UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Helena Fernanda Kray

O USO DE BLOCKCHAIN PARA AUDITORIA DE DADOS: UM ESTUDO DE CASO EM CLOUD COMPUTING

Santa Maria, RS
2023

Helena Fernanda Kray

O USO DE BLOCKCHAIN PARA AUDITORIA DE DADOS:
UM ESTUDO DE CASO EM CLOUD COMPUTING

Monografia apresentada ao Curso de Graduação em Engenharia de Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Graduado em Engenharia de Computação** . Defesa realizada por videoconferência.

Orientadora: Prof.ª Giliane Bernardi

Santa Maria, RS
2023

Helena Fernanda Kray

O USO DE BLOCKCHAIN PARA AUDITORIA DE DADOS:
UM ESTUDO DE CASO EM CLOUD COMPUTING

Monografia apresentada ao Curso de Graduação em Engenharia de Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Graduado em Engenharia de Computação** .

Aprovado em 3 de agosto de 2023:

_____

Giliane Bernardi, Dra. (UFSM)
(Presidenta/Orientadora)

_____

Andre Zanki Cordenonsi, Dr. (UFSM)

_____

José Eduardo Baggio, Dr. (UFSM)

Santa Maria, RS
2023

DEDICATÓRIA

Dedico esse trabalho a toda minha família, sobretudo a minha mãe e irmã, por sempre acreditarem no meu potencial, mesmo nos momentos em que eu já não acreditava em mim mesma durante essa caminhada.

AGRADECIMENTOS

Agradeço ao meu pai e cunhado, por servirem de inspiração na escolha da minha carreira e sempre aconselharem quando necessário.

A professora Giliane, por ter sido minha orientadora e ter me guiado durante o desenvolvimento desse trabalho.

A todos que contribuíram, de alguma forma, durante esse semestre, mesmo que com palavras de apoio e incentivo.

Só se pode alcançar um grande êxito quando nos mantemos fiéis a nós mesmos.

(Friedrich Nietzsche)

RESUMO

O USO DE BLOCKCHAIN PARA AUDITORIA DE DADOS:
UM ESTUDO DE CASO EM CLOUD COMPUTING

AUTORA: Helena Fernanda Kray
Orientadora: Giliane Bernardi

O presente trabalho tem como objetivo analisar as aplicações da tecnologia Blockchain dentro de Cloud Computing (computação em nuvem). A pesquisa partiu de uma revisão teórica, contextualizando os principais conceitos e pilares envolvendo sistemas Cloud e a tecnologia Blockchain, além de trazer outros trabalhos correlatos desta integração, que serviram de ponto de partida para a idealização desta monografia. No decorrer do estudo, foram identificados os principais desafios enfrentados pelos sistemas Cloud, como por exemplo questões de segurança envolvendo violação de dados, acessos indevidos e perdas permanentes de dados. A partir disso, foi possível traçar quais características da Blockchain poderiam suprir esta demanda, como os níveis de autorizações definidos para cada usuário e um livro-razão contendo todo histórico dos dados. Os resultados demonstraram que a tecnologia da Blockchain tem relevância no meio Cloud, podendo servir como forma de auditoria em casos de desconfiança da nuvem, garantindo que clientes e usuários tenham acesso a um registro completo e imutável de todas inclusões e alterações de dados. Assim, é esperado que este trabalho possa contribuir em futuros estudos e desenvolvimentos relacionados a integração destas duas tecnologias emergentes, atraindo o interesse de profissionais da área.

Palavras-chave: Computação em Nuvem. Blockchain. Segurança. Auditoria de Dados. Compartilhamento de Dados

ABSTRACT

THE USAGE OF BLOCKCHAIN FOR DATA AUDITING:
A CASE STUDY IN CLOUD COMPUTING

AUTHOR: Helena Fernanda Kray
ADVISOR: Giliane Bernardi

This work explores the applications of Blockchain technology within Cloud Computing context. The research started with a literature review, contextualizing the main concepts and characteristics involving Cloud Computing and Blockchain technology, in addition to bring related works to this integration, that served as a starting point for the idealization of this monograph. During the study, the main challenges faced by Cloud systems were identified, such as security issues involving data violation, unauthorized access and permanent data loss. Moreover, it was possible to trace which Blockchain characteristics could meet this demand, such as the authorization levels defined for each user and a ledger containing all data history. The results showed that Blockchain technology is indeed relevant in the Cloud context, and can be useful as a means of audit in cases of cloud mistrust, ensuring that customers and users have access to a complete and immutable record of all data inclusions and changes. Therefore, it is expected that this work can contribute to future studies and developments related to the integration of these two emerging technologies, bringing the interest of professionals in the area.

Keywords: Cloud Computing. Blockchain. Security. Data Audit. Data Sharing

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Com o avanço da tecnologia, a cada dia mais vemos a necessidade de sistemas de fácil acesso que possibilitam o gerenciamento de dados a qualquer momento. O imediatismo do Sistema Cloud, ou seja, a possibilidade de utilizar um sistema a qualquer momento simplesmente por estar conectado à internet, popularizou esta forma de gerenciamento de informações. Sua principal vantagem é que, por estar rodando em sistemas externos, os usuários não necessitam de poder computacional On Premise (onde o software roda no servidor local) para fazer uso dos serviços. Do Google Drive à Netflix, o Cloud vem cada vez mais tomando seu espaço no dia a dia de pessoas dentro e fora da Tecnologia da Informação (TI), mas, apesar de ser uma ideia que, à primeira vista, parece fascinante, atualmente grandes empresas de nuvem enfrentam desafios para conquistar o público. O motivo disso se deve, principalmente, pela falta de visibilidade e controle dos dados, assim como apresentado no Relatório de Segurança em Nuvem publicado pela Fortinet (2022), empresa de serviços de cibersegurança.

Em contrapartida, temos uma outra tecnologia conhecida por sua segurança e privacidade, a Blockchain (CHICARINO et al., 2017). Esta é um bloco de dados compartilhado que faz um rastreamento completo e imutável de todos os dados presentes nos blocos. Isso acontece pois, com a sua forma de sistema distribuído, onde cada servidor conectado na rede (chamado de nós) contém uma cópia do arranjo, torna-se impossível fazer alguma alteração neste banco sem modificar todos os outros nós. Sempre que alguma informação precise ser alterada ou adicionada, um novo bloco de dados é encadeado nos nós, assim, o histórico nunca é perdido e não pode ser adulterado. Cada bloco é selado e recebe um identificador criptográfico do tipo *hash*, uma função que transforma uma determinada entrada varável em uma saída padronizada. Além da criptografia, o algoritmo *hash* gera uma sequência praticamente exclusiva, onde um único caractér diferente na entrada geraria uma saída completamente distinta, garantindo a integridade do bloco. Para tornar possível uma alteração corruptiva, seria necessário modificar os dados de todos os nós e ainda alterar o *hash*.

Neste trabalho, será realizada a criação de uma Blockchain e uma análise de como esta tecnologia pode ser utilizada como forma de auditoria de dados em um sistema Cloud. O objetivo é estudar possíveis aplicabilidades da Blockchain dentro de um cenário de banco de dados em Cloud, como inclusão de informações de diferentes nós, alterações e verificações. É esperado que este trabalho possa contribuir para trazer notoriedade a interessados da área, e, assim, aumentar o desenvolvimento de estudos e aplicações na vida real dessas tecnologias emergentes.

2 REFERENCIAL TEÓRICO

O presente capítulo têm como objetivo apresentar uma revisão dos principais conceitos, desafios e aplicações das tecnologias Cloud e Blockchain, e como uma integração entre as mesmas pode trazer benefícios nos tempos atuais.

2.1 CLOUD COMPUTING

Quando falamos em Cloud Computing (computação em nuvem no português), normalmente pensamos em uma tecnologia que evoluiu muito nos últimos anos, e vem sendo cada vez mais presente no dia a dia de grandes empresas e usuários finais. Apesar disso, a melhor definição deste termo continua sendo a da NIST (*National Institute of Standards and Technology*), em 2011:

> A computação em nuvem é um modelo para permitir o acesso onipresente, conveniente e sob demanda a uma rede compartilhada de recursos de computação configuráveis (por exemplo, redes, servidores, armazenamento, aplicativos e serviços) que podem ser provisionados e liberados rapidamente com o mínimo de esforço de gerenciamento ou interação com o provedor de serviços.

Ainda segundo NIST (2011), podemos apresentar esse tipo de sistema em três partes, sendo elas: características essenciais, modelo de serviço e modelo de implantação. Dentre as características essenciais, temos:

- *On-demand self-service* (Autosserviço por demanda): o consumidor pode fazer uso dos recursos computacionais sem precisar de uma interação humana para realizá-lo;
- *Broad network acces* (Acesso amplo à rede): disponibilidade de realizar o acesso dos recursos por meio de diferentes plataformas (por exemplo, computadores, smartphones e/ou tablets);
- *Resource pooling* (Agrupamento de recursos): os recursos computacionais do provedor são agrupados de forma a servir diversos consumidores com um modelo *multi-tenant* (uma instalação que atende múltiplos clientes simultaneamente). Além disso, a localização das instalações é independente, fazendo com que os consumidores não tenham conhecimento sobre a localização exata do provedor, e, sim, apenas informações gerais, como país ou região. Alguns exemplos de recursos são: armazenamento, processamento, memória e banda larga;

- *Rapid elasticity* (Elasticidade rápida): as capacidades podem ser instantaneamente providas e lançadas, de acordo com a demanda, fazendo com que o cliente final sinta que possui recursos ilimitados a qualquer momento e em qualquer quantidade; e

- *Measured service* (Serviços mensurados): os uso dos recursos são automaticamente controlados e otimizados, utilizando uma medição em algum nível de abstração apropriado para o tipo de serviço (por exemplo, armazenamento, processamento e banda larga). Esse uso é monitorado e reportado, de forma a garantir transparência para o provedor e consumidor.

Em questão de modelos de serviço, temos as três opções a seguir (NIST, 2011):

- *Software as a Service* (Software como serviço): as aplicações são providas para o consumidor por meio de uma infraestrutura Cloud. Essas aplicações podem ser acessadas pelos clientes por um navegador ou programa de interface. O cliente não possui controle sobre a infraestrutura, sendo esta gerenciada e mantida pelo provedor, assim como o sistema operacional, servidores e a aplicação em si;

- *Platform as a Service* (Plataforma como serviço): o consumidor faz uso da infraestrutura Cloud, porém mantém suas próprias aplicações criadas com linguagens de programação e bibliotecas disponibilizadas pelo provedor. Neste caso, o cliente também não possui acesso a infraestrutura, sistema operacional e servidores, porém, têm controle sobre as aplicações; e

- *Infrastructure as a Service* (Infraestrutura como serviço): neste modelo, o provedor apenas disponibiliza a infraestrutura Cloud, sendo o consumidor responsável por criar e manter seu próprio software e recursos. O cliente não possui controle sobre a infraestrutura, mas gerencia o software, sistema operacional e armazenamento.

Por fim, pode ser escolhido um modelo de implantação dentre as quatro possibilidades a seguir (NIST, 2011):

- *Private Cloud*: a infraestrutura Cloud é provida exclusivamente para um cliente ou organização. Assim, a manutenção e gerenciamento do mesmo pode ficar a cargo do próprio consumidor ou de terceiros;

- *Community Cloud*: a infraestrutura Cloud é provida para uma comunidade de consumidores de uma organização que possuem interesses em comum. Assim como no modelo anterior, também pode ser gerenciada e mantida pela organização ou terceiras partes.

- *Public Cloud*: a infraestrutura Cloud é provida para o público em geral. Pode ser gerenciada e mantida pela empresa, organização governamental ou universidade; e

- *Hybrid Cloud*: a infraestrutura Cloud é composta por dois ou três modelos de implantação (*Private*, *Community* ou *Public Cloud*), vinculados por meio de uma tecnologia padronizada que permite a portabilidade dos dados.

Desta forma, os clientes interessados em contratarem produtos Cloud possuem uma vasta gama de possibilidades quando vão assinar estes serviços, podendo selecionar a opção que mais encaixa em suas necessidades de negócio.

2.1.1 As vantagens dos Sistemas Cloud Computing

Além de trazer um novo conceito para o mundo da tecnologia, o modelo de computação em nuvem vêm chamando muito a atenção do público pela sua praticidade e relação custo-benefício. Algumas de suas principais vantagens, segundo Silva et al. (2020), podem ser destacadas:

- Menor custo: ao contrário da computação convencional, o modelo Cloud funciona como um serviço utilitário, por meio de assinaturas. Sendo assim, como todos recursos necessários ficam a encargo do provedor, os clientes precisam apenas organizar seus sistemas para começar a acessar o serviço;
- Custo operacional reduzido: uma vez que o provedor é responsável pela infraestrutura, os assinantes não precisam se preocupar com custos operacionais relacionados à manutenção do sistema, energia 24x7 (ou seja, garantir que o sistema esteja conectado 24 horas por dia, 7 dias na semana), administração e suporte de refrigeração das instalações;
- Responsabilidade de gerenciamento de sistema reduzida: ao contratar serviços Cloud, o usuário consegue evitar o trabalho extra de configurar e manter seu sistema, já que estes gerenciamentos são providos pelo fornecedor;
- Poder de computação e armazenamento ilimitados: os usuários conseguem ter acesso a um poder computacional muito superior, uma vez que estas super máquinas são adquiridas pelo provedor e disponibilizadas a um custo razoável;
- Disponibilidade: os grandes provedores de computação em nuvem prometem disponibilidade perto de 24x7;
- Acesso independente do lugar: o Cloud está disponível aos seus usuários independente de sua localização geográfica ou horário, bastando ter um dispositivo computacional e acesso a internet;

- Implantação: como a oferta de sistemas Cloud é imediata e automática, a rápida implantação se torna um grande atrativo quando comparado à sistemas tradicionais, em que os serviços podem levar uma grande quantidade de tempo para se tornarem produtivos;
- Atualização automática de software: dentro do contexto de computação em nuvem, as atualizações são feitas automaticamente pelo fornecedor, evitando o trabalho extra de executar correções periódicas; e,
- Amigo do meio ambiente: uma vez que os recursos computacionais são compartilhados com múltiplos usuários, o Cloud promove a computação verde, minimizando a geração de lixo eletrônico.

É importante frisar, também, que normalmente esse modelo de computação em nuvem é oferecido contratualmente junto a acordos de nível de serviço (*Service Level Agreements* - SLA), onde são especificados alguns pontos a serem cumpridos pela provedora, como por exemplo tempos de entrega e requisitos de desempenho.

2.1.2 Os Desafios do Cloud Computing

Apesar do mercado da computação em nuvem estar avaliado este ano em USD 483.98 bilhões, e uma estimativa de crescer cerca de 14.1% de 2023 até 2030 (Grand View Research, 2022), este modelo ainda enfrenta grandes desafios para ganhar a total confiança do público. Segundo uma pesquisa conduzida pela Nutanix (2023), empresa americana especializada em produtos em nuvem, 49% dos consumidores da região das Américas alega que sua principal preocupação com a aderência do Cloud é relacionada à segurança de seus dados. Explorando mais este contraponto, a seguir estão alguns dos desafios e problemas de segurança em Cloud, de acordo com Hiran et al. (2019):

- Violação de Dados: chamamos de violação de dados quando uma pessoa externa consegue acesso aos dados mantidos em uma infraestrutura Cloud por meio de ações fraudulentas, fazendo uso da mesma sem autorização das partes envolvidas contratualmente;
- Localização dos Dados: ao contrário de instalações On Premise, os consumidores Cloud não recebem informações relacionadas a localização de onde seus dados estão sendo mantidos;
- Acesso: os clientes Cloud podem acessar seus serviços a qualquer momento, porém, esta facilidade pode ocasionar problemas de acesso por usuários não autorizados;

- Vulnerabilidades do Sistema: possíveis bugs do sistema podem abrir portas para ataques cibernéticos. Atualizações devem ser feitas periodicamente afim de evitar isso;
- *Account Hijacking*: é um ataque cibernético onde uma parte maliciosa tenta se passar por uma fonte confiável a fim de roubar credenciais e identidades. Esta é uma preocupação relevante no conceito Cloud, uma vez que o acesso aos serviços é normalmente provido por meio de login e senha; e
- Perda permanente de Dados: possíveis desastres naturais, panes no sistema ou exclusões não intencionais podem ocasionar uma perda permanente de dados.

Uma possível solução para os problemas de violação de dados, acesso irrestrito e perda permanente de dados seria o uso da tecnologia Blockchain, que possui funcionalidades como configuração de permissão de usuários e um histórico completo e imutável de todos dados.

2.2 BLOCKCHAIN

Em 2008, um autor anônimo, conhecido pelo seu pseudônimo Satoshi Nakamoto, escreveu o *White Paper* intitulado 'Bitcoin: A peer-to-peer electronic cash system' e revolucionou o conceito de transações monetárias e dinheiro com a criptomoeda Bitcoin (NAKAMOTO, 2008). A ideia introduzida por Nakamoto (2008) foi a de criar uma moeda virtual utilizando o modelo *Peer-to-Peer*, onde as transações financeiras não precisam passar por uma terceira parte (instituição financeira), garantindo descentralização e independência (ULRICH, 2014).

Mas como realizar isso na prática? A resposta está na tecnologia Blockchain, que, apesar de ter recebido notoriedade com o Bitcoin, surgiu muito antes disso. Bayer, Haber e Stornetta (1992) trouxeram uma proposta de uma corrente de blocos com uma forte criptografia que guardava informações de data e hora de documentos. A ideia dessa implementação era que não fosse possível alterar ou violar estes dados sem comprometer toda cadeia de blocos e, assim, garantir a segurança dos dados (NASCIMENTO et al., 2022).

> A ideia do Blockchain é que o usuário inicie sua transação usando sua assinatura digital. Em seguida, usuários transmitem essa transação para os nós. Um ou mais nós começam a validar a transação. Após a transação ser validada, ela é registrada em um bloco. E esses blocos são constantemente enviados pela rede aos outros nós encadeando com outros blocos já existentes, pelo protocolo do Blockchain (MORAES, 2021).

Assim, conforme Nakamoto (2008) idealizou, a Blockchain seria uma espécie de livro-razão distribuído, que contêm todas informações registradas sobre as transações

ocorridas na rede. Para garantir a consistência e autenticidade das transações, cada usuário possui uma chave privada e uma pública. Nesse caso, um indivíduo A cria uma transação com a chave pública do indivíduo B (que irá receber), assinada com sua chave privada. Desta forma, ao checar o livro-razão e o nó, é possível verificar que a chave pública do indivíduo A realizou uma transação para o indivíduo B e o mesmo foi autenticado (ULRICH, 2014).

Dentre as características desta tecnologia, Lucena (2016) dividiu a Blockchain em 5 principais pilares:

- Funções de mão única: é uma função onde é possível apenas realizar operações em um sentido, não sendo possível aplicar uma engenharia reversa para voltar ao arquivo original. Tipicamente, nas tecnologias Blockchain, é utilizada a função Hash para este fim, onde uma entrada variável é mapeada para uma sequência de valores de comprimento fixo. Este algoritmo funciona de uma forma que uma diferença de 1 bit na entrada gere uma saída completamente diferente;
- *Timestamp* (registro do momento da criação/alteração do arquivo): com o objetivo de impedir alguma possível fraude temporal, o timestamp guarda a informação do exato momento em que qualquer arquivo foi criado ou alterado;
- Assinatura digital do autor da alteração no arquivo: tem por finalidade garantir que qualquer alteração no nó foi de fato realizada pelo dono do nó, por meio da verificação das chaves públicas e privadas;
- Rede descentralizada *peer-to-peer*: com este modelo, cada alteração (ou criação) na Blockchain pode ser verificada e, assim, aceita ou rejeitada pela maioria dos *peers*, impossibilitando atividades fraudalentas na rede; e
- Mecanismo que gera o novo bloco encadeado da Blockchain: por se tratar de um bloco em cadeia, toda vez que alguma alteração é feita, um novo bloco é criado e vinculado a rede da Blockchain, guardando o histórico completo do nó.

Assim, apesar da Blockchain estar amplamente vinculada ao Bitcoin, seus conceitos fundamentais podem ser explorados em diversos contextos além das criptomoedas, o que busca-se explorar na próxima seção.

2.2.1 Aplicações da Blockchain

Como já citado anteriormente, a Blockchain veio como uma facilitadora que tornou possível o uso e criação do Bitcoin. Porém, esta tecnologia vem sendo estudada com diversas aplicações no mundo da TI. Segundo Simão, Silva e Paiva (2018), no contexto

de educação, a blockchain poderia prover uma grande contribuição quando utilizada para guardar históricos escolares e certificados de conclusão de curso. Além desta tecnologia ajudar com a diminuição do uso de papeis, sua aplicação possui grande relevância para evitar fraudes destes documentos e permitir a validação dos mesmos em diferentes instituições.

Um outro eixo onde a Blockchain ganhou alta notoriedade foi no âmbito judicial, principalmente com os chamados *Smart Contracts* (contratos inteligentes, em português). Esta forma de uso permitiria que certidões de nascimento, vendas de propriedades e outros registros legais ficassem guardados em uma Blockchain sem a possibilidade de alterações e atividades fraudulentas (MARCHSIN, 2022).

Por fim, conforme será explorado neste trabalho, a Blockchain pode ser aplicada como uma camada de segurança em um software, uma vez que a IoT (*Internet of Things*, que também inclui *Cloud Computing*) é um meio vulnerável e sujeito a falsificações, deixando uma variedade de possibilidades neste quesito (LAURENCE, 2019).

2.2.2 BlockChain Privada e Pública

Embora a Blockchain seja mais reconhecida por sua propriedade de descentralização, onde as transações são feitas a partir de um protocolo comum por todos usuários da rede, permitindo que qualquer nó registre informações na base de dados, isto é uma característica de uma Blockchain pública, como mencionado por Preukschat et al. (2017).

Porém, esta tecnologia possui um outro modelo que, apesar de não trazer a ideia de uma rede pública e descentralizada, ainda assim possui o mesmo funcionamento de registro de livro-razão, e este seria a Blockchain Privada. Segundo Guegan (2017), uma Blockchain Privada é aquela onde o acesso e as transações só podem ser efetuados por aqueles que têm as autorizações para tal.

Neste trabalho, será abordado o conceito de Blockchain Privada, justamente por estar sendo estudada uma aplicação da mesma no meio Cloud, onde os serviços são comercializados e é necessário que se tenha um controle de acessos. Com isso, os usuários devem ter permissões explícitas para atuarem na Blockchain.

2.3 TRABALHOS CORRELATOS

Gupta et al. (2019) exploraram a usabilidade da Blockchain em Cloud Computing como uma forma de segurança, ao comparar as características da Blockchain com os requisitos de Cloud, como a escalabilidade. Enquanto a Blockchain é escalável justamente pela adição/remoção de novos nós, sistemas Cloud são providos para diversos

usuários/nós que estão usando seus serviços. Durante este estudo, foram analisadas as seguintes três possibilidades de uso de Blockchains em Cloud:

- Registro aberto: os usuários possuem acesso a todos serviços providos pelo sistema Cloud, além de terem acesso aos níveis de segurança e SLAs vendidos pelo fornecedor;
- Registro distribuído: todos os usuários conseguem checar uma cópia do livro-razão. Este, por sua vez, guarda todos serviços utilizados na nuvem; e
- *Smart Contract* descentralizado: neste caso, todos os usuários do modelo Cloud tem uma cópia dos contratos e, quando o pagamento do serviço é concluído, o acesso ao sistema é liberado. Todos os usuários Cloud poderiam verificar e prevenir se alguma das partes quisesse fazer alguma alteração no registro.

Já Dorsala, Sastry e Chapram (2021), apresentaram alguns desafios quando estudada a aplicação conjunta de Cloud e Blockchain. Isso porque, ao passo que o Cloud é um sistema centralizado com baixa transparência, a Blockchain parte justamente do princípio da descentralização. Além disso, os fornecedores de Cloud costumam oferecer a privacidade dos dados por terceiras partes, o que é um problema no contexto de Blockchain, onde as informações estão disponíveis na rede. Por fim, uma das grandes vantagens da nuvem, já abordadas anteriormente, é o acesso imediato aos seus serviços, enquanto na Blockchain a inclusão de novos blocos pode causar certos atrasos temporais.

Considerando os desafios apresentados por Dorsala, Sastry e Chapram (2021), a utilização de uma Blockchain com acesso restrito a determinados usuários, ao invés de nós públicos, poderia mitigar o problema, e esse foi o modelo utilizado no desenvolvimento deste trabalho. Ainda, seguindo a pesquisa de Gupta et al. (2019), no presente trabalho o conceito de registro distribuído foi considerado como o principal para o estudo da tecnologia Blockchain como forma de auditoria em sistemas Cloud. O próximo capítulo apresenta o percurso metodológico seguido para a implementação da Blockchain, foco de estudo esta proposta.

3 ASPECTOS METODOLÓGICOS

Para que fosse possível implementar uma Blockchain e estudar diferentes aplicações no contexto Cloud, o presente trabalho iniciou a partir de uma revisão bibliográfica, detalhada no capítulo 2, para ambientar e definir a linha de pesquisa deste estudo. Com esse intuito, o capítulo 2 foi separado entre a contextualização do Cloud e seus desafios, uma explicação da Blockchain e suas aplicações e, por fim, trabalhos correlatos sobre a integração dessas tecnologias.

Na sequência, partiu-se para o desenvolvimento e preparação da Blockchain, apresentado no capítulo 4. Para isso, foi escolhido fazer uso da plataforma Multichain[1], que permite criar e testar cadeias de dados. A MultiChain possui a versão *Community* (comunidade), que é gratuita com funcionalidades restritas, e a versão *Enterprise* (empresarial), que é paga e possui recursos avançados. Como este é um trabalho acadêmico com objetivo de estudos, a versão *Community* foi selecionada.

O modelo de Blockchain selecionado para este trabalho foi o privado. Como estamos pensando em uma aplicação em Cloud Computing, para armazenamento de dados pessoais e possíveis auditorias, a rede privada possui a grande vantagem de precisar selecionar autorizações explícitas para cada nó operar na sua rede. Em contrapartida, a Blockchain pública permite que qualquer nó da rede altere informações, deixando em aberta a possibilidade de que um único usuário controle diversos nós e corrompa as informações da Blockchain.

Apesar da Multichain ter a possibilidade de rodar nos principais Sistemas Operacionais (Linux, Windows e MacOS), o Linux foi escolhido neste trabalho por se ter mais recursos ao lidar com linhas de comando. Além disso, dois computadores na mesma rede foram utilizados para criar os nós em servidores diferentes.

O desenvolvimento foi dividido em 4 etapas, sendo elas:

- Instalação da plataforma Multichain: o primeiro passo para poder utilizar os recursos da Multichain é instalar a mesma nas máquinas locais. Para isso, foi seguida a documentação[2] disponibilizada pela própria plataforma;

- Preparação das Máquinas: foi necessário, primeiramente, criar um projeto de Blockchain na rede Multichain e então conectar a máquina secundária à ela. Este passo é importante pois é quando os parâmetros da Blockchain são selecionados, além de prover as devidas autorizações de leitura e/ou escrita para cada nó;

- Criação de uma *Stream*: aqui, foi criada uma *stream* (sequência de dados) que conterá todas informações de cadastro de pessoas, como um banco de dados. Também,

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

[2] Disponível em: https://www.multichain.com/download-community/

foi necessário novamente prover e configurar permissões de acesso para a máquina secundária; e

- Inclusão de dados na *Stream*: por fim, a última etapa do desenvolvimento foi alimentar a *stream* já criada com alguns dados fictícios por meio do tipo JSON, que se trata de um formato compacto para troca de dados. As informações contidas nesta etapa serviram de base para o estudo de caso posterior.

Uma vez que a Blockchain já estava funcional e alimentada, foi possível partir para o desenvolvimento de alguns cenários de uso pensando em uma integração com sistemas Cloud, de forma a discutir suas contribuições. Esta etapa foi subdividida em 3 partes, sendo elas:

- Cenário 1 - Alteração de Dados: demonstração de como a Blockchain se comporta quando uma informação precisa ser alterada, e como registra essa ação;
- Cenário 2 - Inclusão de Dados por outro servidor: aqui foi estudado como as informações podem ser adicionadas de outras máquinas e como isso é reconhecido pela Blockchain; e
- Cenário 3 - Verificação da Blockchain: por fim, uma análise foi realizada, buscando discutir como uma auditoria poderia funcionar no contexto Cloud, demonstrando a grande vantagem da integração das duas tecnologias.

O desenvolvimento e discussão dos cenários de uso é apresentado no capítulo 5.

4 DESENVOLVIMENTO

Este capítulo apresenta como a Blockchain foi implementada em dois servidores Linux para estudo posterior. Para isso, foi necessário ambientar, gerar e alimentar uma Blockchain em duas máquinas separadas.

4.1 INSTALAÇÃO DA PLATAFORMA MULTICHAIN

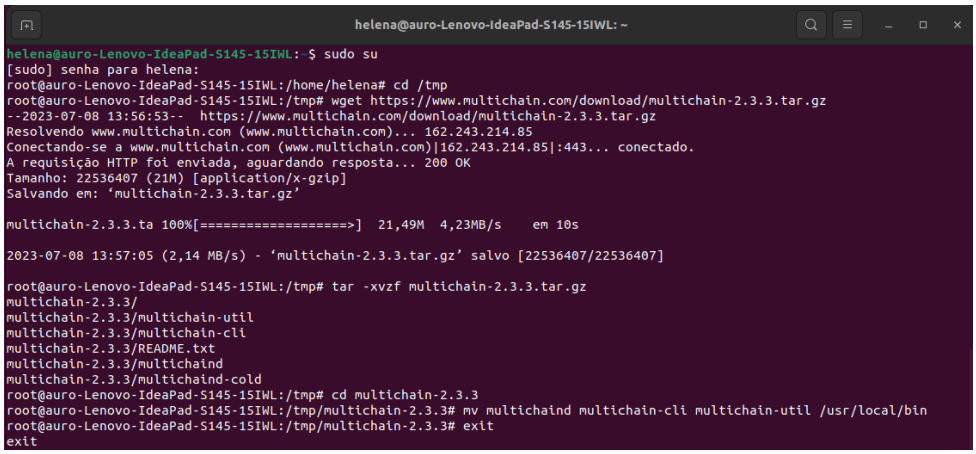
O primeiro passo para construir uma Blockchain é ambientar a máquina local e inclui-la na rede de nós. Assim, foram seguidos os passos descritos na documentação da plataforma MultiChain para preparar o servidor local, conforme Figura 1.

Figura 1 – Passos para instalação da MultiChain.

Fonte: Adaptado de MultiChain (2023)

Já que o sistema operacional utilizado neste trabalho foi o Linux, toda implementação foi feita pelo *Prompt*, por meio de linhas de comando (Figura 2).

Figura 2 – Instalação da rede Blockchain MultiChain no servidor local.



Fonte: Próprio autor.

Após prover a senha do usuário do computador e rodar os comandos necessários, a rede Multichain fica disponível para acesso na máquina.

4.2 PREPARAÇÃO DAS MÁQUINAS

Uma vez que a rede MultiChain está instalada na máquina, foi criada a Blockchain chainProject com o comando *multichain-util create chainProject* no servidor local (Figura 3).

Figura 3 – Criação da Blockchain chainProject.

```
helena@auro-Lenovo-IdeaPad-S145-15IWL:~$ multichain-util create chainProject

MultiChain 2.3.3 Utilities (latest protocol 20013)

Blockchain parameter set was successfully generated.
You can edit it in /home/helena/.multichain/chainProject/params.dat before running multichaind for the first time.

To generate blockchain please run "multichaind chainProject -daemon".
```

Fonte: Próprio autor.

A plataforma MultiChain permite configurar os parâmetros da Blockchain de acordo com as necessidades do usuário. Dentre as possibilidades, estão as restrições de acesso de leitura e escrita nos nós, tamanho máximo de um bloco (em megabytes), nome/descrição do node, entre outras. Para fins de pesquisa, foram mantidos os parâmetros originais [1].

Visto que os parâmetros foram incluídos, foi possível, então, iniciar o *node*, utilizando o comando *multichaind chainProject -daemon* (Figura 4). Uma vez que o *node* foi gerado, somente parâmetros marcados como *Upgradable* podem ser alterados por meio do mecanismo de atualização do MultiChain.

Figura 4 – Inicialização do node.

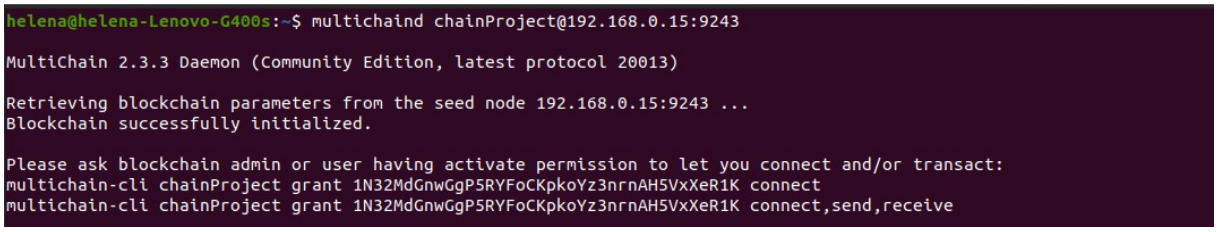
```
helena@auro-Lenovo-IdeaPad-S145-15IWL:~$ multichaind chainProject -daemon

MultiChain 2.3.3 Daemon (Community Edition, latest protocol 20013)

Starting up node...

Looking for genesis block...
Genesis block found

Other nodes can connect to this node using:
multichaind chainProject@192.168.0.15:9243

Listening for API requests on port 9242 (local only - see rpcallowip setting)

Node ready.
```

Fonte: Próprio autor.

Como demonstrado na Figura 4, o *node* chainProject@192.168.0.15:9243 foi criado na rede local, e o mesmo pode ser acessado e conectado por outros *nodes* a partir desta ID. Dado que a tecnologia Blockchain é baseada na ideia *Peer-to-peer*, uma segunda máquina (com diferente servidor e na mesma rede) foi utilizada para conectar no projeto (Figura 5).

[1] Lista disponível em: https://www.multichain.com/developers/blockchain-parameters/

Figura 5 – Conectando a segunda máquina ao chainProject.



Fonte: Próprio autor.

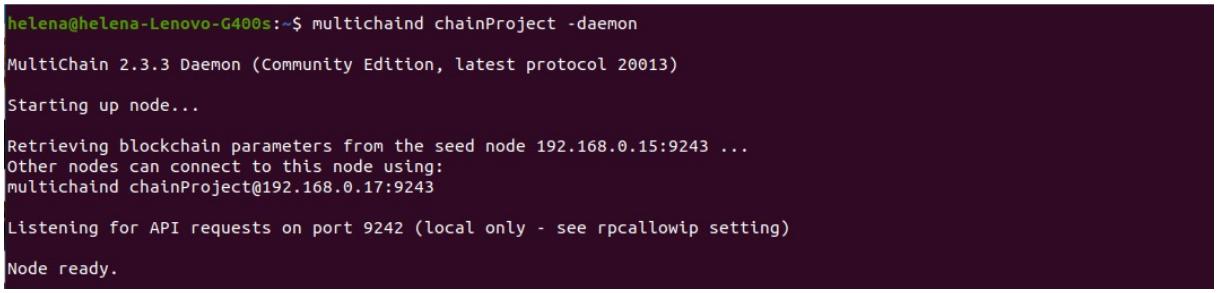
Apesar da Blockchain ter sido inicializada com sucesso e o *address* gerado, o *node* original ainda precisa prover permissão para o servidor secundário (representado pela sequência 1N32MdGnwGgP5RYFoCKpkoYz3nrnAH5VxXeR1K) poder conectar e realizar transações de *send* (envio no português) e *receive* (receber). Para isso, na máquina 1, adicionamos o comando *multichain-cli chainProject grant 1N32MdGnwGgP5RYFoCKpkoY z3nrnAH5VxXeR1K connect,send,receive* (Figura 6).

Figura 6 – Provendo permissões ao segundo servidor.

Fonte: Próprio autor.

Voltando para o segundo servidor, agora é possível criar o outro *node* para se conectar a Blockchain chainProject (Figura 7).

Figura 7 – Iniciando o segundo node no projeto



Fonte: Próprio autor.

A partir deste momento, ambas as máquinas estão plenamente aptas a operar dentro do chainProject, com as permissões necessárias.

4.3 CRIANDO UMA *STREAM*

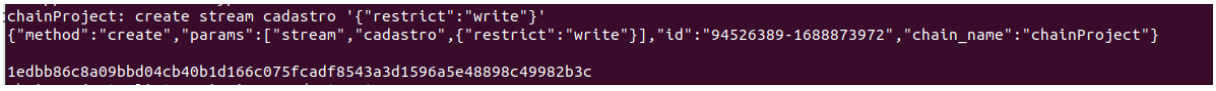
Primeiramente, para aumentar a praticidade, o modo interativo foi ativado com o comando *multichain-cli chainProject* (Figura 8). Desta forma, não é necessário escrever *multichain-cli* toda vez que precisar realizar alguma ação na Blockchain.

Figura 8 – Ativando modo interativo.

Fonte: Próprio autor.

Agora que os dois servidores já estão configurados e habilitados para se conectar e realizar transações na Blockchain chainProject, vamos criar uma *stream* Cadastro para guardar informações de usuários. A restrição *write* significa que somente quem tiver permissões explícitas pode escrever no Cadastro (Figura 9).

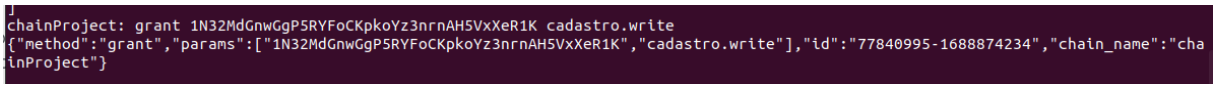
Figura 9 – Criando a *stream* Cadastro.



Fonte: Próprio autor.

Devido à restrição, o servidor secundário precisa receber as autorizações gerais e de escrita na *stream* (Figura 10). O comando *listpermissions cadastro.** pode ser utilizado para verificar todos níveis de permissões de cada *address*.

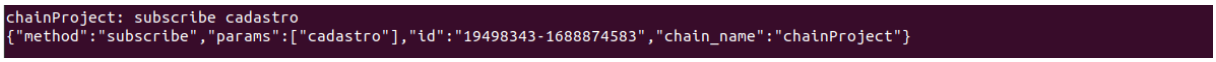
Figura 10 – Autorização geral para *stream* Cadastro.



Fonte: Próprio autor.

Por fim, é necessário habilitar o *node* a começar a acompanhar as atividades da *stream* por meio do comando *subscribe cadastro*. O mesmo deve ser realizado na máquina original e secundária (Figura 11).

Figura 11 – Autorização geral *stream* Cadastro.



Fonte: Próprio autor.

Desta forma, tanto a máquina original quanto a secundária podem, de agora em diante, enviar dados para a *stream* Cadastro. Além disso, a troca de informações acontece em tempo real, então os dados podem ser vistos de ambas máquinas.

4.4 ALIMENTANDO A *STREAM* CADASTRO

A plataforma MultiChain aceita receber dados em formato hexadecimal, textual, *cache* binário ou em JSON (escolhido neste trabalho). O Cadastro foi alimentado com as informações contidas na Tabela 1. A *key* é o objeto em questão para o qual estamos nos referindo, uma chave única para cada registro da tabela. Em situações reais, este campo poderia conter o CPF, por exemplo.

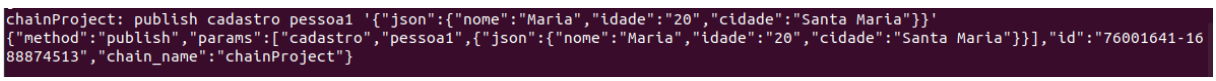
Tabela 1 – Tabela de dados.

Key	Nome	Idade	Cidade
Pessoa1	Maria	20	Santa Maria
Pessoa2	Joao	19	Porto Alegre
Pessoa3	Fernanda	45	Curitiba
Pessoa4	Pedro	58	Canoas
Pessoa5	Ana	30	Rio de Janeiro

Fonte: Próprio autor

Para adicionar os dados no Cadastro, é necessário utilizar o comando *publish cadastro &key '{"json":{"nome":"&Nome","idade":"&Idade","cidade":"&Cidade"}}'*, conforme exemplo da Figura 12, para Maria.

Figura 12 – Adicionando dados na *stream*.



Fonte: Próprio autor.

Uma vez que todas informações foram adicionadas à *stream*, é possível listar todos os dados com o comando *liststreamitems cadastro* para conferência do ID da transação

(*txid*), *publisher*, *timestamp* e demais informações. Também é possível listar por *key* com o comando *liststreamkeyitems cadastro &key*, como demonstrado na Figura 13.

Figura 13 – Listagem por referência.

Fonte: Próprio autor.

A partir deste momento, em que já temos uma Blockchain configurada e com dados iniciais atribuídos, é possível analisar algumas possibilidades de uso e aplicabilidades no contexto Cloud, demonstradas no próximo capítulo.

5 CENÁRIOS DE USO - ESTUDO DE CASO NO CONTEXTO CLOUD

Neste capítulo, serão explorados alguns exemplos de uso do chainProject considerando um cliente Cloud que possui a Tabela 1 como dados em seu sistema. O mesmo pode ser SaaS, PaaS ou IaaS.

Em uma situação real, a integração entre ambas tecnologias poderia ser feita por meio de uma API incluída pelos provedores em seus sistemas Cloud, mantendo a mesma vinculada à Blockchain em tempo real. Já que manter este vínculo exige mais poder computacional e conhecimento técnico, um custo adicional poderia ser incluso para os clientes que desejassem a funcionalidade.

Além disso, caso houvesse a necessidade de uma barreira a mais de segurança, o fornecedor da nuvem ainda teria a possibilidade de enviar os dados criptografados em modelo cache binário (aceito na plataforma MultiChain) e fazer uso das chaves privadas e públicas para garantir a troca de dados.

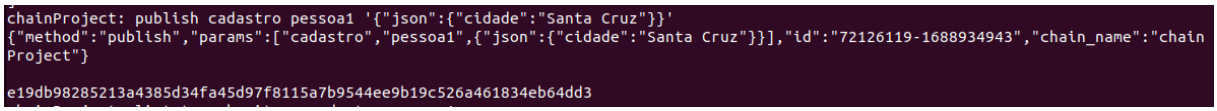
5.1 CENÁRIO 1 - ALTERAÇÃO DE DADOS

Digamos que a Pessoa1 - Maria mudou de cidade, e agora vai passar a residir em Santa Cruz. No contexto da Blockchain, há duas formas de realizar essa alteração, a depender de como o desenvolvedor gostaria que o seu projeto se comportasse.

Na primeira opção, o comando *publish cadastro pessoa1 '{"json":{"nome":"Maria", "idade":"20","cidade":"Santa Cruz"}}'* seria utilizado, gerando um novo bloco na Blockchain com todos os parâmetros. Assim, o último bloco conteria todas as informações, mas seria difícil identificar exatamente qual dado foi alterado.

A outra alternativa seria passar apenas a alteração com o comando *publish cadastro pessoa1 '{"json":{"cidade":"Santa Cruz"}}'* (Figura 14). Desta forma, um bloco contendo somente a alteração da cidade será encadeado na Blockchain. Esta foi a opção selecionada para o estudo, pois, pensando em uma auditoria do sistema Cloud, o cliente poderia visualizar mais facilmente quais dados foram alterados em cada transação, garantindo a consistência das informações. Além disso, desta forma não teremos dados duplicados na cadeia, apenas os originais e alterações posteriores.

Figura 14 – Alteração da cidade.



Fonte: Próprio autor.

Na Figura 15, é possível visualizar o histórico da Pessoa1, incluindo a alteração da cidade.

Figura 15 – Bloco da cidade adicional.

Fonte: Próprio autor.

Desta forma, podemos perceber que o bloco adicional possui apenas a informação que foi alterada, apesar de ainda ser possível visualizar os dados anteriores no primeiro bloco.

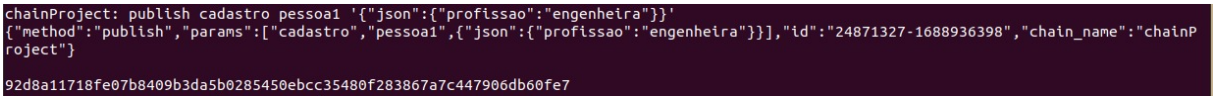
5.2 CENÁRIO 2 - INCLUSÃO DE DADOS POR OUTRO SERVIDOR

Como demonstrado no capítulo anterior, o servidor secundário também recebeu permissões para acessar a *stream* Cadastro e escrever nela. Essa funcionalidade seria

relevante no contexto Cloud para permitir que diferentes servidores tenham autorização para escrever ou ler a Blockchain.

Assim, a partir da segunda máquina, foi adicionada a profissão da Maria com o comando *publish cadastro pessoa1 '{"json":{"profissao": "engenheira"}}'*, como pode ser visto na Figura 16.

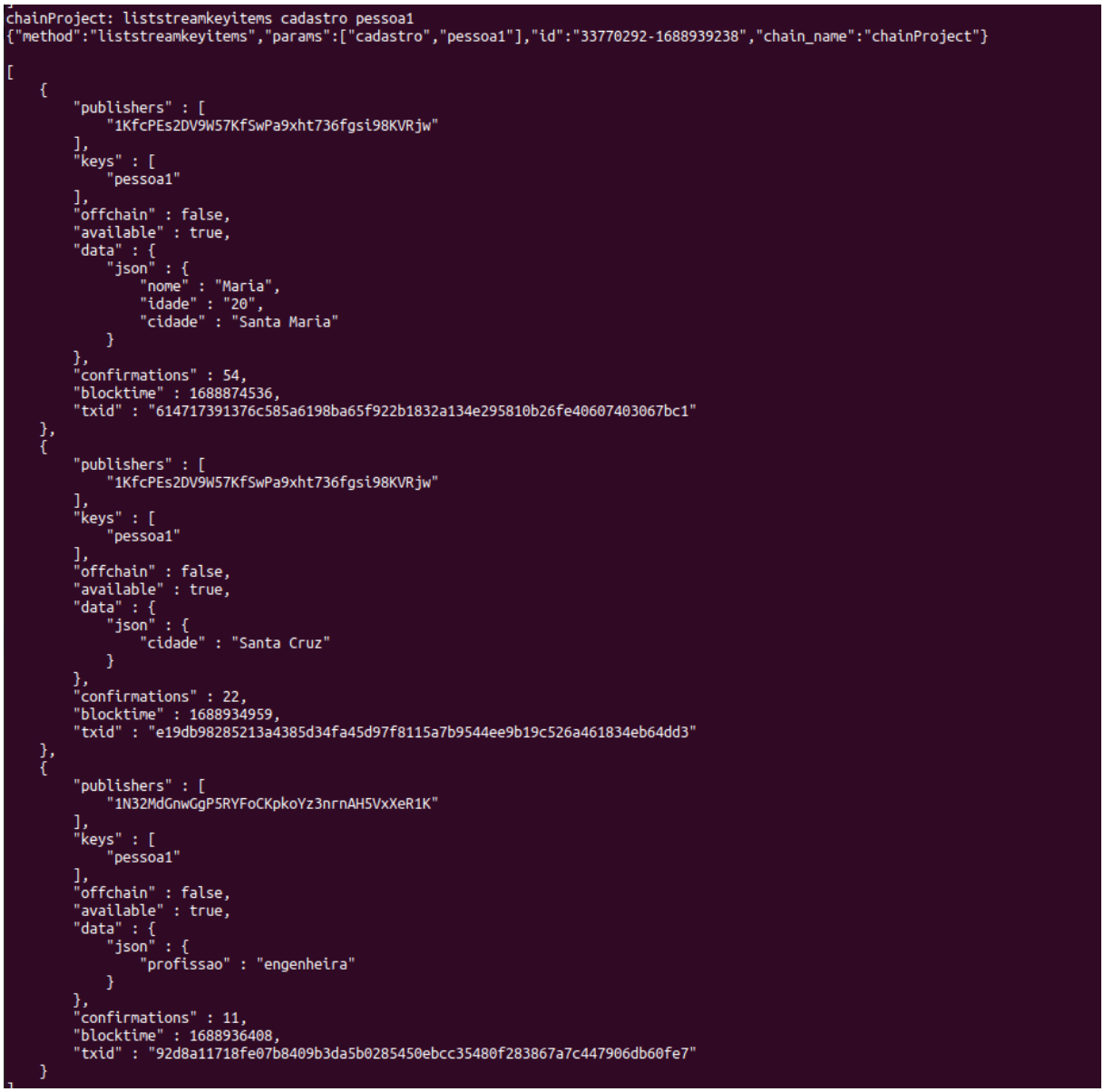
Figura 16 – Inclusão da profissão pela máquina secundária.



Fonte: Próprio autor.

Ao checar a listagem da *stream* Cadastro para a *key* Pessoa1, a inclusão da profissão é gerada por um bloco com o *address* do *publisher* do segundo servidor, como demonstrado na Figura 17.

Figura 17 – Listagem do Cadastro Pessoa1.



```
chainProject: liststreamkeyitems cadastro pessoa1
{"method":"liststreamkeyitems","params":["cadastro","pessoa1"],"id":"33770292-1688939238","chain_name":"chainProject"}

[
    {
        "publishers" : [
            "1KfcPEs2DV9W57KfSwPa9xht736fgsi98KVRjw"
        ],
        "keys" : [
            "pessoa1"
        ],
        "offchain" : false,
        "available" : true,
        "data" : {
            "json" : {
                "nome" : "Maria",
                "idade" : "20",
                "cidade" : "Santa Maria"
            }
        },
        "confirmations" : 54,
        "blocktime" : 1688874536,
        "txid" : "614717391376c585a6198ba65f922b1832a134e295810b26fe40607403067bc1"
    },
    {
        "publishers" : [
            "1KfcPEs2DV9W57KfSwPa9xht736fgsi98KVRjw"
        ],
        "keys" : [
            "pessoa1"
        ],
        "offchain" : false,
        "available" : true,
        "data" : {
            "json" : {
                "cidade" : "Santa Cruz"
            }
        },
        "confirmations" : 22,
        "blocktime" : 1688934959,
        "txid" : "e19db98285213a4385d34fa45d97f8115a7b9544ee9b19c526a461834eb64dd3"
    },
    {
        "publishers" : [
            "1N32MdGnwGgP5RYFoCKpkoYz3nrnAH5VxXeR1K"
        ],
        "keys" : [
            "pessoa1"
        ],
        "offchain" : false,
        "available" : true,
        "data" : {
            "json" : {
                "profissao" : "engenheira"
            }
        },
        "confirmations" : 11,
        "blocktime" : 1688936408,
        "txid" : "92d8a11718fe07b8409b3da5b0285450ebcc35480f283867a7c447906db60fe7"
    }
]
```

Fonte: Próprio autor.

Seguindo a mesma ideia do Cenário 1, o bloco adicional possui somente a nova informação, sem corromper os dados anteriores.

5.3 CENÁRIO 3 - VERIFICAÇÃO DA BLOCKCHAIN

Na medida em que alguns dados foram alterados e informações extras foram incluídas, podemos, por fim, testar o grande objetivo da Blockchain, que é justamente ter um histórico completo e imutável dos dados.

Se, por algum motivo, o cliente Cloud desconfiasse que suas informações foram adulteradas, o mesmo poderia entrar na Blockchain e checar o livro-razão para a Maria, por exemplo.

Cada bloco possui uma lista de informações referentes àquela transação. Algumas como *vout* e *blockindex*, são somente relevantes dentro do contexto de criptomoeda. Para fins de auditoria, os seguintes elementos podem ser verificados:

- *Publisher*: identificador da máquina que realizou a transação;
- *Keys*: chave única para referenciar o objeto em questão;
- *Data*: a informação que foi incluída na Blockchain;
- *Hash*: cada bloco possui seu *hash*, que pode ser verificado para garantir sua integridade;
- *Blockheight*: é o tamanho do bloco, e pode ser utilizado para definir o custo adicional cobrado pelos fornecedores Cloud para disponibilizar a Blockchain, uma vez que quanto maior a rede encadeada, maior poder computacional e de manutenção;
- *Timestamp (blocktime)*: Afim de averiguar o momento em que o bloco foi gerado;
- *Txid*: identificador da transação.

O comando que permite listar essas e demais informações é *liststreamkeyitems cadastro pessoa1 true*, e pode ser visualizado nas Figuras 18, 19 e 20, exibidas a seguir.

Figura 18 – Bloco 1 (entrada original dos dados).

Fonte: Próprio autor.

Figura 19 – Bloco 2 (alteração da cidade).

```
{
    "publishers" : [
        "1KfcPEs2DV9W57KfSwPa9xht736fgsi98KVRjw"
    ],
    "keys" : [
        "pessoa1"
    ],
    "offchain" : false,
    "available" : true,
    "data" : {
        "json" : {
            "cidade" : "Santa Cruz"
        }
    },
    "confirmations" : 22,
    "blockhash" : "00aa2d7370cd75ab1856e31b8db3399a8acf55129dd4f0889df4988932c299f1",
    "blockheight" : 213,
    "blockindex" : 1,
    "blocktime" : 1688934959,
    "txid" : "e19db98285213a4385d34fa45d97f8115a7b9544ee9b19c526a461834eb64dd3",
    "vout" : 0,
    "valid" : true,
    "time" : 1688934943,
    "timereceived" : 1688934943
},
```


Fonte: Próprio autor.

Figura 20 – Bloco 3 (adição da profissão por servidor secundário).

```
{
    "publishers" : [
        "1N32MdGnwGgP5RYFoCKpkoYz3nrnAH5VxXeR1K"
    ],
    "keys" : [
        "pessoa1"
    ],
    "offchain" : false,
    "available" : true,
    "data" : {
        "json" : {
            "profissao" : "engenheira"
        }
    },
    "confirmations" : 11,
    "blockhash" : "006be65b392647bd6ce83e24075854d52c93afe760fe083d767889d003077632",
    "blockheight" : 224,
    "blockindex" : 1,
    "blocktime" : 1688936408,
    "txid" : "92d8a11718fe07b8409b3da5b0285450ebcc35480f283867a7c447906db60fe7",
    "vout" : 0,
    "valid" : true,
    "time" : 1688936398,
    "timereceived" : 1688936398
}
]
```


Fonte: Próprio autor.

Como pode ser observado nas imagens, a informação de *"blockheight"* (tamanho do bloco) é sempre incrementada. O *hash* do novo bloco, por sua vez, é sempre gerado a partir do *hash* anterior, formando a relação de cadeia dos blocos.

É importante frisar, também, que mesmo que um processo de auditoria seja solicitado para verificação dos dados, ainda assim seria impossível que alguém não autorizado tivesse acesso às informações contidas na Blockchain.

6 CONCLUSÃO

Apesar do conceito principal da Blockchain estar relacionado a uma tecnologia descentralizada e pública, já que foi concebida no contexto de Bitcoin, vimos que suas características podem ser aplicadas em um ambiente controlado, onde acessos são restritos e providos de acordo com a necessidade do desenvolvedor.

A Blockchain privada pode ser aplicada no meio do Cloud Computing justamente como uma forma de armazenar os dados desejados (i.e, informações sensíveis e/ou restritas) em um livro-razão imutável, que pode ser consultado a qualquer momento como forma de auditoria.

Desta forma, a tecnologia da Blockchain conseguiria resolver alguns dos problemas descritos por Hiran et al. (2019), como a perda permanente de dados devido a pane nos sistemas ou falhas humanas, uma vez que o cliente poderia ter uma de suas máquinas conectada a Blockchain e garantir uma cópia de seus dados. Outro problema que essa tecnologia poderia auxiliar dentro de Cloud Computing é o acesso não autorizado, já que foi visto que a Blockchain permite configurações de permissões de leitura e escrita. Neste quesito, apesar do sistema Cloud ainda estar suscetível a ataques e violações de dados, a Blockchain estaria segura e não sofreria com fraudes causadas na nuvem, podendo servir de backup para restaurar o sistema.

Por outro lado, problemas relacionados à localização das informações e vulnerabilidades do sistema não são resolvidos com a integração a Blockchain, já que a mesma não teria controle sobre como a infraestrutura Cloud é mantida e onde.

Em questões de fragilidades da Blockchain privada, apesar de não ter o problema da rede pública, onde os nós majoritários conseguem corromper os blocos, ainda assim estaria vulnerável a ataques cibernéticos, por exemplo, se uma das máquinas é invadida. Uma solução para este problema seria deixar o mínimo de nós possíveis com a autorização de escrita, permitindo que estes somente visualizem os blocos. Além disso, em sistemas Cloud com um fluxo muito grande dados, a rede encadeada poderia acabar exigindo um poder computacional muito grande, trazendo possíveis problemas de performance.

Outro ponto importante a ser destacado é que a plataforma Multichain na sua versão *enterprise*, que seria a melhor escolha em uma aplicação real, possui diversas outras funcionalidades, como *Health Check* (verificação de saúde no português), que faz um diagnóstico completo nos nodes para garantir sua integridade. Outra Essa e outras funcionalidades disponíveis para a Multichain podem ser encontradas em sua documentação oficial [1].

Por fim, é importante ressaltar que para uma validação mais completa da proposta apresentada neste trabalho, é necessário testar essas situações em um ambiente real e

[1] Disponível em: https://www.multichain.com/developers/json-rpc-api/

analisar o interesse de empresas ao aderirem uma Blockchain junto a sistemas Cloud. Também, para trabalhos futuros, poderia ser estudada a implementação de uma interface mais amigável para os usuários acessarem a Blockchain.

REFERÊNCIAS

BAYER, D.; HABER, S.; STORNETTA, W. S. Improving the efficiency and reliability of digital time-stamping. 1992. Acesso em 20 jun. 2023. Disponível em: <www.math.columbia.edu-/˜bayer/papers/Timestamp_BHS93.pdf>.

CHICARINO, V. R. L. et al. Uso de blockchain para privacidade e segurança em internet das coisas. **Sociedade Brasileira de Computação**, 2017.

DORSALA, M. R.; SASTRY, V.; CHAPRAM, S. Bloackchain-based solutions for cloud computing: A survey. **Journal of NEtwork and Computer Applications**, v. 196, 2021. Acesso em 2 jul. 2023. Disponível em: <https://doi.org/10.1016/j.jnca.2021.103246>.

FORTINET. Cloud security report. 2022. Disponível em: <https://global.fortinet.com/latam-lp-pr-CloudAWSsummit?utm_source=website&utm_medium=brpubliccloudsecuritypp&utm_campaign=c latam-aws-br&utm_content=cloud-security-report-2022>.

Grand View Research. **Cloud Computing Market Size**: Share trends analysis report by service (saas, iaas), by end-use (bfsi, manufacturing), by deployment (private, public), by enterprise size (large, smes), and segment forecasts, 2023 - 2030. 2022. Acesso em 20 jun. 2023. Disponível em: <https://www.grandviewresearch.com/industry-analysis/cloud-computing-industry?utm_source=prnewswire&utm_medium=referral&utm_campaign=ict_01-sep-21&utm_term=cloud-computing-market&utm_content=rd1>.

GUEGAN, D. Public blockchain versus private blockchain. **HAL**, 2017.

GUPTA, A. et al. Cloud computing security using blockchain. **JETIR**, v. 6, 2019.

HIRAN, K. et al. **Cloud computing: master cloud computing concepts, archicteture and applications with-real world examples and case studies**. New Delhi: BPB Publications, 2019. 332 p.

LAURENCE, T. **Blockchain Para Leigos**. Rio de Janeiro: Alta Books Editora, 2019.

LUCENA, A. U. de. Estudo de arquiteturas dos blockchains de bitcoin e etherum. 2016. Acesso em 29 jun. 2023. Disponível em: <https://diegoazziufabc.files.wordpress.com-/2017/08/estudo-de-arquiteturas-dos-blockchains.pdf>.

MARCHSIN, K. B. K. **Blockchain e smart contracts:**: As inovações no âmbito do direito. São Paulo: Editora Saraiva, 2022.

MORAES, A. F. de. **Bitcoin e Blockchain: a revolução das moedas digitais**. São Paulo: Editora Saraiva, 2021.

MultiChain. **Getting started with MultiChain**. 2023. Acesso em 8 Julho 2023. Disponível em: <https://www.multichain.com/getting-started/>.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. 2008. Acesso em 20 jun. 2023. Disponível em: <https://bitcoin.org/bitcoin.pdf>.

NASCIMENTO, L. B. G. et al. **Criptomoeadas e Blockchain**. Porto Alegre: SAGAH, 2022.

NIST. **The NIST Definition of Cloud Computing**. 2011. Acesso em 15 maio 2023. Disponível em: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>.

Nutanix. **Enterprise Cloud Index**: The explosion of data across clouds. 2023. Acesso em 20 jun. 2023. Disponível em: <https://www.nutanix.com/enterprise-cloud-index#navhero>.

PREUKSCHAT, A. et al. Blockchain: la revolución industrial de internet. 2017.

SILVA, F. R. da et al. **Cloud Computing**. Porto Alegre: SAGAH, 2020.

SIMãO, G. F.; SILVA, G. G.; PAIVA, L. F. A tecnologia blockchain aplicada À educação. **Uniube**, 2018. Acesso em 29 jun. 2023. Disponível em: <repositorio.uniube.br/bitstream/123456789/517/1-/Guilherme\%20Felipe\%20Sim\%c3\%a3o\%20e\%20Gustavo\%20Gomes\%20Silva-.pdf>.

ULRICH, F. Bitcoin: A moeda na era digital. **Mises Brasil**, 2014. Acesso em 20 jun. 2023.

NUP: 23081.099485/2023-28 **Prioridade:** Normal

Homologação de ata de defesa de TCC e estágio de graduação

125.322 - Bancas examinadoras de TCC: indicação e atuação

COMPONENTE

Ordem	Descrição	Nome do arquivo
8	Trabalho de conclusão de curso (TCC) (125.32)	O USO DE BLOCKCHAIN PARA AUDITORIA DE DADOS.pdf

Assinaturas

07/08/2023 15:52:17

HELENA FERNANDA KRAY (Aluno de Graduação - Aluno Regular)
07.09.09.01.0.0 - Curso de Engenharia de Computação - 121624

Universidade Federal de Santa Maria
SEDES SAPIENTIÆ
1960

Código Verificador: 3090278
Código CRC: 70f4ffdf
Consulte em: https://portal.ufsm.br/documentos/publico/autenticacao/assinaturas.html