UNIVERSIDADE FEDERAL DE SANTA CATARINA

HENRIQUE HERMES SCHMITT
PEDRO PAULO CHIARELLI STEIL

PRESERVAÇÃO DE DADOS EM BLOCKCHAIN

FLORIANÓPOLIS
2021/2

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

PRESERVAÇÃO DE DADOS EM BLOCKCHAIN

HENRIQUE HERMES SCHMITT
PEDRO PAULO CHIARELLI STEIL

Trabalho de Conclusão de Curso de Graduação em Sistemas de informação, do Departamento de Informática e Estatística, do Centro Tecnológico da Universidade Federal de Santa Catarina, requisito parcial à obtenção do título de Bacharel em Sistemas de informação.

Orientadora: Dra. Carla Merkle Westphall
Coorientador: Me. Leandro Loffi

FLORIANÓPOLIS
2021/2

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

PRESERVAÇÃO DE DADOS EM BLOCKCHAIN

HENRIQUE HERMES SCHMITT
PEDRO PAULO CHIARELLI STEIL

Trabalho de Conclusão de Curso de Graduação em Sistemas de informação, do Departamento de Informática e Estatística, do Centro Tecnológico da Universidade Federal de Santa Catarina, requisito parcial à obtenção do título de Bacharel em Sistemas de informação.

Orientadora: Dra. Carla Merkle Westphall
Coorientador: Me. Leandro Loffi
Banca examinadora:
Me. Caciano dos Santos Machado
Me. Johann Westphall

FLORIANÓPOLIS
2021/2

RESUMO

A evolução da sociedade em conjunto com a evolução tecnológica propicia uma enorme carga de dados gerados a cada dia. Por conta disso, é necessário garantirmos que os três pilares da segurança da informação estejam sempre presentes, sendo eles a confidencialidade, integridade e disponibilidade (CID). Garantir o CID é um grande desafio para todos os envolvidos, e, quando feito apenas em seu ambiente tecnológico local, o usuário pode ficar refém de problemas de disco, o que poderá causar a perda parcial ou total dos dados. Se feito em conexão com um outro usuário, no qual ambos trocam informações via um canal seguro, pode estar sujeito a ataques de rede, como por exemplo o *man-in-the-middle*, o qual teria acesso aos dados, podendo até modificá-los, violando assim até dois dos três pilares da segurança da informação. Tendo isso em vista, uma alternativa é a aplicação distribuída que é mais segura para esses pontos supracitados. Um exemplo de aplicação distribuída é a *blockchain*. Uma rede *blockchain* pode ser uma alternativa viável, pois mesmo ocorrendo uma mudança no hash de um de seus blocos, ela será avaliada pelos outros participantes da rede. Outra característica que garante maior segurança à *blockchain* é a ausência de uma autoridade central de gerenciamento. Sem uma autoridade central, todos os participantes da *blockchain* possuem uma cópia exata dela e, se um bloco for alterado, o mesmo teria que ser verificado por todos os outros participantes da *blockchain*, os quais iriam invalidá-lo. Além disso, na solução com utilização da *blockchain* também são utilizados *Smart Contracts*, que são programas de computador imutáveis que são armazenados na *blockchain*. Portanto, este trabalho tem como objetivo criar uma rede de *blockchain* para preservação de dados, com o intuito de mantê-los confidenciais, íntegros e disponíveis, respeitando, assim, os três pilares da segurança da informação. Essa rede contará com agentes de controle de acesso e visualização, um banco de dados para comportar os arquivos a serem guardados, e a *blockchain*.

Palavras-chave: *blockchain*, *Smart Contract*, preservação de dados, segurança

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES E SIGLAS

CID	Confidencialidade, integridade e disponibilidade
TTP	*Trusted Third Party*
MSP	*Membership Service Provider*

SUMÁRIO

1. INTRODUÇÃO

Não apenas para organizações como também para pessoas, uma das suas preocupações hoje é com a segurança de seus dados. Desde fotos de seus animais de estimação até dados sensíveis como fotos de documentos e dados financeiros a preocupação é a mesma, então, surgem questionamentos de como manter esses dados confidenciais, íntegros e disponíveis.

De modo geral, uma solução simples e direta que responde a pergunta seria fazer o uso de criptografia nos dados que interessam. Porém, com uma grande variedade de algoritmos, a escolha torna-se complexa. Nem todo algoritmo de criptografia é recomendado para tratar da integridade e segurança de dados. Na verdade são recomendados apenas algumas aplicações de hash, como autenticação de mensagem, assinatura digital e *blockchain*. (STALLINGS, 2014). Com o objetivo de garantir a preservação de dados e responder ao questionamento sobre os pilares da segurança da informação, das três possibilidades foi escolhido o desenvolvimento de uma solução que faça uso de *blockchain*, com a possibilidade de combinação com *Smart Contracts*.

Uma das soluções relacionadas à segurança é o uso de um *Trusted Third Party* (TTP). Essa solução apresenta algumas falhas que podem ser críticas para a disponibilidade como, por exemplo, *single point of failure* (ALHARBI, VAN MOORSEL, 2017). Se há uma falha no TTP, nenhuma operação será feita em cima dos dados. Tendo isso em vista, o uso de *blockchain* combinado com *Smart Contracts* prevê esse tipo de falha, utilizando dos *Smart Contracts* como forma de reforçar a confiança entre os participantes. Por não ter uma autoridade central e muito menos um TTP, todos os participantes da *blockchain* fazem as verificações necessárias que um TTP faria e, por ser distribuída, não é possível ocorrer *single point of failure*.

A proposta deste trabalho consiste no desenvolvimento de uma rede de *blockchain* combinada com *Smart Contracts* para manter dados confidenciais, íntegros e disponíveis. Nesta rede serão possíveis duas ações, a de armazenar um dado e a de visualizar dados já armazenados. Estes dados armazenados na rede de *blockchain* devem a todo momento estar disponíveis, íntegros e confidenciais para o usuário.

1.1 Motivação

O rápido avanço da tecnologia atual criou diversas oportunidades para as pessoas e organizações dentro e fora da grande área de segurança da informação, porém acabam ficando algumas lacunas importantes que ainda não foram bem resolvidas. Uma dessas lacunas é em relação à preservar os dados de maneira que garanta a segurança e veracidade das informações. O presente trabalho tem como motivação a apresentação de um modelo de preservação de dados com o uso da tecnologia de *blockchain* que garanta os três pilares da segurança da informação, podendo ser usado em diferentes contextos, já que suas tecnologias não são restritas a nenhum tipo de atividade.

1.2 Objetivo Geral

Como objetivo geral deste trabalho temos o desenvolvimento de uma rede de *blockchain* em ambiente de computação em nuvem, tendo foco na preservação de dados, com finalidade de garantir os três pilares da segurança da informação - confidencialidade, integridade e disponibilidade. A rede de *blockchain* deve ser distribuída, e com participação de colaboradores conhecidos.

1.3 Objetivos Específicos

A seguir são listados os principais objetivos específicos deste trabalho.

1. Desenvolvimento de uma rede de *blockchain*:
    a. Será utilizada a plataforma Hyperledger Fabric no desenvolvimento da rede de *blockchain*;
    b. A rede terá foco na preservação de dados.
    c. A rede deve ter a possibilidade de ser hospedada em computação em nuvem ou *fog*.

1.4 Organização do trabalho

O trabalho está dividido em 6 capítulos. O primeiro capítulo descreve a introdução, motivação e objetivos deste trabalho. No capítulo 2 é feita a fundamentação teórica, trazendo conceitos que são fundamentais para a compreensão deste trabalho. No capítulo 3 é feito uma análise de trabalhos correlatos, nos quais foram apresentados análises de 4 artigos e uma revisão sistemática baseada em palavras chave. No capítulo 4 é feita a abordagem da proposta, que consta a arquitetura proposta com seus dois possíveis cenários de execução, um

de coleta e outro de visualização dos dados. No capítulo 5 é detalhado o desenvolvimento da solução proposta, apresentando detalhes técnicos contidos na arquitetura da solução. Já no capítulo 6 é feita a análise dos resultados alcançados e a conclusão do trabalho. Por fim, no capítulo 7 é comentado sobre trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Segurança da informação

Segundo STALLINGS, 2011, se entende como Segurança da Informação o conjunto de tecnologias e ferramentas com objetivo de manter as informações de uma organização ou indivíduo confidenciais, íntegras e disponíveis. Esses três conceitos são conhecidos como os pilares da segurança da informação, e são descritos como:

- Confidencialidade: Esse termo é composto por dois conceitos relacionados:

  Confidencialidade de dados: Garante que a privacidade ou confidencialidade de informações estejam disponíveis e acessíveis apenas para indivíduos devidamente autorizados.

  Privacidade: Garante que informações relacionadas a indivíduos possam ser acessadas e armazenadas somente por indivíduos e sistemas devidamente autorizados.
- Integridade: Esse termo é composto por dois conceitos relacionados:

  Integridade de dados: Garante que informações e sistemas possam ser modificados apenas de maneira autorizada.

  Integridade de sistema: Garante que um sistema execute sua função de uma maneira íntegra, livre de manipulação indesejada e não autorizada do sistema.
- Disponibilidade: Garante que o sistema funcione da maneira devida e que seu serviço não seja negado ou impossibilitado à usuários autorizados.

Os três conceitos incorporam os objetivos de segurança fundamentais para dados e para serviços de informação e informática.

2.2 Blockchain

Segundo Casino et al, 2019, uma *blockchain* é definida como uma estrutura de dados distribuídos que possui um carimbo do tempo, a qual permite apenas acréscimos a rede. Uma outra característica crucial da *blockchain* é a imutabilidade, uma vez que um *ledger* é adicionado, ele permanecerá inalterado, garantindo, assim, um preciso histórico de transações.

O funcionamento de uma *blockchain* pode ser definido pelo seguinte fluxo,

representado na Figura 1:

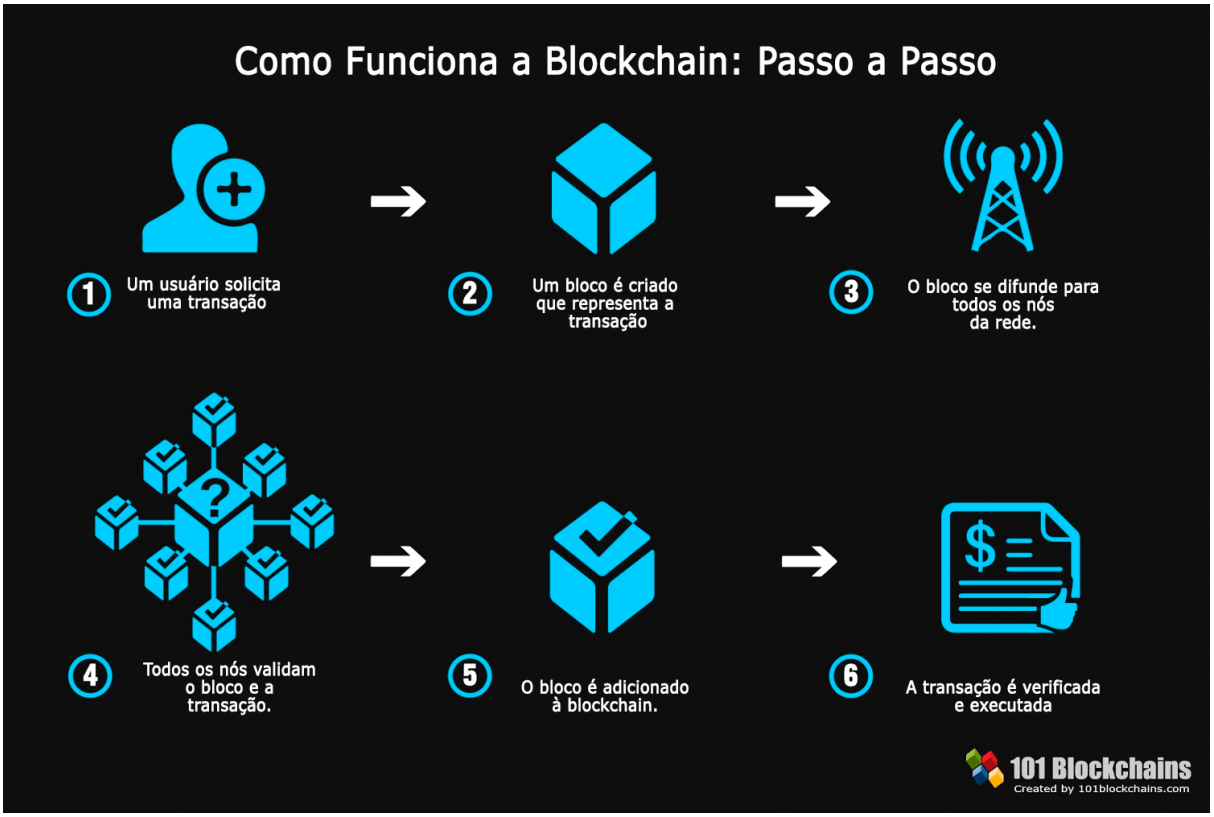

Figura 1: Funcionamento de *blockchain*

Disponível em: https://101blockchains.com/pt/tecnologia-blockchain-guia/

O processo descrito pela Figura 1 é descrito abaixo:

1. Solicitação de transação por um usuário;
2. Um bloco é criado, representando a transação deste usuário;
3. Este bloco é difundido na rede;
4. Todos os outros nós (blocos da rede) validam o novo bloco e a transação requisitada;
5. O bloco do usuário é adicionado à *blockchain*;
6. A transação é verificada e, finalmente, executada.

Além do descrito pela Figura 1, a Figura 2 demonstra como é feito o encadeamento dos blocos na *blockchain*. Os blocos são encadeados por meio do *hash* do bloco anterior, o qual irá ser concatenado juntamente ao conteúdo original. Essa técnica de encadeamento dos blocos garante a imutabilidade da *blockchain*, pois o *hash* presente em um bloco concatenado ao conteúdo deve sempre ser igual ao *hash* do bloco anterior.

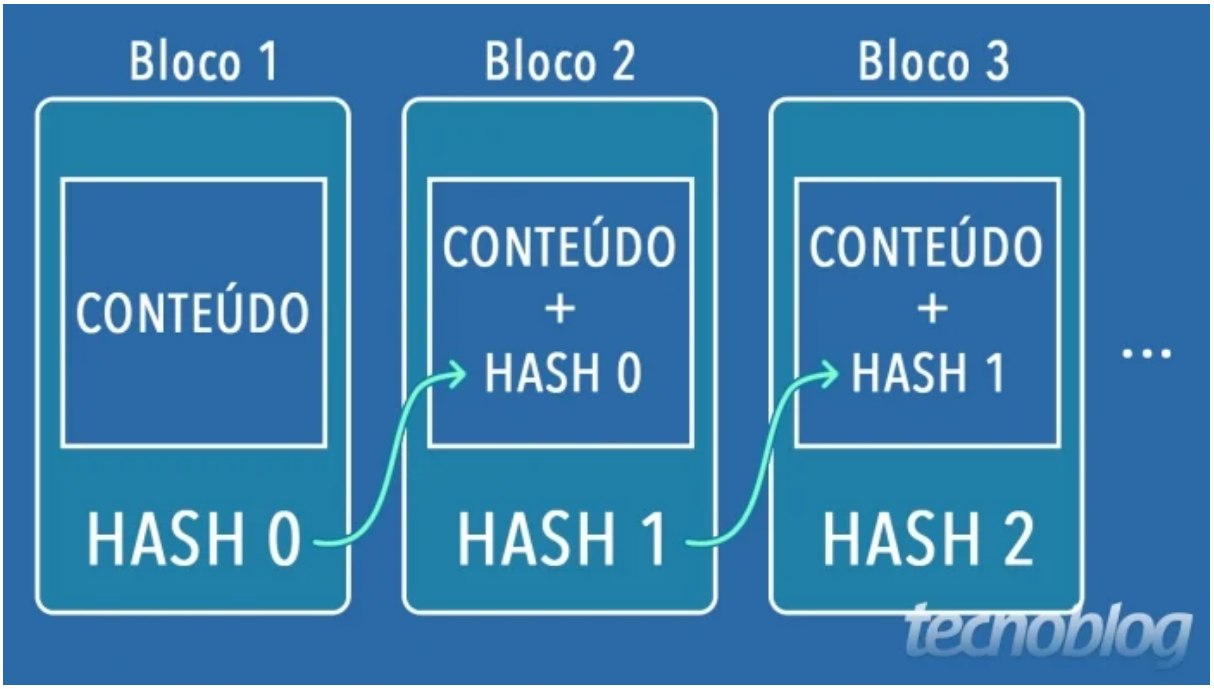

Figura 2: Encadeamento dos blocos da *blockchain*

Disponível em:

https://tecnoblog.net/wp-content/uploads/2017/11/como-funciona-blockchain-700x394.jpg

2.3 Hyperledger Fabric

A Hyperledger Fabric é uma plataforma de código aberto modular e extensível de desenvolvimento de *blockchain*. Ela é um projeto hospedado pela Linux Foundation, que permite a criação de um tipo de *blockchain* privada permissionada (Elli Androulaki et al.). Nela, todos os participantes são conhecidos, diferente de *blockchains* usadas em criptomoedas, como a Bitcoin. Por conta disso, não é necessário protocolos de Proof of Work (PoW), pois os usuários se inscrevem via Membership Service Provider (MSP).

Ele suporta protocolos de consenso modulares e tem como objetivo o fornecimento de um sistema que possa executar *Smart Contracts*, garantindo a segurança e a escalabilidade das aplicações finais, podendo executar aplicações escritas em diferentes linguagens de programação.

Como ele possibilita a criação de uma *blockchain* privada, somente participantes conhecidos e autenticados através de certificado tem acesso à aplicação.

2.4 Smart Contracts

"[Smart Contract] é um protocolo de transação computadorizado que executa os termos de um contrato" (Szabo, 1994), que nos permite traduzir cláusulas contratuais em código embutido, diminuindo participações externas e riscos envolvidos.

Os *Smart Contracts*, ou contratos inteligentes, são soluções para garantir automaticamente a eficácia de um contrato. Ele conta com o uso de execuções de códigos automáticas baseadas nos termos definidos no contrato escrito. Essas execuções são definidas nas cláusulas do documento de contrato, e são executadas mediante consulta realizada pelo próprio *Smart Contract*.

Essa solução garante a segurança para as duas partes do contrato, dado que só serão feitas as devidas transações com a devida prova dos quesitos definidos no contrato. Além disso, como é feito com auto-execução de código, esse tipo de contrato deixa de necessitar do envolvimento de uma terceira parte, como intermediários e testemunhas. Além disso, os *Smart Contracts* são baseados em *blockchain*, assegurando ainda mais as operações pela sua natureza imutável e descentralizada.

No contexto do Hyperledger Fabric, o *smart contract* é chamado de *chaincode*.

2.5 *Ledgers*

Ledger, ou Livro Razão, é um elemento fundamental no Hyperledger . Este livro armazena informações atuais e históricas sobre os objetos específicos do negócio (Elli Androulaki et al.). Estas informações são referentes a todos os fatos ocorridos em cima deste objeto. Esses fatos podem ser assimilados a uma conta bancária, por exemplo. Ao observarmos uma conta bancária podemos verificar nosso saldo (momento atual do objeto) e também podemos verificar o nosso extrato (quais foram os fatos que levaram o saldo ao estado atual). E, assim como um banco, todos os fatos que antecederam o estado atual do objeto são imutáveis, a fim de manter um histórico conciso. Esse constante controle sobre os estados de um objeto em específico, como uma conta bancária, junto com a imutabilidade desse controle faz a *blockchain* ser altamente relevante para instituições financeiras e jurídicas, com o objetivo de aumentar a segurança das operações desses tipos de instituições.

2.6 *Orderers*

Orderers, ou Serviços de Ordens, são o conjunto de nós ordenadores (ou nó de ordenação). Em uma arquitetura de *blockchain* baseada em algoritmos de consenso determinístico, como é o caso da Hyperledger Fabric, qualquer bloco validado pelo par é garantido como final e correto (Elli Androulaki et al.). Os nós ordenadores fazem a ordem das transações na *blockchain*, promovem a finalização e separação do endosso da ordem de execução do *chaincode*, além de também eliminar gargalos que podem ocorrer quando a execução é realizada nesses mesmos nós. Eles são os responsáveis por garantir o consenso nas decisões da rede.

2.7 *Peers*

Peers são elementos fundamentais do Hyperledger, pois armazenam tanto *Ledgers* quanto *Chain Codes*. Eles são membros das organizações que interagem com a rede, podendo hospedar *ledgers* e *smart contracts*.

Esse elemento se torna fundamental para a rede por conta dele ser responsável por expor algumas APIs para que, tanto administradores quanto usuários, interajam com o serviço fornecido. Além disso, um *peer* não se limita apenas a 1 *ledger* e 1 *chain code*, ele pode possuir uma quantidade qualquer, e até mesmo diferente, de ambos. Com isso em mente, o (Elli Androulaki et al.). *peer* também pode ser chamado de um host de *ledgers* e *chain codes* e, sem ele, não seria possível que usuários e administradores executassem operações sobre a rede.

2.8 Docker

Docker é uma plataforma de código aberto que executa aplicativos que são empacotados com todas as suas dependências de suporte necessárias em um formato padrão chamado de *container*, tornando o aplicativo mais fácil de se desenvolver e distribuir. Esses contêineres continuam funcionando de forma isolada um dos outros, em camadas acima do kernel do sistema operacional, tornando sua execução independente do sistema operacional das máquinas que estão hospedando os containers (Babak Bashari Rad, 2017).

Em um ambiente de contêiner no qual os aplicativos são virtualizados e executados, o docker adiciona uma camada extra de mecanismo de implementação em cima dele, de maneira que o docker forneça um ambiente rápido e leve no qual o código pode ser executado de forma eficiente e otimizado.

O docker funciona da maneira ilustrada na Figura 3:

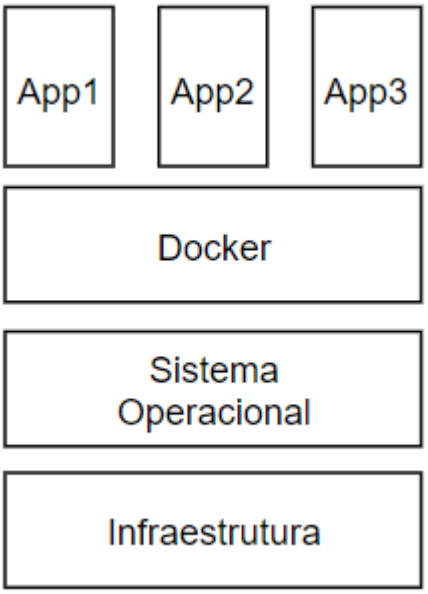

Figura 3: Funcionamento do Docker

Fonte: Elaborada pelos autores

Como comentado anteriormente, o docker é executado em cima do sistema operacional, e divide as aplicações em contêineres isolados, no qual cada aplicação tem apenas o necessário para sua execução, porém sendo executada pelas camadas inferiores na arquitetura.

2.9 HTTP

HTTP é um protocolo que gerencia a lógica por trás da transmissão de páginas da web de um servidor para outro (Karanpreet Singh, 2017). Quase todas as organizações comerciais ou não comerciais com presença na internet possuem um site para fornecer a clientes informações ou produtos com disponibilidade ininterrupta. O HTTP permite a obtenção de dados e operações na web através de requisições GET. A Figura 4 exemplifica um fluxo de obtenção de informações na web através de múltiplas requisições GET, que usam o protocolo HTTP para a comunicação entre o requisitante (cliente) e os servidores.

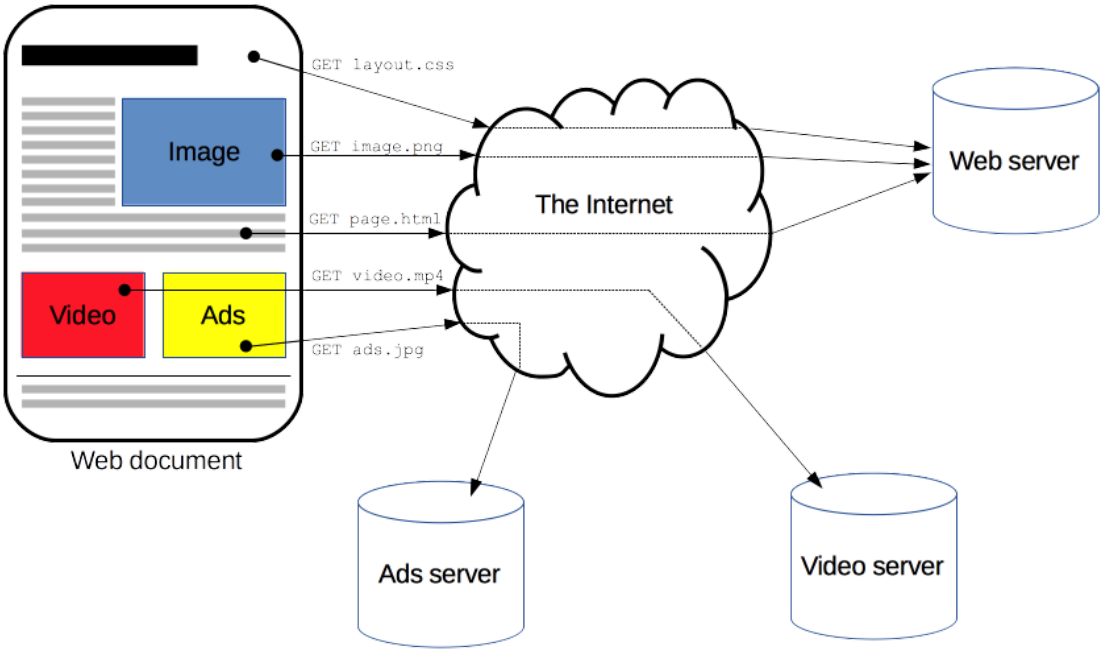

Figura 4: Representação de requisições através de HTTP

Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview

As requisições enviadas pelo cliente para o servidor são chamadas de solicitações, ou *requests*. As mensagens enviadas pelo servidor como resposta são chamadas de respostas, ou *responses*.

2.10 JWT

JSON Web Token (ou simplesmente JWT) é uma solução escalável para controle de acesso de usuários para sistemas distribuídos (Jánoky et al.). Este token segue o padrão RCT 7519, um padrão da indústria para realizar autenticação entre duas partes utilizando um token assinado para autenticar uma requisição. A Figura 5 ilustra o conteúdo do JWT.

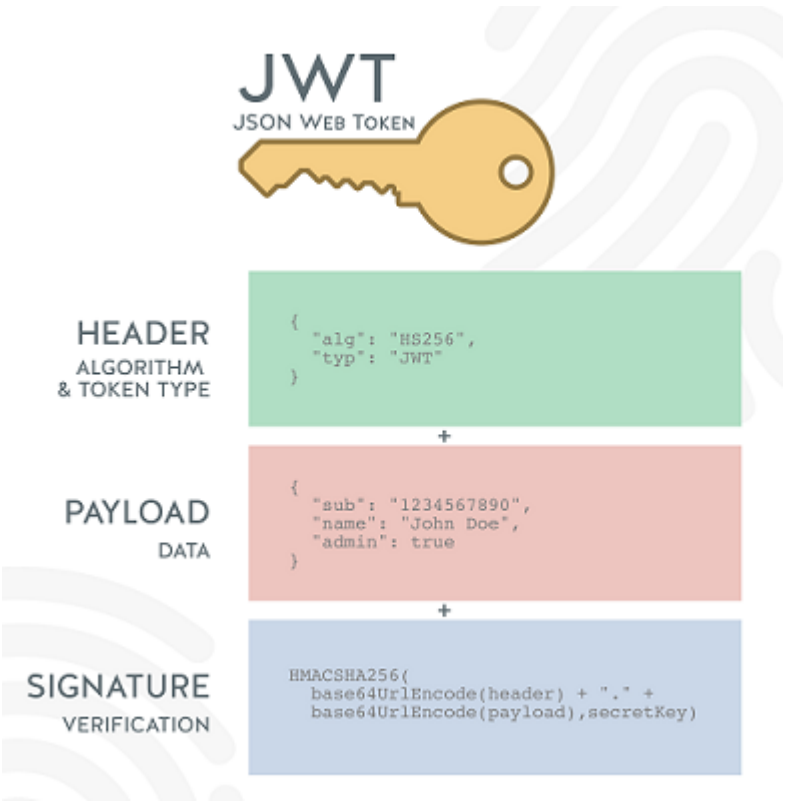

Figura 5: Representação do conteúdo JWT

Disponível em: http://www.macoratti.net/19/06/aspnc_autjwt1.htm

Um JWT é composto por 3 componentes:

- Header: Possui dois campos, *alg*, que informa qual algoritmo é utilizado para criar o *hash* das assinaturas e *typ*, que informa que isto é um JWT;
- Payload: Possui os campos referentes à autenticação, como por exemplo email e senha;
- Signature: Assinatura única de cada token a qual é gerada a partir do algoritmo contido no Header. Seu corpo é criado com base no header, payload e no segredo (uma sequência de caracteres aleatórios) definido pela aplicação.

3. TRABALHOS CORRELATOS

3.1 Revisão Bibliográfica Sistemática

Para a elaboração do presente trabalho foi necessário um levantamento do estado da arte do uso das tecnologias de *blockchain* com o contexto de preservação de dados, além da

tecnologia Hyperledger Fabric, escolhida para o desenvolvimento da proposta do trabalho. A Tabela 1 apresenta palavras-chave pesquisadas no Google Scholar e o total de aparições em artigos e trabalhos. Os dados da Tabela 1 mostraram se as palavras-chave, relacionadas ao tema, são relevantes ou não no mundo acadêmico.

Palavra-Chave	Total
“Data preservation”	3.920.00
“Smart contract”	1.555.000
“Blockchain”	279.000
“Hyperledger Fabric”	10.600
“Blockchain”, “Smart contract”	39.700
“Blockchain”, “Data preservation”	700
“Blockchain”, “Data preservation”, “Smart contract”	324
“Hyperledger Fabric”, “Data preservation”	142
“Hyperledger Fabric”, “Data preservation”, “Smart contract”	118

Tabela 1 - Revisão Bibliográfica Sistemática

Dados da tabela 1 foram retirados do Google Scholar.

A partir dos dados expostos na Tabela 1, pode-se afirmar que a preservação de dados é um tema de extrema relevância, com 3.9 milhões de resultados. Além disso, também pode-se verificar uma grande preocupação com smart contracts e *blockchain*, possuindo, respectivamente, 1.5 milhões e 279 mil resultados. Porém, ao combinar estes três temas, os resultados baixam drasticamente, para apenas 324 e, ao especificar o tipo de *blockchain* para Hyperledger Fabric, os resultados demonstram-se ainda menores, apenas 118. A seguir será apresentado um resumo dos artigos levantados como mais relevantes advindos da revisão bibliográfica sistemática. Os artigos foram escolhidos baseados na sua proximidade de tema com o trabalho desenvolvido.

3.2 A systematic literature review of blockchain-based applications: Current status, classification and open issues

Este artigo conceitua *blockchain* e descreve a situação em que a tecnologia foi desenvolvida, e como ela afetou as tecnologias ao redor dela. Satoshi Nakamoto descreveu como a tecnologia de *blockchain*, uma estrutura distribuída conectada peer-to-peer, poderia resolver problemas que existiam para manter a ordem e a integridade de operações de transição. Quase uma década depois os autores do artigo "A systematic literature review of blockchain-based applications: Current status, classification and open issues" mostram de maneira geral como as tecnologias e aplicações de *blockchain* fizeram para resolver os problemas levantados por Satoshi Nakamoto, como descrevem as mudanças disruptivas que ocorreram em consequência disso também, usando aplicações de negócio e criptomoedas como modelos de exemplificar todas essas mudanças.

Os autores também comentam sobre a quantidade de trabalhos de revisão em relação às tecnologias de *blockchain*, cada uma com seu foco e particularidade, dando exemplo de seus estudos na área de IoT, na gerência de big data de maneira descentralizada, nos desafios de segurança da *blockchain* etc, porém é discutida a atenção limitada do estudo do estado da arte em relação aos aplicativos habilitados para *blockchain*, sendo a motivação para o desenvolvimento do trabalho.

O artigo discorre através de explicações sobre *blockchain* e suas aplicações e seus recursos, mostrando o crescente aumento no interesse da comunidade acadêmica no assunto, e identificando três principais áreas de pesquisa:

1. Classificação do alcance das aplicações baseadas em *blockchain* através de diversos setores acadêmicos e de negócio;
2. Aptidão/possibilidade das tecnologias de *blockchain* de gerar valor nos setores levando em conta as limitações que a tecnologia apresenta;
3. Apresenta um guia para os pesquisadores com interesse na área através de um roteiro de pesquisas promissoras, desafios e oportunidades em que é necessário um avanço na pesquisa.

3.3 Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains

Neste artigo os autores iniciam com uma introdução aos conceitos de *blockchain*. Segundo eles, a *blockchain* é uma forma confiável de realizar transações digitais por conta da criptografia usada. Além disso, também são trazidos os conceitos de *blockchain* permissionada e não permissionada.

Segundo Androulaki et al, 2018, uma *blockchain* não permissionada, ou pública, normalmente envolve criptomoedas (como Bitcoin) como meio de recompensar os esforços dos membros daquela *blockchain*. Essas recompensas são dadas com base nos trabalhos de mineração do usuário, o chamado *Proof of Work* (PoW). O PoW se dá através da resolução de complicados problemas matemáticos utilizados na validação de transações e na mineração de novos tokens. Por conta da complexidade desses problemas, é requerido um alto esforço computacional para resolvê-lo, o que implica que esses tipos de *blockchain* possuem como requisito a participação de dispositivos com capacidade de processamento mais potente.

Já uma *blockchain* permissionada (ou privada) tem como objetivo unir interesse de grupos que não confiam totalmente entre si. Diferentemente da *blockchain* pública, a permissionada requer um convite para que se possa atuar resultando, assim, no conhecimento de todas as partes envolvidas na *blockchain*. Levando em conta essa falta de confiança entre as partes, a *blockchain* privada utiliza o tradicional consenso de Tolerância a Falhas Bizantinas (TFB). Porém, segundo os autores, esses tipos de *blockchain* sofrem algumas limitações. Com essas limitações, surgiu o projeto de uma *blockchain* permissionada com o objetivo de suprir todas estas limitações: *Hyperledger Fabric*. O Fabric é a primeira blockchain a permitir a execução de aplicações distribuídas escritas em linguagem convencional, ocasionando em uma execução consistente entre vários nodos, com a impressão de uma execução centralizada.

Segundo os autores, Fabric introduz uma nova arquitetura *blockchain*, com objetivos como: flexibilidade, escalabilidade e confidencialidade. Para cumprir estes objetivos, Fabric segue um novo paradigma de *execute-order-validate* para ambientes não confiáveis. As transações são separadas, segundo os autores, em três etapas:

1. Execução e verificação da transação, e finalmente aprovando-a;
2. Ordenação via protocolo de consenso, independente das semânticas da transação;
3. Validação de transação por suposições de confiança, evitando condições de corrida, devido a simultaneidade.

3.4 Blockchain-Based Smart Contracts: A Systematic Mapping Study

Neste artigo os autores citam os conceitos principais de um *smart contract*, suas vantagens e seus maiores desafios. Logo nos primeiros parágrafos, os autores conceituam o *smart contracts* como sendo códigos executáveis que são executados em *blockchain*, com o objetivo de facilitar, executar e impor acordo entre partes não confiáveis sem o envolvimento de um *Trusted Third Party* (TTP). Tendo em vista a ausência de um TTP, os *smart contracts* resultam em transações de custo mais baixo. Outro fato interessante trazido pelos autores é que os *smart contracts* comumente são desenvolvidos na *blockchain* Ethereum, por conta do suporte a *Turing-completeness*. Essa característica, segundo os autores, permite o desenvolvimento de smart contracts mais avançados e customizados.

Smart contracts podem ser escritos em diversas linguagens, tais como Java, Javascript, C++ dentre outras. Isto porque a capacidade dos *smart contracts* não depende da linguagem, e sim de instituições políticas, jurídicas e empresariais, que ditam os limites destes contratos. Além disso, estes contratos também possuem alguns problemas, como:

1. Codificação: segundo os autores, *smart contracts* podem ter problemas de codificação pois, nem sempre, os próprios desenvolvedores do contrato conseguem escrevê-lo da maneira que gostariam que funcionasse;
2. Segurança: segundo os autores, *smart contracts* podem ter problemas com a dependência das ordens de transação. Esse problema ocorre quando duas transações, dependentes entre si e que chamam o mesmo contrato, estão contidas no mesmo bloco, o que pode gerar uma brecha para atacantes;
3. Privacidade: segundo os autores, *smart contracts* podem ter problemas com falta de privacidade nas transações. *Blockchains* deixam publicamente visíveis dados como saldo de usuário e transações. Tendo isso em vista, alguns usuários podem escolher por não adotar o uso de *smart contracts*;
4. Desempenho: segundo os autores, *smart contracts* podem ter problemas de desempenho por conta da execução sequencial dos contratos, o que limita a desempenho quando comparado a uma execução paralela.

3.5 A Data Preservation Method Based on Blockchain and Multidimensional Hash for Digital Forensics

Neste artigo os autores começam comentando sobre como é fácil criar, armazenar, transferir e usar dados digitais, e, com essa mesma facilidade, também é possível modificar dados na investigação forense. Porém, é vital que a integridade das evidências digitais sejam

asseguradas pelos seus responsáveis, garantindo que os dados fiquem íntegros e com credibilidade. O crescimento das tecnologias de preservação de dados, como criptografia, assinatura digital, timestamp, dentre outras, foi possível aumentar a maturidade do auxílio digital à processos judiciais, tanto no processo de investigação como durante o tribunal.

Os autores discorrem sobre as principais vantagens que esse tipo de tecnologia trazem, como a possibilidade dos dados de evidências serem perfeitamente preservados e "congelados no tempo", automatização de processos de preservação de maneira não intrusiva, além de deixar a validação e a medição capazes de serem feitas independente do acesso às cenas de crime. Com o crescimento do desenvolvimento desse tipo de tecnologia usada na preservação de dados, a desempenho de seus usos serão cada vez melhores e com o custo cada vez mais otimizado.

A abordagem mais popular na área de preservação de dados é a combinação da criptografia de dados e digestão de dados. Ela usa algoritmos de criptografia simétrica e assimétrica para assinar dados, combinando as informações de timestamp com os dados e, em seguida, gera um hash com o algoritmo de hash. Dessa maneira, quando os dados são usados para finalidade judicial, os investigadores podem usar abordagens idênticas na ordem inversa, com finalidade de validar se os dados sofreram qualquer tipo de manipulação. Apesar desses benefícios, há uma desvantagem nessa abordagem de preservação de dados, que é que todos os processos são executados pelo investigador, e, por causa disso, ninguém pode garantir que os investigadores em si não cometam erros, sendo intencionalmente ou involuntariamente.

Os autores comentam ainda que, mesmo existindo muitos trabalhos que tenham sido feitos sobre preservação de dados, não existe um modelo ou método bem definido para a análise forense digital. Então, são levantados os seguintes problemas:

1. Baixo nível de automação no processo de preservação de dados;
2. Alto nível de risco no processo de preservação de dados;
3. Falta de garantia de segurança dos dados digitais;
4. Falta de confiança mútua entre as partes envolvidas.

4. ABORDAGEM DA PROPOSTA

Neste capítulo serão descritos os detalhes da proposta de coleta, preservação e visualização de evidências digitais com o uso de blockchain. Será apresentado um modelo de arquitetura que terá dois cenários possíveis de uso, que irão se complementam para a solução proposta, que serão reproduzidas com a utilização das ferramentas do Hyperledger Fabric.

A Figura 6 ilustra a arquitetura proposta:

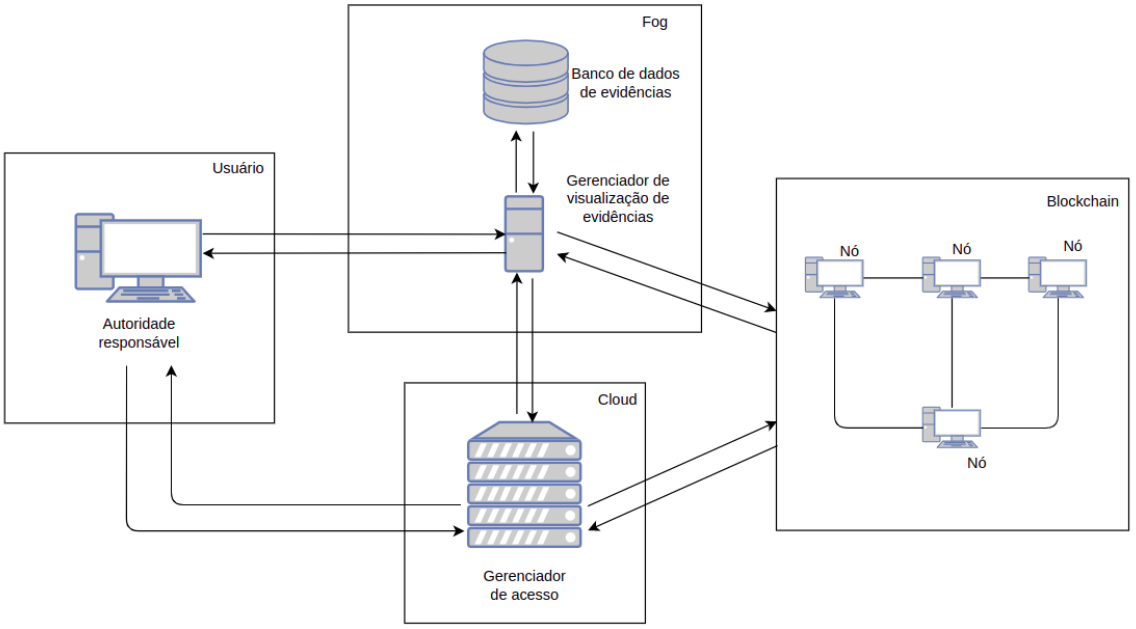

Figura 6: Arquitetura de preservação de evidências digitais

Fonte: Elaborada pelos autores

A arquitetura conta com os seguintes componentes:

- Autoridade responsável, representando os membros autorizados a participarem da rede blockchain;
- Gerenciador de acesso, controlando o acesso às informações na blockchain e direcionando as requisições do banco de dados ao Gerenciador de visualização de evidências;
- Gerenciador de coleta e visualização de evidências, realizando a coleta de novos arquivos, controlando o acesso aos arquivos salvos no banco de dados de evidências e fazendo a requisição de adição de um novo bloco de arquivo adicionado à blockchain;
- Banco de dados de evidências, contendo os arquivos criptografados a serem guardados;
- Blockchain, contendo as informações sobre as transações sobre os arquivos a serem salvos no banco de dados.

Sobre a relação dos componentes com os pilares da segurança da informação, o Gerenciador de acesso e o Gerenciador de coleta e visualização de evidências devem permitir que apenas usuários com as devidas permissões possam acessar as informações na blockchain e os arquivos no banco de dados, garantindo a confidencialidade e integridade na arquitetura, já que usuários sem permissão não conseguiriam fazer operações de visualização nem edição

de informações. Outro componente que ajuda na integridade das informações é a propriedade de imutabilidade dos blocos de informação da blockchain, impedindo que dados já inseridos na blockchain possam ser alterados. Falando sobre a disponibilidade, a característica que ajuda a garantir esse pilar da segurança da informação é o fato de parte da arquitetura ser disponibilizada através da computação em nuvem, além da rede blockchain ser distribuída.

A arquitetura proposta, ilustrada na Figura 6, conta ainda com 4 camadas:

- Blockchain;
- Fog;
- Cloud;
- Usuário.

A camada Blockchain é composta pelos nós de mineração, que são máquinas distribuídas que se conectam à rede para a garantia da disponibilidade e segurança dos dados recebidos. Pelo uso da blockchain é garantido que os dados serão imutáveis e distribuídos.

A camada de *Fog* é composta por um componente atuando como gerenciador de coleta e visualização de evidências e pelo banco de dados de evidências. O gerenciador de coleta e visualização de evidências realiza o gerenciamento da preservação de dados que foram gerados pela camada de sensores e atuadores do sistema, deixando-os também disponíveis para a visualização das evidências, quando solicitado pelo gerenciador de acesso presente na camada de Cloud. Ela tem o objetivo de gerenciar a coleta e a visualização das evidências através de seu armazenamento local.

A camada de usuário é composta por um usuário que pode se conectar e solicitar ao gerenciador de acesso o devido acesso aos seus dados armazenados. Para que o acesso seja concedido, o usuário deve fornecer ao gerenciador de acesso dados que comprovem sua legitimidade perante ao acesso, como um token de acesso que é gerado no login do usuário.

A camada cloud é composta somente por um gerenciador de acesso. Este componente tem o objetivo de atuar como uma porta de entrada ao serviço, ele ficará responsável por se comunicar com todas as outras 3 camadas da arquitetura, tanto enviando quanto recebendo requisições.

Sobre os cenários de uso da arquitetura citados, um deles fará o papel da coleta de evidências digitais e o outro será responsável pela visualização de evidências digitais.

4.1 Cenário de coleta de evidências digitais

Neste primeiro cenário temos como objetivo a coleta e a preservação de evidências vindas do envio manual do usuário dentro da arquitetura. A Figura 7 ilustra o processo de operações que acontecem neste cenário de uso.

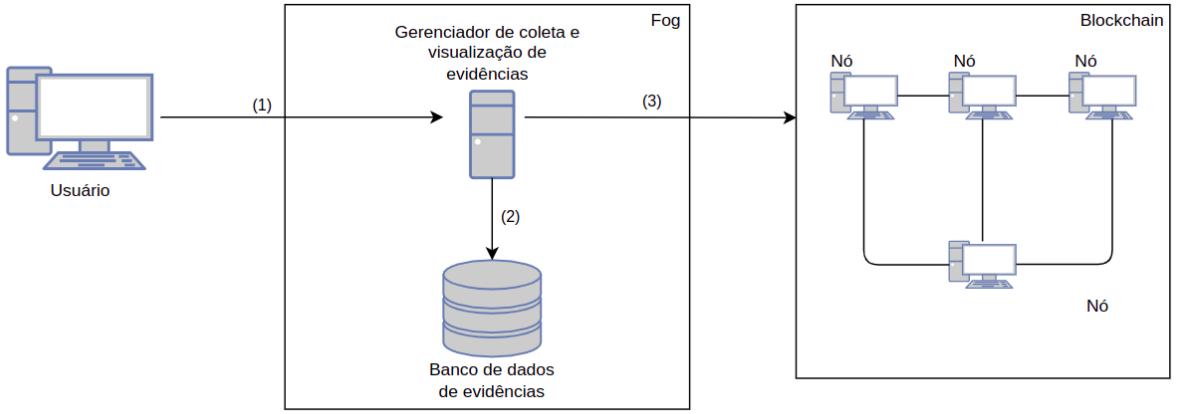

Figura 7: Cenário de coleta de evidências

Fonte: Elaborada pelos autores

Dando início ao processo de envio de dados, um usuário devidamente autenticado faz o envio da informação a ser preservada para o Gerenciador de coleta e visualização de evidências.

Em seguida, o gerenciador de coleta e visualização de evidências criptografa o arquivo enviado e armazena-o no Banco de Dados de Evidências. Após o arquivo ser salvo no banco de dados, é criado e adicionado um novo bloco de informação na blockchain. Esse novo bloco contém um ID único da transação de adição do bloco à blockchain, o nome do arquivo salvo, um resumo criptográfico do conteúdo do arquivo, marca temporal da transação e o usuário dono do arquivo. Com isso, a blockchain adiciona esse novo bloco de informação à rede de blockchain.

4.2 Cenário de visualização de evidências digitais

Neste segundo cenário temos como objetivo a visualização das evidências salvas no banco de dados e a visualização das transações realizadas na arquitetura. A Figura 8 ilustra o processo de operações que acontecem neste cenário de uso.

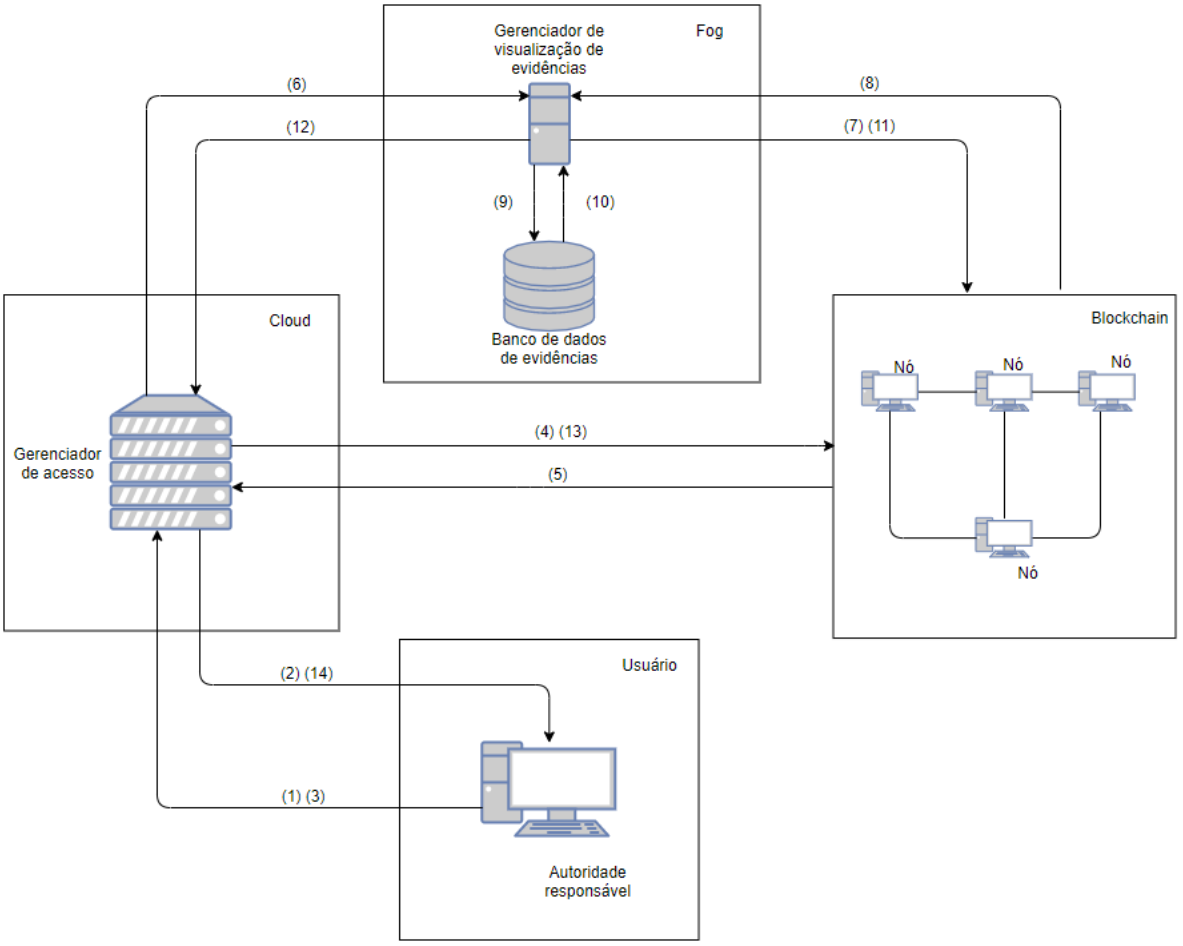

Figura 8: Cenário de visualização de evidências

Fonte: Elaborada pelos autores

Conforme a Figura 8, são necessárias 14 operações para que o usuário tenha acesso aos dados armazenados, são elas:

1. A primeira operação é onde o usuário requer acesso aos dados, no qual ele faz uma requisição ao gerenciador de acesso;
2. A segunda operação o gerenciador de acesso solicita identificação ao usuário via token de acesso;
3. A terceira operação o usuário solicita acesso aos dados armazenados e envia seu token de acesso ao gerenciador de acesso;
4. A quarta operação ocorre quando um certificado digital é fornecido, o qual será utilizado pelo gerenciador de acesso para verificar quais dados estão vinculados ao portador deste certificado;
5. A quinta operação é o retorno da requisição feita à *Blockchain*, ao gerenciador de acesso;

6. A sexta operação conta com o gerenciador de acesso enviando ao GVE uma requisição de acesso aos dados determinado na terceira operação, juntamente com o seu token de acesso;
7. A sétima operação conta com um gerenciador de acesso enviando uma requisição de acesso para o GVE. É feita uma verificação via *smart contract* da validade do certificado digital e quais as permissões de acesso que ele garante;
8. A oitava operação é o retorno da *Blockchain* ao GVE, apresentando informações sobre dados os quais o usuário tem acesso;
9. A nona operação verifica a possibilidade do usuário ter acesso aos dados. Caso tenha, será feita uma busca via ID no banco de dados;
10. A décima operação conta com banco de dados enviando os dados armazenados ao GVE;
11. Na décima primeira operação o GVE realiza uma verificação de integridade dos dados, os quais são enviados à *Blockchain*. A verificação é feita a partir de uma comparação de "Hashes" dos dados presentes na *Blockchain* e da aplicação do algoritmo SHA-3, 256bits, aos dados fornecidos pelo banco de dados. Finalmente, é feito o envio à *Blockchain* algumas informações, são elas:
    - "Hash válido" : variável booleana (aceita apenas os valores verdadeiro ou falso);
    - "Intermediadores": consta o local em que estava armazenado e a identificação do GVE;
    - "Responsável": identificação do usuário que solicitou acesso;
    - "*Timestamp* da transação": momento em que a transação é processada pelo GPE antes do envio à *Blockchain*.
12. Na décima segunda operação o GVE envia os dados, compactados e criptografados, ao gerenciador de acesso;
13. Na décima terceira operação é feito o envio à *Blockchain* as informações: "intermediadores", "responsável" e "*timestamp* da transação";
14. Na décima quarta, e final, operação, o gerenciador de acesso encaminha os dados cifrados ao usuário.

5. DESENVOLVIMENTO DA PROPOSTA

5.1 Blockchain[1][2]

Para a implantação da blockchain é usada a própria biblioteca do Hyperledger Fabric. Com essa biblioteca, possibilitamos que os usuários façam 2 tipos de operação na blockchain, inserção e leitura de blocos. Para instalação dessa biblioteca, é utilizado o comando *curl -sSL https://bit.ly/2ysbOFE | bash -s*. Este comando irá criar uma pasta chamada *fabric-samples*, na qual será possível, então, a instanciação de uma blockchain.

Para que seja feito o lançamento da blockchain é utilizado o comando *./network.sh up createChannel -c channel1 -ca*. O resultado desta operação encontra-se ilustrado na Figura 9:

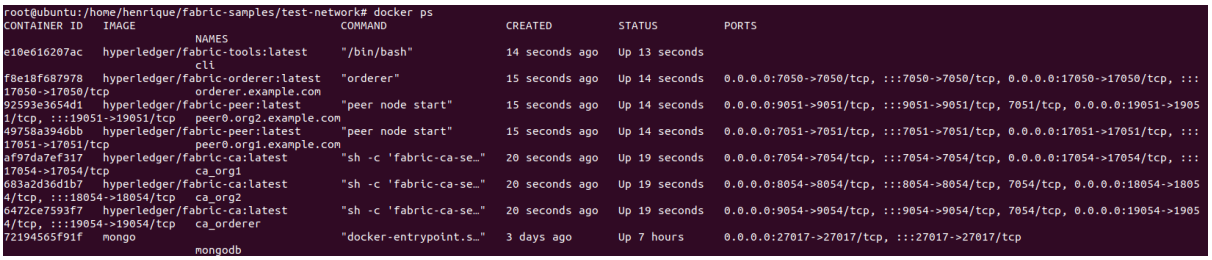

Figura 9: Resultado do comando *./network.sh up createChannel -c channel1 -ca*
Fonte: Elaborada pelos autores

Após instanciados os nodos da blockchain, foi feita a instalação do *chaincode* em ambos os *peers*. Para isso, foram utilizados alguns comandos, como demonstrado na Figura 10, são eles:

- *peer lifecycle chaincode package basic.tar.gz –path ../asset-transfer-basic/chaincode-javascript/ –lang node –label basic_1.0*: Exporta o *chaincode* no *path* especificado para o *path* atual;
- *. ./scripts/envVar.sh*: Possibilitar a execução de comandos para trocar entre a organização 1 e a organização 2;
- *setGlobals n*: Definir para qual organização gostaríamos de utilizar no momento, sendo n igual ao número da organização desejada;
- *peer lifecycle chaincode install basic.tar.gz*: Instalação da *chaincode* na organização atual.

[1] O código fonte da blockchain e da API está disponível nesse endereço: https://github.com/henriquehschmitt1/tcc-blockchain
[2] Um vídeo explicativo do passo a passo de configuração e utilização da rede está disponível nesse endereço: https://www.youtube.com/watch?v=K1OZMchIKpU

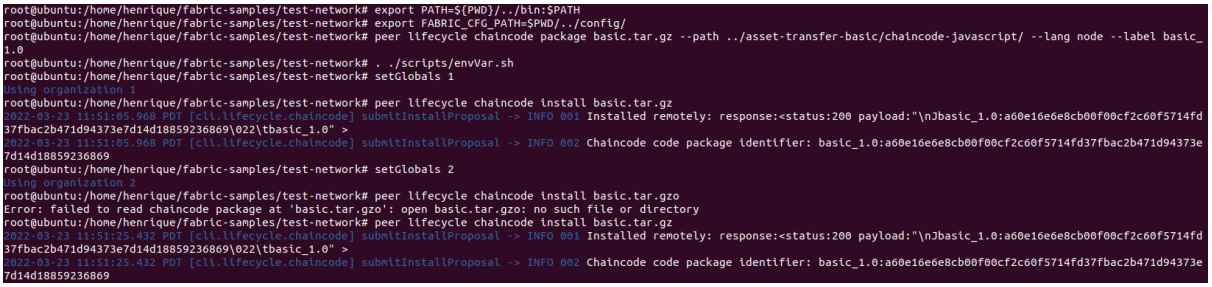

Figura 10: Execução dos comandos de instalação da *chaincode*
Fonte: Elaborada pelos autores

Logo após a instalação da *chaincode* nos *peers*, é feita uma consulta pelo ID do processo do *chaincode* que foi instalado. Este passo é necessário pois este ID será exportado como uma variável de ambiente para que, futuramente, possamos aprovar este *chaincode*. Para isso, mudamos para a organização 1 utilizando o comando setGlobals 1 e executamos dois comandos, como demonstrado na Figura 11:

- *peer lifecycle chaincode queryinstalled --peerAddresses localhost:7051 --tlsRootCertFiles organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt*: Faz uma consulta diretamente pelo ID da *chaincode*;
- export PKGID=basic_1.0:a60e16e6e8cb00f00cf2c60f5714fd37fbac2b471d94373e7d14d18859236869: Exporta o ID em questão para uma variável de ambiente para uso futuro.

Figura 11: Execução dos comandos para exportar o ID da *chaincode*.
Fonte: Elaborada pelos autores

Após exportarmos o ID da *chaincode*, iremos fazer a aprovação dela para ambas as organizações utilizando os comandos, como demonstrado na Figura 12:

- *setGlobals n*: Definir para qual organização gostaríamos de utilizar no momento, sendo n igual ao número da organização desejada;
- *peer lifecycle chaincode approveformyorg -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile $PWD/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem --channelID channel1 --name basic --version 1 --package-id $PKGID --sequence 1*: Aprova a *chaincode* para a organização atual, com base no ID da *chaincode*.

Figura 12: Execução dos comandos para aprovar a *chaincode* pelo ID para ambas as organizações.
Fonte: Elaborada pelos autores

E, por fim, após aprovar a *chaincode* para ambas as organizações, faremos a publicação dela na rede local e iremos validar se ocorreu tudo como o planejado, como mostrado na Figura 13, através dos seguintes comandos:

- *peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile $PWD/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem --channelID channel1 --name basic --peerAddresses localhost:7051 --tlsRootCertFiles $PWD/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt --version 1 --sequence 1*: Publicação das *chaincodes* para ambas as organizações;
- *peer lifecycle chaincode querycommitted --channelID channel1 --name basic --cafile ${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem*: Comando utilizado para verificar se a intalação da *chaincode* ocorreu com sucesso.

```
root@ubuntu:/home/henrique/fabric-samples/test-network# peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile $PWD/organi
zations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem --channelID channel1 --name basic --peerAddresses localhost:7051 --tl
sRootCertFiles $PWD/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles organizations/peerOrga
nizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt --version 1 --sequence 1
2022-03-23 11:52:41.291 PDT [chaincodeCmd] ClientWait -> INFO 001 txid [c5040cab2ed38b58e864f6ae7f3134526fd24234bd3a7d19640a2aa1173cf8c7] committed with status (VALID) at localhost:
7051
2022-03-23 11:52:41.297 PDT [chaincodeCmd] ClientWait -> INFO 002 txid [c5040cab2ed38b58e864f6ae7f3134526fd24234bd3a7d19640a2aa1173cf8c7] committed with status (VALID) at localhost:
9051
root@ubuntu:/home/henrique/fabric-samples/test-network# peer lifecycle chaincode querycommitted --channelID channel1 --name basic --cafile ${PWD}/organizations/ordererOrganizations/
example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
Committed chaincode definition for chaincode 'basic' on channel 'channel1':
Version: 1, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
```

Figura 13: Execução dos comandos para publicar a *chaincode* e verificar se houve êxito na publicação.
Fonte: Elaborada pelos autores

Após a execução de todos os comandos citados anteriormente, podemos verificar as imagens criadas em nosso Docker local utilizando o comando *docker ps*. Caso tudo tenha ocorrido com sucesso, é esperado que o resultado deste comando seja igual ao apresentado na Figura 14.

```
root@ubuntu:/home/henrique/fabric-samples/test-network# docker ps
CONTAINER ID   IMAGE
   COMMAND                  CREATED          STATUS          PORTS                                                                                  NAMES
0634451e2649   dev-peer0.org2.example.com-basic_1.0-a60e16e6e8cb00f00cf2c60f5714fd37fbac2b471d94373e7d14d18859236869-5cd1a211cccf3170d32161181dd942a55c1ecfcabae3aebca05a7b229bd0d914
   "docker-entrypoint.s…"   26 seconds ago   Up 25 seconds                                                                                          dev-peer0.org2.exam
ple.com-basic_1.0-a60e16e6e8cb00f00cf2c60f5714fd37fbac2b471d94373e7d14d18859236869
8a001537a987   dev-peer0.org1.example.com-basic_1.0-a60e16e6e8cb00f00cf2c60f5714fd37fbac2b471d94373e7d14d18859236869-c68f0eb3b223fb9d34f8ece9fbc07bb4a4c5d26aced75c22efd9c1642d07dfd7
   "docker-entrypoint.s…"   26 seconds ago   Up 25 seconds                                                                                          dev-peer0.org1.exam
ple.com-basic_1.0-a60e16e6e8cb00f00cf2c60f5714fd37fbac2b471d94373e7d14d18859236869
e10e616207ac   hyperledger/fabric-tools:latest
   "/bin/bash"              3 minutes ago    Up 3 minutes                                                                                           cli
f8e18f687978   hyperledger/fabric-orderer:latest
   "orderer"                3 minutes ago    Up 3 minutes    0.0.0.0:7050->7050/tcp, :::7050->7050/tcp, 0.0.0.0:17050->17050/tcp, :::17050->17050/tcp             orderer.example.com
92593e3654d1   hyperledger/fabric-peer:latest
   "peer node start"        3 minutes ago    Up 3 minutes    0.0.0.0:9051->9051/tcp, :::9051->9051/tcp, 7051/tcp, 0.0.0.0:19051->19051/tcp, :::19051->19051/tcp   peer0.org2.example.
com
49758a3946bb   hyperledger/fabric-peer:latest
   "peer node start"        3 minutes ago    Up 3 minutes    0.0.0.0:7051->7051/tcp, :::7051->7051/tcp, 0.0.0.0:17051->17051/tcp, :::17051->17051/tcp             peer0.org1.example.
com
af97da7ef317   hyperledger/fabric-ca:latest
   "sh -c 'fabric-ca-se…"   4 minutes ago    Up 4 minutes    0.0.0.0:7054->7054/tcp, :::7054->7054/tcp, 0.0.0.0:17054->17054/tcp, :::17054->17054/tcp             ca_org1
683a2d36d1b7   hyperledger/fabric-ca:latest
   "sh -c 'fabric-ca-se…"   4 minutes ago    Up 4 minutes    0.0.0.0:8054->8054/tcp, :::8054->8054/tcp, 7054/tcp, 0.0.0.0:18054->18054/tcp, :::18054->18054/tcp   ca_org2
6472ce7593f7   hyperledger/fabric-ca:latest
   "sh -c 'fabric-ca-se…"   4 minutes ago    Up 4 minutes    0.0.0.0:9054->9054/tcp, :::9054->9054/tcp, 7054/tcp, 0.0.0.0:19054->19054/tcp, :::19054->19054/tcp   ca_orderer
72194565f91f   mongo
   "docker-entrypoint.s…"   3 days ago       Up 7 hours      0.0.0.0:27017->27017/tcp, :::27017->27017/tcp                                                        mongodb
```

Figura 14: Execução do comando *docker ps*.
Fonte: Elaborada pelos autores

5.2 Autoridade Certificadora

No cenário proposto, a Autoridade Certificadora (AC) é responsável por possibilitar a comunicação entre a *blockchain* e a API. Utilizando métodos do próprio Hyperledger Fabric, são gerados certificados que permitem com que o usuário consiga interagir com a *blockchain* via API. Esta ação pode ser chamada de permissionamento, que é quando se concede acesso a um usuário externo conhecido à *blockchain*.

5.3 API

No cenário proposto, a API fará tanto o papel de GCVE (Gerenciador de coleta e visualização de evidências) quanto o de GA (Gerenciador de acessos). A comunicação entre a API desenvolvida e a *blockchain* é feita utilizando certificados digitais previamente gerados utilizando a AC instanciada pelos passos de configuração da rede. A api, então, atuará como ponto de entrada para que o usuário possa se registrar, autenticar e fazer as operações propostas de armazenamento de arquivo no banco de dados, registro de transações na *blockchain* e auditar todas as transações referentes a um arquivo específico. Para que todas essas ações sejam possíveis, foram criadas um total de 7 rotas que podem ser acessadas por métodos HTTP pelo usuário, são elas:

- Rota de cadastro de usuário: Acessada pelo método HTTP POST, essa rota possui 2 objetos requeridos no corpo da requisição: *email* e *password*. Se os valores foram inseridos corretamente, o usuário será inserido no banco de dados, e poderá fazer login no sistema;
- Rota de login: Também acessada pelo método HTTP POST, essa rota requer os mesmos objetos no corpo da requisição que a rota de cadastro. No entanto, ela retorna um token JWT que será utilizado para autenticação de todas as outras operações;
- Rota de cadastro de arquivo: Também acessada pelo método HTTP POST, essa rota passa a requerer um *header* (ou cabeçalho) chamado "*x-access-token*". Este token JWT é obtido ao realizar, com sucesso, login na aplicação. Além disso, também é necessário passar dois objetos no corpo da requisição: *owner* e *fileName*. *Owner* é o nome atrelado ao arquivo. *FileName* é o nome com que este arquivo será salvo no banco de dados. Se os valores foram inseridos corretamente, o conteúdo do arquivo será criptografado utilizando o método criptográfico AES-256-CBC. Após obter o resumo criptográfico do conteúdo do arquivo, são feitos dois registros, um no banco de dados, para manter o estado atual do registro, e um na blockchain, para fins de histórico transacional.
- Rota para consultar um arquivo: Acessada pelo método HTTP GET, essa rota requer, assim como a rota de cadastro de arquivo, um *header* "*x-access-token*". Além disso,

no path também é necessário passar o *fileName*. Se os valores foram inseridos corretamente, esta rota retornará o conteúdo do arquivo descriptografado;

- Rota para consultar todo o histórico transacional: Acessada pelo método HTTP GET, essa rota requer apenas o *header* "*x-access-token*". Caso o *header* esteja correto, esta rota retornará todas as transações feitas na *blockchain*;
- Rota para consultar o histórico transacional de um arquivo específico: Acessada pelo método HTTP GET, essa rota requer o *header* "*x-access-token*". Além disso, no path também é necessário passar o *fileName*. Caso os valores foram inseridos corretamente, esta rota retornará o histórico transacional do arquivo solicitado no *path*;
- Rota para atualizar o conteúdo de um arquivo: Acessada pelo método HTTP PUT, essa rota requer o *header* "*x-access-token*". Além disso, no *path* também é necessário informar o ID do arquivo presente no banco de dados. Finalmente, também é necessário informar o novo arquivo no corpo da requisição. Caso os valores sejam inseridos corretamente, é feita a atualização do conteúdo do arquivo no banco de dados. Além disso, também é feito um novo registro na *blockchain*, contendo o novo conteúdo do arquivo.

6. DIFICULDADES ENFRENTADAS

A principal dificuldade enfrentada foi a configuração da integração entre todos os componentes da arquitetura. No caso prático o maior empecilho foi a comunicação da API desenvolvida com a rede *blockchain*. Tal fato acarretava em problemas na execução do usuário às operações da *blockchain* de maneira distribuída. Com a impossibilidade de execução de maneira distribuída, a *blockchain* perde uma de suas características, que é a participação de usuários permissionados de maneira descentralizada. Porém, essa dificuldade foi contornada com a utilização de certificados de autorização gerados na Autoridade Certificadora para permitir a comunicação. Assim, foi possível comunicar devidamente as requisições entre todos os componentes da rede.

7. CONCLUSÃO

Neste trabalho foi proposto uma arquitetura de preservação de evidências digitais com o uso de *blockchain*. Para isso, foi desenvolvida uma API para intermediar a comunicação do usuário com a rede de blockchain. A API faz o papel de gerenciador de acessos às operações da blockchain e de gerenciador de coleta e visualização de evidências no banco de dados.

Também foi desenvolvida a rede de *blockchain* permissionada baseada no Hyperledger Fabric, uma organização de desenvolvimento de tecnologia de blockchain *open source*.

Com tal implementação foi possível fazer uma análise sobre a viabilidade da arquitetura proposta com o objetivo de coleta e visualização de evidências digitais. É possível ver com o funcionamento da arquitetura proposta que ela é eficiente no que propõe sobre a preservação de dados. Ele guarda conteúdo de arquivos de maneira criptografada, sendo possível sua decriptografia apenas quando requisitada por usuários devidamente autorizados. Além disso, a *blockchain* reúne informações sobre todas as operações feitas envolvendo a coleta de evidências digitais, fator de extrema importância em termos de auditoria para organizações. Finalmente, podemos afirmar que o *software* desenvolvido respeitou os três pilares da segurança da informação com êxito, utilizando uma rede de *blockchain* permissionada.

REFERÊNCIAS

ALHARBY, Maher; VAN MOORSEL, Aad. Blockchain Based Smart Contracts: a systematic mapping study. **Computer Science & Information Technology (Cs & It)**, [S.L.], v. 1, n. 1, p. 125-140, 26 ago. 2017. Academy & Industry Research Collaboration Center (AIRCC). http://dx.doi.org/10.5121/csit.2017.71011. Acesso em: 27 abr. 2021.

ANDROULAKI, Elli et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the thirteenth EuroSys conference. 2018.

CASINO, Fran; DASAKLIS, Thomas K.; PATSAKIS, Constantinos. A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and informatics, 2019.

STALLINGS, William. Network Security Essentials, 4 Edição, 2011, Prentice Hall.

RAD, Babak B., BHATTI Harrison J., AHMADI Mohammad. An Introduction to Docker and Analysis of its Performance. Asia Pacific University of Technology and Innovation Technology Park Malaysia, Kuala Lumpur, Malaysia. IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.3, March 2017.

SZABO, Nick. Smart contracts. 1994.

SINGH, Karanpreet , SINGH, Paramvir, KUMAR, Krishan. Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges. Computers & Security, Volume 65, 2017, Pages 344-372.

Jánoky LV, Levendovszky J, Ekler P. An analysis on the revoking mechanisms for JSON Web Tokens. International Journal of Distributed Sensor Networks. September 2018.

Apêndice A - Artigo Preservação de dados em Blockchain

Preservação de dados em Blockchain

Henrique H. Scmitt[1], Pedro Paulo C. Steil[1], Carla M. Westphall[1], Leandro Loffi[1]

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 15.064 – 91.501-970 – Florianópolis – SC – Brazil

Abstract. *The evolution of society together with technological evolution provides an enormous load of data generated every day. Because of this, it is necessary to ensure that the three pillars of information security are always present, namely confidentiality, integrity and availability (CID). In this context, this work aims to create a blockchain network for data preservation, in order to keep it confidential, intact and available, thus respecting the three pillars of information security. This network will have access control and visualization agents, a database to hold the files to be kept, and the blockchain.*

Resumo. *A evolução da sociedade em conjunto com a evolução tecnológica propicia uma enorme carga de dados gerados a cada dia. Por conta disso, é necessário garantirmos que os três pilares da segurança da informação estejam sempre presentes, sendo eles a confidencialidade, integridade e disponibilidade (CID). Neste contexto, este trabalho tem como objetivo criar uma rede de blockchain para preservação de dados, com o intuito de mantê-los confidenciais, íntegros e disponíveis, respeitando, assim, os três pilares da segurança da informação. Essa rede contará com agentes de controle de acesso e visualização, um banco de dados para comportar os arquivos a serem guardados, e a blockchain.*

1. Introdução

Não apenas para organizações como também para pessoas, uma das suas preocupações hoje é com a segurança de seus dados. Desde fotos de seus animais de estimação até dados sensíveis como fotos de documentos e dados financeiros a preocupação é a mesma, então, surgem questionamentos de como manter esses dados confidenciais, íntegros e disponíveis.

De modo geral, uma solução simples e direta que responde a pergunta seria fazer o uso de criptografia nos dados que interessam. Porém, com uma grande variedade de algoritmos, a escolha torna-se complexa. Nem todo algoritmo de criptografia é recomendado para tratar da integridade e segurança de dados. Na verdade são recomendados apenas algumas aplicações de hash, como autenticação de mensagem, assinatura digital e blockchain. (STALLINGS, 2014). Com o objetivo de garantir a preservação de dados e responder ao questionamento sobre os pilares da segurança da informação, das três possibilidades foi escolhido o desenvolvimento de uma solução que faça uso de blockchain, com a possibilidade de combinação com Smart Contracts.

Uma das soluções relacionadas à segurança é o uso de um Trusted Third Party (TTP). Essa solução apresenta algumas falhas que podem ser críticas para a disponibilidade como, por exemplo, single point of failure (ALHARBI, VAN MOORSEL, 2017). Se há uma falha no TTP, nenhuma operação será feita em cima dos dados. Tendo isso em vista, o uso de blockchain combinado com Smart Contracts prevê esse tipo de falha,

utilizando dos Smart Contracts como forma de reforçar a confiança entre os participantes. Por não ter uma autoridade central e muito menos um TTP, todos os participantes da blockchain fazem as verificações necessárias que um TTP faria e, por ser distribuída, não é possível ocorrer single point of failure.

A proposta deste trabalho consiste no desenvolvimento de uma rede de blockchain combinada com Smart Contracts para manter dados confidenciais, íntegros e disponíveis. Nesta rede serão possíveis duas ações, a de armazenar um dado e a de visualizar dados já armazenados. Estes dados armazenados na rede de blockchain devem a todo momento estar disponíveis, íntegros e confidenciais para o usuário.

2. Levantamento bibliográfico

Para a elaboração do presente trabalho foi necessário um levantamento do estado da arte do uso das tecnologias de blockchain com o contexto de preservação de dados, além da tecnologia Hyperledger Fabric, escolhida para o desenvolvimento da proposta do trabalho. A Tabela 1 apresenta palavras-chave pesquisadas no Google Scholar e o total de aparições em artigos e trabalhos. Os dados da Tabela 1 mostraram se as palavras-chave, relacionadas ao tema, são relevantes ou não no mundo acadêmico.

Tabela 1. Revisão Bibliográfica Sistemática

Palavra-Chave	Total
“Data preservation”	3.920.00
“Smart contract”	1.555.000
“Blockchain”	279.000
“Hyperledger Fabric”	10.600
“Blockchain”, “Smart contract”	39.700
“Blockchain”, “Data preservation”	700
“Blockchain”, “Data preservation”, “Smart contract”	324
“Hyperledger Fabric”, “Data preservation”	142

“Hyperledger Fabric”, “Data preservation”, “Smart contract”	118

A partir dos dados expostos na Tabela 1, pode-se afirmar que a preservação de dados é um tema de extrema relevância, com 3.9 milhões de resultados. Além disso, também pode-se verificar uma grande preocupação com smart contracts e blockchain, possuindo, respectivamente, 1.5 milhões e 279 mil resultados. Porém, ao combinar estes três temas, os resultados baixam drasticamente, para apenas 324 e, ao especificar o tipo de blockchain para Hyperledger Fabric, os resultados demonstram-se ainda menores, apenas 118, mostrando uma boa oportunidade para o desenvolvimento de trabalhos acadêmicos neste nicho.

3. Proposta de arquitetura para preservação de dados

Neste capítulo serão descritos os detalhes da proposta de coleta, preservação e visualização de evidências digitais com o uso de blockchain. Será apresentado um modelo de arquitetura que terá dois cenários possíveis de uso, que irão se complementam para a solução proposta, que serão reproduzidas com a utilização das ferramentas do Hyperledger Fabric.

A Figura 1 ilustra a arquitetura proposta:

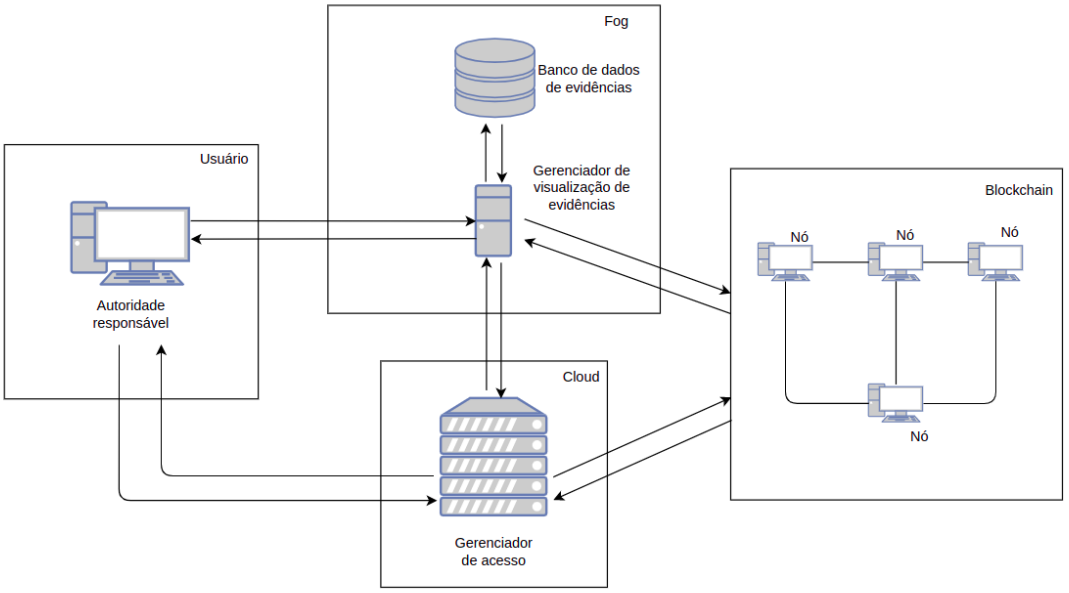

Figura 1: Arquitetura de preservação de evidências digitais

A arquitetura conta com os seguintes componentes:

- Autoridade responsável, representando os membros autorizados a participarem da rede blockchain;
- Gerenciador de acesso, controlando o acesso às informações na blockchain e direcionando as requisições do banco de dados ao Gerenciador de visualização de evidências;
- Gerenciador de coleta e visualização de evidências, realizando a coleta de novos arquivos, controlando o acesso aos arquivos salvos no banco de dados de evidências e fazendo a requisição de adição de um novo bloco de arquivo adicionado à blockchain;
- Banco de dados de evidências, contendo os arquivos criptografados a serem guardados;
- Blockchain, contendo as informações sobre as transações sobre os arquivos a serem salvos no banco de dados.

Sobre a relação dos componentes com os pilares da segurança da informação, o Gerenciador de acesso e o Gerenciador de coleta e visualização de evidências devem permitir que apenas usuários com as devidas permissões possam acessar as informações na blockchain e os arquivos no banco de dados, garantindo a confidencialidade e integridade na arquitetura, já que usuários sem permissão não conseguiriam fazer operações de visualização nem edição de informações. Outro componente que ajuda na integridade das informações é a propriedade de imutabilidade dos blocos de informação da blockchain, impedindo que dados já inseridos na blockchain possam ser alterados. Falando sobre a disponibilidade, a característica que ajuda a garantir esse pilar da segurança da informação é o fato de parte da arquitetura ser disponibilizada através da computação em nuvem, além da rede blockchain ser distribuída.

A arquitetura proposta, ilustrada na Figura 1, conta ainda com 4 camadas:

- Blockchain;
- Fog;
- Cloud;
- Usuário.

A camada Blockchain é composta pelos nós de mineração, que são máquinas distribuídas que se conectam à rede para a garantia da disponibilidade e segurança dos dados recebidos. Pelo uso da blockchain é garantido que os dados serão imutáveis e distribuídos.

A camada de Fog é composta por um componente atuando como gerenciador de coleta e visualização de evidências e pelo banco de dados de evidências. O gerenciador de coleta e visualização de evidências realiza o gerenciamento da preservação de dados que foram gerados pela camada de sensores e atuadores do sistema, deixando-os também disponíveis para a visualização das evidências, quando solicitado pelo gerenciador de acesso presente na camada de Cloud. Ela tem o objetivo de gerenciar a coleta e a visualização das evidências através de seu armazenamento local.

A camada de usuário é composta por um usuário que pode se conectar e solicitar ao gerenciador de acesso o devido acesso aos seus dados armazenados. Para que o

acesso seja concedido, o usuário deve fornecer ao gerenciador de acesso dados que comprovem sua legitimidade perante ao acesso, como um token de acesso que é gerado no login do usuário.

A camada cloud é composta somente por um gerenciador de acesso. Este componente tem o objetivo de atuar como uma porta de entrada ao serviço, ele ficará responsável por se comunicar com todas as outras 3 camadas da arquitetura, tanto enviando quanto recebendo requisições.

Sobre os cenários de uso da arquitetura citados, um deles fará o papel da coleta de evidências digitais e o outro será responsável pela visualização de evidências digitais.

3.1 Cenário de coleta de evidências digitais

Neste primeiro cenário temos como objetivo a coleta e a preservação de evidências vindas do envio manual do usuário dentro da arquitetura. A Figura 2 ilustra o processo de operações que acontecem neste cenário de uso.

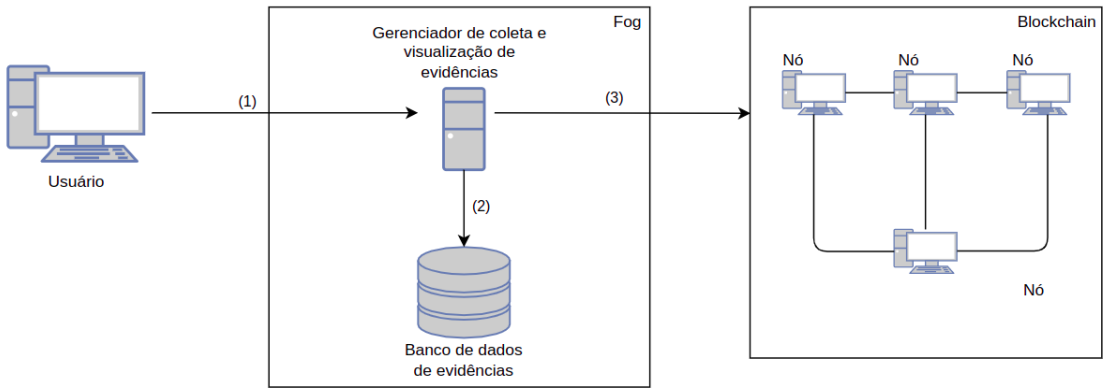

Figura 2: Cenário de coleta de evidências

Dando início ao processo de envio de dados, um usuário devidamente autenticado faz o envio da informação a ser preservada para o Gerenciador de coleta e visualização de evidências.

Em seguida, o gerenciador de coleta e visualização de evidências criptografa o arquivo enviado e armazena-o no Banco de Dados de Evidências. Após o arquivo ser salvo no banco de dados, é criado e adicionado um novo bloco de informação na blockchain. Esse novo bloco contém um ID único da transação de adição do bloco à blockchain, o nome do arquivo salvo, um resumo criptográfico do conteúdo do arquivo, marca temporal da transação e o usuário dono do arquivo. Com isso, a blockchain adiciona esse novo bloco de informação à rede de blockchain.

3.2 Cenário de visualização de evidências digitais

Neste segundo cenário temos como objetivo a visualização das evidências salvas no banco de dados e a visualização das transações realizadas na arquitetura. A Figura 3 ilustra o processo de operações que acontecem neste cenário de uso.

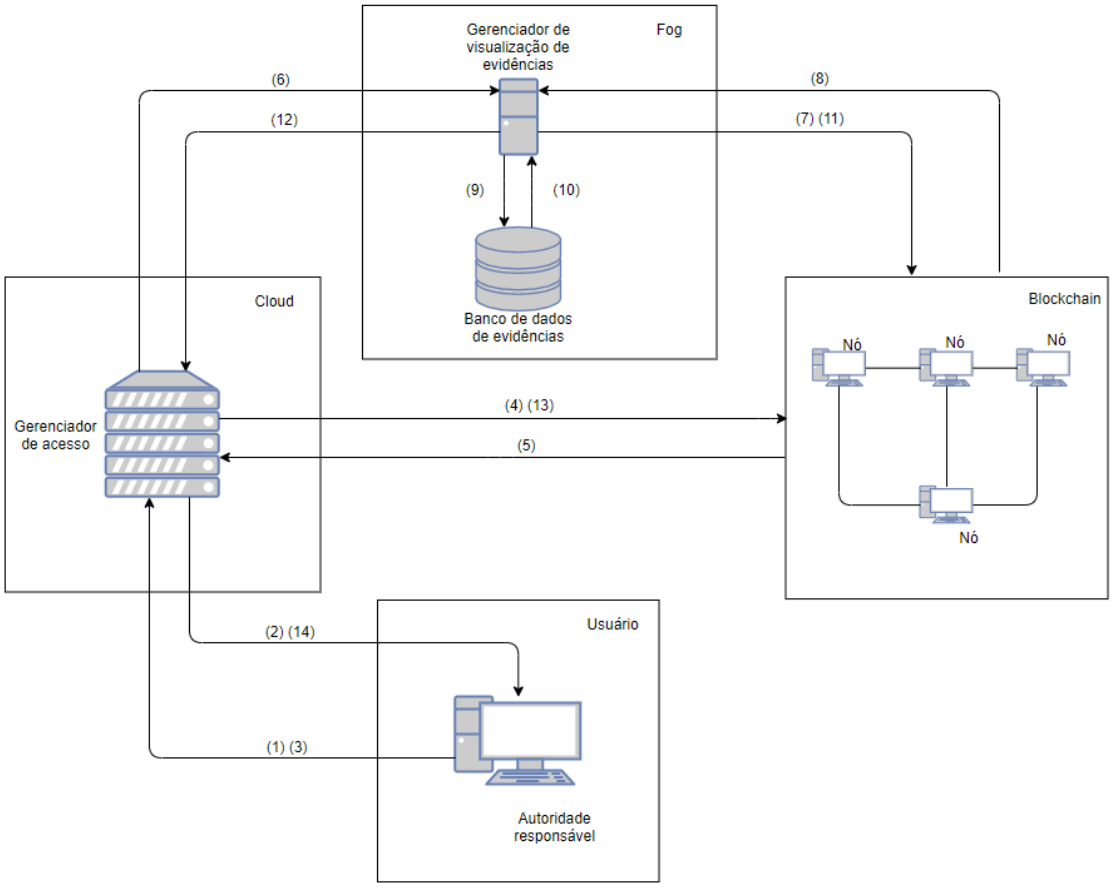

Figura 3: Cenário de visualização de evidências

Conforme a Figura 3, são necessárias 14 operações para que o usuário tenha acesso aos dados armazenados, são elas:

1. A primeira operação é onde o usuário requer acesso aos dados, no qual ele faz uma requisição ao gerenciador de acesso;
2. A segunda operação o gerenciador de acesso solicita identificação ao usuário via token de acesso;
3. A terceira operação o usuário solicita acesso aos dados armazenados e envia seu token de acesso ao gerenciador de acesso;

4. A quarta operação ocorre quando um certificado digital é fornecido, o qual será utilizado pelo gerenciador de acesso para verificar quais dados estão vinculados ao portador deste certificado;
5. A quinta operação é o retorno da requisição feita à Blockchain, ao gerenciador de acesso;
6. A sexta operação conta com o gerenciador de acesso enviando ao GVE uma requisição de acesso aos dados determinado na terceira operação, juntamente com o seu token de acesso;
7. A sétima operação conta com um gerenciador de acesso enviando uma requisição de acesso para o GVE. É feita uma verificação via smart contract da validade do certificado digital e quais as permissões de acesso que ele garante;
8. A oitava operação é o retorno da Blockchain ao GVE, apresentando informações sobre dados os quais o usuário tem acesso;
9. A nona operação verifica a possibilidade do usuário ter acesso aos dados. Caso tenha, será feita uma busca via ID no banco de dados;
10. A décima operação conta com banco de dados enviando os dados armazenados ao GVE;
11. Na décima primeira operação o GVE realiza uma verificação de integridade dos dados, os quais são enviados à Blockchain. A verificação é feita a partir de uma comparação de "Hashes" dos dados presentes na Blockchain e da aplicação do algoritmo SHA-3, 256bits, aos dados fornecidos pelo banco de dados. Finalmente, é feito o envio à Blockchain algumas informações, são elas:
    - "Hash válido" : variável booleana (aceita apenas os valores verdadeiro ou falso);
    - "Intermediadores": consta o local em que estava armazenado e a identificação do GVE;
    - "Responsável": identificação do usuário que solicitou acesso;
    - "Timestamp da transação": momento em que a transação é processada pelo GPE antes do envio à Blockchain.
12. Na décima segunda operação o GVE envia os dados, compactados e criptografados, ao gerenciador de acesso;

13. Na décima terceira operação é feito o envio à Blockchain as informações: “intermediadores”, “responsável” e “timestamp da transação”;
14. Na décima quarta, e final, operação, o gerenciador de acesso encaminha os dados cifrados ao usuário.

4 Conclusão

Neste trabalho foi proposto uma arquitetura de preservação de evidências digitais com o uso de blockchain. Para isso, foi desenvolvida uma API para intermediar a comunicação do usuário com a rede de blockchain. A API faz o papel de gerenciador de acessos às operações da blockchain e de gerenciador de coleta e visualização de evidências no banco de dados. Também foi desenvolvida a rede de blockchain permissionada baseada no Hyperledger Fabric, uma organização de desenvolvimento de tecnologia de blockchain open source.

Com tal implementação foi possível fazer uma análise sobre a viabilidade da arquitetura proposta com o objetivo de coleta e visualização de evidências digitais. É possível ver com o funcionamento da arquitetura proposta que ela é eficiente no que propõe sobre a preservação de dados. Ele guarda conteúdo de arquivos de maneira criptografada, sendo possível sua decriptografia apenas quando requisitada por usuários devidamente autorizados. Além disso, a blockchain reúne informações sobre todas as operações feitas envolvendo a coleta de evidências digitais, fator de extrema importância em termos de auditoria para organizações. Finalmente, podemos afirmar que o software desenvolvido respeitou os três pilares da segurança da informação com êxito, utilizando uma rede de blockchain permissionada.

Referências

Alharby, Maher; Van Moorsel, Aad. Blockchain Based Smart Contracts: a systematic mapping study. Computer Science & Information Technology (Cs & It), [S.L.], v. 1, n. 1, p. 125-140, 26 ago. 2017. Academy & Industry Research Collaboration Center (AIRCC). http://dx.doi.org/10.5121/csit.2017.71011. Acesso em: 27 abr. 2021.

Androulaki, Elli et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the thirteenth EuroSys conference. 2018.

Casino, Fran; Dasaklis, Thomas K.; Patsakis, Constantinos. A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and informatics, 2019.

Stallings, William. Network Security Essentials, 4 Edição, 2011, Prentice Hall.

Rad, Babak B., Bhatti Harrison J., Ahmadi Mohammad. An Introduction to Docker and Analysis of its Performance. Asia Pacific University of Technology and Innovation Technology Park Malaysia, Kuala Lumpur, Malaysia. IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.3, March 2017.

Szabo, Nick. Smart contracts. 1994.

Singh, Karanpreet , Singh, Paramvir, Kumar, Krishan. Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges. Computers & Security, Volume 65, 2017, Pages 344-372.

Jánoky LV, Levendovszky J, Ekler P. An analysis on the revoking mechanisms for JSON Web Tokens. International Journal of Distributed Sensor Networks. September 2018.