UNIVERSIDADE ESTADUAL DO CEARÁ

CENTRO DE CIÊNCIAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

RAPHAEL LIMA SARAIVA

UMA SOLUÇÃO BASEADA EM BLOCKCHAIN PARA GERENCIAR REGISTROS DE PROFISSIONAIS MÉDICOS

FORTALEZA – CEARÁ

2018

RAPHAEL LIMA SARAIVA

UMA SOLUÇÃO BASEADA EM BLOCKCHAIN PARA GERENCIAR REGISTROS DE PROFISSIONAIS MÉDICOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. Jerffeson Teixeira de Souza

FORTALEZA – CEARÁ

2018

Deve ser gerada através do preenchimento do Formulário Eletrônico de Elaboração da Ficha Catalográfica, disponível no link: http://www.uece.br/biblioteca/index.php/entrega-de-trabalho.

X000x Sobrenome, Nome do 1º autor. (citado na folha de rosto)
Título principal: subtítulo./Nome completo do 1º autor, Nome completo do 2º autor, Nome completo do 3º autor; orientação [de]. – Local: ano.
Nº de folhas.: il.(se houver ilustração); 30 cm.

Inclui bibliografias: f.(nº da folha em que se encontra)
Trabalho de Conclusão de Curso (Graduação em) – Universidade Estadual do Ceará – (UECE).

1. Assunto. 2. Assunto. 3. Assunto. I. Sobrenome, Nome do 2º autor. II. Sobrenome, Nome do 3º autor. III. Sobrenome, Nome do orientador (orient.). IV. Universidade Estadual do Ceará – UECE. V. Título.

CDU

RAPHAEL LIMA SARAIVA

UMA SOLUÇÃO BASEADA EM BLOCKCHAIN PARA GERENCIAR REGISTROS DE PROFISSIONAIS MÉDICOS

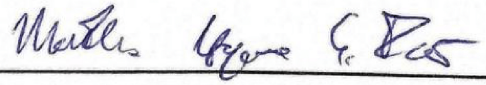
Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Jerffeson Teixeira de Souza (Orientador)
Universidade Estadual do Ceará – UECE

Prof. Dr. Samuel Façanha Câmara
Universidade Estadual do Ceará - UECE



Prof. Dr. Matheus Henrique Esteves Paixão
Universidade Estadual do Ceará - UECE

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

AGRADECIMENTOS

Primeiramente agradeço a Deus, pelo fim de mais essa etapa, pelos sonhos que se concretizam.
Aos meus pais Edilva Lima Saraiva e Gilson Mendes Saraiva e minha avó Terezinha Saraiva Mendes, pelo incentivo, amor e apoio.
Ao prof. PhD. Jerffeson Teixeira de Souza pela oportunidade dada e apoio na elaboração deste trabalho. Um agradecimento especial a Allysson Allex e Pamella Soarez por sua incrível auxilio que contribuiu para conclusão deste trabalho
Aos professores do Mestrado Acadêmico em Ciência da Computação (MACC) que foram importantes em minha vida acadêmica por proporcionaram-me o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional. À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.
Ao Prof. Dr. Matheus Henrique Esteves Paixão e Prof. Dr. Samuel Façanha Câmara, por aceitarem o convite de participação e contribuição em minha banca avaliadora.
Aos meus grandes amigos e companheiros, Márcio Silva, Jefferson Ramos, Sprince Marques, Samuel Araújo, Lucas Melo, Newrislane Costa e Marcos Sombra.
Por fim a todos outros não citados que fizeram parte da minha formação, o meu muito obrigado.

"Quando eu estava na escola, o computador era uma coisa muito assustadora. As pessoas falavam em desafiar aquela máquina do mal que estava sempre fazendo contas que não pareciam corretas. E ninguém pensou naquilo como uma ferramenta poderosa."

(Bill Gates)

RESUMO

As agências fiscais têm enfrentado vários desafios em relação à garantia da profissão jurídica. Em particular, o contexto do exercício da medicina é crucial, uma vez que os profissionais envolvidos nessa área são responsáveis por lidar com a saúde de terceiros. No entanto, podemos notar vários incidentes e fraudes relacionados aos registros de profissionais médicos, como a emissão de diplomas falsos ou a disponibilidade de informações não validadas pelo Conselho Regional de Medicina. Assim, a fim de evitar essas fraudes, supõem-se ser relevante investigar soluções inovadoras visando manter de forma segura e transparente a história profissional de cada médico. Neste contexto, o principal objetivo dete trabalho é apresentar uma solução baseada em blockchain fundamentada na plataforma *Hyperledger* que permita de forma descentralizada e confiável o armazenamento de informações relevantes necessárias para o gerenciamento de registros de profissionais médicos. Além disso, os eventos cobertos por essa solução foram validados por meio de uma entrevista semiestruturada com um diretor de alto nível de um Conselho Regional de Medicina. Por isso, o presente trabalho contribuiu no avanço do conhecimento, esclarecendo o desenvolvimento da solução proposta utilizando a tecnologia *Hyperledger* e, consequentemente, possibilitando o aumento da transparência e confiabilidade dos dados referidos.

Palavras-chave: CRM. Blockchain. Hyperledger.

ABSTRACT

The fiscal agencies have been facing several challenges regarding the guarantee of the legal profession. In particular, the context of medicine exercise poses as a crucial one, since the professionals involved in this area are responsible for dealing with the health of third parties. However, we may notice a number of incidents and frauds related to the records of medical professionals, such as issuing forged diplomas or availability of information not validated by the Regional Council of Medicine. Thus, in order to avoid these frauds, we assume to be relevant to investigate innovative solutions aiming at maintaining in a safe and transparent way the professional history of each doctor. In this context, our main objective is to present a blockchain-based solution based on the Hyperledger platform that allows in a decentralized and reliable fashion the storage of relevant information necessary for managing records of medical professionals. In addition, the events covered by our solution was validated through a semi-structured interview with a high-level director of a Regional Council of Medicine. Therefore, we contribute and advance the knowledge by clarifying the development of the proposed solution using the Hyperledger technology and, consequently, enabling the increase of transparency and reliability of the referred data.

Keywords: CRM. Blockchain. Hyperledger.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE SÍMBOLOS

DLT	*Distributed Ledger Technologie*
CRM	Conselho Regional de Medicina
CEM	Código de Ética Médica
CFM	Conselho Federal de Medicina
P2P	*peer-to-peer*
PoW	*Proof of Work*
PoS	*Proof of Stake*
PoA	*Proof of Activity*
PoI	*Proof of Importance*
PoC	*Proof of Capacity*
DApp	*Decentralized Application*

SUMÁRIO

1 INTRODUÇÃO

Neste capítulo são apresentados os principais fatores que levaram ao desenvolvimento desse trabalho, assim como os objetivos a serem alcançados pelo mesmo. A estrutura do restante do trabalho é apresentada ao final.

1.1 MOTIVAÇÃO

A atuação dos órgãos de fiscalização profissional tem sido de real relevância para a sociedade, tendo em vista a necessidade de elaboração e manutenção de normas específicas e particulares de cada profissão em prol da garantia do exercício legal do referida trabalho. Nesse contexto, pode-se destacar os profissionais relacionados à medicina, dado o impacto e importância social que os mesmos representam. Por se tratar de cuidados com a saúde e integridade física do ser humano, os atuantes deste campo precisam dedicar-se de forma diligente, sem negligenciar os cuidados necessários à terceiros, cumprindo, assim, suas obrigações de maneira zelosa, honesta e legal (BRASIL, c).

Devido a interação entre diferentes instituições, regiões e atores necessária para garantia do exercício legal da medicina, faz-se necessário que os dados, referentes às informações profissionais dos médicos, possam ser gerenciados e mantidos de maneira segura e confiável. Tal motivação advém da necessidade em se evitar fraudes e inconsistências de informações as quais podem propiciar crimes relacionados ao exercício ilegal da medicina, conforme prescrito na Lei 2.848/40 do Código Penal, Artigo 282 (BRASIL, f). Dessa maneira, faz-se necessário o uso de tecnologias que proporcionem uma segurança adequada, amenizando ou, até mesmo, evitando as vulnerabilidades dos dados, possíveis fraudes e os demais infortúnios que são decorrentes delas. Problemas esses que têm afetado e, até mesmo, posto em risco a vida de terceiros. Além disso, o registro, a manutenção e a divulgação adequada dessas informações têm a possibilidade de promover o fomento à cultura da transparência das informações relevantes ao público.

Assim, a busca e o desenvolvimento de tecnologias inovadoras que disponham de segurança para manipulação de dados significativos, como os previamente contextualizados, têm se mostrado cada vez mais relevante para a sociedade. Com os avanços no desenvolvimento dessas tecnologias, pode-se colocar em destaque o uso de livros-razão distribuídos ou DLTs (do inglês *Distributed Ledger Technologies*), com os quais tornou-se possível a definição e o uso de aplicações descentralizadas. As DLTs permitem o registro organizado de informações

diversas a depender da aplicação e semântica que seja dada a tais dados. Um caso especial de tecnologias dessa natureza são as *blockchains*, que implementam a ideia de DLTs por meio de um mecanismo organizacional baseado em encadeamento de blocos (PIRES et al., 2018).

Uma *blockchain* pode ser entendida como uma tecnologia de livros-razão de propósito geral que oferece uma base de dados altamente transparente, segura e resiliente contra falhas (DAVIDSON et al., 2016). A resiliência contra falhas ocorre devido a dois fatores: i) a *blockchain* ser replicada em cada nó que compõe a rede; ii) a existência de robustos mecanismos de consenso entre os nós garantindo a integridade do livro-razão. Assim, as plataformas que utilizam *blockchain* permitem que partes completamente anônimas e que não confiam entre si possam formar uma rede que armazena informações confiáveis (WUST; GERVAIS, 2018). Logo, a tecnologia *blockchain* tornou-se uma alternativa aos tradicionais sistemas centralizados, dispensando a necessidade de um agente intermediário confiável para gerenciar as informações armazenadas.

1.2 OBJETIVOS

Este trabalho tem como objetivo principal desenvolver, utilizando a tecnologia *blockchain*, uma solução que permitirá, ao público de interesse, uma maneira de armazenar as informações relevantes referentes à inscrição no Conselho Regional de Medicina (CRM) dos profissionais da medicina, de modo descentralizado e confiável, dada as vantagens que a tecnologia proporciona para a manipulação e rastreabilidade desses dados. Dadas essas características, a solução proposta mostra-se relevante e útil para prover uma maior segurança e disponibilidade dos dados médicos. Além disso, apresentou-se uma arquitetura que pode ser estendida para diversos outros projetos similares em diferentes domínios. Em relação aos objetivos específicos, pretende-se:

- Realizar um levantamento dos problemas encontrados no controle dos registros médicos do CRM;
- Apresentar o funcionamento da tecnologia *Fabric* em sua estrutura fundamental como base para a resolução do problema;
- Projetar a arquitetura do sistema de modo que possam ser usadas no âmbito do problema promovendo maior confiabilidade e transparência;
- Desenvolver o sistema utilizando uma plataforma adequada para a implementação das suas funcionalidades.

- Realizar uma avaliação da solução proposta junto a um especialista para valida-la.

1.3 ESTRUTURA DO TRABALHO

O trabalho está organizado em oito capítulos, incluindo a presente Introdução. Desta forma, os demais capítulos são resumidos abaixo:

a) **Capítulo 2 - Fundamentação Teórica:** são discutidos os principais conceitos que permeiam a pesquisa desenvolvida. Primeiramente, apresenta-se os conceitos gerais sobre os conselhos regionais de medicina e as informações que devem ser manipuladas para sua divulgação. Em seguida, a tecnologia DLT é apresentada para que, logo em seguida, os fundamentos e particularidades da *blockchain* sejam explanados. A partir disso, também é realizada uma breve discussão entre os diferentes tipos de redes existentes (pública x permissionada). Posteriormente, aborda-se o funcionamento da rede permissionada *Hyperledger*.

b) **Capítulo 3 - Trabalhos Relacionados:** são analisados os principais trabalhos relacionados ao presente estudo e quais aplicações estão disponíveis no mercado utilizando *blockchain* como diferencial.

c) **Capítulo 4 - Procedimentos Metodológicos:** caracteriza-se a pesquisa metodologicamente, incluindo as etapas e métodos explorados.

d) **Capítulo 5 - Arquitetura Proposta:** apresenta a concepção proposta por este trabalho para o gerenciamento de registros médicos. Na primeira seção é mostrada uma visão geral do funcionamento da abordagem proposta. Em seguida, é descrita a modelagem adotada para a solução proposta, destacando-se os detalhes relativos como estão organizados e os elementos que a compõem e por fim na Seção seguinte, é apresentada uma demonstração de funcionamento da solução proposta.

e) **Capítulo 6 - Implementação do Sistema:** apresenta como foi implementado o sistema web desenvolvido comportar a arquitetura de *blockchain* proposta. Nesse capitulo são apresentados os atores do sistema, a descrição dos requisitos funcionais e não funcionais, os diagramas relevantes e o projeto do sistema, finalizando com as perspectivas tecnológicas utilizadas.

f) **Capítulo 7 - Demonstração do Sistema:** primeiramente apresenta uma demonstração do sistema desenvolvido, avaliando as funcionalidades e telas da aplicação.

Em seguida apresenta uma avaliação de desempenho do sistema.

g) **Capítulo 8 - Considerações Finais:** são discutidas as últimas considerações, contribuições alcançadas e limitações da pesquisa. Ao final, elenca-se oportunidades para trabalhos futuros

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão apresentados alguns conceitos teóricos ligados a este trabalho. Inicialmente, na Seção 2.1, serão mostrados os aspectos gerais à respeito das normas relacionadas ao exercício legal de profissionais da medicina e as informações relevantes para seus devidos registros nos órgãos fiscalizadores. Na Seção 2.2 será introduzido o conceito de livro-razão distribuído e como essa tecnologia pode ser utilizada para resolver problemas que envolvem a transação de uma grande quantidade de dados sem a necessidade de uma entidade centralizadora. Em seguida (Seção 2.3), será apresentada a tecnologia *blockchain* como uma forma de livro-razão distribuído, sendo feita a distinção entre os diferentes tipos de implementação dessa tecnologia (*blockchains* públicas e permissionadas). Na Seção 2.4 será apresentado a tecnologia *Hyperledger*, suas ferramentas e como estas podem ser utilizadas no desenvolvimento de *Blockchains* permissionadas. Por fim, na Seção 2.5 serão abordados os principais conceitos relativos à Aplicações Descentralizadas.

2.1 REGISTROS MÉDICOS NO CONSELHO REGIONAL DE MEDICINA

As responsabilidades éticas do profissional da medicina no Brasil são indicadas pelo Código de Ética Médica (CEM), segundo a Resolução n.1.246, de 8 de janeiro de 1988, do Conselho Federal de Medicina (CFM), como também inspiradas por documentos internacionais. No CEM, as normas redigidas tratam desde a relação e responsabilidade ética dos médicos para com os pacientes e seus familiares, como também sobre o relacionamento entre os próprios médicos e os direitos dos mesmos, publicidade e trabalhos científicos, além de uma série de disposições gerais (BRASIL, c).

Apesar de todos os deveres descritos nos regimentos, e das obrigações a serem cumpridas, principalmente em relação ao bem estar do paciente, são inúmeros os casos de ilegalidade de atuação dos profissionais da medicina, seja esse relacionado ao exercício ilegal na área, procedimentos cirúrgicos ilícitos, prescrições de receitas erradas, quebra de sigilo quanto às informações confidenciais.

Para lidar com esses tipos de impasses, os Conselhos de Medicina atuam como órgãos fiscalizadores com o intuito de realizar o devido controle tanto para com entidades físicas como jurídicas públicas e privadas a fim de alcançar o perfeito desempenho ético da Medicina em todos os aspectos possíveis, e penalizar, quando cabível, àqueles que infringirem as leis

propostas. Segundo a Resolução no 1.541/98 do CFM, no Anexo sobre o Estatuto para os Conselhos de Medicina, título I, artigo primeiro, define o Conselho Federal de Medicina e os Conselhos Regionais de Medicina (CRM) como sendo "órgãos supervisores, normatizadores, disciplinadores, fiscalizadores e julgadores da atividade profissional médica em todo o território nacional" .

Para cada tipo de conselho são dadas as devidas atribuições, e ambos formam o Conselho Pleno. A organização e funcionamento deles são estabelecidos conforme o artigo 5 do Estatuto para os Conselhos de Medicina:

> Art. 5. O Conselho Federal, com jurisdição sobre todo o território nacional, é sediado na capital da República e os Conselhos Regionais, com sede em cada capital de Estado, Território e no Distrito Federal, serão denominados de acordo com suas áreas de jurisdição. (Redação dada pela Resolução CFM nº 1.541/98 de 18 de dezembro de 1998).

Dentre uma série de atribuições incumbidas a cada CRM, pode-se destacar a responsabilidade em auxiliar no registro das informações necessárias para o exercício profissional legal de pessoa física e as atividades de pessoas jurídicas de direito público ou privado como medida protetiva à saúde da coletividade. Especificamente, os serviços oferecidos por cada CRM pode se enquadrar em cinco diferentes funções, as quais são descritas a seguir:

- **Judicante**: permite que o CRM realize o monitoramento e garantia do exercício médico da profissão. Em outras palavras, o CRM atua também juridicamente, recebendo denúncias de infração ao CEM e agindo sobre elas de acordo com os procedimentos estabelecidos nos manuais;
- **Cartorial**: deve garantir o registro do diploma do médico recém-formado para comprovar sua formação em medicina e autorizar por lei seu exercício na referida profissão. Além disso, tem-se também o registro cartorial de especialidades, no qual o médico deve cadastrar também sua especialidade de atuação;
- **Normativa**: permite que o CRM crie resoluções que estabeleçam as diretrizes para o exercício da profissão que não estejam contidas no CEM;
- **Fiscalizatória**: responsável por supervisionar as irregularidades existentes. Esta função fiscaliza serviços de saúde, hospitais, postos de saúde, clínicas e consultórios. Isso é realizado por meio de um corpo de fiscais que verificam se as atividades estão sendo feitas dentro das normas estabelecidas

- **Pedagógica**: responsável por incentivar políticas públicas, as quais estão relacionadas à discussões sobre saúde pública de qualidade, ambientes seguros para o exercício da profissão médica, dentre outros assuntos.

Os médicos, por sua vez, a fim de garantir o seu o exercício da medicina, o cumprimento, a perfeita execução do código e, consequentemente, a transparência das informações necessárias, têm o dever de realizar a sua inscrição e o registro das informações relevantes nos órgãos especializados. A Lei n 3.268/57 expressa em seu artigo 17 que:

> Art. 17. Os médicos só poderão exercer legalmente a medicina, em qualquer de seus ramos ou especialidades, após o prévio registro de seus títulos, diplomas, certificados ou cartas no Ministério da Educação e Cultura e de sua inscrição no Conselho Regional de Medicina, sob cuja jurisdição se achar o local de sua atividade. (Redação dada pela Lei n 3.268/57 de 30 de setembro de 1957).

Além disso, é dever do profissional comunicar ao Conselho Regional de Medicina os fatos de que tenham conhecimento e que qualifiquem provável violação às normas que regulam o exercício da Medicina. A Resolução no 1.541/98 do CFM, no Anexo sobre o Estatuto para os Conselhos de Medicina, título V, artigo 32, estabelece outras seguintes condições para que um médico exerça legalmente a Medicina:

> Art. 32. [...] §2 - No caso de médico estrangeiro, a inscrição será feita após cumprimento das exigências legais pertinentes. §3 - Poderão ser isentos do pagamento da anuidade, mantidos os direitos e deveres, os médicos que completarem 70 (setenta) anos naquele exercício. §4 - Nos casos em que o profissional tenha que exercer temporariamente a Medicina em outra jurisdição, este fato deverá ser comunicado por escrito ao Conselho Regional de sua jurisdição original. §5 - Se o médico inscrito no Conselho Regional de um Estado passar a exercer, de modo permanente, atividade em outra região, assim se entendendo o exercício da profissão por mais de 90 (noventa) dias na nova jurisdição, ficará obrigado a requerer a inscrição secundária no quadro respectivo, ou para ele se transferir, sujeito, em ambos os casos, à jurisdição do Conselho local pelos atos praticados em qualquer jurisdição [...]. (Redação dada pela Resolução CFM nº 1.541/98 de 18 de dezembro de 1998).

Um registro de dados realizado de maneira adequada, com as informações mencio-

nadas acima, é de grande relevância para a garantia da cultura da transparência na administração pública. Especificamente, de como encontra-se o atual estado do exercício de um profissional da Medicina afim de se evitar fraudes, ilegalidades e informações inconsistentes. Além disso, conforme dispõe o artigo 3º da Lei no 12.527/2011, é necessário "assegurar o direito fundamental de acesso à informação, que devem ser executados em conformidade com os princípios básicos da administração pública".

A Resolução CFM nº 2.180/2018 estabelece quais as informações podem ser disponibilizadas em consultas eletrônicas relativas aos registros dos médicos inscritos no sistema do Conselhos de Medicina para fins de divulgação de informações de interesse público, esses dados são apresentados a seguir: (i) nome completo; (ii) o número do CRM; (iii) Unidade da Federação; (iv) o sexo; (v) a data de inscrição; (vi) a fotografia; (vii) o endereço comercial e o telefone profissional; (viii) o tipo de inscrição; (ix) a situação da inscrição; (x) especialidades registradas e respectivas áreas de atuação e, por fim (xii) as informações sobre inscrições em outras unidades da Federação. No próprio endereço eletrônico do CFM é possível realizar a busca por médicos e obter essas informações mencionadas anteriormente. A Figura 1 apresenta o formulário de busca com os campos requeridos para efetuar a pesquisa.

Como o próprio formulário indica, quanto mais informações preenchidas, mais rápido e fácil será de encontrar o médico pelo qual busca-se encontrar. Os tipos de inscrições dos médicos podem ser:

- **Principal:** é a principal inscrição que o médico possui, podendo, no máximo, uma delas estar ativa, sendo está a responsável pelo controle das demais inscrições;
- **Secundária:** é a inscrição que o médico possui em outros Estados da Federação mantendo sua Inscrição principal ativa;
- **Provisório:** é a inscrição por medida judicial (revalidação do diploma, o registro ou a reintegração de registro);
- **Estudante médico:** visto temporário de estudo. Normalmente, é concedida a médicos estrangeiros e/ou brasileiro formados no exterior, cujo diploma ainda não foi revalidado no Brasil, e que se encontra em processo de formação no território nacional.

O campo de busca considerado como um dos mais relevantes para se ter conhecimento a respeito e, supostamente, evitar fraudes e ilegalidades, é o denominado "Situação". Este informa o estado de capacidade e possibilidade de atuação legal de um profissional da Medicina, o qual pode encontrar-se em situação "Ativa" ou "Inativa" que, por sua vez, podem

Figura 1 – Formulário de busca por médicos do endereço eletrônico do CFM

Fonte – *Website* do Conselho Federal de Medicina (CFM).

ser classificadas de acordo como apresentado nos Quadros 1 e 2.

Quadro 1 – Classificação dos tipos de situações ativas

Situação Ativa	
Regular	Médico que está regularmente inscrito no Conselho Regional de Medicina e se encontra apto ao exercício da medicina.
Inoperante	Médico que não recolhe anuidades há mais de cinco anos ou com paradeiro desconhecido.
Suspenso parcialmente	Médico suspenso parcialmente de exercer determinada atividade médica por decisão administrativa do Conselho de Medicina em decorrência de doença incapacitante.
Suspenso parcialmente (ordem judicial)	Médico suspenso parcialmente de exercer determinada atividade médica em decorrência de decisão judicial.
Interdição cautelar (parcial)	Médico interditado parcialmente de exercer determinada atividade médica em decorrência de decisão administrativa do Conselho Regional de Medicina.

Fonte – *Website* do Conselho Federal de Medicina.

Como mencionado anteriormente, tais informações são imprescindíveis para fomentar a cultura de transparência na administração pública em prol da divulgação de informações ao público de interesse e, principalmente, assegurar que informações relevantes sobre a atividade profissional de um médico estejam consistentes e de acordo com as leis e as normas

Quadro 2 – Classificação dos tipos de situações inativas

Situação Inativa	
Transferido	Médico que solicitou transferência de Conselho Regional de Medicina (CRM) de seu estado de origem para outros estados.
Aposentado	Médico com inscrição cancelada por aposentadoria.
Cancelado	Médico que teve sua inscrição cancelada por não apresentar diploma médico no CRM no prazo de 120 dias ou a pedido próprio, em decorrência de viagem ao exterior ou encerramento da atividade profissional.
Cassado	Médico apenado com cassação do exercício trabalhista em decorrência de processo ético-profissional (artigo 22, letra "e" da Lei 3.238/57) e, com sentença judicial transitada em julgado.
Falecido	Médico falecido.
Interdição cautelar (total)	Médico interditado para o exercício trabalhista por decisão administrativa do Conselho Regional/Federal de Medicina.
Suspenso total	Médico suspenso do exercício da medicina por decisão administrativa do Conselho de Medicina em decorrência de doença incapacitante.
Suspenso total (ordem judicial)	Médico suspenso do exercício da medicina em decorrência de decisão judicial.
Suspenso temporariamente	Médico suspenso por tempo determinado, de até trinta dias, do exercício da medicina por ter sido apenado em processo ético-profissional (artigo 22, letra "d" da Lei 3.268/57).

Fonte – *Website* do Conselho Federal de Medicina.

que regulamentam a profissão para seu exercício legal. Apesar de informações como estas já serem disponibilizadas eletronicamente, ainda tem-se observado variados tipos de irregularidades e ilegalidades as quais são apresentadas na mídia em geral. Fatos como estes têm causado, até mesmo, mortes à terceiros, como são os que acontecem com pessoas que se submetem à procedimentos estéticos e têm complicações sérias e fatais, por exemplo[1] [2].

Os médicos que geralmente se envolvem nesse tipo de delito, não têm o registro de seus títulos de forma a respeitar as normas estabelecidas, e acabam por falsificar seus currículos e realizarem procedimentos ilegais. Dentre os diversos casos, pode-se destacar o de Denis Furtado, médico que não estava com seus registros em conformidade com a lei, e foi preso por realizar um procedimento estético ilícito que ocasionou no óbito da paciente [3] . Portanto, faz-se necessária uma rigorosa fiscalização do exercício profissional dos médicos por funcionários exclusivos para o setor e com capacitação específica (BRASIL, e).

[1] Notícia fornecida por G1 AL Disponível em: <https://g1.globo.com/al/alagoas/noticia/2018/08/22/mp-al-denuncia-denuncia-tres-pessoas-por-exercicio-ilegal-da-medicina-estelionato-e-falsidade-ideologica.ghtml>. Acesso em: 16 jan 2019.

[2] Notícia fornecida pelo Jornal da EPTV 2ª Edição. Disponível em: <https://g1.globo.com/sp/ribeirao-preto-franca/noticia/vereador-vira-reu-em-processo-por-exercicio-ilegal-da-medicina-em-ribeirao-preto-sp.ghtml>. Acesso em: 15 jan 2019.

[3] Notícia fornecida pelo site da BBC em: <https://www.bbc.com/portuguese/brasil-44901624>. Acesso em: 15 de janeiro de 2019.

2.2 LIVRO-RAZÃO DISTRIBUÍDO

Um livro-razão distribuído é um banco de dados distribuído por vários nós ou dispositivos de computação[4]. Cada nó replica e salva uma cópia idêntica do livro-razão e pode atualizar-se de forma independente. A característica inovadora da tecnologia de livro-razão distribuída é que este não é mantido por nenhuma autoridade central. Logo, as atualizações para o livro-razão são construídas independentemente e registradas por cada nó. Os nós, por sua vez, votam nessas atualizações para garantir que a maioria concorde com a conclusão alcançada. Essa votação e acordo em uma cópia do livro-razão é chamada de consenso e é conduzida automaticamente por um **Algoritmo de Consenso** definido. Uma vez alcançado o consenso, o livro-razão distribuído é atualizado e a última versão acordada do livro-razão é salva em cada nó, separadamente.

As tecnologias de livro-razão distribuído reduzem drasticamente o custo financeiro uma vez que sua arquitetura pode ajudar a mitigar a dependência de bancos, governos, advogados, notários e funcionários responsáveis pela conformidade regulatória. Um exemplo disso é o Corda (HEARN, 2016), um projeto de código aberto desenvolvido pelo consórcio R3, onde é mantido um livro-razão distribuído que gerencia e executa contratos financeiros entre mais de 75 das maiores instituições financeiras do mundo.

Portanto, o conceito de livro-razão distribuído pode ser visto como um paradigma que trata como as informações são coletadas e comunicadas, de forma a revolucionar o modo como indivíduos, empresas e governos realizam transações uma vez que as partes envolvidas em uma transação possuam a mesma visão da verdade, assegurada através de regras de consenso definidas pelo próprio modelo de negócio.

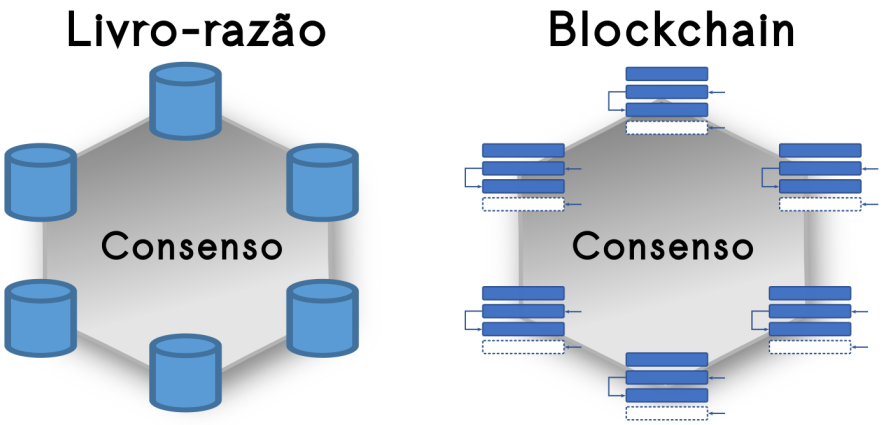
2.3 BLOCKCHAIN

O conceito do *blockchain* foi introduzido em 2008, por um grande motivo: afastar-se de um sistema centralizado onde as instituições financeiras, o estado e os reguladores não eram confiáveis (NORTON, 2016). Uma *blockchain* trata-se de um tipo de livro-razão distribuído gerenciado por redes *peer-to-peer* (P2P). Como é um livro-razão distribuído, ele pode existir sem uma autoridade centralizada ou servidor que o gerencie. A segurança de seus dados pode ser mantida pela replicação do banco de dados e assegurada pelas regras de consenso. Entretanto,

[4] Notícia fornecida por TOWARDSDATASCIENCE Disponível em: <https://towardsdatascience.com/ the-difference-between-blockchains-distributed-ledger-technology-42715a0fa92>. Acesso em: 16 jan 2019.

enquanto em um livro-razão os dados são apenas mantidos em cada nó por meio de um banco de dados, na *blockchain* os dados são agrupados e organizados em blocos, sendo estes ligados entre si e protegidos por meio de criptografia. Essa comparação pode ser vista na Figura 2.

Figura 2 – Livro-razão x *Blockchain*



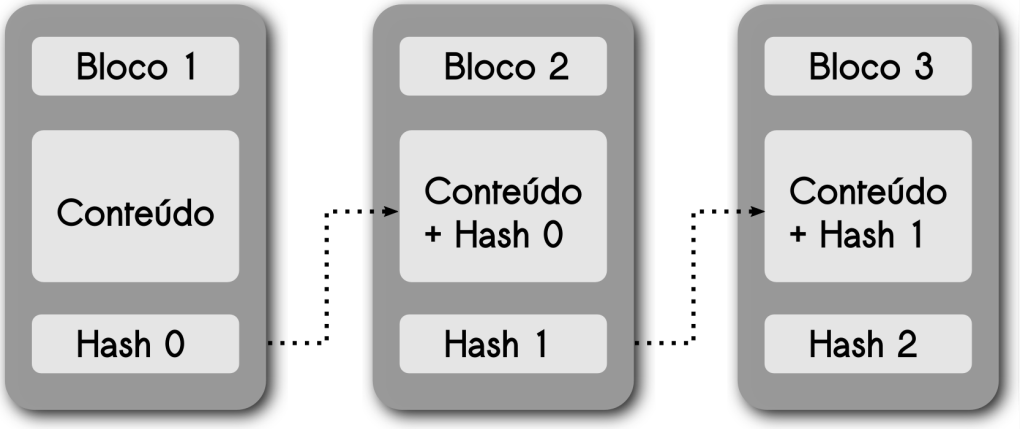
Fonte – Elaborado pelo autor.

Em uma *blockchain*, existe um algoritmo responsável por mapear todo o conteúdo armazenado em cada bloco para uma sequência de bits de comprimento fixo denominado *hash*. O *hash* responsável por representar o bloco anterior sempre será utilizado na construção do *hash* do atual bloco, como pode ser visto na Figura 3. Assim, o valor do *hash* anterior sempre será armazenado no bloco atual e, por conta disso, é possível rastrear qualquer bloco anterior na *blockchain*. Eventualmente, pode-se chegar até o bloco **gênese**, que é o nome dado ao primeiro bloco criado na blockchain, e que provê um ponto de partida para que a mesma possa ser construída (ANTONOPOULOS, 2014).

A tecnologia *blockchain*, portanto, pode ser utilizada para uma série de situações que possam demandar registro de eventos, gerenciamento de registros, processamento de transações, rastreamento de ativos de qualquer espécie, dentre outras aplicações. Especificamente, pode-se citar uma das aplicabilidades para a qual essa tecnologia tem auxiliado, que é o sistema de votação (LEE KIBIN JAMES; KIM, 2016). Por possuir uma estrutura somente de anexação, os dados sempre serão adicionados ao banco de dados, e é impossível alterar ou excluir dados inseridos nos blocos anteriores.

Quando projeta-se uma blockchain, é necessário levar em consideração diversos pontos relacionados a sua organização, por exemplo, o intuito da rede, se a rede projetada será de propósito geral ou específico assim como também a forma que a rede deve lidar com o processo

Figura 3 – Representação de um conjunto de blocos interligados pelas referências de *hash* do bloco anterior.



Fonte – Adaptado de (SOUSA et al., 2018).

de encadeamento dos blocos nos diversos nós que mantém a rede sem perda de integridade. Nas próximas subseções, esses pontos serão melhor discutidos.

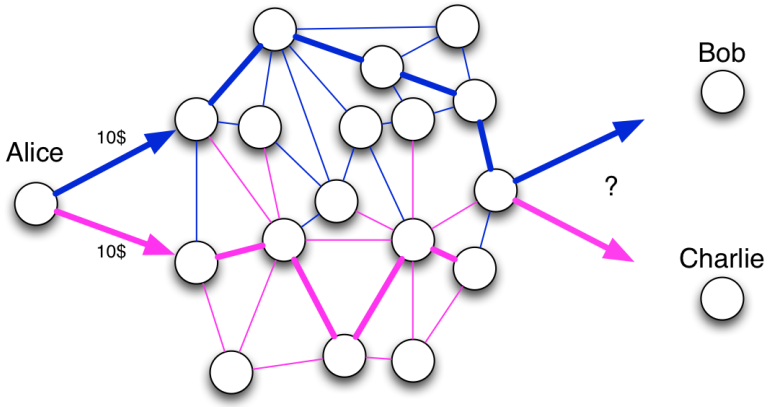
2.3.1 Algoritmos de Consenso

A *blockchain* permite manter as informações íntegras, porém, assim como em qualquer outro sistema distribuído, a *blockchain* possui o problema de resolução de conflitos. Ou seja, se dois fatos incompatíveis chegarem no mesmo instante, o sistema deve possuir regras que determinem qual dos fatos será considerado válido. A Figura 4 exemplifica o problema de resolução de conflitos onde Alice envia dez dólares para Bob e os mesmos dez dólares para Charlie. O problema está no fato de que Alice possui somente dez dólares e está tentando gastar duas vezes este mesmo valor. Este problema é chamado **Gasto Duplo** (HOEPMAN, 2015).

Uma maneira de resolver este problema é ordenando os eventos em que, o primeiro que for registrado, é o vencedor. Porém, ambos os eventos podem aparecer em ordens diferentes em nós distantes um do outro. Para que toda a rede concorde na ordem dos fatos e preserve sua integridade é necessário um sistema de sincronização de dados que, na literatura, é conhecido como Algoritmo de Consenso (MENDANHA et al., 2016).

Segundo Schwartz et al. (2014), existem diversos algoritmos de consenso, embora cada um possa ser mais adequado do que outro para determinado sistema distribuído, a depender dos requisitos, eles devem ser robustos o suficiente para tolerar falhas como as descritas no "Problema dos Generais Bizantinos" (GMYTRASIEWICZ; DURFEE, Alpharetta, USA). Por

Figura 4 – Problema do gasto duplo.



Fonte – Zaninotto (2016)

exemplo, segundo GARAY et al.(2015) a ideia por trás de consenso *Proof-of-work* (PoW) é limitar a taxa de criação de novos blocos resolvendo um quebra-cabeça criptográfico, ou seja, ao executar, a CPU leva um certo tempo para resolver o quebra-cabeça. Sendo assim, para que um nó possa anexar um bloco na blockchain, ele é forçado a encontrar um *hash* criptográfico N de maneira que esse *hash* tenha um valor inferior que um determinado valor limite L. O primeiro nó que apresenta essa solução tem seu bloco anexado ao livro-razão. Assim, desde que um participante mal-intencionado não controle mais da metade do poder computacional total presente na rede, ou seja, maior que 50%, o consenso PoW impede que esse participante crie novos blocos mais rapidamente do que os demais participantes da rede. Alguns outros algoritmos estão sendo mostrados no Quadro 3.

Quadro 3 – Exemplos de algoritmos de consenso

Algoritmos de Consenso	
Proof of Stake (PoS)	Usa um processo de eleição pseudo-aleatória para selecionar o nó que será o validador do próximo bloco. Essa eleição é baseada no tempo de posse da moeda e da riqueza do nó.
Proof of Activity (PoA)	Abordagem híbrida que combina a Prova de Trabalho PoW e PoS, a mineração começa da mesma forma que a prova de trabalho tradicional porém a validação do bloco é feita por um grupo aleatório de validadores. Quanto mais moedas no sistema possui um validador, maior é a chance de ser escolhido.
Proof of Importance (PoI)	Incentiva a participação ativa na rede atribuindo ao nó uma pontuação de importância, o que determina com que frequência esse nó pode validar blocos
Proof of Capacity (PoC)	Segue o princípio do PoW, porém permite que os dispositivos de mineração da rede usem o espaço disponível nos seus discos rígidos para a mineração de criptomoeda. Quanto maior for a capacidade do disco rígido, mais valores podem ser armazenados aumentando assim as chances do minerador de encontrar na sua lista o valor de hash necessário.

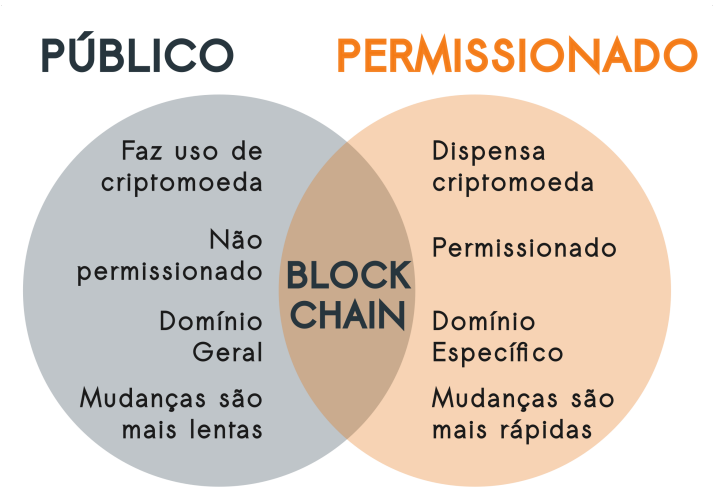
Fonte – Elaborada pelo Autor

2.3.2 Blockchains Públicas e Permissionadas

As diversas possibilidades de modernização de processos tradicionais via *blockchain* têm feito surgir uma grande variedade de plataformas e *tokens* que focam em finalidades que abrangem vários domínios. Portanto, para se adequar aos tipos de sistemas que têm sido desenvolvidos, podem-se usar *blockchains* públicas e/ou permissionadas, a depender da necessidade de aplicação. Segundo (VICTORIA, 2016) as *blockchains* públicas são as mais populares devido ao grande interesse no desenvolvimento de aplicações de código aberto, entretanto as blockchain permissionadas destacam-se por fornece uma maneira de proteger as interações entre um grupo de entidades que têm um objetivo comum, mas que não confiam totalmente umas nas outras, como empresas que trocam bens ou informações (ANDROULAKI et al., 2018).

Diferentemente de uma blockchain pública, na qual qualquer um pode participar da rede para visualizar e realizar transações, em uma *blockchain* permissionada é preciso ter permissão da instituição ou grupo de instituições responsável pela rede. Dessa forma, em uma indústria de alimentos, por exemplo, os participante incluídos na rede permissionada poderiam ser: produtores, distribuidores, empresas de logística e, até mesmo, o supermercado para acompanhar todo o caminho de seus produtos, desde a origem até o destino definido.

Figura 5 – Características da *blockchain* pública x permissionada



Fonte – Elaborado pelo Autor.

Como é mostrado na Figura 5, as redes públicas utilizam mecanismos como moedas virtuais chamadas criptomoedas, para incentivar a rede. As transações na *blockchain* são valida-

das por membros da rede denominados de “mineradores” os quais são pagos, em criptomoedas, de acordo com as taxas definidas na rede (NARAYANAN et al., 2016). Esse tipo de esquema de validação para transações não é necessário para o caso de um uma rede permissionada. Em outras palavras, não há necessidade de incentivar a rede usando criptomoedas, pois a própria rede permite que os participantes decidam quem serão os validadores e que tipo de políticas e regras devem ser usadas para validar a transação.

Outro diferencial a ser considerado encontra-se no fato de que as *blockchains* públicas têm o benefício de permitir que o livro-razão seja gerenciado de forma totalmente descentralizada, ou seja, qualquer nó disposto a manter uma cópia do livro-razão pode tentar criar novos blocos para isso. Porém, o esforço computacional associado ao algoritmo de consenso é demorado, mesmo que o hardware especializado seja usado no processo de validação do *blockchain*, este mecanismo ainda impõe um limite na latência das transações tornando-as bastante lentas [5].

Por outro lado, as *blockchains* permissionadas possuem um conjunto de nós encarregados por criar novos blocos e realizar a execução de protocolos de consenso para decidir a ordem pela qual os blocos são inseridos no livro-razão (ANDROULAKI et al., 2018; BUCHMAN, 2016; MARTINO, 2016). Portanto, *blockchains* permissionadas não utilizam a quantidade de recursos que *blockchains* públicas despendem, tornando-as assim uma melhor alternativa quando buscasse priorizar o desempenho da rede. Além disso, em *blockchains* permissionadas é possível controlar o conjunto de participantes encarregados por manter o livro-razão, o que pode fazer esse tipo de *blockchain* uma solução mais atraente para grandes corporações.

2.4 HYPERLEDGER

A plataforma *Hyperledger* trata-se do resultado de um esforço colaborativo entre indústrias de diversas áreas para criar uma *blockchain* permissionada de código aberto. A plataforma incorpora tecnologias, incluindo *frameworks*, *smart contract*, interfaces gráficas e amostras de aplicações. Atualmente o projeto conta com 130 membros e 5 *frameworks*, *hyperledger Burrow*, *Fabric*, *Iroha*, *Sawtooth* e *Indy*. Alguns exemplos de empresas envolvidas são a Cisco, IBM, Intel e também empresas financeiras como: ANZ Bank, CLS Group e CME Group. De maneira geral seu objetivo é promover a adoção em massa da tecnologia *blockchain*, reutilizando recursos em comum para acelerar processo de inovação (BLUMMER; BOHAN,

[5] Notícia fornecida por ITFORUM AL Disponível em: <https://www.itforum365.com.br/gestao/ smartchain-blockchain-permissionado/>. Acesso em: 16 jan 2019.

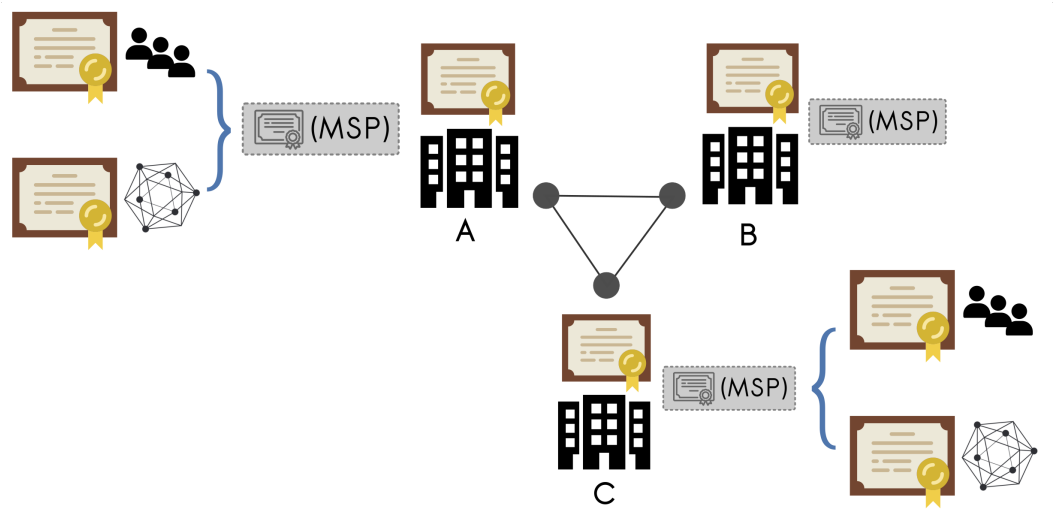
2018).

Segundo os membros do projeto Blummer e Bohan (2018), as *blockchains* públicas seriam incapazes de resolver problemas como a escalabilidade e falta de suporte para transações privadas. Vale destacar que o *Hyperledger* tem o foco na indústria, mais especificamente nas relações *Business-to-business* (B2B) e *Business-to-consumer* (B2C). Em geral, empresas têm parceiros B2B, como por exemplo, fornecedores de matéria-prima ou compradores de mercadorias.

Isso significa que, diferentemente de redes públicas, para ter-se acesso aos sistemas baseados em *blockchains* construídas em *Hyperledger* é necessário que possua uma credencial. Um provedor de serviço de associação (do inglês *Membership Service Provider*, ou MSP) trata de gerar essas credenciais para os diversos tipos de participantes. Uma aplicação define papéis que são atribuídos aos participantes, e o acesso é concedido ou restrito por meio desses papéis.

As credenciais da rede *Hyperlegder* são atribuídas por meio de um certificado do tipo X509, como mostrado na Figura 6. Cada organização recebe um certificado e, dependendo de seu nível de autoridade, pode usar uma MSP para emitir certificados para todos os componentes e participantes da infraestrutura em sua organização.

Figura 6 – Estrutura de emissão de certificados para a rede *Hyperledger*



Fonte – Elaborado pelo autor.

Assim, uma organização A que recebe o certificado da rede *Hyperledger*, e através dele pode gerar certificados para todos seus membros, de acordo com o nível de importância deles para a empresa. Dessa forma, um funcionário comum não terá acesso a recursos importantes da empresa como o controle de estoque, por exemplo. Além disso, é possível que a organização

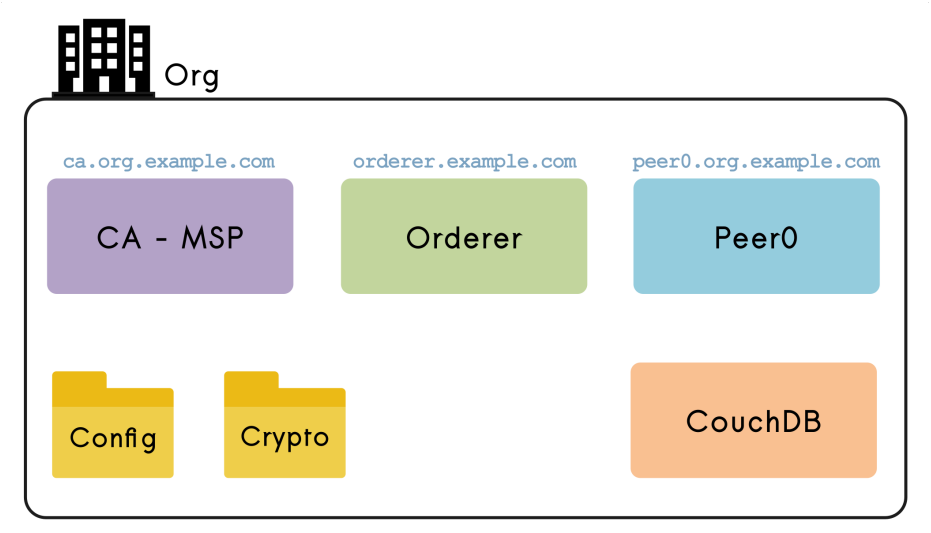
autorize sub-redes dentro da rede principal por meio do certificado emitido permitindo que a organização tenha um controle de quais outras organizações (B ou C) podem acessar aquela informação.

Por fim, o *Hyperledger* em si pode ser utilizado por meio de três ferramentas principais, *Fabric*, *Composer* e *Playground*. Sendo essas responsáveis por criar a rede, definir o comportamento logico da rede e testá-la, respectivamente. Nas próximas subseções será discutido mais detalhadamente o propósito de cada uma dessas ferramentas.

2.4.1 Hyperledger Fabric

Dentre os cinco *frameworks* apresentados na seção anterior, Androulaki et al. (2018) afirma que o que mais chama atenção é o *hyperledger fabric*, pois este já foi usado no desenvolvimento de mais de 400 diferentes protótipos como sistemas para logística comercial, compensação cambial, segurança alimentar, gestão de contratos, negociação de títulos e gerenciamento de identidade. O *Hyperledger Fabric* foi projetado para o desenvolvimento de aplicações com uma arquitetura modular, permitindo que serviços sejam associados utilizando o método *plug-and-play* (FERREIRA, 2016). De acordo com Blummer e Bohan (2018) é um projeto de código aberto desenvolvido dentro do espaço colaborativo do *Hyperledger*. Trata-se de um sistema de *blockchain* permissionada projetada para dar suporte à implementações de diferentes componentes, tais como mecanismos de consenso e serviços de associação para os membros.

Figura 7 – Estrutura do *Hyperledger Fabric*



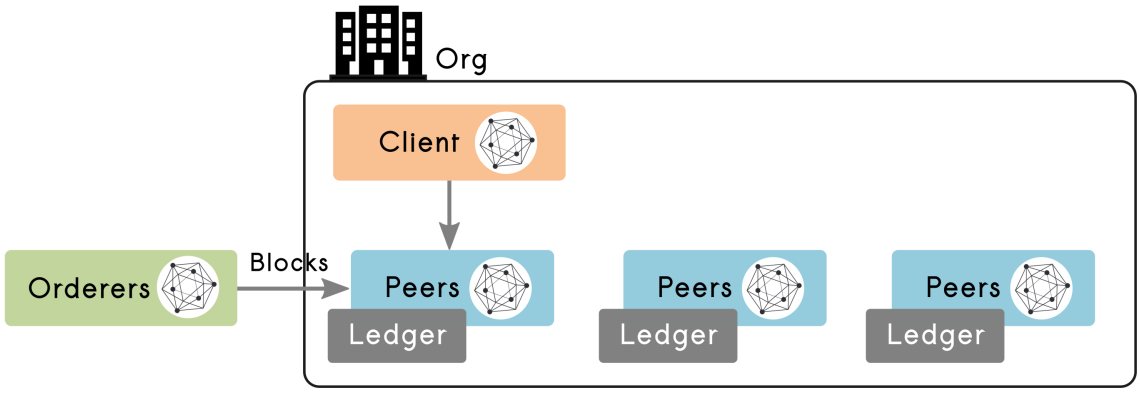
Fonte – Elaborado pelo autor.

Na Figura 7, é apresentada a configuração do ambiente no qual são definidos os vários contêineres que compõem os componentes da infraestrutura para a organização. Existem, predominantemente, quatro contêineres que são instanciados ou criados como parte do desenvolvimento do ambiente. O primeiro e segundo são os contêineres MSP e *Orderer*, responsáveis pelo serviço de associação de novos membros e o serviço de criação de novos blocos, respectivamente. O terceiro é o contêiner *Peer* onde fica o livro-razão e depende do quarto contêiner **CouchDB** para guardar os dados de estado. A configuração de todos esses contêineres é gerenciada em dois diretórios. O diretório ***crypto*** mantém todos os arquivos criptografados e o diretório ***config*** contém os arquivos de configuração para todos os contêineres.

De acordo com Blummer e Bohan (2018), o *Hyperledger Fabric* permite que os clientes gerenciem transações usando ***Chaincodes***, ***Order services*** e ***Peers***.

- ***Chaincode:*** consiste em um código escrito em Go, node.js ou Java implantado na rede do *Fabric*. Normalmente lida com a lógica de negócios acordada pelos membros da rede, onde é executado e validado. Assim, essa tecnologia pode ser vista como uma contraparte aos contratos inteligentes utilizados em *blockchains* públicas.
- ***Ordering Service:*** é o serviço encarregado de criar blocos para o livro-razão distribuído, bem como a ordem pela qual cada bloco é anexado ao livro-razão.
- ***Peer:*** responsável por receber atualizações de estado na forma de blocos do *Ordering Service* e manter o livro-razão.

Figura 8 – Relação *orderer* x *peers*



Fonte – Elaborado pelo autor.

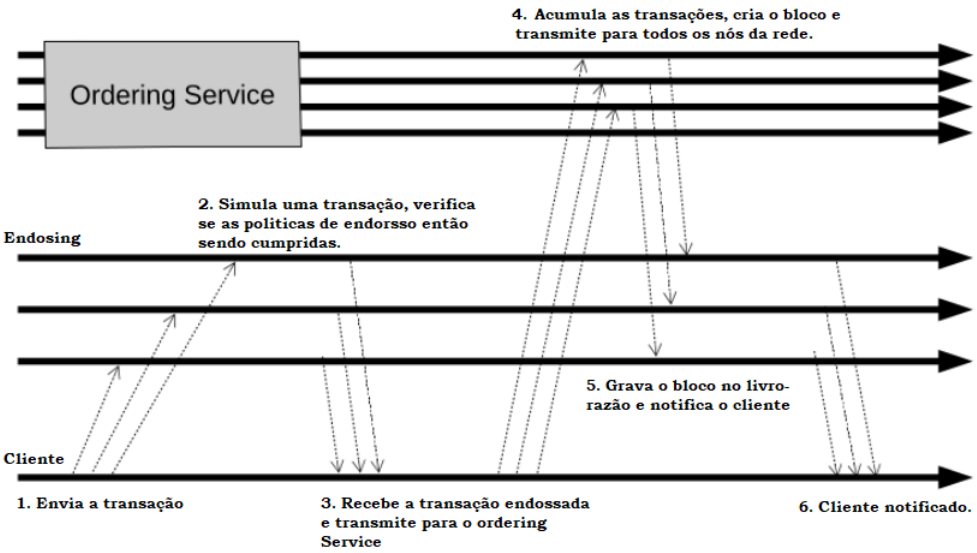
A Figura 8 ilustra a relação entre *orderers* e *peers*. O cliente representa a entidade que age em nome de um usuário final. Ele pode conectar-se a qualquer nó de sua escolha para comunicar com a *blockchain* e criar transações. Diferentemente da arquitetura das redes

não-permissionadas, os nós na rede *Hyperledger* não são homogêneos, podendo variar de acordo com funcionalidades específicas. Os mesmos podem ser classificados em dois tipos:

- ***Ancho:*** responsável por receber o bloco enviado pelo *Ordering Service*, ele atualiza os outros nós membros. Para evitar que haja um único ponto de falha, uma organização pode criar vários nós *Ancho* gerando vários pontos de ancoragem que são, por padrão, detectáveis pelo *Ordering Service*.
- ***Endorsing:*** responsável por avaliar o *Chaincode*, manter o livro-razão e o banco de dados (modelado como armazenamento de chave/valor). O principal objetivo do *Endorsing* é proteger a rede. Isso não significa apenas proteger contra ataques mal-intencionados na rede, mas também significa que ele precisa proteger a rede de um nó mal configurado. Como cada organização é responsável por configurar seus próprios nós, existe a possibilidade de que um nó mal configurado possa ser adicionado inadvertidamente à rede.

Em relação ao algoritmo de consenso, a forma como os blocos são criados ocorre diferentemente de como apresentado nas redes públicas. Em uma rede permissionada, como o Fabric, é necessário que a transação seja validado por $N/2+1$ nós *Endorsing*, onde N é o número de nós *Endorsing* presentes na rede. Caso uma transação seja considerada inválida pela maioria dos nós, ela é descartada. Segundo (CACHIN, 2016) o consenso no *Hyperledger Fabric* é dividido em três fases: Endorsso, Ordenação e Validação. Cada uma dessas etapas pode ser vista na Figura 9.

Figura 9 – Fluxo de uma transação no Hyperledger Fabric



Fonte – Elaborado pelo autor.

Na primeira fase denominada **endosso**, a transação é enviada pelo cliente para os nós *Endorsing*. Neles, a transação é simulada e avaliada por políticas de validação definidas pela rede. Caso a transação seja aceita pelos $N/2+1$ nós *Endorsing*, ela é considerada uma transação endossada e pode ser enviada pelo cliente para o *Ordering Services*. Na etapa de ordenação das transações, ocorre a criação dos bloco com as transações enviadas e o envio deste bloco para todos os nós *Endorsing*. Por fim na etapa validação, uma verificação de **Gasto Duplo** é feita por todo o bloco. Caso nada errado ocorra nessa etapa, o bloco é adicionado na *Blockchain* e o cliente é notificado.

2.4.2 Hyperledger Composer

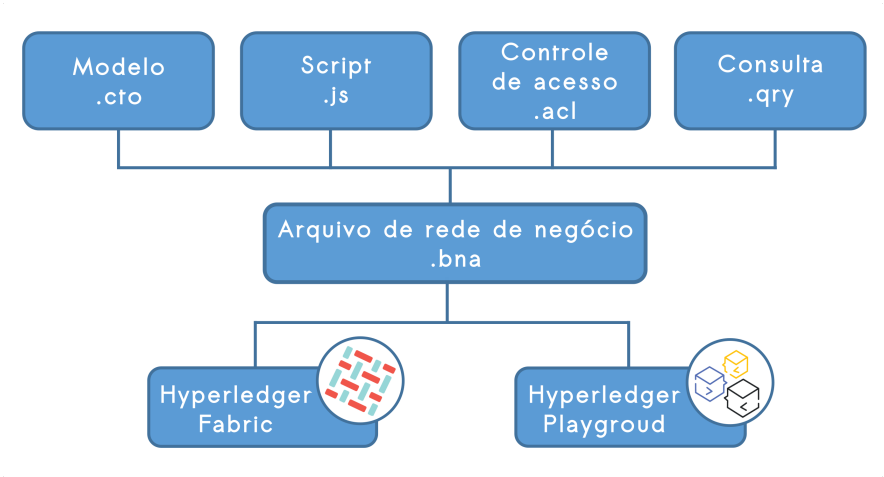
O *Hyperledger Composer* é um extenso conjunto de ferramentas desenvolvido para facilitar a criação de aplicativos utilizando a tecnologia *Hyperledger* e suportar a infra-estrutura existente no *Fabric* (BLUMMER; BOHAN, 2018). Logo, o objetivo principal desse projeto é facilitar a integração da *blockchain* aos sistema sem desenvolvimento. O *Composer* pode ser usado para desenvolver rapidamente a modelagem de casos de uso, além de permitir a integração da *blockchain* com sistemas e dados existentes, como aplicativos web.

Sendo assim, pode-se usar o *Composer* para modelar rapidamente uma rede comercial, contendo os serviços e/ou bens existentes e as transações relacionadas a eles. Como parte do modelo da rede do negócio, definem-se as transações que podem interagir com os serviços. Redes de negócios também incluem os participantes que interagem com a *blockchain*, cada um é associado a uma identidade única. O *Composer* também suporta protocolos de consenso para garantir que as transações sejam validadas de acordo com a política definida pelos participantes da rede de negócios designada. A Figura 10 apresenta a estrutura de um projeto criado no *Composer*.

Em suma, a estrutura do projeto pode ser dividida nos quatro módulos descritos abaixo:

- **Arquivo de Modelo:** para criação do modelo é utilizada uma linguagem de modelagem definida pelo próprio *Composer*. Essa linguagem foi feita para ser simples e de fácil entendimento, pois a intenção é que até mesmo um analista de negócios seja capaz de criar o modelo de domínio para uma aplicação empresarial. Um modelo de domínio criado com essa linguagem define a representação de diversos recursos, como os participantes, os ativos com os quais esses participantes devem lidar, as transações que são realizadas pelos participantes nos ativos e os eventos que são emitidos como resultado dessas transações.

Figura 10 – Estrutura de um projeto no *Composer*



Fonte – Elaborado pelo autor.

Os modelos são definidos em arquivos com uma extensão *.cto*.

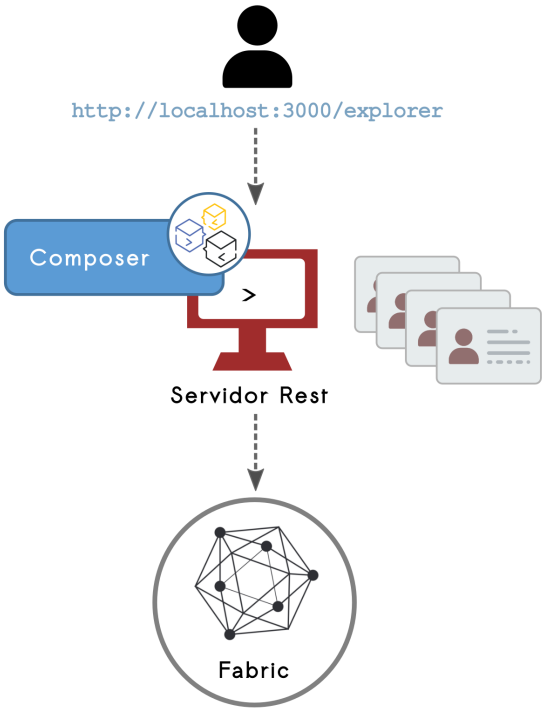
- **Arquivo de Script:** o modelo que define todas as transações que podem ser executadas pelos participantes. A lógica de cada transação é codificada em um ou mais *scripts* na linguagem Javascript. Anotações são usadas para conectar funções do código ao modelo.
- **Arquivo de Consulta:** os recursos utilizados são gerenciados por registros que podem ser consultados em tempo de execução usando uma linguagem de consulta do *Composer* semelhante ao SQL. Todas as possíveis consultas são definidas em um único arquivo com a extensão *.qry*.
- **Controle de Acesso:** é possível criar regras de controle para os vários recursos da aplicação através da linguagem de controle de acesso definida pelo *Composer*. Essa linguagem tem uma sintaxe semelhante ao do JSON. Logo para se definir uma regra, é necessário usar uma palavra-chave seguida do nome da regra e das chaves entre as quais fornecer os vários elementos da regra. Assim, cada recurso possuirá uma regra associada a ele definida em um arquivo com uma extensão *.acl*.

Os quatro módulos apresentados são integrados em um arquivo final com a extensão *.bna*. Este arquivo será a entrada do *Fabric* ou do *Hyperledger Composer Playground* apresentado na subseção 2.4.3.

Por fim, com relação a realização das transações nos recursos implantados no *Hyperledger Fabric*, o *Composer* disponibiliza uma API REST para manipular as transações. A API é projetada para usar verbos do HTTP específicos para operar os recursos, dependendo do tipo de operação. O verbo **GET** é usado para ler o estado do recurso. **PUT** é usado para atualizar o

estado do recurso. **POST** para criar um novo recurso e **DELETE** é usado para uma operação que excluirá o recurso. Para que um usuário possa interagir em tempo de execução com o *Fabric* usando o servidor REST é necessário um *business network card* que encapsula as informações de todas as credenciais, chaves e certificados e os perfis de conexão. Um usuário pode ter vários *business network card* configurados no servidor por meio de uma carteira. O gerenciamento desses cartões pode ser feito por meio do *Composer* em tempo de execução usando a RAM, banco de dados ou qualquer tipo de armazenamento em nuvem, por exemplo. Logo, ao executar a API REST, um cartão é associado a um usuário. Como a API atende a vários usuários, cada um terá sua própria carteira. A Figura 11 representa o processo citado ao demonstrar um usuário interagindo com o servidor rest via requisições http.

Figura 11 – Estrutura de servidor REST usados em um projeto no Composer

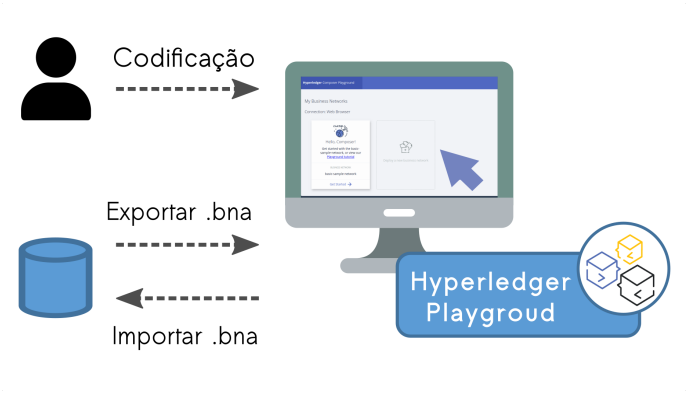


Fonte – Elaborado pelo autor.

2.4.3 Hyperledger Composer Playground

O *Hyperledger Composer Playground* é um ambiente de testes web, que fornece uma interface de usuário para configuração, implementação e teste de uma *blockchain* permissionada. Os recursos do *Playground* permitem que os usuários gerenciem a segurança da rede de negócios, convidem os participantes para redes e conectem-se a várias outras redes.

Figura 12 – Interface do *Hyperledger Composer Playground*

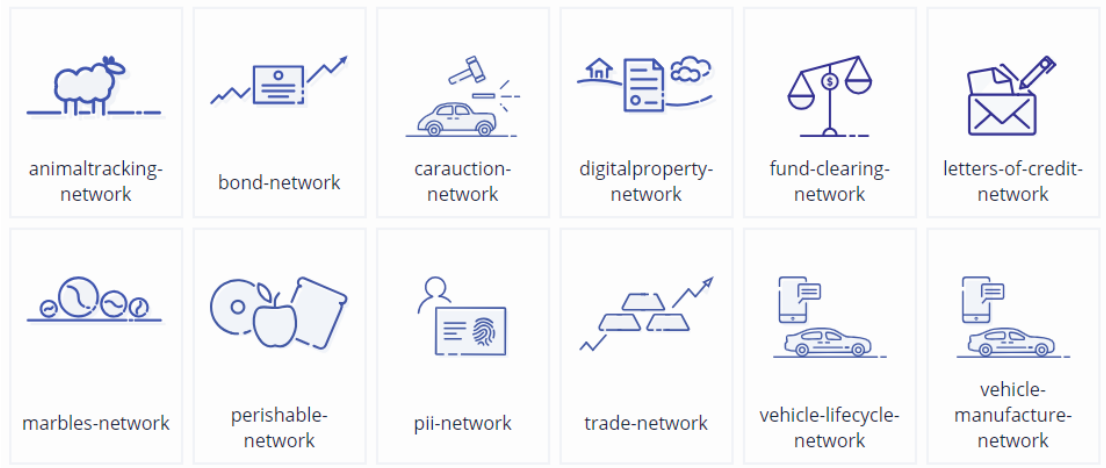


Fonte – Elaborado pelo autor.

Como ilustrado na Figura 12, os desenvolvedores podem codificar as transações usando a interface do *Playground*. A plataforma não só codifica as transações e atualizam o modelo, mas também podem exportar o modelo atualizado para o armazenamento local usando o recurso Exportar do *Playground*. Este arquivo pode ser armazenado no sistema de controle de código ou os desenvolvedores podem fazer alterações no modelo e o código e, em seguida, usar o recurso de implantação para colocá-lo de volta no *Playground* para testes.

Há também diversos exemplos disponíveis na plataforma que contemplam vários aspectos da definição de uma rede de negócios. Assim, o usuário pode testar exemplos pré-definidos, criando ou definindo o modelo através da interface de usuário. A Figura 13 apresenta os exemplos disponíveis.

Figura 13 – Exemplos disponíveis no *Hyperledger Composer Playground*



Fonte – *Website* do Hyperledger Composer.

2.5 USO DA TECNOLOGIA BLOCKCHAIN EM APLICAÇÕES

A maioria das aplicações conhecidas atualmente são consideradas centralizadas, ou seja, existe uma relação cliente-servidor onde as aplicações são controlados por uma única entidade que guarda e fornece todos os dados, gerando assim uma dependência dele para utilizar o *software*. O *Ethereum* começou como uma maneira de fazer um blockchain de propósito geral que poderia ser programado para uma variedade de usos. Mas rapidamente, a visão da Ethereum se expandiu para se tornar uma plataforma para programar aplicações descentralizadas (em inglês Decentralized Applications ou DApps). Mais amplamente, um DApp é um aplicativo da Web que é construído sobre serviços de infraestrutura abertos, descentralizados e P2P, possuindo as seguintes características:

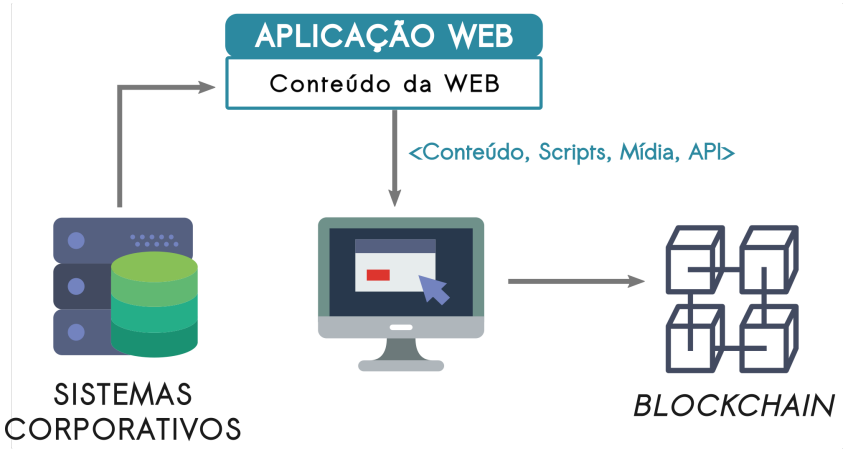
- Possuem código aberto, de forma que sua segurança pode ser auditada, e operam de forma autônoma, sem uma autoridade em particular no controle;
- Qualquer alteração dos dados somente é executada depois que se atinge um consenso entre os participantes da rede;
- Protocolos e informações são armazenadas em um *blockchain*, protegidos por criptografia e acessíveis pela rede descentralizada;
- Geram criptomoedas para mineradores e para quem transaciona na rede.

Segundo Porru et al. (2017), os investimentos em capital de risco em startups de blockchain têm aumentado constantemente, de US $93,8 milhões em 2013 para US$ 315 milhões em 2014 e US$ 490 milhões em 2015. Portanto os DApps passaram a representar uma perspectiva mais ampla do que simples contratos inteligentes, pois no mínimo, possuem um contrato inteligente e uma interface de usuário na web. Para entender mais sobre os DApps, o *whitepaper* da Ethereum (WOOD, 2017), norteia as principais categorias de DApps:

- **Aplicações financeiras:** Sistemas que permitem aos usuários manipular seus recursos existentes na blockchain.
- **Semi-Aplicações Financeiras:** Sistemas onde uma das partes envolve dinheiro, porém a outra não envolve, como exemplo, uma prestação de serviços.
- **Outros:** Sistemas com fins não financeiros como votações e governança descentralizada.

A Figura 14 apresenta a estrutura básica de um DApp. Nela o *frontend* é desenvolvido com as tecnologias tradicionais, como angular, HTML, JS e CSS, e é carregado como um aplicativo da web tradicional que pode receber scripts, arquivos de mídia, e tem acesso a APIs da mesma maneira que as aplicações centralizadas. Porém as transações feitas pelo *backend* dos

Figura 14 – Exemplo de estrutura de um DApp



Fonte – Elaborada pelo autor.

Dapps são transmitidas para a *Blockchain* por meio de APIs especializadas, o que difere das aplicações centralizadas, onde as transações podem ser armazenada no banco de dados de um servidor central.

Por fim, segundo Antonopoulos e Wood (2004), o conceito de DApps pode levar a internet a um próximo estágio natural de evolução, introduzindo a descentralização com protocolos P2P em todos os aspectos de uma aplicação web e gerando, ainda de acordo com Wood, uma nova versão da internet onde o foco está em aplicativos web baseados em protocolos descentralizados.

2.6 CONCLUSÕES DO CAPÍTULO

O presente capítulo apresentou uma fundamentação teórica sobre os principais temas que dirigem esse trabalho. Acredita-se que tais discussões irão propiciar uma melhor compreensão da proposta que será apresentada adiante.

Primeiramente, foi realizada uma breve discussão sobre o cadastro de registros médico para, logo em seguida, apresentar os conceito de livro-razão, mostrando de uma forma geral como livros-razões distribuídos são utilizados e como essa tecnologia pode ser útil para resolver diversos problemas. Na seção 2.3, foi apresentado a ideia de blockchain como uma forma de livro-razão distribuído, sua importância e os diferentes tipos (público x permissionado). A tecnologia *Hyperledger* também foi discutida na Seção 2.4, mostrando sua estrutura e ferramentas, assim como sua importância no desenvolvimento de *blockchains* permissionados. Por fim alguns conceitos sobre Aplicações Descentralizadas foram apresentados nas Seção 2.5.

3 TRABALHOS RELACIONADOS

Nesta seção, serão apresentados trabalhos relacionados à presente pesquisa. Primeiramente, serão citadas pesquisas diretamente relacionados aos conceitos de *blockchain* e a tecnologia *Hyperledger* para, em seguida, contextualizar diversos aplicativos disponíveis no mercado que utilizam *blockchain* como uma solução para resolver problemas encontrados na sociedade.

3.1 ARTIGOS CIENTÍFICOS

3.1.1 Blockchains

O conceito de *blockchain* foi originalmente introduzido pelo Bitcoin para resolver o problema dos gastos duplos associados à criptomoeda em redes P2P públicas (NAKAMOTO, 2008). Desde o início do Bitcoin e sua adoção generalizada, outras plataformas baseadas em *blockchain* PoW surgiram. Dentre estas redes a Ethereum apresentada em (WOOD, 2017), destacam-se por seu apoio ao uso de contratos inteligentes.

Blockchains são principalmente utilizadas na área financeira em grande parte como tecnologia base para definição de criptomoedas para criptomoedas. Como apresentado por Underwood (2016), *blockchains* têm sido estudadas e aplicadas dentre outros campos, na área financeira, comercial e desenvolvimento de políticas de transparência. McConaghy et al. (2016) apresentam uma proposta para gestão de grandes bases de dados distribuídas incorporando características intrínsecas a *blockchains* à estes sistemas. Em Hammer-Lahav e Hardt (2016) os autores analisam a aplicação combinada de *blockchain*, aplicações de economia compartilhada e Internet das Coisas. Axon (2015) explora o uso de blockchain para apresentar um modelo de infraestrutura de chave pública baseado em tal tecnologia.

No trabalho desenvolvido por LIANG *et al.* (2017) é proposto o uso de uma *blockchain* como uma base de dados para armazenamento de metadados relacionados a criação e operação de objetos de dados em nuvens, garantindo um carimbo de tempo para os registros, bem como a imutabilidade dos dados. Além disso, Tapscott e Tapscott (2016) afirmam que a tecnologia de blockchain pode ser utilizada para melhorar a prestação de serviços ao mesmo tempo que garante integridade e transparência das informações. São exemplos de uso dessa tecnologia em serviços de governos: armazenamento antifraude de registros públicos, como

propriedades privadas e antecedentes criminais; identificação digital de pessoa física ou jurídica; e digitalização da moeda nacional.

3.1.2 Hyperledger Fabric

Devido às penalidade de desempenho associadas aos algoritmos de PoW apresentadas na Seção 2.3.2 e ao fato de que a tecnologia *Blockchain* estar ganhando a atenção de muitas indústrias, a ideia de *blockchains* permissionadas está rapidamente ganhando utilidade e popularidade. Androulaki et al. (2018) apresentam o *Fabric* para desenvolvimento de *blockchains*. Os autores preocupam-se em mostrar a arquitetura do *Fabric* e a lógica por trás das diversas decisões de design, sua implementação em seus aspectos proeminentes, bem como seu modelo de programação de aplicações distribuídas. Além disso, é feita uma avaliação do *Fabric* implementando e comparando com um *benchmarking* de uma moeda digital inspirada no Bitcoin. O *Fabric* mostrou alcançar taxa de transferência *end-to-end* de mais de 3500 transações por segundo utilizando implantações ditas populares.

Em relação aos mecanismos de consenso para o *Fabric*, diversos trabalhos propõem implementações de mecanismos de consenso baseados no problema dos generais bizantinos para evitar falhas na ordem que os blocos são adicionados na *blockchain*. Por exemplo, o *software* Tendemint implementa o protocolo *Byzantine fault-tolerant* projetado em (BUCHMAN, 2016) e Martino (2016) usa uma variante do protocolo de consenso de jangada (ONGARO; OUSTERHOUT, 2014) adaptado para o problema dos generais bizantinos. Porém uma pesquisa mais recente (SVINIVAS, 2017) faz uma comparação entre alguns desse protocolos de consenso para redes permissionadas e aponta o algoritmo BFT-SMART (BESSANI et al., 2014) como o candidato mais promissor para implementar o protocolo de consenso em redes como o *Hyperledger*.

3.2 APLICAÇÕES DESCENTRALIZADAS

Entre os DApps, o jogo *CryptoKitties* [1] obteve bastante destaque. O mesmo permite os usuários colecionem gatos virtuais denominados "*criptokitties*", que podem ser bem caros de acordo com sua raridade. Cada gato é representado por um código único, que define sua aparência e personalidade. Os jogadores podem até usar seus mascotes para procriarem e gerar novos *criptokitties*. A brincadeira, desenvolvida na rede *Ethereum*, já alcançou 12 milhões em

[1] Disponível em: <https://www.cryptokitties.co/>. Acesso em: 03 de janeiro de 2019.

vendas em seu mercado descentralizado.

A plataforma *Golem* [2] é um mercado global de poder de processamento, criado na rede *Ethereum*. Por meio do sistema, qualquer um que tenha capacidade computacional ociosa pode cedê-la à rede Golem, em troca de tokens. Esse recurso é muitas vezes utilizado por diversos artistas para renderizar animações feitas com computação gráfica. Outra plataforma criada na rede *Ethereum*, *Aragon* [3] permite criar organizações autônomas descentralizadas. São organizações cujas regras são especificadas através de contratos inteligentes, os quais são executados e validados pela rede *Ethereum*. Como exemplos de utilização tem-se: arbitragem, gerenciamento e transferências de tokens, captação de recursos entre outros. O *Aragon Network Token* permite a participação das pessoas na operação e tomada de decisões da rede.

Como uma alternativa descentralizada ao Twitter, a rede *Peepeth* [4], possui um sistema imune à censura que funciona de modo bastante parecido com o microblog, mas tudo fica registrado em *blockchain*, em vez de servidores como em outras companhias. É preciso pagar em Ether para se registrar no *Peepeth* e também a cada ação realizada. São valores baixos, equivalentes a poucos centavos.

Em relação à *blockchains* permissionadas usando a tecnologia *Hyperledger*, a *MonetaGo* [5] é uma empresa de desenvolvimento de sistemas e membro do *Hyperledger* com sede em Nova York, e trabalha com instituições financeiras e bancos centrais em todo o mundo para fornecer soluções através de uma *blockchain* permissionada. Em apenas três meses após o lançamento, a rede já atende a uma porcentagem significativa do mercado de fomento mercantil na Índia, processando milhares de contas diariamente sem financiamento duplicado.

Outro caso de sucesso usando o *Hyperledger*, é o *Cambio Coffee* [6], uma empresa que possui uma *blockchain* responsável por conectar milhões de seus produtos relacionados com café à Internet, dando a cada um desses produtos uma identidade digital única na rede. Com seu código QR à prova de cópia, a empresa traz confiança, transparência e rastreabilidade à sua cadeia de suprimentos. O *Cambio Coffee* implementou os códigos QR exclusivos em seus pacotes desde maio de 2018. Atualmente, o responsável pela torragem do café e a empresa de entrega inserem dados no blockchain, porém o plano é lançar o recurso para a empresa de transporte e eventualmente, os agricultores, cobrindo assim toda a cadeia de suprimentos. No futuro, a

[2] Disponível em: <https://golem.network/>. Acesso em: 03 de janeiro de 2019.
[3] Disponível em: <https://aragon.org>. Acesso em: 03 de janeiro de 2019.
[4] Disponível em: <https://peepeth.com/welcome> Acesso em: 04 de janeiro de 2019.
[5] Disponível em: <https://www.monetago.com/> Acesso em: 10 de janeiro de 2019.
[6] Disponível em: <https://www.cambiocoffee.com/> Acesso em: 10 de janeiro de 2019.

empresa deseja expandir para outras iniciativas apoiadas pela *blockchain*, como a chamada "*Tip Your Farmer*", um meio de incetivo aos agricultores por meio da própria *blockchain*.

Por fim, a Associação Nacional de Agentes Imobiliários da América também aprimorou seus serviços com a tecnologia *Hyperledger*, as várias associações locais e estaduais são corporações independentes trabalhando juntas para fornecer informações sob um Código de Ética que tem mais de cem anos de idade. Existe um banco de dados centralizado que contém informações de faturamento dos membros. Embora seja essencial compartilhar informações sobre o envolvimento dos membros, é difícil fazer com que 1200 organizações concordem em contribuir e renunciar ao controle de seus dados para um único banco de dados. O protótipo denominado *BlockR* [7] foi desenvolvido por seis associações. Na semana seguinte à apresentação da primeira versão, 51 associações pediram para fazer parte, pois antes a associação de novos membros que costumava levar horas via e-mails ou telefonemas, agora levavam apenas alguns segundos.

3.3 CONCLUSÕES DO CAPITULO

Este capítulo teve o objetivo de mostrar alguns dos trabalhos relacionados a essa pesquisa. A seção foi dividida em duas partes: a primeira apresentando artigos científicos relacionados diretamente a *Blockchain* e *Hyperledger* enquanto a segunda parte apresentou um mapeamento de diversas aplicações descentralizadas utilizando a tecnologia *blockchain*.

[7] Disponível em: <https://www.nar.realtor/blockr/>. Acesso em: 13 de janeiro de 2019.

4 PROCEDIMENTOS METODOLÓGICOS

O presente trabalho pode ser classificado como uma pesquisa de natureza qualitativa e exploratória. Qualitativa, pois a pesquisa possui uma preocupação fundamental com o estudo e a análise do mundo empírico em seu ambiente natural através de uma análise com enfoque subjetivo (GODOY, 1995). Neste caso, será adotado o uso de entrevista em profundidade por meio de um questionário semi-estruturado, conforme será discutido posteriormente sobre a validação da solução. Em relação a natureza exploratória, o trabalho segue o ideal proposto por Piovesan e Temporini (1995) onde o investigador deve definir o problema de pesquisa e formular sua hipótese com maior precisão. Neste caso, a pesquisa exploratória reflete-se através da apresentação de uma nova solução para o Controle de Registros Médicos.

Seguindo a classificação definida por Wazlawick (2017), o presente trabalho se enquadra na opção “Apresentação de um produto”, o qual caracteriza-se por apresentar um novo produto para resolver um problema relevante que ainda não foi solucionado por outros produtos, ou ainda resolver tais problemas de uma maneira mais eficiente, visando agregar novos conhecimentos para a área e comparar o trabalho apresentado com trabalhos existentes. Nas próximas seções são apresentados os procedimentos metodológicos fundamentais adotados nesta pesquisa. Nelas, serão informadas, de forma sintética, as quatro etapas que compõem o trabalho.

4.1 REVISÃO TEÓRICA

Inicialmente, realizou-se uma revisão da literatura de caráter exploratório visando adquirir os conhecimentos necessários para a compreensão do domínio do projeto. A revisão permitiu a identificação, leitura e análise de publicações relevantes ao desenvolvimento do projeto e que fornecem a sustentação teórica para o estudo desenvolvido nas demais fases.

Assuntos referentes à leis sobre o exercício legal da medicina e a devida inscrição de registro dos médicos no CRM foram fundamentais para o entendimento do problema em questão no início da pesquisa. Também houve uma revisão sobre conceitos de livro-razão distribuído que é a base da tecnologia que será utilizada para solucionar o referido problema e, mais especificamente, os conceitos de um tipo de livro-razão distribuído, o *Blockchain*, também foram compreendidos. Grande parte dos conceitos de *Blockchain* presentes atualmente na literatura foram explorados, incluindo os seus tipos, como é o caso das *Blockchains* Permissionadas, as qual serão utilizadas como modelo de solução tecnológica, especialmente, relacionado à

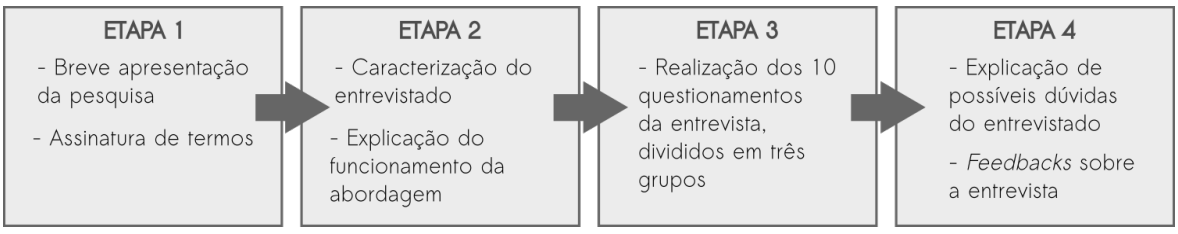
tecnologia *Hyperledger*.

Uma pesquisa a respeito do projeto *Hyperledger* foi realizada e envolveu os conceitos principais da tecnologia, seus componentes, as ferramentas necessárias para implementação (*Fabric*, *Composer* e *Playground*) e, até mesmo, o desenvolvimento da *blockchain* para estudo e testes da rede. Assuntos referentes à aplicações descentralizadas foram compreendidos a fim de se ter o aprofundamento sobre serviços de infraestrutura abertos e descentralizados. Por fim, apesar da escassez de literatura, foi realizada uma breve revisão sobre a subárea denominada de Engenharia de Software orientada a *Blockchain* que vem emergindo, a qual busca relacionar a adoção dos procedimentos oriundos da Engenharia de Software no contexto de Blockchain.

4.2 VALIDAÇÃO DA PROPOSTA

Após a compreensão do domínio do projeto, com o objetivo de validá-lo, foi realizada uma entrevista em junto a um diretor do Conselho Regional de Medicina do Estado do Ceará (CREMEC), a fim de obter as percepções subjetivas do entrevistado sobre a referida proposta. Este procedimento é adequado para avaliá-la pois permite a percepção individual de um pequeno número de pessoas sobre uma determinada ideia, programa ou situação (BOYCE; NEALE, 2006). Assim, foi possível compreender as diversas questões relevantes para a aceitação e implantação do produto. A estrutura da entrevista seguiu quatro etapas, conforme a Figura 15, as quais serão descritas a seguir.

Figura 15 – Etapas da entrevista em profundidade



Fonte – Elaborada pelo autor.

Primeiramente, uma breve explicação sobre a pesquisa foi feita ao entrevistado e assinatura de termos sobre acordos de confidencialidade e não divulgação foram coletados (visto em Apêndice B). Em uma segunda etapa, a fim de obter a caracterização do entrevistado e sobre o Conselho Regional da Medicina como instituição, onze perguntas foram realizadas as quais podem ser visualizadas no Apêndice C. Além disso, uma detalhada explicação sobre o

funcionamento da abordagem, levando em consideração todos os componentes e a relação entre eles, foi apresentada ao participante conforme descrito na Seção 5.4.

Na terceira etapa, deu-se início aos questionamentos da entrevista. Em geral, foram realizadas dez perguntas as quais foram agrupadas em três pontos (visto em Apêndice D). O primeiro grupo de perguntas, com cinco questões, estava relacionado aos serviços prestados pelo Conselho Regional de Medicina. Estas questões abrangiam desde os serviços oferecidos aos médicos até o relacionamento do CRM com outras entidades, como Instituições de Ensino Superior (IES) e/ou sociedades filiadas à Associação Médica Brasileira (AMB) e entidades de saúde credenciadas à Comissão Nacional de Residência Médica (CNRM). Em relação ao segundo grupo, com três perguntas, o mesmo lidava com questões no âmbito de armazenamento e compartilhamento dos dados entre os CRMs. Por fim, as últimas duas perguntas do questionário tinham o objetivo de obter do entrevistado as principais considerações positivas e negativas a respeito da abordagem.

Finalmente, a quarta etapa foi estruturada para esclarecer possíveis dúvidas e coletar *feedbacks* gerais sobre o processo de entrevista (visto em Apêndice E). A entrevista foi feita individualmente, autorizada a ser gravada em áudio e transcrita em texto para, posteriormente, explorar a riqueza do material coletado. Apesar das perguntas definidas, por se tratar de uma entrevista semi-estruturada, algumas dúvidas e questões extras sugiram no decorrer da entrevista e que puderam ser exploradas também na validação. Em média, a entrevista durou cerca de uma hora e foi realizada no próprio ambiente de trabalho do entrevistado, no Conselho Regional de Medicina do Ceará (CREMEC). Vale ressaltar que, para concedimento da entrevista, foi necessária a solicitação por meio de um oficio o qual foi protocolado, processo que durou cerca de três semanas após a entrega da solicitação.

4.2.1 Realização da Entrevista

Conforme apresentado anteriormente, para validar a solução proposta, realizou-se uma entrevista em profundidade de, aproximadamente, uma hora de duração. A entrevista para validação da proposta consistiu em quatro etapas cujas seções estão sendo apresentadas no Apêndice A. De acordo com Robson (2002) a principal ameaça ao fornecimento de uma descrição válida do que se viu ou ouviu está embasada na falta de precisão ou completude dos dados. Portanto seguindo as recomendação de Runeson e Höst (2009), e com a devida permissão do entrevistado, a entrevista teve seu áudio gravado como já mencionado na Seção 4.2, ação essa

que serve como uma estratégia para mitigação a ameaça de uma descrição errônea. A transcrição foi realizada com a utilização das ferramentas Parlatype e Google Docs.

No início da entrevista foi reforçado o aspecto sigiloso das informações prestadas, salientando que não seriam publicadas, mas sim utilizada para validar a solução a que o estudo se propõe. Segundo as informações concedidas na Etapa 2, o entrevistado tem 49 anos e foi disponibilizado pelo Conselho Regional de Medicina do Estado do Ceará (CREMEC) para conceder a entrevista. O participante é médico e possui uma formação acadêmica de Doutorado em Ciências Médicas, com 30 anos de experiência na área médica e há seis meses está vinculado ao CREMEC como um dos principais membros da diretoria. Em relação às informações gerais do CREMEC, a instituição funciona há 60 anos e conta, atualmente, com 20 a 99 funcionários trabalhando nas diversas funções oferecidas pelo CREMEC, estas funções foram detalhadas pelo entrevistado. Na Etapa 3 ocorreu a aplicação das questões ao entrevistado que dividiu-se em três agrupamentos, de acordo com a Seção 4. Na Etapa 4 o entrevistado respondeu um formulário a respeito da explicação da abordagem por parte do entrevistador, a qual ele considerou clara.

4.2.2 Serviços prestados pelo Conselho Regional de Medicina (CRM)

Inicialmente, buscou-se confrontar as respostas obtidas na entrevista quanto ao funcionamento do CRM em relação à base teórica levantada na Seção 2.1 a qual serviu de base para a presente proposta. Segundo como explanado pelo participante, constatou-se que os fluxos adotados pelo presente trabalho estão condizentes com os processos realizados na prática pelos CRMs. Além disso, o mesmo acrescentou informações sobre a organização do CRM na realização de suas atividades, sendo essa dividida em cinco funções: (i) cartorial; (ii) judicante; (iii) normativa; (iv) fiscalizatória e (v) pedagógica. Essas informações, inclusive, foram utilizadas para incrementar a Seção 2.1, onde está sendo descrita cada uma das funções mencionadas acima. Em particular, este trabalho tem foco na função cartorial, a qual é responsável pelas atividades que garantem o registro do diploma do médico para comprovar sua formação em medicina e autorizar, por lei, seu exercício na referida profissão. O entrevistado salienta a importância dessa

função na frase:

> *"O CREMEC é o local onde o médico deve registrar seu diploma após formar-se em medicina, caso isso não ocorra, ele não está autorizado a exercer a profissão. Isso vale, não somente para médico, mas para pessoas jurídicas como clínicas, serviços públicos e privados."*

A afirmação acima mostra que não são apenas os médicos que podem usufruir de tais serviços prestados pela solução proposta, mas também serviços à setores públicos e privados. Por exemplo, para empresas, instituições, entidades ou estabelecimentos prestadores e/ou intermediadores de assistência à saúde com personalidade jurídica. Isso abre margem para melhoria e ampliação futura da abordagem nesse sentido.

Em relação ao processo de inscrição realizado no CREMEC, o entrevistando afirmou que a inscrição deve ser feita de maneira presencial, ou mediante procuração, com os seguintes documentos: RG, CPF, comprovante de endereço, diploma de graduação, certificado de Especialização (caso possua) em um período de 30 até 60 dias após a formatura. No caso de pessoa jurídica, o diploma e os documentos do responsável técnico pela empresa devem ser fornecidos. Essas afirmações corroboram com a importância do processo descrito na Seção 5.4 pois, uma vez que esses dados podem ser adicionados à *blockchain* através dos participantes CRM, IES e EE, a solicitação de uma primeira inclusão poderia ser automatizada, sem a necessidade de ser obrigatoriamente presencial.

A possibilidade de CRMs terem funções diferentes de acordo com cada estado foi uma preocupação no momento da modelagem da solução. Entretanto, quando questionado sobre os serviços oferecidos entre os CRM do país, o entrevistado afirmou:

> *"Todos os CRMs prestam o mesmo serviço, pois são padronizados. Existe um manual de procedimento técnico para os CRMs elaborado pelo CFM que os CRMs são obrigados a seguir. Todos prestam os mesmos serviços: cadastro de pessoa física, cadastro de pessoa jurídica, acolhimento de denúncia, sindicância, procegimentagem profissionais, processos".*

Essa padronização demonstra consonância com a forma de como os CRMs são definidos na solução proposta. Isto porque um nó definido como um participante do tipo CRM, recebe o mesmo nível de acesso a todos os serviços dos demais de mesmo tipo, os quais estão sendo modelados de mesmo modo na Seção 5.

Quando questionado sobre a existência de algum protocolo que as EEs devam seguir

a fim de que os certificados possam ser válidos para o CRM, o entrevistado afirmou que a atuação das EEs é autônoma em relação ao CRM. Segundo o diretor:

> *"A AMB reúne todas as sociedades de especialidade que, por sua vez, aplicam as provas de títulos. O resultado dos que são aprovados é enviado para AMB que emite um certificado, transcorrendo sem qualquer interferência do CRM. Em outras palavras, o CRM não tem gerência nem ingerência sobre os processos dessa entidade".*

Essa assertiva indica que a EE realmente pode ser vista como um participante particular no processo de registro do médico. Logo, a forma como as restrições de acesso são definidas na Subseção 5.2.2 para as EEs, estão condizentes com o que realmente ocorre na prática.

4.2.3 Armazenamento e compartilhamento dos dados

Visando salientar a importância de uma abordagem descentralizada para o problema em questão, procurou-se investigar a atual arquitetura adotada pelo CRM para armazenamento e gerenciamento dos dados. Conforme hipotetizado, o entrevistado contextualizou que os dados são registrados de forma centralizada. Ou seja, o CRM de cada estado envia os dados para um único banco de dados persistido pelo CFM. Segundo o participante:

> *"Os dados do CRM são compartilhados com o CFM. Porém os CRMs não compartilham os dados entre si. [...] Caso precise de informações de um médico de outro estado, o CRM local envia um ofício solicitando todas as informações necessárias. [...] O CRM apenas envia dados para o ponto central e não recebe dados sem autorização. Inclusive, o regional que autoriza a disponibilização de dados por parte do federal, através de contato direto."*

A partir disso, identifica-se uma burocratização ao acesso à informações importantes que, muitas vezes, deveriam ser acessadas com certa prontidão a depender do serviço que está sendo solicitado. Além disso, visto que todas as informações estão armazenadas de maneira centralizada, a segurança da rede torna-se sujeita à um único ponto de falha. Portanto, observa-se a relevância de uma possível solução descentralizada de modo a evitar esses impasses.

Outro ponto a ser considerado é a questão do compartilhamento de informações entre os participantes (CRM, EE e IES). Segundo o entrevistado, "não há nenhum tipo comunicação

do banco de dados entre as EEs e os CRMs". Ele explica que o CRM já solicitou maneiras de compartilhamento das informações por parte das entidades emissoras de certificados de especialidade a fim de realizar verificações no momento do registro médico e de outros processos. Essas informações seriam, por exemplo, certificações de especialidades recém emitidas pela AMB ou CNRM. Nesse caso, como apresentado na Seção 5.4, a proposta também demonstra grande potencial em atender esse aspecto, haja vista que uma das principais características da solução é prover o compartilhamento e a transparência dos dados.

4.2.4 Opinião do entrevistado em relação à proposta

Por fim, considerando a opinião do entrevistado em relação a solução proposta, como ponto negativo, o entrevistado levantou a questão da operacionalização da proposta, pois, há a necessidade da especialização de profissionais para lidar com a tecnologia *blockchain*, e afirma: "Esse é a único ponto negativo que eu vejo: você depender de profissionais especializados. Talvez, isso seja um gargalo". Porém, logo após, o próprio entrevistado acrescentou que não considera a solução verdadeiramente um ponto negativo, mas apenas uma barreira. Essa opinião demonstra que a solução proposta neste trabalho pode ter desafios relevantes a serem solucionados a fim de que a mesma seja implementada. Isto porque essa arquitetura, de certa forma, causaria uma considerável mudança no funcionamento interno dos CRMs, e entre os CRMs e as outras instituições participantes da rede, o que pode prejudicar sua aceitação por parte do órgão.

Entretanto, o mesmo frisou a questão da segurança como o principal aspecto positivo da abordagem, segundo ele "quanto mais mecanismos de segurança houverem que resguardem nossos dados, isso é ótimo". Reforçando, assim, a necessidade que os CRMs possuem por mecanismos de segurança como a *blockchain* para proteção dos dados.

4.3 IMPLEMENTAÇÃO DA SOLUÇÃO

Após uma validação do domínio e os requisitos levantados advindas das etapas anteriores, dar-se-a início a fase implementação da solução. Para desenvolvimento da solução, será utilizado a linguagem *Python* juntamente com o *framework flask* (GRINBERG, 2018). Conforme apresentado na Figura 16, a arquitetura genérica do aplicativo se dividirá em duas partes, uma interface web que possibilita o acesso por computador e a *blockchain* responsável por armazenar as informações das inscrições dos registros dos médicos.

Figura 16 – Exemplo de estrutura do sistema proposto

Fonte – Elaborada pelo autor.

Como forma de hospedagem remota da aplicação será utilizado a plataforma *Heroku*, tendo a vantagem de não se preocupar com a instalação de servidores físicos, e oferecendo uma economia substancial a curto, médio e longo prazo (KRISHNAN; GONZALEZ, 2015). Adicionalmente, a plataforma *Firebase* é utilizada para o armazenamento do cadastro dos usuários do sistema web, sendo que este pode adaptar-se de acordo com as necessidades em um simples passo, basta apenas solicitar a alteração dos limites de armazenamento. Por fim, o *framework Hyperledger* será utilizado para criar a *blockchain*, garantindo uma plataforma robusta e fácil de usar. Mais detalhes de como é implementada a solução podem ser vistos na Seção 5.2. Mais detalhes sobre como o sistema web foi implementado e todos os *stack* de desenvolvimento utilizados podem ser vistos na Seção 6.

5 ARQUITETURA DA SOLUÇÃO

Nesta seção serão discutidos os detalhes referentes ao escopo e arquitetura da solução proposta neste trabalho. Na primeira seção é mostrada uma visão geral do funcionamento da abordagem proposta. Em seguida, é descrito a modelagem adotada para a solução proposta, destacando-se os detalhes relativos como estão organizados e os elementos que a compõem. Por fim, na seção seguinte é apresentado uma demonstração de como a solução funciona.

5.1 SOLUÇÃO PROPOSTA

A solução proposta neste trabalho é baseada nos atuais processos necessários para a inscrição do médico e de suas informações relevantes no CRM, podendo este realizar eventos diversos, como a sua primeira inscrição, reinscrição, transferências, cancelamentos e registro de especialidade, por exemplo. Assim sendo, faz-se necessário levar em consideração as entidades que compõem o cenário de registro de informações. Em consonância aos processos descritos na seção anterior, propõe-se as seguintes categorias as quais representam os nós participantes da rede:

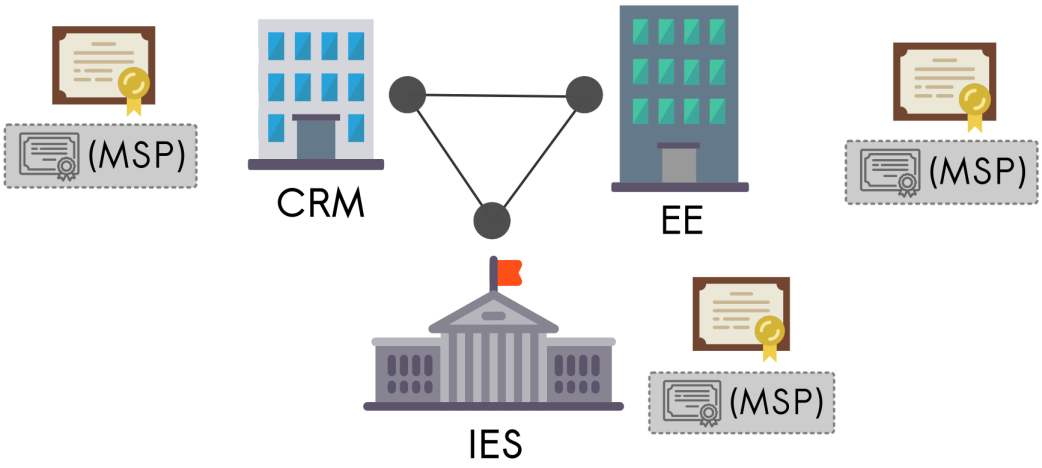
- **Conselho Regional de Medicina (CRM):** é a entidade responsável por fiscalizar, apurar e julgar irregularidades contra médicos no Estado. Além disso, é o órgão responsável pelo registro de diplomas, títulos de especialidade e informações relevantes referentes ao profissional;
- **Instituição de Ensino Superior (IES):** é uma instituição que promove educação em nível superior que, de acordo com suas características, são classificadas como universidade e faculdades de ensino superior;
- **Entidade de Especialidade (EE):** é uma instituição que emite os certificados de residência médica e títulos de especialidades. Esses documentos podem apenas ser emitidos pelas sociedades de especialidades filiadas à Associação Médica Brasileira (AMB) e/ou instituições de residência médica credenciados pela Comissão Nacional de Residência Médica (CNRM) (BRASIL, b). Essas instituições tem a finalidade de permitir a qualificação do médico graduado nas diferentes especialidades da medicina em nível de pós-graduação, no qual os aprendizes desenvolvem competências específicas para a aplicação de suas funções de forma adequada.

Deve-se ressaltar que cada uma dessas categorias é composta por um conjunto de nós.

Por exemplo, existem 27 nós da categoria CRM que representam o CRM de cada estado brasileiro e o distrito federal. Semelhantemente, existem nós que representam cada IES, instituição de saúde e sociedade de especialidade do Brasil. Assim, forma-se uma grande rede composta de nós que correspondem a cada uma dessas entidades.

A presente proposta utiliza uma *blockchain* permissionada da tecnologia *Hyperledger* apresentada na Seção 2.4, onde cada nó define papéis que são atribuídas aos participantes e o acesso é concedido ou restrito por meio desses papéis. Assim, um provedor de serviço MSP trata de gerar as credenciais para um dos diversos tipos de participantes como apresentado na Figura 17.

Figura 17 – Visão geral



Fonte – Elaborada pelo autor.

Entretanto, no caso especifico da solução proposta, ela segue um modelo mono-organizacional, onde existe apenas uma única rede, e todos os nós devem ser certificados como um dos três tipos de participantes. Para o cadastro nesse modelo, apenas o um dos nós CRM recebe o certificado diretamente da blockchain *Hyperledger* e, portanto, ficando responsável de através dele, permitir que o serviço MSP gere os certificados para os outros dois participantes (IES e EE) como apresentado na Figura 18.

Dessa forma, um membro específicos (IES ou EE) apenas terá acesso a recursos importante após a validação do certificado emitido pelo MSP através do CRM. É importante salientar que o CRM apenas emite os certificados, ele não tem acesso aos recursos utilizados pelos demais participantes, por exemplo, o cadastro do Diploma. Na próxima Subseção será discutido mais detalhadamente a estrutura do projeto e como o mecanismo de autorização de

Figura 18 – Emissão de Certificados pelo CRM

Fonte – Elaborada pelo autor.

acesso ao recurso é implementado.

5.2 ESTRUTURA DA SOLUÇÃO

A estrutura de uma solução utilizando a tecnologia *hyperledger*, pode ser dividida em diferentes etapas, diante disso a presente subseção divide a estrutura da solução em três partes: 1) a definição dos componentes que a *blockchain* deve possuir, apresentando como a rede é montada. 2) a modelagem da parte lógica, onde é definido como cada componente deve ser comportar e 3) a estrutura utilizada para comunicação do *front-end* da aplicação com a *blockchain*.

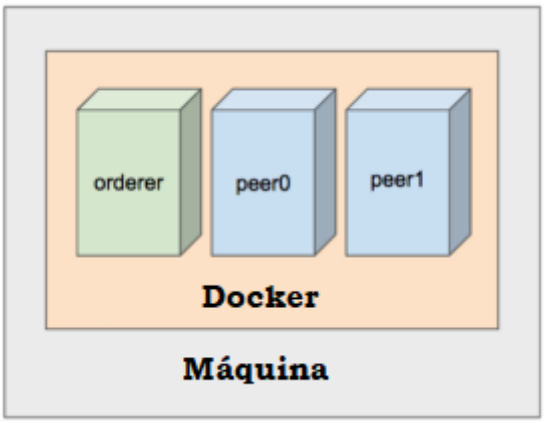
5.2.1 Definição dos componentes da rede

Para criação de cada nó da *blockchain*, foi utilizado a tecnologia Docker [1]. Docker é uma plataforma que automatiza a implantação de aplicações dentro de ambientes isolados denominados contêineres (TURNBULL, 2014). Trata-se portanto, de uma solução para desenvolvedores cujo objetivo é proporcionar múltiplos ambientes isolados dentro do mesmo servidor, mas acessíveis externamente via tradução de portas. No caso em questão, como mostrado na Figura 19, vários contêineres Docker são utilizada para armazenar os serviços seguindo o modelo apresentado na subseção 2.4.1. Assim, diversos serviço são mantidos em contêineres separado dentro do mesmo servidor.

Entre esses serviços podem estar por exemplo: o *Ordering Service* (orderer) responsável pela criação dos blocos, o nó *Anchor* (peer0) responsável por receber o bloco enviado pelo *Ordering Service* e o nó *Endorsing* (peer1) que armazena o Livro-razão. Entretanto, como os participantes da *blockchain* tem funções diferentes, os mesmos são compostos por diferentes nós, como segue:

[1] https://www.docker.com/

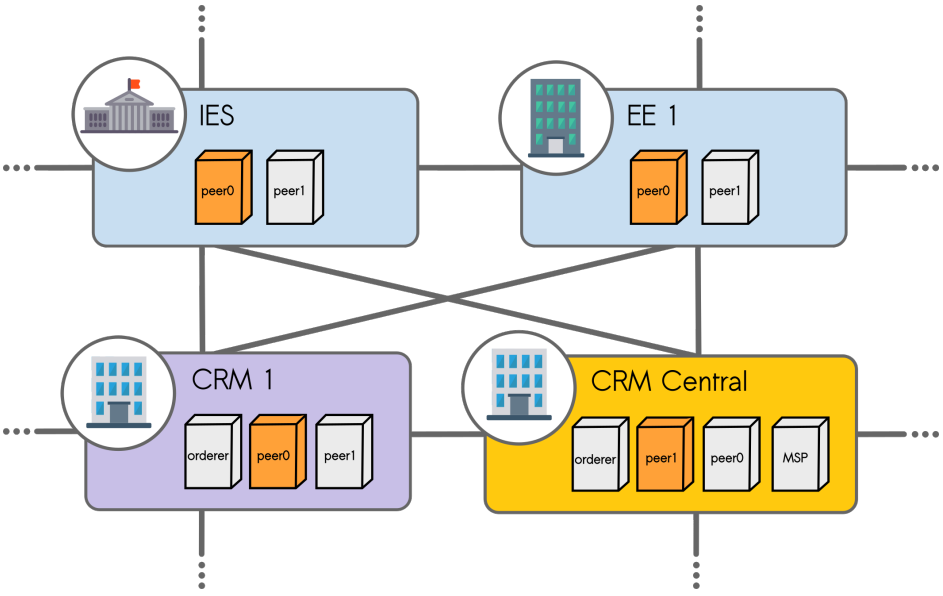
Figura 19 – Exemplo de contêiner utilizado em um nó do CRM.



Fonte – Elaborada pelo autor.

- **CRM:** representam os CRMs, como apresentados na Figura 18. Contém o serviço de criação de blocos (*Ordering Service*) e os nós *Endorsing* e *Anchor*. Entretanto, existe apenas um desse tipo na rede considerado como CRM central (Distrito Federal). A diferença é que, além dos nós já mencionados, esse participante também contém o serviço de certificação (MSP).
- **EE e IES:** possui apenas os contêiners com os nós *Endosing* e *Anchor*.

Figura 20 – Estrutura da solução proposta



Fonte – Elaborada pelo autor.

A Figura 20 ilustra a relação entre esses os participantes. O CRM central é gerado quando a *blockchain* é criada, como mencionado é o único dos CRMs que possui um contêiner com um nó com o serviço MSP, sendo assim o responsável por gerar os certificados que irão

adicionar os demais participantes à rede.

Todos os participantes da *blockchain* possuem uma copia do livro-razão garantindo assim a integridade dos dados caso um deles seja atacado. Entretanto, apenas as instâncias do CRM, possuem o *Ordering Service* que criam blocos e adicionam na rede. Logo, o sistema é montado de modo que quando deseja-se adicionar uma informação à blockchain, a instância mais próxima da requisição, pertencente a um dos 27 CRM, cria o bloco e distribui para os demais.

Outro aspecto importante é a comunicação entre os participantes da rede. Para que uma instância possa distribuir um bloco dentro da rede, é necessário que esse comunique-se com máquinas que estão em locais diferentes. Para permitir essa comunicação, foi repassado um arquivo de configuração **.yml** para todos os contêineres da rede, informando o endereço ip de todas as máquinas presentes na rede. A Figura 21 demostra uma parte do arquivo **.yml** e como os endereços são passados para um nó *anchor* da rede.

Figura 21 – Parte do arquivo .yml usado para configurar a rede

```
peer0.org1.example.com:
  container_name: peer0.org1.example.com
  image: hyperledger/fabric-peer
  environment:
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    - CORE_PEER_ID=peer0.org1.example.com
    - CORE_LOGGING_PEER=info
    - CORE_CHAINCODE_LOGGING_LEVEL=info
    - CORE_PEER_LOCALMSPID=Org1MSP
    - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/peer/
    - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
    - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=${COMPOSE_PROJECT_NAME}_basic
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric
  command: peer node start
  ports:
    - 7051:7051
    - 7053:7053
  volumes:
      - /var/run/:/host/var/run/
      - ./crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp:/etc/hyperledger/msp/peer
      - ./crypto-config/peerOrganizations/org1.example.com/users:/etc/hyperledger/msp/users
      - ./config:/etc/hyperledger/configtx
  extra_hosts:
    - "peer1.org1.example.com:172.31.44.235"
    - "peer2.org1.example.com:172.31.45.83"
    - "peer3.org1.example.com:172.31.33.22"
    - "peer4.org1.example.com:172.31.16.226"
  depends_on:
    - orderer.example.com
    - couchdb
  networks:
    - basic
```

Fonte – Elaborada pelo autor.

5.2.2 Modelagem lógica da rede

Em relação a modelagem lógica da rede, como explicado na Subseção 2.4.2, é feita através da ferramenta **composer** e utilizando um arquivo .bna. Esse arquivo é composto por quatro sub-arquivos: cto, acl,js e qry, responsáveis respectivamente pela definição dos participantes e transações, controle de acesso, implementação lógica das transações e definição

das consultas que podem ser realizadas.

Figura 22 – Exemplo de arquivo CTO usado na solução proposta

```
participant abstract Membro identified by Id {
  o String Id
  o String Nome
  ...
}

participant IES extends Membro {
  ...
}

participant IE extends Membro {
  ...
}

participant CRM extends Membro {
  o String estado
  ...
}

asset diploma identified by Id {
  o String Id
  ...
}

asset especialidade identified by Id {
  o String Id
  ...
}

asset registroMedico identified by Id {
  o String Id
  --> asset diploma
  --> asset especialidade
  ...
}

transaction RegistrarDiploma {
  --> diploma ativo
  ...
}

transaction RegistrarEspecialidade {
  --> especialidade ativo
  ...
}

transaction RegistrarDadosMedicos {
  --> registroMedico ativo
  ...
}
```

Fonte – Elaborada pelo autor.

A Figura 22 exemplifica o arquivo *.cto* utilizado na solução proposta. Nele são definidos: os tipos participantes utilizados (IES, EE e CRM) com seus atributos, os ativos que serão manipulados por esses participantes (Diploma, Especialidade e Registro Médico) e que tipo de transações podem ser feitas com esses ativos. Salienta-se que o ativo **registroMedico** tem a possibilidade de guardar as informações do médico e os ponteiros para os ativos Diploma e Especialidade do respectivo médico.

Na Figura 23 demonstram-se como as transações são definidas no arquivo js. Quando um dos participantes realiza a adição de um registro na rede, uma dessas três funções é invocada. A principal finalidade dessas funções consiste em: 1) Criar um objeto Javascript correspondente ao recurso que está sendo invocado e preenche-lo com os dados enviado pela transação. 2) Acessar o registro de ativos da *blockchain* e adicionar o novo registro na rede pelo método **.add(obj)**.

A relação de quais participantes pode invocar qual transação foi definida de modo que um tipo específico de participante possa manipular cada ativo, sendo essa manipulação restringida pelo arquivo de controle de acesso. A Figura 24 demonstra a primeira parte do arquivo ACL que define as regras de controle de acesso.

As restrições são utilizadas para permitir que apenas determinados participantes possam realizar operações em recursos específicos. Por exemplo, a regra **ManipularRegistrar-Diploma** permite que apenas participantes do tipo IES possam ter acesso a função **RegistrarDi-ploma**. Por sua vez, a restrição **ManipularDiploma** permite que o ativo **Diploma** seja invocado pelo participante IES quando está utilizando a função **RegistrarDiploma**.

Figura 23 – Exemplo de arquivo .js usado na solução proposta

```
async function RegistrarDiploma(tx) {

    const factory = getFactory();
    const NS = 'org.crm';
    const id = tx.Id;

    const obj = factory.newResource(NS, 'Diploma', id+'');
    obj.NomeDoCurso = tx.NomeDoCurso;
    obj.Data = tx.Data;
    ...

    const RegistroDiploma =
    await getAssetRegistry(NS + '.Diploma');

    await RegistroDiploma.add(obj);
}
```

```
async function RegistrarEspecialidade(tx) {

  const factory = getFactory();
    const NS = 'org.crm';
    const id = tx.Id;

    const obj = factory.newResource(NS, 'Especialidade', id+'');
    obj.NomeDaEspecialidade = tx.NomeDaEspecialidade;
    obj.Data = tx.Data;
    ...

    const RegistroEspecialidade =
    await getAssetRegistry(NS + '.Especialidade');

    await RegistroEspecialidade.add(obj);
}
```

```
async function RegistrarDadosMedicos(tx) {

    const factory = getFactory();
    const NS = 'org.crm';
    const id = tx.Id;

    const obj = factory.newResource(NS, 'RegistroMedico', id+'');
    obj.dip = tx.dip;
    obj.esp = tx.esp;
    ...

    const RegistroEspecialidade =
    await getAssetRegistry(NS + '.RegistroMedico');

    await RegistroEspecialidade.add(obj);
}
```

Fonte – Elaborada pelo autor.

Figura 24 – Exemplo de ACL usado no Controle de Acesso

```
rule ManipulaDiploma {
    description: "Apenas IES podem acessar informações do diploma"
    participant: "org.example.crm.IES"
    operation: ALL
    resource: "org.example.crm.Diploma"
    action: ALLOW
}
```

```
rule ManipulaRegistrarDiploma {
    description: "Apenas IES podem manipular informações do diploma"
    participant: "org.example.crm.IES"
    operation: ALL
    resource: "org.example.crm.RegistrarDiploma"
    action: ALLOW
}
```

```
rule ManipularEspecialidade {
    description: "Apenas EE podem acessar informações da Especialidade"
    participant: "org.example.crm.EE"
    operation: ALL
    resource: "org.example.crm.Especialidade"
    action: ALLOW
}
```

```
rule ManipularRegistarEspecialidade {
    description: "Apenas EE podem maripular informações da Especialidade"
    participant: "org.example.crm.EE"
    operation: ALL
    resource: "org.example.crm.RegistrarEspecialidade"
    action: ALLOW
}
```

```
rule ManipularDadosMedicos{
    description: "Apenas CRM podem acessar informações do diploma"
    participant: "org.example.crm.CRM"
    operation: ALL
    resource: "org.example.crm.RegistroMedico"
    action: ALLOW
}
```

```
rule ManipularRegistrarDadosMedicos{
    description: "Apenas CRM podem manipular informações do diploma"
    participant: "org.example.crm.CRM"
    operation: ALL
    resource: "org.example.crm.RegistrarDadosMedicos"
    action: ALLOW
}
```

Fonte – Elaborada pelo autor.

Além das regras que permitem a manipulação dos dados por participantes específicos, a Figura 25 apresenta outras regras também importantes para o funcionamento da rede. A regra **LerDados** permite que todos os participantes que tenham acesso apenas a leitura das informações dos presentes na rede, sendo bloqueada a possibilidade de escrita ou alteração desses dados diretamente. As demais permitem que apenas o CRM cadastre novos IES, EE ou CRM na rede.

Figura 25 – Outras regras de Controle de Acesso

```
rule LerDados {
    description: "Todos os participantes podem ler dados da blockchain"
    participant: "**"
    operation: READ
    resource: "**"
    action: ALLOW
}

rule AddEE {
    description: ""
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.crm.EE"
    action: ALLOW
}
```

```
rule AddIES {
    description: "Adiciona novo IES à rede"
    participant: "org.crm.CRM"
    operation: ALL
    resource: "org.crm.IES"
    action: ALLOW
}

rule AddCRM {
    description: "Adiciona novo CRM à rede"
    participant: "org.crm.CRM"
    operation: ALL
    resource: "org.crm.CRM"
    action: ALLOW
}
```

Fonte – Elaborada pelo autor.

Na Figura 26 é ilustrado um exemplo de arquivo *.qry* responsável pelas consultas aos registros dos profissionais médicos. A priori, foram definidos dois tipos de consulta: **todosRegistros** retorna um *json* com todos os registros médicos armazenados na *blockchain* e **registroEspecifico** que retorna um *json* com o registro baseado no Id passado no momento da consulta. Vale ressaltar que consultas por outras informações do médico (nome, endereço, situação) também podem ser adicionadas.

Figura 26 – Exemplos de consultas utilizadas

```
query todosResgistros {
  description: "Seleciona todos os Registros Médicos"
  statement:
      SELECT crm.registroMedico
}


query registroEspecifico {
  description: "Seleciona o registro de um médico Especifico"
  statement:
      SELECT crm.registroMedico
          WHERE (cadId ==_$id)
}
```

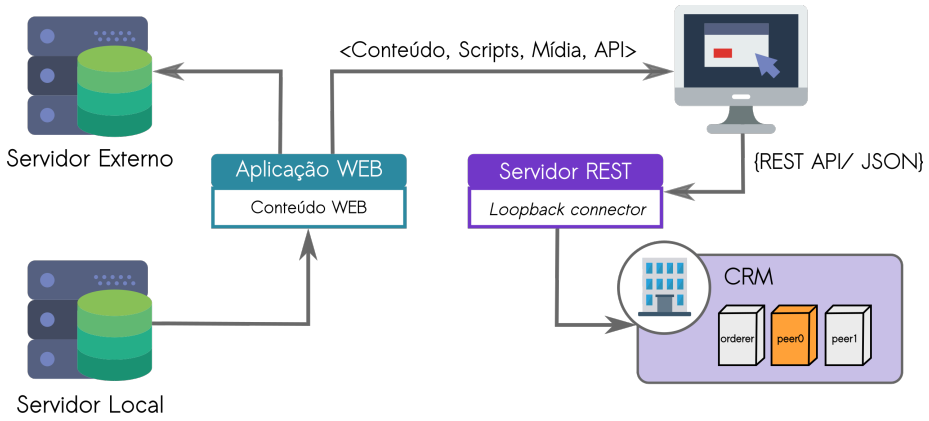
Fonte – Elaborada pelo autor.

5.2.3 Integração com o *front-end*

Definida a parte lógica da rede, é necessário também uma estrutura de comunicação entre a aplicação web e a *blockchain*. Para montar essa estrutura foi utilizado o estrutura de *API Rest* apresentado na Subseção 2.4.2, fornecida pela ferramenta **composer**. O design adotado pode ser visto na Figura 27.

Assim, uma aplicação web mantida localmente por um servidor do CRM como a apresentada na Figura 28, conecta-se em tempo de execução com a rede por meio de um

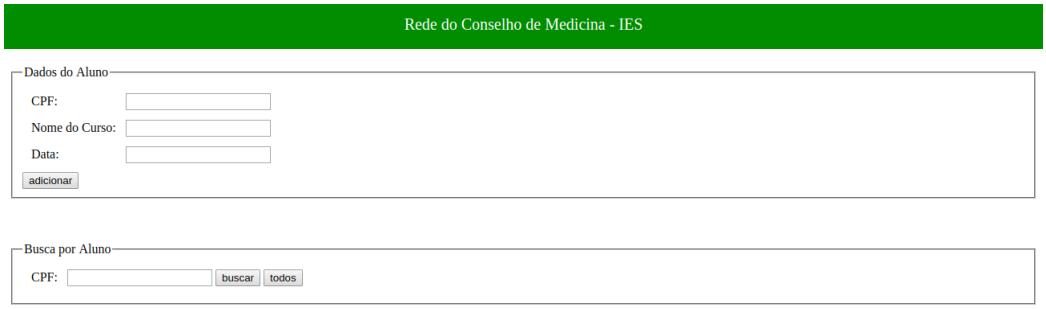
Figura 27 – Estrutura de comunicação com a blockchain.



Fonte – Elaborada pelo autor.

servidor REST. Este servidor possui as credenciais necessárias para poder interagir com a rede, convertendo as requisições *.json* enviados pela aplicação em transações para um nó *Anchor* presente nos servidores da rede.

Figura 28 – Exemplo de *Front-end* da Solução Proposta



Fonte – Elaborada pelo autor.

Para que um usuário possa conectar a aplicação com a *blockchain* é necessário uma autenticação realizada pelo servidor REST. A primeira chamada sem autenticação levará a um erro 401 pois não há autorização sendo recebida pela servidor. Nesse ponto, o aplicativo redirecionará o usuário para uma página de login na qual o usuário fornecerá suas credenciais. A página de login fará uma chamada a um provedor de identidade **OAUTH2.0** (HAMMER-LAHAV; HARDT, 2011) para validar as credenciais. Se tudo correr bem, o provedor de identidade emitirá um *token* de acesso e, por meio de um URL de retorno, enviará o *token* de acesso de volta a aplicação web.

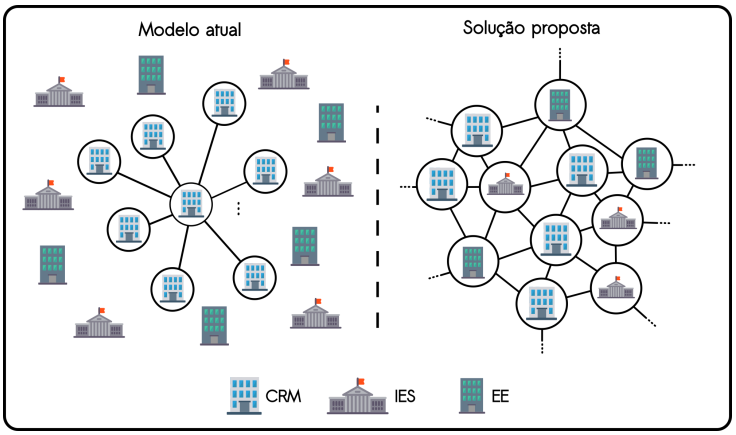
Deste ponto em diante, toda a API REST incluirá o *token* de acesso recebido do provedor de identidade. O servidor REST ao receber uma chamada realizará uma validação do token de acesso em tempo de execução, e se o *token* for considerado válido, o acesso a *blockchain*

será concedido. Opcionalmente, a aplicação Web também pode conectar-se a servidores externos, para buscar ou guardar informações que não precisam necessariamente estar na *blockchain*.

5.3 VANTAGENS E LIMITAÇÃO DA SOLUÇÃO PROPOSTA

A solução proposta, por utilizar a tecnologia *blockchain*, apresenta algumas vantagens em relação a estrutura atual de comunicação existente entre os CRMs. Inicialmente, conforme pode ser observado na Figura 29 (lado esquerdo), o atual modelo de compartilhamento dos dados ocorre de forma que os CRMs transmitem todas as informações de maneira exclusiva para um servidor central. No caso do modelo em questão, o servidor central está presente no Distrito Federal. Sendo assim, as informações de todos os CRMs estão reunidas em apenas um único ponto. Caso algum CRM requeira dados de um CRM de outro estado, é necessário que o mesmo faça solicitações ao servidor central a fim de obter a informação desejada. Esse modelo pode favorecer ataques maliciosos, o que pode afetar os dados de toda a rede. Em contra-partida, a solução proposta neste trabalho evita esse tipo de situação pois os dados são armazenados descentralizadamente, o que elimina a necessidade de um servidor central para proteger e autenticar os dados. Dessa forma, cada nó da rede possui uma cópia de todas as informações, facilitando assim, o acesso aos dados entre os participantes da rede (EEs, IESs, CRMs), como pode ser visto na Figura 29 (lado direito). Além disso, sistemas distribuídos permitem que os dados sejam recuperados caso um nó sofra alguma falha e perca esses dados (PRESSMAN, 1995).

Figura 29 – Contraste entre Modelo Atual e Solução Proposta.



Fonte – Elaborada pelo autor.

Outra vantagem da solução proposta é a imutabilidade dos dados, pois os blocos

de dados, uma vez criados, não podem ser modificados posteriormente. Ou seja, quando os dados são registrados em uma *blockchain*, não é possível removê-los ou alterá-los. Dada essa imutabilidade, o uso de *blockchain* demonstra-se pertinente para viabilizar a rastreabilidade das informações, visto que pode-se promover transparência aos dados, desde sua concepção até um estado atual. Sendo assim, cada nó pode obter todo o histórico de um profissional da medicina e, a partir disso, analisá-lo a fim de obter alguma informação relevante.

Além dos benefícios mencionados anteriormente, pode-se considerar a automação de grande parte dos processos, principalmente aqueles que necessitam de uma interação entre os diferentes participantes da rede. Tais vantagens podem influenciar potencialmente na desburocratização de determinados processos internos entre os participantes, assim como na redução de fatores importantes como custos e tempo de atendimento.

Ademais, poderosos mecanismos de consenso existentes na tecnologia *blockchain* garantem também que a informação seja adicionada apenas quando todos nós concordarem. Esse mecanismo evita que informações não consensuais sejam adicionadas em um nó por pessoas não autorizadas evitando, assim, possíveis fraudes na manipulação dos dados inseridos.

Em relação a limitações, a principal limitação encontrada na solução atual, trata-se do fato da solução considerar a rede apenas em um contexto mono-organizacional, ou seja, todos os participantes definidos (CRM, EE e IES) são tratados como membros de uma única rede. Isso impossibilita a criação de sub-redes impedido por exemplo, a criação de regras e níveis de privilégios que sejam validos apenas entre participante de um determinado tipo.

5.4 DEMONSTRAÇÃO DA SOLUÇÃO

A Figura 30 apresenta os fluxos de processos relacionados ao registro de informações do médico no CRM. Como já mencionado, para o médico exercer legalmente a medicina é necessário que este realize o registro de sua inscrição a qual é a primeira concedida ao médico após sua colação de grau junto à IES. O processo para realizar outros serviços disponíveis aos médicos, como reinscrição, inscrição secundária, inscrição temporária, transferência ou cancelamento, é semelhante aos passos a seguir no quesito documentação, no qual o CRM pode verificar o registro do diploma e das informações inicialmente registradas na *blockchain*.

1. A fim de realizar sua primeira inscrição, o recém-graduado em medicina depende da emissão de seu diploma pela sua respectiva IES. Assim, o início do fluxo de funcionamento da abordagem proposta necessita que a IES realize registro do diploma na *blockchain*.

Figura 30 – Fluxo do Registro de Informações Médicas ao CRM.

Fonte – Elaborada pelo autor.

2. O médico solicita o registro de sua inscrição ao CRM de acordo com as normas previamente estabelecidas.
3. O CRM verifica a validade do diploma registrado pela IES ou certificados de especialidade ao consultar a *blockchain*.
4. Após o CRM verificar o registro do diploma na *blockchain*, dados os devidos procedimentos, o órgão registra ou atualiza as informações do médico na *blockchain*.

A presente demonstração destaca e detalha outro evento em específico: o registro de especialidades do médico. Esse destaque é dado pelo fato de que o mesmo envolve órgãos terceiros que emitem os certificados e títulos de especialidades, documentos esses que também precisam ter o devido cuidado quanto sua validação e integridade das informações. Portanto, sugere-se que, assim como o diploma emitido pela IES, os certificados de especialidades também sejam registrados na *blockchain*. Os passos a seguir demonstram um cenário de funcionamento da abordagem para tal fim.

5. As EEs consultam as informações do profissional na *blockchain* com o propósito de verificar o diploma e as informações registradas no CRM.
6. Após verificar se as informações necessárias estão válidas, a instituição de especialidade realiza o registro do certificado de residência médica, caso seja uma instituição credenciada na CNRM ou certificado de prova de título, caso seja uma sociedade filiada à AMB.

7. O médico realiza a solicitação do registro de especialidade de acordo com as normas estabelecidas.

Semelhantemente ao fluxo do processo de realizar a primeira inscrição, os passos 3 e 4 se repetem após o passo 7, visto que o CRM deve consultar a validade das informações e registrar a especialidade do profissional na *blockchain*.

Além dos fluxos apresentados na Figura 30, pode-se considerar também o acesso a essas informações pelos próprios profissionais da medicina, visto que esses podem consultar se os documentos oficiais que garantem sua habilitação e cumprimento das exigências necessárias à obtenção de seus títulos foram adicionadas à *blockchain*. Por exemplo, o médico pode acompanhar se seu diploma foi registrado pela IES consultando pela plataforma *blockchain* implementada. Assim, o mesmo pode realizar os processos posteriores de forma facilitada.

5.5 CONCLUSÕES DO CAPITULO

Neste capítulo apresentou-se toda configuração necessária para aplicação da arquitetura proposta e justificou-se tal abordagem e foram apresentados os benefício em relação a atual abordagem existente nos CRMs.

Primeiramente definiu-se as três entidades que fariam parte da rede para em seguida apresentar as três principais etapas de definição da arquitetura proposta. A primeira etapa definiu como a rede é criada, como são construídas as instancias que fazem parte dessa rede e como elas estão distribuídas pelos participantes existentes. Na segunda Etapa apresentou-se os arquivos lógicos que definem o comportamento da rede, restrições em relação ao que cada participantes pode utilizar e as funções que são executas para adicionar um novo dado a rede. A terceira etapa apresenta a estrutura de comunicação criada para que uma um aplicação *front-end* criada possa se interagir com a *blockchain* desenvolvida e adicionar dados à mesma.

Por fim apresentou-se uma demonstração da solução visando esclarecer seu comportamento. Essa demonstração foi apresentada por meio de um passo-a-passo de como é feito o registro para um médico.

6 IMPLEMENTAÇÃO DA SOLUÇÃO

Este trabalho apresenta um sistema que implementa a solução para os conselhos de medicina. A fim de possibilitar a execução da solução proposta, para a presente Seção optou-se por definir o escopo do sistema *web* responsável pela comunicação das entidades com a *blockchain*. Dessa forma, quatro perspectivas complementares serão discutidas durante esta seção, são elas: *Stakeholders*, Requisitos do Sistema, Projeto do Sistema e Perspectiva Tecnológica. Logo, será apresentado na Seção 6.1 uma breve descrição dos *stakeholders*; na Seção 6.2 os requisitos funcionais e de qualidade do sistema; na Seção 6.3, o projeto do sistema e na Seção 6.4 a perspectiva tecnológica do sistema.

6.1 DESCRIÇÃO DOS STAKEHOLDERS

Os *stakeholders* do sistema *web* podem ser divididos em dois grupos: o administrador, responsável pelo desenvolvimento e manutenção do sistema, e os funcionários, que poderão executar as funcionalidades do sistema de acordo com o que foi definido para cada entidade. Assim, esses funcionários podem ser incluídos em seus respectivos sub-grupos conforme os tipos de participantes da *blockchain* apresentados na Seção 5.1, são eles:

- **Funcionários das IES:** responsáveis por adicionar o diploma do médico recém-formado junto com seus dados para a rede;
- **Funcionários das EE:** responsáveis por adicionar aos dados médicos na rede o registro de especialização para um médico;
- **Funcionários dos CRM:** responsáveis por validar o cadastro médico no CRM;

A princípio, todos os funcionários acessam o mesmo sistema, porém as telas visualizadas são adaptadas de acordo com o sub-grupo ao qual o funcionário pertence.

6.2 REQUISITOS DO SISTEMA

A partir das análises das necessidades identificadas e discutidas na Seção 4, decidiu-se estabelecer os seguintes requisitos funcionais (RF) e de qualidade (RNF) para o projeto. Os requisitos são indicados e referenciados no formato [RFxx] ou [RNFxx], onde xx se refere ao número do requisito. Tais requisitos são descritos na Quadro 4.

Com a finalidade de mostrar uma visão geral das funcionalidades propostas para o sistema, apresenta-se o Diagrama de Caso de Uso na Figura 31 com a descrição dos dois tipos

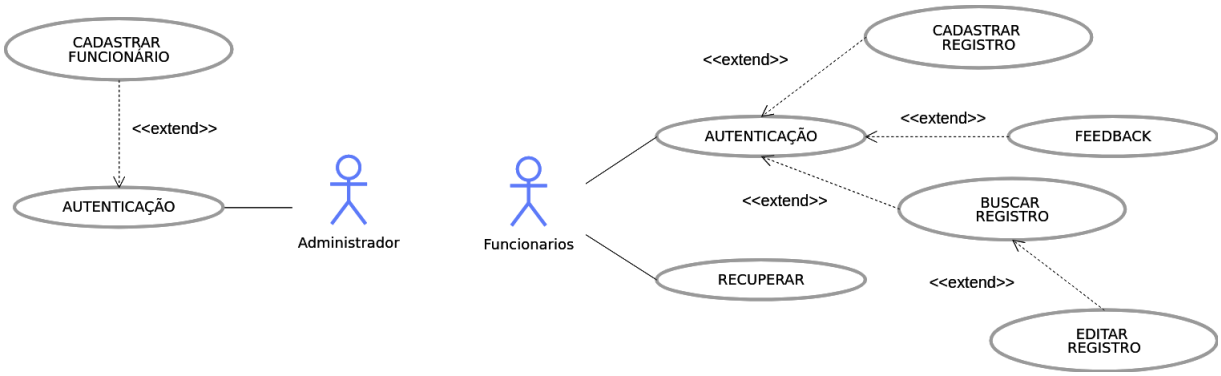
Quadro 4 – Requisitos Funcionais e Não Funcionais do Sistema

ID	Requisito	Descrição	Caso de Uso
RF01	Adicionar Funcionário	Gera uma conta para um funcionário	Cadastrar Funcionário
RF02	Login do Funcionário	Permite que o funcionário entre no sistema	Autenticação
RF03	Manter dados na Blockchain	Permitir que o funcionário possa incluir e listar ds registros presentes na blockchain	Cadastrar e Buscar e Editar Registros
RF04	Recuperar senha do Funcionário	Permite que o funcionário receba uma nova senha para o sistema	Recuperar
RF05	Enviar Feedback	Permite que o funcionário possa enviar feedback do sistema para o administrador	Feedback
RNF01	Compreensão de Usabilidade	O funcionario deverá compreender o funcionamento da aplicação com o mínimo possível de explicações.	
RNF02	Segurança dos Dados	O sistema não apresentará aos funcionários quaisquer dados que não sejam de acordo com seu subgrupo	
RNF03	Telas adaptadas	O sistema deve adaptar as telas após o login de acordo com o grupo pertencente o funcionário	

Fonte – Elaborado pelo autor.

de *stakeholders* apresentados na presente abordagem: administrador e funcionário.

Figura 31 – Diagrama de Casos de Uso do Sistema



Os casos de uso informados na figura acima são detalhados no Apêndice F, com as seguintes informações: nome, atores, prioridade, objetivo, entradas, pré-condições, pós-condições, fluxo normal de eventos, fluxo alternativo de eventos e fluxo excepcional de eventos, de acordo com a Quadro 5.

Primeiramente, ambos os *stakeholders* precisam utilizar o caso de uso autenticar

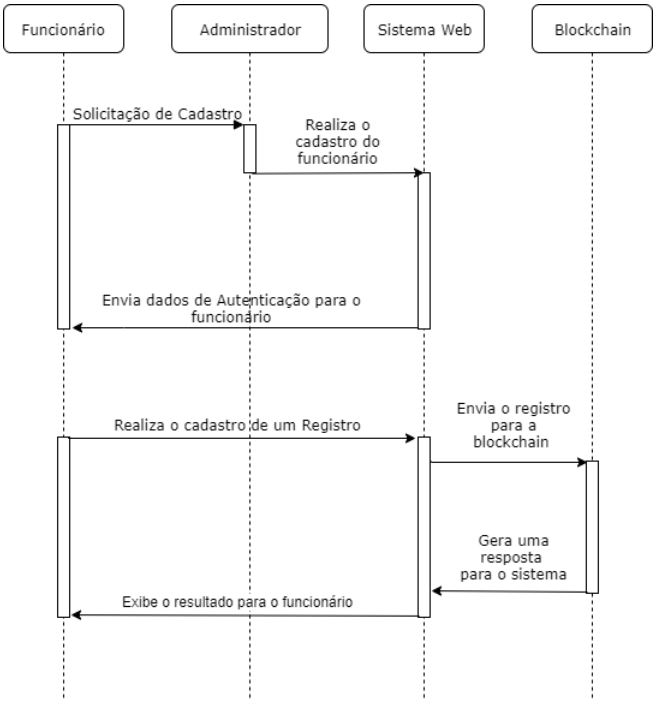
Quadro 5 – Exemplo de documentação dos casos de uso

Nome
Atores
Prioridade
Entradas
Pré-condições
Pós-condições
Fluxo normal de eventos [FN]
Fluxo alternativo de eventos [FA]
Fluxo excepcional de eventos [FE]

Fonte – Elaborado pelo autor.

para poderem logar no sistema. O administrador é responsável por gerenciar os casos de uso referentes à inclusão dos funcionários na rede [Cadastrar Funcionário], enquanto o funcionário está relacionado aos casos de edição de registros na *blockchain* [Cadastrar, Buscar, Editar Registros] e administrativos [Autenticação, Recuperação de Senha]. Também é permitido ao Funcionário enviar *feedbacks* de erro para o administrador, esta função está descrita no caso de uso [Feedback].

Figura 32 – Diagrama de Sequência



Fonte – Elaborado pelo Autor.

A Figura 32 apresenta o diagrama de sequência. Nele são descritas o passo a passo das ações realizadas para que um funcionário possa acessar o sistema e adicionar um registro na *blockchain*. O processo é iniciado com o administrador do sistema acessado o sistema para

cadastrar um funcionário no sistema. Após o funcionário ser registrado no sistema, é permitido a ele realizar o login no sistema e realizar solicitações de adição de registro na *blockchain*. O sistema recebe essa solicitação e encaminha um objeto **json** contendo essas informações para um dos nós da *blockchain*. Quando a informação chega em um dos nós ele é recebido por um servidor REST que realiza uma validação de acordo como apresentado na subseção 5.2.3 e transmite a informação aos demais nós da *blockchain*. Por fim, o servidor REST retorna um objeto **json** resposta para o sistema, alertando do sucesso ou falha da operação, essa resposta então é tratada e exibida para o funcionário por meio de um alerta.

6.3 PROJETO DO SISTEMA

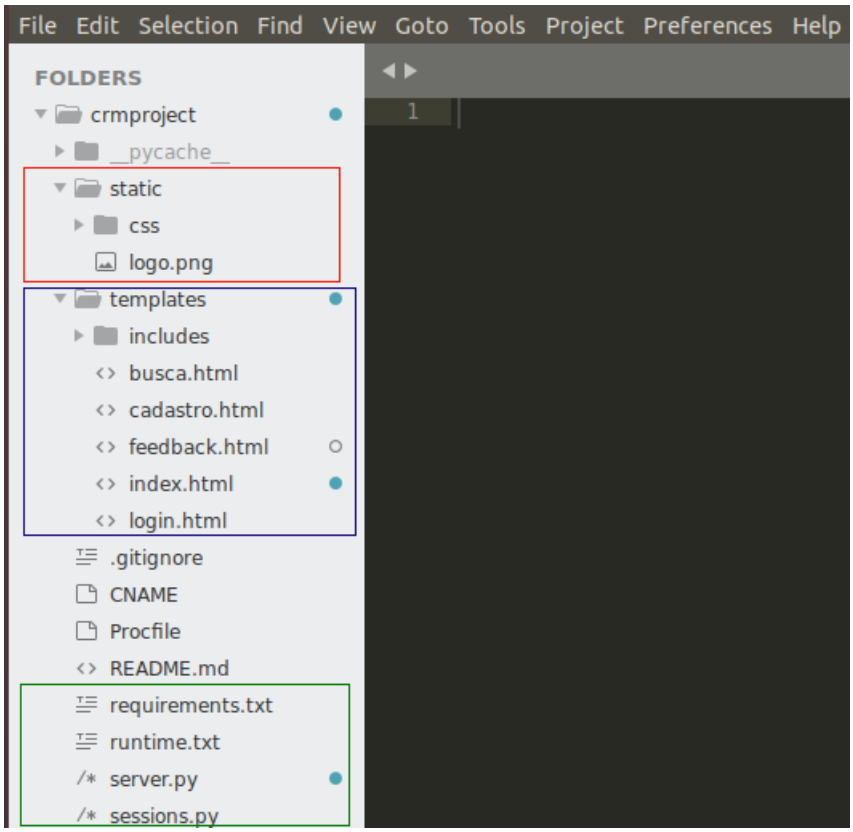
Na Figura 33 esta sendo apresentado o projeto do sistema proposto para comunicar-se com a *blockchain* usando o framework *Flask*. Na parte selecionada pelo quadrado azul, tem-se a pasta *templates* e quatro arquivos compondo-a. Nesta pasta estão implementadas as páginas responsáveis pelas telas utilizadas pelos usuários. Enquanto que o *css*, que abriga a implementação do estilo da interface das páginas, está armazenado na pasta *static* selecionada pelo quadrado vermelho.

A página *Login* é a responsável por comunicar-se com o banco e logar o *stakeholder* de acordo com o tipo definido. Caso o *stakeholder* tenha esquecido sua senha, é possível recuperá-la também por essa página, reenviando uma senha nova para o e-mail.

Em relação as páginas *Buscar* e *Cadastro*, estas funcionam como um canal de comunicação com a *blockchain*. A primeira apresenta informações sobre os dados contidos na *blockchain* de acordo com o perfil de *stakeholder* que está acessado o sistema, ou seja, caso um Funcionário de uma IES acesse o sistema, será buscado apenas dados referentes da IES cadastrada. Em relação à segunda, faz a solicitação de um cadastro na *blockchain*, enviando os dados por meio de um objeto json.

Além das pastas e páginas apresentadas é necessário também os arquivos que configuram o servidor da aplicação. Pode-se destacar quatro arquivos selecionados pelo quadrado verde: *requirements*, *runtime*, *server* e *session*. *Requirements* contém a lista de bibliotecas que estão sendo utilizadas no projeto, *runtime* define qual versão do python está sendo utilizada pelo projeto e *session* é responsável por fazer o controle da sessão de cada usuário. Por fim, o arquivo *server* é o responsável pelo servidor, nele se faz a gestão das páginas e comunicação com o banco de dados de usuários que podem logar no sistema.

Figura 33 – Projeto do sistema proposto



Fonte – Elaborado pelo autor.

Figura 34 – Banco de Dados do Sistema Web

Fonte – Elaborado pelo autor.

O sistema *web* ainda utiliza o *framework Firebase* como Sistema de Gerenciamento de Banco de Dados (SGBD) o qual está sendo mostrado na Figura 34. Este armazena todas as informações referentes aos funcionários de forma que cada funcionário está relacionado a uma chave única e inalterável criada pelo SGBD. Além disso, como mostrado na Figura 35, o gerenciamento de autenticação dos funcionários também é feito pela própria plataforma do *Firebase* sendo estes definidos para serem autenticados apenas por e-mail. Logo, os dados de autenticação, login e senha, disponibilizados para o funcionário pelo administrador estão cadastrados no banco de dados e só é permitido fazer login no sistema se os dados apresentados na tela específica para este fim forem válidos.

Figura 35 – Sistema de Autenticação

Fonte – Elaborado pelo autor.

6.4 PERSPECTIVA TECNOLÓGICA

Além das tecnologias apresentadas na Seção 5. Nesta seção são apresentadas as principais ferramentas e tecnologias utilizadas no desenvolvimento do sistema web que estará integrado com a *blockchain*.

6.4.1 JSON

JSON é uma formatação usual atualmente. É simples tanto para humanos compreenderem, como máquinas. Ele é baseado em um subconjunto da Linguagem de Programação JavaScript e é um formato de texto completamente independente do idioma, mas usa convenções que são familiares aos programadores de C, C++, C#, Java, JavaScript, Perl, Python e muitos outros. Usando a Application Programming Interface (API) específica, esse formato pode ser acessado por qualquer linguagem e utiliza o modelo atributo/valor para representar os dados.

6.4.2 HTML e CSS

HTML é comumente utilizado para a criação de páginas online e aplicações da WEB. Os navegadores atuais recebem documentos em HTML que são processados renderização e apresentação do conteúdo online. O nome HTML é uma abreviação para a expressão inglesa de HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto. Em relação ao CSS, trata-se de um simples mecanismo para adicionar estilo (cores, fontes, espaçamento, etc.) ao documento HTML. Em vez de colocar a formatação dentro do HTML, o CSS cria um link para a página que contém os estilos. Quando quiser alterar a aparência do projeto, basta modificar apenas o arquivo CSS.

6.4.3 Heroku

Heroku é uma plataforma em nuvem como serviço (PaaS) que suporta várias linguagens de programação. A rede Heroku executa os aplicativos do cliente em contêineres virtuais "Dynos"em um ambiente de tempo de execução confiável. Esses Dynos podem executar códigos escritos em diversas linguagens, tais como Python, por exemplo.

6.4.4 Firebase

O Firebase é um SGBD que facilita a manipulação de informações em aplicativos e sistemas *web*. Ele pode ser usado sem custo e possui funcionalidades como: banco de dados em tempo real, autenticação, armazenamento, hosting, entre outros. Ele funciona como um grande arquivo JavaScript Object Notation (JSON) que pode ser gerenciado com uma simples API fornecendo ao aplicativo o valor atual dos dados e qualquer atualização deles

6.4.5 Flask

Flask é um micro-*framework* para Python, que provê um modelo simples para desenvolvimento web. Uma vez importando no Python, Flask pode ser usado para economizar tempo de construção nas aplicações web.

6.4.6 Gunicorn

O Gunicorn é um servidor HTTP amplamente compatível com as várias estruturas da Web implementado de maneira simples e rápida os recursos de um servidor. Quando instalado, um comando gunicorn é disponibilizado para iniciar o processo servidor. Na sua forma mais simples, o gunicorn somente precisa ser chamado com a localização do módulo contendo o objeto da aplicação nomeado app. Neste caso para chamar o servidor localmente no python deve-se usar o comando: **gunicorn server:app**

6.5 CONCLUSÕES DO CAPITULO

Neste capítulo definiu-se o escopo implementado para o sistema *web* responsável pela comunicação com a *blockchain* por meio de quatro perspectivas complementares.

Inicialmente definiu-se os tipos de *stakeholders* responsáveis pela utilização do sistema, que podem influenciar ou serem impactados pelo mesmo. Logo após, apresentou-se o levantamento dos requisitos definidos para o sistema (Funcionais e de Qualidade) e como eles interagem com os casos de uso definidos para o sistema. Em seguida, foi apresentado o do sistema, toda a relação de arquivos que compõe o projeto e por fim foram listadas todas as tecnologias utilizadas no desenvolvimento do sistema *web*.

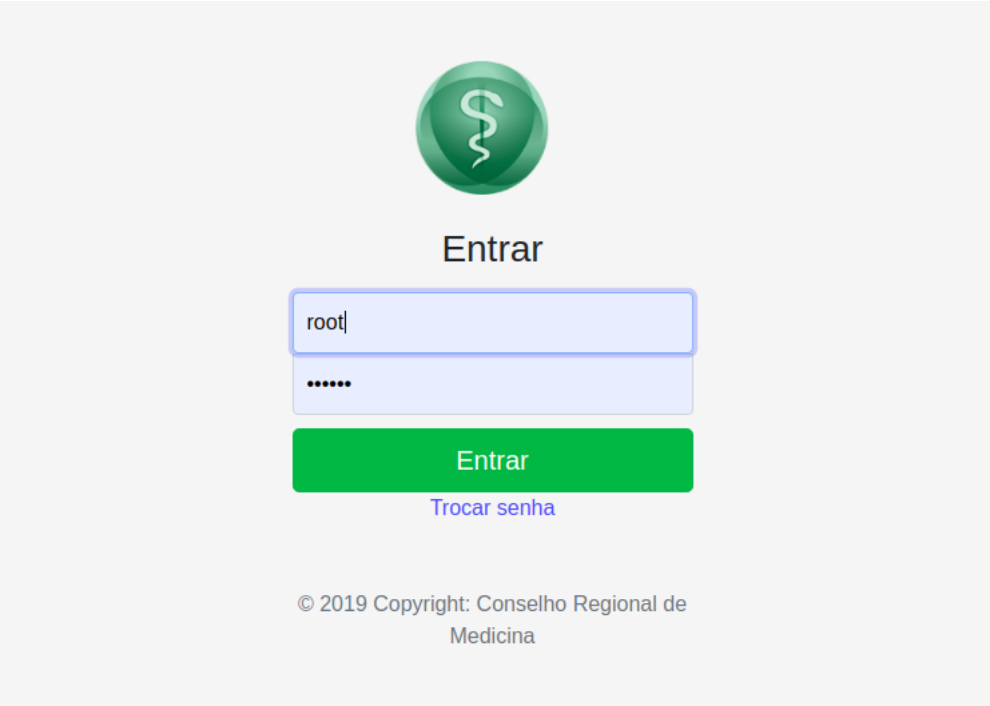
7 DEMONSTRAÇÃO DO SISTEMA

Este capitulo apresenta uma demonstração do sistema desenvolvido utilizando as ferramentas apresentadas na Seção 6 e seguindo a solução apresentado no Seção 5. Para facilitar a visualizações dos resultados, este capitulo é divido em três seções. As duas primeiras correspondem a uma apresentação dos diferentes *stalkeholders* que usarão o sistema e suas responsabilidades, visando apresentar diferentes cenários de uso do sistema. Na ultima seção é apresentada uma avaliação de desempenho da solução.

7.1 ADMINISTRADOR

A função de administrador é atribuída a um ator que fará o gerenciamento da aplicação. Sua principal responsabilidade trata-se de adicionar o *stakeholder* funcionário no sistema, ou seja, somente poderá acessar o sistema, o funcionário cujo cadastro foi adicionado pelo ator administrador. Primeiramente, após uma solicitação de cadastro por um funcionário, o administrador entrará no sistema utilizando um login e senha pré-definidos, como mostrado na Figura 36.

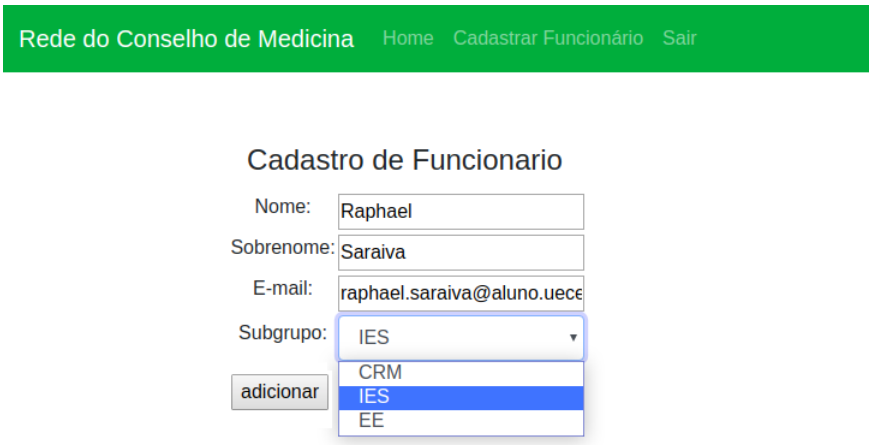
Figura 36 – Tela inicial do sistema



Fonte – Elaborado pelo autor.

A Figura 37 apresenta a tela de cadastro do funcionário. Esta permite a adição das seguintes informações: nome, sobrenome, e-mail e sub-grupo. O campo e-mail define para qual e-mail será enviado a senha provisória para o novo funcionário e o campo sub-grupo define que tipo de funcionário está sendo cadastrado.

Figura 37 – Tela de Cadastro do Funcionário



Fonte – Elaborado pelo autor.

A partir da senha encaminhada e utilizado o e-mail informado para o administrador, o funcionário já podem utilizar o sistema.

7.2 FUNCIONÁRIO

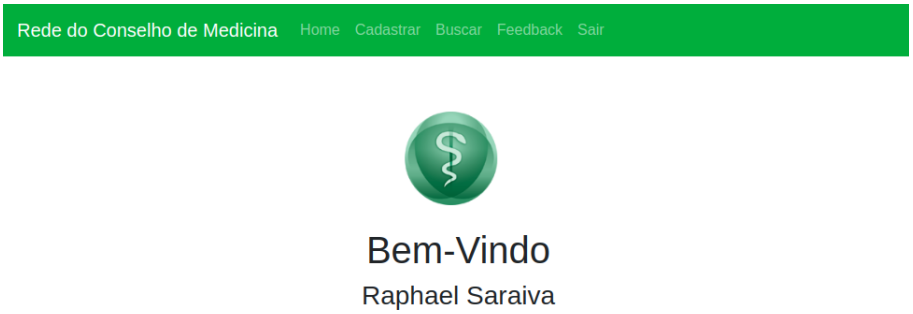
Todos os funcionários cadastrados tem um ator denominado funcionário que faz o gerenciamento do domínio ao qual está relacionado. Nas subseções a seguir será mostrada a descrição de como é processada a solicitação de registro do ponto de vista de dois tipos de funcionário: um funcionário de uma IES e o funcionário do CRM. O primeiro demostrará como um diploma é adicionado na *blockchain* enquanto o segundo mostrá como a solicitação de registro de um médico é realizado. Por fim, é demostrado a tela de *feedback* ao administrador.

7.2.1 Funcionário da IES

O funcionário do IES poderá realizar o login com os dados recebidos do administrador acessando a mesma tela apresentado na Figura 36. Caso ele deseje alterar senha, poderá altera-lá clicando no link apresentado abaixo do botão de login. Nesse caso um e-mail será

encaminhado pelo banco, por onde ele poderá trocar a senha. Na Figura 38 é apresentada a tela principal do aplicativo sendo esta visível independentemente do sub-grupo ao qual o funcionário pertence.

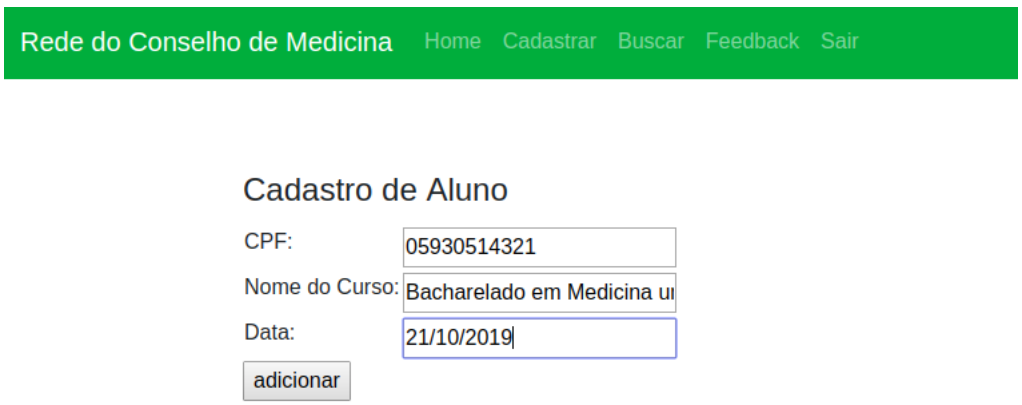
Figura 38 – Tela principal



Fonte – Elaborado pelo autor.

Na Figura 39 é exibida a tela de realização de cadastro do diploma para a *blockchain*, esta tela é ajustada de acordo com o sub-grupo que o funcionário pertence. A tela em questão é exibida de forma personalizada para que o funcionário da IES possa preencher com os dados do diploma que deseja enviar para a para blockchain.

Figura 39 – Tela de cadastro de aluno



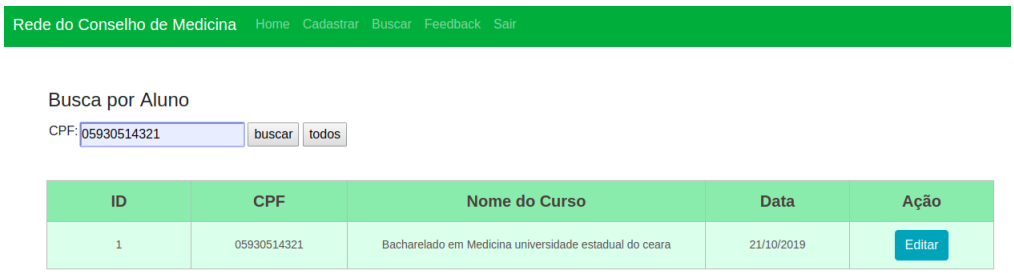
Fonte – Elaborado pelo autor.

Após preencher os campos e clicar em adicionar, será estabelecida uma conexão

entre o servidor da aplicação web e a blockchain. Se não correu nenhum erro, será exibida um alerta notificando que o dado foi adicionado e uma mensagem com o **id** do diploma adicionado é encaminhado para o e-mail do aluno cadastrado. Entretanto, caso ocorra algum erro, um alerta é disparado informando o erro ocorrido.

Realizado o cadastrado do registo do diploma de um aluno na *blockchain*, é possível que o funcionário do IES possa realizar a busca de um ou mais registros adicionados por meio da tela de busca apresentada na Figura 40. Por meio dela, o funcionário digita o **cpf** pertencente ao aluno cujo diploma está procurando. O sistema *web* realizará uma requisição de consulta para a blockchain e retornará as informações do aluno por meio de uma tabela.

Figura 40 – Tela de busca por aluno



Fonte – Elaborado pelo autor.

É possível ainda por meio do botão de editar exibido na coluna de ações, alterar os dados do aluno caso seja necessário, por meio desse botão será exibida uma tela semelhante a tela de cadastro da Figura 39 onde os dados do aluno podem ser editados e salvados novamente. Eventualmente, um cenário para um funcionário de uma EE seguiria os passos apresentados nessa sub-seção, sendo a Tela de Cadastro adaptada para o nome da especialização.

7.2.2 Funcionário do CRM

O funcionário do CRM realizará os mesmos passos de login como na Figura 36 e acessar a tela principal como na Figura 38. Entretanto a tela de castrado de registro médico apresentada na Figura 41 difere da tela exibida para o Funcionário de uma IES. Nessa tela, o Funcionário do CRM irá adicionar o dados referentes ao registro médico que deseja adicionar.

Para o registro do médico, no momento do cadastro o funcionário do CRM deve adicionar o id do diploma referente ao cpf do médico que deseja registrar. Alternativamente, caso o médico que está sendo sendo cadastrado, tenha especializações, é possível repassar o id

Figura 41 – Tela de cadastro dos dados médico

Fonte – Elaborado pelo autor.

de uma ou mais especializações de que o médico tenha, sendo esse campo não obrigatório.

O funcionário do CRM pode realizar a busca por registros de médicos cadastrados. Neste caso como mostrado na Figura 42, o Funcionário do CRM realiza a busca pelo CPF do médico cadastrado assim como o funcionário do IES e visualizar os dados por meio de uma tabela. Também é possível editar as informações no registro do medico por meio do botão de editar, exibido na coluna de ações. Entretanto, diferentemente da tela de busca do funcionário da IES, também é possível visualizar as informações referentes ao diploma e especialização por meio do botão exibir presente em suas respectivas colunas.

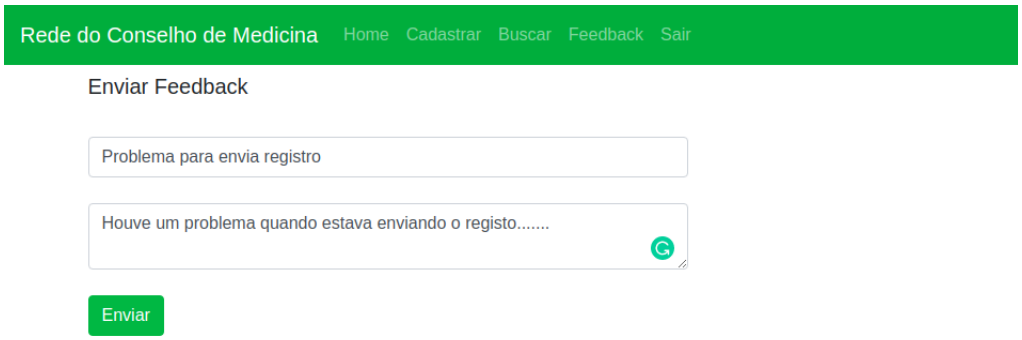
Figura 42 – Tela de Busca de médico

Fonte – Elaborado pelo autor.

7.2.3 Feedback

Independentemente do sub-grupo ao qual o funcionário pertence, todos podem enviar uma mensagem de *feedback* para o administrado do sistema. A Figura 43 apresenta a tela de *feedback* com um exemplo de uma possível mensagem que os funcionários podem enviar para o Administrador. Após clicar no botão enviar, essa mensagem é encaminhada para o e-mail cadastrado na conta de administrador do sistema. Dessa forma, todos os funcionários tem a possibilidade de se comunicar com o ator administrador, para informar eventuais erros, ou dar sugestões de melhorias para o sistema.

Figura 43 – Tela de *feedback*



Fonte – Elaborado pelo autor.

7.3 AVALIAÇÃO DE DESEMPENHO DO SISTEMA

A *blockchain* pode ser considerada um gargalo no desempenho da solução proposta, uma vez que para a aplicação funcionar corretamente é necessário que a *blockchain* tenha um tempo de resposta razoável independentemente dos fatores que possam afetar seu desempenho. Portanto nesta subseção foi projetado um experimento buscando analisar o desempenho da solução apresentada, avaliando seu tempo de resposta.

Como apresentando no Quadro 8, para esse experimento foram definidas duas variáveis independentes: tipo de solicitação e número de clientes e uma variável dependente: latência de resposta. A latência de resposta que busca-se analisar neste trabalho é considerada a variável dependente do experimento, pois seu valor depende de como as variáveis independentes apresentadas estão sendo manipuladas.

Como variável independente, tem-se o **número de clientes** que estão fazendo as

Quadro 6 – Variáveis do Experimento

Variáveis Independentes		Variáveis dependentes
Tipo de requisição	Número de Clientes	
GET	1	Latência de Resposta
	50	
	100	
	150	
	200	
	400	
POST	1	
	50	
	100	
	150	
	200	
	400	

Fonte – Elaborado pelo autor.

requisições. A medida que o número de clientes aumenta, podem afetar o desempenho do sistema fazendo este diminuir. Por esse motivo, o número de clientes foi selecionado como uma variável independente analisada neste experimento. Em relação a variável independente **tipo de requisição**, esta pode afetar diretamente o desempenho do sistema dependendo do tipo de requisição escolhido. Existem dois tipos de requisição HTTP relacionados ao aplicativo atual: **GET** e **POST**, responsáveis pela pela leitura e escrita dos dados médicos na *blockchain* respectivamente. Mais especificamente, o método **GET** tende a ser mais rápido, porque está apenas adquirindo informações da *blockchain*, o que não altera o estado do livro-razão. Diferentemente deste, o método **POST** requer a gravação de informações no *blockchain*, mudando o estado do livro-razão, o que o torna mais lento.

7.3.1 Configurações do Experimento

Como a principal ferramenta de avaliação de desempenho foi usado JMeter 5.0 (NEVEDROV, 2006). Este trata-se de um software de código aberto, projetado para testar o comportamento e medir o desempenho de aplicações. No caso em questão é usando para simular uma carga e enviar para a blockchain visando analisar seu desempenho. Um computador com Windows foi usado como simulador do cliente. A especificação do computador é mostrada no Quadro 7. Em relação ao nó da *blockchain*, este foi implantado no servidor da amazon através da ferramenta *Amazon Web Service* (AWS). A especificação da máquina na nuvem é mostrada no Quadro 8.

Este nó também foi configurado para não exigir as credenciais necessárias apresentada na Subseção 5.2.3, afim de facilitar o processo de avaliação do desempenho. Por fim, o

Quadro 7 – Especificações do Computador do Cliente

Sistema Operacional	Ubuntu 16.04 LTS
Modelo do Computador	Dell Inspiron 3000
Processador	Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz ×4
Memoria	4.0GB
Tipo do Sistema	64-bit, x64-based processor

Fonte – Elaborado pelo autor.

Quadro 8 – Especificações do Servidor AWS

Sistema Operacional	Ubuntu 18.04 LTS
Tipo de Maquina	n1-highmem-2 (2 vCPUs, 13 GB memory)
CPU	Intel Haswell (Intel Xeon E5 v3 @2.3GHz)
Zona	us-east-2c

Fonte – Elaborado pelo autor.

experimento foi configurado usando a interface gráfica do JMeter e para todas as medições foi utilizado um contador de interação definido como 1s.

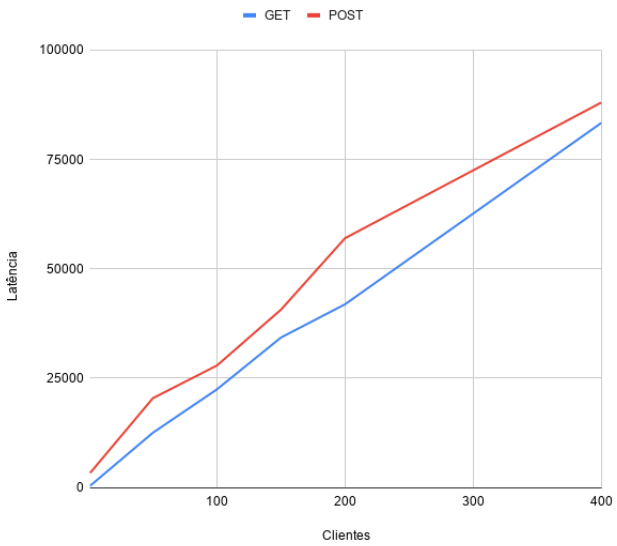
7.3.2 Resultado

O desempenho geral é mostrado na Figura 44. A latência para uma requisição do tipo **GET** para 1 cliente é de 415 ms, enquanto a requisição **POST** é de 3352 ms. As requisições do tipo **POST** são mais lentas quando comparadas com as requisições do tipo **GET** pode ser vista em todos os casos independentemente da variação no número de clientes. Isso confirmou a previsão de que as requisições **POST** são mais lenta em comparação com a **GET**, pois exigem que novas informações sejam gravadas no *blockchain* e atualize o status do livro-razão, o que ocasiona um tempo extra.

Pode-se perceber também que à medida que o número de clientes aumenta, a latência aumenta linearmente. Para a solicitação **GET**, há um aumento linear quase perfeito na latência à medida que o número de clientes aumenta, onde a latência aumentou de 415 ms para 83355 ms. A solicitação **POST** mostrou uma tendência semelhante, onde a latência aumentou de 3352 ms para 88015 ms. Assim, as requisições **GET** e **POST** confirmaram a segunda previsão, que à medida que o número de clientes cresce linearmente, o desempenho diminui (ou seja, aumenta a latência da resposta), considerando que cada cliente adicional precisa levar uma certa quantia de tempo e recursos.

Além disso, a diferença geral entre a requisição **POST** e **GET** é mostrada na Figura 45. Esta foi adquirida subtraindo a latência geral das requisições **GET** da latência geral da requisições de **POST**. A diferença média geral é aproximadamente 389 ms com o Desvio Padrão de aproximadamente 3899 ms, sendo esta a estimativa de tempo que a *blockchain*

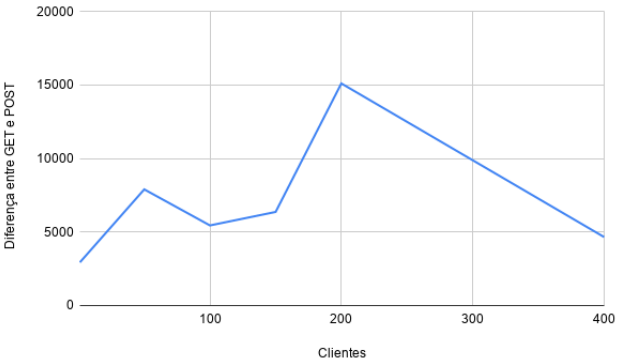
Figura 44 – Desempenho Geral do Sistema



Fonte – Elaborado pelo autor.

leva para gravar informações e alterando o livro-razão. Caso um novo nó seja adicionado, ele funcionará da mesma forma que esse nó, pois diferentes nós são executando paralelamente e independentemente, motivo pelo qual o número de nós não foi considerado como uma variável dependente do sistema.

Figura 45 – Diferença geral entre a requisições **POST** e **GET**



Fonte – Elaborado pelo autor.

As latências de repostas apresentadas são relativamente boas. Com o número de clientes aumentado, a latência aumenta, porém com 200 clientes fazendo solicitações ao mesmo tempo, o desempenho ainda está abaixo de 1 minuto e com 400 clientes, o desempenho ainda está abaixo de 2 minutos. Isso pode ser adequado para uma rede como o CRM, uma vez que os dados médicos não são adicionados com uma frequência muito alta. A latência média para

o sistema blockchain gravar informações em blockchain e alterar o livro-razão é inferior a 8 segundos dentro do intervalo de tempo para os clientes testados. Isso é significativamente mais rápido comparado aos as *blockchains* públicas como o bitcoin por exemplo, onde leva-se cerca de 10 minutos para concluir uma transação.

8 CONSIDERAÇÕES FINAIS

Neste capítulo serão feitas as considerações finais deste trabalho de dissertação. Serão apresentadas as principais contribuições da pesquisa bem como algumas limitações da mesma. Posteriormente são apontadas direções de pesquisa para trabalhos futuros.

8.1 CONTRIBUIÇÕES

As responsabilidades éticas dos profissionais de medicina no Brasil são indicadas pelo Código de Ética Médica (CEM), que trata a relação entre médicos e pacientes, assim como seus direitos e deveres. Apesar desses deveres serem descritos no CEM, são inúmeros os casos onde corre a atuação de ilegal de profissionais. Diante disso, faz-se necessário a atuação de órgãos de fiscalização, como os Conselho Regional de Medicina (CRM), encarregados por garantir que as normas do CEM sejam cumpridas. Tais órgãos armazenam de maneira centralizada, um registro de todos médicos, procurando assim manter sempre um controle de como os médicos estão atuando na sociedade.

A motivação desta pesquisa adveio da possibilidade que o uso da tecnologia do *blockchain* podem proporcionar uma maior segurança necessária para as informações evitando, assim, fraudes e infortúnio decorrentes da quebra de segurança. Além disso, o uso de tal tecnologia promove a divulgação adequada dessas informações e fomenta à cultura da transparência. Portanto, o principal objetivo tratou-se de desenvolver, utilizando a tecnologia *blockchain*, uma solução que permite, ao público de interesse, uma maneira de armazenar as informações relevantes, referentes à inscrição no CRM dos profissionais da medicina, de modo descentralizado e confiável, dada as vantagens que a tecnologia proporciona para a manipulação e rastreabilidade desses dados.

A principal contribuição deste trabalho consiste na proposta de uma solução baseada em uma *blockchain* permissionada para o controle dos registros profissionais de médicos. Nela é apresentado um mecanismo que permite a integração de diferentes entidades por meio de uma *blockchain* usando *Hyperledger*. Consequentemente, a contextualização relacionada ao desenvolvimento de uma solução através do *Hyperledger* também se manifesta como uma importante contribuição, haja vista o relevante papel da referida tecnologia para o mercado e academia.

Finalmente, a solução proposta neste trabalho visa facilitar o processo de inscrição de

um novo médico a medida que mantém um controle rígido, evitando assim, possíveis fraudes que venham a acontecer devido a problemas durante o processo de registro profissional. Por tratar-se de uma nova forma de armazenamentos dos registros profissionais de médicos, buscou-se obter uma validação da solução através da realização de uma entrevista semi-estruturada com um membro da diretoria do CREMEC.

8.2 LIMITAÇÕES

Acredita-se que todos os objetivos listados por esta pesquisa foram atingidos ao longo do seu desenvolvimento. Apesar disso, o trabalho ainda apresenta algumas limitações tais como:

- Ausência de uma avaliação de desempenho da solução, buscando validar a solução em um cenário de uso. Por exemplo, o uso de métricas que avaliam a robustez do sistema.
- A falta de uma visão dos usuários finais da solução em termos de usabilidade do sistema.
- Baixo número de entrevistados na validação da solução.

8.3 TRABALHOS FUTUROS

Este trabalho foca no uso de uma blockchain para a gestão de registro de profissionais médicos e suas consequências. Porém é um assunto bastante novo e complexo; portanto ainda há muitos podem que merecem ser exploradas a partir das ideias aqui apresentadas. Portanto, em relação aos trabalhos futuros, constituem-se oportunidades para trabalhos futuros:

- Validar a proposta empiricamente e obter uma visão da usabilidade da solução operando em diferentes contextos.
- Adaptar a solução proposta para um contexto de pessoa jurídica, adicionando um registro para serviços públicos e privados como: clinicas, hospitais, etc.
- Ajustar a solução proposta para uma ambiente multi-organizacional permitindo a criação de sub-redes para cada participante.
- Evoluir a aplicação permitindo a possibilidade de acesso via dispositivos móveis, através da integração com tecnologias de desenvolvimento *Mobile*.

REFERÊNCIAS

ANDROULAKI, E.; BARGER, A.; BORTNIKOV. A distributed operating system for permissioned blockchains. In: EUROSYS CONFERENCE,18, 2018, Porto, Portugal. **Anais...** New York, NY, USA: ACM, 2018. p. 30:1–30:15.

ANTONOPOULOS, A. M. **Mastering Bitcoin: unlocking digital cryptocurrencies**. USA: O'Reilly Media, Inc., 2014. 298 p.

ANTONOPOULOS, A. M.; WOOD, G. **Mastering Ethereum: Building Smart Contracts and Dapps**. USA: O'Reilly Media, Inc., 2004. 415 p.

AXON, L. Privacy-awareness in blockchain-based pki. **University of Oxford Journal**, v. 3, n. 1, p. 1–6, 2015.

BESSANI, A.; SOUSA, J.; ALCHIERI, E. State machine replication for the masses with bft-smart. In: 44TH ANNUAL IEEE/IFIP INTERNATIONAL CONFERENCE ON DEPENDABLE SYSTEMS AND NETWORKS,44, 2014, Atlanta, GA, USA. **Anais...** New York, NY, USA: IEEE, 2014. p. 355–362.

BLUMMER, T.; BOHAN, S. **An Introduction to Hyperledger**. Mountain View, California: Creative Commons, 2018. 33 p.

BOYCE, C.; NEALE, P. Conducting in-depth interviews: A guide for designing and conducting in-depth interviews for evaluation input. **IEEE Software**, v. 1, n. 1, p. 1–16, Aug 2006.

BRASIL. Conselho Federal de Medicina. **Lei No 3.268/57.** Disposição sobre os Conselhos de Medicina, e das outras providências, Portal do Planalto, Brasília, [199-]. Disponível em: <http://www.planalto.gov.br/Leis/L3268.html>. Acesso em: 15 jan. 2019.

BRASIL. Conselho Federal de Medicina. **Parecer CFM No 5/2017.** Dispõe sobre os tipos de pós-graduações médicas lato sensu no Brasil, Portal CFM, Brasília, [199-]. Disponível em: <https://sistemas.cfm.org.br/normas/ arquivos/pareceres/BR/2017/5_2017.pdf>. Acesso em: 02 jan. 2019.

BRASIL. Conselho Federal de Medicina. **Resolução CFM No 2.217/2018.** Aprova o Código de Ética Médica, Portal CFM, Brasília, [199-]. Disponível em: <https://sistemas.cfm.org.br/normas/visualizar/resolucoes>. Acesso em: 07 jan. 2019.

BRASIL. Conselho Federal de Medicina. **Resolução No 1.541/98.** Aprova o Estatuto para o Conselho de Medicina, Portal CFM, Brasília, [199-]. Disponível em: <https://portal.cfm.org.br/images/stories/documentos/resolucocfmn1541estatutodosconselhos.pdf>. Acesso em: 02 jan. 2019.

BRASIL. Decreto-lei. **Lei No 12.527.** Regula o acesso a informações previsto no inciso XXXIII do art. 5o, Portal do Planalto, Brasília, [199-]. Disponível em: <http://www.planalto.gov.br/ccivil_03.htm>. Acesso em: 02 jan. 2019.

BRASIL. Decreto-Lei. **Lei No 2.848/40.**. Código Penal Brasileiro, Portal da Câmara dos Deputados, Brasília, [199-]. Disponível em: <http://www2.camara.leg.br/legin/fed/declei/1940-1949/decreto-lei-2848-7-dezembro-1940-412868-publicacaooriginal-1-pe.html>. Acesso em: 15 jan. 2019.

BUCHMAN, E. **Tendermint: Byzantine fault tolerance in the age of blockchains.** 2016. 134 f. Tese (Doutorado) — The University of Guelph, Canada, 2016.

CACHIN, C. Architecture of the hyperledger blockchain fabric. In: WORKSHOP ON DISTRIBUTED CRYPTOCURRENCIES AND CONSENSUS LEDGERS,16, 2016, Chicago, USA. **Anais...** New York, NY, USA: IEEE, 2016. p. 1–4.

DAVIDSON, S.; FILIPPI, P. D.; POTTS, J. Economics of blockchain. **HAL**, v. 1, n. 1, p. 1–24, Nov 2016.

FERREIRA, J. E. **Blockchain para Criação de Novos Modelos de Negócio e Seus Impactos na Indústria de Serviços Financeiros, 2016,** 51 f. Tese (Doutorado) — Universidade Federal de Pernambuco, Recife, Brazil, 2016.

GARAY, J.; KIAYIAS, A.; LEONARDOS, N. The bitcoin backbone protocol: Analysis and applications. In: ANNUAL INTERNATIONAL CONFERENCE ON THE THEORY AND APPLICATIONS OF CRYPTOGRAPHIC TECHNIQUES, 2015, Sofia, Bulgaria. **Anais...** Berlin, Heidelberg: Springer, 2015. p. 1–46.

GMYTRASIEWICZ, P. J.; DURFEE, E. H. Decision-theoretic recursive modeling and the coordinated attack problem. In: PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE PLANNING SYSTEMS, 1992, College Park, Maryland. **Anais...** Alpharetta, USA: Elsevier, Alpharetta, USA. p. 88–95.

GODOY, A. S. Uma revisão histórica dos principais autores e obras que refletem esta metodologia de pesquisa em ciências sociais. **Revista de Administração de Empresas**, v. 23, n. 2, p. 57–63, Out 1995.

GRINBERG, M. **Flask web development: developing web applications with python**. 1. ed. Califórnia, EUA: O'Reilly Media, 2018. 316 p.

HAMMER-LAHAV, D.; HARDT, D. The oauth2 authorization protocol. **IETF Internet Draft**, v. 1, n. 1, p. 1–10, Jun 2011.

HAMMER-LAHAV, D.; HARDT, D. Internet of things, blockchain and shared economy applications. **Elsevier**, v. 98, n. 1, p. 461–466, Jul 2016.

HEARN, M. Corda: A distributed ledger. **R3**, v. 1, n. 1, p. 1–10, Jan 2016.

HOEPMAN, J.-H. Distributed double spending prevention. In: INTERNATIONAL WORKSHOP ON SECURITY PROTOCOLS, 2007, Sofia, Bulgaria. **Anais...** Berlin, Heidelberg: Springer, 2015. p. 1–46.

KRISHNAN, S.; GONZALEZ, J. L. U. **Building Your Next Big Thing with Google Cloud Platform: A Guide for Developers and Enterprise Architects**. 2. ed. New York, EUA: Appress, 2015. 396 p.

LEE KIBIN JAMES, T. G.; KIM, H. J. Electronic voting service using blockchain. **Journal of Digital Forensics, Security and Law**, v. 11, n. 2, p. 8–12, Jan 2016.

LIANG, X.; SHETTY, S.; TOSH, D.; KAMHOUA, C.; KWIAT, K.; NJILLA, L. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: PROCEEDINGS OF THE 17TH IEEE/ACM INTERNATIONAL SYMPOSIUM

ON CLUSTER, CLOUD AND GRID COMPUTING, 2015, Sofia, Bulgaria. **Anais...** New York, NY, USA: IEEE, 2017. p. 468–477.

MARTINO, W. The first scalable, high performance private blockchain. **sammantics**, v. 5, n. 2, p. 61–66, Dez 2016.

MCCONAGHY, T.; MARQUES, R.; MULLLER, A. Bigchaindb: a scalable blockchain database. **Journal of Digital Forensics, Security and Law**, v. 3, n. 1, p. 1–12, Nov 2016.

MENDANHA, G. O.; CRUZ, L. A.; MAGALHAES, R. Uma ferramenta para assegurar a propriedade e imutabilidade de documentos digitais. In: SIMPóSIO BRASILEIRO DE BANCOS DE DADOS, 2016, Uberlândia, Brazil. **Anais...** New York, NY, USA: IEEE, 2016. p. 157–167.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. v. 1, n. 1, p. 1–10, Nov 2008.

NARAYANAN, A.; BONNEAU, J.; FELTEN, E.; MILLER, A.; GOLDFEDER, S. **Bitcoin and cryptocurrency technologies: a comprehensive introduction**. 1. ed. Nova Jersey, EUA: Princeton University Press, 2016.

NEVEDROV, D. Using jmeter to performance test web services. **dev2dev**, v. 1, n. 1, p. 1–11, fev 2006.

NORTON, S. Cio explainer: What is blockchain. **The Wall Street Journal**, v. 2, n. 1, p. 1–10, Dez 2016.

ONGARO, D.; OUSTERHOUT, J. In search of an understandable consensus algorithm. In: USENIX ANNUAL TECHNICAL CONFERENCE,14, 2014, Philadelphia, PA. **Anais...** Berkeley, Califórnia, EUA: USENIX, 2014. p. 305–319.

PIOVESAN, A.; TEMPORINI, E. R. Pesquisa exploratória: procedimento metodológico para o estudo de fatores humanos no campo da saúde pública. **Revista de Saúde Pública**, v. 29, n. 4, p. 318–325, Aug 1995.

PIRES, M.; SOUZA, D.; COSTA, R.; LEMOS, G. Uma abordagem baseada em brokers para registro de transações em multiplos livros-razão distribuidos. In: I WORKSHOP EM BLOCKCHAIN: TEORIA, TECNOLOGIAS E APLICAçõES (WBLOCKCHAIN-SBRC 2018),1, 2018, Campos do Jordão, São Paulo. **Anais...** Porto Alegre, Rio Grande do Sul, Brasil: SBC, 2018. p. 14.

PORRU, S.; PINNA, A.; MARCHESI, M.; TONELLI, R. Blockchain-oriented software engineering: challenges and new directions. In: 39TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING COMPANION (ICSE-C),39, 2017, Buenos Aires, Argentina. **Anais...** New York, NY, USA: IEEE, 2017. p. 169–171.

PRESSMAN, R. S. **Sistemas Distribuidos - Conceitos e Projeto**. 8. ed. Porto Alegre: Bookman Editora, 1995. 790 p.

ROBSON, C.; MCCARTAN, K. **Real world research**. 1. ed. Nova Jersey: John Wiley & Sons, 2016. 560 p.

RUNESON, P.; HÖST, M. Guidelines for conducting and reporting case study research in software engineering. **Empirical software engineering**, v. 14, n. 2, p. 131, Dec 2008.

SCHWARTZ, D.; YOUNGS, N.; BRITTO, A. et al. The ripple protocol consensus algorithm. **Ripple Labs Inc White Paper**, v. 5, n. 8, p. 8, 2014. Disponível em: <https://cryptoguide.ch/cryptocurrency/ripple/whitepaper.pdf>. Acesso em: 20 jun. 2019.

SOUSA, J.; BESSANI, A.; VUKOLIC, M. A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In: 48TH ANNUAL IEEE/IFIP INTERNATIONAL CONFERENCE ON DEPENDABLE SYSTEMS AND NETWORKS (DSN) ,48, 2018, Luxembourg City, Luxembourg. **Anais...** New York, NY, USA: IEEE, 2018. p. 51–58.

SVINIVAS, N. Blockchains consensus protocols in the wild. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 1, p. 1–16, 2017. Disponível em: <https://drops.dagstuhl.de/opus/volltexte/2017/8016/pdf/LIPIcs-DISC-2017-1.pdf>. Acesso em: 17 jul. 2019.

TAPSCOTT, D.; TAPSCOTT, A. **Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world**. 2. ed. Londres: Penguin, 2016. 368 p.

TURNBULL, J. **The Docker Book: Containerization is the new virtualization**. 1. ed. New York: James Turnbull, 2014.

UNDERWOOD, S. Blockchain beyond bitcoin. **Communications of the ACM**, v. 59, n. 11, p. 15–17, Out 2016.

VICTORIA, K. **Impact from the blockchain technology on the Nordic capital market**. 1. ed. Uppsala: Uppsala Universitet, 2016. 59 p.

WAZLAWICK, R. **Metodologia de pesquisa para ciência da computação**. 2. ed. Rio de Janeiro: Elsevier Brasil, 2017. 150 p.

WOOD, G. Ethereum: A secure decentralised generalised transaction ledger. **Ethereum project yellow paper**, v. 151, n. 17, p. 1–34, Set 2017.

WUST, K.; GERVAIS, A. Do you need a blockchain? In: CRYPTO VALLEY CONFERENCE ON BLOCKCHAIN TECHNOLOGY (CVCBT),18, 2018, Zug, Switzerland. **Anais...** New York, NY, USA: IEEE, 2018. p. 45–54.

ZANINOTTO, F. The blockchain explained to web developers, part 1: The theory. **marmelab**, v. 2, n. 1, p. 1–10, 2016.

APÊNDICES

APÊNDICE A – Roteiro da entrevista

UMA SOLUÇÃO BASEADA EM BLOCKCHAIN PARA CONTROLE DE REGISTROS MÉDICOS PROFISSIONAIS

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

Universidade Estadual do Ceará

SUMÁRIO

APÊNDICE B – Termo de consentimento da entrevista

SEÇÃO 1 - ASSINATURA TERMOS

UMA SOLUÇÃO BASEADA EM BLOCKCHAIN PARA GERENCIAR REGISTROS DE PROFISSIONAIS MÉDICOS

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

Universidade Estadual do Ceará

TERMO DE CONSENTIMENTO

Eu,_____,
sendo conhecedor(a) do tema e metodologia utilizados pela aluno(a) do Mestrado em Ciência da Computação da Universidade Estadual do Ceará (UECE), consinto em participar da pesquisa conduzida pela mesma.

Entendo que toda e qualquer informação prestada por mim no decorrer da(s) entrevista(s) pode ser utilizada na escritura de relatórios referentes à pesquisa. Entendo também que as entrevistas podem ser gravadas. É acertado entre mim, signatário(a) deste termo, e a aluno Raphael Lima Saraiva, que todas as possibilidades de **identificação enquanto entrevistado estão autorizadas**.

Fortaleza (CE), _____de _____de 2019.

_____

Assinatura do Voluntário(a)

APÊNDICE C – Caracterização do entrevistado

SEÇÃO 2 - CARACTERIZAÇÃO DO ENTREVISTADO

UMA SOLUÇÃO BASEADA EM BLOCKCHAIN PARA GERENCIAR REGISTROS DE PROFISSIONAIS MÉDICOS

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

Universidade Estadual do Ceará

QUESTIONÁRIO

Q1. Qual seu nome?

_____

Q2. Qual sua idade? _____

Q3. Qual seu estado civil? _____

Q4. Qual sua formação acadêmica? Em qual área? Há quanto tempo?

_____

Q5. Quanto tempo de experiência você possui na área? _____

Q6. Qual cargo você exerce atualmente na empresa em que trabalha? Há quanto tempo? _____

Q7. Há quanto tempo está vinculado à sua atual empresa? _____

Q8. Quanto tempo de funcionamento a empresa possui? _____

Q9. Principais atividades desenvolvidos?

_____

_____

Q10. Há algum nicho de cliente que sua empresa atende? Se sim, qual e por quê?

_____

_____

Q11. Em relação a quantidade de funcionários, em qual opção abaixo sua empresa se enquadra?

( ) até 19 empregados

( ) de 20 a 99 empregados

( ) 100 a 499 empregados

( ) mais de 500 empregados

APÊNDICE D – Questionário realizado na Entrevista

SEÇÃO 4 - ENTREVISTA

UMA SOLUÇÃO BASEADA EM BLOCKCHAIN PARA GERENCIAR REGISTROS DE PROFISSIONAIS MÉDICOS

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

Universidade Estadual do Ceará

SERVIÇOS PRESTADOS PELO CONSELHO REGIONAL DE MEDICINA (CRM)

1. Quais os serviços do CREMEC são oferecidos aos médicos?
2. Como se dá o processo de inscrição do médico no CRM dentre as modalidades existentes?
3. Quais as documentações são comuns para a maioria dos serviços?
4. Existe alguma diferença no processo dos serviços oferecidos entre os CRM's do país?
5. Existe algum protocolo que deva ser realizado pelas entidades emissoras de certificados (sociedades filiadas à AMB e entidades de saúde credenciadas à CNRM) para que os certificados gerados pelas mesmas sejam válidos para o CRM?

ARMAZENAMENTO E COMPARTILHAMENTO DOS DADOS

6. Após a realização de algum serviço realizado ao médico, como se dá o processo de armazenamento dos dados?
7. As informações médicas de cada estado são compartilhadas entre os CRM's?
8. Existe algum tipo de comunicação ou compartilhamento de dados entre o CRM e as entidades emissoras de certificados de especialidades?

OPINIÃO DOS ENTREVISTADOS EM RELAÇÃO À PROPOSTA

9. Quais seriam os possíveis pontos negativos que você poderia identificar nesta proposta?
10. Quais seriam os possíveis pontos positivos que você poderia identificar nesta proposta?

APÊNDICE E – Fechamento da Entrevista

SEÇÃO 5 - FECHAMENTO

UMA SOLUÇÃO BASEADA EM BLOCKCHAIN PARA GERENCIAR REGISTROS DE PROFISSIONAIS MÉDICOS

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

Universidade Estadual do Ceará

FEEDBACKS

Q1. O que você achou das explicações antes da avaliação?

( ) Suficiente

( ) Indiferente

( ) Insuficiente

Q2. Feedback aberto:

_____

_____

_____

_____

_____

_____

_____

APÊNDICE F – Documentação dos casos de uso [UC-01] do sistema

Caso de uso [UC-01]	
Nome:	Cadastrar Funcionário
Atores:	Administrador
Prioridade:	Importante
Entradas:	login e senha pré-definidas
Pré-condições:	
Pós-condições:	Um funcionário recebe um e-mail com um login e senha para acessar o sistema
Fluxos de eventos	
Fluxo normal de eventos [FN]	1. O administrador realiza o login e visualiza o tela inicial do sistema. 2. O administrador clica no link cadastrar funcionário e é direcionado para a página de cadastro de funcionário. 3. O administrador inseri as informações nos respectivos campos e clica em salva.
Fluxo alternativo de eventos [FA]	
Fluxo excepcional de eventos [FE]	1. O administrador não está conectado à internet e não consegue fazer login.

APÊNDICE G – Documentação dos caso de uso [UC-02] do sistema

Caso de uso [UC-02]	
Nome:	Autenticação
Atores:	Funcionário
Prioridade:	Importante
Entradas:	Login e senha.
Pré-condições:	O funcionário solicitou cadastro, e recebeu email com dados.
Pós-condições:	O funcionário pode acessar a tela principal do sistema.
Fluxos de eventos	
Fluxo normal de eventos [FN]	1. O funcionario acessa a tela de login do sistema. 2. Cliente inseri os dados de autenticação e faz login.
Fluxo alternativo de eventos [FA]	
Fluxo excepcional de eventos [FE]	

APÊNDICE H – Documentação dos caso de uso [UC-03] do sistema

Caso de uso [UC-03]	
Nome:	Cadastrar Registro
Atores:	Funcionário
Prioridade:	Essencial
Entradas:	informações do registro
Pré-condições:	O Funcionário deve estar logado.
Pós-condições:	As informações no registro serão adicionados a blockchain.
Fluxos de eventos	
Fluxo normal de eventos [FN]	1. O funcionário realiza o login e visualiza o tela inicial do sistema. 2. O funcionário clica no link cadastrar e é direcionado para a página de cadastro. 3. O funcionário inseri as novas informações nos respectivos campos e clica em salva. 4. O cliente é direcionado para a tela inicial.
Fluxo alternativo de eventos [FA]	O funcionário preenche os campos com dados inválidos, visualiza mensagem de erro e não acessa a tela principal do sistema.
Fluxo excepcional de eventos [FE]	1. O administrador não está conectado à internet e não consegue fazer login.

APÊNDICE I – Documentação dos caso de uso [UC-04] do sistema

Caso de uso [UC-04]	
Nome:	Buscar Registro
Atores:	Funcionário
Prioridade:	Importante
Entradas:	Algum dado do registro para busca
Pré-condições:	O Funcionário deve estar logado.
Pós-condições:	informações no registro serão visíveis na tela.
Fluxos de eventos	
Fluxo normal de eventos [FN]	1. O funcionário realiza o login e visualiza o tela inicial do sistema. 2. O funcionário clica no link buscar e é direcionado para a página de buscar. 3. O funcionário clica no link buscar e é direcionado para a página de buscar. 4. O funcionário um dado (CPF) do registro que deseja buscar.
Fluxo alternativo de eventos [FA]	
Fluxo excepcional de eventos [FE]	1. Cliente não está conectado à internet e não consegue fazer login.

APÊNDICE J – Documentação dos caso de uso [UC-05] do sistema

Caso de uso [UC-05]	
Nome:	Editar Registro
Atores:	Funcionário
Prioridade:	Importante
Entradas:	informações no registro que serão alteradas.
Pré-condições:	O Funcionário deve estar logado.
Pós-condições:	As informações alteradas estarão visíveis na blockchain
Fluxos de eventos	
Fluxo normal de eventos [FN]	1. O funcionário realiza o login e visualiza o tela inicial do sistema. 2. O funcionário clica no link buscar e é direcionado para a página de buscar. 3. O funcionário clica no link buscar e é direcionado para a página de buscar. 4. O funcionário um dado (CPF) do registro que deseja buscar. 5. O funcionário clica em editar e inseri as novas informações nos respectivos campos e salva. 6. O cliente é direcionado para a tela inicial.
Fluxo alternativo de eventos [FA]	
Fluxo excepcional de eventos [FE]	1. Cliente não está conectado à internet e não consegue fazer login.

APÊNDICE K – Documentação dos caso de uso [UC-06] do sistema

Caso de uso [UC-06]	
Nome:	Recuperar
Atores:	Funcionário
Prioridade:	Desejável
Entradas:	
Pré-condições:	O cliente precisa ter acesso ao email que foi cadastro no sistema.
Pós-condições:	Cliente receberá novamente o email com login e senha.
Fluxos de eventos	
Fluxo normal de eventos [FN]	1. O funcionário não lembra seu email ou senha e apagou o e-mail enviado pelo sistema com suas informações. 2. O funcionário clica no botão com o título 'Recuperação de senha' na página login.html. 3. O sistema reenvia o email de autenticação enviado no cadastro.
Fluxo alternativo de eventos [FA]	1. O funcionário envia email com o título 'Recuperação de email e senha' para o administrador, através do seu email cadastrado no sistema.
Fluxo excepcional de eventos [FE]	

APÊNDICE L – Documentação dos caso de uso [UC-07] do sistema

Caso de uso [UC-07]	
Nome:	feedback
Atores:	Funcionário
Prioridade:	Desejável
Entradas:	Mensagem a ser enviada.
Pré-condições:	
Pós-condições:	O administrador receberá um e-mail com a mensagem enviada.
Fluxos de eventos	
Fluxo normal de eventos [FN]	1. Clica no link ‘feedback’ no menu superior. 2. Funcionário é direcionado para a página feedback.html. 3. Usuário/Cliente digita a mensagem e envia.
Fluxo alternativo de eventos [FA]	
Fluxo excepcional de eventos [FE]	