagem dos estabelecimentos associados à sua conta conforme mostra a Figura 11.

Figura 10 – Tela de Login



2:48 AM

Gestor Prontuário Mobile

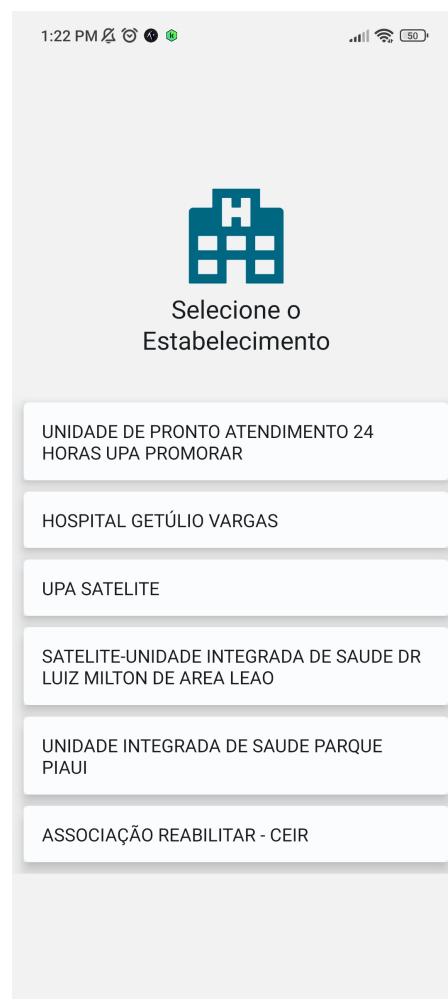
Usuário

Senha

Login

Fonte: Próprio Autor.

Figura 11 – Tela da Lista de Estabelecimentos Vinculados à Conta do Usuário



1:22 PM

Selecione o Estabelecimento

UNIDADE DE PRONTO ATENDIMENTO 24 HORAS UPA PROMORAR

HOSPITAL GETÚLIO VARGAS

UPA SATELITE

SATELITE-UNIDADE INTEGRADA DE SAUDE DR LUIZ MILTON DE AREA LEO

UNIDADE INTEGRADA DE SAUDE PARQUE PIAUI

ASSOCIAÇÃO REABILITAR - CEIR

Fonte: Próprio Autor.

Ao selecionar o estabelecimento, uma janela é aberta, a qual pedirá ao usuário que informe de que posto, setor ou especialidade deseja visualizar as evoluções, conforme a Figura 12. Pelo menos uma dessas informações deverá ser preenchida para que o usuário avance para a tela inicial. A intenção dessa seleção é mostrar ao usuário somente as evoluções pertencentes ao posto, setor ou especialidades selecionadas. Vale destacar que o usuário poderá alterar essas informações quando quiser, por meio da opção contida no menu lateral. Após selecionar as informações e

pressionar o botão “Entrar”, o usuário é levado à tela inicial do aplicativo, no qual contém três *cards* clicáveis referentes ao histórico médico, de enfermagem e multidisciplinar. A tela inicial também contém um botão *hamburger* que dá acesso ao menu lateral. No menu lateral estão contidos o nome do profissional logado no sistema, no qual foi adquirido através do login, o perfil, o nome do estabelecimento e também o posto, especialidade ou setor escolhidos. Essas informações estão armazenadas globalmente, a fim de servirem durante a gravação ou filtro de registros.

As Figuras 13 e 14 apresentam a tela inicial e o menu lateral, respectivamente.

Figura 12 – Janela para Seleção do Posto, Setor ou Especialidade

2:49 AM

Selecione o Posto/Setor

Posto
POSTO 2

Especialidade
Selecione o item

Setor
CLINICO GERAL

CLINICO GERAL

Cancelar Entrar

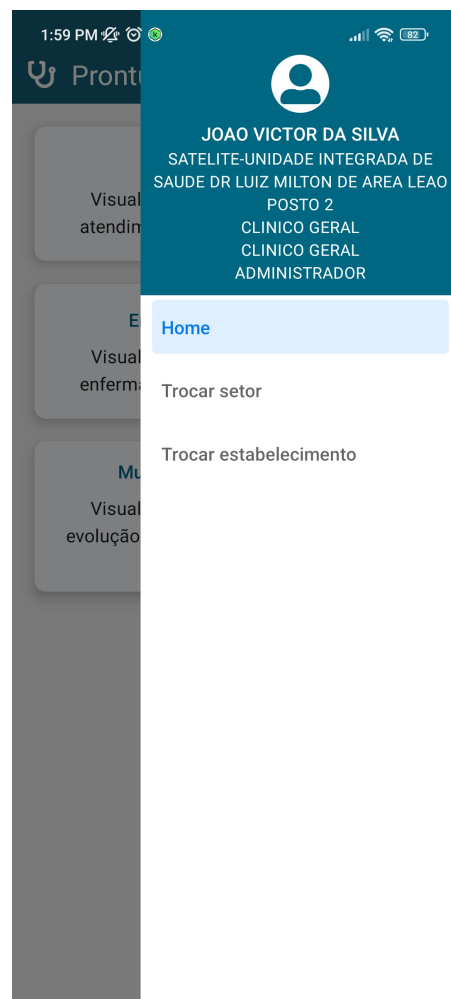
Fonte: Próprio Autor.

Figura 13 – Tela Inicial do Aplicativo



Fonte: Próprio Autor.

Figura 14 – Menu Lateral do Aplicativo



Fonte: Próprio Autor.

Clicando no primeiro *card*, que diz respeito à evolução médica, então o usuário é direcionado à listagem de evoluções médicas dos pacientes internados, conforme a Figura 15.

No *header* da tela há um ícone de lupa, que apertando-o, é aberta uma caixa de texto, no qual o profissional pode filtrar os registros pelo o número do prontuário, laudo de Solicitação de Internação Hospitalar (SIH) ou nome do paciente. Este leiaute se repete também nas evoluções de enfermagem e multidisciplinar.

Figura 15 – Listagem dos Pacientes Internados



Fonte: Próprio Autor.

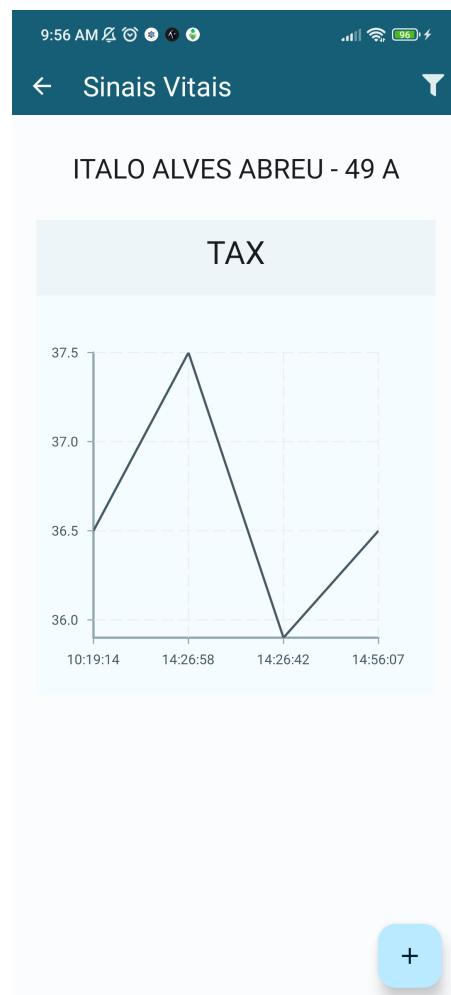
Em cada item da listagem dos pacientes, há dois botões rotulados “Sinais Vitais” e “Solicitações de Visita”. O botão “Sinais Vitais” leva o usuário à tela de histórico dos sinais vitais do paciente selecionado. Esta tela dispõe os dados através de gráficos de linha que representam evolução do paciente durante um período selecionado. Cada sinal vital possui um gráfico, na Figura 17 o sinal de temperatura axilar (Tax) está sendo representado. O eixo horizontal representa o horário da coleta do dados, enquanto o vertical informa o valor coletado. O botão de rótulo “Solicitações de Visita” leva o usuário para a listagem das solicitações já registradas (Figura 16), colocando um ícone de *check* duplo na cor verde para as solicitações visualizadas pelo destinatário (assemelhando-se ao utilizado nos aplicativos de mensagens instantâneas), e um *check* único na cor cinza para as solicitações ainda não visualizadas.

Figura 16 – Tela de Listagem das Solicitações de Visita



Fonte: Próprio Autor.

Figura 17 – Tela contendo o Gráfico de Temperatura Axilar



Fonte: Próprio Autor.

Além disso, cada uma das telas das Figuras 16 e 17 possuem um botão flutuante, o qual leva às telas para solicitar um nova visita ou à adição de sinais vitais, representados, respectivamente pelas Figuras 18 e 20. Na Figura 18, o profissional deverá solicitar os profissionais que pertencem ao CBO (Classificação Brasileira de Ocupações) indicado, por exemplo, um médico clínico geral, enfermeiro, fisioterapeuta, como mostrado na Figura 19. Após selecionar o CBO, o profissional informará a justificativa que o fez realizar a solicitação, vale destacar que todos os campos são obrigatórios. Uma vez preenchidos os campos, o profissional salvará a solicitação clicando no botão de *check* no canto superior direito, então a solicitação é enviada para todos profissionais que correspondem ao CBO especificado.

Figura 18 – Tela para a Solicitação de Visitas

10:33 AM

Solicitação de Visita

66 - Conselho Regional de Enfermagem

Observação

Fonte: Próprio Autor.

Figura 19 – Listagem de CBOs

10:55 AM

Solicitação de Visita

66 - Conselho Regional de Enfermagem

× CBO

- ☒ Médico Clínico Geral
- ☐ Enfermeiro
- ☐ Dentista
- ☐ Fisioterapeuta
- ☐ Técnico de Enfermagem
- ☐ Assistente Social na Área da Saúde
- ☐ Psicólogo Clínico
- ☐ Farmacêutico
- ☐ Nutricionista
- ☐ Terapeuta Ocupacional

Fonte: Próprio Autor.

Figura 20 – Formulário de Adição dos Sinais Vitais

10:33 AM

← Adicionar Sinais Vitais ✓

ITALO ALVES ABREU - 49 A

Data: 26/11/2023

Horário Inicial: 10:33:48

TAX	36.5
Freq. Cardíaca	50
Freq. Respiratória	60
PA Sistólica	12
PA Diastólica	4
Glicemia Capilar	90
Diurese	200
Débitos	80

Fonte: Próprio Autor.

A Figura 20 apresenta o formulário para adição de um novo registro de sinais vitais. Ao acessar a tela, os valores padrões mais comuns já são definidos para cada sinal vital. Além disso, os campos “Data” e “Horário Inicial” já vêm preenchidos com a data e horário atual, respectivamente. Isso foi pensado para agilizar ainda mais o preenchimento do formulário, aliado ao uso de *slides* como entrada dos dados de cada sinal vital. Após o preenchimento do formulário, o usuário poderá gravar os dados pressionando o botão de *check*, localizado no canto superior direito. Realizada a gravação, o usuário é direcionado novamente à tela de histórico de sinais vitais, agora, com o novo registro adicionado aos gráficos.

Seguindo o fluxo da Figura 15, quando um registro de evolução médica é selecionado, todo o histórico de evoluções médicas do paciente é carregado em formato de *cards*, dispostos em ordem cronológica, dos mais recentes aos mais antigos. A Figura 21 representa a evolução propriamente dita e solicitações de parecer. A Figura

22, as solicitações de procedimentos (juntamente com seus resultados, se houver) e, por último, a Figura 23 ilustra as prescrições. É importante ressaltar que nem todos esses elementos são obrigatórios em cada registro, uma vez que refletem o progresso do paciente ao longo de sua internação.

No *header* da tela de evoluções há o nome do paciente e sua idade separados por hífen. A idade é composta pelo número correspondente e a unidade de medida, no caso da Figura 21, em anos (A).

Figura 21 – *Cards* representando a Solicitação de Parecer e a Evolução Médica

Histórico Médico
ITALO ALVES ABREU - 49 A

Solicitação de Parecer - 31/10/2023 15:05:01

Especialidade: CARDIOLOGIA
Data: 31/10/2023

Justificativa
O paciente possui frequentes dores de cabeça

Data da Resposta: 31/10/2023
Horário da Resposta: 15:08:48
Profissional da Resposta: SERGIO ROBERTO SILVA RODRIGUES

Resposta
O paciente possui sopro no coração

Evolução Médica - 31/10/2023 14:38:12

Temperatura: 36 °C **Pulso:** 100 ppm
Peso: 90 Kg
Pressão Arterial: 120 x 80 mmHg
Diurese: 120 ml **Glicemia:** 99 mg/dL
Respiração/Saturação O2: 94%

Evolução Diária
O paciente encontra-se nervoso e com bastante dor de cabeça

Queixas
Dor de cabeça

Fonte: Próprio Autor.

Figura 22 – *Card* representando a Solicitação de Procedimentos

Histórico Médico
ITALO ALVES ABREU - 49 A

Solicitação de Procedimentos - 31/10/2023 14:54:02

0202010287 - DOSAGEM DE COLESTEROL LDL ^
Situação: Aguardando realização **Quant.:** 1

0202020380 - HEMOGRAMA COMPLETO ^
Situação: Aguardando realização **Quant.:** 1

0301010072 - CONSULTA MEDICA EM ATENCAO ESPECIALIZADA ^
Situação: Realizado - Resultado Disponível
Quant.: 1
CBO
LABIO PALATINO - PALATO LABIAL - SEGUIMENTO
Data resultado: 31/10/2023 **Hora:** 14:54:00
Resultado
O paciente possui dificuldades para respirar

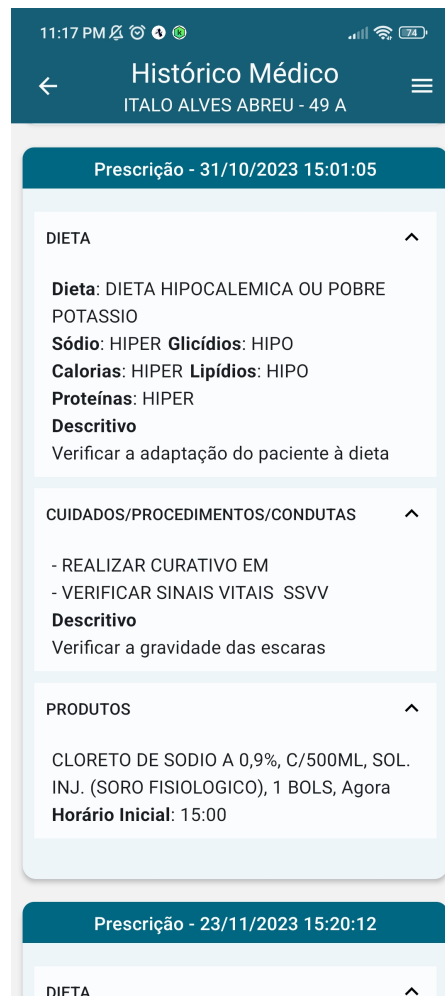
Prescrição - 31/10/2023 15:01:05

DIETA v

CUIDADOS/PROCEDIMENTOS/CONDUTAS v

Fonte: Próprio Autor.

Figura 23 – Card representando a Prescrição Médica



Fonte: Próprio Autor.

Indo para a Evolução de Enfermagem, os registros nele contidos correspondem aos dados da Sistematização da assistência de enfermagem (SAE), no qual organiza o trabalho profissional quanto ao método, pessoal e instrumentos, tornando possível a operacionalização do Processo de Enfermagem (Santos *et al.*, 2016). São dispostos horizontalmente devido ao volume de informações de cada registro, como ilustra a Figura 24 . Adentrando em cada registro, cada seção da evolução é organizada por componentes que se retraem ou expandem, a fim de que o usuário tenha o foco somente nos dados que seja visualizar naquele momento.

Indo para a Evolução de Enfermagem, os registros de evolução são dispostos horizontalmente devido ao volume de informações de cada registro, como ilustra a Figura . Adentrando em cada registro, cada seção da evolução é organizada por componentes que se retraem ou expandem, a fim de que o usuário tenha o foco somente nos dados que seja visualizar naquele momento.

Figura 24 – Card representando a Evolução de Enfermagem



Fonte: Próprio Autor.

Figura 25 – Tela contendo a Seção Cardiovascular/Respiratório e Neurológico



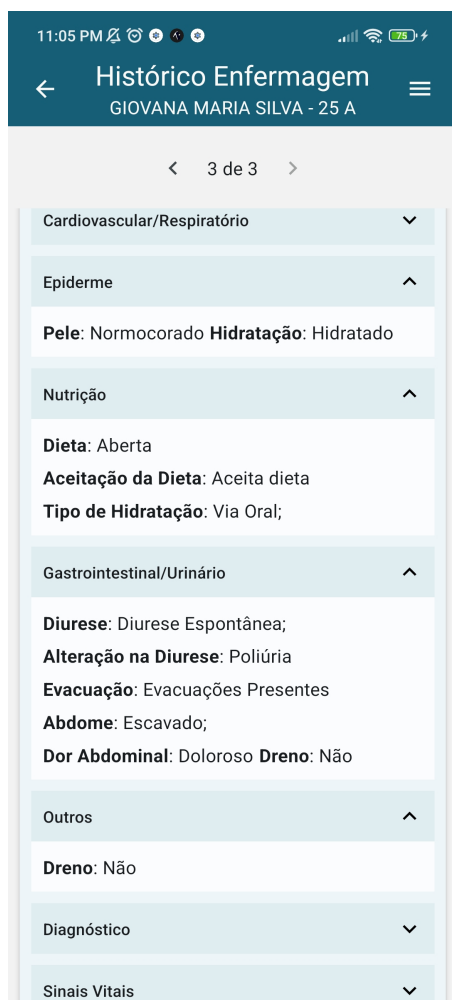
Fonte: Próprio Autor.

Na Figura 25, a seção “Neurológico” contém os campos “Nível de Consciência”, campos relacionados à escala de *Glasgow*⁸, “Locomoção”, “Comunicação” e “Sono”. Ainda na mesma figura é mostrado a seção “Cardiovascular/Respiratório” que contém os campos “Ritmo Cardíaco”, “Respiração”, “Respiração Através”, “Respiração Auxílio”, os três últimos mencionados dizem respeito, respectivamente à alteração nos padrões respiratórios, se o indivíduo respira através de alguma aparelho e se precisa de auxílio de oxigênio. A seção “Epiderme” contém os campos “Pele” e “Hidratação”, que representam informações sobre a pele do indivíduo, como mostradas na . A seção “Nutrição” contém os campos “Dieta”, “Aceitação Dieta” e “Tipos de hidratação”, que informam sobre seu grau de nutrição durante a estadia no hospital. A seção Gastroin-

⁸ É através dessa escala que é possível mensurar o nível de consciência dos pacientes (Ministério da Saúde, 2023)

testinal/Urinário formada pelos campos “Diurese”, “Alteração diurese”, “Evacuações”, “Alteração evacuações”, “Abdome”, “Dor abdome”, representam informações sobre o sistema digestivo e urinário, no que tange à eliminação dos resíduos. A seção “Outros” corresponde aos campos que não adequaram-se às outras seções mencionadas. Os campos contidos nessa seção são “Dreno”, “Localização do dreno”, e “Tipo do dreno”. As seções “Epiderme”, “Nutrição”, “GastroIntestinal/Urinário” e “Outros” estão representadas na Figura 26.

Figura 26 – Seções Epiderme, Nutrição, GastroIntestinal/Urinário e Outros



Fonte: Próprio Autor.

Figura 27 – Seções Diagnóstico e Sinais Vitais



Fonte: Próprio Autor.

A seção “Diagnóstico” é dividida em duas partes: a condição clínica, e os cuidados prescritos para a condição clínica informada. Cada condição clínica é representada por uma seção, que contém os cuidados correspondentes, por exemplo, na Figura 27, para o diagnóstico de “Ansiedade”, há os cuidados “Verificar Sinais Vitais - SSVV” e “Verificar Infusão Venosa”, com seus respectivos campos associados, isto é, “Intervalo

prescrição”, “Turno” e “Horário Início”, o qual informam a frequência que o cuidado deverá ser realizado.

Por fim, a seção “Sinais Vitais” (Figura 27) contém os mesmo campos “Horário”, “Tax”, “Frequência Cardíaca/Respiratória”, “Diurese Volume”, “Diurese Volume”, “Pressão Arterial Sistólica/Diastólica”, “Débitos”, esse último é referente ao débito cardíaco, uma medição do fluxo sanguíneo produzido pelo o coração à cada batimento. Essa seção da SAE, entretanto, é pouco utilizada pelos profissionais, conforme já relatado na seção 1.2.

A evolução multidisciplinar (Figura 28) é mais simples do que as demais, visto que, pelo seu caráter multidisciplinar, é difícil mapear campos que correspondam a todas as áreas relacionadas. Dessa forma, as informações são adicionadas em texto livre pelo o profissional. Essas informações estão representadas pelo o campo “Descrição”.

Figura 28 – Evolução Multidisciplinar

A imagem é uma captura de tela de um aplicativo móvel. No topo, há uma barra de status com o horário 11:40 PM e ícones de bateria, Wi-Fi e sinal. Abaixo, uma barra de cabeçalho azul escuro contém o título "Histórico Multidisciplinar" e o nome da paciente "GIOVANA MARIA SILVA - 25 A". Um botão de voltar (seta esquerda) está à esquerda e um menu hambúrguer (três linhas) à direita. O conteúdo principal é um cartão branco com o título "Registro Multidisciplinar - 01/10/2023". Dentro do cartão, há o nome do profissional "Profissional: SERGIO ROBERTO SILVA RODRIGUES" e um campo rotulado "Descrição" com o texto "A paciente possui dificuldade em comunicar-se".

Fonte: Próprio Autor.

5 CONCLUSÃO

Este trabalho apresenta o projeto Gestor Prontuário Mobile, uma ferramenta útil para profissionais de saúde que utilizam o sistema Gestor Saúde. O aplicativo é uma extensão do sistema mencionado e obtém seus dados a partir dele, uma vez que consome as informações coletadas pelos profissionais por meio do software e as exibe no aplicativo.

Além disso, o aplicativo permite o registro periódico dos sinais vitais, apresentando-os de forma gráfica e estruturada. Isso facilita a visualização e análise dos sinais vitais, tornando-se um critério de avaliação da evolução do paciente entre os profissionais. Outra funcionalidade adicional é a possibilidade de solicitar visitas de profissionais, se necessário. Essas duas funcionalidades não estão presentes no Gestor Saúde e, inicialmente, serão exclusivas do aplicativo, embora os dados sejam armazenados na mesma base de dados que o Gestor Saúde consome.

Através do aplicativo, espera-se resolver o atual problema de mobilidade dos profissionais de saúde já discutido em seções anteriores. A visita à cada leito exige agilidade e eficiência, pois cada profissional tem vários pacientes sob seus cuidados. Quanto melhor e mais eficiente for o processo de visualização do histórico do paciente, maior será a eficiência da análise médica. Além de ajudar os profissionais, o aplicativo também desempenha seu papel social oferecendo mecanismos que facilitam a análise clínica, o diagnóstico médico e o acompanhamento do paciente.

O presente trabalho buscou propor o aplicativo Gestor Prontuário Mobile como solução para uma atual demanda de mobilidade entre profissionais de saúde. Ademais, procurou justificar a sua implementação levando em consideração a sua importância para a sociedade, representada pelos pacientes, que serão contemplados com os benefícios, e também para a empresa Supra Tecnologia, proprietária do produto, visto que não deixa de ser um novo produto que entrará em produção na companhia.

5.1 TRABALHOS FUTUROS

Após a conclusão das funcionalidades propostas neste trabalho, o próximo passo será a disponibilidade do aplicativo nas lojas de aplicativos mais utilizadas em cada plataforma escolhida, isto é *Google Play Store* para *Android* e *App Store*, para *IOS*. Vale destacar que cada loja possui exigências de segurança para que um novo aplicativo possa adentrar nela, por isso, caso a solução, *a priori*, não esteja adequada à essas regras, será refatorada até alcançar o objetivo. Em seguida, será realizada a apresentação e implantação do aplicativo nos hospitais, aliado do suporte aos profissionais de saúde acerca do uso do aplicativo, defendendo o uso da solução

como uma aliada ao seu trabalho e também para melhorar o atendimento aos pacientes. Através do contato dos profissionais com o aplicativo, estes poderão sugerir melhorias e adição de funcionalidades que não foram mapeadas inicialmente. Este *feedback* é valioso pois define o grau de aceitação dos usuários ao aplicativo e estabelece um vínculo de colaboração para o avanço do mesmo.

É notório o avanço da Inteligência Artificial (IA) e nos últimos anos várias soluções vêm surgindo, como é o caso do *ChatGPT*¹ que foi lançada em 2022, e popularizou a experiência de lidar com IA para a resolução de problemas e de igual forma fomentou discussões na área de aprendizado de máquinas e ciência de dados em carácter social, jurídico, acadêmico, entre outros. Na área da saúde, IA já está sendo utilizada há alguns anos, no qual um dos seus usos está no auxílio à tomada de decisão com base na análise de dados coletados, e através dessa análise oferecer resultados que sejam válidos para os profissionais de saúde. Considerando a coleta de sinais vitais presentes na solução proposta, posteriormente pretende-se adicionar ferramentas de IA para análise dos SSVV coletados, permitindo prever agravos do quadro do paciente e notificar os profissionais responsáveis por ele acerca dos riscos, levando a um cuidado direcionado ao estado de saúde do indivíduo e permitindo intervenções precoces a fim de preservar a saúde do mesmo.

¹ ChatGPT é um programa de inteligência artificial que gera diálogos. Criado pela *OpenAI*, o *chatbot* utiliza algoritmos de aprendizado de máquina para processar e analisar grandes quantidades de dados a fim de gerar respostas às perguntas dos usuários. Durante o ano deste documento, o produto está na sua versão 4 (University of Central Arkansas, 2023)

REFERÊNCIAS

- AUTH0. **JSON Web Token Introduction**. 2023. Disponível em: <<https://www.jwt.io/introduction#what-is-json-web-token>>. Acesso em: 11 nov. 2023.
- BEZERRA, S. M. **Prontuário Eletrônico do Paciente: uma ferramenta para aprimorar a qualidade dos Serviços de Saúde**. 2009. Revista Meta: Avaliação, Rio de Janeiro, v. 1, n. 1. Disponível em: <<https://revistas.cesgranrio.org.br/index.php/metaavaliacao/article/view/12>>. Acesso em: 23 jun. 2023.
- BIGIO, M. **Introducing Hot Reloading**. 2016. Disponível em: <<https://reactnative.dev/blog/2016/03/24/introducing-hot-reloading>>. Acesso em: 20 nov. 2023.
- BRASIL. **Resolução-RDC nº 63, de 25 de novembro de 2011**. 2011. Dispõe sobre os Requisitos de Boas Práticas de Funcionamento para os Serviços de Saúde. Disponível em: <https://bvsms.saude.gov.br/bvs/saudelegis/anvisa/2011/rdc0063_25_11_2011.html>. Acesso em: 21 jun. 2023.
- BRASIL. **Lei nº 13.709, de 14 de agosto de 2018: Lei Geral de Proteção de Dados Pessoais (LGPD)**. 2018. Diário Oficial da União: seção 1, Brasília, DF, p. 59. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm>. Acesso em: 20 nov. 2023.
- BRASIL. **Lei nº 13.787, de 27 de dezembro de 2018**. 2018. Dispõe sobre a digitalização e a utilização de sistemas informatizados para a guarda, o armazenamento e o manuseio de prontuário de paciente. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13787.htm>. Acesso em: 21 jun. 2023.
- CONSELHO FEDERAL DE MEDICINA. **Resolução CFM nº 1.638/2002**: Define prontuário médico e torna obrigatória a criação da comissão de revisão de prontuários nas instituições de saúde. 2002. Disponível em: <<https://sistemas.cfm.org.br/normas/visualizar/resolucoes/BR/2002/1638>>. Acesso em: 24 jun. 2023.
- CONSELHO FEDERAL DE MEDICINA. **LGPD: a Lei Geral de Proteção de Dados Pessoais e a atuação do profissional da medicina**. 2022. Disponível em: <https://cdn-flip3d.sflip.com.br/temp_site/issue-fd4d801731725513a4d77aa9bb35534b.pdf>. Acesso em: 23 jun. 2023.
- FLUXO ESTÚDIO DE DESIGN. **O que é kits de design**. 2023. Disponível em: <<https://fluxodesign.com.br/glossario/o-que-e-kits-de-design-importancia/>>. Acesso em: 20 nov. 2023.
- FUNDAÇÃO GETÚLIO VARGAS. **Panorama do Uso de TI no Brasil - 2022**. 2022. Fundação Getúlio Vargas. Disponível em: <<https://portal.fgv.br/artigos/panorama-uso-ti-brasil-2022>>. Acesso em: 22 jun. 2023.
- LURACAST. **Restler: Simple and effective multi-format Web API Server**. 2023. Disponível em: <<https://github.com/Luracast/Restler>>. Acesso em: 28 nov. 2023.

MARIN, H. de F.; MASSAD, E.; NETO, R. S. de A. Prontuário eletrônico do paciente: Definições e conceitos. **O prontuário eletrônico do paciente na assistência, informação e conhecimento médico**, v. 322, n. 10, p. 1–15, 2003. Disponível em: <http://www.sbis.org.br/biblioteca_virtual/prontuario.pdf>. Acesso em: 21 jun. 2023.

META PLATFORMS. **About the New Architecture - React Native**. 2023. Disponível em: <<https://reactnative.dev/docs/next/the-new-architecture/why>>. Acesso em: 20 nov. 2023.

META PLATFORMS. **Cross Platform Implementation - React Native**. 2023. Disponível em: <<https://reactnative.dev/architecture/xplat-implementation>>. Acesso em: 27 nov. 2023.

META PLATFORMS. **React Native: Learn once, write anywhere**. 2023. Disponível em: <<https://reactnative.dev/>>. Acesso em: 20 nov. 2023.

MINISTÉRIO DA SAÚDE. **Escala de Coma de Glasgow**. 2023. Disponível em: <[https://linhasdecuidado.saude.gov.br/portal/acidente-vascular-cerebral-\(AVC\)-no-adulto/glasgow](https://linhasdecuidado.saude.gov.br/portal/acidente-vascular-cerebral-(AVC)-no-adulto/glasgow)>. Acesso em: 27 nov. 2023.

NEGRI, F. D. **As tecnologias da informação podem revolucionar o cuidado com a Saúde?** 2020. IPEA. Disponível em: <<https://www.ipea.gov.br/cts/pt/central-de-conteudo/artigos/artigos/107-as-tecnologias-da-informacao-podem-revolucionar-o-cuidado-com-a-saude>>. Acesso em: 21 jun. 2023.

OKAFOR, N. **React Native's New Renderer: Fabric**. 2022. Disponível em: <https://dev.to/theadultnoble_6/react-natives-new-renderer-fabric-5898>. Acesso em: 20 nov. 2023.

OLIVEIRA, G. N. *et al.* **Alteração de sinais vitais e desfecho clínico de pacientes admitidos em unidade de emergência**. 2022. Fundação Getúlio Vargas. Disponível em: <<https://periodicos.ufsm.br/reufsm/article/view/42559/html>>. Acesso em: 22 jun. 2023.

PEÇA, C. M. K. **Análise e interpretação de tabelas e gráficos estatísticos utilizando dados interdisciplinares**. Curitiba: [s.n.], 2008. Programa de Desenvolvimento Educacional – PDE, Secretaria de Estado da Educação do Paraná. Orientação: Profa. MSc. Simone Crocetti – UTFPR. Disponível em: <<https://www.diaadiaeducacao.pr.gov.br/portals/pde/arquivos/1663-8.pdf>>. Acesso em: 23 jun. 2023.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ. **Conselhos profissionais**. 2023. Disponível em: <<https://ead.pucpr.br/blog/conselhos-profissionais>>. Acesso em: 23 nov. 2023.

POSTGRES SQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL: The World's Most Advanced Open Source Database**. 2023. Disponível em: <<https://www.postgresql.org/>>. Acesso em: 27 nov. 2023.

PROSTT, M. E. Interface web utilizando design responsivo: um estudo de caso aplicado a smartphones, tablets, computadores e televisores. **Rev. Latino-Am. Enfermagem**,

2013. Disponível em: <<http://repositorio.utfpr.edu.br/jspui/handle/1/19887>>. Acesso em: 24 jun. 2023.

RED HAT. **API REST - O que é API REST?** 2023. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>. Acesso em: 20 nov. 2023.

SAJDAK, M. **JWT (JSON Web Token) (in)security**. 2019. Disponível em: <<https://research.securitum.com/jwt-json-web-token-security/>>. Acesso em: 11 nov. 2023.

SANTOS, I. M. F. *et al.* Sae - sistematização da assistência de enfermagem: Guia prático. **Conselho Regional de Enfermagem da Bahia**, 2016. Disponível em: <<https://biblioteca.cofen.gov.br/sistematizacao-assistencia-enfermagem-guia-pratico/>>. Acesso em: 23 jun. 2023.

SECRETARIA DE ESTADO DA SAÚDE DO ESPÍRITO SANTO. **SAME arquiva mais de 300 mil documentos em papel**. 2014. <<https://saude.es.gov.br/same-arquiva-mais-de-300-mil-documentos-em-pa>>. Acesso em: 20 nov. 2023.

SILVA, A. d. A. P. d. **Design responsivo: técnicas, frameworks e ferramentas**. 2014. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro. Orientador: Mariano Pimentel. Disponível em: <<https://bsi.uniriotec.br/wp-content/uploads/sites/31/2020/05/201412Almeida.pdf>>. Acesso em: 24 jun. 2023.

SILVA, C. R. da. História do prontuário médico: Evolução do prontuário médico tradicional ao prontuário eletrônico do paciente – pep. **Research, Society and Development**, p. 7, 2021. Disponível em: <<https://rsdjournal.org/index.php/rsd/article/view/18031>>. Acesso em: 23 jun. 2023.

SOMMERVILLE, I. **Engenharia de Software**. [S.l.]: Pearson Education, 2011.

SOUZA, B. T. *et al.* Identificação dos sinais de alerta para a prevenção da parada cardiorrespiratória intra-hospitalar*. **Rev. Latino-Am. Enfermagem**, 2019. Disponível em: <<https://www.scielo.br/j/rlae/a/jfkTNXZ5BwjrqHmGJtBFzKQ/?lang=en>>. Acesso em: 21 jun. 2023.

STATCOUNTER. **Desktop vs Mobile vs Tablet Market Share Worldwide**. 2023. Disponível em: <<https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/#monthly-202203-202303>>.

THE PHP GROUP. **PHP: Hypertext Preprocessor**. 2023. Disponível em: <<https://www.php.net/>>. Acesso em: 26 nov. 2023.

TONELLO, I. M. S.; NUNES, R. M. da S.; PANARO, A. P. Prontuário do paciente: a questão do sigilo e a lei de acesso à informação. **Informação & informação**, v. 18, n. 2, p. 193–210, 2013. Disponível em: <<https://ojs.uel.br/revistas/uel/index.php/informacao/article/view/16169>>. Acesso em: 24 nov. 2023.

UNIVERSITY OF CENTRAL ARKANSAS. **Chat GPT: What is it?** 2023. Disponível em: <<https://uca.edu/cetal/chat-gpt/>>. Acesso em: 22 nov. 2023.

WESOLOWSKI, E. N. **Melhoria no processo administrativo do prontuário do paciente (PEP) para otimização do atendimento nas unidades de atenção básica na área de saúde**. Curitiba: [s.n.], 2016. Trabalho de Especialização (Gestão Pública Municipal) – Universidade Federal do Paraná. Orientador: Joel Souza e Silva. Disponível em: <<https://acervodigital.ufpr.br/handle/1884/51444>>. Acesso em: 23 jun. 2023.

WRIKE. **What is the Scrum methodology?** 2023. Disponível em: <<https://www.wrike.com/scrum-guide/scrum-methodology/>>. Acesso em: 22 nov. 2023.

ANEXO A – ESPECIFICAÇÃO DOS CASOS DE USO

Caso de Uso 01

- **Caso de Uso:** Visualizar histórico médico
- **Atores:** Profissional de Saúde
- **Precondições:** O profissional deve estar autenticado no sistema e realizado a filtragem do paciente
- **Pós-condições:** Visualização do histórico do paciente pesquisado
- **Fluxo Principal:**
 1. O profissional loga no sistema;
 2. O profissional seleciona o estabelecimento no qual deseja visualizar os pacientes;
 3. O profissional seleciona o *card* “Médica”, na tela inicial da aplicação;
 4. O profissional seleciona o setor a fim de iniciar uma pré-filtragem dos pacientes;
 5. O profissional procura um paciente filtrando o setor, número de prontuário, laudo SIH ou nome do mesmo.
 6. Após a filtragem o profissional poderá selecionar o item retornado;
 7. O profissional tem acesso ao histórico médico do paciente.
- **Fluxos Alternativos:**
 - Se a comunicação entre o aplicativo e o sistema falhar:
 1. Uma mensagem será exibida ao usuário com o seguinte conteúdo: “Comunicação com o servidor não estabelecida”.
 - Profissional não possui nenhum estabelecimento associado a seu perfil:
 1. Uma mensagem será exibida ao usuário com o seguinte conteúdo: “Usuário sem estabelecimentos associados para perfil hospitalar”.
 2. Retorna ao fluxo principal, passo 1.
 - Estabelecimento escolhido pelo profissional não possui setores associados:
 1. O fluxo principal será seguido ignorando o passo 4;

2. A pré-filtragem mencionada no passo 5 do fluxo principal ocorrerá sem condicionar o resultado pelo setor.
- Nenhum paciente é retornado na filtragem:
 1. Uma mensagem será exibida ao profissional com o seguinte conteúdo:
“Nenhum paciente encontrado”.

Caso de Uso 02

- **Caso de Uso:** Visualizar solicitações de visita
- **Atores:** Profissional de Saúde
- **Precondições:** O profissional deve estar autenticado no sistema
- **Pós-condições:** Visualização das visitas requisitadas ao profissional autenticado
- **Fluxo Principal:**
 1. O profissional loga no sistema;
 2. O profissional seleciona o estabelecimento no qual deseja visualizar os pacientes;
 3. O profissional abre a tela de Solicitações de Visitas, na tela inicial da aplicação;
 4. O profissional tem acesso às solicitações de visita, dos últimos 30 dias, associadas ao seu código de conselho.
- **Fluxos Alternativos:**
 - Se a comunicação entre o aplicativo e o sistema falhar:
 1. Uma mensagem será exibida ao usuário com o seguinte conteúdo:
“Comunicação com o servidor não estabelecida”.
 - Profissional não possui nenhuma solicitação de visita associado a seu perfil:
 1. É apresentada um tela de *fallback* informando que não há solicitações para o profissional.

Caso de Uso 03

- **Caso de Uso:** Registrar sinais vitais
- **Atores:** Profissional de Saúde
- **Precondições:** Caso de Uso 01

- **Pós-condições:** Um novo registro de sinal vital é associado ao paciente
- **Fluxo Principal:**
 1. O profissional seleciona o botão para adicionar sinais vitais;
 2. O profissional preenche os campos de sinais vitais presentes no formulário;
 3. O profissional salva os dados preenchidos;
 4. O registro é criado;
 5. O profissional é notificado sobre a criação do registro.
- **Fluxos Alternativos:**
 - O profissional descartar as informações preenchidas:
 1. O formulário de criação dos sinais vitais é fechado;
 2. As informações preenchidas não são gravadas;
 3. Retorna para o passo 7 do Caso de Uso 01.
 - Caso exista algum erro de validação de entradas:
 1. Todos os erros serão retornados ao usuário;
 2. Retorna para o passo 2.

Caso de Uso 04

- **Caso de Uso:** Solicitar visita
- **Atores:** Profissional de Saúde
- **Precondições:** Caso de uso 01
- **Pós-condições:** Notificar profissionais pertencentes ao código de conselho selecionado pelo o ator, sobre registro criado
- **Fluxo Principal:**
 1. O profissional seleciona o botão para solicitar visita;
 2. O profissional preenche os campo Conselho de Classe e Observação;
 3. O profissional salva os dados preenchidos;
 4. O registro é criado;
 5. O profissional é notificado sobre a criação do registro;
 6. Os profissionais pertencentes ao conselho de classe são notificados.
- **Fluxos Alternativos:**

- O profissional descartar as informações preenchidas:
 1. O formulário de criação das solicitações de visitas é fechado;
 2. As informações preenchidas não são gravadas;
 3. Retorna para o passo 7 do Caso de Uso 01.
- Caso exista algum erro de validação de entradas:
 1. Todos os erros serão retornados ao usuário;
 2. Retorna para o passo 2.

Caso de Uso 05

- **Caso de Uso:** Visualizar gráfico de sinais vitais
- **Atores:** Profissional de Saúde
- **Precondições:** Caso de uso 01
- **Pós-condições:** Exibir gráfico sobre a evolução do paciente em função dos sinais vitais coletados
- **Fluxo Principal:**
 1. O profissional seleciona o botão para abrir o gráfico;
 2. O profissional visualiza os sinais vitais.
- **Fluxos Alternativos:**
 - O paciente não possui sinais vitais coletados:
 1. Uma tela de *fallback* é exibida ao profissional, informando que não há sinais coletados para o paciente.

Caso de Uso 06

- **Caso de Uso:** Visualizar histórico de enfermagem
- **Atores:** Profissional de Saúde
- **Precondições:** O profissional deve estar autenticado no sistema e realizado a filtragem do paciente
- **Pós-condições:** Visualização do histórico de enfermagem do paciente pesquisado
- **Fluxo Principal:**

1. O profissional loga no sistema;
2. O profissional seleciona o estabelecimento no qual deseja visualizar os pacientes;
3. O profissional seleciona o *card* “Enfermagem”, na tela inicial da aplicação;
4. O profissional procura um paciente filtrando o setor, número de prontuário, laudo SIH ou nome do mesmo.
5. Após a filtragem o profissional poderá selecionar o item retornado;
6. O profissional tem acesso ao histórico de enfermagem do paciente.

- **Fluxos Alternativos:**

- Se a comunicação entre o aplicativo e o sistema falhar:
 1. Uma mensagem será exibida ao usuário com o seguinte conteúdo: “Comunicação com o servidor não estabelecida”.
- Profissional não possui nenhum estabelecimento associado a seu perfil:
 1. Uma mensagem será exibida ao usuário com o seguinte conteúdo: “Usuário sem estabelecimentos associados para perfil hospitalar”;
 2. Retorna ao fluxo principal, passo 1.
- Nenhum paciente é retornado na filtragem:
 1. Uma mensagem será exibida ao profissional com o seguinte conteúdo: “Nenhum paciente encontrado”.

Caso de Uso 07

- **Caso de Uso:** Visualizar histórico multidisciplinar
- **Atores:** Profissional de Saúde
- **Precondições:** O profissional deve estar autenticado no sistema e realizado a filtragem do paciente
- **Pós-condições:** Visualização do histórico multidisciplinar do paciente pesquisado
- **Fluxo Principal:**
 1. O profissional loga no sistema;
 2. O profissional seleciona o estabelecimento no qual deseja visualizar os pacientes;
 3. O profissional seleciona o *card* “Multidisciplinar”, na tela inicial da aplicação;

4. O profissional procura um paciente filtrando o setor, número de prontuário, laudo SIH ou nome do mesmo.
5. Após a filtragem o profissional poderá selecionar o item retornado;
6. O profissional tem acesso ao histórico multidisciplinar do paciente.

• **Fluxos Alternativos:**

- Se a comunicação entre o aplicativo e o sistema falhar:
 1. Uma mensagem será exibida ao usuário com o seguinte conteúdo:
“Comunicação com o servidor não estabelecida”.
- Profissional não possui nenhum estabelecimento associado a seu perfil:
 1. Uma mensagem será exibida ao usuário com o seguinte conteúdo:
“Usuário sem estabelecimentos associados para perfil hospitalar”;
 2. Retorna ao fluxo principal, passo 1.
- Nenhum paciente é retornado na filtragem:
 1. Uma mensagem será exibida ao profissional com o seguinte conteúdo:
“Nenhum paciente encontrado”.

ANEXO B – DOCUMENTAÇÃO DA API REST

Buscar evoluções de enfermagem

Retorna as evoluções de enfermagem correspondentes ao valor preenchido no campo de pesquisa “paciente”.

- **URL**

- /evolucao-enfermagem/paciente/filtro

- **Método**

- POST

- **Parâmetros**

- Nenhum parâmetro

- **Corpo**

- estabelecimentoCnes: String
- paciente: String - Valores correspondentes ao nome do paciente ou número de prontuário (Opcional)

- **Resposta de Sucesso**

- Código: 200 OK
- Conteúdo: JSON contendo a chave “pacientes” formada por um Array com elementos da seguinte estrutura:

```
{“id”: Integer, “data”: String, “data_evolucao”: String, “paciente_nome”:  
String, “prontuario”: String, “leito_nome”: String, “nome_mae”: String,  
“data_nascimento”: String, “paciente_idade”: Integer}
```

- **Resposta de Erro**

- Código: 500 Erro Interno do Servidor
- Conteúdo: JSON com o seguinte conteúdo:

```
{“status”:“error”,“message”:“Não foi possível completar a requisição”}
```

Retornar Evoluções de Enfermagem do Paciente

Retorna as evoluções de enfermagem referentes ao prontuário dado.

- **URL**
 - /evolucao-enfermagem/{prontuarioid}
- **Método**
 - POST
- **Parâmetros**
 - prontuarioid: String - Representa o número de prontuário do paciente.
- **Resposta de Sucesso**
 - Código: 200 OK
 - Conteúdo: Array JSON contendo as evoluções de enfermagem em ordem cronológica a partir do número de prontuário informado no parâmetro.
- **Resposta de Erro**
 - Código: 500 Erro Interno do Servidor
 - Conteúdo: JSON com o seguinte conteúdo:

```
{“status”:“error”,“message”:“Não foi possível completar a requisição”}
```

Buscar evoluções multidisciplinares

Retorna as evoluções multidisciplinares correspondentes ao valor preenchidos no campo de pesquisa “paciente”.

- **URL**

- /evolucao-multidisciplinar/paciente/filtro

- **Método**

- POST

- **Parâmetros**

- Nenhum parâmetro

- **Corpo**

- estabelecimentoCnes: String
- paciente: String - Valores correspondentes ao nome do paciente ou número de prontuário (Opcional)

- **Resposta de Sucesso**

- Código: 200 OK
- Conteúdo: JSON contendo a chave “pacientes” formada por um Array com elementos da seguinte estrutura:

```
{“id”: Integer, “data_evolucao”: String, “paciente_nome”: String,
  “prontuario”: String, “leito_nome”: String, “nome_mae”: String,
  “data_nascimento”: String, “prontuario”: String, “paciente_idade”: Integer}
```

- **Resposta de Erro**

- Código: 500 Erro Interno do Servidor
- Conteúdo: JSON com o seguinte conteúdo:

```
{“status”:“error”,“message”:“Não foi possível completar a requisição”}
```

Retornar Evoluções Multidisciplinar do Paciente

Retorna as evoluções multidisciplinares referentes ao prontuário dado.

- **URL**
 - /evolucao-multidisciplinar/{prontuarioid}
- **Método**
 - POST
- **Parâmetros**
 - prontuarioid: String - Representa o número de prontuário do paciente.
- **Resposta de Sucesso**
 - Código: 200 OK
 - Conteúdo: Array JSON contendo as evoluções multidisciplinares em ordem cronológica com a seguinte estrutura:

```
{“id”: Integer, “data_evolucao”: String, “profissional”: String, “descricao”: String}
```
- **Resposta de Erro**
 - Código: 500 Erro Interno do Servidor
 - Conteúdo: JSON com o seguinte conteúdo:

```
{“status”:“error”, “message”:“Não foi possível completar a requisição”}
```

Adicionar Sinais Vitais do Paciente

Adiciona um registro de sinal vital ao paciente.

- **URL**
 - sinais-vitais/adicionar
- **Método**
 - POST
- **Parâmetros**
 - Nenhum parâmetro.
- **Corpo**
 - prontuarioAtendimentoId: String
 - tax: Float
 - frequenciaCardiaca: Float
 - frequenciaRespiratoria: Float
 - paSistolica: Float
 - paDiastolica: Float
 - diureseVolume: Float
 - glicemiaCapilar: Float
 - debito: Float
 - outrosDebitos: String
 - data: String
 - horario: String
 - usuarioCadastro: Integer
 - usuarioCadastroCnes: String
 - usuarioAlterar: Integer
 - usuarioAlterarCnes: String
 - dataCadastro: String
 - dataAlterar: String

Adicionar Sinais Vitais do Paciente

- **Resposta de Sucesso**

- Código: 200 OK
- Conteúdo: JSON retornando o novo registro adicionado.

- **Resposta de Erro**

- Código: 500 Erro Interno do Servidor
- Conteúdo: JSON com o seguinte conteúdo:
`{"status": "error", "message": "Não foi possível completar a requisição"}`

Retornar Sinais Vitais do Paciente

Retorna os sinais vitais referentes ao prontuário dado.

- **URL**

- /sinais-vitais/{prontuarioid}

- **Método**

- GET

- **Parâmetros**

- prontuarioid: String - Representa o número de prontuário do paciente.

- **Resposta de Sucesso**

- Código: 200 OK
- Conteúdo: Array JSON contendo os sinais vitais em ordem cronológica a partir do número de prontuário informado no parâmetro.

- **Resposta de Erro**

- Código: 500 Erro Interno do Servidor
- Conteúdo: JSON com o seguinte conteúdo:
`{"status": "error", "message": "Não foi possível completar a requisição"}`

Solicitar Visita

Adiciona uma solicitação de visita.

- **URL**
 - solicitacao-visita/adicionar
- **Método**
 - POST
- **Parâmetros**
 - Nenhum parâmetro.
- **Corpo**
 - prontuarioAtendimentoId: String
 - cboCodigo: String
 - descricao: String
 - usuarioCadastro: Integer
 - usuarioCadastroCnes: String
 - usuarioAlterar: Integer
 - usuarioAlterarCnes: String
 - dataCadastro: String
 - dataAlterar: String
- **Resposta de Sucesso**
 - Código: 200 OK
 - Conteúdo: JSON retornando o novo registro adicionado.
- **Resposta de Erro**
 - Código: 500 Erro Interno do Servidor
 - Conteúdo: JSON com o seguinte conteúdo:

```
{“status”:“error”,“message”:“Não foi possível completar a requisição”}
```

Atualizar solicitação de visita

Atualizar o registro, setando valores para colunas não preenchidas na criação, como a coluna “lida”, o qual seu valor é alterado para *true* quando o usuário destinatário lê a solicitação.

- **URL**

- solicitacao-visita/

- **Método**

- PUT

- **Parâmetros**

- Nenhum parâmetro.

- **Corpo**

- solicitacaoVisitaId: Integer
- lida: Boolean
- usuarioAltera: Integer
- usuarioAlteraCnes: String
- dataAltera: String

- **Resposta de Sucesso**

- Código: 200 OK
- Conteúdo: JSON retornando o registro atualizado.

- **Resposta de Erro**

- Código: 500 Erro Interno do Servidor
- Conteúdo: JSON com o seguinte conteúdo:

```
{“status”:“error”,“message”:“Não foi possível completar a requisição”}
```

Retornar Solicitação de Visita do Paciente

Retorna as solicitações de visita do paciente em ordem cronológica.

- **URL**
 - /solicitacao-visita/{prontuarioid}
- **Método**
 - GET
- **Parâmetros**
 - prontuarioid: String - Representa o número de prontuário do paciente.
- **Resposta de Sucesso**
 - Código: 200 OK
 - Conteúdo: Array JSON contendo as solicitações em ordem cronológica a partir do número de prontuário informado no parâmetro.
- **Resposta de Erro**
 - Código: 500 Erro Interno do Servidor
 - Conteúdo: JSON com o seguinte conteúdo:

```
{“status”:“error”,“message”:“Não foi possível completar a requisição”}
```