INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FLUMINENSE

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À ENGENHARIA E GESTÃO

LUIS GUILHERME CARVALHO DE OLIVEIRA

UMA ARQUITETURA PARA AUTENTICAÇÃO E DISPONIBILIZAÇÃO DE DOCUMENTOS VIA BLOCKCHAIN E ARMAZENAMENTO ASSOCIATIVO PEER-TO-PEER ENDEREÇÁVEL AO CONTEÚDO

Campos dos Goytacazes/RJ
2020

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FLUMINENSE

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À ENGENHARIA E GESTÃO

LUIS GUILHERME CARVALHO DE OLIVEIRA

UMA ARQUITETURA PARA AUTENTICAÇÃO E DISPONIBILIZAÇÃO DE DOCUMENTOS VIA BLOCKCHAIN E ARMAZENAMENTO ASSOCIATIVO PEER-TO-PEER ENDEREÇÁVEL AO CONTEÚDO

Mark Douglas de Azevedo Jacyntho
(Orientador)

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão (MPSAEG), como parte dos requisitos necessários à obtenção do título de Mestre em Sistemas Aplicados à Engenharia e Gestão.

Campos dos Goytacazes/RJ
2020

Biblioteca Anton Dakitsch
CIP - Catalogação na Publicação

O48a

Oliveira, Luis Guilherme Carvalho de
Uma arquitetura para autenticação e disponibilização de documentos via blockchain e armazenamento associativo peer-to-peer endereçável ao conteúdo / Luis Guilherme Carvalho de Oliveira - 2020.
72 f.: il. color.

Orientador: Mark Douglas de Azevedo Jacyntho

Dissertação (mestrado) -- Instituto Federal de Educação, Ciência e Tecnologia Fluminense, Campus Campos Centro, Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão, Campos dos Goytacazes, RJ, 2020.
Referências: f. 69 a 72.

1. Blockchain. 2. Armazenamento associativo P2P. 3. Ethereum. 4. IPFS. 5. Aplicação descentralizada. I. Jacyntho, Mark Douglas de Azevedo, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da Biblioteca Anton Dakitsch do IFF com os dados fornecidos pelo(a) autor(a).

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL FLUMINENSE
CAMPUS CAMPOS CENTRO
RUA DOUTOR SIQUEIRA, 273, PARQUE DOM BOSCO, CAMPOS DOS GOYTACAZES / RJ, CEP 28030130
Fone: (22) 2726-2903, (22) 2726-2906

ATA N° 12/2020 - CMSAEGCC/DEPPGCC/DGCCENTRO/REIT/IFFLU

Mestrado em Sistemas Aplicados à Engenharia e Gestão - SAEG

ATA DE DEFESA DA DISSERTAÇÃO DE MESTRADO

Aos 17 dias do mês de junho de 2020, no horário de 15 horas às 17 horas e 25 minutos, foi realizada, via webconferência do IFFluminense, a defesa pública da dissertação do(a) mestrando(a) **Luis Guilherme Carvalho de Oliveira**, em Sistemas Aplicados à Engenharia e Gestão, intitulada "**Uma Arquitetura para Autenticação e Disponibilização de Documentos via Blockchain e Armazenamento Associativo Peer-to-peer Endereçável ao Conteúdo**".

A Banca Examinadora, presidida pelo(a) professor(a) orientador(a) Mark Douglas de Azevedo Jacyntho e constituída pelos professores Aline Pires Vieira de Vasconcelos (IFFluminense), Philippe Leal Freire dos Santos (IFFluminense) e Thiago Ribeiro Nunes (IFFluminense), após a exposição da pesquisa pelo(a) mestrando(a) e a sua arguição oral, emitiu o seguinte resultado final:

(X) APROVADO(A) ( ) APROVADO(A) COM RESTRIÇÕES ( ) REPROVADO(A)

Parecer: **O mestrando deverá acatar as sugestões propostas pela banca.**

Nota: (9) nove.

Eu, PRESIDENTE DA BANCA, orientador(a) da pesquisa, lavrei a presente Ata que segue por mim assinada e pelos demais membros da Banca Examinadora.

Dr. Mark Douglas de Azevedo Jacyntho (IFFluminense) - Presidente da Banca;

Drª. Aline Pires Vieira de Vasconcelos (IFFluminense) - Membro Interno;

Dr. Philippe Leal Freire dos Santos (IFFluminense) - Membro Externo;

Dr. Thiago Ribeiro Nunes (IFFluminense) - Membro Externo.

*No caso de **aprovação com restrições**, o Presidente da Banca Examinadora deverá preencher, em anuência com os membros da banca, a Folha de Exigências com as considerações e alterações que deverão ser cumpridas pelo mestrando no prazo não superior a 30 dias.*

FOLHA DE EXIGÊNCIAS DA DISSERTAÇÃO DE MESTRADO

O(A) autor(a), **Luis Guilherme Carvalho de Oliveira** da dissertação de mestrado intitulada "**Uma Arquitetura para Autenticação e Disponibilização de Documentos via Blockchain e Armazenamento Associativo Peer-to-peer Endereçável ao Conteúdo**" deverá cumprir as seguintes exigências:

O mestrando deverá acatar as sugestões propostas pela banca.

O prazo para cumprimento das exigências é de **30** dias, sendo responsável(eis) o(s) professore(s): **Mark Douglas de Azevedo Jacyntho.**

O(s) professore(s) responsável(eis) atesta(m) que foram cumpridas as exigências realizadas pela Banca Examinadora.

Documento assinado eletronicamente por:

- Thiago Ribeiro Nunes, PROFESSOR ENS BASICO TECN TECNOLOGICO, COORDENAÇÃO DO CURSO DE ELETRÔNICA PROEJA, em 24/06/2020 08:43:35.
- Aline Pires Vieira de Vasconcelos, PROFESSOR ENS BASICO TECN TECNOLOGICO, COORDENAÇÃO DO CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO, em 24/06/2020 02:57:47.
- Philippe Leal Freire dos Santos, PROFESSOR ENS BASICO TECN TECNOLOGICO, COORDENAÇÃO DO CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO, em 23/06/2020 21:05:26.
- Mark Douglas de Azevedo Jacyntho, PROFESSOR ENS BASICO TECN TECNOLOGICO, COORDENAÇÃO DO CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO, em 23/06/2020 20:34:27.

Este documento foi emitido pelo SUAP em 23/06/2020. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.iff.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 154967
Código de Autenticação: f45656dce6

RESUMO

Este projeto propõe uma arquitetura para autenticação e disponibilização de documentos via *blockchain* e armazenamento associativo *peer-to-peer* (P2P) endereçável ao conteúdo, respectivamente. Como prova de conceito da arquitetura proposta, uma aplicação descentralizada (*DApp*) foi desenvolvida, utilizando a *blockchain Ethereum* e o sistema de arquivos associativo P2P *InterPlanetary File System* (IPFS). Esta iniciativa visa minimizar a necessidade de terceiros confiáveis (p.ex. cartórios) para a autenticação de documentos, e assim, reduzir burocracia, fraudes e custos inerentes a este processo. Além dos documentos propriamente ditos, opcionalmente, arquivos com metadados semânticos podem ser associados a estes, permitindo que máquinas possam compreender seu conteúdo, aumentando, pois, a interoperabilidade. A eficácia da arquitetura e da aplicação desenvolvida, abrangendo desde o *backend* (contrato inteligente *Ethereum* e armazenamento distribuído IPFS) até o *frontend* Web, foi corroborada por meio de exemplos exploratórios usando tanto uma *blockchain* local quanto *blockchains* de teste públicas (*testnets*), amplamente reconhecidas pela comunidade de desenvolvedores *Ethereum*. Por fim, um estudo de caso realista é descrito, com o propósito de ilustrar o uso da arquitetura e da aplicação.

Palavras-chave: *Blockchain*, Armazenamento associativo P2P, *Ethereum*, IPFS, Aplicação descentralizada, Contratos inteligentes.

ABSTRACT

This project proposes an architecture for authenticating and providing documents via blockchain and content-addressing peer-to-peer (P2P) associative storage, respectively. As a proof of concept of the proposed architecture, a decentralized application (DApp) was developed, using the Ethereum blockchain and the P2P associative file system InterPlanetary File System (IPFS). This initiative aims to minimize the need for trusted third parties (e.g. notaries) to authenticate documents, and thus reduce bureaucracy, fraud, and costs inherent in this process. In addition to the documents themselves, files with semantic metadata can be optionally associated with them, allowing machines to understand their content, thus increasing interoperability. The effectiveness of the developed architecture and application, ranging from the backend (Ethereum smart contract and IPFS distributed storage) to the Web frontend, has been corroborated through exploratory examples using both a local blockchain and public test blockchains (testnets) widely recognized by the Ethereum developer community. Finally, a realistic case study is described, in order to illustrate both the use of architecture and the application.

Keywords: Blockchain, P2P associative storage, Ethereum, IPFS, Decentralized application, Smart contracts.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

Atualmente, contratos e transações (e seus respectivos registros) fazem parte da estruturação de sistemas econômicos, legais e políticos na sociedade moderna, uma vez que resguardam ativos e determinam limites organizacionais, além de estabelecer e verificar identidades e eventos cronológicos. Todavia, os mecanismos de gerenciamento supracitados e as burocracias arquitetadas para administrá-los não acompanharam a transformação digital da economia. A tecnologia *blockchain* surge como alternativa promissora para solução desse problema (IANSITI; LAKHANI, 2017).

Uma transação financeira autêntica representa uma operação entre duas partes (pessoas físicas ou jurídicas), realizada de forma centralizada por um terceiro confiável (intermediário). Exemplos desta forma de operação são as transferências bancárias e aquisições via cartão de crédito, uma vez que são autenticadas, respectivamente, pelo banco e pela operadora de cartão de crédito, ao custo de taxas adicionais por intermédio (YLI-HUUMO *et al.*, 2016).

Com o emprego da tecnologia *blockchain,* não se faz mais necessária a aprovação por terceiros, uma vez que *blockchain* constitui um banco de dados ou livro razão distribuído em que os registros dos dados são imutáveis e confirmados pelos nós constituintes da rede. Os dados de todas as operações são registrados publicamente e disponibilizados para todos os nós, permitindo que estes certifiquem futuras operações. *Bitcoin* foi a primeira aplicação a utilizar a tecnologia *blockchain*. A criptomoeda possui um sistema descentralizado, no qual os participantes podem realizar transações com a moeda digital (YLI-HUUMO *et al.*, 2016).

A mesma lógica pode ser aplicada a contratos. Em 1994, Nick Szabo introduziu o conceito de contratos inteligentes (*smart contracts*), sendo um protocolo de transação computadorizado que executa os termos de um contrato. Os principais objetivos de um contrato inteligente são: cumprir condições contratuais comuns (como condições de pagamento, ônus e confidencialidade), mitigar exceções maliciosas e acidentais, e minimizar a necessidade de intermediários confiáveis. Em relação a objetivos econômicos, incluem-se redução de perda por fraude, custos de arbitragem e execução, e outros custos de transação (SZABO, 1994).

No contexto de *blockchain*, contratos inteligentes são programas de computador armazenados nesta, e por isso, possuem um endereço próprio. Para interagir com um contrato, é preciso realizar uma transação com destino a este endereço. Em seguida, ele é executado de maneira independente e automática, na forma prescrita, em todos os nós da rede, de acordo com os dados incluídos na transação de acionamento (CHRISTIDIS; DEVETSIKIOTIS, 2016). Sistemas mais recentes baseados em *blockchain* permitem que os usuários especifiquem códigos em transações e contratos para auxiliar aplicações que ultrapassam os limites de simples transações financeiras, como, por exemplo, a *blockchain* da rede *Ethereum*[1] (YLI-HUUMO *et al.*, 2016).

Apesar das diversas potencialidades inerentes a uma rede *blockchain* pública, como o sistema de consenso descentralizado, níveis extremos de tolerância a falhas e imutabilidade, há taxas a serem pagas à rede. O usuário que realiza a operação deve pagar por cada etapa da transação, incluindo as computacionais e de armazenamento de memória. Quando um contrato, por exemplo, é executado através de uma mensagem ou transação, cada instrução é executada em todos os nós da rede, e para cada operação realizada há um custo especificado (ETHEREUM COMMUNITY, 2016).

Devido ao custo operacional de uma *blockchain* pública, como a rede *Ethereum*, é importante que seja registrado apenas o que é estritamente essencial na *blockchain*, a fim de evitar gastos desnecessários. Assim, demanda-se um sistema de armazenamento de dados subjacente que tenha sinergia com a *tecnologia blockchain*. Alinhados ao caráter distribuído, imutável e redundante de *blockchains*, vem ao encontro deste propósito, o surgimento dos sistemas de armazenamento associativo *peer-to-peer* (P2P) endereçável ao conteúdo, que são sistemas de arquivos distribuídos P2P, onde cada arquivo tem um endereço global único gerado a partir de seu conteúdo e partes deste arquivo são distribuídas pelos nós da rede, fornecendo resiliência e maior velocidade de acesso. Estes novos sistemas de arquivos impulsionam um movimento que vem sendo chamado de *Web de conteúdo distribuído* e, dentre eles, destaca-se o *InterPlanetary File System* (IPFS)[2], no qual é possível publicar grandes quantidades de dados e registrar somente seus respectivos endereços em uma

[1] https://www.ethereum.org/
[2] https://ipfs.io/

transação na *blockchain*, sem que seja necessário inserir na *blockchain* os dados do arquivo em sua plenitude (PROTOCOL LABS, 2019a).

Desta forma, é possível disponibilizar documentos por meio de sistemas de armazenamento associativo (P2P) endereçável ao conteúdo, e autenticá-los, ao publicar seus respectivos endereços na *blockchain*. Sob esta luz, este trabalho propõe uma arquitetura para autenticação de documentos quaisquer via *blockchain*, bem como sua disponibilização usando este novo protocolo P2P de armazenamento associativo de arquivos, endereçáveis com base nos respectivos conteúdos.

1.1 Justificativa

A motivação central deste trabalho é eliminar a necessidade de uma terceira parte confiável (p.ex. cartório) para autenticar documentos, reduzindo, pois, fraudes, tempo e custo. Para este propósito, a tecnologia *blockchain*, com suas provas invioláveis de autenticação, é muito bem-vinda.

Outra motivação que surge naturalmente é o compartilhamento eficiente e resiliente destes documentos, de forma distribuída, diminuindo o tempo de acesso e, sobretudo, eliminando o ponto central de falhas inerente às arquiteturas cliente-servidor convencionais. Neste sentido, os sistemas de arquivos P2P da chamada *Web de conteúdo distribuído* são uma solução bastante interessante.

A tecnologia *blockchain* vem apresentando crescimento exponencial ao longo dos anos, uma vez identificado o seu potencial. Em um levantamento realizado na maior base de publicações científicas, Scopus[3], foram identificadas 12.177 publicações relacionadas a esta tecnologia na data de 20/06/2020 – TITLE-ABS-KEY (blockchain) –. Além disso, apenas 150 (≅1,2%) possuem participações de universidades brasileiras, o que instiga a realização de novos projetos nesta temática, a fim de fomentar seu desenvolvimento em âmbito nacional.

[3] https://scopus.com/

Como já supramencionado, contratos inteligentes, por meio da *blockchain*, são capazes de operar contratos automaticamente, e reduzir de forma substancial a necessidade de terceiros confiáveis, o que implica a diminuição de fraudes e custos operacionais. No Brasil, estas disposições podem ser aplicadas às atividades notariais e registrais, como serviços de reconhecimento de firma e cópia autenticada.

Em 2017, os cartórios arrecadaram R$ 15,74 bilhões (CNJ, 2018) através da prestação de serviços notariais e registrais. Ainda, são diversos os casos envolvendo fraudes por parte destes profissionais, dotados de fé pública.

Através de contratos inteligentes bem desenvolvidos, é possível reduzir drasticamente, não só os custos inerentes aos serviços notariais e registrais prestados e as falhas humanas, intencionais ou não, como também a burocracia, pois, uma vez que é uma solução digital e automática, possui menos intermediários e encurta a distância entre a parte requerente e o produto final.

Ademais, faz-se necessário discutir sobre a validade jurídica da tecnologia *blockchain* no Brasil. O juiz de direito e acadêmico Alexandre Morais da Rosa[4] discorre acerca da temática, à luz da Medida Provisória 2.200-2/2001[5], que institui a Infraestrutura de Chaves Públicas Brasileira, ICP-Brasil, a fim de "garantir a autenticidade, a integridade e a validade jurídica de documentos em forma eletrônica, das aplicações de suporte e das aplicações habilitadas que utilizem certificados digitais, bem como a realização de transações eletrônicas seguras."

> No caso das ferramentas que promovem a utilização da *blockchain* como base de dados para autenticação de documentos, como dito, um "livro razão" descentralizado, transparente, público e totalmente auditável, que, após o registro das informações em sua rede, torna-se imutável o documento ali escrito, entendemos pela plena viabilidade jurídica e validade das provas ali produzidas. Isso porque o artigo 10 da MP 2.200-2/2001 prevê que outras formas de assinaturas ou provas de autenticidade podem se reputar válidas, ainda que não prescritas na referida MP, o que dá pleno respaldo à utilização da rede para os fins aqui discutidos. (ROSA, 2019)

[4] http://lattes.cnpq.br/4049394828751754
[5] http://www.planalto.gov.br/ccivil_03/mpv/antigas_2001/2200-2.htm

Assim, o juiz afirma ainda que, devido às características pertinentes à tecnologia *blockchain*, como a alta segurança, a integridade, a auditabilidade, a imutabilidade e a transparência dos dados, as provas documentais geradas pela *blockchain* possuem, indubitavelmente, validade jurídica (ROSA, 2019).

1.2 Objetivo geral

O presente projeto propõe a elaboração de uma arquitetura racional capaz de: (a) autenticar/validar documentos via *blockchain,* a fim de eliminar a necessidade de terceiros confiáveis, e assim, reduzir burocracia, custos e fraudes relativos ao processo de autenticação; (b) disponibilizar estes documentos por meio de um sistema de arquivos distribuído, com o propósito de obter um compartilhamento eficiente e resiliente de documentos via armazenamento associativo peer-to-peer endereçável ao conteúdo.

1.3 Objetivos específicos

Para atingir o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Empregar um algoritmo criptográfico (*hashe*) conhecido a fim de reduzir a quantidade de informação armazenada na *blockchain*, diminuindo, pois, os custos inerentes à transação executada na rede;
- Utilizar uma *blockchain* consagrada programável de propósito geral, que ofereça uma máquina virtual de Turing completa, capaz de executar contratos inteligentes (programas de computador);
- Para armazenar e disponibilizar os documentos, utilizar um sistema de arquivos distribuído P2P endereçável ao conteúdo, comprovadamente testado e largamente utilizado, que, ao invés de armazenar os arquivos em um servidor central, permita que os arquivos sejam obtidos a partir de qualquer nó da rede P2P, reduzindo tempo de resposta e aumentando resiliência;
- Além do documento original propriamente dito, registrar também, opcionalmente, um documento RDF[6] com metadados semânticos,

[6] https://www.w3.org/TR/rdf11-primer/

viabilizando que máquinas consigam compreender o conteúdo do documento registrado. RDF é modelo de dados padrão da Web Semântica definida pelo consórcio W3C[7];

- Desenvolver uma aplicação descentralizada (*DApp*) para disponibilização e autenticação/validação de documentos, contendo: contrato inteligente como *backend*, interface Web como *frontend* e sistema de arquivos P2P endereçável ao conteúdo como armazenamento;
- Contribuir para a expansão da Web Semântica, da Web descentralizada, bem como da tecnologia disruptiva *blockchain*, por meio do legado deixado por este trabalho.

1.4 Estrutura do documento

Além desta introdução, este documento encontra-se estruturado da seguinte forma:

- Capítulo 2: fundamentos necessários para a compreensão dos conceitos abordados, como *blockchain* e armazenamento associativo P2P endereçável ao conteúdo;
- Capítulo 3: trajetória metodológica e a proposição da arquitetura para a autenticação e disponibilização de documentos.
- Capítulo 4: implementação da arquitetura proposta através de uma aplicação descentralizada (*DApp*), que inclui a descrição das ferramentas utilizadas, o desenvolvimento da aplicação propriamente dita, além de testes exploratórios e um estudo de caso;
- Capítulo 5: resultados e discussões acerca da arquitetura, da aplicação e dos testes realizados;
- Capítulo 6: alguns trabalhos relacionados que, ora corroboram a originalidade do projeto, ora reiteram sua importância;

[7] https://www.w3.org/

- Capítulo 7: considerações finais sobre o trabalho desenvolvido, bem como alguns trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são abordados, tanto conceitos essenciais ao embasamento da arquitetura proposta, como *blockchain*, armazenamento associativo P2P endereçável ao conteúdo e aplicações descentralizadas, quanto conceitos complementares, como *hashes* e modelo de dados RDF, a fim de reduzir possíveis lacunas de conhecimento inerentes a este projeto.

2.1 Hashes

Antes de discorrer sobre as tecnologias disruptivas utilizadas neste projeto, é importante conceituar o algoritmo criptográfico de *hashes.*

Hashes são funções que, a partir de um valor de entrada, retornam como saída um valor de comprimento fixo. Este valor depende do algoritmo utilizado (SHA-1, SHA-256, BLAKE1, por exemplo), porém, um algoritmo *hash* sempre retornará o mesmo valor de saída para um determinado valor de entrada. Assim, a criptografia por *hashes* apresenta quatro características significativas (PROTOCOL LABS, 2019b):

- Determinística: como dito anteriormente, o mesmo valor de entrada sempre retornará o mesmo valor de saída (*hash*).
- Não há correlação: qualquer mínima alteração no valor de entrada resultará em uma *hash* de saída completamente diferente.
- Singularidade: é inviável obter a mesma *hash* através de duas entradas diferentes.
- Sentido único: é inviável obter o valor de entrada através da *hash* de saída.

Tabela 1 – Exemplos de aplicação da função de hash pelo algoritmo SHA-256.

SHA-256	
Entrada	Saída
é um teste	5633f7bfb51fc57f4dfc3978a65c8f7288db2979b98cd5d209c289e97916f41c
É um teste	28ddb99a3c1e1ea26a8ad97db979773367a735fd482d7e08b84867222b1e1dde
É um teste.	a1fadce73fa05118573c78d5fd504f186ccae983bf7fe0f49bbde9e774acc1ea

Fonte: Elaboração própria (2019).

Por meio da Tabela 1, é possível observar o princípio de não correlação da função de *hash*, uma vez que mínimas alterações no texto a ser codificado originam uma nova *hash* totalmente distinta. Nota-se também que a *hash* gerada pela função SHA-256 possui sempre 64 caracteres, independente dos dados de entrada. Ressalta-se ainda que qualquer um é capaz de comprovar a validade das *hashes* geradas nesta tabela; ao utilizar da função de *hash* pelo algoritmo SHA-256 e atribuir os mesmos valores de entrada, serão obtidos os mesmos valores (*hashes*) de saída, corroborando seu princípio determinístico.

2.2 Tecnologia Blockchain

Blockchain é um livro razão compartilhado e imutável que facilita o processo de registro de transações e rastreamento de recursos em uma rede de negócios, sejam tangíveis, como casa, carro ou dinheiro, ou intangíveis, como propriedade intelectual e patente. Praticamente qualquer coisa de valor pode ser rastreada e negociada em uma rede *blockchain*, reduzindo riscos e custos para as partes envolvidas (GUPTA, 2018).

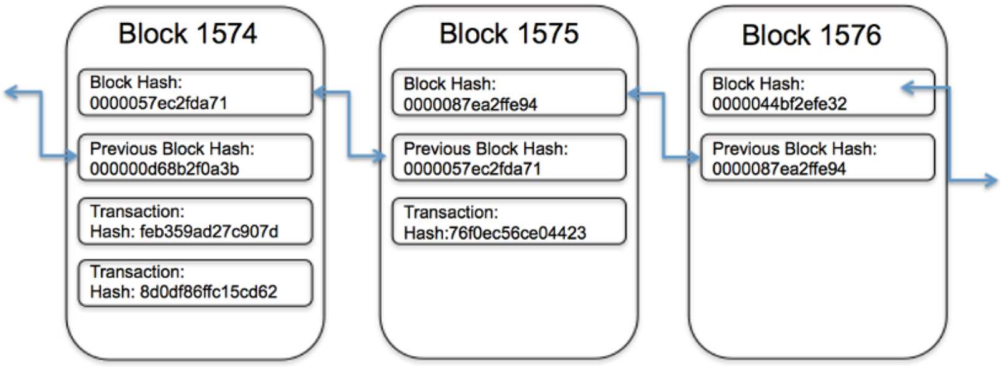

Figura 1 - Registros de transações armazenados em uma cadeia de blocos (blockchain).
Fonte: Blockchain for dummies (GUPTA, 2018).

Nota-se, na Figura 1, que cada bloco possui sua própria *hash* (identificador), registros temporais de transações válidas e *hash* do bloco anterior. A *hash* do bloco anterior permite a vinculação entre os blocos, de modo a impedir que sejam alterados, ou que um bloco seja inserido entre os blocos vinculados. Desta forma, cada bloco subsequente reforça a verificação do bloco anterior, e consequentemente, toda a cadeia de blocos. Este método torna *blockchain* inviolável, caracterizando sua imutabilidade (GUPTA, 2018).

Como dito anteriormente, transações são armazenadas ao serem registradas em um bloco; apenas transações válidas são adicionadas à *blockchain*, e esta validação é, tecnicamente, o processo de identificação da *hash* do bloco. Desta forma, os blocos são inseridos na *blockchain* apenas após a validação. O processo de validação é realizado pelos nós participantes, denominados, nesta condição, de mineradores, que são recompensados financeiramente pelo esforço ou poder computacional utilizado (CYBROSYS, 2018).

Existem diferentes protocolos *blockchain*, chamados de protocolos de consenso (*consensus protocols*), que determinam qual nó minerará o bloco; o protocolo utilizado atualmente nas principais redes públicas de *blockchain,* como *Bitcoin* e *Ethereum,* é o protocolo prova de trabalho (*Proof of Work*). Neste protocolo, os nós competem, através de poder computacional, para resolver um quebra-cabeças criptográfico. O primeiro nó a resolver o quebra-cabeças tem o direito de criar um novo bloco (ZHANG; LEE, 2019).

No que tange à tecnologia *blockchain*, há cinco princípios básicos, a saber (IANSITI; LAKHANI, 2017):

- Base de dados distribuída: cada participante na *blockchain* tem acesso completo à base de dados e seu histórico. Nenhuma parte é capaz de controlar dados ou informações. Além disso, todos são capazes de apurar quaisquer transações sem intermediação;
- Sistema P2P (*Peer-to-Peer*): de maneira oposta a um sistema centralizado (servidor central), a comunicação ocorre diretamente entre os nós da rede. Cada nó armazena as informações e as transmite aos demais;
- Pseudotransparência: ainda que todas as operações e seus respectivos valores estejam disponíveis a qualquer um que tenha acesso ao sistema, cada usuário possui um endereço de identificação alfanumérico. Desta forma, o usuário é capaz de optar por permanecer anônimo ou fornecer provas de associação a seu respectivo endereço. Ressalta-se que as transações são realizadas entre esses endereços;
- Imutabilidade dos registros: uma vez que determinada transação faz parte do banco de dados, ela não pode ser alterada ou excluída, pois estão ligadas a

todas as transações predecessoras. Diversos algoritmos e abordagens computacionais são instaurados para garantir que a gravação na base de dados seja permanente, organizada cronologicamente e disponível a todos na *blockchain*;

- Lógica computacional: a natureza digital da *blockchain* permite vincular suas transações à lógica computacional, isto é, programação. Desta maneira, os usuários podem configurar algoritmos e regras que acionam automaticamente transações entre nós.

Nos últimos anos, esta tecnologia chamou a atenção para as criptomoedas, principalmente nas redes *Bitcoin* e *Ethereum*. Atualmente, a tecnologia *blockchain* vem desempenhando um papel maior no mercado financeiro no que tange à cadeia de suprimentos, validação de documentos, internet das coisas e armazenamento, por exemplo, de forma descentralizada (WANG; ZHANG; ZHANG, 2018).

2.3 Rede Ethereum

Derivado do primeiro livro razão distribuído (*Bitcoin*), *Ethereum* o reaproveita para modelar uma máquina virtual capaz de fornecer a códigos de operação a mesma segurança encontrada nas transações da rede *Bitcoin*; isto é, assim como é possível garantir a precisão e confiabilidade das operações e seus respectivos registros temporais (*timestamps*) na rede *Bitcoin*, a mesma premissa é aplicada a rede *Ethereum*, assegurando que instruções de máquina iniciadas através do protocolo sejam executadas (DAMERON, 2019). Em outras palavras, enquanto a rede *Bitcoin* é retratada como uma base de dados de contas (ou carteiras) que armazenam criptomoedas, a rede *Ethereum* possui uma estrutura mais sofisticada, capaz de armazenar códigos computacionais e executar suas respectivas aplicações através do poder de processamento da máquina virtual *Ethereum* (EVM), uma máquina de Turing completa (MARR, 2018).

A plataforma busca desenvolver a generalização da tecnologia *blockchain*, na qual conceitos de máquina baseados em transações e mudanças de estado podem ser construídos.

Por meio da rede *Ethereum*, usuários são capazes de se reunir em um sistema operacional social, seguro e descentralizado (WOOD, 2018).

Ethereum é uma plataforma *blockchain* em código aberto que permite a qualquer pessoa desenvolver e usar aplicações descentralizadas (*DApps*) que são executadas através da tecnologia *blockchain*. A rede foi projetada para ser flexível e adaptável, uma vez que se delineou como uma plataforma baseada em contratos inteligentes sem mediação (ETHEREUM COMMUNITY, 2016). Tais atributos são possíveis porque esta *blockchain* opera através de uma linguagem de programação Turing-completa integrada, na qual qualquer um pode criar suas próprias regras arbitrárias de propriedade, formatos de transação e funções de transição de estado (BUTERIN, 2013).

2.4 Contratos inteligentes

Como dito anteriormente, contratos inteligentes (*smart contracts*) são códigos de programação inseridos na *blockchain* e possuem um endereço próprio. Para interagir com um contrato, é necessário realizar uma transação com destino a este endereço. Assim, ele é executado de maneira independente, automática e estrita, na forma que foi desenvolvido.

No caso da rede *Ethereum*, isto é possível porque existem dois tipos de contas, essencialmente: contas de propriedade externa, controladas por chaves privadas (usuários), e contas de contrato, controladas pelos códigos de seus respectivos contratos. Contas de propriedade externa não possuem código, e usuários podem realizar transações através delas, enquanto as contas de contrato têm seu respectivo código executado quando transações são destinadas a elas (BUTERIN, 2013).

Na rede *Ethereum*, o desenvolvimento de contratos inteligentes é realizado por meio de linguagens de programação de alto nível, estaticamente escritas e orientadas a contratos, dentre as quais, a mais utilizada é *Solidity* (ETHEREUM FOUNDATION, 2019). O código de alto nível criado com *Solidity* é compilado em *bytecodes* da EVM e estes *bytecodes* são o código do contrato inteligente publicado na rede *Ethereum*, no respectivo endereço, a partir do qual funções do contrato inteligente podem ser invocadas.

Com a linguagem *Solidity*, usuários são capazes de criar contratos inteligentes incorporando diversas funcionalidades, como votações, financiamentos colaborativos, leilões ocultos e carteiras com várias assinaturas (“Solidity Documentation”, 2019).

```
1  pragma solidity >=0.4.0 <0.7.0;
2
3  contract SimpleStorage {
4      uint storedData;
5
6      function set(uint x) public {
7          storedData = x;
8      }
9
10     function get() public view returns (uint) {
11         return storedData;
12     }
13 }
```

Figura 2 – Exemplo de contrato inteligente de armazenamento de dados.
Fonte: Solidity Documentation (“Introduction to Smart Contracts — Solidity 0.5.15 documentation”, 2019).

A Figura 2 exemplifica um contrato inteligente simples de armazenamento de dados. É possível observar inicialmente que o contrato pode ser compilado pela versão 0.4.0 do compilador *Solidity* até a versão imediatamente anterior a versão 0.7.0 (linha 1). O contrato, intitulado *SimpleStorage* (linha 3), possui a variável de estado *storedData* (linha 4). Ao acionar a função *set* (linha 6), uma variável *x,* atribuída pelo requerente do contrato, é registrada, modificando a variável de estado do contrato *storedData* (linha 7). Enquanto isso, a função *get* (linha 10) é capaz de retornar o valor atual da variável de estado *storedData* (linha 11) (“Introduction to Smart Contracts — Solidity 0.5.15 documentation”, 2019).

Cabe reiterar que, no exemplo exposto, qualquer um pode acessar e acionar as funções do contrato inteligente, registrando um novo número inteiro (*uint*) à variável *x* e alterando a variável de estado *storedData*, uma vez que não há nenhuma restrição de acesso descrita no contrato. Ressalta-se ainda que, apesar da mutabilidade da variável de estado *storedData*, todos os seus valores já registrados podem ser acessados através do histórico da *blockchain*. Isto acontece porque toda vez que a função *set* for acionada, uma transação é registrada junto à *blockchain* com todas as informações pertinentes a operação, como o valor atribuído (*uint x*), quem a realizou (chave pública), seu registro temporal (*timestamp*), etc.

2.5 Armazenamento associativo *peer-to-peer* (P2P) endereçado ao conteúdo

Um sistema de arquivos de armazenamento associativo endereçado ao conteúdo oferece armazenamento persistente de dados que são compartilhados entre os nós da rede P2P. Os arquivos são criados diretamente e podem sobreviver ao tempo de vida de processos e nós até a exclusão explícita (indisponibilidade). Para garantir a integridade dos dados é utilizada a criptografia por *hashes*, fornecendo um identificador (endereço) único e número de caracteres constante, gerado a partir do conteúdo do arquivo. Colisões, isto é, obter a mesma *hash* para agrupamento de dados diferentes, é virtualmente impossível, tornando a criptografia por *hashes* um meio de proteção à adulteração de dados e assegurando a imutabilidade. Além disso, arquivos são deduplicados, isto é, dados duplicados são eliminados automaticamente, reduzindo a utilização real do espaço de armazenamento e otimizando a disponibilidade através dos nós (BLOMER, 2015).

Uma vez que não existe um servidor de arquivos centralizado, este tipo de sistema de armazenamento aprimora duas características importantes, a saber: resiliência e velocidade. Resiliência porque existe backup de um mesmo arquivo em vários nós da rede. Velocidade porque partes de um mesmo arquivo podem ser obtidas, paralelamente, de diversos nós da rede.

2.6 InterPlanetary File System (IPFS)

InterPlanetary File System (IPFS) é um sistema de arquivos de armazenamento associativo P2P endereçado ao conteúdo que busca conectar todos os dispositivos de computação com o mesmo sistema de arquivos. De certa forma, o IPFS é semelhante à Web, mas pode ser visto como um único conjunto de *BitTorrent*, trocando objetos em um repositório *Git*. Em outras palavras, o IPFS fornece um modelo de armazenamento em bloco endereçado a conteúdo de alto rendimento, com *hiperlinks* endereçados a conteúdo. Isto forma uma estrutura de dados sobre a qual é possível construir sistemas de arquivos com versão, *blockchains* e até uma Web permanente (BENET, 2019).

Figura 3 – Exemplo de arquivos publicados no IPFS.
Fonte: Elaboração própria (2019).

A Figura 3 apresenta quatro documentos publicados no sistema de arquivos distribuído IPFS e suas respectivas *hashes.* Nota-se a deduplicação realizada pelo sistema no arquivo *FOAF RDF* que, ao ser publicado duas vezes, ainda que com extensões de formato diferentes (.rdf e .txt), apresentam a mesma *hash* de identificação, por possuírem exatamente o mesmo conteúdo. Ressalta-se ainda que, caso qualquer usuário queira acessar o documento pelo navegador, basta utilizar o *gateway* fornecido pelo IPFS, acrescido da *hash* de identificação do documento, neste caso, <https://ipfs.io/ipfs/QmeVRmyWzTwjijmGuqC4XPw7xJpQMHoyypK5bkb5q7Rizi>.

Reforça-se ainda a importância da criptografia por função de *hash*: caso o documento em questão seja minimamente alterado e republicado, este é reconhecido como um novo documento e lhe é atribuída uma nova *hash* completamente diferente e, por conseguinte, um novo endereço, garantindo a imutabilidade do documento original; isto é, a correlação entre o documento original e sua respectiva *hash* originalmente gerada.

2.7 Aplicações descentralizadas (DApps)

Andreas Antonopoulos, um dos maiores especialistas em *blockchain* do mundo, e Gavin Wood, cofundador da rede *Ethereum*, em Antonopoulos e Wood (2018), descrevem

aplicações descentralizadas (*DApps*, sigla em inglês) como aplicações majoritária ou inteiramente descentralizadas, devendo-se considerar diversos aspectos neste quesito, como *backend, frontend* e armazenamento de dados. Cada um destes aspectos pode ser, de alguma forma, centralizado ou descentralizado.

Em uma aplicação descentralizada, contratos inteligentes são utilizados para armazenar a lógica do negócio, e, de forma simplificada, substituem os componentes *backend* presentes em uma aplicação comum. Depois de implementado, um contrato inteligente pode ser operado por diversos desenvolvedores e usuários através da rede *Ethereum*.

Diferente da lógica do negócio (contratos inteligentes), que requerem maior entendimento sobre a máquina virtual do *Ethereum* e novas linguagens de programação, como *Solidity*, a interface do usuário (*frontend*) pode ser realizada por meio de tecnologias Web tradicionais (HTML, CSS, *JavaScript*, etc.). O *frontend* geralmente é vinculado ao *Ethereum* por meio da biblioteca *JavaScript web3.js*, que é incluída aos recursos do *frontend* e é fornecida a um navegador por um servidor Web.

Devido a altos custos em *gas* (taxa necessária para realizar com êxito uma transação ou executar um contrato na plataforma *blockchain Ethereum*), contratos inteligentes não são adequados para armazenar ou processar dados em larga escala. Por esta razão, a maioria das aplicações descentralizadas utilizam plataformas de armazenamento fora da *blockchain*. Estas plataformas podem ser centralizadas, como bancos de dados na nuvem, por exemplo, ou descentralizadas em um sistema P2P, como IPFS ou *Swarm*[8].

Além disso, Antonopoulos e Wood (2018) discernem sobre as vantagens que uma aplicação descentralizada possui em relação a uma arquitetura tradicionalmente centralizada:

- Resiliência: o fato de a lógica do negócio ser controlada por um contrato inteligente torna o *backend* da aplicação descentralizada totalmente distribuído e gerenciado na *blockchain*. Assim, enquanto a *blockchain* estiver operando, a aplicação permanecerá disponível e sem tempo de inatividade;

[8] https://swarm-gateways.net/bzz:/theswarm.eth/

- Transparência: uma vez registrada na *blockchain*, qualquer um pode verificar o código da aplicação e suas funções. Além disso, todas as interações com a *DApp* são armazenadas permanentemente.
- Resistência à censura: enquanto tiver acesso a um nó da rede *Ethereum*, nenhuma espécie de controle centralizado impedirá o usuário de interagir com a aplicação. Ninguém, nem mesmo o dono do contrato, pode alterar o código, uma vez implementado à rede.

2.8 Web Semântica

Na Web original, a máquina não compreende a semântica dos documentos (páginas) publicados, limitando-se a exibi-los para que o ser humano os interprete. A Web Semântica (BERNERS-LEE *et al.*, 2001 apud AZEVEDO; JACYNTHO, 2014) vislumbra ir além, publicando dados formalmente estruturados, bem como estabelecendo relacionamentos semânticos entre dados de diferentes fontes de dados (*Linked data*). Um espaço global de dados (Web de Dados Ligados), com semântica explícita, compreensível tanto por seres humanos, mas, sobretudo, por agentes de software que então podem tomar decisões "inteligentes" para nos auxiliar (JACYNTHO, 2012).

Nesta Web de dados semânticos, cada recurso (entidade) da vida real é identificado por um *Uniform Resource Indentifier* (URI). Um URI é um endereço Web que, ao ser acessado, retorna uma descrição (propriedades e valores) do recurso em um formato estruturado, inteligível por máquina, denominado *Resource Description Framework* (RDF). Desta forma, essencialmente, os recursos são capazes de se conectar semanticamente por meio de seus endereços Web (URI), e são descritos por um modelo de dados estruturado (RDF) (AZEVEDO; JACYNTHO, 2014).

2.9 Resource description framework (RDF): modelo de dados da Web Semântica

O modelo de dados RDF, comumente utilizado em aplicações relacionadas à Web Semântica, permite a interligação de dados em grafo. Além de se enquadrar muito bem com

algoritmos e técnicas estatísticas desenvolvidas para grafos, o modelo possui forte capacidade de consulta através da linguagem de consulta SPARQL (JACYNTHO; SCHWABE, 2016).

Um grafo RDF é um conjunto de triplas (*recurso-propriedade-valor* ou *sujeito-predicado-objeto*) descrevendo um ou mais recursos (JACYNTHO, 2012). Para ilustrar, o grafo RDF da Figura 4 descreve os recursos "Bob" e a famosa pintura "The Mona Lisa" de Leonardo da Vinci.

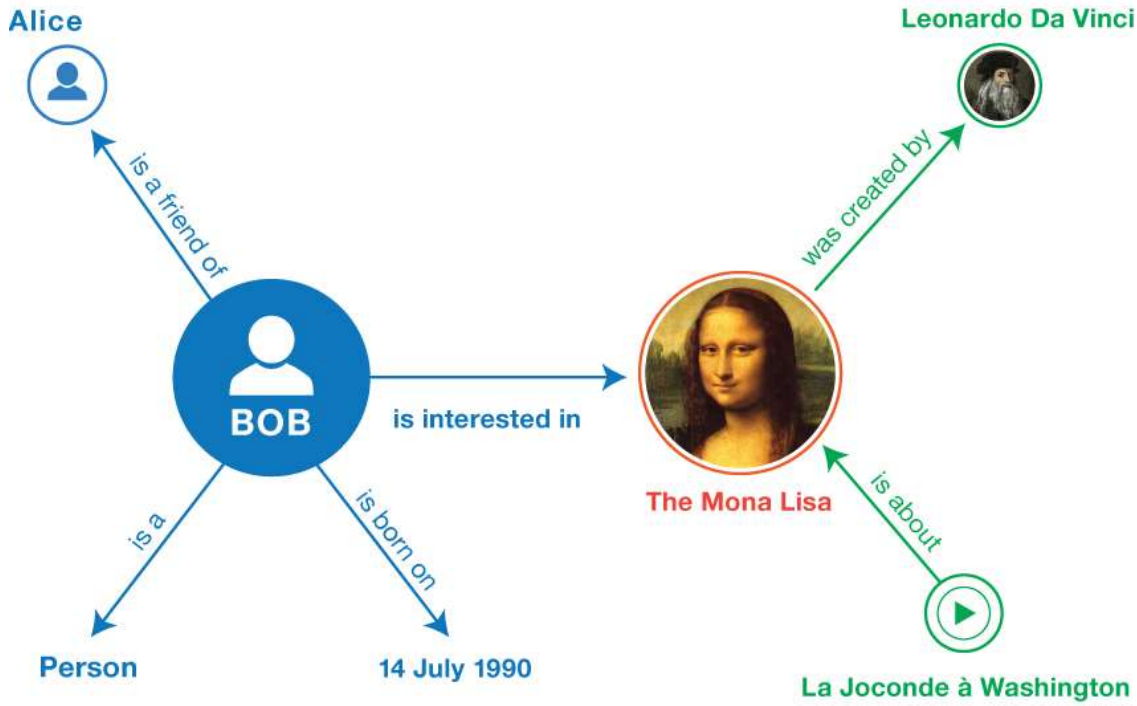

Figura 4 – Grafo RDF em triplas inerentes aos recursos "Bob" e "The Mona Lisa".
Fonte: RDF 1.1 Primer (MANOLA; MILLER; MCBRIDE, 2014).

A partir da perspectiva de "Bob" como recurso, verifica-se, por meio das triplas RDF, que o mesmo é uma pessoa (propriedade: "is a" – valor: "Person"), nasceu em 14 de julho de 1990 (propriedade: "is born on" – valor: "14 july 1990"), é amigo de Alice (propriedade: "is a friend of" – valor: "Alice"), e possui interesse na pintura "The Mona Lisa" (propriedade: "is interested in" – valor: "The Mona Lisa"). É possível observar também que "The Mona Lisa" foi criada por Leonardo da Vinci (recurso: "The Mona Lisa" – propriedade: "was created by" – valor: "Leonardo da Vinci") e que o vídeo "La Joconde à Washington" é sobre "The Mona Lisa" (recurso: "La Joconde à Washington" – propriedade: "is about" – valor:

"The Mona Lisa"). Além disso, torna-se relevante ressaltar que "The Mona Lisa" é o recurso em uma tripla e o valor em duas triplas. A capacidade de um recurso em estar em ambas as posições proporciona maior conectividade entre triplas; característica significativa de um modelo RDF (MANOLA; MILLER; MCBRIDE, 2014). Vale ressaltar que em um grafo RDF real, nós e arestas são identificados (rotulados) por URIs, garantindo a ausência de ambiguidade.

A representação gráfica de um grafo RDF contribui ao entendimento e interpretação por humanos, todavia, é inadequado ao processamento por máquinas (FERREIRA; SANTOS, 2013). Para a publicação factual de grafos (arquivos) RDF na Web, faz-se necessária a definição de sintaxes padrão, como RDF/XML[9], *Turtle*[10], RDFa[11], etc. (AZEVEDO; JACYNTHO, 2014).

É possível publicar arquivos RDF na Web de duas formas distintas, a saber: arquivo RDF puro (RDF/XML, *Turtle*, etc.) e RDF embutido em páginas HTML (RDFa). A sintaxe RDFa permite anotar semanticamente páginas HTML, ou seja, por meio dos atributos dos elementos HTML, inserir metadados RDF que explicitem o significado do conteúdo da página. Com RDFa, ferramentas de busca na Web (Google, Yahoo, Bing, etc.) são capazes de compreender o valor semântico subjacente aos elementos dos documentos publicados em HTML (HERMAN et al., 2015).

Desta forma, com RDF é possível adicionar informações legíveis por máquinas a páginas da Web, viabilizando tanto exibições em um formato aprimorado nos mecanismos de pesquisa, quanto processamentos automáticos em aplicações semânticas. Ademais, fornece um padrão para interligar informações entre bancos de dados, inclusive dentro da mesma organização, a serem realizadas através de uma consulta federada pela linguagem SPARQL (MANOLA; MILLER; MCBRIDE, 2014).

[9] https://www.w3.org/TR/rdf-syntax-grammar/
[10] https://www.w3.org/TR/turtle/
[11] https://www.w3.org/TR/rdfa-syntax/

3. ARQUITETURA PROPOSTA PARA AUTENTICAÇÃO E DISPONIBILIZAÇÃO DE DOCUMENTOS

Este capítulo apresenta a arquitetura proposta, bem como uma breve descrição da trajetória metodológica percorrida para sua elaboração, implementação e teste.

3.1 Trajetória metodológica

A arquitetura, que contempla a autenticação/validação e disponibilização de documentos por meio de *blockchain* e de sistema de arquivos associativo P2P endereçável ao conteúdo, respectivamente, foi delineada ao longo dos seguintes passos:

- Estudo das tecnologias envolvidas: criptografia, *blockchain* e contratos inteligentes, aplicações descentralizadas modernas, armazenamento associativo P2P, Web Semântica (especialmente o modelo de dados RDF);
- Identificação dos componentes arquiteturais e modelagem da arquitetura racional para o registro e validação de documentos, propriamente dita;
- Seleção da *blockchain* e do sistema de arquivos associativo P2P a serem utilizados. Em seguida, seleção e aprendizagem de ferramentas para a implementação da arquitetura por meio de uma aplicação descentralizada (*DApp*), que inclui a elaboração do contrato inteligente e sua integração e interação com a *blockchain*;
- Desenvolvimento do contrato inteligente;
- Desenvolvimento da primeira versão da aplicação descentralizada, implementada em uma *blockchain* de testes local;
- Aprimoramento da aplicação, por meio de testes em *blockchains* de teste públicas, geograficamente distribuídas.

3.2 Módulos da Arquitetura proposta de autenticação e disponibilização de documentos

A arquitetura proposta é dividida em três módulos, a saber: registro de documentos; obtenção de documentos autênticos; e autenticação de documentos.

3.2.1 Módulo de registro dos documentos

Este módulo consiste, basicamente, na geração de *hashes* de autenticação e endereçamento, com base no conteúdo dos arquivos a serem registrados e, em seguida, no registro destas *hashes* junto à *blockchain*, como pode ser observado na Figura 5.

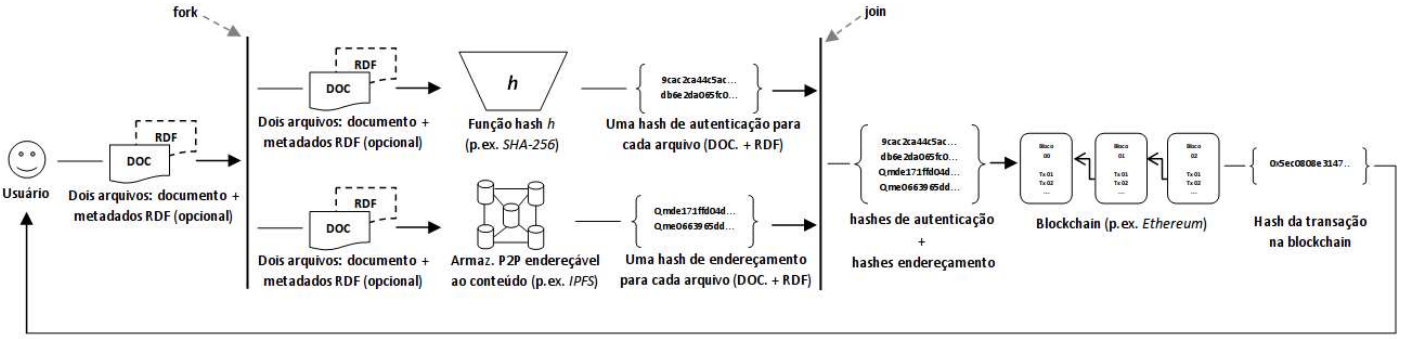

Figura 5 – Módulo de registro dos documentos.
Fonte: Elaboração própria (2019).

Seja o usuário (p.ex. uma instituição), em posse do documento digital a ser registrado (p.ex. um PDF) e, opcionalmente, um arquivo com metadados semânticos em RDF sobre o documento, este módulo da arquitetura contempla as seguintes etapas:

(1) Em paralelo, gerar tanto as *hashes* de autenticação quanto as *hashes* de endereçamento do documento e do arquivo RDF.

- Aplicar função de *hash* (p.ex. SHA-256) no documento e no arquivo RDF, gerando suas respectivas *hashes* que serão usadas para propósito de autenticação;
- O documento e o arquivo RDF são publicados no sistema de armazenamento associativo P2P endereçável ao conteúdo (p.ex. IPFS), garantindo, pois, sua disponibilidade. Como resultado desta publicação, duas *hashes* de endereçamento são geradas

automaticamente pelo sistema de armazenamento P2P, uma para o documento e outra para o arquivo RDF associado. Estas *hashes* são usadas para formar os endereços Web que permitem acessar globalmente ambos os arquivos.

Assim, ao fim desta etapa, são obtidas quatro *hashes*: uma de autenticação e outra de endereçamento, tanto para o documento quanto para o arquivo com metadados RDF associado.

(2) Registrar as quatro *hashes* na *blockchain* por meio de uma transação cujo destino é um contrato inteligente. Obter-se-á, ao fim desta operação, a *hash* de transação; isto é, o identificador do registro na *blockchain*, por meio do qual as quatro *hashes* poderão ser consultadas e sua validação verificada. Cumprida esta etapa, temos o registro imutável e inviolável de que um determinado usuário (p.ex. instituição) reconhece a existência do respectivo documento, por meio da publicação de suas *hashes*.

Embora não seja padrão na arquitetura, uma heurística atrativa seria gerar as *hashes* do documento (DOC) através da função SHA-256 e do registro no sistema de armazenamento P2P (p.ex. IPFS), antes da elaboração do arquivo RDF, para que estas *hashes* possam ser referenciadas no arquivo RDF, uma como identificador do documento (*hash* de autenticação) e outra como endereço Web (URL) de acesso ao mesmo (*hash* de endereçamento), estabelecendo uma ligação explícita entre os dois arquivos. O emprego desta heurística é ilustrado na seção 4.3.1 do estudo de caso realista de uso da arquitetura.

3.2.2 Módulo de obtenção de documentos autênticos

Uma vez realizada a fase de registro dos documentos, estes podem ser facilmente obtidos a partir de suas *hashes* de endereçamento registradas na *blockchain*. Uma vez que ambos, a *blockchain* e o sistema de armazenamento P2P endereçável ao conteúdo, são imutáveis, temos a garantia de que o documento obtido é autêntico. Este módulo da arquitetura é apresentado na Figura 6.

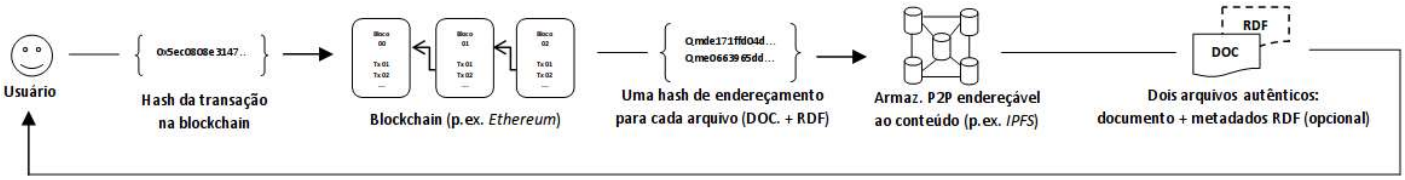

Figura 6 – Módulo de obtenção de documentos autênticos.
Fonte: Elaboração própria (2019).

O usuário, de posse da *hash* da transação da *blockchain*, deve seguir as seguintes etapas:

(1) Acessar a transação na *blockchain*, usando a *hash* da transação;

(2) A partir do registro da transação na *blockchain*, extrair as *hashes* de endereçamento do documento e do arquivo RDF associado;

(3) Com as *hashes* de endereçamento, acessar os endereços Web dos arquivos e obtê-los no sistema de arquivos P2P. Por exemplo, no caso do IPFS, o endereço Web do arquivo no seguinte formato: <*https://ipfs.io/ipfs/hash*>.

Vale ressaltar que neste módulo não são utilizadas *hashes* de autenticação (geradas, por exemplo, pela função SHA-256). De novo, graças ao caráter imutável tanto da *blockchain* quanto do sistema de arquivos P2P endereçável com base no conteúdo, os arquivos obtidos por meio dos endereços registrados na *blockchain* são, por definição, autênticos.

3.2.3 Módulo de autenticação de documentos

Este módulo da arquitetura tem por objetivo autenticar documentos que tenham sido obtidos de fontes quaisquer que não os endereços oficiais registrados na *blockchain*. Como se tratam de fontes não garantidas, estes documentos devem ser verificados quanto a sua autenticidade. Este módulo da arquitetura é descrito na Figura 7.

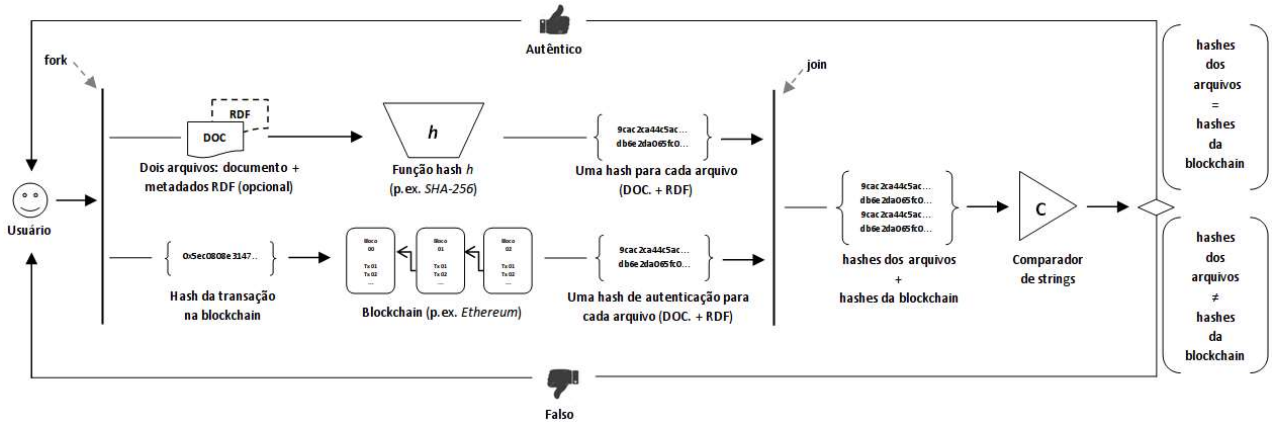

Figura 7 – Modelo de validação dos documentos através do SHA-256.
Fonte: Elaboração própria (2019).

Neste módulo, a autenticação é realizada por meio das *hashes* de autenticação previamente registradas na *blockchain*, geradas pela função de *hash* (p.ex. SHA-256). O usuário que deseja verificar a autenticidade de um documento deve possuir, tanto o documento, quanto a *hash* da transação da *blockchain*:

(1) De posse do documento e do arquivo RDF opcional que se queira autenticar, aplica-se a função de *hash* (p.ex. SHA-256) aos arquivos, obtendo-se as respectivas *hashes*;

(2) Em paralelo, a partir da *hash* de transação da *blockchain*, acessam-se os dados da referida transação na *blockchain*, extraindo as *hashes* de autenticação previamente registradas;

(3) A *hash* de cada arquivo a ser autenticado é então comparada com a *hash* de autenticação correspondente extraída da *blockchain*. Caso a comparação tenha sucesso, ou seja, a *hash* do arquivo seja idêntica à *hash* da *blockchain* o arquivo é autêntico, caso contrário o arquivo é falso.

4. IMPLEMENTAÇÃO DA ARQUITETURA PROPOSTA

Como prova de conceito, este capítulo descreve uma implementação da arquitetura proposta, por meio de uma aplicação descentralizada (*DApp*), usando a *blockchain Ethereum* e o sistema de armazenamento associativo P2P endereçável ao conteúdo IPFS. A rede *Ethereum* foi escolhida por ser uma *blockchain* de propósito geral, com uma máquina virtual de Turing completa, com uma comunidade massiva de desenvolvedores, que cresce cada vez mais. Como resultado desse crescimento, o *Ether*, a criptomoeda da rede *Ethereum*, é a segunda criptomoeda em capitalização de mercado, ficando atrás apenas da criptomoeda *Bitcoin*. Já o IPFS é um sistema robusto, sendo um dos mais utilizados para o seu propósito, e inclusive, visa superar o HTTP, na Web distribuída do futuro.

Este capítulo aborda a seleção das ferramentas utilizadas, a elaboração de um contrato inteligente, o desenvolvimento da aplicação em ambiente simulatório local e em ambientes descentralizados, além de demonstrações em exemplos exploratórios e realistas. Ademais, a aplicação descentralizada (*DApp*), assim como o seu código, na íntegra, podem ser acessados no *GitHub*[12].

4.1 Ferramentas utilizadas

Para implementar a referida arquitetura, algumas ferramentas foram cuidadosamente avaliadas e selecionadas.

4.1.1 Remix IDE

Remix[13] é um ambiente de desenvolvimento integrado (IDE) escrito em *JavaScript* e em código aberto que permite escrever, testar, depurar e implementar contratos inteligentes em *Solidity*, linguagem mais popular da plataforma *Ethereum*, direto do navegador. O editor do *Remix* recompila o código cada vez que o contrato é alterado, além de realçar a sintaxe mapeada à linguagem *Solidity*. A ferramenta possibilita testar as funções do contrato e

[12] https://github.com/lguicarvalho/Inbox
[13] https://remix.ethereum.org/

realizar transações que podem posteriormente ser depuradas através do seu *console* (“Remix documentation”, 2018).

4.1.2 Truffle

Além de um ambiente de desenvolvimento de primeira classe, *Truffle*[14] é também um *framework* de testes para *blockchains* utilizando a máquina virtual *Ethereum*, a fim de tornar mais fácil a vida dos desenvolvedores. Com *Truffle*, é possível compilar contratos inteligentes e testá-los de forma automatizada para rápido desenvolvimento, por exemplo. Além disso, o *framework* possui gerenciamento de rede para implantação em qualquer rede pública ou privada (“Truffle documentation”, 2019).

4.1.3 Ganache

Orientado à *blockchain Ethereum*, *Ganache*[15] funciona como uma *blockchain* privada e é utilizado para implementar contratos, desenvolver aplicações descentralizadas e executar testes. Por simular uma *blockchain* em memória, é possível acompanhar as saídas de log (*log outputs*) geradas pelas transações, estabelecer configurações avançadas de mineração e verificar todos os blocos e transações com o propósito de compreender o funcionamento do sistema (“Ganache documentation”, 2019).

4.1.4 Lite-server

Utilizado para desenvolver aplicações Web de forma mais acessível, *Lite-server*[16] é um servidor Web simples com suporte para aplicações de uma única página (SPAs, sigla em inglês), que atualiza quando há alteração no HTML ou *Javascript*, insere alterações de CSS e tem uma página de *fallback* quando uma rota não é encontrada (PAPA, 2019).

[14] https://www.trufflesuite.com/truffle
[15] https://www.trufflesuite.com/ganache
[16] https://github.com/johnpapa/lite-server

4.1.5 MetaMask

MetaMask[17] é uma extensão que possibilita realizar transações na rede *Ethereum* através do navegador, permitindo que usuários gerenciem contas e suas chaves (privadas e públicas) de várias maneiras, incluindo carteiras de hardware, isolando-as do contexto do site (KUMAVIS et al., 2019).

A ferramenta injeta uma instância *Web3* em *JavaScript* no navegador, agindo como um cliente RPC (*gateway*) que se conecta a uma variedade de *blockchains Ethereum*, como a rede principal, e redes de testes, como *Ropsten*, *Kovan*, *Rinkeby*, ou um nó RPC local, como *Ganache* (ANTONOPOULOS; WOOD, 2018).

4.2 Desenvolvimento da aplicação descentralizada (DApp)

Esta seção se destina ao desenvolvimento da aplicação descentralizada (*DApp*), que inclui a elaboração do contrato inteligente, o desenvolvimento da *DApp* em *blockchain* privada e pública, além de testes exploratórios em ambos os níveis de privacidade.

Primeiramente foi feito um protótipo, em *blockchain* privada. Este protótipo foi paulatinamente aprimorado, gerando versões cada vez mais completas e robustas da aplicação, que serão apresentadas a seguir.

4.2.1 Elaboração do contrato inteligente

Para desenvolver a aplicação foi preciso, em primeiro lugar, codificar um contrato inteligente que cumprisse os requisitos, ou seja, que registrasse, na transação da *blockchain Ethereum,* as quatro *hashes* necessárias para a validação e disponibilização de um documento. O código elaborado para este contrato pode ser observado na Figura 8.

[17] https://metamask.io/

```
1  pragma solidity >=0.4.16 <0.6.0;
2
3  contract Inbox {
4
5      address publisher;
6      string PDF1;
7      string RDF1;
8      string PDF2;
9      string RDF2;
10
11
12     event LogStoreHash(
13       address indexed _publisher,
14       string hash1,
15       string hash2,
16       string hash3,
17       string hash4);
18
19     function storeHash
20     (string memory hash1,
21      string memory hash2,
22      string memory hash3,
23      string memory hash4)
24      public {
25         publisher = msg.sender;
26         PDF1 = hash1;
27         RDF1 = hash2;
28         PDF2 = hash3;
29         RDF2 = hash4;
30
31         emit LogStoreHash(publisher, hash1, hash2, hash3, hash4);
32     }
33
34     function getMessage() public view returns
35     (address _publisher,
36      string memory hash_do_PDF,
37      string memory hash_do_RDF,
38      string memory PDF_no_IPFS,
39      string memory RDF_no_IPFS)
40      {
41         return (publisher, PDF1, RDF1, PDF2, RDF2);
42     }
43
44 }
```

Figura 8 – Código do contrato inteligente para registrar hashes na blockchain Ethereum.
Fonte: Elaboração própria (2019).

O contrato desenvolvido, denominado *Inbox* (linha 3), opera através de cinco variáveis de estado (linhas 5 a 9). Enquanto a variável *publisher* é do tipo *address* (endereço), as variáveis *PDF1, RDF1, PDF2* e *RDF2* são do tipo *string.* Variáveis do tipo *address* contêm um endereço de conta *Ethereum*.

A função *storeHash* (linha 19) tem por objetivo armazenar as variáveis *hash1, hash2, hash3* e *hash4*, nas respectivas variáveis de estado do contrato (linhas 26 a 29). Ainda, ressalta-se que, na linha 25, a variável de estado *publisher* equivale ao remetente da operação (*msg.sender*), ou seja, o endereço da conta *Ethereum* que iniciou a transação que invocou esta operação do contrato.

Por conseguinte, a função *getMessage* retorna os valores atribuídos na função *storeHash*, assim como o endereço da conta que realizou a transação (linha 41).

Além destas funções, aprecia-se a utilização de eventos no contrato. Eventos são registros acionados por transações e incorporados à estrutura da *blockchain*, associados ao endereço do contrato inteligente a que se referem ("Contracts — Solidity 0.5.11 documentation", 2019). Eventos neste contrato inteligente possibilitam a realização de consultas aos registros de transações que invocaram este contrato, diretamente pela aplicação descentralizada, de acordo com as variáveis de estado estabelecidas no evento, codificado no contrato (linhas 13 a 17). Assim, um evento *LogStoreHash* é disparado ao final da execução da função *storeHash* (linha 31), armazenando as informações pertinentes à transação na *blockchain*, como as variáveis estabelecidas e outros dados, como a *hash* da transação e a *hash* do bloco, em um formato estruturado de dados, permitindo que estas informações sejam gerenciadas pela aplicação descentralizada.

É importante reiterar a verificação do código do contrato por meio da ferramenta *Remix* IDE, que permite apurar erros de programação, assim como inconsistências inerentes à versão do compilador *Solidity* utilizado.

4.2.2 Desenvolvimento da aplicação descentralizada em um ambiente simulatório local

Após a elaboração do contrato inteligente, foi realizada, através do comando *truffle migrate --compile-all --reset --network ganache*, sua compilação junto ao *framework Truffle* e registro na *blockchain Ethereum* privada e local *Ganache*, como mostra a Figura 9.

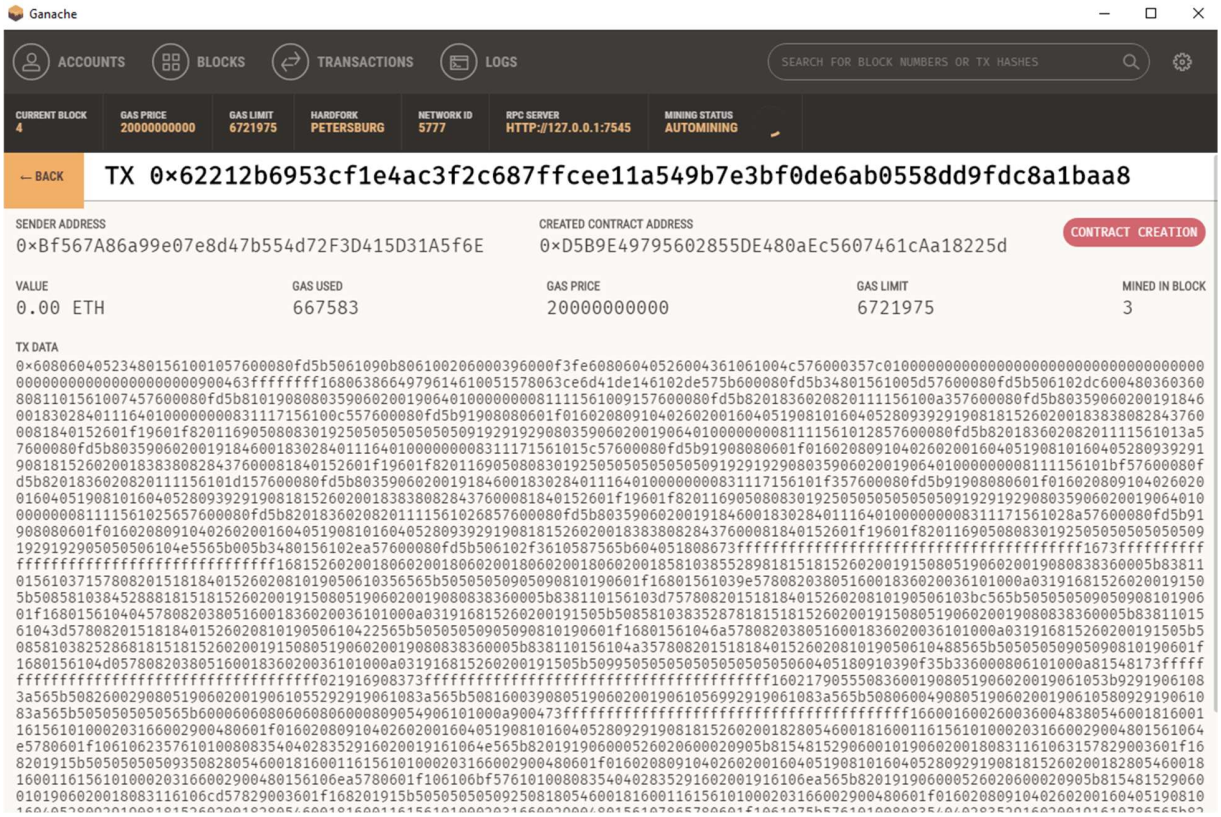

Figura 9 – Exibição do registro de implementação do contrato no simulador Ganache.
Fonte: Elaboração própria (2019).

O registro de criação do contrato atenta a informações relevantes, como a *hash* de transação de criação do contrato (*TX*), o endereço da conta que criou a transação (*sender address*), o endereço do contrato criado (*created contract address*) e a quantidade de *gas* consumida para implementar o contrato na *blockchain* (*gas used*).

O último passo para o desenvolvimento da aplicação descentralizada envolveu a estruturação de uma interface Web em HTML para que usuários interajam diretamente com o contrato inteligente registrado na *blockchain Ganache*. Este *frontend* foi elaborado através do *Lite-server*, e pode ser observado na Figura 10.

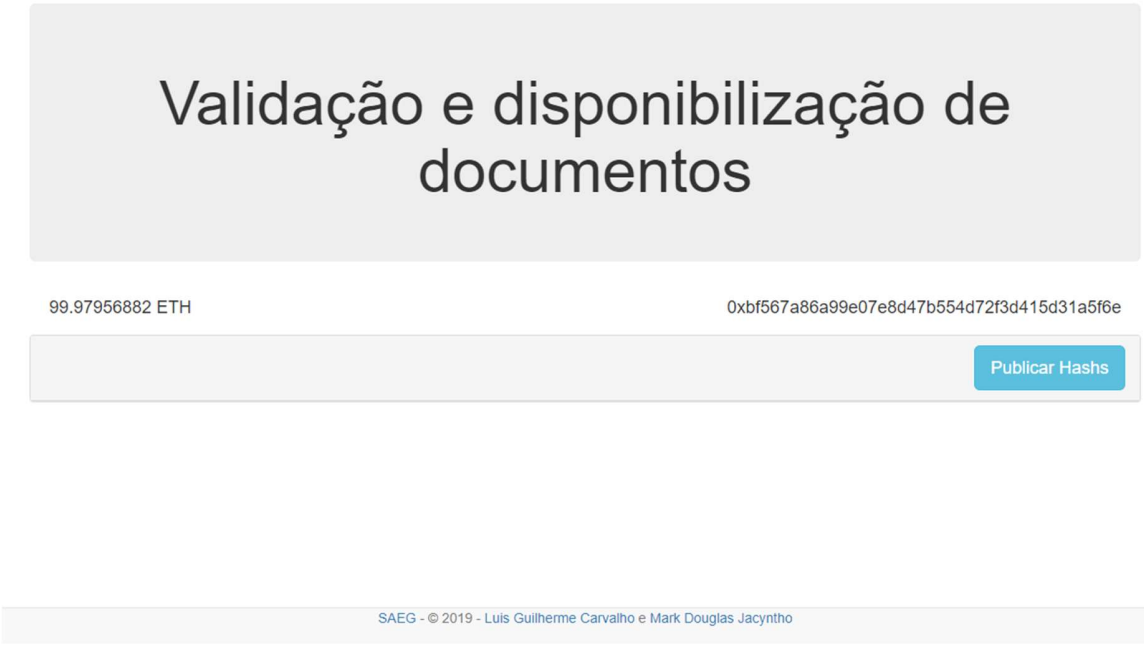

Figura 10 – Interface web da aplicação descentralizada (DApp) em HTML.
Fonte: Elaboração própria (2019).

A interface apresenta elementos referentes à conta *Ethereum* pré-selecionada para realizar a operação: endereço (chave pública) e balanço em *ether*. É importante salientar que esta foi uma das dez contas geradas automaticamente pelo *Ganache* com balanço inicial de 100 *ethers* (ETH), e a mesma foi utilizada para criar o contrato (Figura 9). Desta forma, devido aos custos de implementação, seu balanço atualizado na interface é de aproximadamente 99,98 *ethers*.

O botão *Publicar Hashes* referencia a função *storeHash* do contrato inteligente (Figura 8), permitindo registrar na *blockchain* as *hashes* inerentes ao documento a ser validado e disponibilizado. Ao pressionar este botão, as informações são registradas automaticamente na *blockchain Ganache* através do contrato inteligente (*backend*), caracterizando a aplicação descentralizada, e são exibidas na interface (*frontend*).

4.2.2.1 Exemplo exploratório de uso da versão piloto da aplicação

Para testar a aplicação, foram selecionados o documento PDF[18] de especificação da ontologia FOAF e sua versão RDF (BRICKLEY; MILLER, 2014). Ao calcular as *hashes* de autenticação pela função SHA-256, foram obtidos os seguintes resultados:

- PDF: 53dc735a544a826d9e5133ec0fce0a9e308bd5e0053f1f2374b2182831b66268;
- RDF: 3d859b5d92a2c3d041545d014fa826f682ca06d056af8c7b31e32d930abf2bc5.

Em seguida, os documentos foram submetidos ao sistema de arquivos P2P IPFS, e as seguintes *hashes* de endereçamento foram extraídas:

- PDF: QmYNT2J6b1vjJod87a5J8ktpYytc4HTV3FbDuaBHLrTznX;
- RDF: QmeVRmyWzTwjijmGuqC4XPw7xJpQMHoyypK5bkb5q7Rizi.

Reitera-se que, para visualizar determinado arquivo em um navegador, basta acessar <*https://ipfs.io/ipfs/hash-do-arquivo*>.

De posse das quatro *hashes*, torna-se possível a realização do registro através da aplicação descentralizada, como mostra a Figura 11.

[18] Foi utilizado um arquivo PDF, mas poderia ter sido utilizado qualquer tipo de arquivo.

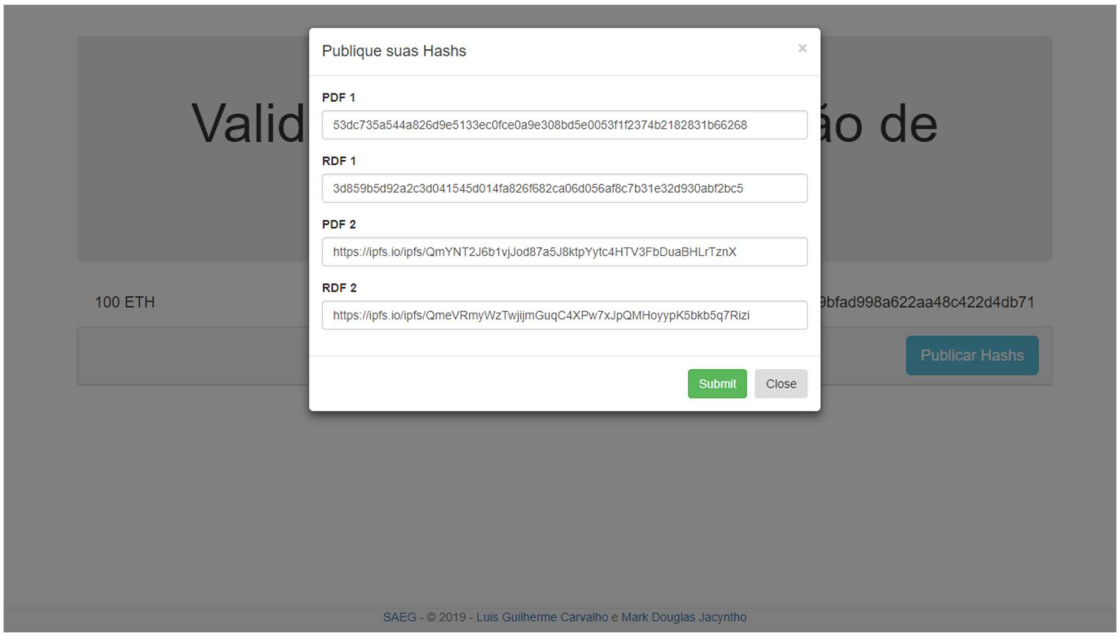

Figura 11 – Tela de registro das hashes na aplicação descentralizada (DApp).
Fonte: Elaboração própria (2019).

Nota-se que cada uma das *hashes* corresponde a uma variável de estado mencionada no desenvolvimento do contrato inteligente. Ademais, optou-se por publicar os endereços Web (*https*) dos arquivos no IPFS, ao invés de *hashes* simples, permitindo o acesso direto pela aplicação.

Ao clicar no botão *Submit*, a função *storeHash* do contrato inteligente é ativada e, neste momento, é exibida uma janela para confirmação através da extensão *MetaMask*, com informações acerca da interação com o contrato, como mostra a Figura 12.

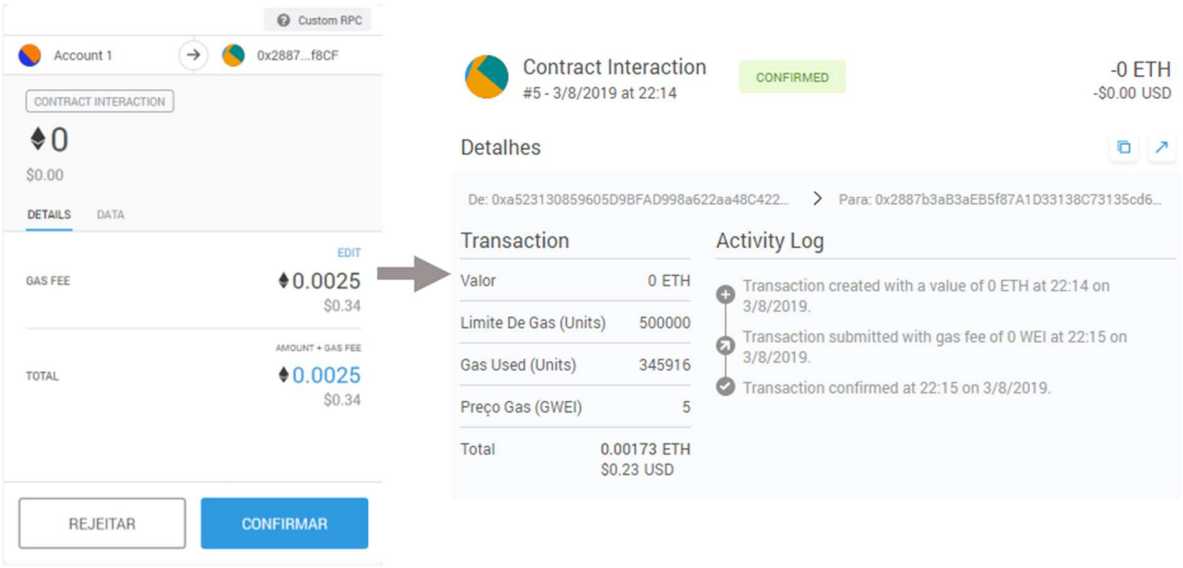

Figura 12 – Telas pré e pós-confirmação da operação *storeHash* do contrato inteligente.
Fonte: Elaboração própria (2019).

Na tela pré-operação, constatam-se dados pertinentes, como a conta a realizar a operação (*account 1*), o endereço do contrato (*0x2887...f8CF*), a quantidade de *ether* a ser enviada (*0 ether*, por se tratar de uma operação de registro, e não uma transação financeira) e a estimativa de custos operacionais (*gas fee*).

Uma vez confirmada, são então geradas informações definitivas a respeito da transação, uma espécie de extrato, como os endereços do remetente (*0xa523...*) e do destinatário – contrato (*0x2887b...*) –, os custos operacionais reais (*gas used*) e o registro de atividades (*activity log*). Ao acessar o registro de atividades, o usuário é direcionado ao *site etherscan.io*, uma plataforma de análise e exploração de blocos da rede *Ethereum* (TAN et al., 2019). Ressalta-se que o *etherscan.io* disponibiliza informações somente sobre *blockchains* públicas. Para acessar as informações completas da transação nesta aplicação, é preciso acessá-las no gerenciador da *blockchain Ganache*, como mostra a Figura 13.

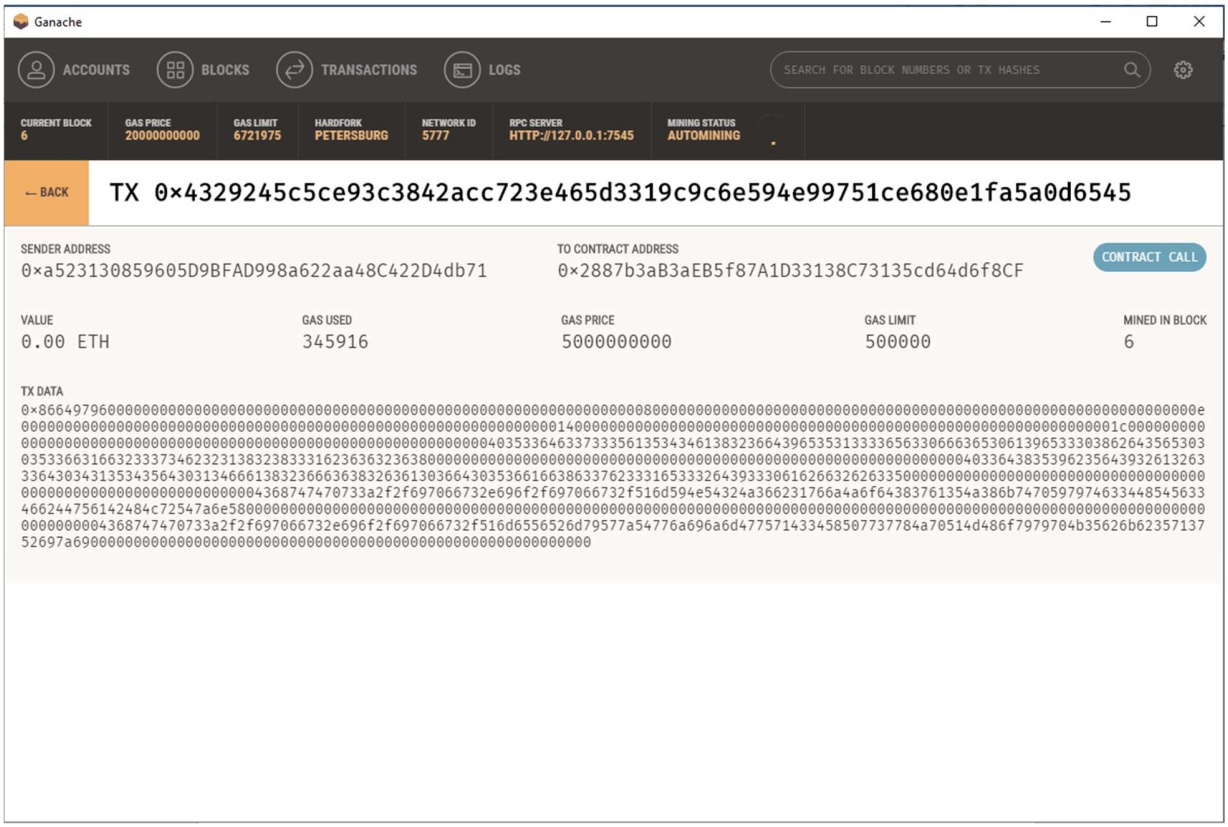

Figura 13 – Registro da transação na blockchain privada Ganache.
Fonte: Elaboração própria (2019).

Além das informações já percebidas anteriormente, o registro também fornece os dados de entrada, isto é, as *hashes* atribuídas através da aplicação (*tx data*). Observa-se que os dados de entrada estão no formato hexadecimal; para melhor visualização, basta convertê-los ao formato texto.

A fim de estabelecer um ambiente mais amigável e funcional, os dados de entrada também são exibidos no *frontend* da aplicação após a submissão das *hashes*, e podem ser contemplados na Figura 14.

Figura 14 – Exibição da aplicação descentralizada (DApp), após a publicação das hashes.
Fonte: Elaboração própria (2019).

Além das *hashes* de autenticação geradas pelo algoritmo SHA-256 (*PDF1* e *RDF1*) e das *hashes* de endereçamento IPFS, acrescidas do *gateway* público do IPFS (*PDF2* e *RDF2*), também é exibido o endereço público da conta que as publicou (*publicado por*). Caso a conta ativa (*0xa523...b71*) na carteira do *MetaMask* seja a mesma que publicou as *hashes*, a variável retorna "Você".

Vale salientar que a versão apresentada nesta seção é um protótipo. A utilização de uma *blockchain* de teste local (*Ganache*) permitiu a análise da interação entre o usuário e o contrato inteligente integrado a uma *blockchain*, sem a exposição de dados ao mundo externo. Ainda que neste projeto piloto da aplicação descentralizada as *hashes* tenham sido registradas em uma *blockchain* com sucesso, a geração das *hashes* foi realizada por ferramentas externas. Neste ínterim, estimou-se a integração destas ferramentas à aplicação, de modo que a aplicação seja a única ferramenta necessária a cumprir todas as etapas da arquitetura proposta, reduzindo o número de interações até a validação, e melhorando, sobremaneira, a experiência do usuário. Este e outros aprimoramentos serão abordados a seguir.

4.2.3 Desenvolvimento da aplicação descentralizada em *blockchains Ethereum* públicas de teste (*testnets*)

Após o uso da aplicação em uma *blockchain* local e centralizada, tornou-se importante experimentá-la em ambientes descentralizados, próximos à realidade da rede principal da *blockchain Ethereum*.

A comunidade *Ethereum* mantém várias redes de teste. Elas são utilizadas pelos desenvolvedores para testar suas aplicações sob diversas condições realistas, sem custo, antes de implantar na rede principal do *Ethereum* (“Developers | Ethereum”, 2019).

Neste contexto, a aplicação foi implementada em duas redes de teste (*testnets*): *Rinkeby* e *Ropsten*. Enquanto a rede *Ropsten* utiliza o protocolo *Proof of Work* (o mesmo da rede principal do *Ethereum*), no qual os mineradores devem resolver problemas matemáticos através de processamento computacional, a rede *Rinkeby* possui o protocolo *Proof of Authority*; isto é, uma rede centralizada, na qual a validação das transações é realizada por um grupo específico de pessoas (“Developers | Ethereum”, 2019).

A Figura 15 exibe a aplicação no seu estado atual. Como explicitado na seção anterior, fez-se importante a incorporação das ferramentas de geração das *hashes* à aplicação, a qual foi realizada, e pode ser observada a seguir.

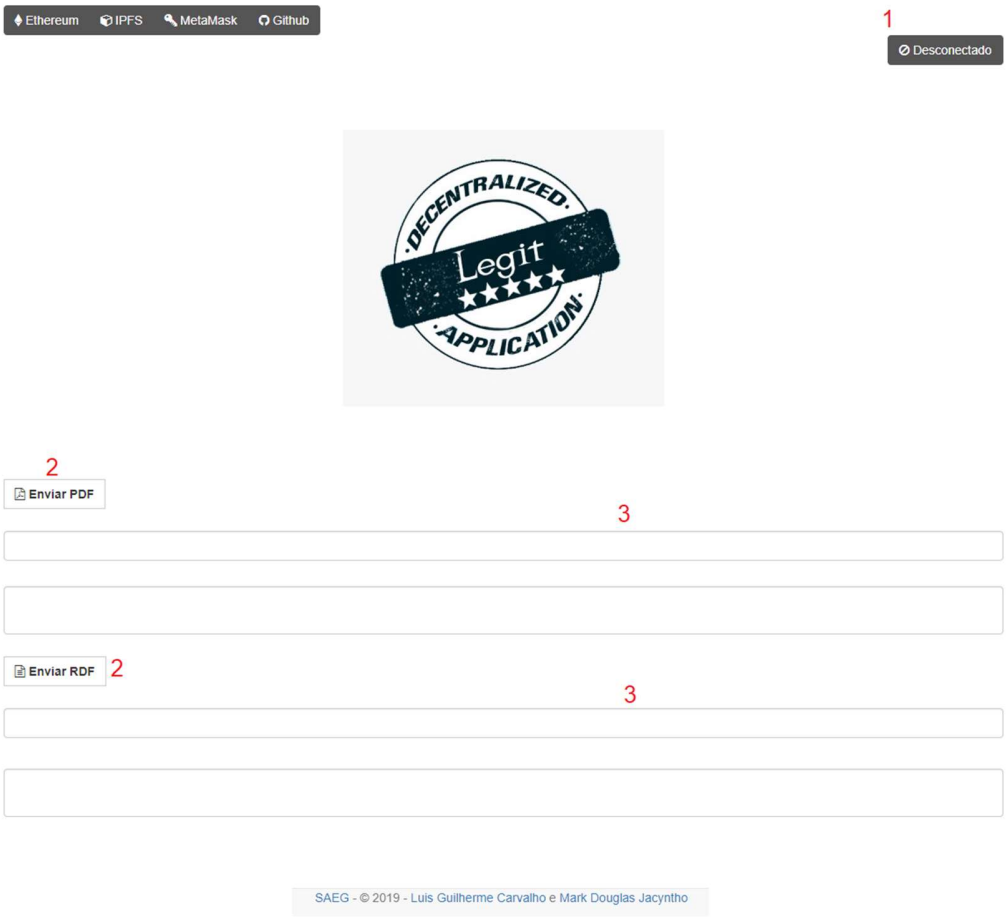

Figura 15 – Versão atual da aplicação descentralizada (DApp), desconectada.
Fonte: Elaboração própria (2019).

É possível observar na imagem que a *DApp* se encontra desconectada (item 1) da rede *Ethereum*; isto é, não possui qualquer provedor de *Web3* ativo. Nesta configuração apresentada, o usuário pode utilizar tão somente as ferramentas de geração das *hashes* IPFS e SHA256; basta enviar um arquivo PDF ou RDF (item 2), e então as *hashes* são geradas automaticamente através de *Javascript* (item 3).

Enquanto a geração da *hash* SHA256 de um arquivo implica apenas a aplicação de um algorítimo criptográfico, uma *hash* gerada pelo IPFS significa que o arquivo já se encontra publicado no sistema de arquivos P2P e pode ser acessado por qualquer um que tenha conhecimento da respectiva *hash*, desde que o arquivo esteja hospedado por um nó.

Cabe ressaltar que registrar arquivos no IPFS não implica a sua validação; apenas a sua disponibilização. Para que se torne um arquivo crível, é preciso registrar as *hashes*

geradas na *blockchain Ethereum*, e isto só pode ser realizado utilizando um provedor *Web3,* como a extensão de navegador *MetaMask.*

Ao acionar a extensão *MetaMask* no navegador, é injetada uma instância *Web3* no contexto de *Javascript* do navegador, permitindo a comunicação entre a *DApp* e a rede *Ethereum* selecionada. Além da rede, a extensão também possibilita ao usuário estabelecer a carteira a ser utilizada, na qual consta a conta a realizar as transações. Após definir estes parâmetros, a *DApp* apresenta a configuração apresentada na Figura 16.

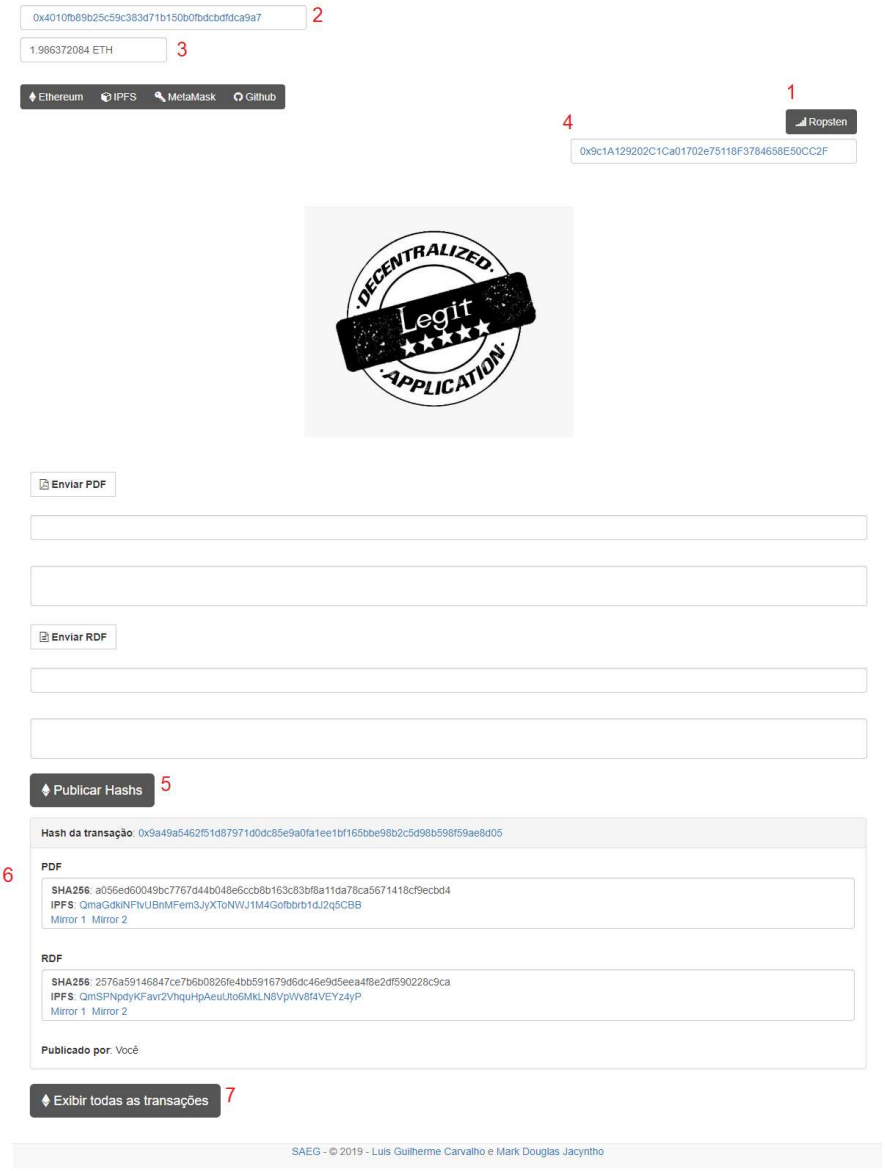

Figura 16 – Versão atual da aplicação descentralizada (DApp), conectada.
Fonte: Elaboração própria (2019).

A *DApp* então apresenta novas informações oriundas da rede *Ethereum*, mais especificamente do contrato inteligente associado à aplicação que fora registrado na referida *blockchain*. Seguindo a numeração da Figura 16, temos:

1. Rede selecionada: apresenta o nome da rede definida na extensão *MetaMask*. Frisa-se que, para a interação, é necessário que o contrato inteligente desenvolvido tenha sido registrado (publicado) na rede selecionada. Caso a rede seja alterada, a aplicação é automaticamente atualizada para refletir o estado atual da extensão. Na Figura 16, a rede selecionada é a *Ropsten*;

2. Conta selecionada: exibe o endereço público da conta definida na extensão *MetaMask*. Caso a conta seja alterada, a aplicação é automaticamente atualizada para refletir o estado atual da extensão;

3. Saldo em *ether*: expõe o saldo em *ether* da conta selecionada;

4. Contrato implementado: endereço do contrato inteligente associado à aplicação, registrado na rede selecionada. O contrato apresenta um endereço diferente para cada rede em que foi registrado;

5. Publicar *hashes*: o botão aciona a função *storeHash* do contrato inteligente, permitindo registrar as *hashes* inerentes aos documentos enviados;

6. Última transação: apresenta a última operação realizada dentro do contrato. Nesta tabela, podem-se observar informações importantes, como a *hash* da transação que invocou o contrato e as *hashes* de autenticação SHA256 e de endereçamento IPFS dos documentos PDF e RDF, respectivamente;

7. Exibir todas as transações: o botão aciona o evento *LogStoreHash* do contrato inteligente, permitindo consultar todas as transações que invocaram o contrato, da respectiva rede.

Desta forma, todas as informações pertinentes ao esquema de disponibilização e autenticação desenvolvido são exibidas na aplicação, tornando-a autocontida. Adicionalmente, para fins de confirmação, existem *hiperlinks* associados às *hashes* IPFS e endereços públicos, redirecionando o usuário a *gateways* de acesso ao documento e ao *blockchain explorer etherscan.io*, respectivamente.

4.2.3.1 Exemplo exploratório de uso da versão atual da aplicação

Inicia-se o teste da *DApp*, utilizando os mesmos documentos do exemplo anterior (vide seção 4.2.2.1), como mostra a Figura 17. Ao enviar os arquivos, é possível observar as mesmas *hashes* geradas.

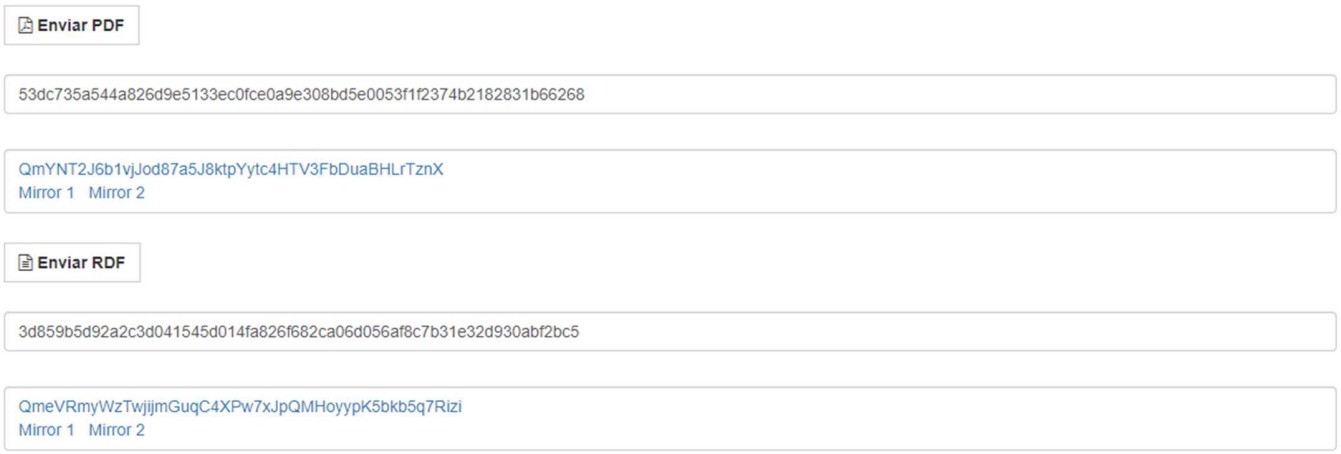

Figura 17 – Envio de documentos na versão atual da aplicação descentralizada (DApp).
Fonte: Elaboração própria (2019).

Reitera-se que, na seção 4.2.2.1, a geração das *hashes* foi realizada de forma segregada à aplicação; isto é, ferramentas de terceiros para geração da *hash* SHA256 e a instalação local de um nó IPFS para envio do documento ao respectivo sistema de arquivos distribuído. Nesta seção, as *hashes* são geradas de forma integrada à aplicação; há códigos em *Javascript* para o algoritmo criptográfico SHA256 e para o envio de arquivos ao IPFS.

Além disso, é possível, a partir da *hash* IPFS, acessar os arquivos hospedados neste sistema de arquivos, através de três diferentes *gateways* gerados pela aplicação.

Após acionar o botão “Publicar *Hashes*” e confirmar a operação junto ao *MetaMask*, as *hashes* registradas são disponibilizadas no HTML, como mostra a Figura 18.

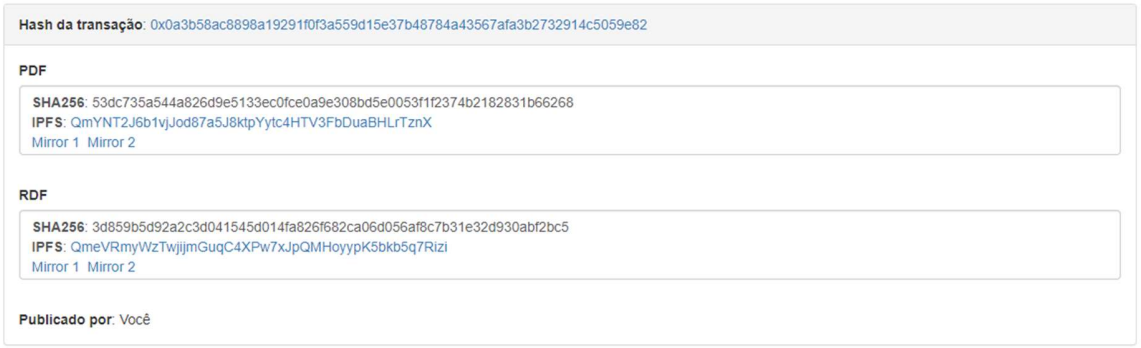

Figura 18 – Exibição da função *getMessage* na versão atual da aplicação descentralizada (DApp).
Fonte: Elaboração própria (2019).

A Figura 18, que contém informações registradas na última transação realizada na *blockchain*, apresenta o retorno da chamada função *getMessage* do contrato inteligente utilizado (vide Figura 8).

Ao acessar o *link* “*Hash* da transação”, o usuário é redirecionado ao *blockchain explorer etherscan.io*, como mostra a Figura 19.

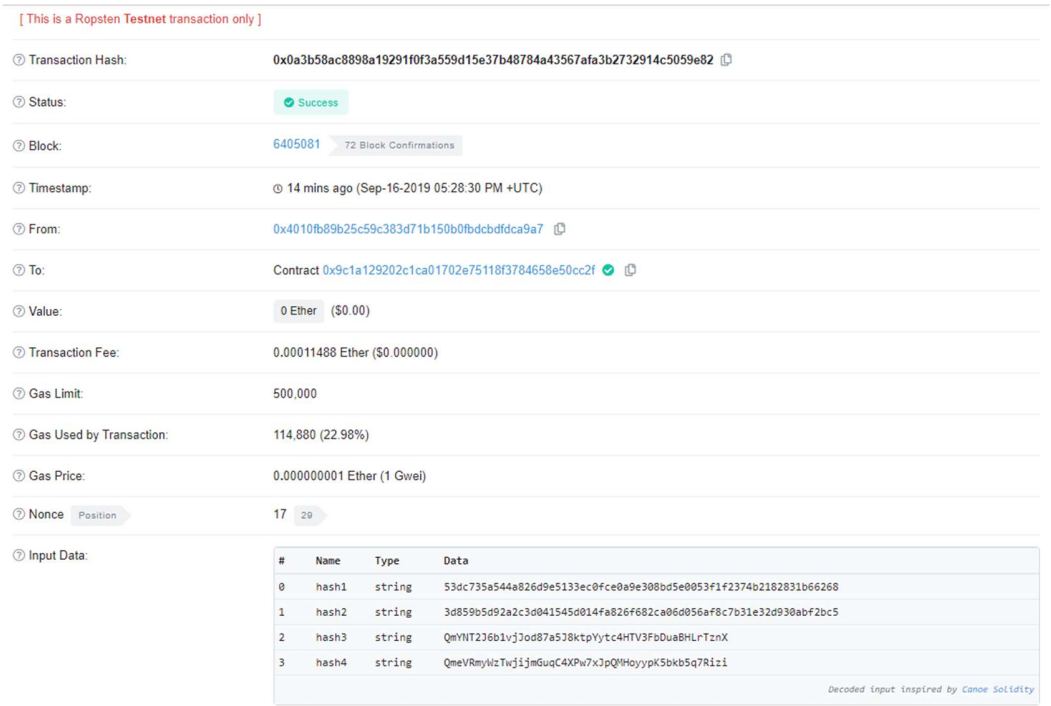

Figura 19 – Exibição dos detalhes da transação.
Fonte: Blockchain Explorer Etherscan (ETHERSCAN.IO, 2019).

Assim, o usuário é capaz de obter a confirmação de que a transação foi realizada com sucesso. Além disso, é possível extrair informações complementares, como o registro temporal (*Timestamp*) e o *gas* efetivamente consumido para realizar a transação (*Gas Used by Transaction*).

Concluídas as etapas anteriores e registrados os documentos, é permitido ao usuário verificar todas as transações realizadas dentro do contrato vigente nesta aplicação, seja pelo *etherscan.io* ou pela própria aplicação.

Utilizando o *etherscan.io,* basta acessar o *link* do contrato exibido na Figura 16 (*<0x9c1a129202c1ca01702e75118f3784658e50cc2f>*). Pela aplicação, o botão "exibir todas as transações" aciona o evento *LogStoreHash* do contrato inteligente, possibilitando a visualização das transações, como mostra a Figura 20.

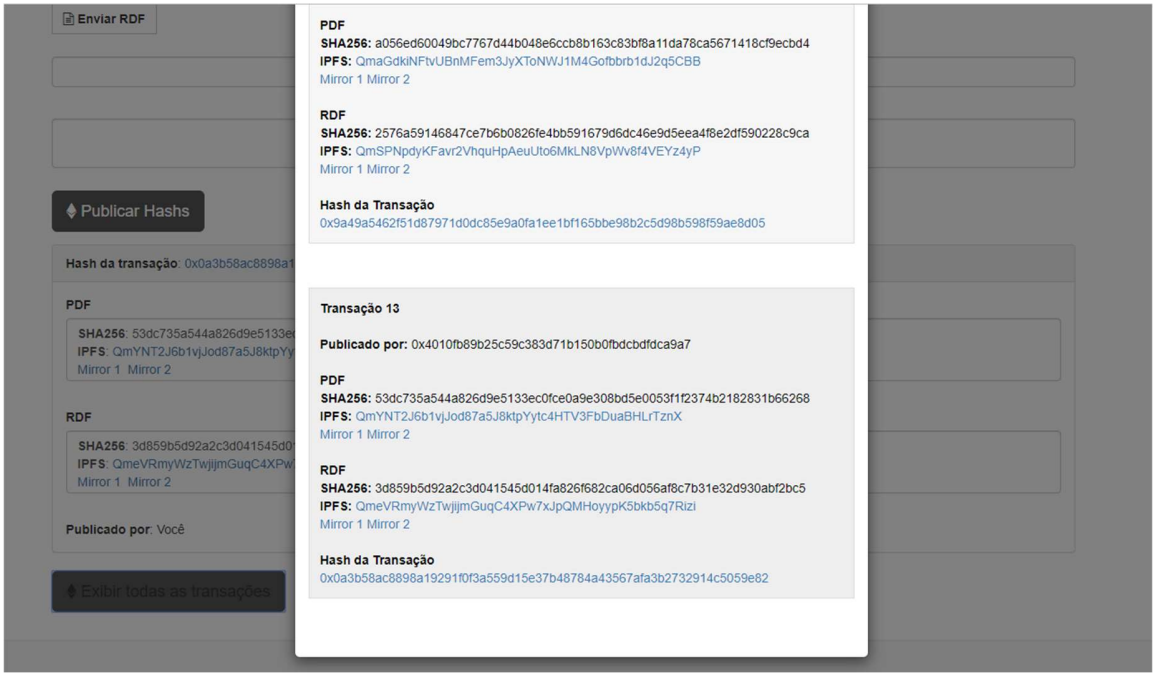

Figura 20 – Exibição do evento *LogStoreHash* na versão atual da aplicação descentralizada (DApp).
Fonte: Elaboração própria (2019).

Desta forma, é permitido ao usuário analisar as transações realizadas, a partir das variáveis definidas no evento pelo contrato inteligente.

Os aprimoramentos realizados, do protótipo à versão atual, almejam a integração das ferramentas requisitadas e o gerenciamento de informações à cerca do contrato inteligente e da *blockchain Ethereum*, de modo a garantir a completude da aplicação descentralizada

(*DApp*), permitindo que o usuário consiga realizar todas as tarefas de publicação/acesso e autenticação dos documentos, previstas na arquitetura proposta, dentro da própria aplicação, sem precisar do auxílio de outras aplicações externas.

4.3 Estudo de caso realista de uso da arquitetura e da aplicação (DApp)

Após discorrer sobre as funcionalidades da versão atual da *DApp*, para ilustrar ainda mais sua aplicabilidade e a arquitetura proposta, cabe explorá-la por meio de um estudo de caso realista e funcional de uso: a autenticação de um diploma emitido por uma instituição de ensino.

4.3.1 Envio e registro do documento digital

A Figura 21 mostra um exemplo de documento digital, um diploma de mestrado fictício (porém realista), a ser registrado pela referida instituição.

Figura 21 – Exemplo de diploma a ser registrado na blockchain pela instituição.
Fonte: Elaboração própria (2020).

É possível também apurar outras informações, como o nome do aluno, a data de reconhecimento, e a pessoa responsável pela autenticação diploma. Além de possuir este

documento digital em PDF, cabe à instituição, opcionalmente, elaborar a versão RDF do documento. Um exemplo factível de metadados RDF pode ser observado na Figura 22.

```
@prefix schema: <http://schema.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://ensino.edu.br/mestrado/curso/documentos/diploma/45784>
  a schema:EducationalOccupationalCredential ;
  schema:about <http://ensino.edu.br/aluno/JoseDasFlores> ;
  schema:credentialCategory <http://dbpedia.org/resource/Degree> ;
  schema:educationalLevel <http://dbpedia.org/resource/Master_degree> ;
  schema:accountablePerson <http://ensino.edu.br/professor/MariaSilva> ;
  schema:recognizedBy <http://ensino.edu.br> ;
  schema:dateCreated "2020-05-31"^^xsd:date ;
  schema:identifier "45784", "e35a18e171f1ec061aa391043159ffacedf0ee682c439626ac7c01baf75635cd" ;
  schema:URL <https://ipfs.io/ipfs/Qmbj2GUgV4Ew9MPuCN9ZGcSCoaBB2xtCUqRMRWwmAQz3qn> .

<http://ensino.edu.br/mestrado/curso>
  a schema:Course, schema:EducationalOccupationalProgram ;
  schema:educationalCredentialAwarded <http://ensino.edu.br/mestrado/curso/documentos/diploma/45784> ;
  schema:provider <http://ensino.edu.br> .
```

Figura 22 – Exemplo de arquivo com metadados RDF relativos ao diploma a ser registrado na blockchain (sintaxe *Turtle*).
Fonte: Elaboração própria (2020).

Cabe destacar o relacionamento entre as informações, expressas em triplas RDF no diploma a ser autenticado. Neste exemplo, o recurso principal é o próprio diploma (<http://ensino.edu.br/mestrado/curso/documentos/diploma/45784>), e as informações presentes no documento original são estruturadas usando a ontologia schema.org[19]. É possível observar, por exemplo, que: o documento retrata um *diploma de mestrado* (propriedade: *educationalLevel*, valor: *<http://dbpedia.org/resource/Master_degree>*); a pessoa responsável pelo reconhecimento do diploma é *Maria Silva* (propriedade: *accountablePerson*, valor: *<http://ensino.edu.br/professor/MariaSilva>*); e a entidade principal do diploma, isto é, o aluno, é *José das Flores* (propriedade: *about*, valor: *<http://ensino.edu.br/aluno/JoseDasFlores>*).

A representação em RDF assegura a compreensão por máquinas, permitindo a realização de inferências e consultas complexas a partir das triplas. Usando uma base de dados semântica, seria possível, por exemplo, por meio de uma consulta SPARQL, verificar todos os diplomas de mestrado que foram reconhecidos na responsabilidade de *Maria Silva*,

[19] http://schema.org/

ou ainda, verificar todos documentos digitais gerados pela instituição de ensino no dia *31 de maio de 2020*.

Destacam-se, ainda, as propriedades *identifier* e *URL*, cujos valores são as *hashes* de autenticação (SHA256) e de endereçamento (IPFS) do documento PDF do diploma, respectivamente. Para tal, o usuário precisa enviar o documento PDF antes da elaboração do modelo RDF, como mostra a Figura 23.

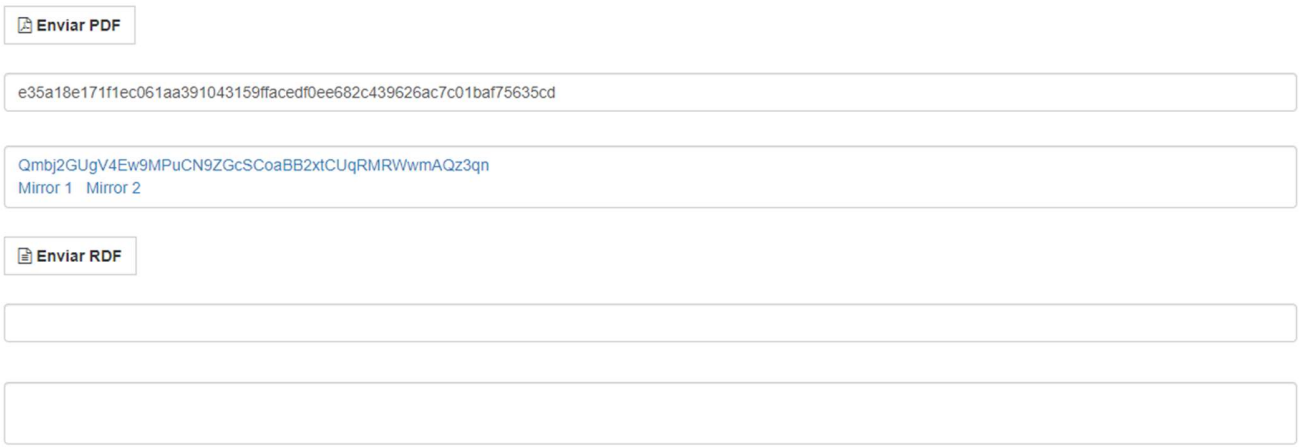

Figura 23 – Duas hashes obtidas após o envio do diploma em PDF à aplicação descentralizada (DApp).
Fonte: Elaboração própria (2020).

Desta forma, o usuário é capaz de enviar o documento PDF, e em seguida, elaborar o modelo RDF adequado, incluindo as *hashes* geradas para o PDF. Após, envia-se também o RDF para a geração das *hashes* correspondentes, como ilustra a Figura 24.

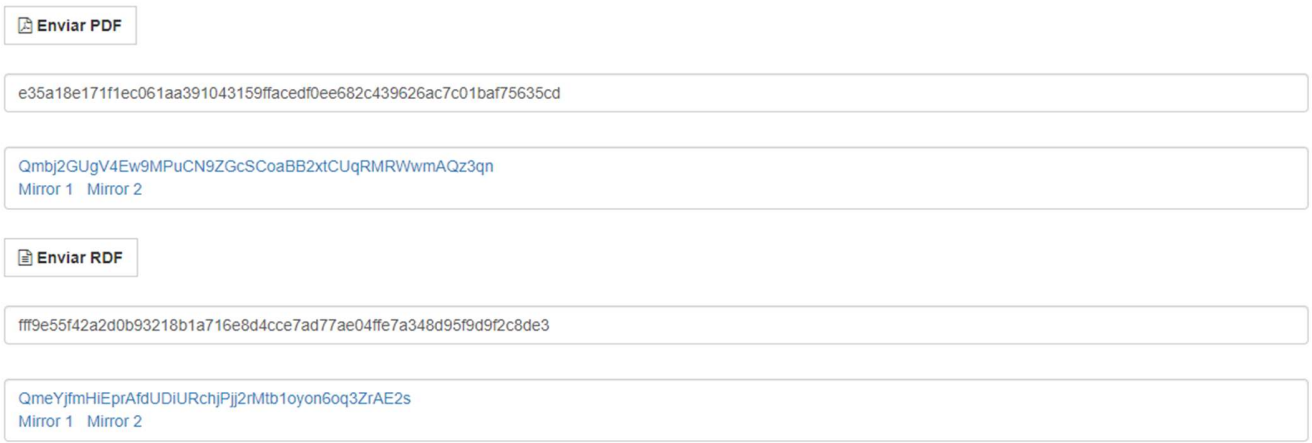

Figura 24 – Quatro hashes obtidas após o envio do diploma (PDF) e seus metadados (RDF).
Fonte: Elaboração própria (2020).

Obtidas as quatro *hashes,* o usuário, representando a instituição de ensino, deve registrá-las na *blockchain*. A aplicação então realiza a operação a partir da conta conectada à extensão *MetaMask*, invocando o contrato inteligente publicado na rede *Ethereum* selecionada (rede *Ropsten*, neste exemplo).

4.3.2 Obtenção/validação do documento digital

Após o registro na *blockchain*, os dados registrados, assim como a *hash* de transação gerada, podem ser observados na aplicação, de acordo com a Figura 25.

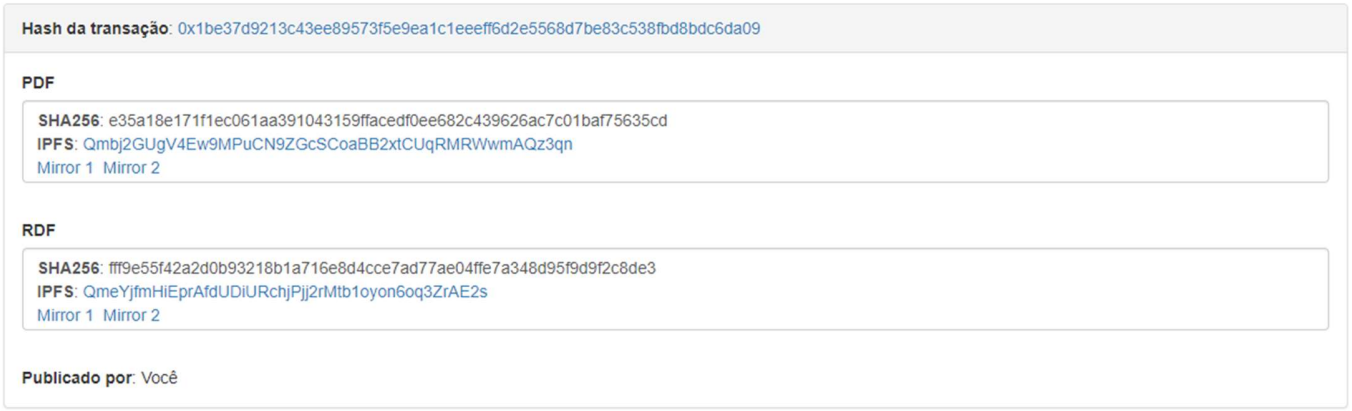

Figura 25 – Exibição da última transação realizada através da aplicação (DApp).
Fonte: Elaboração própria (2020).

É possível acessar os arquivos PDF e RDF através dos *links* disponibilizados, referentes ao sistema de arquivos IPFS. Reitera-se que os dados acima são fornecidos pela *blockchain*, garantindo sua veracidade.

Adicionalmente, pode-se acessar a transação através do *link* na *hash* da transação, diretamente no *explorer etherscan.io* (<*https://ropsten.etherscan.io/tx/0x1be37d9213c43ee89573f5e9ea1c1eeeff6d2e5568d7be83c538fbd8bdc6da09*>). O *explorer* disponibiliza diversas informações relacionadas à transação, como mostra a Figura 26.

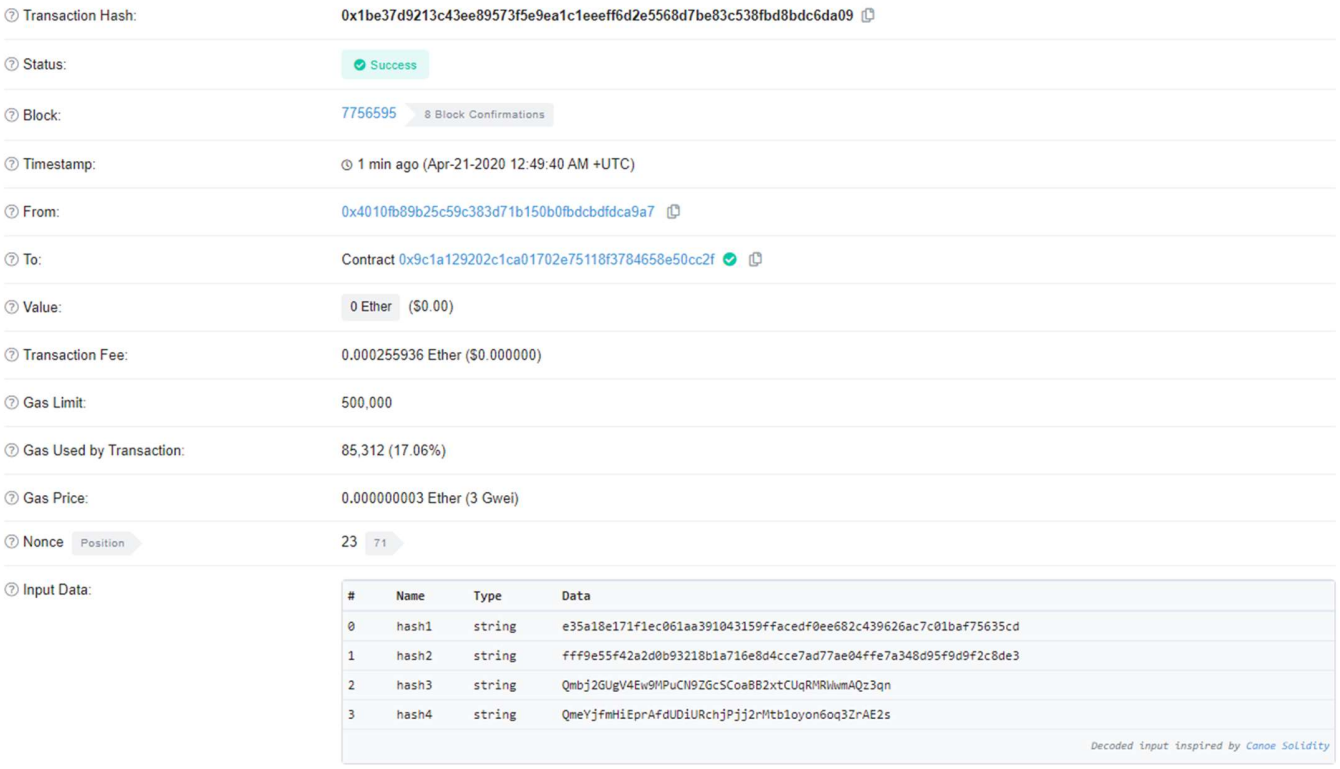

Figura 26 – Detalhes da transação referente ao registro do diploma na blockchain.
Fonte: Blockchain Explorer Etherscan (ETHERSCAN.IO, 2020).

Além da *hash* de transação (*Transaction Hash*), é possível observar o registro temporal (*Timestamp*), o endereço/chave pública do remetente (*from*), o endereço do contrato pelo qual as *hashes* foram registradas (*To*), os custos da transação em *ether* (*Transaction Fee*), o consumo em *gas* (*Gas Used By Transaction*), e as quatro *hashes* registradas (*Input Data*).

Vale ressaltar que, assim como uma conta bancária ou um cartão de crédito, o endereço/chave pública que realizou a transação (*0x4010fb89b25c59c383d71b150b0fbdcbdfdca9a7*) é a prova de que uma pessoa ou organização (isto é, quem se identifica pelo endereço determinado) reconhece aquele documento digital, uma vez que tenha publicado suas *hashes*. Para isso, basta que esse endereço seja fornecido ou reconhecido pela pessoa ou organização.

5. RESULTADOS E DISCUSSÕES

O presente trabalho aborda uma arquitetura para autenticação e disponibilização de documentos via *blockchain* e armazenamento associativo P2P endereçável ao conteúdo, a qual foi corroborada por meio de uma aplicação descentralizada (*DApp*), em *blockchains* de teste privadas e públicas. O intuito deste projeto é a redução de custos, burocracia e fraudes relacionados a processos de autenticação de documentos.

A aplicação descentralizada (*DApp*) foi desenvolvida de modo a integrar todos os requisitos necessários à autenticação de um documento, de acordo com a arquitetura proposta (vide Figura 5). Ao enviar documentos à *DApp*, estes são submetidos a funções SHA-256, e paralelamente, são publicados no sistema de arquivos P2P IPFS, gerando assim, as *hashes* de autenticação SHA-256 e de endereçamento IPFS, respectivamente. Em seguida, as *hashes* são registradas via contrato inteligente implementado na *blockchain Ethereum;* o contrato deve ser registrado em cada rede da *blockchain Ethereum* que se queira usar (neste trabalho, redes de teste *Ropsten* e *Rinkeby*). Reforça-se a importância da biblioteca *JavaScript web3.js*, injetada pela extensão *MetaMask*, que realiza a vinculação da *DApp* à *blockchain Ethereum*.

Efetuado o registro*,* infere-se a autenticação do documento ao acessar a transação na *blockchain* (vide Figura 6), necessitando apenas da *hash* da transação gerada no processo anterior. Após acessá-la, basta extrair as *hashes* IPFS e obter os documentos. Desta forma, entende-se que os documentos foram obtidos através de um registro (transação) que possui uma assinatura digital (endereço/chave pública), confirmando o reconhecimento do seu portador perante estes documentos (serviço notarial denominado "reconhecimento de firma"). Adicionalmente, ressalta-se que o processo de consulta aos registros na *blockchain*, e consequentemente extração das *hashes*, é gratuito, logo, qualquer usuário, em posse da *hash* da transação, é capaz de obter uma cópia autenticada dos documentos ali registrados, livre de custos.

Para contemplar os casos nos quais o documento é obtido de outras fontes diferentes do IPFS (por exemplo, por e-mail), considerou-se prudente o registro adicional de *hashes* de autenticação SHA-256. Como descrito na Figura 7, de posse do documento a ser verificado, deve-se aplicar função SHA-256 no documento, e obter sua *hash* SHA-256. Em paralelo, o usuário precisa acessar a transação na *blockchain* através de sua *hash*, e extrair a *hash* SHA-

256 que foi reconhecida pela parte autora do registro. Por fim, basta comparar a *hash* SHA-256 do documento com a *hash* SHA-256 registrada na *blockchain*. Se forem idênticos, significa que se trata de uma cópia autêntica do documento.

A prototipagem da aplicação em um ambiente controlado (privado), a *blockchain* local *Ganache,* permitiu maior eficiência na realização dos testes necessários ao seu desenvolvimento, garantindo mais robustez à versão atual da aplicação, esta sim tendo seu comportamento observado em *blockchains* públicas de teste, bastante similares a rede principal da *blockchain Ethereum* (*Main Ethereum Network*).

Acerca do objetivo geral deste projeto, é possível discorrer sobre importantes benefícios da arquitetura, a saber:

- Redução de custos: o cálculo dos custos relacionados às transações (registros) realizadas é feita através do consumo de *gas*. O site *ethgasstation*[20] permite calcular a conversão do consumo em *gas* para dólares americanos em tempo real. Este cálculo depende do preço do *gas* em *ether* (atualmente 3 *Gwei*); uma relação de oferta e procura no processo de mineração dentro da *blockchain Ethereum*. No extrato de registro do exemplo da Figura 19, o consumo em *gas* (*gas used*) foi de 114.880, equivalente, atualmente, a aproximadamente US$ 0,06, ou R$ 0,32 (32 centavos). Quanto ao exemplo da Figura 26, o consumo em *gas* foi de 85.312, equivalente a aproximadamente US$ 0,044, ou R$ 0,23 (23 centavos). Observa-se a drástica redução de custos, em comparação ao processo de reconhecimento de firma tradicional fornecido pelos cartórios brasileiros. Além disso, reitera-se que o documento pode ser obtido gratuitamente através do IPFS, criptograficamente reconhecido (autenticado) pela sua *hash*. Em um serviço notarial, este processo possui custos (obviamente), e é denominado “cópia autenticada”. Segundo a portaria CGJ Nº 2881/2019, nos cartórios do estado do Rio de Janeiro, por exemplo, são cobrados os valores de R$ 7,90 por um serviço de reconhecimento de firma

[20] ethgasstation.info

por semelhança, e R$ 8,16 por uma cópia autenticada, isto é, autenticação por documento ou página (GARCEZ, 2019).

- Redução de burocracia: qualquer usuário que possua uma carteira na *blockchain Ethereum,* com saldo na conta ativa para pagar as taxas transacionais, um documento digital e acesso à Internet, é capaz de autenticar/reconhecer documentos digitais através da aplicação, implementação da arquitetura apresentada. Na outra mão, basta a qualquer usuário que necessite verificar/validar a autenticidade de um documento digital, a *hash* da transação. Assim, o usuário pode conferir o endereço/chave pública do autor do registro. Ademais, os documentos digitais gerados a partir da *hash* IPFS são absolutamente equivalentes; não só zeram custos de uma cópia autenticada, como dito anteriormente, como podem ser obtidos de forma eficiente e resiliente a partir do sistema de arquivos P2P IPFS;
- Redução de fraudes: a estrutura da tecnologia *blockchain* garante a incorruptibilidade das transações e dos dados registrados em seus blocos. Além disso, uma vez armazenados na *blockchain*, os dados das transações não podem ser alterados, nem removidos. Estas características asseguram a auditabilidade das informações registradas nesta base de dados distribuída e descentralizada. Na arquitetura proposta, a utilização de *hashes* garante a imutabilidade dos documentos digitais, uma vez que qualquer alteração origina uma nova *hash* completamente diferente, impedindo a validação de documentos digitais fraudulentos. No que tange à *blockchain*, em uma transação de registro de *hashes*, todas as informações são permanentes, como o endereço/chave pública da parte autora e as *hashes* publicadas, tornando a arquitetura uma ótima oportunidade de combate às fraudes. Todavia, insta salientar a preferência semântica de "redução", e não "eliminação" de fraudes; isto porque, ainda que a arquitetura e as tecnologias sejam sistematicamente incorruptíveis, ainda há o fator humano. Caso uma parte mal-intencionada tenha acesso à carteira da parte autora, seria possível a autenticação indevida de documentos digitais. Assim, cabe ao usuário proteger seus dados pessoais de acesso à *blockchain*.

No estudo de caso apresentado (seção 4.3), no qual houve a autenticação de um diploma não real, mas realista, destaca-se a importância da autenticação complementar de sua representação formal em RDF, a fim de realizar futuras inferências e consultas complexas através de uma aplicação semântica. Ademais, faz-se necessária a melhor compreensão acerca do endereço/chave pública da parte autora do registro, pois é esta *hash* – *0x4010fb89b25c59c383d71b150b0fbdcbdfdca9a7* – que permite detectar a autoria da transação, e logo, sua responsabilização legal. Neste exemplo, faria sentido se a *hash* supramencionada pertencesse à *Maria Silva*, tendo em vista que é a responsável pela autenticação do diploma. Nesta hipótese, a *hash* é a comprovação legal de sua responsabilidade, devendo a autora responder legalmente, como por exemplo, em uma posterior comprovação de que o aluno não cumpriu algum dos requisitos necessários à obtenção do diploma, e estas condições adversas devem ser consideradas no desenvolvimento real de um contrato inteligente.

A arquitetura proposta contempla, claramente, a autenticação de documentos digitais e pode ser aplicada a qualquer espécie de documento em que seja necessária a confirmação de sua autenticidade, como: RG, CPF, certidão de nascimento, diploma, contrato de prestação de serviço, boletim de ocorrência, etc. Todavia, documentos de maior complexidade, como contratos, ou o diploma apresentado, ensejam a proposição de novos requisitos específicos, considerando os diversos cenários possíveis, de acordo com as necessidades percebidas pela pessoa ou organização que deseja autenticá-los. Ao utilizar o exemplo anterior, em que ocorre a autenticação de um diploma inválido, tendo em vista o não cumprimento dos requisitos à obtenção do diploma, entende-se que deva haver a revogação do diploma, através de funções oriundas de um contrato inteligente específico para diplomas de instituições de ensino. Assim, estima-se que cada contexto documental apresente seu próprio contrato inteligente, observando suas particularidades, tanto no processo de autenticação, quanto na tratativa de circunstâncias excepcionais.

6. TRABALHOS RELACIONADOS

Visando a redução da falsificação de certificados de graduação, Cheng *et al.* (2018) propõem um sistema de certificação digital baseado na tecnologia *blockchain*. Inicialmente, um certificado digital é gerado a partir de seu respectivo certificado físico; sua *hash* é calculada e armazenada na *blockchain Ethereum* subjacente ao sistema. Em seguida, o sistema cria um *QR-code* e uma *string* para consulta, que são anexados no certificado físico. Isso possilibitará a usuários a verificação de autenticidade do certificado físico junto ao sistema, através de escaneamento pelo celular ou consulta ao site, respectivamente.

Com a premissa de que o volume de dados na *blockchain* cresce continuamente e que a esta alta demanda por espaço de armazenamento impede que novos nós façam parte da rede, Zheng *et al.* (2018) sugerem um modelo de armazenamento de dados para *blockchain* baseado em IPFS. No esquema aplicado à *blockchain Bitcoin*, os mineradores depositam os dados da transação no IPFS, que então tem sua *hash* gerada e retornada ao bloco, reduzindo drasticamente a quantidade de dados armazenados na *blockchain*.

Nizamuddin, Hasan e Salah (2018) propõem uma solução para publicar e autenticar conteúdos digitais online, como livros, músicas e filmes, através do IPFS e contratos inteligentes pela *blockchain*. Enquanto o IPFS é utilizado para armazenar os conteúdos digitais com alta integridade e disponibilidade, os contratos inteligentes da rede *Ethereum* têm por finalidade gerenciar e fornecer rastreabilidade e transparência. Ressalta-se ainda neste projeto, o acesso irrestrito ao código completo do contrato inteligente desenvolvido e a utilização da ferramenta Remix IDE para a realização de testes com suas funções.

Cano-Benito, Cimmino e García-Castro (2019) fornecem uma visão aprofundada da benéfica relação que a Web Semântica e a *blockchain* podem alcançar juntas e relatam diferentes cenários que identificaram na literatura para combiná-las. Os autores apontam benefícios que a tecnologia *blockchain* pode extrair da Web Semântica, como a modelagem de dados em padrões conhecidos, ligação de dados, múltiplos modelos de dados e consultas à *blockchain*; por outro lado, a Web Semântica também pode obter benefícios da tecnologia *blockchain*, como a descentralização, imutabilidade, transparência e publicidade dos dados.

A respeito dos dados abertos ligados (*Linked Open Data* - LOD), Sicilia *et al.* (2016) relatam que sua natureza descentralizadora consiste na nuvem de serviços locais, resultando em problemas de disponibilidade e *links* quebrados. Desta forma, propõem uma abordagem baseada em sistema de arquivos P2P, a fim de alcançar melhor disponibilidade, eficiência e resiliência de dados, enquanto preserva os princípios LOD. O artigo traz um prótotipo baseado no sistema de arquivos distribuído IPFS, de modo que os dados abertos ligados sejam persistentes e imutáveis, independente de quem os publicou originalmente.

Os trabalhos correlatos corroboram a relevância da tecnologia *blockchain* no que tange à lisura de informações e à autenticação de documentos. Todavia, atentam sobre sua debilidade ao atuar com grandes quantidades de dados e reforçam quanto a necessidade de complementá-la com um sistema descentralizado capaz de armazenar arquivos em larga escala, um sistema de arquivos distribuído. Além disso, há o destaque destas tecnologias quanto às suas possíveis contribuições à Web Semântica e ao modelo de dados RDF.

Os trabalhos apresentados neste capítulo se relacionam com este projeto de dissertação, especificamente, da seguinte forma:

- Cheng *et al.* (2018): apesar da validação do diploma, mesmo exemplo utilizado nesta dissertação, os autores restringem seu foco à autenticação do diploma físico, não disponibilizando sua versão digital. Diferente deste trabalho, a arquitetura proposta nesta dissertação não só valida, como permite obter o documento digital de forma prática e ilimitada, como uma "cópia autenticada" do documento validado, além de publicar e validar, adicionalmente, seus metadados em RDF.
- Zheng *et al.* (2018): assim como no presente trabalho, os autores sugerem a publicação de dados no IPFS, e o registro das *hashes* geradas na *blockchain*. No entanto, o projeto dos autores utilizou a *blockchain Bitcoin*, enquanto o esta dissertação utiliza a *blockchain Ethereum*. Atualmente, a *blockchain Ethereum* apresenta melhor estrutura para o gerenciamento de informações do que a rede *Bitcoin*, devido aos contratos inteligentes, que permitem rodar códigos auto-executáveis, um verdadeiro computador de *Turing* mundial.

- Nizamuddin, Hasan e Salah (2018): apesar de apresentarem um objetivo diferente, isto é, a autenticação de mídias digitais, os autores apresentam grande proximidade metodológica com a presente dissertação. A publicação e disponibilização através do IPFS, o registro de suas *hashes* e o gerenciamento dos dados por meio de contratos inteligentes e a rede *Ethereum*. Contudo, enquanto os autores realizaram os testes através da ferramenta Remix IDE, a presente dissertação desenvolve uma aplicação descentralizada como prova de conceito à arquitetura proposta, além de analisar todos os resultados obtidos.
- Cano-Benito, Cimmino e García-Castro (2019): o trabalho fornecido pelos autores evidencia que há sim benefícios oferecidos ao aliar a Web Semântica à tecnologia *blockchain* e vice-versa. Desta forma, devido ao foco desta dissertação, destaca-se a contribuição da *blockchain Ethereum* para a Web Semântica, ao autenticar arquivos RDF, garantindo a descentralização, imutabilidade, transparência e publicidade dos dados autenticados.
- Sicilia *et al.* (2016): ao reiterar a sinergia entre dados abertos ligados e a disponibilização de dados, os autores verificam a necessidade de se utilizar um sistema de arquivos distribuído mais resiliente, o que exalta o trabalho de publicação e disponibilização de arquivos RDF no sistema IPFS, feito nesta dissertação.

Em comparação aos trabalhos relacionados, este trabalho, se diferencia, sobretudo, por fornecer uma arquitetura genérica, auto-contida para a autenticação e disponibilização sistemática de documentos, usando dois dos mais importantes paradigmas disruptivos da atualidade, que são *blockchain* e sistema de arquivos distribuído P2P endereçável ao conteúdo. Uma arquitetura agnóstica com relação a detalhes de implemementação e formato de documento, podendo ser aplicada em quaisquer implementações destes paradigmas. Outro ponto importante é que todos os três módulos da arquitetura proposta foram, plenamente, corroborados por meio de uma aplicação descentralizada (DApp) completa (*frontend* e *backend*), usando tanto *blockchain*s *Ethereum* locais quanto públicas, atestando sua escalabilidade.

7. CONSIDERAÇÕES FINAIS

Este projeto apresentou uma arquitetura genérica para autenticação e disponibilização de documentos, usando a tecnologia *blockchain* e sistema de armazenamento associativo P2P endereçável ao conteúdo, respectivamente, e teve por motivação acabar com a necessidade de uma terceira parte confiável para a validação de documentos, como cartórios, e assim, reduzir burocracia, fraudes e custos correlatos.

A arquitetura foi implementada por uma aplicação descentralizada (*DApp*), na qual foi realizada a autenticação por meio da rede *Ethereum*, que forneceu as funcionalidades dos contratos inteligentes à favor da aplicação, permitindo ótimo desempenho conceitual e prático dos registros.

A funcionalidade de disponibilização foi realizada por meio do sistema de arquivos P2P IPFS, que permite a publicação imutável e resiliente de documentos.

Ressaltam-se ainda as principais ferramentas utilizadas, *Truffle* e *Ganache*, que têm grande notoriedade para a implementação e realização de testes com aplicações descentralizadas na rede *Ethereum*, atualmente. Desta forma, foi possível testar o contrato inteligente elaborado e a aplicação descentralizada desenvolvida, em um ambiente simulado localmente. Em seguida, a aplicação também foi testada em ambientes de teste oficiais da rede *Ethereum* (*Rinkeby* e *Ropsten*), a fim de se analisar seu comportamento em ambientes descentralizados (*blockchains* públicas).

Além disso, destacam-se as implementações e os aperfeiçoamentos desenvolvidos junto à *DApp,* visando a integração da geração de *hashes* (antes feita com ferramentas externas) com a gestão de informações à cerca do contrato inteligente e da *blockchain Ethereum*, conferindo autossuficiência à aplicação descentralizada em prol do escopo estabelecido para o projeto: a disponibilização e a validação de documentos.

Por fim, foi descrito um estudo de caso realista de autenticação e disponibilização de um documento digital, diploma de mestrado, com o propósito de corroborar todo o trabalho desenvolvido: da proposição da arquitetura à sua implementação pela *DApp.*

7.1 Trabalhos futuros

A tecnologia *blockchain* e a descentralização vêm apresentando exponencial evolução ao longo dos anos, e por este motivo, haverá constantemente melhoramentos a serem realizados neste âmbito. Atualmente, existem aprimoramentos reconhecidamente possíveis a serem desenvolvidos, a saber:

- Ainda que informações relacionadas às transações, endereços públicos e dados de entrada possam ser identificados na *blockchain*, é reconhecida a necessidade de um ambiente mais amigável à luz da perspectiva do usuário, para acessar, processar e analisar dados da *blockchain* (PRATAMA; MUTIJARSA, 2018). Por esta razão, foram incluídas ao escopo da aplicação a autenticação e disponibilização de documentos no formato RDF. Assim, estima-se mapeamentos em RDF para diferentes tipos de documentos a serem disponiblizados e validados. Esta incorporação do modelo prevê futuras aplicações semânticas em torno dos metadados RDF, visando estabelecer maior interoperabilidade de dados, para benefício de desenvolvedores e usuários finais;

- Atualmente, a aplicação é majoritariamente descentralizada: possui seu *backend* (contrato inteligente) registrado na *blockchain* e armazena dados em um sistema de arquivos P2P; todavia, seu *frontend* se encontra em um servidor centralizado. Projeta-se, para breve, a descentralização da interface do usuário, para alcançar uma aplicação totalmente descentralizada;

- Embora a arquitetura garanta a autenticação de documentos, e a aplicação provou isso, existem cenários adversos pós-autenticação que devem ser abordados oportunamente, como a revogação de um diploma por má conduta profissional, ou a rescisão de um contrato de prestação de serviços, por exemplo. Pensando nisso, a utilização de variáveis de estado e funções generalizadas para o desenvolvimento do contrato inteligente permitem sua extensão, a fim de fornecer novas funcionalidades e otimizar as já existentes, se pertinentes forem.

- Finalmente, visa-se a migração da aplicação, de ambientes de teste realistas, para a rede principal *Ethereum*, na qual os documentos serão realmente validados e disponibilizados, fidedignamente. Neste caso, obviamente, os custo em *ether* envolvidos serão reais.

REFERÊNCIAS

ANTONOPOULOS, A. M.; WOOD, G. **Mastering Ethereum**. First ed. United States of America: O'Reilly Media, Inc., 2018.

AZEVEDO, R.; JACYNTHO, M. UM MODELO BASEADO EM ONTOLOGIAS LINKED DATA PARA CATALOGAÇÃO DE PROJETOS DE SOFTWARE. **Conferências Ibero-Americanas WWW/Internet e Computação Aplicada**, 2014.

BENET, J. **IPFS - Content Addressed, Versioned, P2P File System**, 2019. Disponível em: <https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf>. Acesso em: 12 dez. 2019

BLOMER, J. A Survey on Distributed File System Technology. **Journal of Physics: Conference Series**, v. 608, p. 012039, 22 maio 2015.

BRICKLEY, D.; MILLER, L. **FOAF Vocabulary Specification**. Disponível em: <http://xmlns.com/foaf/spec/>. Acesso em: 8 mar. 2019.

BUTERIN, V. **A Next-Generation Smart Contract and Decentralized Application Platform**. Disponível em: <https://github.com/ethereum/wiki>. Acesso em: 18 fev. 2019.

CANO-BENITO, J.; CIMMINO, A.; GARCÍA-CASTRO, R. Towards Blockchain and Semantic Web. In: **Business Information Systems Workshops**. Lecture Notes in Business Information Processing. Cham: Springer, 2019. v. 373p. 220–231.

CHENG, J.-C. et al. **Blockchain and smart contract for digital certificate**. 2018 IEEE International Conference on Applied System Invention (ICASI). **Anais**... In: 2018 IEEE INTERNATIONAL CONFERENCE ON APPLIED SYSTEM INNOVATION (ICASI). Chiba: IEEE, abr. 2018Disponível em: <https://ieeexplore.ieee.org/document/8394455/>. Acesso em: 20 dez. 2018

CHRISTIDIS, K.; DEVETSIKIOTIS, M. Blockchains and Smart Contracts for the Internet of Things. **IEEE Access**, v. 4, p. 2292–2303, 2016.

CNJ. **Cartórios: mais de R$ 15 bilhões arrecadados com serviços em 2017 - Portal CNJ**. Disponível em: <http://www.cnj.jus.br/noticias/cnj/86608-cartorios-mais-de-r-15-bilhoes-arrecadados-com-servicos-em-2018>. Acesso em: 14 ago. 2019.

Contracts — Solidity 0.5.11 documentation. Disponível em: <https://solidity.readthedocs.io/en/v0.5.11/contracts.html#events>. Acesso em: 21 ago. 2019.

CYBROSYS. **BLOCKCHAIN.** Cybrosys Limited Edition, 2018.

DAMERON, M. **Beigepaper: An Ethereum Technical Specification**. . Acesso em: 17 fev. 2019.

Developers | Ethereum. Disponível em: <https://www.ethereum.org/developers/#testnets-and-faucets>. Acesso em: 28 ago. 2019.

ETHEREUM COMMUNITY. **Ethereum Homestead 0.1 documentation**. Disponível em: <http://ethdocs.org/en/latest/index.html>. Acesso em: 11 fev. 2019.

ETHEREUM FOUNDATION. **Solidity, the Contract-Oriented Programming Language: ethereum/solidity**. Disponível em: <https://github.com/ethereum/solidity>. Acesso em: 19 fev. 2019.

ETHERSCAN.IO. **Ropsten Transaction Hash (Txhash) Details | Etherscan**. 2019. Disponível em: <http://ropsten.etherscan.io/tx/0x0a3b58ac8898a19291f0f3a559d15e37b48784a43567afa3b2732914c5059e82>. Acesso em: 28 maio. 2020.

ETHERSCAN.IO. **Ropsten Transaction Hash (Txhash) Details | Etherscan**. 2020. Disponível em: <http://ropsten.etherscan.io/tx/0x1be37d9213c43ee89573f5e9ea1c1eeeff6d2e5568d7be83c538fbd8bdc6da09>. Acesso em: 28 maio. 2020.

FERREIRA, J. A.; SANTOS, P. O MODELO DE DADOS RESOURCE DESCRIPTION FRAMEWORK (RDF) E O SEU PAPEL NA DESCRIÇÃO DE RECURSOS. **Informação & Sociedade: Estudos**, p. 11, 2013.

Ganache documentation. Disponível em: <https://trufflesuite.com/docs/ganache/overvi>. Acesso em: 25 fev. 2019.

GARCEZ, B. **PORTARIA CGJ Nº 2881/2019.** Corregedoria Geral de Justiça, 26 dez. 2019. Disponível em: <http://www1.tjrj.jus.br/gedcacheweb/default.aspx?GEDID=00030D72858A2737F2254CABC62E8076AC00F7C4081F3A08>. Acesso em: 20 jun. 2020

GUPTA, M. **Blockchain for dummies**. 2nd IBM Limited Edition ed. 111 River St. Hoboken: John Wiley & Sons, Inc., 2018.

HERMAN, I. et al. **RDFa 1.1 Primer - Third Edition**. Disponível em: <https://www.w3.org/TR/rdfa-primer/>. Acesso em: 31 out. 2019.

IANSITI, M.; LAKHANI, K. R. The Truth About Blockchain. **Harvard Business Review**, p. 11, 2017.

Introduction to Smart Contracts — Solidity 0.5.15 documentation. Disponível em: <https://solidity.readthedocs.io/en/v0.5.15/introduction-to-smart-contracts.html>. Acesso em: 17 dez. 2019.

JACYNTHO, M. **UM MODELO DE BLOQUEIO MULTIGRANULAR PARA RDF**. DOUTOR EM CIÊNCIAS EM INFORMÁTICA—Rio de Janeiro, Brazil: PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO, 2 fev. 2012.

JACYNTHO, M. D.; SCHWABE, D. A multigranularity locking model for RDF. **Journal of Web Semantics**, v. 39, p. 25–46, ago. 2016.

KUMAVIS et al. **MetaMask**. Disponível em: <https://metamask.io/>. Acesso em: 12 mar. 2019.

MANOLA, F.; MILLER, E.; MCBRIDE, B. **RDF 1.1 Primer**. Disponível em: <https://www.w3.org/TR/rdf11-primer/>. Acesso em: 25 mar. 2019.

MARR, B. **What Is The Difference Between Bitcoin and Ethereum?** Disponível em: <https://www.forbes.com/sites/bernardmarr/2018/02/05/what-is-the-difference-between-bitcoin-and-ethereum/>. Acesso em: 19 fev. 2019.

NIZAMUDDIN, N.; HASAN, H. R.; SALAH, K. IPFS-Blockchain-Based Authenticity of Online Publications. In: CHEN, S.; WANG, H.; ZHANG, L.-J. (Eds.). . **Blockchain – ICBC 2018**. Cham: Springer International Publishing, 2018. v. 10974p. 199–212.

PAPA, J. **Lightweight node server. Contribute to johnpapa/lite-server development by creating an account on GitHub**. [s.l: s.n.].

PRATAMA, F. A.; MUTIJARSA, K. **Query Support for Data Processing and Analysis on Ethereum Blockchain**. 2018 International Symposium on Electronics and Smart Devices (ISESD). **Anais**... In: 2018 INTERNATIONAL SYMPOSIUM ON ELECTRONICS AND SMART DEVICES (ISESD). Bandung: IEEE, out. 2018Disponível em: <https://ieeexplore.ieee.org/document/8605476/>. Acesso em: 12 mar. 2019

PROTOCOL LABS. **IPFS is the Distributed Web**. Disponível em: <https://ipfs.io/>. Acesso em: 13 fev. 2019a.

PROTOCOL LABS. **Hashes – IPFS Documentation**. Disponível em: <https://docs.ipfs.io/guides/concepts/hashes/>. Acesso em: 5 mar. 2019b.

Remix documentation. Disponível em: <https://remix.readthedocs.io/en/latest/>. Acesso em: 5 mar. 2019.

ROSA, A. M. **Qual a validade jurídica dos documentos pela rede blockchain?** Disponível em: <https://www.conjur.com.br/2019-jan-11/limite-penal-qual-validade-juridica-documentos-rede-blockchain>. Acesso em: 17 dez. 2019.

SICILIA, M.-A.; SÁNCHEZ-ALONSO, S.; GARCÍA-BARRIOCANAL, E. Sharing Linked Open Data over Peer-to-Peer Distributed File Systems: The Case of IPFS. In: GAROUFALLOU, E. et al. (Eds.). . **Metadata and Semantics Research**. Cham: Springer International Publishing, 2016. v. 672p. 3–14.

Solidity Documentation. Disponível em: <https://solidity.readthedocs.io/en/v0.5.5/>. Acesso em: 20 fev. 2019.

SZABO, N. **Smart Contracts**. Disponível em: <http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html>. Acesso em: 9 fev. 2019.

TAN, M. et al. **Ethereum (ETH) Blockchain Explorer**. Disponível em: <https://etherscan.io/>. Acesso em: 9 mar. 2019.

Truffle documentation. Disponível em: <https://trufflesuite.com/docs/truffle/overvi>. Acesso em: 25 fev. 2019.

WANG, S.; ZHANG, Y.; ZHANG, Y. A Blockchain-Based Framework for Data Sharing With Fine-Grained Access Control in Decentralized Storage Systems. **IEEE Access**, v. 6, p. 38437–38450, 2018.

WOOD, G. **ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER**. Disponível em: <https://ethereum.github.io/yellowpaper/paper.pdf>. Acesso em: 17 fev. 2019.

YLI-HUUMO, J. et al. Where Is Current Research on Blockchain Technology?—A Systematic Review. **PLOS ONE**, v. 11, n. 10, p. e0163477, 3 out. 2016.

ZHANG, S.; LEE, J.-H. Analysis of the main consensus protocols of blockchain. **ICT Express**, p. S240595951930164X, ago. 2019.

ZHENG, Q. et al. **An Innovative IPFS-Based Storage Model for Blockchain**. 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI). **Anais**... In: 2018 IEEE/WIC/ACM INTERNATIONAL CONFERENCE ON WEB INTELLIGENCE (WI). Santiago: IEEE, dez. 2018. Disponível em: <https://ieeexplore.ieee.org/document/8609675/>. Acesso em: 11 mar. 2019