



UNIVERSIDADE FEDERAL DE ALAGOAS – UFAL
CAMPUS ARAPIRACA – UNIDADE EDUCACIONAL DE PENEDO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

RENDRIKSON DE OLIVEIRA SOARES

**O USO DA TECNOLOGIA BLOCKCHAIN NO DESENVOLVIMENTO DE
APLICAÇÕES DE SAÚDE: UMA ABORDAGEM PRÁTICA PARA GARANTIR A
SEGURANÇA NO ARMAZENAMENTO DE DADOS E O ACESSO
PERMISSIONADO ÀS INFORMAÇÕES CLÍNICAS DO PACIENTE**

PENEDO

2022



UNIVERSIDADE FEDERAL DE ALAGOAS – UFAL
CAMPUS ARAPIRACA – UNIDADE EDUCACIONAL DE PENEDO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

RENDRIKSON DE OLIVEIRA SOARES

**O USO DA TECNOLOGIA BLOCKCHAIN NO DESENVOLVIMENTO DE
APLICAÇÕES DE SAÚDE: UMA ABORDAGEM PRÁTICA PARA GARANTIR A
SEGURANÇA NO ARMAZENAMENTO DE DADOS E O ACESSO
PERMISSIONADO ÀS INFORMAÇÕES CLÍNICAS DO PACIENTE**

Monografia apresentada como requisito parcial
para obtenção do grau de Bacharel em Sistemas
de Informação da Universidade Federal de
Alagoas – Unidade Educacional de Penedo.

Orientador: Prof. Dr. André Magno Costa de
Araújo

PENEDO

2022



Universidade Federal de Alagoas – UFAL
Campus Arapiraca
Unidade Educacional Penedo
Biblioteca Setorial Penedo-BSP

S676u	<p>Soares, Rendrikson de Oliveira</p> <p>O uso da tecnologia blockchain no desenvolvimento de aplicações de saúde: uma abordagem prática para garantir a segurança no armazenamento de dados e o acesso permissionado às informações clínicas do paciente / Rendrikson de Oliveira Soares. – Penedo, AL, 2022.</p> <p>52 f.: il.</p> <p>Orientador(a): Prof. Dr. André Magno Costa de Araújo.</p> <p>Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Universidade Federal de Alagoas, <i>Campus</i> Arapiraca, Unidade Educacional Penedo, Penedo, AL, 2022.</p> <p>Disponível em: Universidade Digital (UD) – UFAL (<i>Campus</i> Arapiraca).</p> <p>Referências: f. 49-52.</p> <p>1. Arquitetura de software. 2. Blockchain. 3. Registro Eletrônico de Saúde. 4. Segurança de dados. I. Araújo, André Magno Costa de. II. Título.</p> <p>CDU 004</p>
-------	---

Rendrikson de Oliveira Soares

O USO DA TECNOLOGIA BLOCKCHAIN NO DESENVOLVIMENTO DE
APLICAÇÕES DE SAÚDE: UMA ABORDAGEM PRÁTICA PARA GARANTIR A
SEGURANÇA NO ARMAZENAMENTO DE DADOS E O ACESSO
PERMISSIONADO ÀS INFORMAÇÕES CLÍNICAS DO PACIENTE

Monografia apresentada como requisito parcial
para obtenção do grau de Bacharel em Sistemas
de Informação da Universidade Federal de
Alagoas – Unidade Educacional de Penedo.

Data de Aprovação: 04/02/2022

Banca Examinadora

Documento assinado digitalmente
 Andre Magno Costa de Araujo
Data: 18/02/2022 16:13:02-0300
Verifique em <https://verificador.iti.br>

Prof. Dr. André Magno Costa de Araújo

Universidade Federal de Alagoas – UFAL

Campus Arapiraca – Unidade Educacional de Penedo

Orientador

Documento assinado digitalmente
 Davy de Medeiros Baía
Data: 21/02/2022 17:46:39-0300
Verifique em <https://verificador.iti.br>

Prof. Dr. Davy de Medeiros Baía

Universidade Federal de Alagoas – UFAL

Campus Arapiraca – Unidade Educacional de Penedo

Examinador

Documento assinado digitalmente
 ALEXANDRE JOSE BARROS MACHADO
Data: 18/02/2022 21:41:04-0300
Verifique em <https://verificador.iti.br>

Prof. Me. Alexandre José Barros Machado

Universidade Federal de Alagoas – UFAL

Campus Arapiraca – Unidade Educacional de Penedo

Examinador

Agradecimentos

Inicialmente, gostaria de agradecer aos meus pais José Renato Soares e Rosemary de Oliveira Soares, e ao meu irmão Rodrigo de Oliveira Soares, que possibilitaram a realização de um sonho e sempre estiveram ao meu lado prestando total apoio nessa jornada.

Agradeço a minha namorada Emanuelle Rodrigues de Amorim do Anjos que sempre teve compreensão e paciência durante essa trajetória, e me deu forças para seguir firme na realização desse sonho.

Agradeço a todos os professores que fizeram parte da minha vida acadêmica, em especial, o meu orientador André Magno Costa de Araújo. Os seus ensinamentos, conselhos e paciência guiaram o meu aprendizado.

Aos meus amigos, José Vitor de Abreu Silva, Matheus de Melo Surica, Gabriel Roberto Rodrigues, Henrique de Couto Melo e Jose dos Santos Neto, por todo apoio e ajuda demonstrado ao longo de todos os períodos em que me dediquei a esse curso.

Por fim, agradeço a todos que participaram, direta ou indiretamente do desenvolvimento deste trabalho de pesquisa, enriquecendo o meu processo de aprendizado.

Resumo

Com o aumento da digitalização das informações em ambientes de saúde, a necessidade de soluções computacionais com recursos de confiabilidade, autenticidade e integridade dos dados clínicos, motivou várias pesquisas que visam solucionar esse problema. Desse modo, este trabalho de conclusão de curso tem como objetivos desenvolver uma solução computacional capaz de auxiliar as organizações de saúde nos atendimentos clínicos, utilizando o armazenamento na tecnologia Blockchain. Para o desenvolvimento da solução proposta foi desenvolvida uma arquitetura de software que proporciona maior confiabilidade, imutabilidade e autenticidade nas informações clínicas fazendo uso da persistência poliglota de dados. Para o armazenamento das informações foram utilizadas duas bases de dados. A primeira, consiste em uma base de dados relacional comumente utilizada em sistemas de saúde, enquanto a segunda, faz uso de uma rede Blockchain para possibilitar maior segurança dos dados salvos na aplicação. Para avaliar a solução computacional proposta, foi desenvolvido um ambiente de teste capaz de simular um cenário de saúde por meio da execução de cargas de dados para o armazenamento na aplicação de software. Como resultado foi obtida a validação do serviço de persistência de dados que se mostrou eficiente e não apresentou divergências nas avaliações realizadas.

Palavras-chaves: Aplicações de Saúde, Blockchain, Registro Eletrônico de Saúde, Arquitetura de software.

Abstract

With the increase in the digitization of information in health environments, the need for computational solutions with reliability, authenticity and integrity of clinical data, motivated several researches that aim to solve this problem. Thus, this course conclusion work aims to develop a computational solution capable of assisting health organizations in clinical care, using Blockchain storage. For the development of the proposed solution, a software architecture was developed that provides greater reliability, immutability and authenticity in clinical information using polyglot persistence. Two databases were used to store the information. The first consists of a relational database commonly used in health systems, while the second consists of a Blockchain network to enable greater security of the data saved in the application. In order to evaluate the computational solution, a test environment capable of simulating a real scenario was developed by executing loads of data for storage in the application. As a result, the validation of the data persistence service was obtained, which proved to be efficient and did not present divergences in the evaluations performed.

Keywords: Healthcare Applications, Blockchain, Electronic Health Record, Software Architecture.

Lista de Figuras

Figura 1 - Arquitetura de RES mínimo com padrões OpenEHR.....	16
Figura 2 - Lista encadeada de Blocos na rede Blockchain.....	17
Figura 3 - Composição de um Bloco Blockchain.	18
Figura 4 - transações de registros de blocos Blockchain	19
Figura 5 - Transações com Contratos Tradicionais	20
Figura 6 - Transações com Contratos Inteligentes	20
Figura 8 – Diagrama de Caso de Uso do médico	27
Figura 9 – Diagrama de Caso de Uso do Paciente	28
Figura 10 - Diagrama de sequência de uma das funcionalidades da aplicação	28
Figura 11 – Interface Gráfica da Consulta.....	30
Figura 12 - Padrão das funções chamadas nas rotas.....	31
Figura 13 - Estrutura de uma rota com autenticação.....	32
Figura 14 - Lógica da função de autenticação.....	32
Figura 15 - Tabelas da base de dados relacional	33
Figura 16 - Informações do contrato após a persistência na rede Blockchain .	35
Figura 17 - Função de criação do contrato inteligente das consultas.....	35
Figura 18 - Fluxo de dados para o permissionamento do contrato	36
Figura 19 - Componente React JS	38
Figura 20 - Função de criação e armazenamento do contrato inteligente	39
Figura 21 - Contrato inteligente da consulta.....	40
Figura 22 - Interface do Ganache.....	41
Figura 23 - Script de geração de cargas de armazenamento	42
Figura 24 – Informações da consulta retornadas com o acesso permissionado	45
Figura 25 – Erro de acesso não permissionado	4

Lista de Tabelas

Tabela 1 - Carga e tempo de duração no armazenamento na Blockchain	44
Tabela 2 - Relação de requisições armazenadas e não armazenadas na Blockchain	45
Tabela 3 - Resultado da avaliação de acesso permissionado.....	45

Lista de Abreviaturas e Siglas

RES	Registro Eletrônico de Saúde
EHR	Registro Eletrônico de Saúde
PEP	Prontuário Eletrônico do Paciente
SIS	Sistema de Informação de Saúde
SGBD	Sistema de Gerenciamento de Banco de Dados
DApps	Aplicativos Decentralizados
PHR	Registro Pessoal de Saúde
ADN	Aplicação Dedicada a nó
KSI	Assinatura sem chave
API	Interface de Programação de Aplicativo
SQL	Linguagem de Consulta Estrutura
SPA	Aplicação de Página Única
REST	Transferência de Estado Representacional
RESTful	Transferência de Estado Representacional com Todas as Restrições Definidas
HTTP	Protocolo de transferência de HyperTextos
POST	Método de Requisição do Protocolo HTTP para Envio de Dados
GET	Método de Requisição do Protocolo HTTP para Consulta de dados
PUT	Método de Requisição do Protocolo HTTP para Atualização de dados
PATCH	Método de Requisição do Protocolo HTTP para Atualização Parcial de dados
HTML	Linguagem de Marcação de Texto
CSS	Folha de Estilo em Cascata
JSX	Formato do Arquivo Utilizado em Aplicações com React JS

Sumário

1. Introdução	10
1.1 Contextualização	10
1.2 Objetivo	11
1.2.1 Objetivos Gerais	12
1.2.2 Objetivos Específicos.....	12
1.3 Metodologia	12
1.4 Organização do trabalho	13
2. Conceitos Básicos	14
2.1 Registro Eletrônico de Saúde	14
2.2 Tecnologia Blockchain	16
2.3 Trabalhos Relacionados.....	21
3. Aplicação de Saúde com Armazenamento em Tecnologia Blockchain	26
3.1 Arquitetura e Visão Geral.....	26
3.2 Prontuário da consulta	29
3.3 Serviço de Persistência de Dados.....	30
3.3.1 Esquema Básico das Rotas	31
3.3.2 Descrição das Rotas.....	33
3.3.3 Persistência de dados na rede Blockchain	34
3.4 Acesso de Dados Permissionado.....	36
3.5 Tecnologias Utilizadas	37
4. Avaliação da Solução Proposta	42
4.1 Cenário da Avaliação	42
4.2 Resultados Encontrados.....	44
5. Conclusão	47
5.1 Considerações Finais	47
5.2 Trabalhos futuros	47
Referência	49

1. Introdução

Este capítulo apresenta uma visão geral sobre o problema de pesquisa que motivou o desenvolvimento de uma solução computacional para o armazenamento de dados de registro eletrônico de saúde, fazendo uso da tecnologia Blockchain e da persistência poliglota. A organização deste capítulo está disposta a seguir: A seção 1.1 descreve a importância do uso de novas tecnologias no desenvolvimento de sistemas de informação de saúde. A seção 1.2 descreve os objetivos gerais e específicos deste trabalho. A seção 1.3 apresenta a metodologia utilizada para o desenvolvimento e elaboração desta pesquisa. Por último, a seção 1.4 descreve a estrutura dos demais capítulos.

1.1 Contextualização

A tecnologia da informação em nossa sociedade tornou-se uma ferramenta chave para o seu funcionamento como a conhecemos hoje. A informação em meio a este contexto, passou a receber maior notoriedade em diversos setores, devido a larga quantidade de dados processados nos *softwares* presentes no mundo. No setor de saúde não foi diferente, em virtude da popularização do armazenamento do registro eletrônico de saúde (RES), muito se tem discutido sobre o uso de normas, padrões e a aplicação de novas tecnologias da informação e comunicação no desenvolvimento de sistemas de informação de saúde (SIS) (Bowman, 2013). Os SIS processam dados de diferentes tipos de aplicações (e.g., prontuário eletrônico do paciente, sistema de análises clínicas e patologia) e auxiliam os profissionais de saúde nas atividades de cuidados clínicos do paciente. Por estas razões, um SIS deve prover mecanismos de segurança no armazenamento dos dados do RES e recursos de autenticidade e confiabilidade na manipulação dos dados (Hölbl et al, 2018; Quaini et al, 2018; Liang et al, 2020).

Durante o processo de desenvolvimento e manutenção de um SIS, o projeto de banco de dados representa uma importante fase no ciclo de vida de um sistema de software, pois permite identificar os requisitos de dados da aplicação, projetar a arquitetura e os componentes de software e escolher a melhor forma de armazenamento e recuperação dos dados (Araújo et al, 2019). No setor de saúde, comumente os desafios identificados no estado da arte na área de banco de dados

referem-se ao uso de diferentes paradigmas de Sistemas de gerenciamento de banco de dados (SGBD) para armazenar os requisitos de dados (i.e., estruturados e não estruturados) que compõem o RES (Araújo et al, 2020), a utilização de tecnologias big data para gerenciar grandes volumes de dados (Kaur et al, 2017) e a adoção de padrões de saúde para garantir a interoperabilidade do RES entre diferentes SIS (Zheng et al, 2017; Huang et al, 2003).

O uso da tecnologia Blockchain tem sido utilizada nas mais diversas áreas de conhecimento, tais como: mercado financeiro (Nakamoto, 2009), segurança pública (Xu et al, 2019) e no setor de saúde (Bowman, 2013). Na área da saúde, ferramentas computacionais foram desenvolvidas para apoiar o armazenamento de dados clínicos utilizando Blockchain (Surya et al, 2020) e *Smart Contracts*. *Smart Contracts* podem ser definidos como um código executado na Blockchain que facilita, executa e impõe os termos de um contrato entre partes não confiáveis (Maher et al, 2017).

Tendo em vista os fatos apresentados, foi identificado no estado da arte uma carência de soluções computacionais no setor da saúde que façam uso da persistência poliglota em conjunto com a tecnologia Blockchain. Desse modo, este trabalho propõe o desenvolvimento de um sistema de informação de saúde, capaz de realizar o gerenciamento dos RES, armazenando suas informações por meio da persistência poliglota, utilizando para isso, uma base de dados relacional e a tecnologia Blockchain. Além disso, a solução proposta também é capaz de realizar o gerenciamento de permissão de acesso às informações armazenadas na rede Blockchain. Por fim, para avaliar a aplicação desenvolvida, foi realizada a validação do serviço de persistência de dados, gerenciamento de acesso e eficiência de uso, mediante a aplicação de testes locais.

1.2 Objetivo

Esta seção de objetivos é dividida em duas subseções. A subseção 1.2.1 apresenta os objetivos gerais deste trabalho, enquanto a Subseção 1.2.2 lista os objetivos específicos utilizados para desenvolver a solução computacional descrita neste documento.

1.2.1 Objetivos Gerais

O trabalho proposto tem por finalidade o desenvolvimento de uma solução computacional que possa auxiliar as organizações do setor de saúde, mais especificamente os médicos, em suas atividades de atendimento a pacientes. Essa solução computacional deve fazer uso das tecnologias de armazenamento Blockchain para garantir a integridade, confiabilidade e interoperabilidade dos dados clínicos inseridos pelos profissionais da saúde.

Desse modo, serão realizados estudos na literatura para realizar a identificação e mapeamento das principais soluções computacionais utilizadas em SIS, e avaliar as soluções que mais se identificarem com a solução proposta neste trabalho.

1.2.2 Objetivos Específicos

Os objetivos específicos listados abaixo representam as principais ações de trabalho para o desenvolvimento do trabalho de conclusão de curso descrito neste documento, que são:

- Entender o funcionamento e características da tecnologia blockchain;
- Projetar a arquitetura de software da solução proposta;
- Desenvolver o serviço de geração de transações blockchain;
- Construir os smart contracts;
- Modelar os requisitos e desenvolver a aplicação de saúde;
- Avaliar o serviço de geração de transações blockchain;
- Avaliar o serviço de permissionamento de contratos na blockchain.

1.3 Metodologia

A Metodologia utilizada para a elaboração deste trabalho de conclusão de curso, consiste na revisão da literatura e na pesquisa exploratória. Por meio da revisão da literatura, foram identificados modelos computacionais fornecidos pela *openEHR*

(i.e. *Demographic Information Model*, *Data Structures Information Model* e *OpenEHR Architecture*), no qual disponibilizou os componentes necessários para a especificação e desenvolvimento da solução computacional aqui proposta; do mesmo modo, foram identificados no estado da arte pesquisas que abordam o uso da tecnologia Blockchain no armazenamento de dados clínicos e persistência poliglota em sistemas de saúde. A pesquisa exploratória possibilitou o desenvolvimento da aplicação de gerenciamento de RES que faz uso do armazenamento poliglota junto à tecnologia Blockchain para a persistência dos dados.

1.4 Organização do trabalho

Os demais capítulos deste trabalho estão distribuídos da seguinte forma: O Capítulo 2 aborda sobre os conceitos básicos das principais tecnologias utilizadas no desenvolvimento da aplicação e os trabalhos correlatos identificados na literatura; O Capítulo 3 apresenta a aplicação de gerenciamento de RES desenvolvida neste trabalho; O Capítulo 4 discorre sobre as avaliações realizadas nas aplicações e expõe os resultados obtidos; O Capítulo 5 finaliza esse trabalho com as considerações finais e propostas de trabalhos futuros.

2. Conceitos Básicos

Este capítulo descreve os conceitos básicos utilizados no desenvolvimento deste trabalho e está organizado da seguinte forma. A Seção 2.1 descreve o que são Registros Eletrônicos de Saúde e a sua importância de uso. A Seção 2.1 descreve uma visão geral do funcionamento da tecnologia Blockchain e dos *Smart Contracts*. Por fim, a Seção 2.3 aborda sobre os principais trabalhos que estão sendo desenvolvidos no estado da arte com relação ao uso da tecnologia Blockchain nos mais diversos setores, incluindo o setor da saúde.

2.1 Registro Eletrônico de Saúde

Com o aumento da digitalização das informações de saúde, registros tradicionais realizados de forma escrita passaram a não atender as demandas clínicas presentes em instituições de saúde. Os Registros Eletrônicos de Saúde (RES) são registros capazes de armazenar informações clínicas da saúde e doenças de forma digital, facilitando o acesso às informações, compartilhamento interno dos dados e interoperabilidade de dados entre instituições médicas (Braga, 2016). A utilização dos RES ao longo da vida do paciente permite o histórico completo de seus dados clínicos, oferecendo uma redução de custos em novos exames, informações mais precisas para os novos profissionais que o atendem e redução de tempo para ambos. Os profissionais da saúde têm informações mais precisas para serem utilizadas em consultas e tratamentos, podendo agregar novas informações no registro do paciente (Braga, 2016).

No setor de saúde, é comum a existência de diferentes tipos de aplicações para o armazenamento de dados clínicos, como diagnóstico por imagem, análise clínicas e Prontuário Eletrônico do Paciente (PEP) (Bezerra, 2019). A heterogeneidade dos dados entre os RES dificulta o compartilhamento mútuo das informações clínicas entre as instituições de saúde. Em virtude dessa problemática, estudos estão sendo realizados para o desenvolvimento e criação de padrões de interoperabilidade dos dados entre instituições de saúde. Atualmente os padrões ISO/EM 13606 (ISO 2008), HL7 (Noumeir, 2010) e OpenEHR (Beale et al, 2007) são referências encontradas no mercado que buscam solucionar problemas relacionados a padrões no

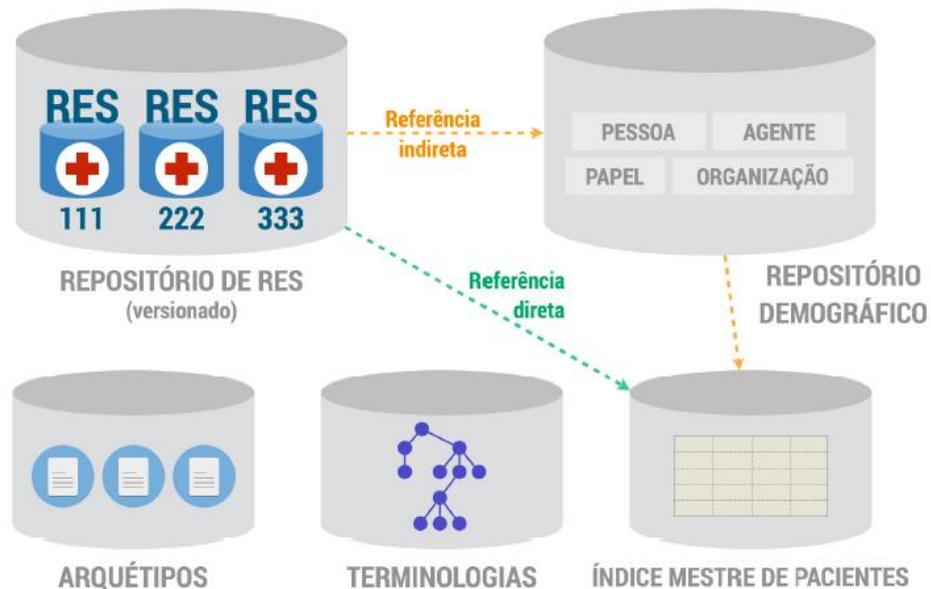
desenvolvimento de RES e interoperabilidade semântica dos dados clínicos entre instituições de saúde.

A ISO 13606 tem como objetivo o compartilhamento de extratos de RES. Esse modelo possui duas camadas de modelagem devido a utilização da *Application dedicated node* (ADN) do *openEHR* (ISO, 2008), entretanto, esse padrão possui problemas com a baixa fluidez em suas atualizações, fazendo com que fique atrás do padrão *openEHR*. O conjunto de normas internacionais HL7 (*Health Level Seven*) tem como objetivo a representação e transferência de dados clínicos e administrativos utilizados em RES (interoperabilidade semântica). O HL7 possui 7 camadas de aplicação, baseadas na estrutura de comunicação entre computadores do modelo OSI (Noumeir, 2010).

O conjunto de padrões *openEHR* faz uso de diversos tipos de arquétipos e *templates* para analisar as diferentes etapas durante o atendimento. O desenvolvimento do RES baseado em arquétipo e *template* cria mecanismos de interoperabilidade entre as instituições médicas, possibilitando a expansão do uso desses sistemas. Um arquétipo pode ser definido como uma expressão computacional representada por restrições de domínio, que modelam os atributos de dados e dão significado semântico ao RES (Araújo, 2018), enquanto *templates* são interfaces gráficas do usuário que possui o agrupamento e refinamento dos arquétipos (Hoy, 2007). Arquétipos e *templates* na saúde são adotados para remodelar os conceitos clínicos de sistemas legados, implementar o RES em sistemas de banco de dados e definir os requisitos de dados e as terminologias de sistemas de informação em saúde (Araújo, 2018).

O padrão *OpenEHR* consiste em conjunto de especificações e ferramentas livres (Bacelar, 2015). Esses conjuntos e ferramentas realizam a separação das características genéricas que modelam a estrutura do Registro Eletrônico em Saúde (RES), também conhecido como modelo de referência, das restrições e padrões associados aos dados clínicos, denominado modelo de conhecimento. O seu principal objetivo é permitir a criação de RES que possibilitem a interoperabilidade semântica e comutabilidade em e-saúde (Bacelar, 2015). A Figura 1 mostra a arquitetura de um RES mínimo que utiliza o conjunto de padrões *OpenEHR*

Figura 1 - Arquitetura de RES mínimo com padrões OpenEHR.



Fonte: Bacelar, 2015.

A arquitetura de um RES mínimo que é baseado no *OpenEHR* é formada por um conjunto de repositórios, no qual serão armazenadas as informações pertinentes a cada estrutura (Bacelar, 2015). Os repositórios presentes nessa arquitetura são: repositório de RES, repositório de Arquétipos, repositório demográfico e terminologias se possível. Essa arquitetura permite a materialização da proposta de utilização do padrão *OpenEHR* que segundo (Braga, 2016) é a realização da persistência de dados clínicos nos RES, junto com a interoperabilidade semântica entre instituições de saúde.

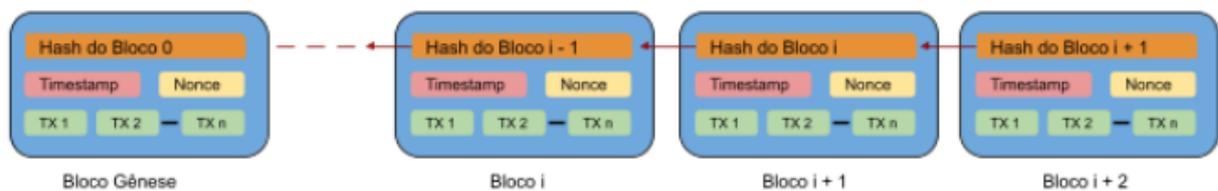
2.2 Tecnologia Blockchain

O *white paper* “*Bitcoin: A Peer-to-Peer Electronic Cash System*” de Satoshi Nakamoto, publicado em 2008, apresentou a utilização da tecnologia Blockchain na realização das transações da moeda digital Bitcoin. Essa tecnologia possui a capacidade de remover uma vulnerabilidade crucial em projetos de rede, que é o comprometimento das políticas de gerenciamento de confiança da rede (Alcazar, 2017). Dessa forma, é possível a realização de transações entre partes não confiáveis

de forma segura. O uso do Blockchain oferece descentralização, imutabilidade e consenso via criptografia e teoria dos jogos.

O Blockchain é um banco de dados distribuído que possui uma arquitetura de computação descentralizada, no qual armazena uma lista crescente de transações ordenadas, que são agrupadas em blocos continuamente reconciliados para manter as informações atualizadas. Essa lista encadeada é composta no início por um bloco gênese, e os seguintes blocos possuem uma referência ao anterior como é exposto na Figura 2. Somente um bloco pode ser adicionado por vez a essa corrente, sendo cada bloco matematicamente verificado através da utilização de criptografia para comprovar que os seguintes blocos seguem a sequência dos anteriores. Esse é o processo comumente chamado de “mineração” ou também “Prova de Trabalho” (Proof of work)(NAKAMOTO,2008).

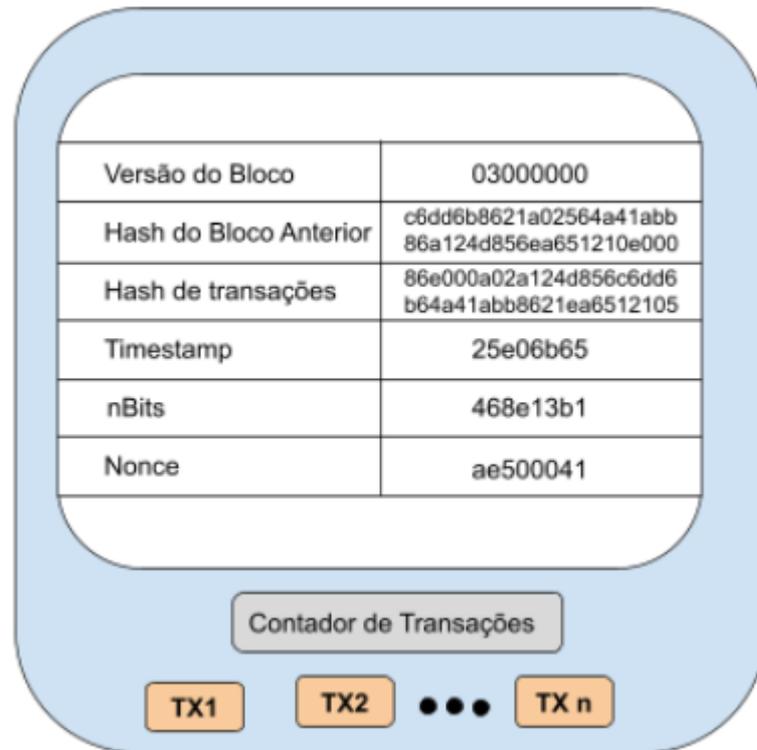
Figura 2 - Lista encadeada de Blocos na rede Blockchain.



Fonte: Zheng, 2018.

Um bloco Blockchain possui em sua composição o cabeçalho do bloco e o corpo do bloco. O cabeçalho do bloco possui a versão do bloco, responsável por indicar qual o conjunto de regras de validação é adotado por o bloco; depois temos a *hash* do bloco anterior; uma *hash* para as transações do bloco; um *Timestamp* que marca o horário em segundos; um *nBits*, responsável por armazenar a *hash* do bloco com tamanho compacto; e por último o *Nonce* formato por um campo de 4 bytes utilizado no algoritmo de consenso. O corpo do bloco possui um contador de transações. O tamanho máximo de transações de um bloco depende do conteúdo armazenado, do tamanho da transação e do tamanho do bloco (Zheng et al, 2018). A Figura 3 demonstra a composição de um bloco.

Figura 3 - Composição de um Bloco Blockchain



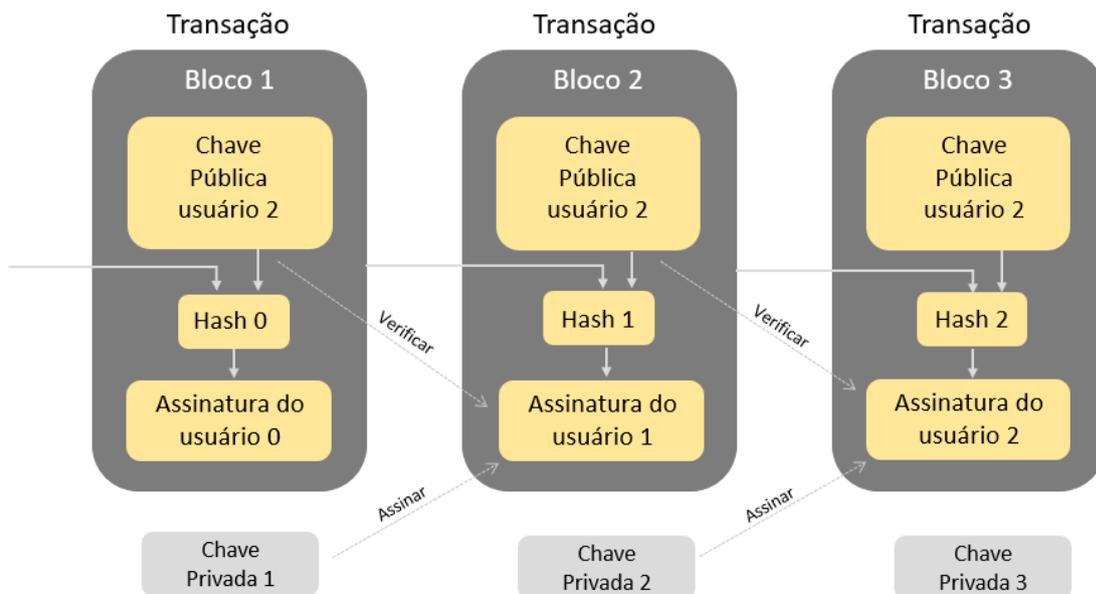
Fonte: Zheng, 2018.

As transações realizadas com a tecnologia Blockchain, são baseadas em uma cadeia de assinaturas digitais. A transação será realizada por assinaturas digitais de um *hash* da transação anterior, um *hash* é um código que transforma as informações presente em um bloco em uma sequência numérica hexadecimal de tamanho fixo. As transações Blockchain são separadas em blocos, cada bloco possui uma chave pública do usuário e um *hash* baseado no bloco anterior, cada usuário possui uma chave privada que será responsável por confirmar sua autenticidade.

As transações somente são realizadas após a utilização de três assinaturas digitais, a primeira contém o *hash* da transação anterior, a segunda contém a chave pública do usuário que vai receber a transação, e a última assinatura digital contém a chave privada que libera o conteúdo presente na transação para o próximo usuário. Devido a esse mecanismo, ao tentar fraudar um dos blocos que já está presente na rede Blockchain, será gerado um *hash* incompatível com os anteriores,

impossibilitando a modificação do seu conteúdo. A Figura 4 mostra a transação de registros de blocos Blockchain

Figura 4 - transações de registros de blocos Blockchain



Fonte: Nakamoto, 2008.

Realizar transações somente com as assinaturas digitais não garante que os dados chegarão ao usuário mesmo que a transação seja realizada. Logo, (Nakamoto, 2018) afirma que uma forma de confirmar a ausência de uma transação é estar atento a todas as transações que ocorrem. Por esse motivo, foi criado o servidor de carimbo de tempo em uma base ponto-a-ponto que se utiliza de sistemas de prova de trabalho. O servidor de carimbos de tempo é responsável por gravar a data e hora de uma função *hash* que será carimbada. A Prova de trabalho é adicionada ao bloco *Nonce*. *Nonce* é um número aleatório que será usado junto com o *hash* para garantir que não haja a repetição de *hash's* anteriores. Esse processo finaliza quando a *hash* do bloco chega aos zero bits exigidos. Isso causa a necessidade da finalização de todo o trabalho realizado no bloco, sendo necessário realizar todo o trabalho novamente caso seja preciso mudar algo no bloco. Dessa forma, para fraudar um conteúdo de um bloco é preciso fraudar todos os blocos presentes no conjunto da rede Blockchain, fazendo

com que essa prática não seja viável ou possível com os recursos computacionais atuais.

O Blockchain fornece as bases para diversos domínios de aplicações, incluindo criptomoedas, IOT, Reputação de Sistemas e Redes Sociais (Zheng, 2018) e aplicativos descentralizados (*DApps*) (JOHNSTON et al, 2014). Os *DApps* são aplicações distribuídas que utilizam em suas transações a tecnologia Blockchain e os *Smart Contracts* (Contratos Inteligentes). Os *Smart Contracts* podem ser definidos como um código executado no Blockchain que facilita, executa e impõem os termos de um contrato entre partes não confiáveis (Maher, 2017). Essa tecnologia não necessita de uma terceira parte confiável para validar uma transação, como nos contratos tradicionais. Diante disso, o custo por transação realizada, torna-se menor comparado a esses. São atribuídos aos *Smart Contracts* um endereço exclusivo de 20 bytes para não ser possível a alteração do seu conteúdo após a inserção na Blockchain. Dessa forma, quando o usuário deseja realizar uma transação com o contrato, é preciso apenas enviar o conteúdo da transação para o endereço do contrato, no qual será realizada a prova de consenso para em seguida ser atualizado em conformidade. Segundo (Maher, 2017), os contratos são capazes de receber, ler / escrever informações para seu armazenamento privado, armazenar dinheiro em seu saldo de conta, enviar / receber mensagens ou dinheiro de usuários / outros contratos ou até mesmo criar novos contratos. As Figura 5 e 6 mostram a diferença entre as transações com contratos usuais e os contratos inteligentes.

Figura 5 - Transações com Contratos Tradicionais.



Figura 6 - Transações com Contratos Inteligentes



Fonte: Osetskyi, 2018.

Para o desenvolvimento dos *Smart Contracts*, são identificadas diversas plataformas Blockchain usadas para essa finalidade, no entanto a *Ethereum* é a plataforma mais utilizada para o desenvolvimento dos mesmos (Maher, 2017). A *Ethereum* é uma plataforma global de código aberto para aplicações descentralizadas que permite o desenvolvimento e execução dos contratos inteligentes, como por exemplo aplicações de criptomoedas, redes sociais que requerem algum tipo de transações e ativos digitais, entre outros (Ante, 2021). Seu uso em contratos inteligentes é conhecido devido a flexibilidade do sistema de *Turing-completeness*, que permite a personalização dos contratos inteligentes em seu desenvolvimento (Buterin, 2015).

2.3 Trabalhos Relacionados

Em razão da sua capacidade de armazenamento descentralizado e seguro, o uso da tecnologia Blockchain vem sendo aplicado em diversas áreas de conhecimento e setores estratégicos que requerem a manipulação de diversas categorias de dados. Essa tecnologia ficou conhecida principalmente pelas criptomoedas, sendo o Bitcoin a mais famosa, porém o Blockchain é utilizado em outras áreas. Mohan, C (2017) apresenta a utilização da Blockchain em diferentes cenários, tais como: no gerenciamento de cadeias de suprimentos, na utilização de *smart contracts* para validação de acordo entre partes não confiáveis, no processamento de derivativos e no gerenciamento de proveniência.

É possível observar o uso da tecnologia Blockchain no contexto militar em (Alcazar, 2017.), como uma extensão na inteligência militar, garantindo sigilo e anonimato, em operações de ordem e logística, assim como dados confidenciais. Nagothu (2018), propõe um sistema de vigilância inteligente, que se baseia na arquitetura de microsseviços e na tecnologia Blockchain, devido a sua segurança e descentralização para o armazenamento dos dados.

Sendo a privacidade, segurança e a descentralização as principais características dessa tecnologia, diversas outras áreas utilizam-na, autores como (Zheng, 2017.) explanam os benefícios e as oportunidades que essa nova tecnologia oferece, fazendo uma análise teórica fornecendo a taxonomia do Blockchain.

Com o desenvolvimento dessa tecnologia, novos incrementos surgiram, um deles são os chamados contratos inteligentes (*Smart contracts*), tratam-se de protocolos utilizados para proporcionar confiabilidade em transações online, como em (Alharby et al, 2017), que se objetiva a identificar tópicos de pesquisa no que se refere aos contratos inteligentes baseados em Blockchain e os desafios enfrentados para estudos futuros. A associação da tecnologia do blockchain com outros paradigmas como o Big Data também é objetivo de pesquisas. Em (Rubio, 2018), é feita a análise de uma cadeia de suprimentos que gera um grande volume de dados para garantir o rastreamento dos ativos em meio a um grande volume de dados.

Na área da saúde, o uso da tecnologia Blockchain vem sendo direcionado nos Registros Eletrônicos de Saúde (RES) com o uso dos contratos inteligentes para permitir acesso, gerenciamento e interoperabilidade dos dados dos pacientes (Da Conceição, 2018). A interoperabilidade dos dados dos RES entre Sistema Integrado de Saúde e a interação efetiva entre usuários e aplicações médicas é abordado junto com o uso da tecnologia Blockchain com foco no tráfego dos dados dos pacientes entre organizações de forma segura (Zhang Peng, 2017). Legaz-García (2015) apresenta uma solução complementar para a interoperabilidade semântica por meio de uma ferramenta, a qual busca o desenvolvimento de um processo organizacional e de governança para alcançar a interoperabilidade semântica de modelos clínicos.

Um dos fatores principais para a continuidade da interoperabilidade dos RES é a arquitetura *openEHR*, que permite a associação de arquétipos e *templates* as terminologias internas e externas, ponto fundamental para chegar ao alto nível de interoperabilidade semântica dos dados (Miranda, N., 2014). Um outro sistema eletrônico de saúde encontrado no estado da arte são os PHR (Registro de Saúde

Pessoal) que junto com a tecnologia Blockchain traz maior segurança e confiabilidade dos dados clínicos dos pacientes aos médicos. Seguindo essa linha de pesquisa, (Roehrs, 2019) realiza o estudo da implementação e avaliação de um protótipo utilizando arquitetura *OmniPHR* com o objetivo de reunir registros de integridade distribuídos usando o padrão de interoperabilidade *openEHR* e a Blockchain para melhorar o gerenciamento e o compartilhamento dos PHR.

Outro benefício gerado na área da saúde proporcionada pela Blockchain é seu uso para assegurar os dados em Sistemas de Informação em Saúde que se utilizam de padrões internacionais. O capítulo um (01) do livro *Advances in Computers* produzido por (Zhang, 2018) identifica questões urgentes na área de saúde, com foco na interoperabilidade e no cuidado centrado no paciente. A interoperabilidade é definida por (Zhang, 2018) como a capacidade de sistemas heterogêneos realizarem a comunicação e transferência de dados entre si. A utilização desse recurso em ambientes de saúde permite que os provedores compartilhem registros médicos de pacientes de forma segura e escalável.

O trabalho descrito em (Lanxinang, 2019) propôs a criação de um mecanismo de busca de dados que realizasse consultas através de índice em um sistema com Blockchain. O processo de construção do índice compreende a extração de identificadores de documentos e o empacotamento desses identificadores para inclusão no contrato inteligente para garantir integridade e segurança dos dados. O trabalho proposto em (Haiping, 2021) traz como proposta a criação de um sistema de *eHealth* baseado em Blockchain, chamado BCES, esse sistema tem como objetivo garantir que a manipulação de *EHRs* possa ser auditada com o uso do Blockchain. Em (Gayathri, 2020), os autores apresentaram uma proposta de criação de um framework que utiliza sua infraestrutura em nuvem de assinatura sem chave (KSI) junto ao armazenamento descentralizado com o Blockchain para o armazenamento dos dados dos EHR. A utilização do Blockchain junto a computação em névoa possibilita a grande transferência de dados para a nuvem graças a sua baixa latência.

Na mesma linha de pesquisa, a solução desenvolvida em (Naveed, 2019) traz em seu trabalho a proposta da construção de uma estrutura de monitoramento e reconhecimento de atividades que é baseada em procedimento de categorização cooperativa multiclasse. A utilização desse procedimento permite melhorar a precisão da classificação de atividades em vídeos que suportam a arquitetura de Blockchain baseada em computação em nuvem ou névoa. Esse sistema foi desenvolvido para o

monitoramento de pacientes que sofrem de doenças mentais ou deficientes que necessitam de assistência. Assim como o uso de computação em nuvem e névoa junto a utilização do Blockchain podem trazer uma grande contribuição aos EHR. Da mesma forma, a aplicação de Algoritmos genéticos junto ao Transformada discreta de *wavelet* e Blockchain podem garantir maior segurança e confiabilidade aos dados. A ferramenta proposta em (Ahmed, 2018) tem como objetivo criar um sistema de gerenciamento e proteção de registros médicos com o Blockchain, este sistema é suportado por uma transformada *Wavelet* Discreta para aumentar a segurança geral na análise de sinais digitais, e uma técnica de Algoritmo Genético para otimizar a técnica de otimização de filas. A introdução deste gerador de chaves criptográficas aumenta a imunidade e o controle de acesso ao sistema, o que permite verificar os usuários com segurança de forma rápida. Em (Gordon et al, 2018), relata-se que a interoperabilidade na área da saúde geralmente se concentra no intercâmbio de dados entre entidades comerciais, no entanto, soluções voltadas para a interoperabilidade e acesso de dados para os pacientes necessitam ser desenvolvidas. Em (Quaini et al., 2018), demonstrou-se ser possível obter a integração e o gerenciamento dos dados do RES por meio da tecnologia Blockchain de maneira distribuída entre organizações de saúde e pacientes.

A solução computacional apresentada em (Dubovitskaya et al., 2017) discute a aplicação de um framework para auxiliar o compartilhamento de dados de pacientes em tratamento de câncer entre organizações de saúde. A tecnologia Blockchain foi utilizada e como principais resultados destaca-se a construção de uma arquitetura computacional em nuvem para conectar hospitais e disponibilizar *API's* para consulta de informações. Uma abordagem para compartilhar informações de saúde de maneira eficaz e segura dentro de uma rede de compartilhamento de dados é proposta por (Peterson et al., 2016). Os autores especificam o desenvolvimento um novo algoritmo de consenso projetado para facilitar a interoperabilidade dos dados em organizações de saúde. Em (Azaria et al., 2016), uma interface de usuário intuitiva que facilita o compartilhamento e a visualização de dados de pacientes utilizando tecnologia Blockchain é especificada, e um sistema de saúde que gerencia o RES de forma descentralizada com tecnologia Blockchain e fornece recursos de interoperabilidade e acessibilidade.

Os trabalhos aqui apresentados trazem impactos positivos e relevantes para o estado da arte e mostram como o uso da tecnologia Blockchain pode contribuir para o avanço tecnológico no setor da saúde. Desse modo, foi observado uma lacuna de trabalhos que abordem a utilização da persistência poliglota com Blockchain no setor de saúde. Dada a importância da integridade das informações clínicas, esse trabalho de conclusão de curso tem o objetivo desenvolver uma solução computacional capaz de auxiliar organizações de saúde, fornecendo maior segurança no armazenamento dessas informações por meio do uso da persistência poliglota e a tecnologia Blockchain.

3. Aplicação de Saúde com Armazenamento em Tecnologia Blockchain

Este capítulo tem a finalidade de apresentar e discutir a aplicação de saúde desenvolvida neste trabalho. A esquematização desse capítulo está organizada da seguinte forma. Na Seção 3.1 é apresentado e detalhada a arquitetura de software utilizada na aplicação e expõe os principais componentes especificados. A seção 3.2 apresenta o prontuário de consulta baseado no openEHR. A Seção 3.3 discute sobre o uso individual e em conjunto dos serviços de persistência de dados SQL e Blockchain. A seção 3.4 aborda sobre a utilização de dados permissionados com o uso dos *Smart Contracts* em sistemas de saúde. A Seção 3.4 apresenta as tecnologias utilizadas no desenvolvimento da aplicação.

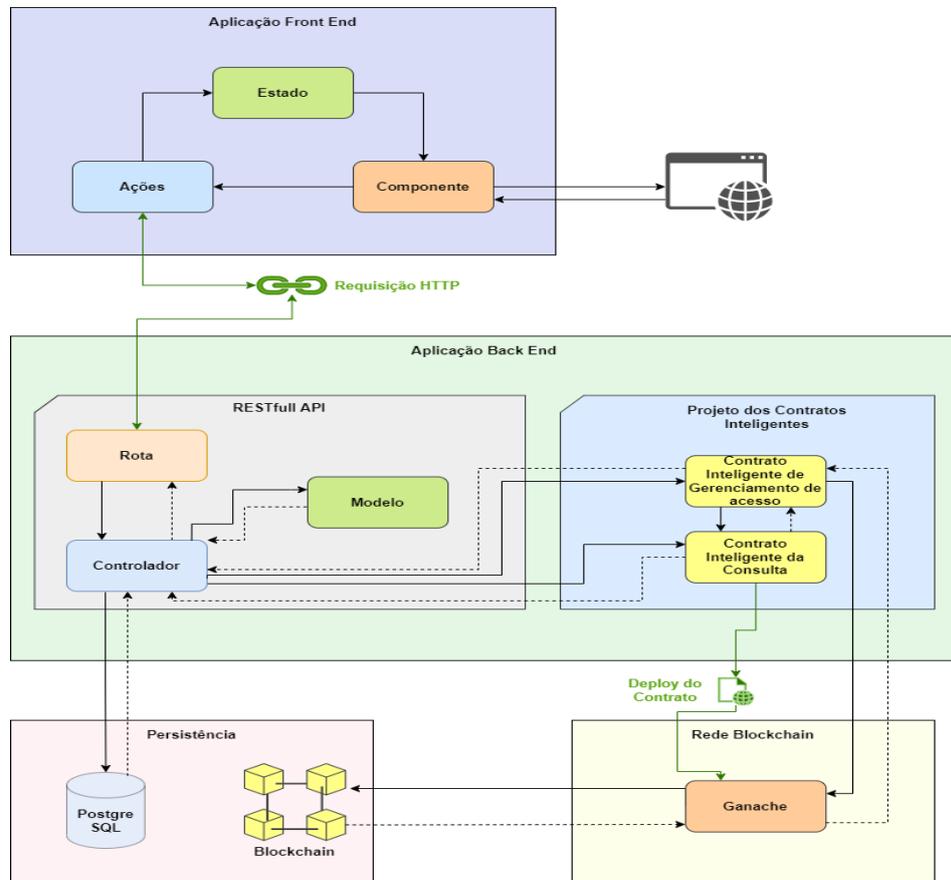
3.1 Arquitetura e Visão Geral

A aplicação proposta neste trabalho possui dois componentes principais, o *Front-end* e o *Back-end*. O *Front-end* é uma aplicação web de página única (*Single Page Applications*) responsável por realizar a interação com os médicos e pacientes que utilizam o sistema, possibilitando aos mesmo o fornecimento e acesso aos seus dados clínicos por meio de uma interface gráfica; ao passo que o *Back-End* realiza o gerenciamento e persistência poliglota em uma base de dados relacional e em uma Rede Blockchain. A Figura 7 apresenta a arquitetura desenvolvida para apresentar uma visão geral do funcionamento da solução proposta.

As funcionalidades da aplicação são divididas em dois cenários, o do médico e o do paciente. No primeiro caso existem três funcionalidades que são: criação de horários de atendimento, realização de atendimento já agendado pelo paciente e criação de consulta. A Figura 8 apresenta o diagrama de caso de uso das funcionalidades disponíveis no cenário de uso do médico na aplicação.

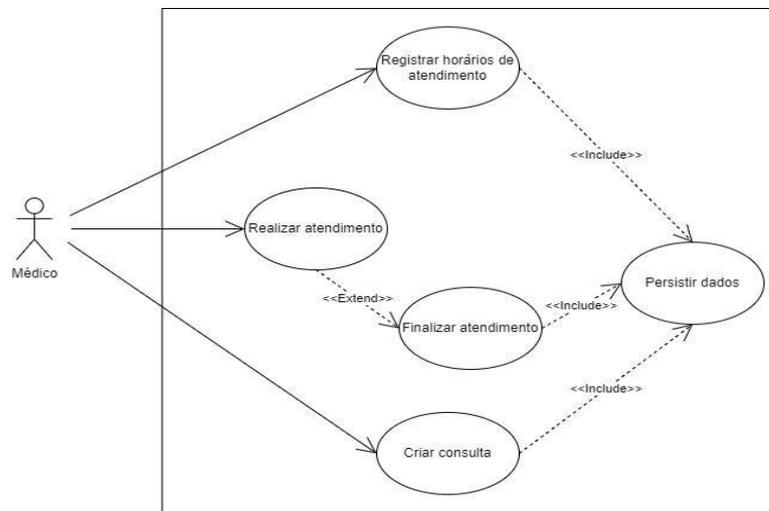
O cenário do paciente é composto por três funcionalidades que são: Agendar consulta, acessar consulta e permitir acesso da consulta. Para melhorar a visualização desse fluxo de atividades foi criada a Figura 9 que apresenta o diagrama de caso de uso do paciente.

Figura 7 - Arquitetura da solução proposta.



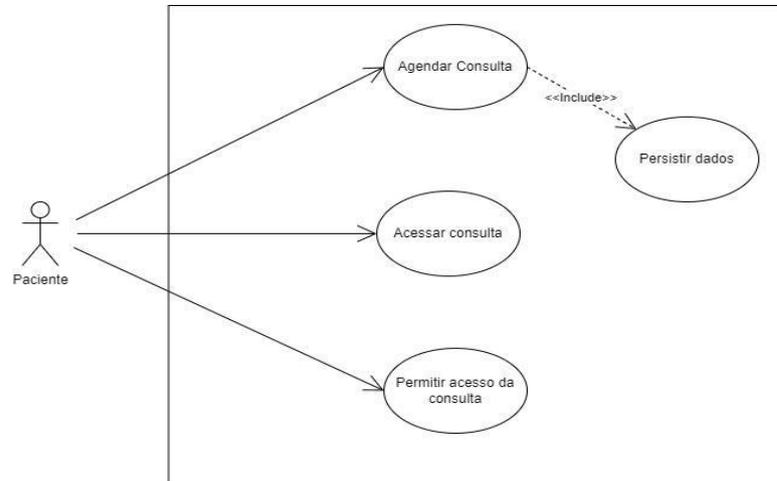
Fonte: Autor.

Figura 8 – Diagrama de Caso de Uso do médico



Fonte: Autor.

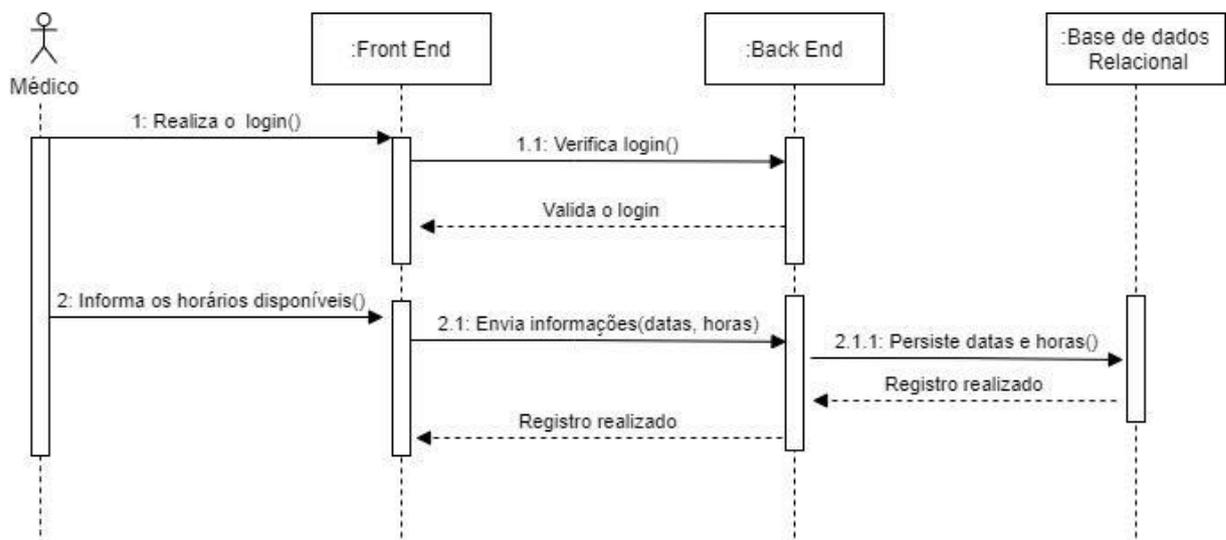
Figura 9 – Diagrama de Caso de Uso do Paciente



Fonte: Autor

Com a intenção de exemplificar a sequência de processos das funcionalidades anteriormente citadas. Foi criada a Figura 10, que apresenta o diagrama de sequência da funcionalidade de criação de horários de atendimento realizado pelo médico.

Figura 10 - Diagrama de sequência de uma das funcionalidades da aplicação



Fonte: Autor.

A arquitetura apresentada na Figura 7 é composta por componentes do *Front-end* e *Back-end*. Nela é possível observar que o *Front-end* é responsável por transmitir e receber as informações para o usuário através de ações e atualização dos valores dos estados e componentes. As ações são responsáveis por realizar a comunicação entre as interfaces mediante requisições HTTP, possibilitando o envio e recebimento de dados das consultas.

O *Backend* da aplicação é composto por dois componentes, o primeiro consiste em uma API *RESTfull* composta por rotas que direcionam os dados enviados para os controladores aplicarem as regras de negócios estabelecidas, verificando por meio dos modelos se os dados passados atendem ao padrão esperado. O segundo componente consiste nos contratos inteligentes que são criados e salvos na rede Blockchain quando a consulta é finalizada, além de permitir o gerenciamento de acesso às informações salvas na rede Blockchain.

A especificação da arquitetura de software desta aplicação permitiu organizar os componentes de modo a garantir um desenvolvimento que ofereça um sistema manutenível e extensível, utilizando-se do padrão arquitetural de comunicação por meio de interfaces. As interfaces são responsáveis por estabelecer a forma de comunicação entre os componentes e permite que modificações sejam realizadas em cada componente de forma isolada aos demais.

3.2 Prontuário da consulta

O padrão openEHR é composto por especificações e ferramentas livres, o qual permitem a construção de Registros Eletrônicos de Saúde (RES) que suportam a interoperabilidade semântica dos dados por meio de uso de arquétipo e templates capazes de atender o maior número de situações relacionadas aos mesmos. Desse modo, foi desenvolvida a funcionalidade de atendimento a consultas no *Front-end* da aplicação no qual realiza a criação de registros eletrônicos de saúde baseado no arquétipo *Problem/Diagnosis* e o template *Patient's background leading to the diagnosis of AMD*. Os campos utilizados para o desenvolvimento do prontuário de consultas da aplicação foram: Histórico Familiar, Doença Crônica, Alergia, Sintomas, Exames, Medicamentos e Diagnósticos. A Figura 11 apresenta a interface gráfica utilizada nas realizações das consultas.

Figura 11 – Interface Gráfica da Consulta

The screenshot displays the 'Blockchain EHR' interface. At the top, there is a navigation bar with 'Agenda', 'Criar Consulta', and 'Sair' options. The main content is divided into two sections: 'Informações do Paciente' and 'Dados Clínicos'.

Informações do Paciente:

- Nome:** Matheus dos Santos
- Ocupação:** Estudante
- Data de Nascimento:** 26/07/1995
- Sexo:** Masculino

Dados Clínicos:

- Histórico Familiar:** Familiares Maternos possuem histórico de doenças cardíacas
- Doença Crônica:** Não possui
- Alergia:** Dipirona
- Sintomas:** Febre
- Exames:** ...
- Medicamento:** Ibuprofeno
- Diagnóstico:** ...

At the bottom of the 'Dados Clínicos' section, there are two buttons: 'Atualizar' (blue) and 'Finalizar Consulta' (red).

Fonte: Autor.

3.3 Serviço de Persistência de Dados

O gerenciamento e armazenamento de dados da aplicação são realizados com a aplicação *Back-end* utilizando as tecnologias *JavaScript* com *NodeJS* para o gerenciamento de negócio, e uma base de dados relacional e a rede blockchain como sistema de armazenamento. A API faz uso das tecnologias de armazenamento da seguinte forma: o PostgreSQL é utilizado para persistência dos dados do médico e paciente, como autenticação, horários de atendimento, consultas realizadas e dados clínicos presente nas consultas em andamento. O Ganache é a Rede Blockchain no qual é realizado o armazenamento permanente das informações contidas nas consultas finalizadas.

3.3.1 Esquema Básico das Rotas

As informações são recebidas pela *API* através de requisições *HTTP* enviadas da aplicação *Front-end* utilizando as rotas desenvolvidas para o acesso às informações do atendimento. Todas as rotas recebem uma requisição *HTTP* e executam a função correspondente a rota retornando uma resposta a essa requisição, utilizando-se o padrão exibido na Figura 12.

Figura 12 - Padrão das funções chamadas nas rotas.

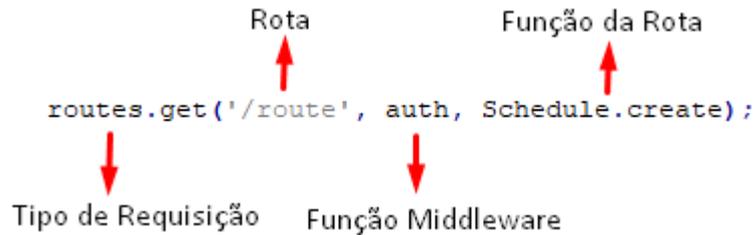
```
1 //Função assíncrona que recebe a requisição HTTP
2 async functionName(req, res, next){
3     try{
4         //ação a ser executada
5     }catch(error){
6         //ação executada em erros de execução
7         return res.status(500).json({error: 'error'});
8         next(error);
9     }
10 }
```

Fonte: Autor

Para realizar o tratamento de erros foi utilizado o bloco de tratamento de exceção *Try/catch*. O bloco *Try* contém toda a lógica da função a ser executada, retornando uma resposta em formato JSON com as informações solicitadas caso não seja identificado algum erro durante o processo; já o bloco *Catch* é o bloco de erros, esse bloco é chamado sempre que algum erro é retornado no bloco *try*, permitindo que seja retornado para a aplicação *Front-end* a mensagem de erro na solicitação da informação.

Para verificar a autenticidade do usuário solicitante da informação, algumas rotas foram protegidas utilizando-se uma função *middleware* responsável pela autenticação dos usuários e a chamada da requisição solicitada pelo mesmo. Sua chamada é realizada entre a requisição da rota e a chamada da função da rota, realizando a verificação se o usuário solicitante possui ou não um token válido para acesso a mesma, conforme exibido no modelo da Figura 13.

Figura 13 - Estrutura de uma rota com autenticação.



Fonte: Autor

A Figura 14 mostra o código da função de autenticação utilizada na API. As linhas 4 e 5 são responsáveis por verificar o recebimento do token enviado na requisição. A lógica presente entre as linhas 7 e 11 é responsável por receber o token enviado na requisição e verificar se o mesmo é válido para acessar as informações da rota. Por fim, na última linha é exportada a função para ser utilizada como middleware nas rotas protegidas.

Figura 14 - Lógica da função de autenticação.

```

1  const jwt = require('jsonwebtoken');
2
3  const auth = (req, res, next) => {
4      const token_header = req.headers.auther;
5      if (!token_header) return res.send({ error: 'Token não enviado' });
6
7      jwt.verify(token_header, 'blockchainEHR', (err, decode) => {
8          if (err) return res.send({ error: 'Token inválido' });
9          res.locals.auth_data = decode;
10         return next();
11     })
12 }
13
14 module.exports = auth;

```

Fonte: Autor

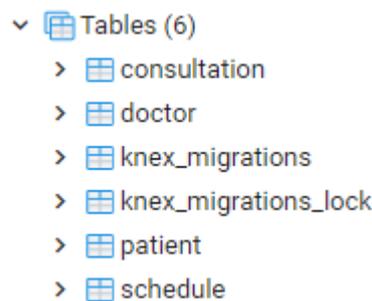
Esse padrão foi implementado em todas as rotas que realizam consultas às informações presentes na base de dados relacional da aplicação.

3.3.2 Descrição das Rotas

Rotas que estão diretamente ligadas a funcionalidades de manipulação de dados e regras de negócios, fazem uso do sistema de autenticação. Esse mecanismo exige que o usuário esteja devidamente autenticado para que possa ter acesso a rota solicitada. Para realizar esse procedimento é necessário enviar uma requisição do tipo **POST** para o endereço */doctor/login* caso seja um médico ou */patient/login* se for um paciente, incluindo os dados de login e senha no corpo da requisição.

As rotas são divididas de acordo com a divisão das tabelas da base de dados relacional, são elas: Doctor, Patient, Schedule e Consultation. Como mostrado na Figura 15. É importante ressaltar que as tabelas que iniciam com o nome knex são geradas automaticamente pelo query build do knex e não fazem parte do esquema do modelo relacional do banco de dados.

Figura 15 - Tabelas da base de dados relacional.



Fonte: Autor

A criação de novos usuários é realizada com o método **POST**. A rota */createDoctor* possui a funcionalidade de criação de contas do tipo médico e a rota */createPaciente* é responsável pela criação de contas do tipo paciente. As rotas que possuem funcionalidade de busca a informações utilizam o método **GET**. A rota */searchDoctor/:name* tem a função de realizar a buscar de profissionais com o nome passado como parâmetro. Na rota */searchDoctor/:id* tem funcionalidade de retornar o médico que possui o ID passado no endereço da rota. A busca por pacientes é realizada com a rota */searchPatient/:name* e retorna todos os registros com o nome passado no parâmetro.

Para ser realizado o gerenciamento dos agendamentos foram desenvolvidas rotas especializadas para essa funcionalidade. A rota `/createSchedule` tem a funcionalidade de criar novos agendamentos e faz uso do método **POST**. A busca por um paciente em específico pode ser realizada com a rota `/schedule/:id` com a utilização do método **GET**. A verificação de horários disponíveis é realizada com o método **GET** na rota `/hoursFree/:id?date`, no qual é passada a ID do médico como parâmetro e a data da consulta como query. Rotas que utilizam o método **PUT | PATCH** realizam alterações e registros já armazenados no banco de dados. Esse é o caso da rota `/pacienteSchedule/:id` que é responsável por realizar o agendamento do paciente na consulta selecionada.

As rotas de gerenciamento de consultas têm como funcionalidades a criação de consultas com o **POST** `/createConsultation`; a busca por uma consulta específica é possível com o **GET** `/consultation/:id`; e para atualizar a mesma é possível através do **PUT | PATCH** `/updateConsultation/:id`, passando a id da consulta como parâmetro.

As rotas do processo de finalização de consulta fazem uso do método **POST**. A rota `/createContract` finaliza a consulta com o armazenamento das informações na base de dados relacional e na Blockchain. A rota `/allowAccess` permite ao proprietário da consulta, dar acesso temporário para outro usuário. O usuário que recebeu acesso a consulta pode visualizar as suas informações através da rota `/external` que faz uso do método **PUT | PATCH**.

3.3.3 Persistência de dados na rede Blockchain

A metodologia utilizada para a transição dos dados entre a base PostgreSQL e a Rede Blockchain do Ganache é isolada do funcionamento restante do *Back-end* da aplicação.

As transições das informações no Blockchain são iniciadas com o comando de finalização de consultas no *Front-end*, no qual realizará a persistência no banco relacional e posteriormente iniciará o processo de armazenamento na rede Blockchain. O processo de armazenamento anteriormente citado é realizado da seguinte forma. Primeiro a rota de criação do contrato inteligente é chamada e recebe todas as informações da consultada no corpo da requisição. Em seguida, o contrato é criado e o deploy do mesmo é realizado na rede Blockchain. Por último, os dados da

Figura 17 - Função de criação do contrato inteligente das consultas.

```

1  const makeConsultation = async (consultationData) => {
2      const contract = new eth.Contract(consultationContract.abi);
3
4      try {
5          const deployedContract = await contract.deploy({
6              data: consultationContract.bytecode
7          }).send({
8              from: '0x66675CB43DC48b590F8EC3e99BC09446B81C9DB8',
9              gas: 6721975,
10             gasPrice: 2000000000
11         });
12
13         const result = await deployedContract.methods
14             .setConsultation(consultationData)
15             .send({
16                 from: '0x66675CB43DC48b590F8EC3e99BC09446B81C9DB8',
17                 gas: 6721975,
18                 gasPrice: 2000000000
19             });
20
21         return result
22     } catch (error) {
23         console.log(error)
24     }
25 }

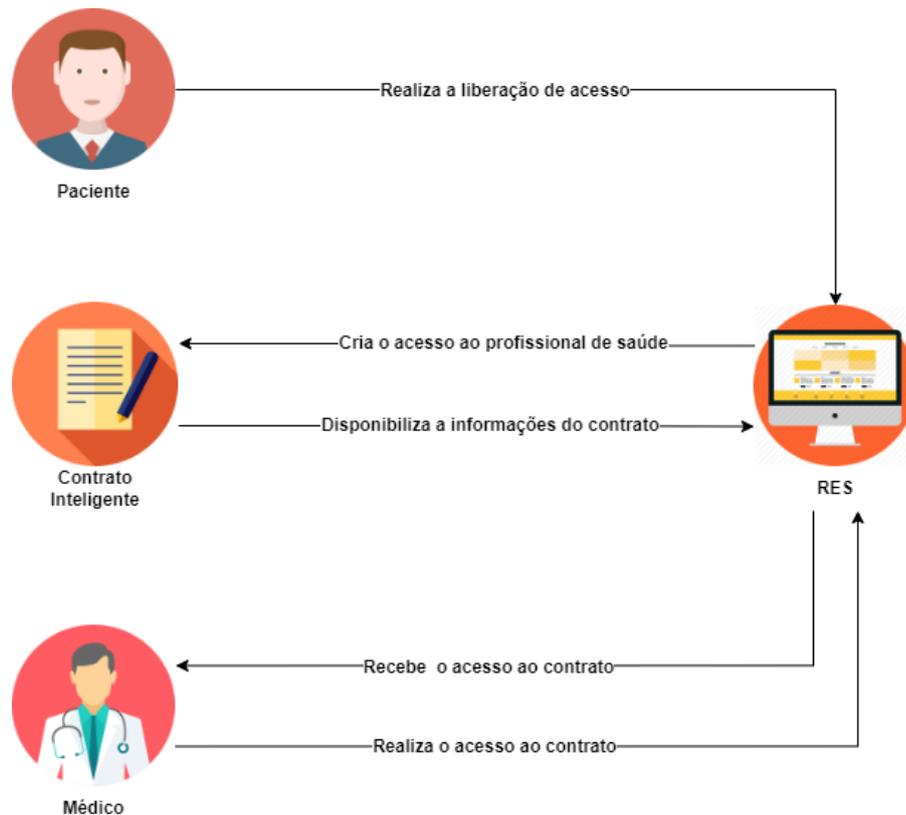
```

Fonte: Autor

3.4 Acesso de Dados Permissionado

O objetivo do permissionamento dos contratos da aplicação é permitir ao proprietário (paciente) do contrato compartilhar suas informações clínicas somente com profissionais da saúde que foram autorizados. Por padrão, os contratos inteligentes em uma rede Blockchain podem ser chamados por qualquer participante da rede ou outros contratos que possuam permissão para o mesmo. Com o objetivo de gerar mais confidencialidade e segurança nas informações armazenadas, o contrato desenvolvido nessa aplicação possui um sistema de restrição de acesso, no qual o acesso ao contrato somente é autorizado para o proprietário ou usuários com permissão de acesso. A Figura 18 mostra o fluxo dos dados para o permissionamento do contrato.

Figura 18 - Fluxo de dados para o permissionamento do contrato.



Fonte: Autor

O permissionamento da consulta é realizado através da autorização do paciente proprietário da mesma. O Paciente acessa a funcionalidade de permissão de acesso da aplicação, passando o endereço da conta do médico e o tempo de acesso à consulta. Em seguida é disponibilizado o acesso as informações da consulta e o médico as visualizam mediante a interface da aplicação.

3.5 Tecnologias Utilizadas

Para o desenvolvimento da aplicação aqui proposta foram utilizadas ferramentas e tecnologias que permitiram o desenvolvimento e execução do software. O desenvolvimento da interface gráfica foi realizado com o uso da Linguagem de marcação de texto HTML que possibilita a criação da estrutura das páginas web por meio de *tags*. A estilização da mesma foi realizada com o CSS uma linguagem de aplicação de estilos de páginas web. A tecnologia responsável pela interação do

usuário com o sistema é o *JavaScript* junto com a biblioteca *React JS*, no qual permite a criação de sites através da utilização de componentes. Segundo (Moraes, P. J, 2019) os componentes são funções *JavaScript* que recebem uma entrada e retornam novos elementos JSX, possibilitando o desenvolvimento de Interfaces Gráficas de Usuários reutilizáveis e com partes independentes. A Figura 19 mostra um dos componentes JSX desenvolvidos no trabalho.

Figura 19 - Componente React JS

```
1  import React from "react";
2
3  import { NavLink } from 'react-router-dom';
4
5  import { Li } from './style';
6
7  const navigationItem = (props) => (
8      <Li>
9          <NavLink
10             to={props.link}
11             exact={props.exact}
12             >{props.children}</NavLink>
13      </Li>
14  )
15
16  export default navigationItem;
```

Fonte: Autor

Para a manipulação e tratamento dos dados fornecidos na interface gráfica de usuário foi desenvolvido uma *Application Programming Interface* (API) utilizando o estilo de arquitetura *Representational State Transfer* (REST), que permite o desenvolvimento de *API's* capazes de serem utilizadas por outras aplicações. *API's* que utilizam o estilo REST possuem uma abstração dos elementos arquitetônicos em um sistema hipermídia distribuído, no qual ignora os detalhes da implementação do componente e da sintaxe do protocolo para dar ênfase nas funções dos componentes, nas restrições de suas interações com outros componentes e na interpretação de elementos de dados significativos, e na comunicação via JSON (Fielding, 2000). O desenvolvimento da API proposta neste trabalho foi realizado com a utilização da linguagem de programação *JavaScript* e o ambiente de execução *Node.js*, que

permite o desenvolvimento de aplicações escaláveis de rede. A Figura 20 demonstra a função responsável por finalizar uma consulta e criar o contrato inteligente para o armazenamento dos dados na rede Blockchain.

Figura 20 - Função de criação e armazenamento do contrato inteligente

```

1  async createContract(req, res, next) {
2      try {
3          const {
4              name,
5              doctor,
6              symptoms,
7              exam,
8              diagnosis,
9              familyHistory,
10             chronicDiseases,
11             allergies,
12             ownerAddress } = req.body
13
14             const consultationData = [
15                 name,
16                 doctor,
17                 symptoms,
18                 exam,
19                 diagnosis,
20                 familyHistory,
21                 chronicDiseases,
22                 allergies,
23                 ownerAddress
24             ];
25
26             const data = await ethController.makeConsultation(consultationData);
27             if (data === undefined) {
28                 return res.status(400).json({ error: 'error storing the information' });
29             }
30
31             return res.json(data);
32
33         } catch (error) {
34             console.log(error);
35             return res.status(500).json({error: 'error'});
36             next(error)
37         }
38     },

```

Fonte: Autor

A linguagem de programação *Solidity* foi utilizada para o desenvolvimento dos Contratos Inteligentes *Ethereum*. Essa tecnologia consiste em uma linguagem de programação de alto nível, estaticamente tipadas com suporte a bibliotecas, tipos complexos definidos pelo usuário e criação de contratos personalizados. O *Solidity* possui influência das linguagens de programação C++, *JavaScript* e *Python*, e foi desenvolvido para funcionar no compilador *Ethereum virtual Machine*(EVM). A Figura

21 apresenta o contrato inteligente desenvolvido para o armazenamento das consultas na Rede Blockchain.

Figura 21 - Contrato inteligente da consulta.

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity >=0.8.10;
3
4 import "../entities/ConsultationStruct.sol";
5 import "../definitions/ConsultationData.sol";
6
7 contract Consultation is IconsultationData {
8     address public owner;
9     consultationStruct clinicalData;
10
11     constructor() {
12         owner = msg.sender;
13     }
14
15     modifier onlyBy(address _address) {
16         require(
17             msg.sender == _address || tx.origin == _address,
18             "you are not the owner of the contract"
19         );
20         _;
21     }
22
23     modifier onlyExternal() {
24         require(
25             msg.sender != tx.origin,
26             "only another contract can call this method"
27         );
28         _;
29     }
30
31     modifier validAddress(address _addr) {
32         require(_addr != address(0), "Invalid address");
33         _;
34     }
35
36     function setConsultation(consultationStruct memory _clinicalData) public {
37         clinicalData = _clinicalData;
38         changeOwner(_clinicalData.patientAddress);
39     }
40
41     function changeOwner(address _newOwner)
42     private
43     onlyBy(owner)
44     validAddress(_newOwner)
45     {
46         owner = _newOwner;
47     }
48
49     function getConsultation()
50     public
51     view
52     onlyBy(owner)
53     returns (consultationStruct memory)
54     {
55         return clinicalData;
56     }
57
58     function getConsultation(address _reader)
59     public
60     view
61     onlyBy(_reader)
62     onlyExternal
63     returns (consultationStruct memory)
64     {
65         return clinicalData;
66     }
67 }

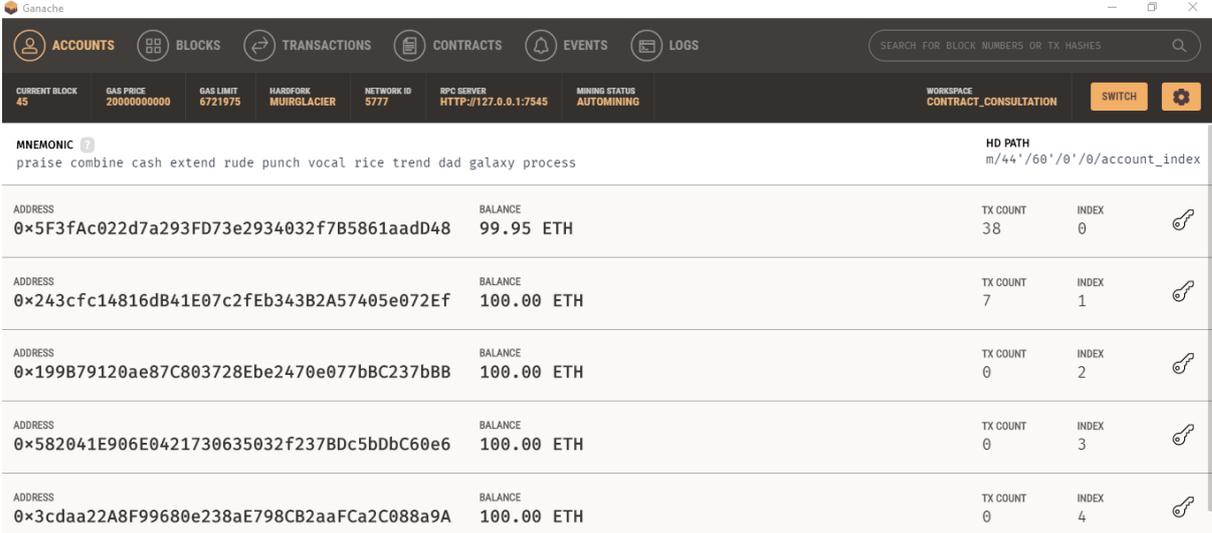
```

Fonte: Autor

Os dados da API são registrados através de um sistema de Armazenamento de Poliglota. Esse consiste no uso de diferentes bases de dados para o armazenamento dos mesmos (Kaur et al, 2015). Para os registros das informações no banco de dados Relacional foi utilizado o Sistema de Gerenciamento de Banco de Dados (SGBD)

PostgreSQL devido a sua velocidade de armazenamento, extensibilidade e escalabilidade. Para o armazenamento na Blockchain foi utilizada a aplicação Ganache, uma rede Blockchain pessoal para desenvolvimento de *DApps Ethereum* que possibilita a criação de Rede Blockchain pré-configurada e disponibiliza contas de usuários para realização das transações. A Figura 22 mostra a tela inicial da Rede Blockchain, com as opções de visualização dos blocos, transações, contratos, eventos, log, contas de usuário e configurações da rede no menu abaixo.

Figura 22 - Interface do Ganache



The screenshot shows the Ganache interface with a dark theme. At the top, there are navigation tabs for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. Below the tabs, a status bar displays various network metrics: CURRENT BLOCK (45), GAS PRICE (2000000000), GAS LIMIT (6721975), HARDFORK (MUIRGLACIER), NETWORK ID (5777), RPC SERVER (HTTP://127.0.0.1:7545), MINING STATUS (AUTOMINING), and WORKSPACE (CONTRACT_CONSULTATION). A search bar is located on the right side of the status bar. The main content area displays account information, including a mnemonic phrase, an HD path, and a list of accounts with their addresses, balances, transaction counts, and indices.

MNEMONIC		HD PATH	
praise combine cash extend rude punch vocal rice trend dad galaxy process		m/44'/60'/0'/0/account_index	
ADDRESS	BALANCE	TX COUNT	INDEX
0x5F3fAc022d7a293FD73e2934032f7B5861aadD48	99.95 ETH	38	0
0x243cfc14816dB41E07c2fEb343B2A57405e072Ef	100.00 ETH	7	1
0x199B79120ae87C803728Ebe2470e077bBC237bBB	100.00 ETH	0	2
0x582041E906E0421730635032f237BDc5bDbC60e6	100.00 ETH	0	3
0x3cdaa22A8F99680e238aE798CB2aaFCa2C088a9A	100.00 ETH	0	4

Fonte: Autor

4. Avaliação da Solução Proposta

Este capítulo tem como objetivo apresentar as etapas de avaliação da aplicação proposta e está dividido da seguinte forma. Na seção 4.1 são apresentados os principais pontos para o desenvolvimento do cenário de avaliação. Na seção 4.2 são apresentados os resultados das avaliações de persistência na Blockchain utilizando os contratos inteligentes, e o permissionamento de acesso aos dados clínicos das consultas.

4.1 Cenário da Avaliação

Para mensurar a capacidade de funcionamento e desempenho da aplicação a respeito do armazenamento dos dados, foi realizada a implementação de recursos de análise dos dados persistentes na rede Blockchain. Para validar a integridade dos dados foi realizada a comparação das informações enviadas para armazenamento na Blockchain com as informações registradas.

O computador utilizado como servidor web para a realização dos testes possui como configuração, um processador i5-7200U com quatro núcleos de processamento e 8GB de memória RAM. Com todos os requisitos de funcionamentos atendidos, foi realizada a inicialização do servidor através do comando *npm start* e da rede Blockchain com a inicialização do software do *Ganache*.

Os testes locais foram realizados utilizando um script desenvolvido para realizar as cargas de dados e visualizar a aplicação em funcionamento. Ao todo, foram utilizadas três cargas de dados. A primeira com mil registros, a segunda com dez mil registros, e a terceira com vinte mil registros.

O script utilizado para a execução das cargas é responsável por realizar a criação da consulta e o seu armazenamento através da geração de informações randômicas pré estabelecidas. Cada atendimento é gerado e enviado gradualmente, e quando um atendimento é registrado o servidor emite uma resposta para que o seguinte atendimento seja realizado. A Figura 23 mostra a lógica do script desenvolvido.

Figura 23 - Script de geração de cargas de armazenamento

```

1  const fetch = require('node-fetch');
2
3  const appointmentLog = (err, ok, numberRequisition) => {
4    console.warn(`[${new Date()}] :\n -->`);
5    console.log(`Erros: ${err}, ok: ${ok}, Requisição número: ${numberRequisition}`);
6  };
7
8  const makeAppointmentData = () => {
9    const name = (Math.random().toString(36).substring(7));
10   const doctor = (Math.random().toString(36).substring(7));
11   const symptoms = (Math.random().toString(36).substring(7));
12   const exam = (Math.random().toString(36).substring(7));
13   const diagnosis = (Math.random().toString(36).substring(7));
14   const familyHistory = (Math.random().toString(36).substring(7));
15   const chronicDiseases = (Math.random().toString(36).substring(7));
16   const allergies = (Math.random().toString(36).substring(7));
17   const medicine = (Math.random().toString(36).substring(7));
18   const ownerAddress = "0x809d6e80c823080827A4Ee5e066262B6e343056E";
19
20   return {
21     name,
22     doctor,
23     symptoms,
24     exam,
25     diagnosis,
26     familyHistory,
27     chronicDiseases,
28     allergies,
29     medicine,
30     ownerAddress
31   }
32 }
33
34 const sendAppointment = async (headers, data) => {
35
36   return await (
37     await fetch('http://localhost:3001/createContract', {
38       method: "POST",
39       body: JSON.stringify(data),
40       headers
41     })
42   )
43 }
44
45 async function main() {
46   const headers = { 'Content-Type': 'application/json' };
47
48   let ok = 0;
49   let error = 0;
50   const InitialHour = 'Initial Hour: ${new Date()}';
51
52   for (let i = 0; i < 4; i++) {
53     setTimeout(async () => {
54       const appointment = makeAppointmentData();
55
56       const result = await sendAppointment(headers, appointment);
57
58
59       if (result.status === 200) {
60         ok = ok + 1;
61       } else {
62         error = error + 1;
63       }
64
65       appointmentLog(error, ok, i);
66
67       if (i == 3) {
68         console.log(InitialHour)
69         console.log('Finish hour' + new Date())
70       }
71     }, 200 * i);
72   }
73
74
75
76 }
77
78 main();

```

Fonte: Autor.

O segundo cenário de avaliação foi desenvolvido para avaliar o sistema de permissionamento de dados na Blockchain. Para isso, foram utilizados alguns dos contratos gerados no teste anterior e executado o teste de permissionamento do contrato.

4.2 Resultados Encontrados

Os resultados obtidos no cenário de teste de persistência na Blockchain foram capazes de mensurar a estabilidade, confiabilidade e eficiência de uso da aplicação. A tabela 1 exibe a relação entre carga e tempo de duração para o armazenamento na Blockchain. Através dos resultados é possível notar a viabilidade técnica de utilização do sistema com relação ao tempo de duração para a finalização do armazenamento.

A avaliação de taxas de erros das trinta e um mil requisições enviadas obteve como resultado 100% de êxito no armazenamento dos dados. A Tabela 2 exibe a relação de requisições armazenadas e não armazenadas na Blockchain.

Tabela 1 - Carga e tempo de duração no armazenamento na Blockchain

Nº de Requisições	Duração	Tempo Médio por Requisição
1000	00:17:16 (hh:mm:ss)	1,03 segundo
10000	02:29:08 (hh:mm:ss)	0,89 segundo
20000	03:02:35 (hh:mm:ss)	0,54 segundo

Fonte: Autor.

Tabela 2 - Relação de requisições armazenadas e não armazenadas na Blockchain

Nº de Requisições Enviadas	Nº de Armazenamentos Realizados	Nº de Armazenamentos não Realizados
31000	31000	0

Fonte: Autor.

No cenário de avaliação de permissionamento dos dados, todas as requisições de acesso com contas autorizadas foram realizadas com sucesso. A Tabela 3 mostra os resultados obtidos nos cenários de avaliação do permissionamento dos dados. Enquanto a Figura 24 apresenta as informações retornadas após o acesso permissionado ser realizado e a Figura 25 apresenta o erro retornado caso o usuário não tenha permissão de acesso.

Tabela 3 - Resultado da avaliação de acesso permissionado

Nº de Acessos concedidos	Nº de acessos realizados	Nº de erros de acesso
150	150	0

Fonte: Autor.

Figura 24 – Informações da consulta retornadas com o acesso permissionado.

```

1 = {
2   "name": "0bguei",
3   "doctor": "b7xq3g",
4   "symptoms": "3fv3yn",
5   "exam": "uvkswa",
6   "diagnosis": "mcegrf",
7   "familyHistory": "8qpufr",
8   "chronicDiseases": "ztwwd7",
9   "allergies": "qofdet",
10  "medicine": "reqeax",
11  "patientAddress": "0x809d6e80c823080827A4Ee5e06626286e343056E"
12 }

```

Fonte: Autor.

Figura 25 – Erro de acesso não permissionado

```
1 {  
2   "error": "you are not allowed to read this clinical data"  
3 }
```

Fonte: Autor.

Ao final das avaliações foi possível constatar o pleno funcionamento da aplicação sem problemas que impedissem o seu uso. Dessa forma, além da utilização completa da aplicação, pode-se realizar apenas o uso da API por outros aplicativos de saúde, a fim de realizar o armazenamento dos dados clínicos da consulta na Blockchain e o seu permissionamento com médicos de outras instituições.

5. Conclusão

Este capítulo apresenta, na seção 5.1, as considerações finais com uma visão geral acerca da solução computacional desenvolvida neste Trabalho de Conclusão de Curso e a seção 5.2 apresenta as indicações de trabalhos futuros.

5.1 Considerações Finais

Esse trabalho apresentou o desenvolvimento de uma solução computacional capaz de auxiliar as organizações do setor de saúde nos atendimentos clínicos, utilizando o armazenamento na tecnologia Blockchain. As principais contribuições deste trabalho incluem o entendimento do funcionamento e características da tecnologia Blockchain, o desenvolvimento de uma arquitetura de software capaz de ser utilizada por diversos sistemas de armazenamento de dados, e a especificação de contratos inteligentes capazes de armazenar consultas e realizar o gerenciamento de permissões de acesso.

Com o objetivo de avaliar as principais características da aplicação proposta, foram realizados testes de armazenamento com três cargas de dados, no qual foi possível obter o resultado esperado com todas as requisições realizadas e um tempo médio de resposta satisfatório para o armazenamento das informações. Além disso, foi realizado o teste do gerenciamento de permissões de acesso que também alcançou o resultado esperado em todos os testes.

A utilização de soluções computacionais que vise trazer maior confiabilidade dos dados armazenados no setor de saúde é de fundamental importância devido à sensibilidade das informações neles armazenadas. Diante do exposto foi reconhecida a viabilidade técnica do uso da solução computacional aqui proposta, para utilização em ambientes reais, bem como, a utilização da API por outros sistemas de saúde presentes no mercado.

5.2 Trabalhos futuros

Como trabalhos futuros é possível destacar a utilização de criptografia no envio das mensagens entre as aplicações; a avaliação da aplicação desenvolvida em um cenário real; a utilização de mecanismos que visem reduzir o tempo e custo por

armazenamento e realizar um estudo e análise do que foi desenvolvido na solução computacional está de acordo com a Lei Geral de Proteção de Dados Pessoais (LGPD).

Referência

- Bowman, S. (2013) "Impact of Electronic Health Record Systems on Information Integrity: Quality and Safety Implications". *Perspectives in health information management / AHIMA, American Health Information Management Association*. vol. 10, 1c.
- Hölbl, M., Kompara, M., Kamisalic, A., Nemec, L. (2018) "A Systematic Review of the Use of Blockchain in Healthcare". *Symmetry*. Vol. 10, pp. 470. Doi: 10.3390/sym10100470.
- Ante, Lennart. Smart contracts on the blockchain—a bibliometric analysis and review. *Telematics and Informatics*, v. 57, p. 101519, 2021.
- Quaini, T., Roehrs, A., André da Costa, C., Righi, R. (2018) "A Model For Blockchain-Based Distributed Electronic Health Records". *IADIS INTERNATIONAL JOURNAL ON WWW/INTERNET*. vol. 16, pp. 66-79. Doi:10.33965/ijwi_2018161205.
- Liang, W., Fan, Y., Li, K., Zhang, D., Gaudiot, J. (2020) "Secure Data Storage and Recovery in Industrial Blockchain Network Environments". *IEEE Transactions on Industrial Informatics*. Doi: 10.1109/TII.2020.2966069.
- Araújo, A.; TIMES, V. C. ; SILVA, M. U. From Data Requirements to Health Applications: A Tool for Dynamic Generation of Data Schemas and Graphical User Interfaces Using Archetypes. In: *Brazilian Symposium on Databases (SBBD), 2019, Fortaleza. Brazilian Symposium on Databases (SBBD), Demo Track, 2019*. p. 24-29. ISSN: 2016-5170
- Araújo, A., Times, V., Silva, M. (2020) "A Tool for Generating Health Applications Using Archetypes". *IEEE Software*, vol. 37, pp. 60-67. Doi: 10.1109/MS.2018.110162508.
- Kaur, Gagandeep Singh, Jasvir Review on Machine Learning Techniques for Big Data Management and Open Research Challenges, *International Journal of Computer Sciences and Engineering* 10.26438/ijcse/v6i7.10521055
- Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H. (2017) "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends". *IEEE International Congress on Big Data (BigData Congress)*, vol. 85, pp. 557-564. Doi: 10.1109/BigDataCongress.2017.85.
- Huang, E., Hsiao, S., Liou, D. (2003) "Design and implementation of a web-based HL7 message generation and validation system". *International Journal of Medical Informatics*, vol.70, pp. 49-58. Doi: 10.1016/S1386-5056(03)00006-6.
- Nakamoto, S. (2009) "Bitcoin: A Peer-to-Peer Electronic Cash System". Accessed on: Mar. 12, 2020. [Online]. Available: <https://git.dhimmel.com/bitcoin-whitepaper>.
- Xu, R., Nikouei, S., Chen, Y., Blasch, E., Aved, A. (2019) "BlendMAS: A BLockchain-ENabled Decentralized Microservices Architecture for Smart Public Safety". Doi: 10.1109/Blockchain.2019.00082.

Surya, R., Charlyn, G., Latha, C. (2020). "Blockchain: A Panacea for Healthcare Cloud - based Data Security and Privacy". Test Engineering and Management, vol. 82, pp. 6671- 6676.

Maher alharby and Aad van Moorsel, "Blockchain-Based Smart Contracts: A Systematic Mapping Study", Cs & IT-CSCP, 2017;

Alcazar, V. Data You Can Trust: Blockchain Technology, 2017.

Johnston, D; Yilmaz, S. O; Kandah, J; Bentenitis, N; Hashemi, F; Gross, R; Wilkinson, S; and MASON, S. The General Theory of Decentralized Applications DApps, Github, June, 2014.

Zheng, Zibim. Blockchain Challenges and opportunities: A survey. International Journal of Web and Grid Services, 2018.

Buterin, V. "A Next Generation Smart Contract & Decentralized Application Platform", 2015.

Osetskyi, V. What Are Smart Contracts and Their Use Cases in Business, 2018.

Mohan, C. Tutorial: blockchains and databases. Proceedings of the VLDB Endowment, v. 10, n. 12, p. 2000-2001, 2017.

ALCAZAR, Col Vincent. Data you can trust. Air and Space Power Journal, v. 31, n. 2, p. 91-101, 2017.

Nagothu, Deeraj et al. A microservice-enabled architecture for smart surveillance using blockchain technology. In: 2018 IEEE international smart cities conference (ISC2). IEEE, 2018. p. 1-4.

Alharby, Maher; Van Moorsel, Aad. Blockchain-based smart contracts: A systematic mapping study. arXiv preprint arXiv:1710.06372, 2017.

Rubio, María Alejandra; Tarazona, Giovanni Mauricio; Contreras, Leonardo. Big data and blockchain basis for operating a new archetype of supply chain. In: International Conference on Data Mining and Big Data. Springer, Cham, 2018. p. 659-669.

Da conceição, Arlindo Flavio et al. Electronic health records using blockchain technology. arXiv preprint arXiv:1804.10078, 2018.

Zhang, Peng et al. Design of blockchain-based apps using familiar software patterns with a healthcare focus. In: Proceedings of the 24th Conference on Pattern Languages of Programs. 2017. p. 1-14.

Legaz-garcía, María del Carmen et al. Transformation of standardized clinical models based on OWL technologies: from CEM to OpenEHR archetypes. Journal of the American Medical Informatics Association, v. 22, n. 3, p. 536-544, 2015.

MIRANDA, Nelson Júlio de Oliveira; PINTO, Virginia Bentes. O padrão openEHR aplicado como Sistema de Organização do Conhecimento Clínico. 2014.

Roehrs, Alex et al. Analyzing the performance of a blockchain-based personal health record implementation. *Journal of biomedical informatics*, v. 92, p. 103140, 2019.

Zhang, Schmidt, and White, "Blockchain Technology Use Cases in Healthcare", Varian Medical Systems, Palo Alto, CA 2018.

Lanxiang Chen, Wai-Kong Lee, Chin-Chen Chang, Kim-Kwang Raymond Choo, Nan Zhang, "Blockchain based searchable encryption for electronic health record sharing", *Future Generation Computer Systems*, V.95, 2019.

Haiping Huang, Xiang Sun, Fu Xiao, Peng Zhu, Wenming Wang, "Blockchain-based eHealth system for auditable EHRs manipulation in cloud environments", *Journal of Parallel and Distributed Computing*, V.148, 2021.

Gayathri Nagasubramanian, Rakesh Kumar Sakthivel, Rizwan Patan, Amir H. Gandomi, Muthuramalingam Sankayya, Balamurugan Balusamy, "Securing e-health records using keyless signature infrastructure blockchain technology in the cloud", *Neural Computing and Applications*, 2020.

Naveed Islam, Yasir Faheem, Ikram Ud Din, Muhammad Talha, Mohsen Guizani, Mudassir Khalil, "A blockchain-based fog computing framework for activity recognition as an application to e-Healthcare services", *Future Generation Computer Systems*, V. 100, 2019.

Ahmed F. Hussein, N. ArunKumar, Gustavo Ramirez-Gonzalez, Enas Abdulhay, João Manuel R.S. Tavares, Victor Hugo C. de Albuquerque, "A medical records managing and securing blockchain based system supported by a Genetic Algorithm and Discrete Wavelet Transform", *Cognitive Systems Research*, V.52, 2018.

Gordon, William J.; Catalini, Christian. *Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability*. *Computational and structural biotechnology journal*, v. 16, p. 224-230, 2018.

Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M. and Wang, F. (2017) 'Secure and Trustable Electronic Medical Records Sharing using Blockchain', *AMIA Annual Symposium proceedings*. AMIA Symposium, AMIA, Washington, DC, USA, Vol. 2017, pp. 650–659.

Peterson, K. J., Rammohan, D., Pradip, K. and Mayo, K. (2016) 'A Blockchain-Based Approach to Health Information Exchange Networks', *Proc. NIST Workshop Blockchain Healthcare*, Vol. 1, pp. 1-10.

Azaria, A., Ekblaw, A., Vieira, T. and Lippman, A. (2016) 'MedRec: Using Blockchain for Medical Data Access and Permission Management', *2nd International Conference on Open and Big Data (OBD)*, IEEE, Vienna, Austria, pp. 25–30.

ISO. *Health informatics-Personal health device communication Part 10407: Device specialization - Blood pressure monitor*, 2008.

Zhang, Peng et al. FHIRChain: applying blockchain to securely and scalably share clinical data. Computational and structural biotechnology journal, v. 16, p. 267-278, 2018.

Beale, T. and HEARD, S. OpenEHR - Architecture Overview. The OpenEHR Foundation, p-79, 2007.

Noumeir, R. and PAMBRUN, J. F. Hands-on approach for teaching {HL7} version 3. Information Technology and Applications in Biomedicine, 2010.

Bacelar, Gustavo; CORREIA, Ricardo. openEHR. 2015.

Araújo, André Magno Costa de. Um framework para criação de esquemas de dados, geração de interfaces gráficas de usuário e persistência poliglota do RES utilizando arquétipos. 2018.

Hoy, Derek et al. A feasibility study on clinical templates for the National Health Service in Scotland. In: Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems. IOS Press, 2007. p. 770.

Braga, Renata Dutra; De Lucena, Fábio Nogueira; de Sá Leitão-júnior, Plínio. Registro Eletrônico em Saúde Interoperável: os desafios do padrão OpenEHR. Journal of Health Informatics, v. 8, n. 3, 2016.

Moraes, João Paulo, Componentes React: Componentes de classe e funcional sem estado, Geekhunter, 2019. Disponível em: <<https://blog.geekhunter.com.br/criando-componentes-react-componentes-de-classe-e-funcional-sem-estado/>>. Acessado em: 22/12/2021.

Kaur, K., RANI, R. Managing Data in Healthcare Information Systems: Many Models, One Solution; IEEE Computer Society, 2015.

Fielding, Roy Thomas. Architectural styles and the design of network-based software architectures. University of California, Irvine, 2000.