UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

NATÁLIA RODRIGUES JUNQUEIRA

Concessão de Permissão a Dados de Saúde Baseada em Contratos Inteligentes em Plataforma de Blockchain

Goiânia
2020

sibi sistema de bibliotecas ufg

PRPG
PRÓ-REITORIA DE PÓS-GRADUAÇÃO

UFG

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das dissertações e teses disponibilizados são de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o autor e o orientador firmam o compromisso de que ele não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico: [X] **Dissertação** [ ] **Tese**

2. Identificação da Tese ou Dissertação:

Nome completo do autor: Natália Rodrigues Junqueira

Título do trabalho: Concessão de Permissão a dados de Saúde Baseada em Contratos Inteligentes em Plataforma de Blockchain

3. Informações de acesso ao documento:

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

Independente da concordância com a disponibilização eletrônica, é imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.

Natália Rodrigues Junqueira
Assinatura do(a) autor(a)[2]

Ciente e de acordo:

Assinatura do(a) orientador(a)[2]

Data: 04/03/2020

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

[2] As assinaturas devem ser originais sendo assinadas no próprio documento, imagens coladas não serão aceitas.

NATÁLIA RODRIGUES JUNQUEIRA

Concessão de Permissão a Dados de Saúde Baseada em Contratos Inteligentes em Plataforma de Blockchain

Dissertação apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Prof. Dr. Sérgio Teixeira de Carvalho

Goiânia
2020

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Rodrigues Junqueira, Natália
Concessão de Permissão a Dados de Saúde Baseada em Contratos Inteligentes em Plataforma de Blockchain [manuscrito] / Natália Rodrigues Junqueira. - 2020.
LXXXVII, 87 f.: il.

Orientador: Prof. Dr. Sérgio Teixeira de Carvalho .
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2020.
Bibliografia. Apêndice.
Inclui siglas, fotografias, gráfico, lista de figuras, lista de tabelas.

1. Blockchain. 2. Personal Health Records. 3. Pacient Centric Agent. 4. Concessão de Permissão. 5. Confiança. I. , Sérgio Teixeira de Carvalho, orient. II. Título.

CDU 004

NATÁLIA RODRIGUES JUNQUEIRA

Concessão de Permissão a Dados de Saúde Baseada em Contratos Inteligentes em Plataforma de Blockchain

Dissertação defendida no Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Ciência da Computação, aprovada em 07 de Fevereiro de 2020, pela Banca Examinadora constituída pelos professores:

Prof. Dr. Sérgio Teixeira de Carvalho
Instituto de Informática – UFG
Presidente da Banca

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº **04/2020** da sessão de Defesa de Dissertação de Natália Rodrigues Junqueira, que confere o título de Mestra em **Ciência da Computação**, na área de concentração em **Ciência da Computação**.

Aos sete dias do mês de fevereiro de dois mil e vinte, a partir das nove horas, na sala 150 do Instituto de Informática, realizou-se a sessão pública de Defesa de Dissertação intitulada **"Concessão de Permissão a Dados de Saúde Baseada em Blockchain"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Sérgio Teixeira de Carvalho (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Iwens Gervásio Sene Júnior (INF/UFG), membro titular externo; Professor Doutor Roberto Vito Rodrigues Filho (Lancaster University), membro titular externo. Durante a arguição os membros da banca fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido a candidata **aprovada** pelos seus membros. Proclamados os resultados pelo Professor Doutor Sérgio Teixeira de Carvalho, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos sete dias do mês de fevereiro de dois mil e vinte.

TÍTULO SUGERIDO PELA BANCA

Concessão de Permissão a Dados de Saúde Baseada em Contratos Inteligentes em Plataforma de Blockchain

Documento assinado eletronicamente por **Sérgio Teixeira De Carvalho**, **Professor do Magistério Superior**, em 07/02/2020, às 11:57, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Iwens Gervasio Sene Junior**, **Professor do Magistério Superior**, em 07/02/2020, às 11:58, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Roberto Vito Rodrigues Filho**, **Usuário Externo**, em 07/02/2020, às 12:02, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1148564** e o código CRC **84918388**.

Referência: Processo nº 23070.047867/2019-19 SEI nº 1148564

Agradecimentos

Agradeço à minha família pelo apoio e incentivo, sempre me animando e acreditando no meu potencial.

Agradeço aos meus amigos que tiveram paciência e compreensão nos momentos que tive que me dedicar ao trabalho.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo auxílio financeiro.

Agradeço ao meu orientador Sérgio Teixeira de Carvalho pelo tempo e dedicação cedidos durante o desenvolvimento desse trabalho.

Resumo

JUNQUEIRA, N. R.. **Concessão de Permissão a Dados de Saúde Baseada em Contratos Inteligentes em Plataforma de Blockchain**. Goiânia, 2020. 89p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

O surgimento de dispositivos vestíveis tem permitido às pessoas o monitoramento do seu próprio estado de saúde mesmo estando em um ambiente domiciliar. Além disso, também tem havido uma conscientização da população que a saúde é algo fundamental, fazendo com que cresça o interesse das pessoas pelo controle da própria saúde, pelo rastreamento e análise desses dados pessoais de saúde. A abordagem da medicina centrada na pessoa surgiu com o objetivo de possibilitar controle desses dados de saúde para as pessoas a quem eles realmente pertencem, ou seja, ao próprio paciente. Desse modo, um paciente poderia não só acompanhar esses dados, mas efetivamente decidir quem pode ou não ter acesso a eles. Blockchain é uma tecnologia que surgiu com o intuito de realizar transações seguras pela internet sem a necessidade de um terceiro de confiança; no contexto de dados de saúde o terceiro de confiança seria um hospital ou uma entidade que armazene e controle os dados coletados dos pacientes. Esta pesquisa visa desenvolver uma solução arquitetural baseada em blockchain para concessão de permissão de acesso aos dados de saúde do paciente coletados por um sistema de monitoramento remoto de pacientes por meio de relacionamentos numa rede social. No sentido de validar a solução foram implementados contratos inteligentes (smart contracts) para identificar, rastrear dados e permitir a concessão de permissão de acesso aos dados do paciente. Os contratos inteligentes foram desenvolvidos e testados utilizando a plataforma blockchain Hyperledger Fabric.

Palavras–chave

Blockchain, Personal Health Records, Pacient Centric Agent, Concessão de permissão, Confiança.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Este capítulo apresenta a motivação para realização deste trabalho, o problema, as hipóteses de solução do problema e os objetivos. Também é apresentada a organização dos capítulos dessa dissertação.

1.1 Motivação

Durante toda a vida de uma pessoa dados de saúde são produzidos e armazenados em bancos de dados aos quais os próprios pacientes não têm acesso, não controlam quem pode acessá-los ou o que pode ou não ser feito com esses dados [11, 64]. Com os avanços tecnológicos e a conscientização da população de que a saúde é algo fundamental, tem crescido o interesse das pessoas pelo controle da própria saúde, rastreamento e análise desses dados pessoais de saúde. O uso de dispositivos vestíveis e sensores para realizar o monitoramento contínuo dos dados de saúde, tem permitido aos pacientes acompanhar seu estado de saúde por meio de aplicativos que se comunicam com esses dispositivos [27, 53].

A medicina centrada na pessoa tem como objetivo devolver o controle desses dados de saúde a quem eles realmente pertencem, ou seja, ao próprio paciente. Desse modo poderia não só acompanhar esses dados, mas efetivamente decidir quem pode ou não ter acesso a eles [61, 17]. Uma pessoa nesse contexto pode ser um paciente com alguma doença crônica que utiliza um sistema de monitoramento remoto de pacientes para ter a assistência domiciliar à saúde [31, 30, 54, 55], ou uma pessoa que tenha interesse em monitorar seus dados de saúde continuamente. Paciente, nessa pesquisa, é a pessoa monitorada por sensores e dispositivos vestíveis no contexto de um sistema de monitoramento remoto de pacientes.

Um sistema de monitoramento remoto de pacientes (SMRP) é uma especialização de um sistema sensível a contexto, ou seja, um sistema capaz de prover serviços e informações baseados no contexto [23]. De acordo com Dey e Abowd [5], contexto se refere a "qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, um local ou um objeto relevante para a interação

entre o usuário e a aplicação, incluindo os próprios usuários e aplicações". Um SMRP, no contexto dessa dissertação, é aquele que utiliza dispositivos (sejam vestíveis, sensores, ou algo do tipo) para coletar dados remotamente, enviá-los para uma central de diagnósticos, onde após processados geram resultados que podem ser disseminados através de notificações em uma rede social [5, 23].

Essa disseminação de notificações pode ser realizada por meio de uma rede social, onde os relacionamentos entre os usuários dessa rede são interpretados como informações contextuais. O relacionamento na rede social é, portanto, utilizado como uma forma de descrever uma interação entre os seus usuários.

O trabalho realizado em [54, 55] foi desenvolver um módulo de rede social (UbiCare Social) para disseminação das notificações do SMRP denominado UbiCare [30, 23, 54, 55], onde um relacionamento entre um profissional de saúde e um paciente significa que o profissional possui interesse em receber dados daquele paciente, ao mesmo tempo em que significa que o paciente concede acesso aos seus dados de saúde àquele profissional.

Esse contexto de monitoramento remoto de pacientes onde a rede social específica para dados de saúde é utilizada para disseminar os dados coletados do paciente para seus familiares ou profissionais de saúde é o cenário em que essa dissertação está inserida.

1.2 Problema

A partir desse contexto, em que os pacientes têm interesse em conceder acesso aos seus dados de saúde a outras pessoas e querem ter a garantia de que somente pessoas às quais ele autorizou terão acesso aos seus dados, surge, como uma forma de demonstrar esse interesse, o uso dos relacionamentos entre os usuários de uma rede social, de maneira similar à utilizada por [54, 55] no UbiCare Social.

Uma questão não tratada pelo UbiCare Social [54, 55], no entanto, está relacionada à privacidade desses dados de saúde coletados dos pacientes. Essa questão é algo fundamental para aumentar a confiança dos pacientes em abordagens de monitoramento remoto de pacientes, pois sem a garantia de que somente pessoas autorizadas devem ter acesso a esses dados, além de comprometer a privacidade dessa pessoa que está sendo monitorada, pode comprometer a sua segurança.

Por exemplo, uma pessoa mal intencionada com acesso não autorizado aos dados de saúde de um paciente pode usar esses dados para descobrir a sua localização, se ele possui alguma doença, qual a rotina de cuidado e tratamento, se ele toma algum medicamento, etc.

A privacidade dos dados de saúde de um paciente é algo tão crítico que os médicos se comprometem a guardar silêncio sobre as informações de seus pacientes "como um

segredo religioso"(juramento hipocrático) [29]. Essa privacidade é algo assegurado pela nossa constituição federal e regido pelo nosso código penal. Portanto, os dados de saúde de um paciente devem estar em princípio sempre protegidos. Isso tem uma função social muito importante para manter a confiabilidade na relação médico-paciente.

Em se tratando de um registro eletrônico desses dados de saúde com uma abordagem centrada no paciente, para garantir a privacidade desse paciente é necessário que o poder de decisão de quem acessa ou não seus dados fique a cargo do próprio paciente. É necessário, portanto, que esse paciente confie que somente pessoas autorizadas por ele poderão obter acesso. Para que seja estabelecida essa relação de confiança é necessário ainda que esses dados de saúde sejam armazenados em um local que garanta a privacidade. Posto esse cenário, o problema tratado por essa dissertação é: como conceder acesso a esses dados sem comprometer a privacidade deles?

1.3 Hipótese

A solução que se poderia utilizar para armazenar e conceder permissão de acesso a dados de saúde dos pacientes deve garantir a privacidade, de modo que apenas pessoas autorizadas pelo paciente consigam acessá-los.

A tecnologia *blockchain* funciona como um livro contábil distribuído (também conhecido como *ledger*) onde todos os nós que compõem a rede *blockchain* possuem um histórico completo de todas as transações que ocorreram, tornando-as auditáveis e transparentes. As transações são armazenadas em blocos no *ledger*, e cada bloco possui em seu cabeçalho um *hash* do bloco anterior [50, 18, 2, 19, 32, 70].

Uma cadeia de blocos (*blockchain*) imutável é formada por meio de um protocolo de consenso que registra as transações com confiança sem a necessidade de se confiar em um terceiro(*e.g.*, banco). Além disso, essa tecnologia também utiliza o conceito de contratos inteligentes implementados na forma de *scripts* escritos em uma determinada linguagem de programação e armazenados na rede *blockchain*. As regras descritas no contrato inteligente funcionam de forma similar a uma lei, pois uma rede *blockchain* é praticamente imutável e os contratos armazenados nela herdam essa característica. Em outras palavras, o que for acordado em um contrato é exatamente o que será executado pela rede [50, 18, 2, 19, 32, 70].

Os contratos inteligentes podem ser usados como uma forma de personalizar o funcionamento da rede *blockchain*. Além disso, podem ser utilizados para estabelecer as regras de concessão de permissão de acesso aos dados de saúde do paciente garantindo a privacidade dos mesmos. Eles poderiam ser construídos com base nos relacionamentos entre usuários de uma rede social, onde um relacionamento entre um paciente e um profissional de saúde significaria a concessão de permissão de acesso a dados de saúde

desse paciente. O relacionamento de um profissional de saúde com um paciente nessa rede social significaria o interesse desse profissional de saúde em ter acesso a dados de saúde desse paciente. A tecnologia *Blockchain* e os Contratos Inteligentes são vistos com detalhes na 2.1.4 do Capítulo 2.

1.4 Objetivos

O objetivo geral desse trabalho é investigar a viabilidade da implementação dos contratos inteligentes implantados em um blockchain, que permitam ao paciente a concessão de acesso a seus dados pessoais de saúde coletados por um sistema de monitoramento remoto de pacientes, através de relacionamentos em uma rede social garantindo privacidade. Os objetivos específicos deste trabalho são:

- Investigar o estado da arte sobre a tecnologia *Blockchain* aplicada a dados de saúde organizados na forma de registros eletrônicos de saúde;
- Investigar uma forma de representar relacionamentos numa rede social como contratos inteligentes de concessão de permissão;
- Desenvolver uma solução arquitetural que permita ao paciente conceder permissão de acesso aos seus dados pessoais de saúde;
- Avaliar as plataformas *Ethereum* e *Hyperledger Fabric* como alternativas de implementação;
- Implementar contratos inteligentes que concedam permissão de acesso a dados de saúde com privacidade na plataforma *Hyperledger Fabric*;

1.5 Método da Pesquisa

A Figura 1.1 apresenta o método utilizado para realização da pesquisa desde a identificação do problema até a sua conclusão.

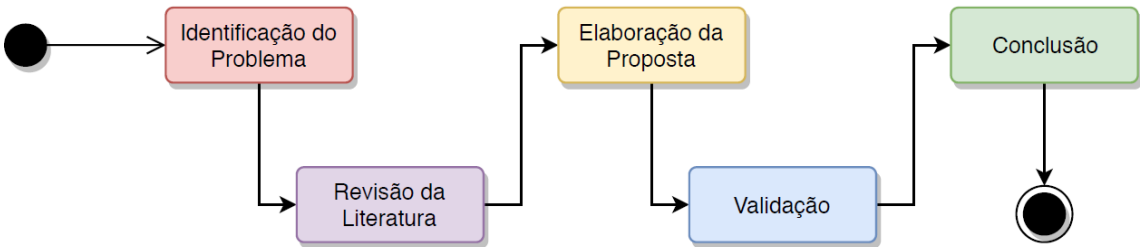


Figura 1.1: Método da Pesquisa desta Dissertação

1. *Identificação do Problema:* A pesquisa iniciou-se com a identificação, a definição do problema e o estudo sobre a tecnologia blockchain.
2. *Revisão da Literatura:* A revisão da literatura foi realizada com o objetivo de identificar de que forma a tecnologia blockchain vem sendo aplicada a dados de saúde.
3. *Elaboração da Proposta:* A apresentação do modelo de domínio e da arquitetura da rede social em conjunto com o sistema de monitoramento remoto de pacientes (SMRP), que foram adaptados com intuito de garantir a privacidade dos dados do paciente, mapeando entidades e relacionamentos para contratos inteligentes blockchain.
4. *Validação:* Implementação dos contratos inteligentes na plataforma *Hyperledger Fabric* e da simulação de sua execução por meio de *scripts* que invocam os contratos inteligentes para: (I) identificar os usuários da rede social, (II) rastrear dados do paciente desde a coleta dos dados através de sensores até o acesso das pessoas autorizadas, (III) conceder acesso aos dados a quem o paciente autorizar mantendo um histórico imutável de todas essas transações.
5. *Conclusão:* A conclusão refere-se à análise dos resultados obtidos no trabalho, identificação das suas limitações e definição de trabalhos futuros.

1.6 Organização

Este trabalho está organizado em cinco capítulos, além deste introdutório. No Capítulo 2 são apresentadas as bases para a realização deste trabalho, quais sejam, SMRP e UbiCare Social, a tecnologia blockchain, contratos inteligentes e plataformas blockchain, juntamente com a revisão da literatura. No Capítulo 3 é apresentada a proposta deste trabalho, contendo o modelo de domínio, a arquitetura e os contratos inteligentes. No Capítulo 4 é detalhada uma implementação dos contratos inteligentes, incluindo a configuração da rede *Hyperledger Fabric* e uma discussão sobre o desenvolvimento dos contratos. No Capítulo 5 estão descritos os trabalhos relacionados. Por fim, no Capítulo 6 é apresentada a conclusão, limitações e trabalhos futuros.

CAPÍTULO 2

Referencial teórico

Este capítulo apresenta alguns dos conceitos fundamentais para compreensão do trabalho como um todo. A primeira seção é a que trata dos conceitos que são a base deste trabalho de pesquisa e, na sequência, está a seção que apresenta a revisão da literatura que deu início a este trabalho. Por fim, as considerações finais.

2.1 Conceitos

Essa seção apresenta os conceitos de monitoramento remoto de pacientes (SMRP), abordagem centrada na pessoa, redes sociais e os relacionamentos entre as entidades de um SMRP. Também são abordados os princípios de segurança da informação. A tecnologia *blockchain* também é apresentada com mais detalhes em conjunto com os contratos inteligentes, pois ambos são a base para nossa abordagem de concessão de permissão.

2.1.1 Sistema de Monitoramento Remoto de Pacientes e a Abordagem Centrada na Pessoa

Neste trabalho é utilizada uma abordagem centrada na pessoa(Figura 2.1). Isso implica numa visão onde os dados de saúde são privados e pertencem à própria pessoa. Portanto, todos que tenham interesse nos dados dessa pessoa devem pedir permissão à própria pessoa para acessá-los [1, 61, 17, 66].

Nesse contexto, um Sistema de Monitoramento Remoto de Pacientes (SMRP) pode ser tratado como uma especialização de sistemas sensíveis ao contexto, seguindo a sua arquitetura geral, como apresentada na Figura 2.2. Nessa arquitetura a camada de sensores é a responsável por coletar os dados daquele contexto, passá-los para a camada de aquisição dos dados que determina como será realizada a coleta dos mesmos. Os dados são também encaminhados para a camada de pré-processamento.

Depois, esses dados são passados para a camada de armazenamento e gerência dos dados, importante para este trabalho, pois controla como os dados são armazenados e

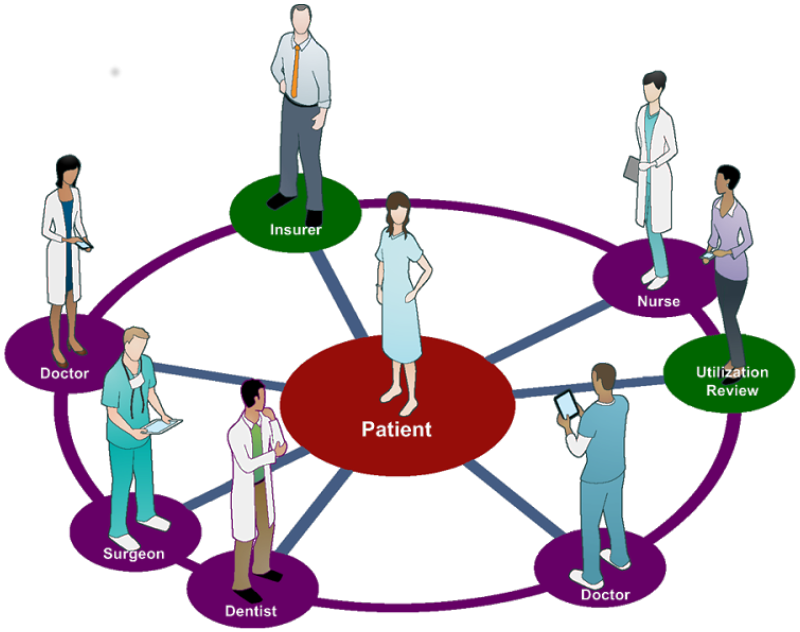


Figura 2.1: Medicina com Abordagem Centrada na Pessoa [1]

Figura 2.2: Arquitetura Geral de Sistemas Sensíveis a Contexto [54, 55, 23]

quem pode acessá-los. A camada de aplicações mantém as aplicações que dependem dos serviços de todas as camadas anteriores para executar suas funções.

No Instituto de Informática - UFG - há um Projeto de Pesquisa e Desenvolvimento intitulado "Aplicação de Técnicas de Computação Ubíqua, no Contexto do Monitoramento de Pacientes Domiciliares". Durante o desenvolvimento desse projeto foi criada uma plataforma para o Monitoramento Remoto de Pacientes denominado UbiCare[30, 31, 54, 55, 13]. O projeto teve como objetivo investigar e propor a aplicação de técnicas de computação ubíqua e pervasiva, de algoritmos inteligentes, de padrões de Informática em Saúde, e de arquitetura de software, no contexto de uma solução computacional para o monitoramento remoto de pacientes domiciliares.

Visando atingir esses objetivos, o UbiCare foi idealizado como uma solução arquitetural com o propósito de realizar de forma transparente operações no contexto da assistência domiciliar à saúde e monitoramento remoto de pacientes, tanto para os pacientes

e os cuidadores, quanto para os profissionais de saúde. Ela tem como base a arquitetura orientada a serviços, de modo a coordenar as operações das aplicações, garantir a interoperabilidade, manter um baixo acoplamento e permitir o reuso de informações comuns entre elas.[30, 31, 54, 55, 13]

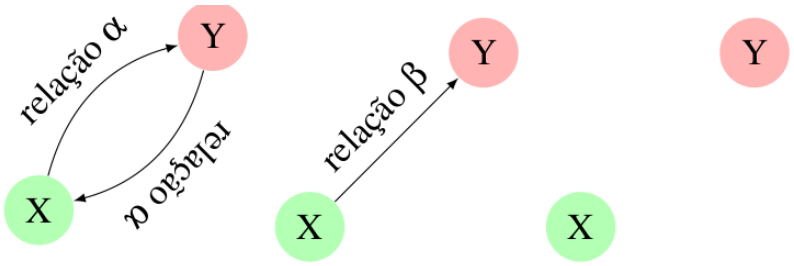
Essa dissertação está inserida no contexto deste projeto, utilizando como base o UbiCare[30], o Simulador de Sensores Fisiológicos[13] e o UbiCare Social[55], com o propósito de tratar a questão da privacidade dos dados do paciente.

2.1.2 Redes Sociais e Relacionamentos entre as entidades do SMRP

Um modo de aperfeiçoar os sistemas de monitoramento remoto de pacientes (SMRP) é o envolvimento das pessoas em torno do indivíduo que necessita de cuidados. Essas pessoas, sejam elas cuidadoras formais (profissionais de saúde) ou cuidadoras informais (familiares e amigos), formam uma rede de cuidadores [40] ou comunidade de interesse [10]. Espera-se que essa comunidade auxilie o paciente no que diz respeito ao seu tratamento como uma rede de amparo àquela pessoa. Além disso, estas comunidades possuem o potencial de conscientizar-se sobre condições de saúde.

Uma forma de permitir a manutenção dessa rede de cuidadores é a integração de serviços de redes sociais a sistemas de monitoramento remoto de pacientes [54, 55]. Os relacionamentos definidos no serviço de rede social podem ser usados para direcionar a rede de cuidadores e as notificações referentes ao estado de saúde de um paciente.

A Figura 2.3 mostra os tipos de relacionamentos entre dois nós em uma rede social, sendo eles: relação mútua, onde ambas as partes estão se relacionando tornando assim a relação bilateral; assimétrica, onde a relação é unilateral - isto implica em apenas um dos nós que compõe a relação enviar informações, e, por fim, relação nula onde não há relação entre os nós.



(a) *Mútua* (b) *Assimétrica* (c) *Nula*

Figura 2.3: Relacionamentos em uma Rede Social entre dois nós representados por grafos denominados díades [23]

A Figura 2.4 mostra visão sobre estes conceitos utilizada nesse trabalho. Indivíduo se refere à pessoa que faz parte de uma Rede Social, um conjunto de indivíduos

relacionados pelas suas conexões sociais como familiares e demais pessoas que aquele indivíduo queira se relacionar. O serviço de rede social representa uma abstração que permite aos indivíduos gerenciarem suas relações sociais. A mídia social, por sua vez, representa o meio através do qual algum indivíduo pode trocar dados baseados nas conexões de sua rede social.

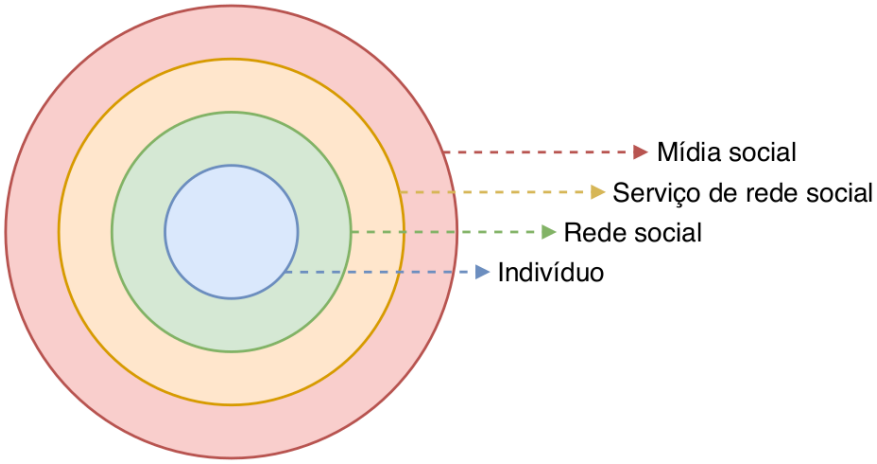


Figura 2.4: Serviços de Rede Social [23]

O Módulo da plataforma UbiCare desenvolvido com base em redes sociais, denominado UbiCare Social, permite o encaminhamento de notificações relacionadas ao monitoramento contínuo de sinais vitais de pacientes, para profissionais de saúde e familiares utilizando redes sociais. Este módulo realiza a integração destas redes sociais no contexto de um SMRP permitindo que os usuários demonstrem interesse em receber notificações por meio da manutenção de relacionamentos[54, 55, 23].

Esse Módulo utiliza como base a abstração de serviço de rede social aplicando assim as redes sociais para gerenciar os relacionamentos entre as pessoas. Essa dissertação amplia esse grau de abstração para o da mídia social, concedendo permissão de acesso aos dados com base nas conexões entre as pessoas, permitindo assim que um paciente ao se relacionar com um profissional de saúde em uma rede social específica para este fim, esteja concedendo acesso aos seus dados, enquanto o profissional de saúde demonstraria interesse em receber esses dados por meio também desse relacionamento.

A Figura 2.5 mostra a arquitetura do UbiCare Social em conjunto com o fluxo que vai desde a coleta de dados pelos sensores, sua transmissão para o UbiCare até a disseminação de notificações aos interessados nela. Os componentes dessa arquitetura são:

- EN-API - Emissor de Notificações: contém os recursos que permitem a entrega de notificações aos diferentes usuários do sistema.
- GR-API - Gerenciador de Relacionamentos: disponibiliza recursos que permitem a manutenção dos relacionamentos entre os usuários do sistema.

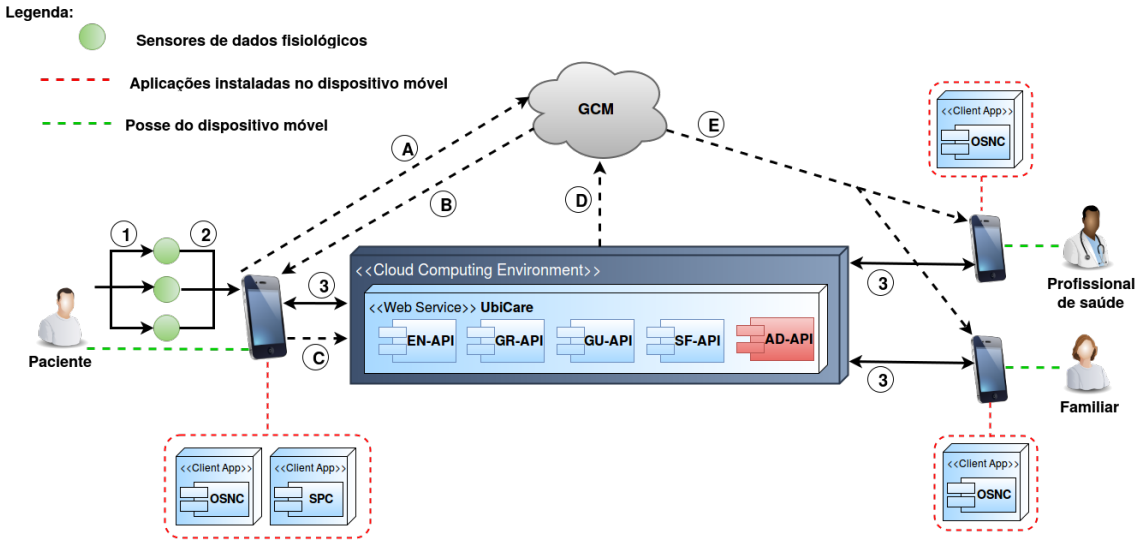
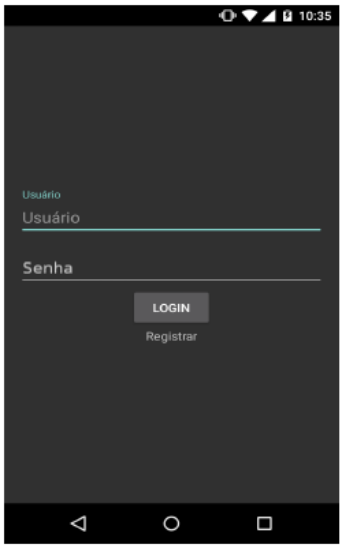


Figura 2.5: Arquitetura do UbiCare Social [54, 55, 23]

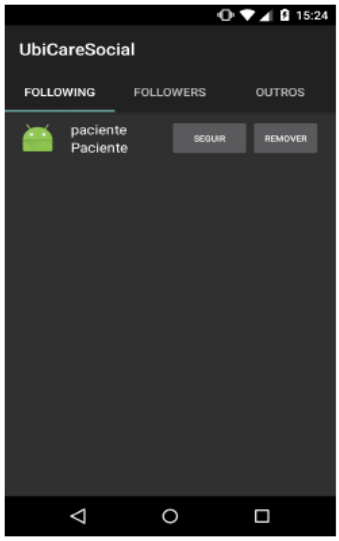
- GU-API - Gerenciador de Usuários: disponibiliza recursos que permitem a autenticação e o controle de usuários.
- SF-API - Sensores Fisiológicos: disponibiliza os recursos que viabilizam o registro de dados fisiológicos do paciente, tais como pressão arterial e peso.
- AD-API - Analisador de Dados: deveria ser responsável por realizar a análise dos dados (ainda não desenvolvido).
- OSNC - Cliente de Rede Social: aplicativo para a plataforma Android que permite que os pacientes, os profissionais de saúde e os familiares gerenciem seus relacionamentos, por meio das funcionalidades do serviço GR-API, e recebam as notificações referentes a essas relações.
- SPC: responsável pela coleta de dados de sensores fisiológicos.
- GCM - Google Cloud Messaging: responsável por auxiliar no encaminhamento das notificações.

A Figura 2.6 apresenta o comportamento do protótipo do aplicativo da rede social, denominado UbiCareSocial [23], na visão de um usuário do tipo interessado, como, por exemplo, um Profissional de Saúde. Nessa visão, o aplicativo apresenta a tela de login; uma aba onde estão localizados os usuários com os quais o usuário logado possui um relacionamento na rede social; uma aba onde estão localizados os usuários que estão interessados no usuário logado; e uma aba com os usuários aos quais o usuário logado não possui um relacionamento.

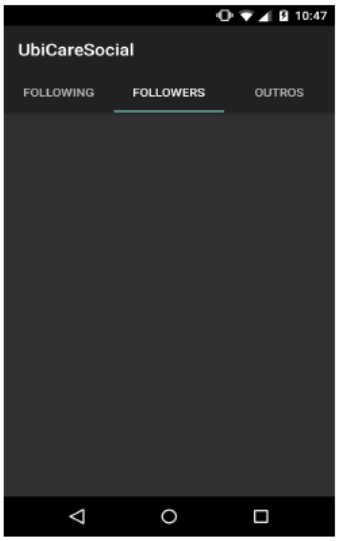
Este trabalho modifica os componentes de GR-API e de GU-API alterando o *back-end* do aplicativo de rede social com o intuito de investigar a viabilidade da aplicação



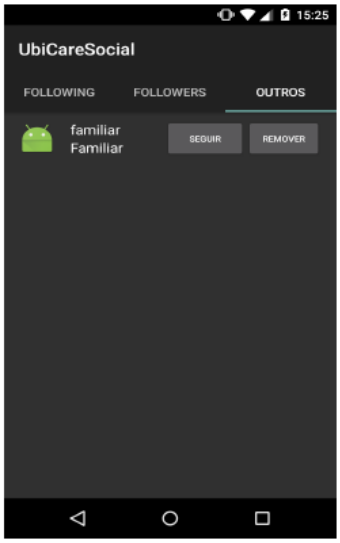
(a) *Tela de Login*



(b) *Usuários de interesse do usuário logado*



(c) *Usuários interessados no usuário logado*



(d) *Demais usuários presentes no sistema*

Figura 2.6: Comportamento do aplicativo UbiCareSocial, considerando a visão de um usuário do tipo interessado, como, por exemplo, um Profissional de Saúde [54, 55, 23]

de contratos inteligentes para a gerências de usuários e relacionamentos nessa rede social, como apresentado com mais detalhes nos capítulos 3 e 4.

2.1.3 Princípios de Segurança da Informação

Os princípios de segurança da informação, conhecidos por CID (confidencialidade, integridade e disponibilidade), são definidos pela ISO 27000 [36].

O princípio da confidencialidade permite que apenas usuários autorizados aces-

sem, usem ou copiem informações. Em outras palavras, as informações não são disponibilizadas ou divulgadas a indivíduos, entidades ou processos não autorizados [36]. Se a informação protegida for acessada por um usuário não autorizado, ocorre o que se denomina quebra de confidencialidade.

A privacidade é tratada no contexto da confidencialidade, no entanto, devido aos avanços tecnológicos e à grande demanda de dados privados atualmente existentes, a privacidade dos dados de uma pessoa se tornou uma questão crítica e tem sido tratada como uma área específica da segurança da informação.

Privacidade é a propriedade das informações pertencerem a uma pessoa [36]. Uma das formas de garantir a privacidade é permitindo ao dono da informação decidir como, quando e por quem seus dados serão manipulados. Além disso, deve ser fornecida ao dono detalhes sobre quais informações estão sendo coletadas e como esses dados estão sendo tratados e armazenados. Uma informação pode ser privada e não confidencial, mas quem decide a confidencialidade dessa informação é a pessoa a qual ela pertence.

Integridade é a propriedade de exatidão e completude de uma informação [36]. É o princípio que protege os dados e não permite que sejam corrompidos, alterados, excluídos ou recriados sem a devida autorização.

Disponibilidade, por sua vez, é a propriedade de ser acessível e utilizável sob demanda por uma entidade autorizada [36]. É esse o princípio que garante que as informações estejam sempre disponíveis para o usuário final quando ele precisar delas.

Além desses princípios, outras duas propriedades importantes para a segurança da informação são a autenticidade e o não-repúdio. Autenticidade garante que uma entidade é realmente o que afirma ser [36], e o não-repúdio caracteriza-se pela capacidade de provar a ocorrência de um evento ou de uma ação, por meio da identificação de suas entidades de origem [36].

Nessa dissertação, o foco está em garantir a privacidade dos dados do paciente, dando a ele o poder de decisão sobre a confidencialidade dos mesmos, e a escolha de quem conceder ou não acesso. A abordagem centrada na pessoa fundamenta que os dados de saúde são da pessoa, logo, isso justifica que ela tenha o controle sobre eles. Além disso, uma das abordagens para se garantir privacidade é citada pela lei geral de proteção aos dados pessoais (LGPD) [3] que permite o próprio usuário decida quando compartilhar ou não um dado. Importante ressaltar que o ideal seria conseguir garantir todos os princípios básicos citados, no entanto à complexidade que isso iria agregar, escolhemos reduzir o escopo do nosso trabalho e deixar os demais princípios como trabalhos futuros.

2.1.4 Blockchain

Blockchain em português significa cadeia de blocos. O termo surgiu em 2008 quando um artigo foi transmitido para uma lista de e-mails divulgando uma criptomoeda denominada *Bitcoin*[50]. Esse artigo apresenta uma explicação sobre o funcionamento dessa tecnologia, responsável por tornar segura a realização de transações na internet sem depender ou confiar em um terceiro, como, por exemplo, um banco.

O *blockchain* do *Bitcoin* é um livro contábil (ledger) distribuído público do qual toda a rede *Bitcoin* depende. Este livro contábil é onde são registradas as transações, de forma na ordem em que ocorreram, formando um histórico de todas as transações que resultaram no estado atual da rede *blockchain*. Outra característica de *blockchain* é que cada nó da rede possui uma cópia desse livro contábil distribuído, eliminando assim a possibilidade de um único ponto de falha[50, 32].

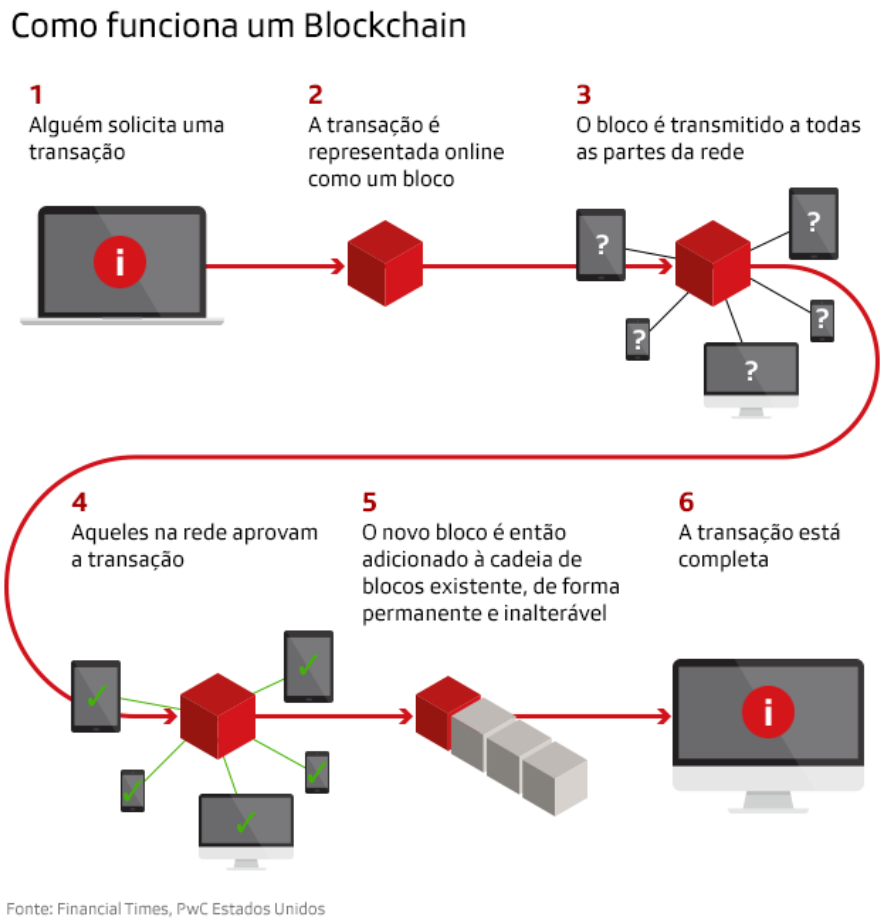


Figura 2.7: Funcionamento de um *blockchain* em alto nível de abstração [7]

A Figura 2.7 apresenta em um alto nível de abstração o funcionamento de um *blockchain*, desde a solicitação de uma transação até o seu término. No passo 1 alguém solicita uma transação; no passo 2, a transação é representada como um bloco; no passo 3,

esse bloco é disseminado por todos os nós da rede; no passo 4, os nós que compõem a rede aprovam a transação; no passo 5, o novo bloco é adicionado na cadeia de blocos de forma permanente e imutável; e no passo 6, a transação está completa. Em um *blockchain* como o do *Bitcoin*, o fluxo de uma transação é um pouco diferente do apresentado pois requer mais passos, sendo portanto mais complexo. Nele, um bloco é um conjunto de transações criado através do processo de mineração executado por nós denominados mineradores que competem entre si por meio de um mecanismo de consenso para determinar quais nós terão suas transações publicadas no *blockchain*. Após a criação de um novo bloco, ele é disseminado para todos os nós da rede, os quais devem executar a verificação do bloco para validá-lo. Caso mais da metade da rede aprove o bloco como válido, o bloco é validado e adicionado à cadeia de blocos já existente[7].

O *blockchain* utiliza uma estrutura de dados similar a de uma lista encadeada para a construção da cadeia de blocos, com cada bloco contendo um hash do bloco anterior em seu cabeçalho, como apresentado na Figura 2.8, formando assim uma cadeia de blocos imutável que tem início no bloco gênesis (o primeiro bloco da cadeia) até o mais recentemente adicionado. Sempre que um novo bloco de transações é criado torna-se necessário atualizar todos os outros nós da rede com aquele novo bloco de transações. Esse processo é realizado por meio de um algoritmo de consenso que, no caso do *Bitcoin* é o algoritmo de prova de trabalho (PoW) [50, 32].

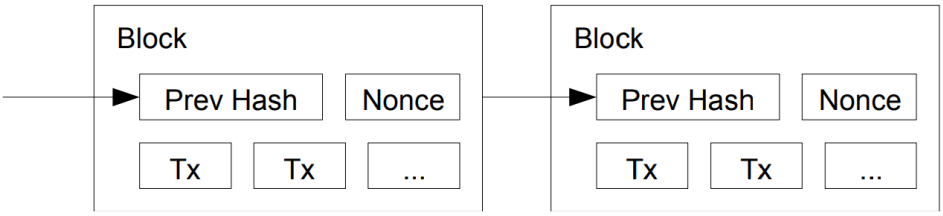


Figura 2.8: *Blockchain* [50]

O algoritmo PoW (Proof of Work), inicialmente proposto por Adam Back [12], tem como ideia principal desmotivar ataques cibernéticos. Para atingir este objetivo, o usuário deve provar por meio de uma prova de trabalho que gastou um certo tempo para encontrar uma resposta que satisfaça a um requisito. A tarefa de encontrar a resposta é baseada em dois princípios: a prova de trabalho tem que ser difícil e trabalhosa, mas não impossível; e a verificação da prova de trabalho dessa prova deve ser rápida e fácil de ser realizada. Os nós responsáveis por realizar essa prova de trabalho, com o intuito de validar aquele conjunto de transações e provar que foi o primeiro a resolver esse problema computacionalmente difícil, são denominados mineradores (*miners*), os quais competem para realizar o trabalho de mineração de um novo bloco [12].

O minerador que conseguir realizar o trabalho de mineração mais rápido ganha

um prêmio, em geral, uma certa quantidade de criptomoeda como incentivo para continuar minerando. O minerador é responsável também por informar aos outros nós que foi criado um novo bloco com determinado conjunto de transações válidas e adicionar o bloco à cadeia. O consenso ocorre nesse momento por meio da escolha da cadeia mais longa como sendo a verdadeira [12].

Em um nível de abstração mais alto, a tecnologia *blockchain* utiliza mecanismos conhecidos da ciência da computação como listas encadeadas e redes distribuídas; reúne primitivas criptográficas como *hashing*, assinaturas digitais, chaves públicas/privadas; e combina conceitos financeiros tais como *ledgers*.

A tecnologia *blockchain* foi escolhida para ser utilizada nesse trabalho devido ao seu potencial de garantir alguns princípios de segurança apenas por utilizá-la e também pela característica de manter um histórico imutável dos dados.

Blockchain por padrão já fornece as propriedades de imutabilidade dos dados, integridade, transparência, não-repúdio e direitos iguais para os seus participantes [69]. Porém, a garantia ou não de cada propriedade pode variar conforme a configuração do blockchain escolhida. O artigo [69] apresenta mais detalhes sobre isso.

Contratos Inteligentes

A tecnologia *blockchain* também utiliza o conceito de contratos inteligentes, que são implementados na forma de *scripts* escritos em uma determinada linguagem de programação e armazenados na rede *blockchain*. As regras descritas no contrato inteligente funcionam de forma similar a uma lei, pois uma rede *blockchain* é praticamente imutável e os contratos armazenados nela herdam essa característica. Em outras palavras, o que for acordado em um contrato é exatamente o que será executado pela rede [50, 32, 70].

Em [70] um contrato inteligente é definido como uma coleção de código e de dados que são implantados em uma rede *blockchain*, por exemplo, *Ethereum*[1], *Hyperledger Fabric*[2], dentre outros. A cada transação enviada para o *blockchain*, dados podem ser enviados para os métodos públicos oferecidos pelo contrato inteligente. O contrato executa o método apropriado com os dados do usuário fornecidos para executar um serviço. O código, sendo implantado no *blockchain*, é imutável e, portanto, utilizado como um terceiro de confiança para transações, como, por exemplo, uma transação financeira. Nesse caso, um contrato inteligente pode executar cálculos, armazenar informações e enviar automaticamente fundos para outras contas. Os contratos inteligentes podem ser usados como uma forma de personalizar o funcionamento da rede *blockchain*.

[1] https://www.ethereum.org
[2] https://hyperledger-fabric.readthedocs.io/en/release-1.4/

O Código abaixo mostra um exemplo de um contrato inteligente escrito na linguagem Solidity e que pode ser implantado na *Ethereum*.

```
pragma solidity ^0.4.19;

contract ZombieFactory {

event NewZombie(uint zombieId, string name, uint dna);

uint dnaDigits = 16;
uint dnaModulus = 10 ** dnaDigits;

struct Zombie {
string name;
uint dna;
}

Zombie[] public zombies;

mapping (uint => address) public zombieToOwner;
mapping (address => uint) ownerZombieCount;

function _createZombie(string _name, uint _dna) internal {
uint id = zombies.push(Zombie(_name, _dna)) - 1;
zombieToOwner[id] = msg.sender;
ownerZombieCount[msg.sender]++;
NewZombie(id, _name, _dna);
}

function _generateRandomDna(string _str) private view returns (uint) {
uint rand = uint(keccak256(_str));
return rand % dnaModulus;
}

function createRandomZombie(string _name) public {
require(ownerZombieCount[msg.sender] == 0);
uint randDna = _generateRandomDna(_name);
randDna = randDna - randDna % 100;
_createZombie(_name, randDna);
}

}
```

Listing 2.1: Exemplo de um Contrato Inteligente em Solidity - Adaptado de [51]

Esse contrato tem o intuito de servir como uma fábrica de Zombies que servem como base para um jogo chamado CryptoZombies o qual é desenvolvido em um tutorial

interativo para aprender a desenvolver contratos inteligentes em Solidity por meio da construção do seu próprio jogo cripto-colecionável[51].

Além disso, esse contrato define a estrutura de dados que vai representar um Zombie, as funções que vão criar um Novo Zombie, para gerar um DNA aleatório, para criar um Zombie aleatório, também define variáveis para definir quem é o dono do Zombie e quantos Zombie um mesmo dono possui, dentre outras coisas.

Os contratos inteligentes são utilizados nessa dissertação para identificar usuários, rastrear os dados e conceder acesso aos dados do paciente conforme descrito no Capítulo 3 e 4.

2.1.5 Plataformas de Desenvolvimento de Contratos Inteligentes

Uma rede *blockchain* pode ser categorizada com base em seu modelo de permissão, que determina quem pode mantê-la (por exemplo, publicar blocos). Há duas categorias: sem permissão e com permissão [32]. Uma rede *blockchain* sem permissão ou pública é aquela onde qualquer pessoa pode publicar um novo bloco (por exemplo, *Bitcoin*, *Ethereum*). A rede *blockchain* com permissão ou privada é onde apenas usuários específicos podem publicar blocos (como a *Hyperledger Fabric*). Em termos simples, uma rede *blockchain* com permissão é como uma intranet corporativa controlada, enquanto uma rede *blockchain* sem permissão é como a Internet pública, onde qualquer pessoa pode participar. Redes de *blockchain* com permissões são frequentemente implantadas para um grupo de organizações e indivíduos, tipicamente referido como um consórcio.

Ethereum [67] foi a primeira plataforma *blockchain* para desenvolvimento de contratos inteligentes. Nela os contratos são escritos na linguagem de programação Solidity e implantados em um *blockchain* sem permissão. Ela usa como algoritmo de consenso a PoW em suas primeiras versões, e tanto os contratos inteligentes como os dados armazenados são, por padrão, públicos, e, portanto, sem privacidade alguma [67]. Além disso, essa plataforma cobra gás (unidade monetária obtida a partir da criptomoeda *ether*) pela execução de todas as suas transações, incluindo a execução de contratos inteligentes que varia de acordo com o grau de complexidade do contrato e tempo de execução.

Hyperledger Fabric [8] é também uma plataforma *blockchain* que permite o desenvolvimento de contratos inteligentes, que podem ser escritos nas linguagens de programação Go, Java ou Node JS, e são implantados em um *blockchain* com permissão modular. Um *blockchain* modular tem suas características representadas em módulos, como, por exemplo, os mecanismos de consenso que no caso do *Hyperledger Fabric* podem ser acoplados a ele. Tanto os contratos como os dados armazenados são também

públicos aos participantes da rede, porém causam um impacto menor quanto à privacidade por se tratar de um *blockchain* com permissão [8].

Um protocolo *blockchain* que traz a solução para essa falta de privacidade nos contratos inteligentes é o *Enigma*[3]. A rede *blockchain* do *Enigma* permite realizar computação em dados criptografados sem a necessidade de descriptografar, tornando possível a criação de contratos secretos que com base nessa propriedade não violam a privacidade dos dados que armazenam. Esses contratos secretos são desenvolvidos na linguagem de programação RUST [75]. Essa plataforma está em fase de desenvolvimento, sendo que a versão atual roda como um fork da plataforma Ethereum e por isso mantém a característica de cobrar gás pelas transações. Além disso, é fortemente acoplada a plataforma Ethereum.

A Tabela 2.1 apresenta uma comparação das plataformas de desenvolvimento de contratos inteligentes. Esta comparação foi desenvolvida com base na documentação apresentada por cada uma das plataformas em relação às principais características necessárias para a escolha de uma plataforma de desenvolvimento de contratos inteligentes [8, 67, 75].

2.2 Revisão da Literatura

Essa seção tem como objetivo apresentar a Revisão da Literatura que foi o ponto de partida para o desenvolvimento deste trabalho. Ela foi realizada para investigar o estado da arte sobre *Blockchain* aplicado a Registros Eletrônicos de Saúde (RES)[4], visando responder as seguintes questões de pesquisa:

- Como *Blockchain* está sendo aplicado a Registros Eletrônicos de Saúde?
- Quais os desafios da implementação de Registros Eletrônicos de Saúde podem ser satisfeitos pela aplicação da Tecnologia *Blockchain*?

2.2.1 Métodos

Visando encontrar as respostas para as nossas questões de pesquisa levantadas, foram realizadas buscas nas bases de dados da IEEE e da PubMed, na língua inglesa, até o mês de maio de 2018. As palavras chaves utilizadas nessas buscas foram:

- *BLOCKCHAIN* AND (EMR OR EHR OR PHR OR "HEALTH RECORD"OR "HEALTH RECORDS"OR "MEDICAL RECORD"OR "MEDICAL RECORDS")

[3] https://enigma.co

[4] Os registros eletrônicos de saúde foram utilizados nessa pesquisa com o intuito de buscar soluções que envolvessem armazenamento dos dados de saúde.

Tabela 2.1: Comparação das Plataformas de Desenvolvimento de Contratos Inteligentes

	Ethereum	**Hyperledger Fabric**	**Enigma**
Próposito	Plataforma de propósito geral desenvolvidada para B2C e aplicações generalizadas	Plataforma modular desenvolvida para B2B	Protocolo Implementado em um fork da plataforma ethereum, versão atual lançada é formetemente acoplada ao Ethereum
Confidenciali-dade	Transparente	Transações confidenciais	Contratos inteligentes Secretos
Modo de Participação dos Pares	Público/Privado e sem permissão	Privado e com permissão	Privado e com permissão
Mecanismo de Consenso	Prova de Trabalho(PoW)	Algoritmos de Consenso Plugável	-
Linguagem de Programação de Contratos	Solidity	Goland, Java, Node JS	Rust
Cryptomoeda	Ether	não tem, mas pode ser implementada através dos contratos	ENG
Gerencia Identidades, Possui Unidade Certificadora, Membership	Não possui	Realiza esse trabalho	Realiza esse trabalho
Dados Privados	Não possui	Através de SideDB onde apenas o hash dos dados vai para os demais nós	Contratos inteligentes Secretos

Essas palavras chaves foram escolhidas com base nas questões de pesquisa, pois elas estão relacionadas com *Blockchain* e Registros Eletrônicos de Saúde; então as palavras chaves foram escolhidas para encontrar os artigos que tratam de ambos. Por esse motivo foram escolhidas *Blockchain* e *Electronic Health Records*, mas devido ao estudo realizado sobre Registros Eletrônicos de Saúde, foram encontradas algumas ramificações deste termo como *Personal Health Record* e *Eletronic Medical Record*. Além dessas variações para se referir aos Registros Eletrônicos de Saúde também foram utilizadas como palavras chaves as siglas referentes a cada uma dessas ramificações (EHR, PHR e EMR).

Essas buscas nas Bases de Dados retornaram artigos de 2016 a 2018, não tendo sido utilizado nenhum tipo de filtro para buscar os artigos mais recentes. A Figura 2.9 apresenta o resultado obtido durante as buscas, mostrando o número de artigos que foi encontrado categorizando-os por ano de publicação. Como pode ser observado, a maioria

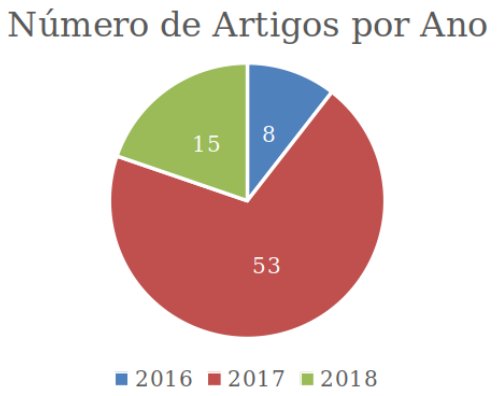


Figura 2.9: Número de artigos encontrados em cada ano

dos artigos é de 2017 e 2018. Isto nos leva a concluir que essa é uma área muito recente que está começando a ser explorada. O total de artigos encontrados foi de 76 sendo 70 deles oriundos da base de dados da IEEE e 6 da base de dados da PubMed. Para auxiliar na realização da Revisão da literatura foi utilizada a ferramenta StArt versão 3.3.

Etapa de Seleção - Na etapa de seleção foi realizada a leitura dos títulos, abstracts e palavras-chaves. Foram considerados os seguintes critérios de inclusão e exclusão para garantir que a busca iria abranger *Blockchain* e Registro Eletrônico de Saúde:

[INCLUSÃO] Publicações que apresentam as Palavras-Chaves;

[INCLUSÃO] Publicações que podem ser relevantes para a pesquisa (publicações que apesar do foco ser apenas uma das palavras chaves apresentam algum conteúdo que pode ser relevante para responder às perguntas de pesquisa, como taxonomias, métricas, etc.);

[EXCLUSÃO] Publicações que não estão relacionadas com *Blockchain*;

[EXCLUSÃO] Publicações que não estão Relacionadas a Registro Eletrônico de Saúde (Principalmente aquelas que não são da área da Saúde para a Financeira);

[EXCLUSÃO] Publicações cujo foco não é nem *Blockchain*, nem Registros Eletrônicos de Saúde.

Após esta Etapa de Seleção foram aceitos 24 artigos, para a etapa de extração.

Etapa de Extração - Nesta etapa foi realizada a leitura completa dos artigos aceitos na etapa anterior. Para esta fase foram atribuídos novos critérios de inclusão e exclusão:

[INCLUSÃO] Publicações que abordam *Blockchain* e Registros Eletrônicos de Saúde;

[INCLUSÃO] Publicações que abordam *Blockchain* aplicado a Dados de Saúde;

[EXCLUSÃO] Publicações cujo foco não é RES e nem *Blockchain* (ele aparece de forma secundária somente, sem dar detalhes de como poderia ser realizada de fato a

aplicação neste contexto).

O resultado desta etapa encontra-se na Tabela 2.2. Um total de 22 artigos foram aceitos e 2 foram rejeitados (estes estão riscados para demonstrar que foram descartados da revisão) com base nos critérios de inclusão e exclusão apresentados anteriormente. Esses artigos aceitos passaram para a etapa de análise dos dados. Após esta etapa foram qualificados conforme a relevância para este estudo.

Tabela 2.2: Resultado da Fase de Extração dos Dados

Título	**Ano**
MedRec: Using *blockchain* for Medical Data Access and Permission Management [11]	2016
Healthcare Data Gateways: Found Healthcare Intelligence on *blockchain* with Novel Privacy Risk Control [71]	2016
Blockchain technology in healthcare: The revolution starts here [48]	2016
~~A Secure System For Pervasive Social Network-Based Healthcare~~ [72]	~~2016~~
Blockchain: Solving the privacy and research availability tradeoff for EHR data: A new disruptive technology in health data management [46]	2017
Advanced block-chain architecture for e-health systems [43]	2017
Enabling Patient Control of Personal Electronic Health Records Through Distributed Ledger Technology [20]	2017
Evaluating Suitability of Applying Blockchain [44]	2017
Towards using *blockchain* technology for eHealth data access management [56]	2017
Integrating *blockchain* for data sharing and collaboration in mobile healthcare applications [42]	2017
Tamper-Resistant Mobile Health Using *blockchain* Technology [35]	2017
Introducing Blockchains for healthcare [6]	2017
A Taxonomy of Blockchain-Based Systems for Architecture Design [69]	2017
~~Secure lightweight context-driven data logging for bodyworn sensing devices~~ [60]	~~2017~~

On the Design of a *blockchain* Platform for Clinical Trial and Precision Medicine [59]	2017
OmniPHR: A distributed architecture model to integrate personal health records [57]	2017
MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain [68]	2017
An Introduction to the *blockchain* and Its Implications for Libraries and Medicine [34]	2017
Metrics for assessing Blockchain-based healthcare decentralized apps [73]	2017
Secure Attribute-Based Signature Scheme with Multiple Authorities for *blockchain* in Electronic Health Records Systems [33]	2018
Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy? [26]	2018
Blockchain: Challenges and applications [62]	2018
Leveraging *blockchain* for retraining deep learning architecture in patient-specific arrhythmia classification [37]	2018
Converging *blockchain* and next-generation artificial intelligence technologies to decentralize and accelerate biomedical research and healthcare [47]	2018

2.2.2 Resultados e Discussão

Os artigos aceitos na etapa de Análise dos Dados foram classificados em seis categorias [5] descritas abaixo:

1. Introdutórios ou Motivacionais: apresentam a tecnologia *Blockchain* citando Registros Eletrônicos de Saúde como uma das possíveis aplicações dela, ou aqueles que tratam do porque seria interessante aplicar a tecnologia *Blockchain* em Registros Eletrônicos de Saúde ou Dados de Saúde.
2. Métricas: propõem alguns requisitos básicos para a aplicação de *Blockchain* para Registros Eletrônicos de Saúde.
3. Taxonomia: apresentam uma taxonomia sobre as diversas configurações possíveis de *Blockchain* e discute as consequências de se utilizar cada uma.
4. Avaliação: propõem uma forma de avaliar se *Blockchain* é apto para um determinado domínio.

[5] A versão completa da análise, classificação dos artigos e dados extraídos deles encontra-se no link https://bit.ly/2qnsieK

5. Apresentam uma Proposta de aplicação de *Blockchain* para dados de saúde.
6. Propõem uma Solução Arquitetural com Protótipo ou Produto.

A Figura 2.10 apresenta o número de artigos encontrados de cada categoria descrita anteriormente.

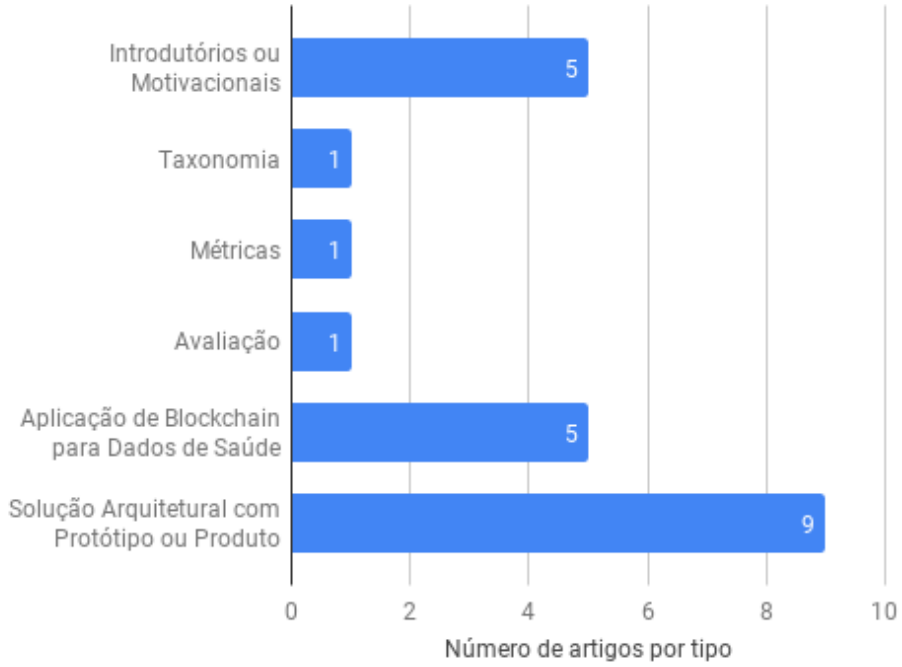


Figura 2.10: Número de artigos por Categoria

Como *Blockchain* está sendo aplicado em Registros Eletrônicos de Saúde? - Por ser uma área de pesquisa muito recente os resultados foram bastante esparsos. Os artigos que auxiliaram na resolução dessas questões de pesquisa foram os que se enquadraram nas categorias 5 e 6 mencionadas anteriormente.As diversas aplicações de *Blockchain* em Registros eletrônicos de Saúde as quais foram apresentadas nos artigos estão listadas a seguir:

A. Auxiliar na construção de um mercado de dados de saúde [47].
B. Armazenar uma cópia dos dados do PHR do paciente [57].
C. Auxiliar na análise de big data [59].
D. Garantir a integridade dos dados [11, 33, 35, 37, 42, 59].
E. Trocar dados de saúde [43, 56, 71].
F. Auxiliar no gerenciamento de dados do paciente [20].
G. Garantir a Privacidade [42, 43, 56].
H. Auxiliar na política de controle de acesso [20, 37, 42].
I. Aumentar a Interoperabilidade [11, 57].

J. Reduzir a fragmentação dos dados [11, 57].
K. Fornecer mais dados para pesquisas [11, 20].
L. Auxiliar na auditoria dos dados [35, 68].

O artigo [47] apresentar uma visão geral das tecnologias de inteligência artificial e de blockchain. Além de introduzir uma nova forma de avaliar custos de registros pessoais, também propõem um mercado de dados pessoais distribuído seguro e transparente que utiliza *blockchain* e *deep learning* para resolver alguns dos desafios encontrados pelos reguladores de dados de saúde e devolver o controle sobre os dados pessoais, incluindo registros médicos, aos indivíduos.

O artigo [57] apresenta um modelo de uma arquitetura para integrar PHRs de forma distribuída. O modelo é baseado em blockchain e utiliza uma rede overlay para distribuir os blocos de dados pelos nós. Algumas características do trabalho: não tratou sobre nenhuma prova de trabalho, não usa contratos inteligentes, o bloco guarda uma cópia dos dados do PHR, o paciente controla o sistema, é interoperável, fornece uma visão unificada do histórico do paciente, tem componentes responsáveis por inserir medidas de segurança e privacidade. Apesar de não abordar especificamente isto, o trabalho está mais focado na interoperabilidade.

O artigo [59] apresenta a ideia de uma plataforma blockchain, construída sobre a rede tradicional de *blockchain* para alavancar seus principais componentes para obter propriedades de transações confiáveis para testes clínicos e medicina de precisão e discute vários aspectos do projeto e fornece alguns *insights* sobre o assunto, requisitos tecnológicos e desafios. Os autores propõem uma arquitetura para essa plataforma *blockchain* com quatro novos componentes: um novo componente de paradigma de computação distribuída e paralela baseado em *blockchain* para desenvolver e estudar paralelamente a análise de *big data*; componente de gerenciamento de dados *blockchain* para garantir integridade de dados, integração de *big data* e integração da disparidade de dados médicos relacionados; componente de gerenciamento de identidades anônimas verificável para privacidade de identidade, incluindo IoT e acesso seguro a dados; componente de gerenciamento de compartilhamento de dados com confiança para permitir um ecossistema de dados médicos para pesquisa colaborativa. Além disso, os autores apresentam uma discussão sobre as metodologias e abordagens de ensaios clínicos e da medicina de precisão como dois casos de uso da plataforma blockchain.

O artigo [43] apresenta uma solução arquitetural de um *blockchain* multi-chain com permissão e uma autoridade certificadora; aplicada para dados de saúde, armazenando estes dados diretamente no blockchain, de modo a facilitar a troca desses dados de saúde. A autoridade certificadora é quem emite os certificados para permitir o acesso ao blockchain. Os pacientes através do uso de IDs e permissões controlam quais dados compartilhar e com quem. Os autores afirmam que essa solução fornece o registro dos

dados de saúde auditáveis, preservando a privacidade conforme as normas da HIPPA, e também garante a segurança do paciente.

O artigo [20] apresenta uma API *blockchain* para auxiliar no gerenciamento de dados do paciente. Essa API foi desenvolvida utilizando uma instância do Ethereum em uma rede privada, permitindo que os pacientes especifiquem quando e como seus registros são acessados para fins de pesquisa. As funcionalidades que cada tipo de usuário pode acessar estão descritas nos contratos inteligentes. O conteúdo do bloco é composto pelas propostas de acesso aos dados ou as respostas dos usuários a essas propostas, não sendo os dados de saúde propriamente ditos.

O artigo [71] apresenta uma proposta de um App baseado na tecnologia *blockchain* para permitir ao paciente controlar e compartilhar os seus dados de forma segura e sem violações de sua privacidade. Esse App é uma combinação de um banco de dados e um gateway proporcionando ao paciente as seguintes funcionalidades: gerenciar os dados de saúde pessoais armazenados no blockchain, avaliar todas as solicitações de dados alavancando o controle de acesso orientado a propósito. Além disso o App utiliza computação multipartidária segura para permitir que terceiros realizem processamento dos dados do paciente sem arriscar a privacidade do mesmo. O *blockchain* é utilizado como um banco de dados nesse caso, os dados são armazenados nele e tem um aplicativo para gerenciar os dados.

O artigo [42] propõe uma solução de compartilhamento de dados de saúde centrada no usuário, utilizando a tecnologia *blockchain* com permissão, para proteger a privacidade desses dados usando o esquema de canais e aprimorar o gerenciamento de identidade usando o serviço de membership suportado pelo blockchain. Com isso os autores buscam garantir a integridade dos dados de saúde, a proteção da privacidade em granularidade fina e uma política de controle de acesso descentralizada.

O artigo [11] apresentar o protótipo denominado MedRec que busca oferecer aos pacientes um registro abrangente, imutável e de fácil acesso a suas informações médicas em provedores de saúde. O intuito é garantir a integridade dos dados de saúde, aumentar a interoperabilidade, reduzir a fragmentação dos dados e fornecer mais dados para as pesquisas em saúde.

O artigo [68] apresentar uma solução, denominada MeDShare, para compartilhamento de dados entre provedores de serviços em nuvem, fornecendo controle de acesso, procedência e auditoria.

O artigo [33] propõe um esquema de assinatura baseado em atributos com múltiplas autoridades para garantir a validade dos EHRs encapsulados no blockchain, para garantir a privacidade dos pacientes, a integridade desses dados e a imutabilidade dos EHRs. Além disso eles provaram matematicamente que solução a proposta é resistente a ataque de conluio(onde pessoas maliciosas se reúnem para tentar tomar o controle do

blockchain).

O artigo [35] desenvolve e avalia um sistema mhealth inviolável usando a tecnologia blockchain. Para a construção desse sistema, optaram por utilizar um *blockchain* privado(uma instância do Hyperledger Fabric), armazenando nesse *blockchain* um hash do banco de dados, garantindo assim a integridade dos dados e facilitar a auditoria dos mesmos.

O artigo [56] realiza uma revisão da literatura para investigar a viabilidade de se propor uma arquitetura baseada em blockchain, ou um modelo, escalável e seguro para aplicações ehealth. Para construção dessa arquitetura os pesquisadores utilizaram um *blockchain* privado (uma instância privada do Ethereum), onde foram armazenadas as referências de onde os dados poderiam ser encontrados, facilitando a troca de dados sem perder a segurança e privacidade em futuras aplicações descentralizadas de ehealth.

O artigo [37] propõe uma solução com base em deep learnig, utilizando autoencoders para desenvolver uma técnica de classificação de arritmias de forma específica para um determinado paciente. A tecnologia *blockchain* é utilizada como gerente de controle de acesso, além disso, para armazenar e acessar com segurança os dados exigidos pela rede neural durante o seu treinamento em tempo real a partir de um armazenamento de dados externo.

Para entender melhor como a tecnologia *blockchain* foi aplicada a registros eletrônicos de saúde é necessário analisar os dados em duas perspectivas: a das configurações que foram utilizadas para a implementação de cada proposta; e para qual fim a tecnologia *blockchain* foi utilizada em cada proposta. Como para fazer essa análise são necessários detalhes, apenas o artigo [43] da categoria 5 foi incluído no comparativo realizado a seguir, os demais pertencem a categoria 6.

A Tabela 2.3 apresenta as configurações do *blockchain* de cada um dos artigos. Eles estão listados com base nas suas referências, tipo de *blockchain* utilizado, se é um *blockchain* com permissão ou sem permissão, se possui uma autoridade de verificação, se a proposta utiliza contratos inteligentes e qual o conteúdo armazenado no bloco do blockchain.

A Tabela 2.4 apresenta a comparação entre os artigos com base nos aspectos de segurança. Eles estão listados com base nas suas referências, nas aplicações da tecnologia *blockchain*, se apresenta aspectos de segurança, se garante privacidade e se estabelece políticas de controle de acesso.

Os resultados foram bastante esparsos, apesar disso foi possível observar que as principais aplicações de *Blockchain* a Registros Eletrônicos de Saúde foram para garantir a integridade dos dados, para trocar dados de saúde e para garantir a privacidade.

Tabela 2.3: Configurações do *blockchain* utilizado

Referência do Artigo	Tipo do *blockchain*	Permissão	Autoridade de verificação	Contratos Inteligentes	Conteúdo do Bloco
[43]	Multi-Chain Público	X	X	X	HL7, códigos LOINC, ICD, e-prescribe, ID do bloco, e assinaturas de quem reconhece o cuidado
[20]	Uma instância privada do Ethereum	-	-	X	propostas de acesso aos dados, respostas dos usuários a essas propostas
[71]	Privado	-	-	-	dados médicos que o paciente escolheu compartilhar
[42]	Privado, construído no *Hyperledger Fabric*	X	-	-	hash dos dados de saúde coletados de dispositivos vestíveis, políticas de controle de acesso, as solicitações de acesso e todas as atividades de acesso
[11]	Uma instância privada do Ethereum	-	-	X	permissões de propriedade e visualização de dados compartilhados por membros em uma rede privada, hash dos dados de saúde
[68]	Multi-Chain Público	-	-	X	informações relacionadas aos solicitantes e aos seus pedidos de acesso aos dados
[33]	Em Consórcio	-	Múltiplas	-	Dados de Saúde
[35]	Privado, construído no *Hyperledger Fabric*	-	-	X	um hash do estado atual do banco de dados
[56]	Uma instância privada do Ethereum	-	-	X	guarda no bloco o endereço (ponteiro/-referência) de onde podem ser encontrados os dados
[37]	Privado, construído no *Hyperledger Fabric*	X	-	X	hash do do banco de dados e políticas de controle de acesso

Tabela 2.4: Comparação entre os artigos com base nos aspectos de segurança apresentados

Referência do Artigo	Utilizado para que?	Segurança	Privacidade	Controle de Acesso
[43]	Trocar dados de saúde	X	X	-
[20]	Auxiliar no gerenciamento de dados do paciente, auxiliar na política de controle de acesso	X	X	X
[71]	Trocar dados de saúde	X	-	-
[42]	Garantir a integridade dos dados, garantir a privacidade, auxiliar na política de controle de acesso	-	X	X
[11]	Garantir a integridade dos dados, aumentar a interoperabilidade, reduzir a fragmentação dos dados, fornecer mais dados para pesquisas	X	-	X
[68]	Auxiliar na auditoria dos dados	X	X	X
[33]	Garantir a integridade dos dados	-	X	-
[35]	Garantir a integridade dos dados, auxiliar na auditoria dos dados	-	X	-
[56]	Trocar dados de saúde	X	X	-
[37]	Garantir a integridade dos dados, auxiliar na política de controle de acesso	X	X	X

Quais os desafios de implementação de Registros Eletrônicos de Saúde podem ser satisfeitos pela aplicação da Tecnologia *Blockchain*? - Os principais desafios de implementação de Registros Eletrônicos de Saúde que podem ser satisfeitos estão interligados com os princípios de segurança discutidos na 2.1.3, afinal, a tecnologia *blockchain* foi desenvolvida para realizar transações na internet de forma segura sem a necessidade de um terceiro confiável como um banco. Portanto, os desafios de registros eletrônicos de saúde que *blockchain* consegue satisfazer estão relacionados ao controle de acesso, à integridade dos dados, à confidencialidade, à privacidade, à necessidade de auditoria dos dados, à fragmentação dos dados, entre outros[26, 46, 48]. Porém, o que um *blockchain* é capaz de garantir ou não depende muito da configuração utilizada para construir a rede *blockchain* e de sua implementação em conjunto com os contratos inteligentes [69], ou seja, apenas o uso de *blockchain* não resolve todos os desafios relacionados à segurança dos dados de saúde.

O artigo [69] apresenta uma taxonomia para auxiliar nas considerações arquitetônicas sobre o desempenho e os atributos de qualidade dos vários tipos de blockchains, demonstrando o quanto a configuração escolhida para a tecnologia *blockchain* pode influenciar nas propriedades as quais ela auxilia a garantir.

2.3 Considerações Finais

Este capítulo começou com a fundamentação teórica, a qual é necessária para descrever o contexto onde este trabalho está inserido: os conceitos de Sistemas de Monitoramento Remoto de Pacientes, da abordagem centrada na pessoa onde os dados de saúde coletados pertencem ao próprio paciente, de redes sociais como um modo de gerenciar relacionamentos entre as entidades de um SMRP, dos princípios de segurança da informação, da tecnologia *blockchain*. Essa tecnologia trouxe junto com ela o conceito de contratos inteligentes que são "scripts"que uma vez implantados no *blockchain* se tornam lei, pois são executados da mesma forma que são escritos e também podem ser utilizados para o desenvolvimento de aplicativos descentralizados. A revisão da literatura foi apresentada logo após os conceitos fundamentais para este trabalho terem sido apresentados. Foi a Revisão da literatura que deu início a este trabalho, buscou investigar o estado da arte sobre a tecnologia *blockchain*, aplicada a registros eletrônicos de saúde (como sendo apenas dados de saúde), com o intuito de investigar como essa tecnologia estava sendo utilizada em conjunto com os dados de saúde.

CAPÍTULO 3

Arquitetura para Concessão de Permissão a Dados de Saúde Baseada em *Blockchain*

Este capítulo apresenta uma solução arquitetural que utiliza a tecnologia *blockchain* em conjunto com os contratos inteligentes, para garantir a privacidade dos dados de saúde de um paciente. Esses dados são coletados por um sistema de monitoramento remoto de pacientes (por exemplo, UbiCare), onde os pacientes manifestam o desejo de conceder acesso aos seus dados e os demais usuários o interesse de receber os dados desse paciente por meio de relacionamentos em uma rede social específica (por exemplo, UbiCare Social). Isso possibilita aos pacientes ter um maior controle dos seus dados de saúde, dando a eles a escolha de a quem conceder acesso a esses dados, seguindo a abordagem de medicina centrada no paciente.

3.1 Modelo de Domínio e Arquitetura

O modelo de domínio envolvendo sistema de monitoramento remoto de pacientes, serviço de rede social e contratos inteligentes é apresentado na Figura 3.1. O retângulo em cor vermelha representa (contratos inteligentes - *blockchain*) as entidades cujo funcionamento pode ser mapeado como contratos para permitir ao paciente conceder acesso a seus dados de saúde com a garantia que o acesso ocorrerá da forma como foi especificado. Essas entidades, mapeadas como contratos inteligentes, são as responsáveis por realizar as restrições do conteúdo, seja de notificações que estão sendo disseminadas ou mesmo de quais dados podem ser acessados e por quem. As restrições podem ser baseadas no papel de cada pessoa (Paciente, Familiar ou Profissional), nos relacionamentos entre as entidades e nas restrições adicionadas pelo próprio paciente no momento de conceder acesso aos seus dados. As entidades apresentadas nesse modelo de domínio são:

- Pessoa: indivíduo que faz parte da relação e que tem o poder de estabelecer restrições sobre ela.
- Relacionamento: representação de um relacionamento entre duas entidades do tipo Pessoa.

- Restrição: determina quais dados se deseja conceder acesso ou em quais há o interesse em acessar.
- Dispositivo: são dispositivos variados que servem como entrada de dados no sistema e pertencem a um determinado usuário; podem ser:

  Sensor: dispositivo que possui a capacidade de sensoriamento.

  Atuador: dispositivo com a capacidade de atuar no meio ao qual está inserido; sua atuação é delimitada pelos dados coletados através de sensores.

  E/S: dispositivo que permite que o usuário interaja com o sistema.
- Analisador: um subsistema responsável por realizar a análise dos dados recebidos por sensores.
- Notificador: responsável por enviar as notificações aos interessados.
- Evento: gerado através da detecção de alguma anomalia nos dados analisados pelo Analisador.
- Notificação: informação que será entregue aos interessados sobre a ocorrência de determinado evento.
- Plano de cuidados: informações referentes ao tratamento do paciente, desde recomendações médicas até o horário de administração de algum medicamento [55, 30]
- Contratos inteligentes: itens que serão mapeados para contratos inteligentes blockchain serão organizados em três contratos, o de identificação de usuário, o de rastreamento de dados e o de concessão de permissão.

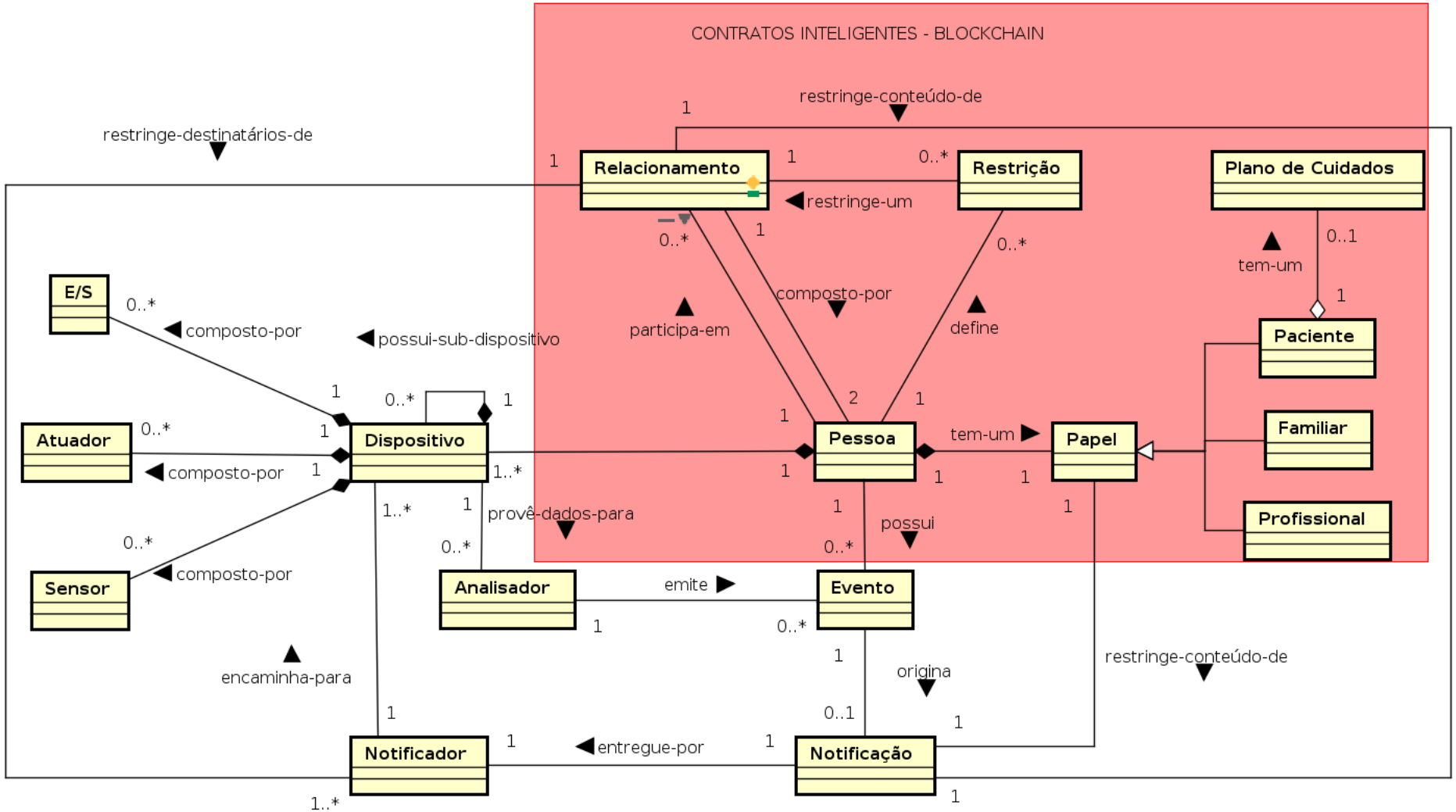


Figura 3.1: Modelo de Domínio com Contratos Inteligentes *Blockchain* - Adaptada de [23]

A Figura 3.2 apresenta o foco deste trabalho no contexto da arquitetura geral que integra o UbiCare e o UbiCare Social. Estamos interessados nos módulos de Gerência de Usuários (GU-API) e de Gerência de Relacionamentos(GR-API), utilizados para construir os nossos contratos inteligentes com o objetivo de preservar a privacidade dos dados do paciente.

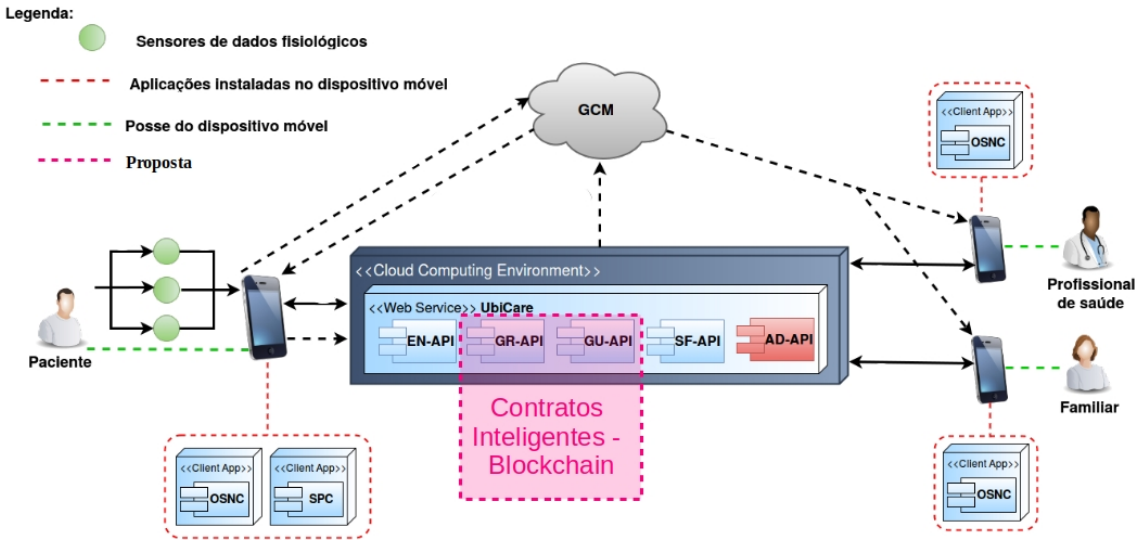


Figura 3.2: Arquitetura do UbiCare Social com a nossa proposta - Adaptada de [54, 55, 23]

A Figura 3.3 apresenta a proposta de solução arquitetural de concessão de permissão a dados de saúde baseada em *blockchain*. Os módulos de Gerência de Usuários e de Gerência de Relacionamentos possuem em sua estruturação contratos inteligentes construídos com o objetivo de preservar a privacidade do paciente que está sendo monitorado, e conceder permissão de acesso a seus dados de saúde a quem ele desejar com a confiança de que apenas quem ele autorizar terá o acesso.

O componente Contratos Inteligentes - *Blockchain* representa todo o conjunto de contratos necessários para realizar as funções de armazenamento dos dados, relacionamento entre as entidades, perfis de usuários, níveis de acesso de cada perfil, restrições determinadas pelo próprio paciente sobre os seus dados, entre outros. Um contrato determina um paciente e os seus respectivos dados monitorados. As atualizações são feitas por meio de chamadas aos métodos *set* definidos no contrato inteligente, os quais criam uma transação para armazenar as informações vindas dos sensores e do analisador de dados, atualizando o estado atual do *blockchain*. Por se tratar de um *blockchain*, as informações anteriores não são apagadas, sendo possível, portanto, manter um histórico completo dos dados nele armazenados.

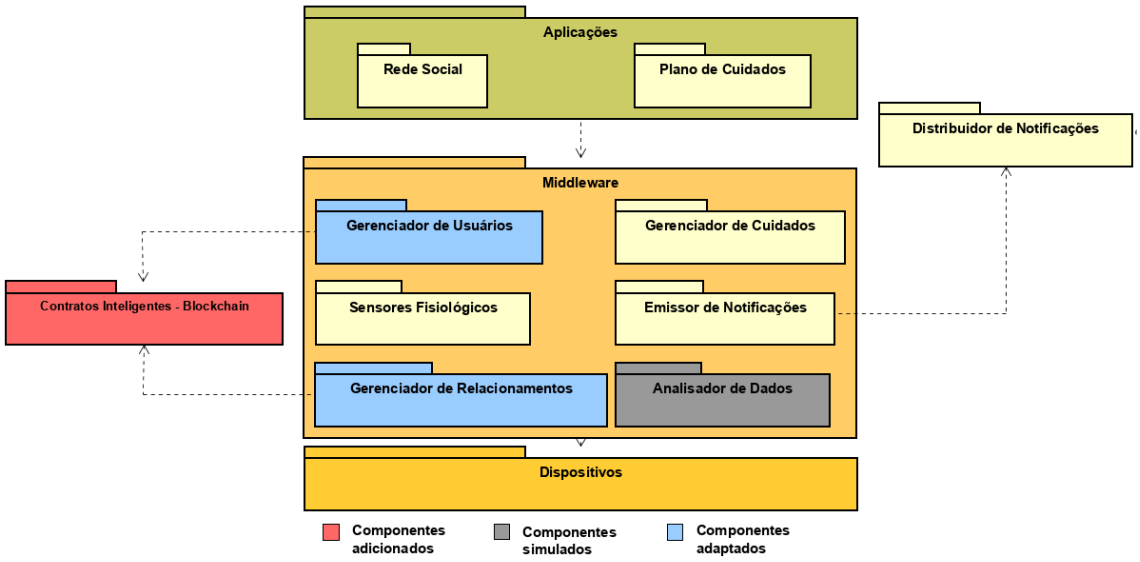


Figura 3.3: Solução Arquitetural de Concessão de Permissão de Acesso baseada em *Blockchain*.

A Figura 3.4 mostra as funcionalidades de cada um dos componentes do UbiCare/UbiCare Social e quais as funcionalidades dos módulos de Gerência de Relacionamentos (GR-API) e Gerência de Usuários (GU-API) foram adaptadas para os contratos inteligentes.

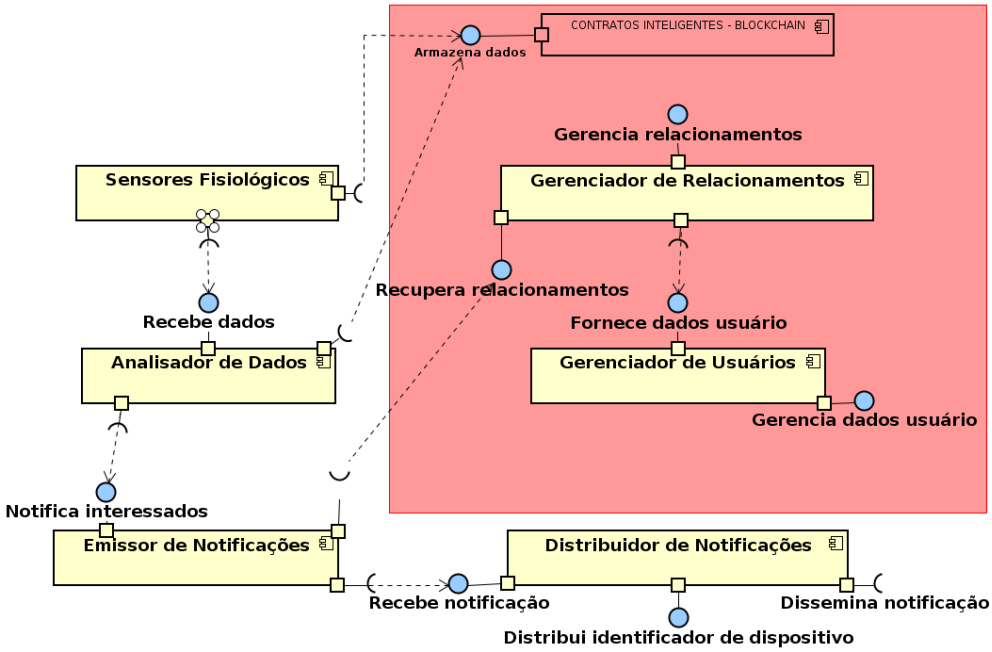


Figura 3.4: Componentes do Ubicare Social com *Blockchain* - Adaptada de [23]

Um contrato que determina quem é um paciente e quais dados estão sendo

monitorados é o mesmo utilizado para atualizar novas informações que chegam dos sensores e do analisador. As atualizações são feitas por meio de chamadas para os métodos *set* que criam uma transação para armazenar as informações mais recentes conforme chegam atualizando o estado atual do *blockchain*.

3.2 Contratos Inteligentes

Para realizar a concessão de permissão de acesso é necessário responder às seguintes questões: Quem é o dono dos dados? Quais são esses dados? Para quem o acesso será concedido? Quais são as restrições para esse acesso? Essas questões são respondidas nessa solução por meio de três tipos de contratos inteligentes: o contrato de identificação, o contrato de rastreamento dos dados e o contrato de concessão de permissão de acesso aos dados.

O contrato de identificação de usuário identifica o paciente, familiar, profissional de saúde ou responsável legal. A política de controle de acesso é implementada com base nesses papéis que os usuários podem assumir no sistema. Para cada papel um conjunto de informações é requisitado para criar e validar a identidade do usuário no sistema. Esse contrato é invocado pela rede social para realizar o cadastro de novos usuários no blockchain por meio de uma chamada ao método do contrato de identificação responsável por criar novas identidades na rede.

Usuários com o papel de paciente devem fornecer informações pessoais, como, por exemplo, nome, RG, CPF. Aqueles usuários com o papel de familiar fazem seu cadastro de dados pessoais e informações sobre a qual paciente está vinculado, comprovando esse vínculo com o anexo de um documento comprobatório no sistema. Essa relação de vínculo não permite a esse familiar o acesso aos dados de um paciente. Para efetivar esse acesso, é necessário que haja um contrato de concessão permissão de acesso aos dados entre o paciente e o familiar. O usuário com o papel de profissional de saúde, além de realizar o cadastro dos seus dados pessoais, também deve inserir seus dados de registros profissionais, como, por exemplo, a identificação junto ao respectivo conselho profissional. Por fim, outro papel possível para usuários é o de responsável legal, para os casos em que o paciente seja incapaz.

O contrato de rastreamento dos dados está vinculado a um usuário com o papel de paciente. Esse contrato é utilizado para manter um histórico das transações que alteram o estado global do ledger. Além disso, todas as operações de leitura e escrita relacionadas aos dados do paciente são chamadas pelos métodos deste contrato deixando assim um histórico completo e auditável sobre a manipulação desses dados, para então executar essas requisições alterando o estado global do ledger. Antes que as chamadas aos métodos desse contrato sejam processadas, consultas são realizadas para verificar se há

um contrato de concessão de permissão entre o paciente dono dos dados e o interessado. O contrato de rastreamento dos dados é invocado pela rede social durante o cadastro de um usuário com o papel paciente para armazenar os dados relacionados a ele, mas também é invocado sempre que um usuário desejar o acesso aos dados de um paciente ou quando for necessária a atualização dos dados vindos do SMRP (e.g., dados de saúde coletados por sensores).

O contrato de concessão de permissão de acesso aos dados está vinculado a um paciente e a um contrato de rastreamento dos dados. Nele, o paciente registra para qual usuário ele está concedendo permissões de acesso a seus dados, especificando as restrições que desejar, tanto em relação ao nível de permissão de acesso quanto a quais dados específicos. A rede social invoca esse contrato sempre que um novo relacionamento é criado entre os usuários, e realiza consultas para verificar se existe um contrato concedendo permissão antes de fornecer o acesso aos dados de um outro usuário.

A rede social deixa transparente para os seus usuários a utilização da tecnologia blockchain e a invocação dos contratos para o seu funcionamento. Os contratos são utilizados no sentido de garantir que as políticas de acesso e de funcionamento da rede sejam executadas da mesma forma que foram programadas. Contratos inteligentes e a tecnologia blockchain ficam no *back-end* da rede social auxiliando no seu funcionamento, no sentido de garantir a privacidade dos dados.

A Figura 3.5 apresenta o passo a passo para o funcionamento da arquitetura de concessão de permissão baseada em *blokchain*, desde o cadastro dos usuários até o estabelecimento de um relacionamento entre eles. No passo 1, o usuário do tipo paciente realiza seu cadastro na rede social que de modo transparente para ele também cria a sua identidade no blockchain por meio do contrato de identificação. No passo 2, o paciente seleciona o SMRP que deseja utilizar, gera uma autorização para os sensores e *wearables* aos quais deseja permitir que atualizem seus dados e armazena no contrato de rastreamento dos dados essas informações. No passo 3, os sensores e *wearables* de tempos em tempos atualizam os dados do paciente aos quais estão vinculados. No passo 4, um usuário do tipo profissional de saúde ou familiar realiza seu cadastro na rede social que de modo transparente também cria sua identidade no blockchain. No passo 5, o paciente estabelece um relacionamento na rede social com esse profissional de saúde ou familiar, estabelecendo as restrições de acesso para esse relacionamento, então a rede social armazena essas informações no blockchain através do contrato de concessão de permissão. No passo 6, o profissional de saúde ou familiar pode consultar os dados do paciente, a rede social realiza essa consulta através do contrato de rastreamento dos dados, que verifica as permissões no contrato de concessão de permissão e então envia os dados para o profissional de saúde ou familiar.

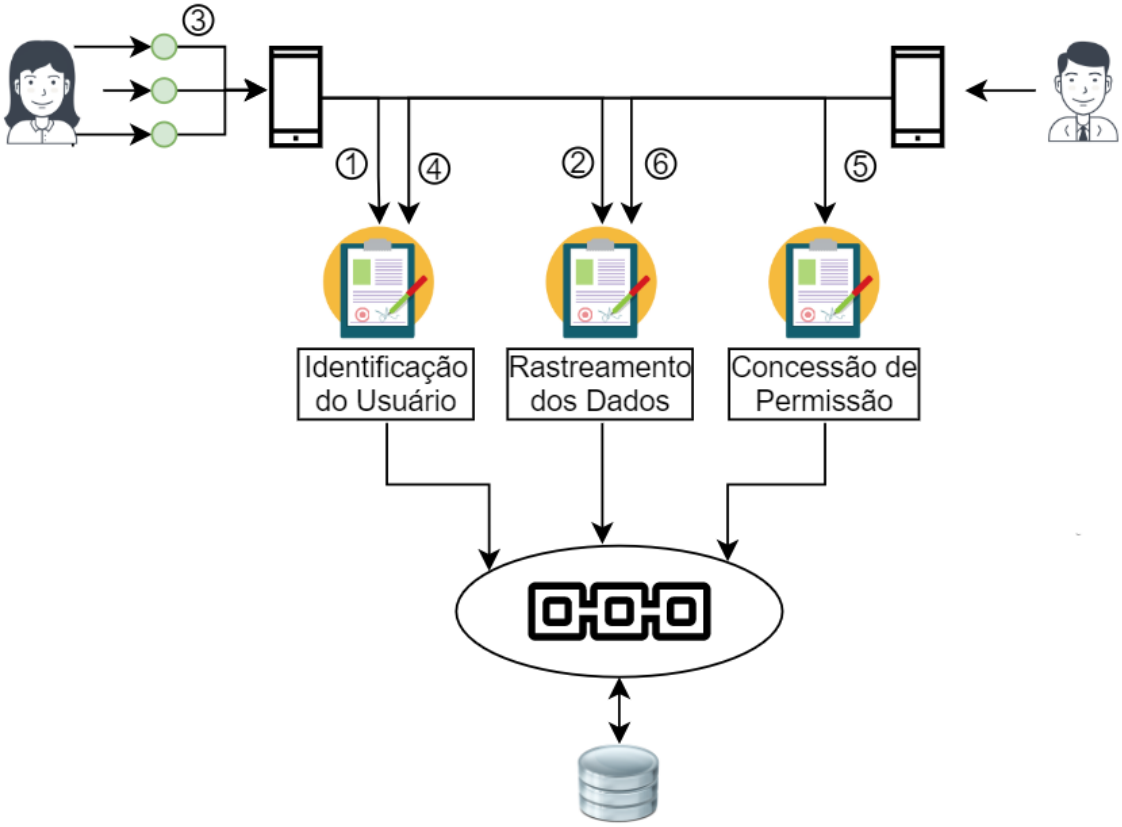


Figura 3.5: Funcionamento da Arquitetura de Concessão de Permissão baseada em *Blockchain*.

3.2.1 Diagramas de Atividades da UML

Essa seção tem o intuito de explicar a interação da aplicação com os contratos inteligentes, demonstrando em que momento o aplicativo da rede social invoca os contratos e o fluxo a ser seguido.

A Figura 3.6 apresenta a solução arquitetural de concessão de permissão de acesso a dados de saúde baseada em *Blockchain* por meio de uma visão dos componentes dessa solução. Essa figura é composta por quatro componentes principais: o aplicativo da rede social denominado UbiCare Social que foi proposto por [55]; uma interface para comunicação entre o componente de rede social e os contratos inteligentes que ainda não foi desenvolvida; os componentes de contratos inteligentes e da rede blockchain que foram desenvolvidos e apresentados nessa dissertação nesse Capítulo e no Capítulo 4. O contexto apresentado nessa figura foi a base para o desenvolvimento dos diagramas de atividades dos contratos inteligentes de identificação do usuário, rastreamento dos dados e concessão de permissão.

O contrato de identificação de usuários foi subdividido em dois, um para tratar o papel de paciente e outro para tratar o papel de interessado nos dados. O papel de paciente

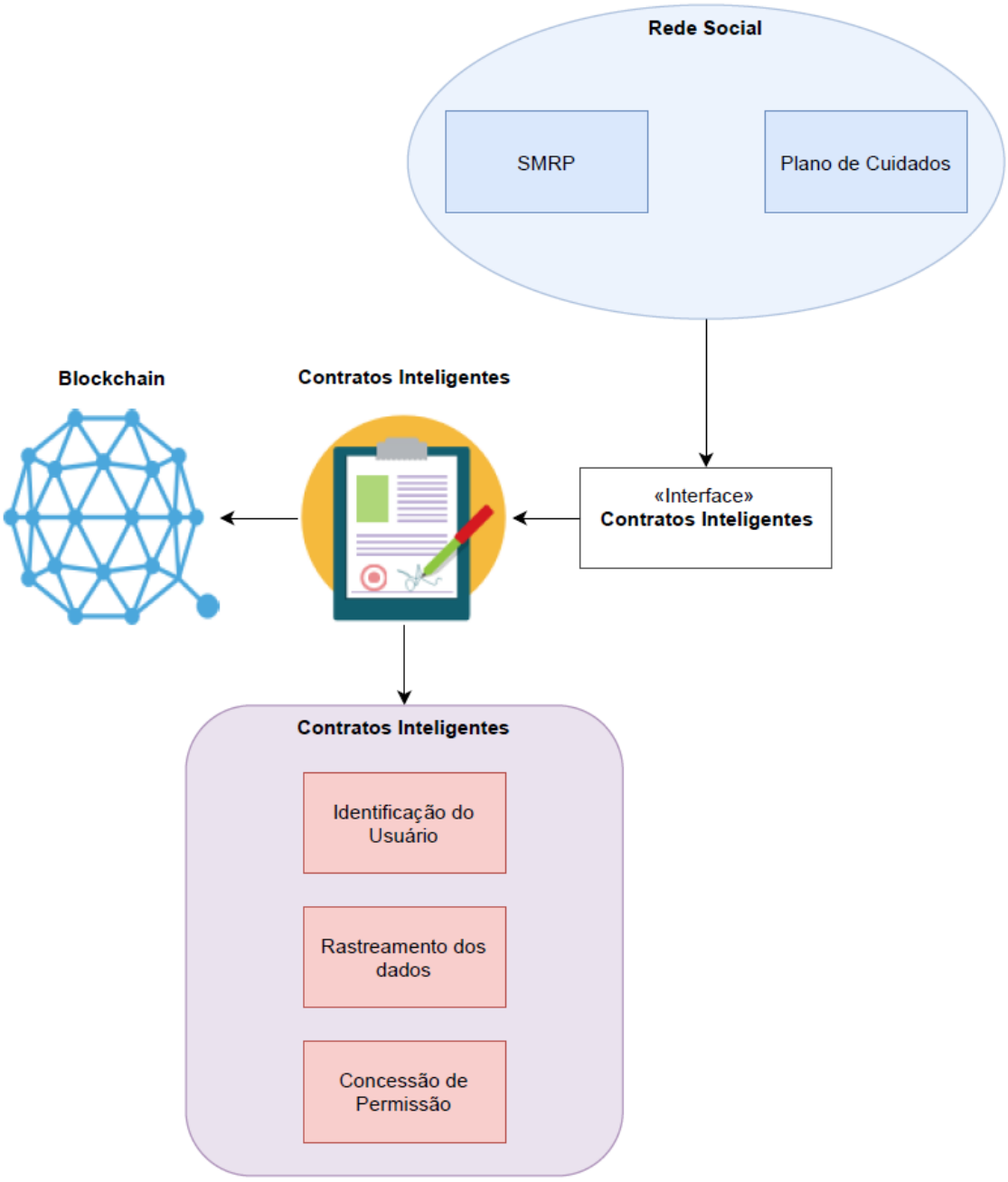


Figura 3.6: Componentes da Arquitetura de Concessão de Permissão de Acesso baseada em *Blockchain*.

é utilizado pela pessoa que utiliza um SMRP para coletar seus dados e tem o interesse de disseminar seus dados com seus familiares ou profissionais de saúde, que fazem parte da sua rede de cuidadores ou comunidade de interesse, utilizando para isso a rede social específica para esse fim. Os familiares e profissionais de saúde foram representados pela entidade interessado.

A Figura 3.7 apresenta o diagrama de atividades da UML do contrato de

identificação de um usuário, neste caso, tratando o caso de um usuário do tipo paciente. O paciente acessa o aplicativo da rede social com o intuito de realizar seu cadastro, então o aplicativo da rede social executa o fluxo de atividades apresentado na Figura 3.7. Iniciando com uma verificação para identificar se aquele paciente já está cadastrado, e, na sequência, se o paciente não é incapaz, se o paciente for capaz então ele poderá inserir seus dados na tela de cadastro do aplicativo da rede social. O aplicativo invoca o contrato de identificação de usuário para criar a nova identidade na rede blockchain. A rede social solicita os dados de acesso ao SMRP para o paciente, que os cadastra, para, por fim, a rede social invocar o contrato de rastreamento dos dados para cadastrar essas informações. O caso do usuário paciente ser incapaz não pode ser tratado via sistema pois dependeria da aprovação de outras pessoas, de uma unidade certificadora [1] ou de alguma regulamentação que ainda não temos.

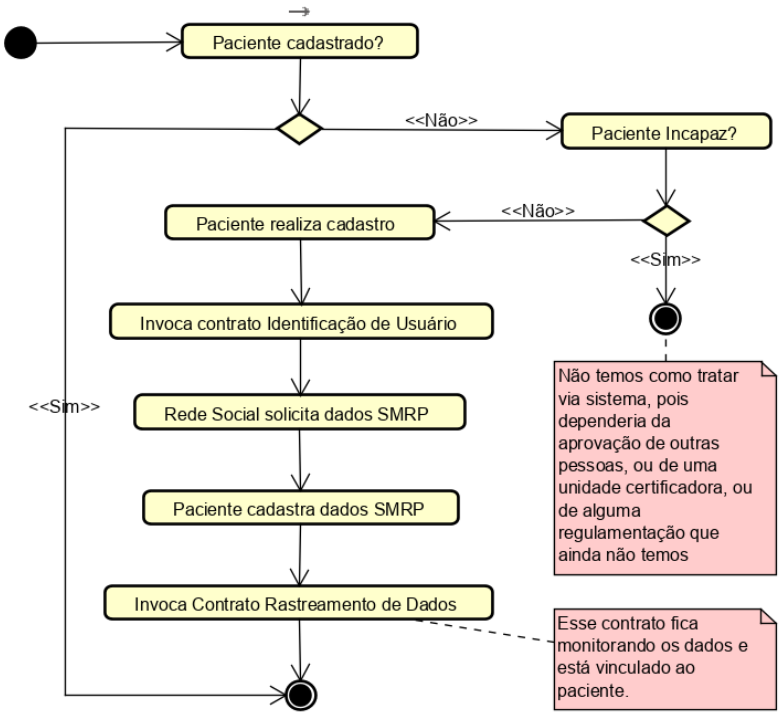


Figura 3.7: Diagrama de Atividades da UML - Identifica Paciente

A Figura 3.8 apresenta o diagrama de atividades da UML do contrato de identificação de um usuário, neste caso, tratando o caso usuário do tipo interessado (Profissional de Saúde, Familiar são os que se enquadram no papel de interessado). Nesse caso é realizado um cadastro simplificado, armazenando assim apenas os dados necessários para identificar aquela pessoa na rede social. O interessado acessa o aplicativo da rede social para efetuar seu cadastro nela, então o aplicativo invoca o contrato de identificação para realizar o cadastro desse novo interessado.

[1] Entidade responsável por criar e gerenciar credenciais na rede blockchain.

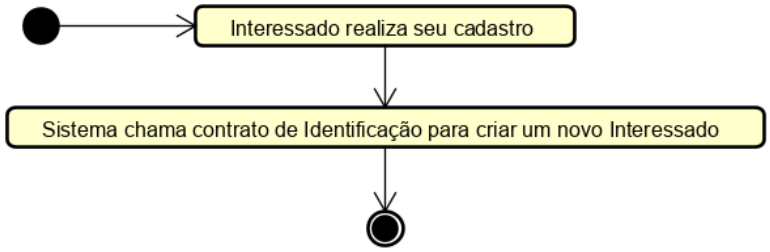


Figura 3.8: Diagrama de Atividades da UML - Identifica Interessado

A Figura 3.9 apresenta o diagrama de atividades da UML de concessão de permissão de acesso aos dados: paciente tem que selecionar para quem ele deseja conceder permissão de acesso, fazendo com que esse usuário participe da rede de interessados no paciente; paciente especifica as restrições para concessão desse acesso; interessado recebe uma notificação para ele aceitar ou não estabelecer esse relacionamento na rede social; rede social invoca o contrato de concessão de permissão para criar essa nova permissão de acesso na rede blockchain.

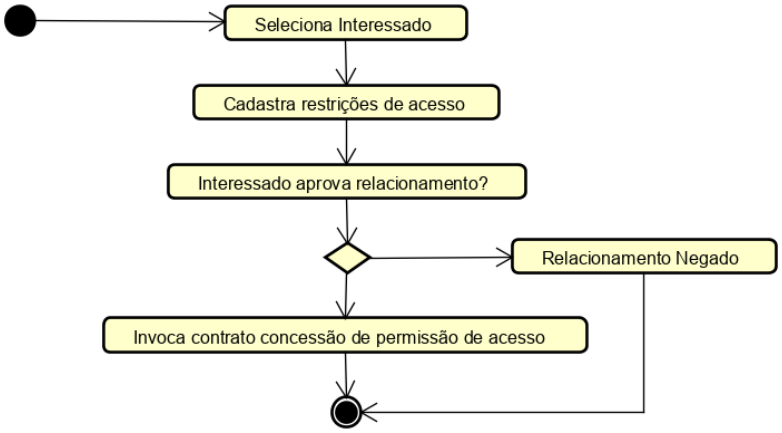


Figura 3.9: Diagrama de Atividades - Concede Permissão

A Figura 3.10 apresenta o diagrama de atividades da UML de acesso aos dados de um paciente: o usuário solicita acesso e a rede social realiza uma chamada ao contrato de Rastreamento dos Dados para obter acesso; o contrato realiza uma consulta para verificar se existe alguma instância do contrato de Concessão de permissão concedendo acesso para esse usuário; caso exista, fornece os dados e registra o acesso no blockchain para manter um histórico de manipulação dos dados de um paciente.

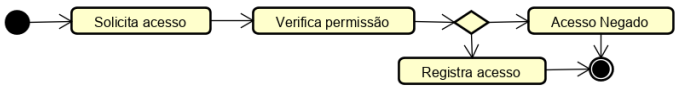


Figura 3.10: Diagrama de Atividades - Acessa Dados Paciente

3.2.2 Pseudocódigos dos Contratos Inteligentes

Essa seção tem o intuito de apresentar os pseudocódigos[2] dos contratos inteligentes. Os contratos de identificação, concessão de permissão e de rastreamento dos dados estão dispostos nessa seção com base nos métodos que cada um precisaria conter. Estes pseudocódigos estão levando em consideração que o aplicativo da rede social já utilizou uma ferramenta para gerar identidades para os seus usuários na plataforma blockchain escolhida, como, por exemplo, uma *wallet*[3] ou uma unidade certificadora[4] para gerar as chaves públicas e privadas, os certificados digitais e assinaturas necessárias para a identificação dos usuários em um blockchain. Neste trabalho foram utilizadas estruturas de dados para representar os usuários do tipo paciente e interessado (o usuário do tipo familiar e profissional de saúde se enquadram aqui).

Contrato de Identificação do Usuário

O contrato de identificação tem o intuito de criar um registro no ledger da identidade dos usuários da rede social. O contrato foi subdividido em dois métodos, sendo um para tratar da identificação do paciente e outro para tratar da identificação de seus familiares, profissionais de saúde ou responsáveis legais que nesse caso assumem papel de interessado nos dados produzidos pelo paciente.

```
1.  IdentificaPaciente(identidadePaciente, dados){
2.      se(ConsultaIdentidade(identidadePaciente) = sucesso)
3.          retorna "pessoa já possui identidade";
4.      paciente = MapeiaDadosParaStruct(dados);
5.      dados_criptografados = Encripta(paciente); //precisa no caso de contratos inteligentes normais
6.      se (ArmazenaIdentidadeBanco(identidadePaciente, dados_criptografados) = sucesso)
7.          retorna "dados inseridos com sucesso";
8.      senão
9.          retorna "erro ao inserir os dados de identidadePaciente";
10. }
```

Figura 3.11: Pseudocódigo que Identifica Paciente

A Figura 3.11 apresenta o método responsável por realizar a identificação do paciente. Este método espera como entrada a identidade do usuário na plataforma block-

[2]Os pseudocódigos podem ser encontrados na integra neste link: encurtador.com.br/bkzVZ

[3]Wallet ou carteira digital são utilizadas para guardar as assinaturas e certificados digitais de usuários blockchain e costumam fornecer uma forma de comunicação com o ledger, também possuem a função de criar um usuário na plataforma blockchain gerando assim esses certificados e assinaturas conforme o requisitado pela plataforma blockchain.

[4]Uma Unidade Certificadora é comum em blockchains com permissão, ela é a unidade responsável por criar e gerenciar os usuários que podem ter acesso ao ledger e também é quem gera os certificados e assinaturas necessárias para criar a identidade do usuário na rede blockchain

chain e os dados necessários para o cadastro dele. Na linha 2 é realizada uma verificação se aquele paciente já está cadastrado, se sim retorna uma mensagem avisando que a identidade na rede social já existe; na linha 4 é criada uma variável paciente (do tipo representado pela *struct* paciente) que espera receber os dados já convertidos para *struct*; na linha 5 é realizada a encriptação dos dados e armazenada na variável *dados_criptografados*. Essa etapa é necessária sempre que se tratar de contratos inteligentes aos quais não realizam por padrão o tratamento de seus dados como dados privados. No caso de contratos inteligentes secretos não seria necessário. Na linha 6 é armazenada a identidade do paciente no banco de dados, adicionando assim um registro no ledger, utilizando a identidade como chave e os dados criptografados como o valor, se essa função executar corretamente é exibida a mensagem de que os dados foram inseridos com sucesso; se não, é retornada uma mensagem de erro.

```
1.  IdentificaInteressado(identidadeinteressado, dados){
2.      se(ConsultaIdentidade(identidadeinteressado) = sucesso)
3.          retorna "pessoa já possui identidade";
4.      interessado = MapeiaDadosParaStruct(dados);
5.      dados_criptografados = Encripta(interessado);
6.      se (ArmazenaIdentidadeBanco(identidadeinteressado, dados_criptografados) = sucesso)
7.          retorna "dados inseridos com sucesso";
8.      senão
9.          retorna "erro ao inserir os dados de identidadeInteressado";
10. }
```

Figura 3.12: Pseudocódigo que Identifica Interessado

A Figura 3.12 tem o intuito de realizar a identificação do interessado de modo similar à descrita para identificar o paciente. A diferença entre elas está apenas no tipo de struct que representa os dados a serem cadastrados.

Contrato de Concessão de Permissão de Acesso aos Dados

O contrato de concessão de permissão de acesso aos dados foi subdividido em dois métodos: um para realizar a concessão e um para revogar a permissão. A Figura 3.13 apresenta o método responsável por realizar a concessão de permissão de acesso aos dados do paciente. Esse método espera como entrada a identificação do paciente e do interessado na plataforma blockchain em conjunto com uma string contendo a lista de dados permitidos. Na linha 2 é realizada uma busca da estrutura de dados do paciente, se ela não for encontrada é exibida uma mensagem de erro; na linha 5 é verificado se o usuário paciente que é o dono dos dados é quem está tentando realizar a concessão (pois é ele quem gerencia o próprios dados), isso é realizado por meio da função *VerificaPermissão()* que verifica se o usuário logado é o usuário dono dos dados, se não

```
1.  ConcederPermissão(identidadePaciente, identidadeInteressado, dadospermitidos){
2.      dados = BuscaPessoaBanco(identidadePaciente);
3.      se(dados = Null)
4.          retorna "paciente não existe";
5.      se(VerificaPermissão(identidadePaciente) != sucesso)
6.          retorna "sem permissão";
7.      novopaciente = Decripta(dados);
8.      idInteressado = BuscaInteressadoPaciente(novopaciente.interessados, identidadeInteressado);
9.      se(novopaciente.interessados[idInteressado] = identidadeInteressado)
10.         retorna "interessado já está na lista";
11.     DeletaEntradaPacienteBanco(identidadePaciente); //precisa por se tratar de um blockchain
12.     novopaciente.interessados = InsereNovoInteressado(novopaciente.interessados, identidadeInteressado,
        dadospermitidos);
13.     dados_criptografados = Encripta(novopaciente); //precisa no caso de contratos inteligentes normais
14.     se (ArmazenaIdentidadeBanco(identidadePaciente, dados_criptografados) = sucesso)
15.         retorna "Permissão concedida a identidadeInteressado com sucesso";
16.     senão
17.         retorna "erro conceder permissão de acesso aos dados para identidadeInteressado";
18. }
```

Figura 3.13: Pseudocódigo que Concede Permissão de Acesso os Dados do Paciente

for o usuário dono dos dados, é retornada uma mensagem de erro; na linha 7 é realizada a decriptação da estrutura de dados paciente e guardada na variável *novopaciente*; na linha 8 é realizada uma chamada para a função que busca um interessado na lista de interessados do paciente e retorna o *id* do interessado para a variável *idinteressado*; na linha 9 é verificado se o interessado para o qual se deseja conceder permissão já está na lista de interessados, se estiver é retornada uma mensagem de erro avisando que ele já está na lista; na linha 11 é atualizado o estado global estabelecendo a estrutura de dados do paciente como deletada. Esse passo é necessário porque um dado inserido em um blockchain não pode mais ser removido[5]; na linha 12 é realizada a chamada à função que adiciona um novo interessado na lista de interessados do paciente passando como parâmetro a identidade do paciente e do interessado em conjunto com uma string com a lista dos dados permitidos, essa função retorna a lista de interessados atualizada; na linha 13 é realizada a encriptação da estrutura de dados do paciente e na sequência a estrutura é armazenada no banco atualizando assim o estado global; por fim o método de concessão retorna que a permissão foi concedida com sucesso ou que houve um erro ao conceder a permissão.

A Figura 3.14 apresenta o método responsável por realizar a revogação da permissão de acesso aos dados. Funciona de forma similar ao método que realiza a concessão até a linha 8 com a diferença que essa função espera como entrada apenas

[5]Para fazer a atualização dos dados é necessário setar o estado daquela referência como removida do estado global para possibilitar armazenar outro valor para aquela mesma variável. Isso não deleta os dados do blockchain apenas altera o estado deles. Ao buscar um histórico, por exemplo, ainda constará os dados que foram setados como removidos e os dados atuais das variáveis.

```
1.  RevogarPermissão(identidadePaciente, identidadeInteressado){
2.      dados = BuscaPessoaBanco(identidadePaciente);
3.      se(dados = Null)
4.          retorna "paciente não existe";
5.      se(VerificaPermissão(identidadePaciente) != sucesso)
6.          retorna "sem permissão";
7.      novopaciente = Decripta(dados);
8.      idInteressado = BuscaInteressadoPaciente(novopaciente.interessados, identidadeInteressado);
9.      se(novopaciente.interessados[idInteressado] != identidadeInteressado)
10.         retorna "interessado não está na lista";
11.     while(novopaciente.interessados[idInteressado]!=Null){
12.         novopaciente.interessados[idInteressado] =  novopaciente.interessados[idInteressado+1];
13.         idInteressado++;
14.     }
15.     dados_criptografados = Encripta(novopaciente); //precisa no caso de contratos inteligentes normais
16.     DeletaEntradaPacienteBanco(identidadePaciente); //precisa por se tratar de um blockchain
17.     se (ArmazenaIdentidadeBanco(identidadePaciente, dados_criptografados) = sucesso)
18.         retorna "Permissão a identidadeInteressado revogada com sucesso";
19.     senão
20.         retorna "erro revogar permissão de acesso aos dados para identidadeInteressado";
21. }
```

Figura 3.14: Pseudocódigo que Revoga Permissão de Acesso os Dados do Paciente

a identidade do paciente e do interessado; na linha 9 é realizada a verificação se o interessado está na lista de interessados do paciente, se não estiver é retornada uma mensagem de erro; na linha 11 a lista de interessados é percorrida da posição onde o interessado foi encontrado até o fim da lista com o intuito de realizar a remoção daquele interessado da lista; na linha 15 de forma similar a apresentada na linha 13 do método de concessão, é realizada a encriptação dos dados, os quais são salvos no banco.

Contrato de Rastreamento dos Dados

```
1.  AtualizaDadosPaciente(identidadePaciente, dados){
2.      se(BuscaPessoaBanco(identidadePaciente) = Null)
3.          retorna "paciente não existe";
4.      se(VerificaPermissão(identidadePaciente) != sucesso)
5.          retorna "sem permissão";
6.      DeletaEntradaPacienteBanco(identidadePaciente); //precisa por se tratar de um blockchain
          novopaciente = MapeiaDadosParaStruct(dados);
7.      dados_criptografados = Encripta(novopaciente); //precisa no caso de contratos inteligentes normais
8.      se (ArmazenaIdentidadeBanco(identidadePaciente, dados_criptografados) = sucesso)
9.          retorna "dados atualizados com sucesso";
10.     senão
11.         retorna "erro ao atualizar os dados de identidadePaciente";
12. }
```

Figura 3.15: Pseudocódigo que Atualiza os Dados do Paciente

O contrato de rastreamento de dados foi subdividido em dois métodos, o de atualização dos dados do paciente e o de consulta do histórico dos dados do paciente. A Figura 3.15 apresenta o método responsável por atualizar os dados do paciente. Esse método espera como entrada a identidade do paciente e os dados. Na linha 2 é verificado se o paciente está cadastrado, se não, é retornada uma mensagem de erro; na linha 4 é verificado se o usuário paciente, dono dos dados, foi quem invocou o método de atualização dos dados[6]; na linha 6, os dados têm seu estado atualizado como removido; na linha 7, os dados são mapeados para a estrutura de dados paciente, e na sequência são encriptados e armazenados no banco, alterando assim o estado global do blockchain.

A Figura 3.16 apresenta o método responsável por realizar a busca do histórico dos dados daquele paciente. Na linha 3 é realizada uma verificação se o paciente está cadastrado, se não retorna uma mensagem de erro; na linha 5, a estrutura de dados paciente é decriptada; na linha 6, é realizada a verificação se o interessado em consultar o histórico é um usuário diferente do próprio paciente; se for diferente do paciente é realizada a consulta do histórico no banco de dados passando a identidade do paciente e uma string com a lista dos dados permitidos para aquele interessado, o retorno da função é salvo na variável *listadadospaciente*; se o interessado for o próprio paciente é realizada a consulta do histórico no banco de dados passando apenas a identidade do paciente; por fim, é retornado o histórico dos dados daquele paciente.

```
1.  HistoricoPaciente(identidadePaciente, identidadeInteressado){
2.      dados = BuscaPessoaBanco(identidadePaciente);
3.      se(dados==Null)
4.          retorna "paciente não existe";
5.      novopaciente = Decripta(dados);
6.      se(identidadePaciente != identidadeInteressado){
7.          idInteressado = BuscaInteressadoPaciente(novopaciente.interessados, identidadeInteressado);
8.          se(novopaciente.interessados[idInteressado]!=identidadeInteressado)
9.              retorna "sem permissão";
10.         dadospermitidos = novopaciente.interessados[idInteressado].dadospermitidos;
11.         listadadospaciente = BuscaHistoricoBanco(identidadePaciente, dadospermitidos);
12.     }
13.     senão
14.         listadadospaciente = BuscaHistoricoBanco(identidadePaciente);
15.     retorna "Os dados do histórico do identidadePaciente são listadadospaciente";
16. }
```

Figura 3.16: Pseudocódigo que Apresenta Histórico completo dos Dados do Paciente

[6]Os sensores de dados fisiológicos podem ser inseridos dentro da carteira digital do paciente e assim na hora de enviar os dados para o blockchain é passada a identidade do paciente que está sendo monitorado pelos sensores.

3.3 Considerações Finais

Este capítulo teve início com a apresentação do modelo de domínio envolvendo sistema de monitoramento remoto de pacientes (SMRP) e o serviço de rede social, com a descrição das entidades do modelo que podem ser mapeadas para contratos inteligentes. Também foi descrita a arquitetura geral que integra o UbiCare (o SMRP) e o UbiCare Social (a rede social), com o destaque para os módulos de gerência de usuários e gerência de relacionamentos, que são os módulos adaptados para se comunicar com os contratos inteligentes com o objetivo de preservar a privacidade dos dados do paciente. Foram apresentados também os contratos inteligentes de identificação do usuário, concessão de permissão de acesso aos dados e rastreamento dos dados em conjunto com os diagramas de atividades, demonstrando assim o funcionamento de cada contrato inteligente. Por fim, foram descritos os pseudocódigos dos contratos inteligentes.

CAPÍTULO 4

Implementação de Contratos Inteligentes

Esse capítulo tem o intuito de apresentar uma discussão sobre as abordagens de implementação do contratos inteligentes, a arquitetura da implementação do componente de Contratos Inteligentes - Blockchain na plataforma *Hyperledger Fabric*, uma descrição das configurações da rede e dos contratos inteligentes. Por fim, uma discussão sobre a implementação realizada.

4.1 Abordagens De Implementação dos Contratos Inteligentes

Para a implementação da proposta desse trabalho algumas decisões de implementação precisaram ser avaliadas como: Qual plataforma seria utilizada? Seria utilizado um blockchain público, privado ou seria construído um blockchain específico para resolver esse problema?

Devido à complexidade e ao tempo dedicado à construção de um blockchain específico para solução do nosso problema de pesquisa, no contexto de uma dissertação de mestrado, a escolha foi por avaliar o uso de plataformas de blockchain públicas e privadas.

Foi considerada a utilização de uma plataforma pública como o *Ethereum* [67], porém a própria definição dessa plataforma vai contra ao que está sendo proposto nessa dissertação, pois nessa plataforma todos os dados são públicos por padrão, e também cobra para efetuar suas transações e execução de contratos inteligentes. Utilizar a plataforma Ethereum implica, portanto, embarcar exclusivamente nos contratos inteligentes toda a complexidade de privacidade e controle de acesso.

A plataforma *Enigma* [75] seria a ideal para o desenvolvimento deste trabalho por tratar por padrão a questão do controle de acesso à rede dos dados privados e ter suporte a contratos secretos, ou seja, executados sem necessidade de de-criptografia. Essa plataforma, porém, está em desenvolvimento e também cobra para realizar suas transações na rede por ser fortemente acoplada ao *Ethereum*.

Diante disso e da análise realizada e apresentada na Tabela 2.1, a plataforma *Hyperledger Fabric* [8] foi a escolhida para o desenvolvimento deste trabalho. Essa escolha se deu ainda por ser uma plataforma mais estável em relação à plataforma *Enigma* que ainda está em desenvolvimento; ser com permissão, ao contrário da Ethereum; ter transações confidenciais e um serviço de gerenciamento de identidades; possuir uma documentação mais completa em relação à *Enigma*; e por não cobrar pelas transações, ao contrário das demais analisadas.

Após a escolha da plataforma foi necessário tomar mais duas decisões sobre a abordagem de implementação a ser utilizada. Onde os dados ficariam armazenados, dentro ou fora do blockchain?

Primeiro foi considerado que os dados ficassem armazenados fora do blockchain, sendo possível garantir sua integridade enviando um hash dos dados para o blockchain. Isso traria ganhos em termos de desempenho, pois não há a necessidade de se aguardar que a transação seja processada pelo blockchain. No entanto, outras propriedades não são garantidas, a não ser por meio dos contratos inteligentes se comunicando diretamente com o banco de dados através de *triggers*. Nesse caso, a questão de desempenho ainda demandaria uma avaliação. Devido à falta de documentações de como realizar isso na prática foi tomada a decisão de não seguir essa abordagem.

Diante disso, os dados nesse trabalho ficam armazenados dentro do blockchain formando assim um histórico imutável de todas as transações de acesso, atualização e inserção de novos dados. Esses dados teriam (I) integridade garantida através do protocolo de consenso; (II) disponibilidade garantida por todos os nós da rede possuírem uma cópia do blockchain; (III) autenticidade e não-repúdio garantidos por ser um blockchain com permissão. Cada usuário na rede tem uma identidade única, precisa ser aprovado para ter acesso a essa rede e qualquer ato seu na rede do *Hyperledger Fabric* é assinado com essa identidade.

4.2 Arquitetura do Componente de *Contratos Inteligentes - Blockchain* na Hyperledger Fabric

Para implementar contratos inteligentes na plataforma *Hyperledger Fabric*, é necessário definir as configurações da rede [4]. Uma rede blockchain na plataforma *Hyperledger Fabric* é composta por entidades do tipo[8]:

- Organização (ORG): as organizações são as representações dos membros da rede; podem ser tão grandes quanto uma empresa com vários usuários dentro da organização ou ser tão pequena quanto um só indivíduo.

- Peer: são nós responsáveis por manter a rede blockchain. Esses Peers pertencem a uma organização e são por onde as organizações enviam transações para o blockchain. Além disso, é onde são instalados e instanciados os chaincodes (termo usado na plataforma para representar contratos inteligentes) para possibilitar a execução deles.
- Orderer: um conjunto de orderer é o responsável por realizar o serviço de ordenação das transações em um bloco. Existem alguns tipos de algoritmos utilizados para realizar a ordenação das transações e a abordagem escolhida impacta diretamente no número de orderers responsáveis pelo serviço de ordenação, sendo no mínimo um.
- Membership Service Provider (MSP): é a entidade responsável por autenticar, autorizar e gerenciar identidades na rede blockchain e está vinculada a uma organização ou a um orderer. Uma organização controla os seus membros, o acesso aos peers que ela mantém e quem tem acesso aos chaincondes instalados nesses peers. Um orderer, por sua vez, controla quem pode emitir transações no canal e os tipos de transações que cada organização pode emitir.
- Chaincode: Um chaincode ou contrato inteligente é um código - invocado por um aplicativo cliente externo à rede blockchain - que gerencia o acesso e as modificações em um conjunto de pares de valores-chave no Estado Global do blockchain.

A Figura 4.1 apresenta a arquitetura da implementação, na plataforma *Hyperledger Fabric*, do componente Contratos Inteligentes - Blockchain apresentado no Capítulo 3. Essa arquitetura da implementação foi construída com o intuito de mapear os módulos de gerência de usuários e gerência de relacionamentos para contratos inteligentes blockchain.

Um canal na rede blockchain representa a rede social, como descrita no Capítulo 2; uma organização é composta por um paciente (como quem produz os dados) e pelos profissionais de saúde e familiares no papel de interessados (no protótipo foram definidas duas organizações denominadas paciente1 e paciente2); cada organização possui dois peers para manter a rede blockchain; esses peers são onde se instala e instancia os chaincodes; o algoritmo utilizado para o serviço de ordenação de transação foi o Solo [4] onde apenas um orderer é utilizado. Mais detalhes sobre a implementação realizada são apresentados nas próximas subseções.

Importante ressaltar que parte da implementação dos Contratos Inteligentes teve origem e foi construída no contexto do projeto final de curso [21], desenvolvido de forma concomitante com essa dissertação.

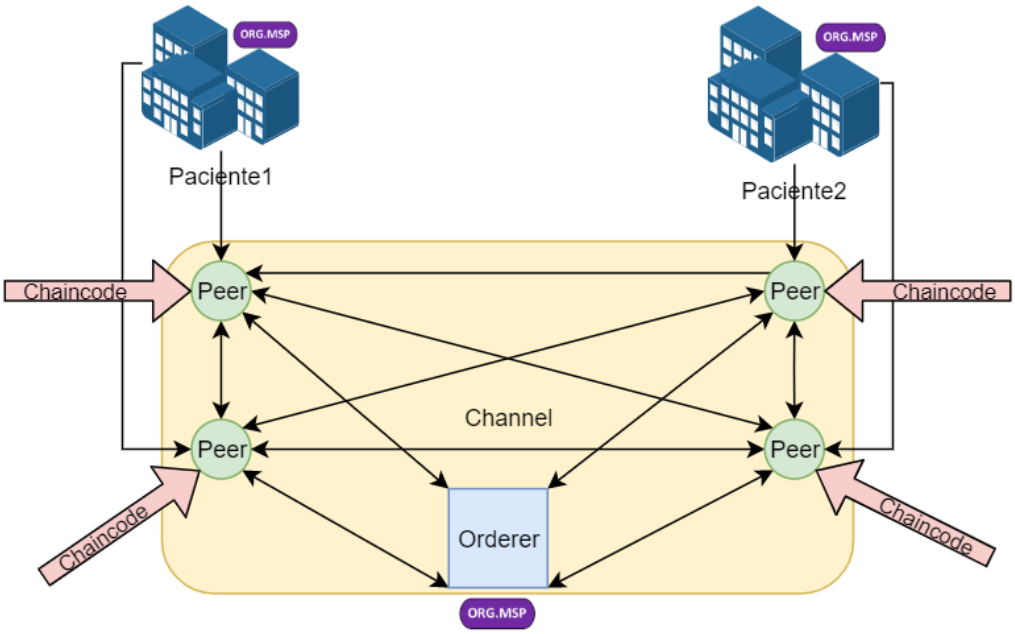


Figura 4.1: Arquitetura do Componente Contratos Inteligentes - Blockchain

4.2.1 Configuração da Rede Hyperledger Fabric

A ferramenta *cryptogen* foi utilizada para gerar o material criptográfico (certificados x509 e assinaturas digitais) para as várias entidades de rede. Esses certificados representam as identidades e permitem que a assinatura/verificação da autenticação ocorra à medida que as entidades se comunicam e realizam transações. A Cryptogen consome um arquivo - crypto-config.yaml - que contém a topologia de rede e permite gerar um conjunto de certificados e chaves para as organizações e os componentes que pertencem a essas organizações. Cada organização recebe um certificado raiz exclusivo (ca-cert) que vincula componentes específicos (pares e pedidos) a essa organização. Ao atribuir a cada organização um certificado CA exclusivo, de forma similar a uma rede típica em que um membro participante usaria sua própria autoridade de certificação, as transações e comunicações no Hyperledger Fabric são assinadas pela chave privada de uma entidade (keystore) e depois verificadas por meio de uma chave pública (signcerts).

A Figura 4.2 apresenta o trecho do arquivo crypto-config.yaml onde é realizada a configuração dos peers por organização que são os responsáveis por manter a rede; nesse caso, estão sendo utilizados dois peers por organização. Depois da execução da ferramenta cryptogen, os certificados e chaves gerados são salvos. Nesse arquivo também são definidos os orderers, responsáveis por ordenar as transações para serem inseridas no ledger.

A Figura 4.3 apresenta o trecho do arquivo crypto-config.yaml com a configuração dos orderers; nesse caso, são definidos cinco orderers e a ferramenta cryptogen gera as chaves e certificados para esses cinco orderers, mas apenas um desses orderers é uti-

```
PeerOrgs:
  # ---------------------------------------------------------
  # Paciente1
  # ---------------------------------------------------------
  - Name: Paciente1
    Domain: paciente1.example.com
    EnableNodeOUs: true
    Template:
      Count: 2
      # Start: 5
      # Hostname: {{.Prefix}}{{.Index}} # default
    # ---------------------------------------------------------
    # "Users"
    # ---------------------------------------------------------
    # Count: The number of user accounts _in addition_ to Admin
    # ---------------------------------------------------------
    Users:
      Count: 1
  # ---------------------------------------------------------
  # Paciente2: See "Paciente1" for full specification
  # ---------------------------------------------------------
  - Name: Paciente2
    Domain: paciente2.example.com
    EnableNodeOUs: true
    Template:
      Count: 2
    Users:
      Count: 1
```

Figura 4.2: Arquivo crypto-config.yaml: Configuração dos Peers por Organização

lizado para realizar o serviço de ordenação das transações do tipo Solo, utilizado nessa implementação.

```
OrdererOrgs:
  # ---------------------------------------------------------
  # Orderer
  # ---------------------------------------------------------
  - Name: Orderer
    Domain: example.com
    # ---------------------------------------------------------
    # "Specs" - See PeerOrgs below for complete description
    # ---------------------------------------------------------
    Specs:
      - Hostname: orderer
      - Hostname: orderer2
      - Hostname: orderer3
      - Hostname: orderer4
      - Hostname: orderer5
```

Figura 4.3: Arquivo crypto-config.yaml: Configuração dos Orderers

A ferramenta configtxgen é utilizada para criar quatro artefatos de configuração: o bloco genesis do orderer; o arquivo que guarda a transação de configuração do canal; e dois peers para ancorar transações - um para cada Organização. O arquivo que contém a transação de configuração do canal é transmitido para o orderer no momento da criação do canal e o peer para ancorar transações especifica o peer ancora de cada organização nesse canal.

Para criar esses artefatos a Configtxgen consome um arquivo - configtx.yaml - que contém a definição de cada tipo de entidade da rede em conjunto o MSP de cada

entidade com as permissões de leitura, escrita e também o registro de quem tem permissão de Admin naquela entidade, sejam essas entidades um orderer, uma organização, um canal ou uma aplicação.

A plataforma Hyperledger Fabric é toda baseada em containers docker-compose, portanto, no momento de de criar uma rede nessa plataforma ela utiliza diversos arquivos de configuração para controlar a criação e funcionamento desses containers. Dois arquivos são utilizados como base para criação da rede, *peer-base.yaml*, com uma descrição padrão para definir um peer (Figura 4.4) e para definir um orderer (Figura 4.5).

```
peer-base:
  image: hyperledger/fabric-peer:$IMAGE_TAG
  environment:
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    # the following setting starts chaincode containers on the same
    # bridge network as the peers
    # https://docs.docker.com/compose/networking/
    - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=${COMPOSE_PROJECT_NAME}_byfn
    - FABRIC_LOGGING_SPEC=INFO
    #- FABRIC_LOGGING_SPEC=DEBUG
    - CORE_PEER_TLS_ENABLED=true
    - CORE_PEER_GOSSIP_USELEADERELECTION=true
    - CORE_PEER_GOSSIP_ORGLEADER=false
    - CORE_PEER_PROFILE_ENABLED=true
    - CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt
    - CORE_PEER_TLS_KEY_FILE=/etc/hyperledger/fabric/tls/server.key
    - CORE_PEER_TLS_ROOTCERT_FILE=/etc/hyperledger/fabric/tls/ca.crt
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
  command: peer node start
```

Figura 4.4: Arquivo peer-base.yaml: Definição de uma entidade Peer

```
orderer-base:
  image: hyperledger/fabric-orderer:$IMAGE_TAG
  environment:
    - FABRIC_LOGGING_SPEC=INFO
    - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
    - ORDERER_GENERAL_GENESISMETHOD=file
    - ORDERER_GENERAL_GENESISFILE=/var/hyperledger/orderer/orderer.genesis.block
    - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
    - ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
    # enabled TLS
    - ORDERER_GENERAL_TLS_ENABLED=true
    - ORDERER_GENERAL_TLS_PRIVATEKEY=/var/hyperledger/orderer/tls/server.key
    - ORDERER_GENERAL_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
    - ORDERER_GENERAL_TLS_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
    - ORDERER_KAFKA_TOPIC_REPLICATIONFACTOR=1
    - ORDERER_KAFKA_VERBOSE=true
    - ORDERER_GENERAL_CLUSTER_CLIENTCERTIFICATE=/var/hyperledger/orderer/tls/server.crt
    - ORDERER_GENERAL_CLUSTER_CLIENTPRIVATEKEY=/var/hyperledger/orderer/tls/server.key
    - ORDERER_GENERAL_CLUSTER_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric
  command: orderer
```

Figura 4.5: Arquivo peer-base.yaml: Definição de uma entidade Orderer

As definições das entidades na rede *Hyperledger Fabric* são utilizadas como base para criação de todos os peers e orderers nas redes blockchain. O arquivo *docker-compose-base.yaml* especifica a criação de cada orderer e peer com as definições apresentadas em *peer-base.yaml*. O arquivo *docker-compose-base.yaml* define o nome do container do

orderer, a localização do seu MSP, onde está o arquivo para criação do bloco genesis (Figura 4.6). Além disso, também especifica o nome de cada peer, a qual organização ele pertence, quais portas na rede são utilizadas para a comunicação com cada um deles. A Figura 4.7 apresenta um exemplo de como fica essa configuração de cada peer. Essa é a última etapa de configuração da rede antes de ser possível focar no desenvolvimento dos contratos inteligentes.

```
  peer0.paciente1.example.com:
    container_name: peer0.paciente1.example.com
    extends:
      file: peer-base.yaml
      service: peer-base
    environment:
      - CORE_PEER_ID=peer0.paciente1.example.com
      - CORE_PEER_ADDRESS=peer0.paciente1.example.com:7051
      - CORE_PEER_LISTENADDRESS=0.0.0.0:7051
      - CORE_PEER_CHAINCODEADDRESS=peer0.paciente1.example.com:7052
      - CORE_PEER_CHAINCODELISTENADDRESS=0.0.0.0:7052
      - CORE_PEER_GOSSIP_BOOTSTRAP=peer1.paciente1.example.com:8051
      - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.paciente1.example.com:7051
      - CORE_PEER_LOCALMSPID=Paciente1MSP
    volumes:
        - /var/run/:/host/var/run/
        - ../crypto-config/peerOrganizations/paciente1.example.com/peers/peer0.paciente1.example.com/
msp:/etc/hyperledger/fabric/msp
        - ../crypto-config/peerOrganizations/paciente1.example.com/peers/peer0.paciente1.example.com/
tls:/etc/hyperledger/fabric/tls
        - peer0.paciente1.example.com:/var/hyperledger/production
    ports:
      - 7051:7051
```

Figura 4.6: Arquivo docker-compose-base.yaml: Definição de uma entidade Peer

```
  orderer.example.com:
    container_name: orderer.example.com
    extends:
      file: peer-base.yaml
      service: orderer-base
    volumes:
        - ../channel-artifacts/genesis.block:/var/hyperledger/orderer/orderer.genesis.block
        - ../crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/msp:/var/
hyperledger/orderer/msp
        - ../crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/tls/:/var/
hyperledger/orderer/tls
        - orderer.example.com:/var/hyperledger/production/orderer
    ports:
      - 7050:7050
```

Figura 4.7: Arquivo docker-compose-base.yaml: Definição de uma entidade Orderer

4.2.2 Implementação dos Contratos Inteligentes

Os contratos inteligentes propostos no Capítulo 3 foram utilizados como base para a implementação apresentada nessa seção. Os chaincodes foram implementados na linguagem Go e armazenados no mesmo arquivo denominado *chaincode.go* que se localiza na pasta *fabric-samples/chaincode/chaincode*. Nesse arquivo foram implementados métodos para executar a identificação dos usuários da rede social de saúde, o rastreamento de dados do paciente e a concessão de permissão de acesso a dados de saúde. A estrutura de dados utilizada para representar a entidade paciente está na Figura 4.8.

```
type paciente struct {
        ObjectType      string                  `json:"docType"`
        Nome            string                  `json:"nome"`
        RG              string                  `json:"rg"`
        CPF             string                  `json:"cpf"`
        DataNascimento  string                  `json:"data_nasc"`
        Sexo            string                  `json:"sexo"`
        NomeMae         string                  `json:"nome_mae"`
        Naturalidade    string                  `json:"naturalidade"`
        Rua             string                  `json:"rua"`
        Numero          int                     `json:"numero"`
        Complemento     string                  `json:"complemento"`
        Bairro          string                  `json:"bairro"`
        Municipio       string                  `json:"municipio"`
        Estado          string                  `json:"estado"`
        CEP             string                  `json:"cep"`
        Interessados    []listaDeInteressados   `json:"interessados"`
        Dados           []string                `json:"dadosInseridos"`
}
```

Figura 4.8: Estrutura de dados do Paciente

A Figura 4.9 apresenta o método "initPaciente", responsável por criar um objeto do tipo paciente, realizar um marshal do json e armazenar no banco atualizando o estado global do blockchain. Essa função, após criar o registro dos dados do paciente no ledger, retorna os dados encriptados (Figura 4.10).

```
paciente := &paciente{
        ObjectType:     Atuacao,
        Nome:           entradaPaciente.Nome,
        RG:             entradaPaciente.RG,
        CPF:            entradaPaciente.CPF,
        DataNascimento: entradaPaciente.DataNascimento,
        Sexo:           entradaPaciente.Sexo,
        NomeMae:        entradaPaciente.NomeMae,
        Naturalidade:   entradaPaciente.Naturalidade,
        Rua:            entradaPaciente.Rua,
        Numero:         entradaPaciente.Numero,
        Complemento:    entradaPaciente.Complemento,
        Bairro:         entradaPaciente.Bairro,
        Municipio:      entradaPaciente.Municipio,
        Estado:         entradaPaciente.Estado,
        CEP:            entradaPaciente.CEP,
        Interessados:   entradaPaciente.Interessados,
        Dados:          entradaPaciente.Dados,
}
pacienteJSONBytes, err := json.Marshal(paciente)
```

Figura 4.9: Trecho de Código da Função InitPaciente

```
encoded := base64.StdEncoding.EncodeToString(pacienteJSONBytes)
return shim.Success([]byte(encoded))
```

Figura 4.10: Retorno InitPaciente

Para invocar esse método do contrato inteligente ou qualquer outro método de um contrato inteligente na plataforma Hyperledger Fabric, é preciso antes criar a identidade

do usuário na plataforma junto com suas chaves públicas e privadas por meio da *fabric-ca* (entidade resposável pelo gerenciamento de identidades na plataforma). A identidade tem que ser incluída no MSP de uma das organizações que compõem o canal, para então invocar o método através de um peer onde o chaincode foi instalado e instanciado, assinando essa transação que invoca o método com sua identidade na rede.

A Figura 4.11 apresenta um trecho de código do método de concessão de permissão de acesso aos dados do paciente. Esse trecho de código é responsável por adicionar um novo interessado na lista de interessados do paciente, definindo seu nome, CPF e a string com a lista de dados permitidos para aquele interessado. Após adicionar o interessado na lista é realizada a chamada do método responsável por deletar a referência aos dados antigos do estado global do blockchain possibilitando assim a chamada do método que irá armazenar a nova entrada no blockchain.

```
pacienteConcedente := &paciente{}
err = json.Unmarshal(pacienteAsBytes, pacienteConcedente) //unmarshal it aka JSON.parse()
var ListanovoInteressado listaDeInteressados
ListanovoInteressado.ObjectType = "interessado"
ListanovoInteressado.Nome = args[1]
ListanovoInteressado.CPF = args[2]
ListanovoInteressado.DadosPermitidos = args[3]
pacienteConcedente.Interessados = append(pacienteConcedente.Interessados, ListanovoInteressado)
pacienteJSONasBytes, _ := json.Marshal(pacienteConcedente)
err = stub.DelState(pacienteConcedente.CPF) //reescreve o paciente
```

Figura 4.11: Trecho de Código do Método Conceder Permissão

Para simular a execução dos contratos inteligentes foram criados *scripts*, cujo os detalhes sobre como realizar a execução dos *chaincodes* por meio desses *scripts* são apresentados no Apêndice A. A invocação do método de concessão de permissão e seu retorno são apresentados na Figura 4.12. O método de concessão é invocado passando como argumento o nome da função "queryConcessao", a identidade do paciente (para simplificar a simulação da execução dos contratos foi representada pelo CPF), o nome do interessado, a identidade do interessado(também representada pelo CPF) e a string com a lista de dados permitidos(o valor para cada um dos dados pode ser 0 quando este não está permitido e 1 quando permitido). O método de concessão retorna a estrutura de dados do paciente encriptada depois de inserido o interessado na lista de interessados daquele paciente.

O método de atualização dos dados do paciente é invocado através da transação apresentada na Figura 4.13. Para que seja possível invocar a transação são passados os certificados do orderer; dos peers da organização. Além disso, é necessário invocar o método do chaincode responsável por atualizar os dados do paciente passando como argumento o nome da função do chaincode. Nesse caso "UpdatePerson"e a estrutura de

Figura 4.12: Retorno do Método Conceder Permissão

dados paciente que deve ser utilizada para atualizar a referência a essa estrutura no estado global do banco de dados.

Figura 4.13: Invocação da Transação para Atualizar os Dados do Paciente

A Figura 4.14 apresenta o retorno da transação que invocou o método de atualização dos dados do paciente. Essa transação retorna o *timestamp* com a data, hora, minuto e segundo que a transação foi processada, e o resultado dela, contendo a mensagem se o chaincode foi invocado com sucesso ou não, e como conteúdo da transação (payload) a estrutura de dados do paciente criptografada. Esse método é invocado sempre que os dados do paciente precisam ser atualizados, seja porque foi adicionado ou removido um interessado ou porque novos dados foram inseridos.

Figura 4.14: Retorno da Transação para Atualizar os Dados do Paciente

O método responsável por buscar o histórico do paciente realiza a busca em todas as transações já adicionadas no blockchain para aquela identidade de paciente, mas

antes de retornar os dados é verificado se o interessado que está requisitando os dados tem ou não permissão de acesso a eles. A Figura 4.15 apresenta a invocação do método "getHistoryPerson"passando como argumento a identidade do paciente e a identidade do interessado. Além disso, também é apresentado o retorno do método que busca o histórico do paciente para quando o interessado não possui permissão de acesso aos dados do paciente.

```
+ peer chaincode query -C mychannel -n mycc -c '{"Args":["getHistoryForPerson","75547333115", "75544
444444"]}'
+ res=1
+ set +x

Error: endorsement failure during query. response: status:500 message:"75544444444 nao tem permissao
 de acesso"
```

Figura 4.15: Retorno do Método que Busca o Histórico do Paciente para Interessado sem Permissão

O retorno do método que busca o histórico do paciente para o caso em que o interessado possui a permissão de acesso aos dados, retorna todos os dados que já foram armazenados no blockchain para aquela identidade de paciente (Figura 4.16).


```
[{"TxId":"47db9b627dd02420b39689fcf3158efa7d15c88a856a92c1730bff4bf2198871", "Value":{"docType":"pac
iente","nome":"nat","rg":"5784049","cpf":"75547333115","data_nasc":"1997-07-25","sexo":"Feminino","n
ome_mae":"Rosangela","naturalidade":"Goiania","rua":"Rua 21","numero":50,"complemento":"Apto","bairr
o":"Jaragua","municipio":"Goiania","estado":"Goias","cep":"74000000","interessados":[{"docType":"int
eressado","nome":"nat","cpf":"75547333115","dados":"11111111111111"}],"dadosInseridos":null}, "Times
tamp":"2020-01-15 02:32:10.374000092 +0000 UTC", "IsDelete":"false"},{"TxId":"b4f8d5da61fef6d0377725
6ffdd80e58e5cabd4fe508e87d29bb14be1592dd73", "Value":{"docType":"paciente","nome":"nat","rg":"578404
9","cpf":"75547333115","data_nasc":"1997-07-25","sexo":"Feminino","nome_mae":"Rosangela","naturalida
de":"Goiania","rua":"Rua 21","numero":50,"complemento":"Apto","bairro":"Jaragua","municipio":"Goiani
a","estado":"Goias","cep":"74000000","interessados":[{"docType":"interessado","nome":"nat","cpf":"75
547333115","dados":"11111111111111"},{"docType":"interessado","nome":"Lindomar","cpf":"75544444444",
"dados":"11111111111111"}],"dadosInseridos":null}, "Timestamp":"2020-01-15 02:32:47.919380468 +0000
UTC", "IsDelete":"false"},{"TxId":"7dca43669f313d34a3341f440d3305c18200e15976a4f91823f05e824d897840"
, "Value":{"docType":"paciente","nome":"nat","rg":"5784049","cpf":"75547333115","data_nasc":"1997-07
-25","sexo":"Feminino","nome_mae":"Rosangela","naturalidade":"Goiania","rua":"Rua 21","numero":50,"c
omplemento":"Apto","bairro":"Jaragua","municipio":"Goiania","estado":"Goias","cep":"74000000","inter
essados":[{"docType":"interessado","nome":"nat","cpf":"75547333115","dados":"11111111111111"},{"docT
ype":"interessado","nome":"Lindomar","cpf":"75544444444","dados":"11111111111111"},{"docType":"inter
essado","nome":"Rosa","cpf":"99999999999","dados":"11111111111111"}],"dadosInseridos":null}, "Timest
amp":"2020-01-15 02:33:09.725237004 +0000 UTC", "IsDelete":"false"}]
```


Figura 4.16: Retorno do Método que Busca o Histórico do Paciente para Interessado com Permissão

4.3 Discussão

A validação desse trabalho foi realizada através da implementação de um conjunto de contratos inteligentes implantados em uma rede *blockchain*, utilizando-a para armazenar os dados produzidos por um sistema de monitoramento remoto de pacientes, garantindo que somente as pessoas, as quais aquele paciente possui um relacionamento

na rede social online tenham acesso aos seus dados de saúde. Esse conjunto de contratos inteligentes contém desde um contrato para definir o que é uma pessoa, quais dados constituem essa entidade pessoa, qual o papel dela na rede social (paciente; profissional de saúde, familiar - no papel de interessado) e os relacionamentos entre os papéis, até para permitir ao paciente criar restrições mais específicas de que uma determinada pessoa poderá acessar os seus dados.

Estes contratos foram desenvolvidos para a plataforma *Hyperledger Fabric* sendo escritos na linguagem de programação Go. Essa plataforma foi escolhida por ser da categoria com permissão e permitir o desenvolvimento de contratos em linguagens de programação de propósito geral. A plataforma *Hyperledger Fabric*, por não garantir a privacidade dos seus contratos inteligentes e dados armazenados neles, necessita de um maior cuidado com a implementação, pois dados sensíveis como os dados de saúde de um paciente necessitam ser criptografados antes de armazenados para evitar que pessoas sem autorização tenham acesso.

Para o desenvolvimento dos contratos inteligentes na plataforma *Hyperledger Fabric* foi utilizada uma máquina virtual com o sistema operacional Ubuntu 64 bits e a versão 1.4.0 da plataforma. Antes de instalar a plataforma e iniciar o desenvolvimento dos contratos inteligentes foi necessário instalar seus pré-requisitos. Um guia desde a instalação dos pré-requisitos até a execução do chaincode está disponível no Apêndice A.

Para testar a execução do chaincode foi construído um script para simular as invocações dos métodos e calcular o tempo de execução das invocações. O tempo para processar 10 invocações ao método foi de 116 segundos e para 100 invocações o tempo foi de 1168 segundos. Para processar 300 invocações foram gastos 3473 segundos e para processar 500 invocações foram gastos 5791 segundos. Embora seja um experimento preliminar, pode-se concluir que a solução não é escalável.

Esse impacto no desempenho da rede ocorre em parte por ter sido simulada em uma máquina virtual, portanto, todos os quatro peers (dois de cada organização) e orderer estão rodando no mesmo computador. A escolha do algoritmo de ordenação de transações Solo, onde apenas um orderer é utilizado, também pode caracterizar a perda do desempenho, uma vez que há um único nó responsável por ordenar as transações, gerando um gargalo na rede blockchain. Uma outra forma de realizar a otimização no desempenho dos contratos inteligente na plataforma *Hyperledger Fabric* é a utilização do CouchDB[1] como banco de dados responsável por armazenar o estado global do blockchain. Para otimizar as buscas é possível realizar a criação de índices para os dados mais buscados,

[1] O tutorial que orienta como utilizar o CouchDB como o banco de dados que armazena o estado global do blockchain pode ser encontrado no link https://hyperledger-fabric.readthedocs.io/en/release-1.4/couchdb_tutorial.html

permitindo assim a consulta dos dados sem a necessidade de percorrer todas as entradas no banco de dados para encontrar o resultado da busca.

CAPÍTULO 5

Trabalhos Relacionados

Esse capítulo apresenta alguns trabalhos relacionados que também utilizam a tecnologia blockchain para o contexto de dados de saúde.

Os autores do trabalho [11] tem como objetivo apresentar um protótipo denominado MedRec que busca oferecer aos pacientes um registro abrangente, imutável e de fácil acesso a suas informações médicas em "provedores"de tratamento (provedores são terceiros que armazenam informações médicas de diversos pacientes, por exemplo um Hospital). O MedRec gerencia a autenticação, a confidencialidade, a responsabilidade e o compartilhamento de dados. A implementação proposta por eles aborda três problemas relacionados a Registros Médicos Eletrônicos (EMR), sendo eles a fragmentação dos dados, a falta de interoperabilidade e a falta de gerência do paciente sobre seus dados. Para isso eles reuniram referências a dados médicos diferentes e codificaram em um *blockchain*, de modo que estas referências fossem organizadas para formar um histórico para a história médica.

No *blockchain* de [11] o conteúdo do bloco representa permissões de propriedade, visualização de dados compartilhados por membros de uma rede privada, referência para os dados em um banco de dados externo e um hash dos dados visando garantir a integridade dos mesmos . Ele foi implementado utilizando uma instância do *blockchain* da *Ethereum*, portanto, é um *blockchain* sem permissão onde todas as transações são públicas e todos que pertencem àquela rede podem participar.

O MedRec organizou o seu funcionamento com base em três tipos de contratos inteligentes. A Figura 5.1 apresenta os seguintes contratos inteligentes: contrato de registro (RC), contrato de sumário (SC) e o contrato de relacionamento provedor-paciente (PPR). O contrato RC tem o intuito de registrar o paciente vinculando o seu registro a seu endereço no Ethereum e a um Contrato do tipo SC. O contrato SC serve para acompanhar o status dos relacionamentos de todos os provedores de dados de saúde com aquele paciente. O contrato PPR tem o objetivo de estabelecer o relacionamento entre um paciente e um provedor de dados de saúde, como, por exemplo, um hospital. No contrato PPR é armazenado quem é o dono da informação, uma referência para realizar o acesso aos dados, as consultas ao EMR, os hashes do banco de dados, a lista de permissão de

acesso, e a recompensa para os mineradores.

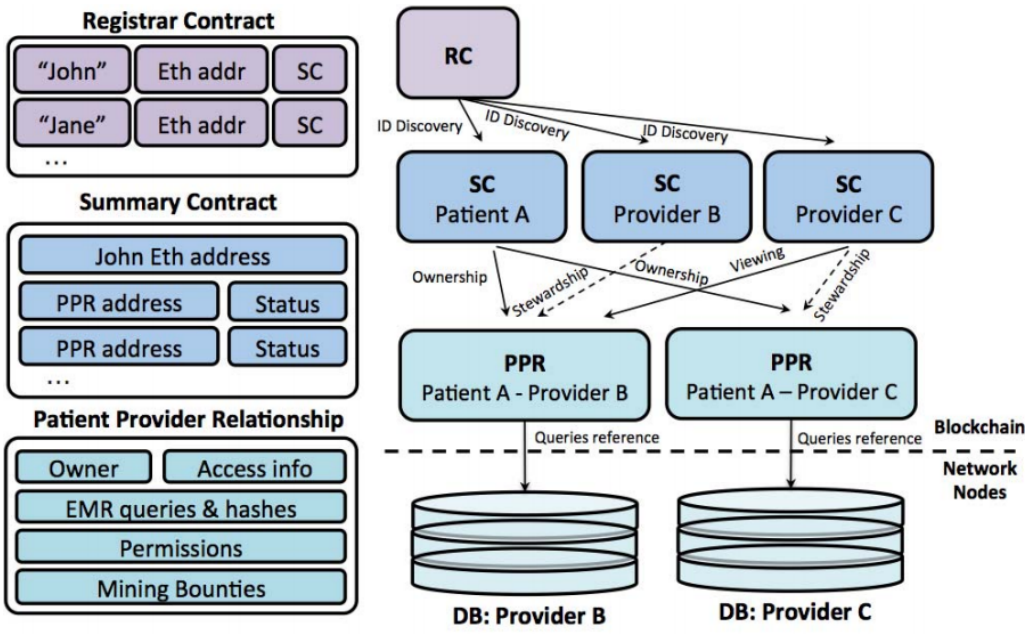


Figura 5.1: MedRec: contratos inteligentes [11]

Em [68], os autores projetam uma solução para o compartilhamento de dados entre provedores de serviços em nuvem, oferecendo controle de acesso a dados, procedência e auditoria. Eles consideram que não existe confiança entre os nós que participam do *blockchain*. A arquitetura da rede *blockchain* deles foi projetada como uma cadeia de blocos multi-chain que além da cadeia normal também possui blocos laterais que fazem parte de cada elemento individual da cadeia principal e utiliza triggers para intermediar a comunicação no sistema com bancos de dados externos onde ficam os dados. Os blocos desse *blockchain* armazenam diferentes instâncias de solicitações de acesso aos dados feitas por um determinado solicitante, onde na cadeia principal estão os solicitantes e na lateral as solicitações de cada.

A Figura 5.2 apresenta o contrato inteligente de monitoramento dos dados proposto no MeDShare. Esse contrato foi utilizado com o intuito de garantir a obtenção de dados, a auditoria desses dados de forma confiável, além de ser responsável por monitorar de perto todas as ações realizadas nos dados, relatar todas as ações realizadas por um solicitante em dados de um proprietário e caso haja alguma violação de permissão o acesso aos dados também é revogado pelo contrato inteligente.

O artigo [42] teve o intuito de garantir a integridade dos dados de saúde, a proteção da privacidade em granularidade fina e uma política de controle de acesso descentralizada. Para isso, eles apresentaram uma solução de compartilhamento de dados de saúde centrada no usuário. Para proteger a privacidade dos dados os autores utilizaram um esquema baseado em canais e para aprimorar o gerenciamento de identidade foi

Algorithm 1 Smart Contract on Data

Require: *Initialization of parameters*:
getAction, getSensitivity, getRequestorID, getOwnerID, getDataID, getKey, getMetaIndex, retrieve, encrypt, comment, report, accessControl;
Ensure: *Setting up functions*:
func (getSensitivity)
func (getAction)
func (comment)
func (accessControl)
MONITORING OF PACKAGE:
for *func* (*getAction*) == *decrypt* **do**
func (*comment*) ← Potray, Data with **DataID** has been decrypted.
retrieve (*getKey*)
encrypt (*comment*)
report (*comment*||*getRequestorID*||*getOwnerID*)
end for
if *func* (*getSensitivity*) == *Low* **then**
func (*getAction*) ← Exemptions on data.
ignore
else if *func* (*getSensitivity*) == *Low* **then**
func (*getAction*) ← Not exemptions on data (violation).
func (*comment*) ← Data violation concatenated with DataID
func (*accessControl*) ← Revokes access to data.
retrieve (*getKey*)
encrypt (*comment*)
report (*comment*||*getRequestorID*||*getOwnerID*)
else {*func* (*getSensitivity*) == *High*}
func (*getAction*) ← Violation.
func (*comment*) ← Data violation concatenated with DataID
func (*accessControl*) ← Revokes access to data.
retrieve (*getKey*)
encrypt (*comment*)
report (*comment*||*getRequestorID*||*getOwnerID*)
end if

Figura 5.2: Contrato Inteligente de monitoramento dos dados do MeDShare [68]

utilizado o serviço de membership da própria plataforma Hyperledger Fabric. Os blocos desse blockchain armazenam um hash dos dados de saúde coletados de dispositivos vestíveis, as políticas de controle de acesso, as solicitações de acesso e todas as atividades relacionadas ao monitoramento dos dados.

A Figura 5.3 apresenta a arquitetura de compartilhamento e colaboração usando um esquema baseado em canais na plataforma Hyperledger Fabric. Nessa arquitetura cada usuário possui seu próprio canal, então seus dados ficam separados dos demais usuários. Porém se uma pessoa conceder acesso para um interessado ao canal, este interessado teria acesso a todos os dados, porque as transações são transparentes para todos com acesso ao canal.

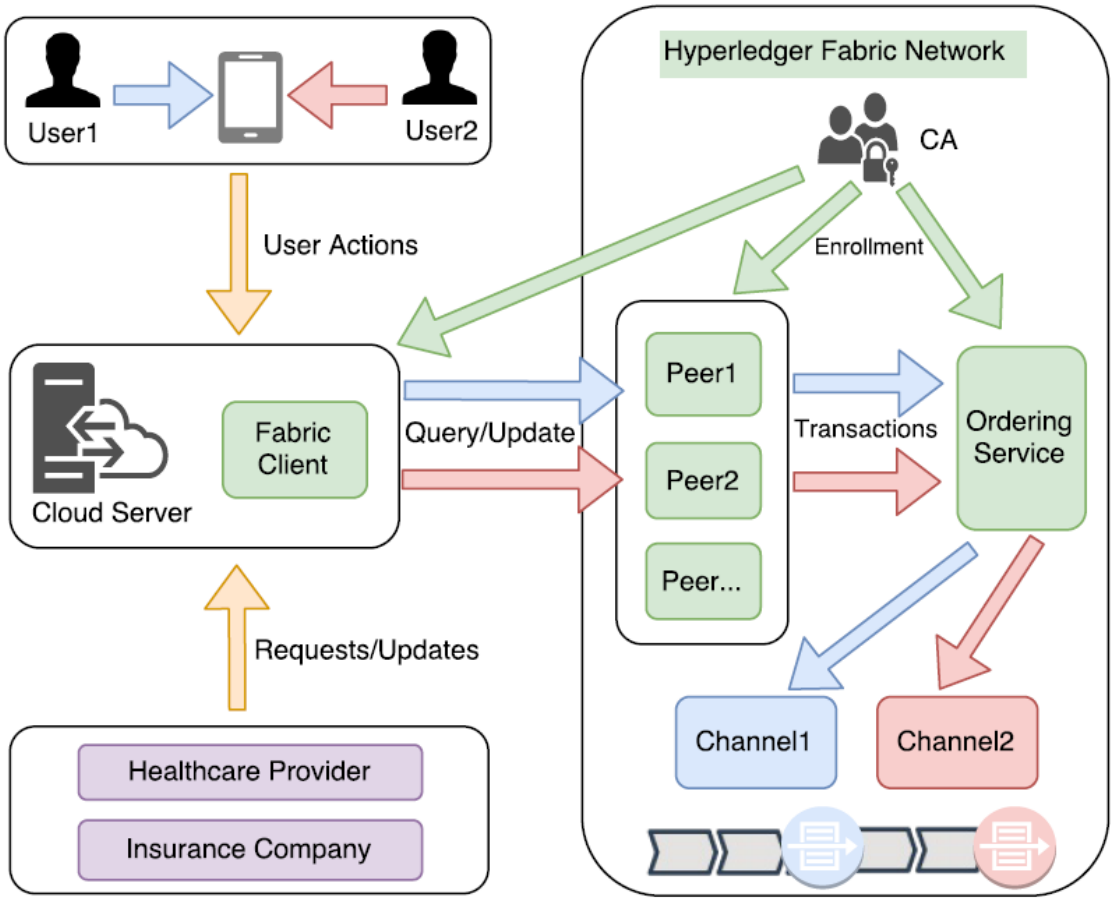


Figura 5.3: Arquitetura proposta por [42]

Em [37], os autores desenvolveram uma técnica com base em redes neurais profundas para classificação de arritmias de forma específica para um determinado paciente com base nos dados coletados do mesmo continuamente. A tecnologia blockchain foi utilizada como gerente de controle de acesso, para armazenar e acessar com segurança os dados exigidos pela rede neural de classificação durante o treinamento da rede neural em tempo real a partir de um armazenamento de dados externo.

A Figura 5.4 apresenta o funcionamento das operações de leitura e escrita dos dados utilizando contratos inteligentes para gerenciar as politicas de controle de acesso [37]. Para realizar a operação de leitura é necessário enviar a assinatura do usuário,

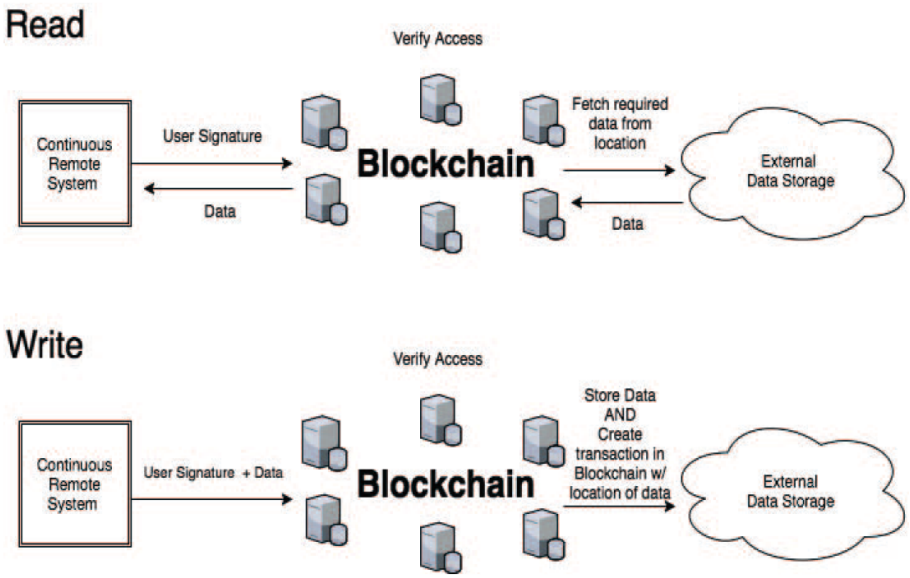


Figura 5.4: Operações de leitura e escrita dos dados utilizando contratos inteligentes para gerencias as politicas de controle de acesso [37].

o blockchain verifica as permissões de acesso, para então requisitar os dados para o banco de dados externo e retornar para o usuário. A operação de escrita precisa da assinatura do usuário mais os dados, o blockchain verifica as permissões de acesso, e envia os dados para o banco de dados externo, guardando apenas um hash dos dados no blockchain.

Algo comum entre a solução apresentada nessa dissertação e as apresentadas no MedRec [11] e no MedShare [68] é a aplicação de contratos inteligentes para monitorar os dados e todas as ações que alteram seu estado.Além disso, há um interesse em controlar quem pode ter acesso aos dados de saúde. Mas o diferencial da solução deste trabalho é em se tratar de dados coletados de um ambiente domiciliar por meio de um SMRP; o paciente ter o poder de conceder, ou não, acesso aos seus dados diretamente para outra pessoa usando relacionamentos em uma rede social, como forma de conceder permissão de acesso a seus dados; por guardar os dados dentro do blockchain e por utilizar para desenvolvimento dos contratos inteligentes a plataforma *Hyperledger Fabric*.

A solução apresentada nessa dissertação, subdivide os contratos em três, com funções semelhantes ao do MedRec. O contrato de identificação do usuário pode ser comparado ao contrato de registro; o contrato de rastreamento pode ser comparado ao contrato sumário e o contrato de concessão de permissão pode ser comparado ao de relacionamento provedor-paciente. O contrato inteligente proposto no MeDShare também tem objetivos similares e pode ser comparado ao contrato de rastreamento dos dados. Porém, na solução proposta nessa dissertação, além da identificação do paciente, também é realizada a identificação dos interessados e os dados são compartilhados de pessoa para pessoa e não entre provedores. O contrato de rastreamento, além de acompanhar o status do relacionamento entre os usuários da rede social, também é utilizado para atualizar os

dados no banco de dados e monitorar o acesso aos dados.

O artigo [42] apresenta uma solução que também utiliza a plataforma *Hyperledger Fabric* com intuito de garantir a privacidade dos dados. Essa solução se difere da nossa proposta por causa das abordagens de implementação utilizadas. Os autores do artigo escolheram realizar a implementação no nível de configurações de rede utilizando as entidades fornecidas pela plataforma, definindo um canal para cada usuário. Na abordagem utilizada por essa dissertação, a implementação se deu em nível de contratos inteligentes, ou seja, em um nível de abstração mais alto diminuindo o acoplamento com a plataforma e tratando o controle de acesso por usuário ao invés de por canal.

Os autores do artigo [37] utilizaram a tecnologia blockchain para gerenciar as políticas de controle de acesso a um banco de dados externo por meio de contratos inteligentes. A solução apresentada nessa dissertação também utiliza os contratos inteligentes para controlar o acesso, ler e escrever dados, mas os dados ficam armazenados dentro do blockchain e o intuito é de gerenciar usuários e o relacionamento entre eles numa rede social.

CAPÍTULO 6

Conclusão

Com a conclusão dessa dissertação, os objetivos propostos foram alcançados, tendo como contribuição uma solução arquitetural de concessão de permissão de acesso a dados de saúde baseada em blockchain[38], onde os relacionamentos entre os usuários de uma rede social específica de saúde são mapeados em contratos inteligentes, com o intuito de conceder permissão de acesso aos dados de saúde de um paciente, garantindo a privacidade desses dados, os quais são coletados através de um SMRP que utiliza essa rede social para disseminar os resultados obtidos por ele. Os resultados parciais dessa dissertação foram publicados no artigo [38].

Ainda em termos de contribuição, pode-se destacar os resultados obtidos pela Revisão apresentada no Capítulo 2, identificando como a tecnologia blockchain tem sido aplicada para dados de saúde e os desafios de implementação nesse contexto, e a adaptação do modelo de domínio de [23] apresentado no Capítulo 3.

O modelo de domínio envolve conceitos de sistema de monitoramento remoto de paciente, serviço de rede social e contratos inteligentes. Esse modelo de domínio tornou possível o desenvolvimento dessa solução arquitetural de concessão de permissão de acesso a dados de saúde baseada em blockchain, mapeando alguns dos componentes do modelo de domínio para contratos inteligentes blockchain.

A solução arquitetural consiste na elaboração de três contratos, o de identificação de usuário, o de rastreamento dos dados e o de concessão de permissão. O contrato de identificação é responsável por identificar o paciente, familiar, profissional de saúde ou responsável legal. A política de controle de acesso é implementada com base nesses papéis que os usuários podem assumir no sistema. Para cada papel um conjunto de informações é requisitado para criar e validar a identidade do usuário no sistema. Esse contrato é invocado pela rede social para realizar o registro de novos usuários dela no banco de dados que armazena o estado global do blockchain, por meio de uma chamada ao método do contrato de identificação responsável por criar novas identidades na rede social. O contrato de rastreamento dos dados está vinculado a um usuário com o papel de paciente. Todas as operações de leitura e escrita relacionadas aos dados do paciente são chamadas pelos métodos deste contrato deixando assim um histórico completo e auditável

sobre a manipulação desses dados. O contrato de concessão de permissão de acesso aos dados está vinculado a um paciente e a um contrato de rastreamento dos dados. Nele, o paciente registra para qual usuário ele está concedendo permissões de acesso a seus dados, especificando as restrições que desejar, tanto em relação ao nível de permissão de acesso quanto a quais dados específicos.

Algumas limitações foram encontradas após o desenvolvimento do trabalho. Uma delas é o fato da rede ter sido implementada em uma máquina virtual onde todos os nós da rede são executados, fazendo com que o desempenho do processador seja subdividido para cada um dos nós. A forma como a rede foi configurada também acarretou impacto no desempenho da rede, pois com apenas um nó ordenando as transações gera-se gargalo. Outra limitação está relacionada à não avaliação da abordagem uma vez integrada a um sistema de monitoramento remoto de pacientes real com sensores fisiológicos físicos ao invés de simulados. O desenvolvimento dos contratos não teve como requisito os padrões de registros eletrônicos de saúde ou a interoperabilidade com esses padrões. Por fim, outra limitação está em não utilizar o gerenciamento de dados privados fornecido pela própria plataforma *Hyperledger Fabric* ao invés de encriptar os dados manualmente.

Como trabalhos futuros as pretensões são de dar continuidade no desenvolvimento da arquitetura e da implementação. Realizar um estudo sobre a viabilidade de se realizar a implementação na plataforma Enigma. Um outro trabalho futuro poderia ser o desenvolvimento de um aplicativo da rede social que seja integrado com os contratos inteligentes apresentados nessa dissertação ou o desenvolvimento da interface de comunicação entre o UbiCare Social e os contratos inteligentes. Além disso, também pode ser feito um upgrade nas configurações da rede para que ela adicione novas organizações dinamicamente e otimize o desempenho da rede. O banco de dados de estado pode ser alterado para o CouchDB criando índices para os dados mais buscados para não ter que percorrer todas as transações para retornar o resultado da busca. Além disso, podem ser utilizadas as funções para tratar dados privados fornecidas pela própria plataforma *Hyperledger Fabric*. Há também a necessidade de se aprofundar em avaliações de desempenho e escalabilidade da tecnologia blockchain, suas plataformas e abordagens de implementação de contratos inteligentes. Por fim executar testes reais com a devida integração com um sistema de monitoramento remoto de pacientes e sensores fisiológicos físicos ao invés de simulados.

Referências Bibliográficas

[1] O paciente irá vê-lo agora, doutor: Como a saúde digital fortalece a mudança na medicina. https://bit.ly/2TPErU6. Acessado: 25/11/2018.

[2] Technical details. http://www.doc.ic.ac.uk/~ma7614/topics_website/tech.html. Acessado: 25/11/2018.

[3] Lei Geral de Proteção de Dados Pessoais (LGPD). http://www.planalto.gov.br/ccivil_03/_ato20152018/2018/lei/L13709.htm, Agosto 2018.

[4] Tutorial para construir sua primeira rede na plataforma *Hyperledger Fabric*, Janeiro 2020.

[5] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, UK, 1999. Springer-Verlag.

[6] Z. Alhadhrami, S. Alghfeli, M. Alghfeli, J. A. Abedlla, and K. Shuaib. Introducing blockchains for healthcare. In *2017 International Conference on Electrical and Computing Technologies and Applications, ICECTA 2017*, volume 2018-Janua, pages 1–4. IEEE, nov 2018.

[7] M. Allen. Como o blockchain pode afetar em breve a vida cotidiana. https://www.swissinfo.ch/por/economia/transa%C3%A7%C3%B5es-digitais_como-o-blockchain-pode-afetar-em-breve-a-vida-cotidiana/43643288. Acessado: 25/11/2018.

[8] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 30. ACM, 2018.

[9] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, and Y. e. a. Manevich. Hyperledger fabric: A distributed operating system for permissioned blockchains. 2018.

[10] S. U. Ayubi and B. Parmanto. Persona: Persuasive social network for physical activity. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2153–2157. IEEE, 2012.

[11] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman. MedRec: Using blockchain for medical data access and permission management. In *Proceedings - 2016 2nd International Conference on Open and Big Data, OBD 2016*, pages 25–30. IEEE, aug 2016.

[12] A. Back et al. Hashcash-a denial of service counter-measure. ftp://sunsite.icm.edu.pl/site/replay.old/programs/hashcash/hashcash.pdf, 2002.

[13] D. Battisti and S. Carvalho. Aplicação do padrão ISO/IEEE 11073 no contexto da assistência domiciliar à saúde. 2016. In: Escola Regional de Informática (ERIGO), 2016, Goiânia. Anais da IV Escola Regional de Informática (ERIGO), 2016.

[14] d. m. boyd and N. B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230.

[15] C. Cachin. Architecture of the hyperledger blockchain fabric. 2016.

[16] S. T. Carvalho, L. Murta, and O. Loques. Variabilities as first-class elements in product line architectures of homecare systems. In *Software Engineering in Health Care (SEHC), 2012 4th International Workshop on*, pages 33–39. IEEE, 2012.

[17] M. Ceron. Habilidades de comunicação: Abordagem centrada na pessoa. *São Paulo: UNA-SUS, UNIFESP*, 2010.

[18] R. Chan. Blockchain data structure. https://www.linkedin.com/pulse/blockchain-data-structure-ronald-chan. Acessado: 25/11/2018.

[19] D. Cosset. Blockchain: what is in a block? https://dev.to/damcosset/blockchain-what-is-in-a-block-48jo. Acessado: 25/11/2018.

[20] J. Cunningham and J. Ainsworth. Enabling patient control of personal electronic health records through distributed ledger technology. In *Studies in Health Technology and Informatics*, volume 245, pages 45–48, 2017.

[21] G. A. da Silva. Implementação de Contratos Inteligentes na Plataforma Hyperledger., 2019. Monografia (Bacharel em Ciência da Computação), UFG (Universidade Federal de Goiás), Goiânia, Brasil.

[22] D. Daglish and N. Archer. Electronic personal health record systems: a brief review of privacy, security, and architectural issues. In *Privacy, Security, Trust and the Management of e-Business, 2009. CONGRESS'09. World Congress on*, pages 110–120. IEEE, 2009.

[23] H. de Almeida Ribeiro. Serviços de redes sociais para a disseminação de informações de saúde em sistemas de monitoramento remoto de pacientes. Master's thesis, Universidade Federal de Goiás, Instituto de Informática - INF/UFG, 2018.

[24] S. T. de Carvalho, A. Copetti, and O. G. Loques Filho. Sistema de computação ubíqua na assistência domiciliar à saúde. *Journal Of Health Informatics*, 3(2), 2011.

[25] L. V. Doering, K. Hickey, D. Pickham, B. Chen, and B. J. Drew. Remote noninvasive allograft rejection monitoring for heart transplant recipients: study protocol for the novel evaluation with home electrocardiogram and remote transmission (new heart) study. *BMC cardiovascular disorders*, 12(1):14, 2012.

[26] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K. K. R. Choo. Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy? *IEEE Cloud Computing*, 5(1):31–37, jan 2018.

[27] A. Fantoni. Dispositivos wearable para o campo da saúde: reflexões acerca do monitoramento de dados do corpo humano. *Temática*, 12(01), 2016.

[28] J. L. Fernández-Alemán, I. C. Señor, P. Á. O. Lozoya, and A. Toval. Security and privacy in electronic health records: A systematic literature review. *Journal of biomedical informatics*, 46(3):541–562, 2013.

[29] C. M. Filho. O juramento de hipócrates e o cÓdigo de ética médica. *Residência Pediátrica*, pages 6(1):45–46, 2016. DOI: https://doi.org/10.25060/residpediatr-2016.v6n1-10.

[30] E. Germano, D. Battisti, H. Ribeiro, and S. Carvalho. Plano de cuidados ubíquo para acompanhamento domiciliar de pacientes. 2016. In: Congresso Brasileiro de Informática em Saúde (CBIS), 2016. Congresso Brasileiro de Informática em Saúde (CBIS), 2016. p. 849-858.

[31] E. Germano, S. Carvalho, and J. De Souza-Zinader. Plano de cuidados ubíquo com sistema de notificações voltado a pacientes domiciliares. *III Escola Regional de Informática de Goiás.*, pages 33–44, 2015.

[32] F. Greve, L. Sampaio, J. Abijaude, A. Coutinho, Í. Valcy, and S. Queiroz. Blockchain e a revolução do consenso sob demanda. In: Minicursos do XXXVI Simpósio Brasileiro

de Redes de Computadores e Sistemas Distribuídos - SBRC, ISBN: 978-85-7669-442-7, Campos do Jordão, SP, Brasil, Maio 2018.

[33] R. Guo, H. Shi, Q. Zhao, and D. Zheng. Secure Attribute-Based Signature Scheme with Multiple Authorities for Blockchain in Electronic Health Records Systems. *IEEE Access*, 6:11676–11686, 2018.

[34] M. B. Hoy. An Introduction to the Blockchain and Its Implications for Libraries and Medicine. *Medical Reference Services Quarterly*, 36(3):273–279, jul 2017.

[35] D. Ichikawa, M. Kashiyama, and T. Ueno. Tamper-Resistant Mobile Health Using Blockchain Technology. *JMIR mHealth and uHealth*, 5(7):e111, jul 2017.

[36] International Organization for Standardization ISO. *ISO 27000, "Information Technology, Security Techniques, Information Security Management Systems, Overview and Vocabulary"*, 2009.

[37] A. Juneja and M. Marefat. Leveraging blockchain for retraining deep learning architecture in patient-specific arrhythmia classification. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 393–397. IEEE, mar 2018.

[38] N. R. Junqueira, G. A. da Silva, and S. T. de Carvalho. Concessão de permissão a dados de saúde baseada em blockchain. In *Anais da Escola Regional de Informática de Goiás*, pages 395–406. SBC, 2019.

[39] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine*, 10(1):156–167, 2006.

[40] C. Khorakhun and S. N. Bhatti. mhealth through quantified-self: a user study. In *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*, pages 329–335. IEEE, 2015.

[41] B. Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 2004.

[42] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li. Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, volume 2017-Octob, pages 1–5. IEEE, oct 2018.

[43] W. Liu, S. Zhu, T. Mundie, and U. Krieger. Advanced block-chain architecture for e-health systems. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6. IEEE, oct 2017.

[44] S. K. Lo, X. Xu, Y. K. Chiam, and Q. Lu. Evaluating Suitability of Applying Blockchain. In *Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems, ICECCS*, volume 2017-Novem, pages 158–161. IEEE, nov 2018.

[45] C. Lovis, S. Spahni, N. Cassoni, and A. Geissbuhler. Comprehensive management of the access to the electronic patient record: Towards trans-institutional networks. *International Journal of Medical Informatics*, 76(5-6):466–470, 2007.

[46] G. Magyar. Blockchain: Solving the privacy and research availability tradeoff for EHR data: A new disruptive technology in health data management. In *IEEE 30th Jubilee Neumann Colloquium, NC 2017*, volume 2018-Janua, pages 135–140. IEEE, nov 2018.

[47] P. Mamoshina, L. Ojomoko, Y. Yanovich, A. Ostrovski, A. Botezatu, P. Prikhodko, E. Izumchenko, A. Aliper, K. Romantsov, A. Zhebrak, I. O. Ogu, and A. Zhavoronkov. Converging blockchain and next-generation artificial intelligence technologies to decentralize and accelerate biomedical research and healthcare. *Oncotarget*, 9(5):5665–5690, jan 2015.

[48] M. Mettler. Blockchain technology in healthcare: The revolution starts here. In *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services, Healthcom 2016*, pages 1–3. IEEE, sep 2016.

[49] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *BMJ*, 2009.

[50] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *In: www.bitcoin.org*, 2008.

[51] L. Network. CryptoZombies: Aprenda a Programar DApps Ethereum Construindo o Seu Próprio Jogo. https://cryptozombies.io/pt/. Acessado: 25/11/2018.

[52] F. F. Reis, A. Costa-Pereira, and M. E. Correia. Access and privacy rights using web security standards to increase patient empowerment. *Studies in health technology and informatics*, 137:275–285, 2008.

[53] J. W. Rettberg. *Seeing ourselves through technology: How we use selfies, blogs and wearable devices to see and shape ourselves*. Springer, 2014.

[54] H. Ribeiro, D. Battisti, E. Germano, and S. Carvalho. Notificações de monitoramento remoto de pacientes usando redes sociais. 2016. In: Anais do XV Congresso Brasileiro de Informática em Saúde (CBIS), 2016. p. 859-868.

[55] H. A. Ribeiro, E. Germano, S. T. Carvalho, and E. S. Albuquerque. Integrating social networks and remote patient monitoring systems to disseminate notifications. In *MEDINFO 2017: Precision Healthcare Through Informatics: Proceedings of the 16th World Congress on Medical and Health Informatics*, volume 245, page 198. IOS Press, 2018.

[56] N. Rifi, E. Rachkidi, N. Agoulmine, and N. C. Taher. Towards using blockchain technology for eHealth data access management. In *2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME)*, pages 1–4. IEEE, oct 2017.

[57] A. Roehrs, C. A. da Costa, and R. da Rosa Righi. OmniPHR: A distributed architecture model to integrate personal health records. *Journal of Biomedical Informatics*, 71:70–81, jul 2017.

[58] L. Røstad. An initial model and a discussion of access control in patient controlled health records. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, pages 935–942. IEEE, 2008.

[59] Z. Shae and J. J. Tsai. On the Design of a Blockchain Platform for Clinical Trial and Precision Medicine. In *Proceedings - International Conference on Distributed Computing Systems*, pages 1972–1980. IEEE, jun 2017.

[60] M. Siddiqi, S. T. All, and V. Sivaraman. Secure lightweight context-driven data logging for bodyworn sensing devices. In *2017 5th International Symposium on Digital Forensic and Security (ISDFS)*, pages 1–6. IEEE, 2017.

[61] M. Stewart, J. B. Brown, W. W. Weston, I. R. McWhinney, C. L. McWilliam, and T. R. Freeman. *Medicina centrada na pessoa: transformando o método clínico*. Artmed Editora, 2017.

[62] P. Tasatanattakool and C. Techapanupreeda. Blockchain: Challenges and applications. In *International Conference on Information Networking*, volume 2018-Janua, pages 473–475. IEEE, jan 2018.

[63] M. G. Testa, C. E. B. de Azevedo Bragança, and E. M. Luciano. Privacidade de informações de pacientes de instituições de saúde: a percepção de profissionais da área de saúde. *Revista Reuna (Belo Horizonte)*, 16(2):89–102, 2011.

[64] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian. Continuous patient monitoring with a patient centric agent: A block architecture. *IEEE Access*, 6:32700–32726, 2018.

[65] H. van der Linden, D. Kalra, A. Hasman, and J. Talmon. Inter-organizational future proof ehr systems: a review of the security and privacy related issues. *International journal of medical informatics*, 78(3):141–160, 2009.

[66] J. Veiga, J. P. Rodriguez, B. Trevizan, M. T. Rebonatto, A. C. B. D. Marchi, et al. Aplicações móveis com interação médico-paciente para um estilo de vida saudável: uma revisão sistemática. 2017.

[67] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

[68] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani. MeDShare: Trust-Less Medical Data Sharing among Cloud Service Providers via Blockchain. *IEEE Access*, 5:14757–14767, 2017.

[69] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba. A Taxonomy of Blockchain-Based Systems for Architecture Design. In *Proceedings - 2017 IEEE International Conference on Software Architecture, ICSA 2017*, pages 243–252. IEEE, apr 2017.

[70] D. Yaga, P. Mell, N. Roby, and K. Scarfone. Blockchain technology overview. *NISTIR*, 8202, 2018.

[71] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang. Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control. *Journal of Medical Systems*, 40(10):218, oct 2016.

[72] J. Zhang, N. Xue, and X. Huang. A secure system for pervasive social network-based healthcare. *IEEE Access*, 4:9239–9250, 2016.

[73] P. Zhang, M. A. Walker, J. White, D. C. Schmidt, and G. Lenz. Metrics for assessing blockchain-based healthcare decentralized apps. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–4. IEEE, oct 2017.

[74] Y. Zhu. Automatic detection of anomalies in blood glucose using a machine learning approach. *Journal of Communications and Networks*, 13(2):125–131, 2011.

[75] G. Zyskind, O. Nathan, and A. Pentland. Enigma: Decentralized computation platform with guaranteed privacy. *arXiv preprint arXiv:1506.03471*, 2015.

APÊNDICE A

Guia para desenvolvedores

Esse apêndice apresenta orientações sobre os pré-requisitos e instalação da rede Hyperledger Fabric e por fim, como realizar a execução dos chaincodes. Esse capítulo foi criado o intuito de facilitar a continuidade deste trabalho e foi construído com base na experiência adquirida durante a utilização da plataforma e também nos tutoriais fornecidos pela Hyperledger Fabric.

A.1 Pré-Requisitos de Instalação

Para implementação deste trabalho foi utilizada uma máquina virtual com o sistema operacional Ubuntu versão 18.04, em seguida foram instalados cada um dos pré-requisitos, para evitar conflitos em decorrência da utilização de versões ou configurações diferentes, estão listados em conjunto com as versões utilizadas.

1. Git 2.17.1: o comando utilizado para sua instalação foi *apt install git-all*.
2. cURL 7.58.0: o comando utilizado para sua instalação foi *apt-get install curl*.
3. Docker 19.03.5: os comandos utilizados para sua instalação foram:
   apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common;
   add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable";
   apt-get install docker-ce docker-ce-cli containerd.io.
4. Docker Compose 1.25.0: os comandos utilizados para sua instalação foram:
   curl -L "https://github.com/docker/compose/releases/download/1.25.0/docker-compose-$(uname -s)-$(uname -m)-o /usr/local/bin/docker-compose;
   chmod +x /usr/local/bin/docker-compose;
   ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose.
5. Go 1.13.5: o comando utilizado para sua instalação foi *tar -C /usr/local -xzf go1.13.5.linux-amd64.tar.gz*. Importante lembrar de exportar a GOPATH no arquivo /etc/profile e depois realizar um source desse arquivo para que

essas mudanças tenham efeito, o comando utilizado para isso foi "export PATH=$PATH:usr/local/go/bin".

6. Node.js 12.14.0: o comando utilizado para sua instalação foi *tar -xJvf node-v12.14.0-linux-x64.tar.xz -C /usr/local/lib/nodejs*. Para o node.js também foi necessário exportar a PATH, o comando utilizado para isso foi "export PATH=$PATH:usr/local/lib/nodejs/bin:$PATH"e depois foi executado o source /etc/profile.
7. Npm 6.13.4: foi instalado junto ao Node.js.
8. Python 2.7.17: o comando utilizado para sua instalação foi *apt-get install python*.

A.2 Instalação da rede Hyperledger Fabric

Após a Instalação dos Pré-requisitos foi possível realizar o clone do código fonte da plataforma Hyperledger Fabric. Para executar o clone foi utilizado o comando: *git clone https://github.com/hyperledger/fabric-samples/tree/release-1.4*. A plataforma pode alterar seu funcionamento em versões posteriores, portanto, é necessário instalar a versão 1.4.0 ou uma que seja compatível com ela. Antes que tentar executar a rede da hyperledger foi preciso dar permissão de execução para os scripts baixados utilizando o comando *chmod 777 -R fabric-samples/*, fazer o download dos binários, imagens do docker e executar dentro da pasta "fabric-samples"o script com o comando *./scripts/bootstrap.sh* que por padrão irá instalar a versão do fabric que foi selecionada na hora do clone do código fonte do github, mas também é possível passar por parâmetro a versão dos binários a qual deseja instalar.

A.3 Execução dos Chaincodes

Para execução dos chaincodes é necessário baixar os arquivos contidos no link "https://bit.ly/2TzL4MG", e exportar dentro da pasta *fabric-samples*, ele irá substituir a pasta *chaincode* e a *first-network*. Após o download dos arquivos é necessário alterar os caminhos para a localização das entidades e artefatos utilizados pela plataforma para subir a rede blockchain. Os arquivos que precisão ser alterados são:

- docker-compose-cli.yaml
- docker-compose-e2e-template.yaml
- docker-compose-paciente3.yaml
- eyfn.sh
- na pasta scripts:
  step1paciente3.sh

step2paciente3.sh
step3paciente3.sh
testpaciente3.sh

Nesses arquivos deve se fazer uma busca pelo caminho */home/nat/fabric-samples* o qual deve ser substituído pelo caminho correspondente a pasta *fabric-samples*, como, por exemplo *.../nomeusuario/fabric-samples*.

Depois da alteração dos caminhos é possível realizar a execução da rede dentro da pasta *fabric-samples/first-network* por meio dos comandos:

- *./byfn.sh generate* para gerar os certificados para as entidades da rede.
- *./byfn.sh up* para subir a rede e executar o chaincode.
- *./byfn.sh down* para deletar todos os arquivos e containers docker gerados para a execução da rede.

A Figura A.1 apresenta a mensagem retornada quando a rede blockchain é executada com sucesso.

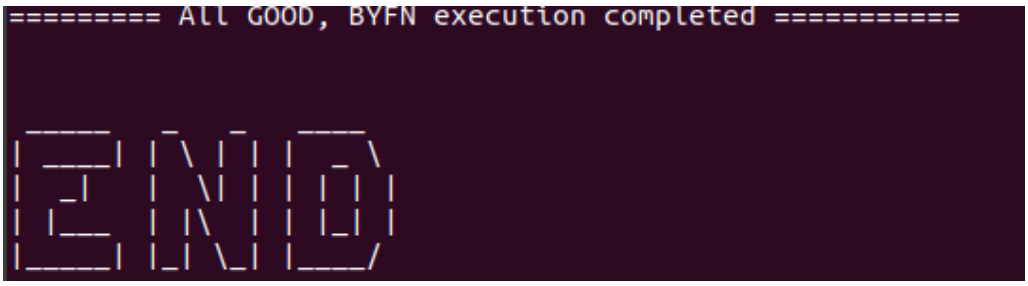


Figura A.1: Mensagem de quando a rede foi executada com sucesso