**ANÁLISE COMPARATIVA ENTRE A TECNOLOGIA BLOCKCHAIN E BANCO DE DADOS**

CASO APLICADO NA COLABORAÇÃO COM O TERCEIRO SETOR

Angelo Rocha Neto

Graduando em Banco de dados pela Fatec Bauru

E-mail: [angelo.neto@fatec.sp.gov.br](mailto:angelo.neto@fatec.sp.gov.br)

Gustavo Bruchi

\*\*. Docente na Fatec Bauru

E-mail: \*\*

**RESUMO**

A presente análise propõe uma investigação detalhada e comparativa entre duas tecnologias de armazenamento de dados que têm desempenhado papéis cruciais em diferentes domínios: a *blockchain* e os bancos de dados relacionais. O estudo busca avaliar suas características intrínsecas, arquiteturas, capacidades de escalabilidade, segurança, eficiência e aplicabilidade em diversas áreas. A partir de um caso de doações para organizações do terceiro setor, que são responsáveis por auxiliar e desenvolver atividades de interesse social sem nenhum fim lucrativo, recebendo apoio por meio de parcerias públicas ou privadas, situação que gera um compromisso de transparência de todos os gastos realizados.

**Palavras-chaves:** banco de dados; *blockchain*; transparência; terceiro setor; contratos inteligentes; *peer-to-peer*; *hash*;

ABSTRACT

The present analysis proposes a detailed and comparative investigation between two data storage technologies that play crucial roles in different domains: blockchain and relational databases. The study seeks to evaluate its intrinsic characteristics, architectures, scalability capabilities, security, efficiency and applicability in several areas. Based on a case of donations to third sector organizations, which are responsible for helping and developing non-profit social interest activities, receiving support through public or private partnerships, a situation that generates a commitment to transparency from all the expenses incurred.

Keywords: database; blockchain; transparency; third sector; smart contracts; peer-to-peer; hash;

**1 INTRODUÇÃO**

Nos últimos anos, as tecnologias de armazenamento de dados têm experimentado diversos avanços, impulsionados pela crescente complexidade das operações de negócios, bem como pela necessidade contínua de garantir a segurança e a integridade dos dados. Duas abordagens distintas, amplamente discutidas e implementadas, surgem como solução para esse cenário em constante evolução: os bancos de dados relacionais tradicionais e a tecnologia blockchain emergente.

Os bancos de dados relacionais, que se estabeleceram como a base da gestão de dados por décadas, são conhecidos por sua estrutura tabular e pelo uso do SQL (Structured Query Language) para consultas e gerenciamento de informações. Esses sistemas são reconhecidos pela robustez e confiabilidade, com propriedades de ACID (Atomicidade, Consistência, Isolamento e Durabilidade) que garantem a integridade das transações.

**2.1 BLOCKCHAIN**

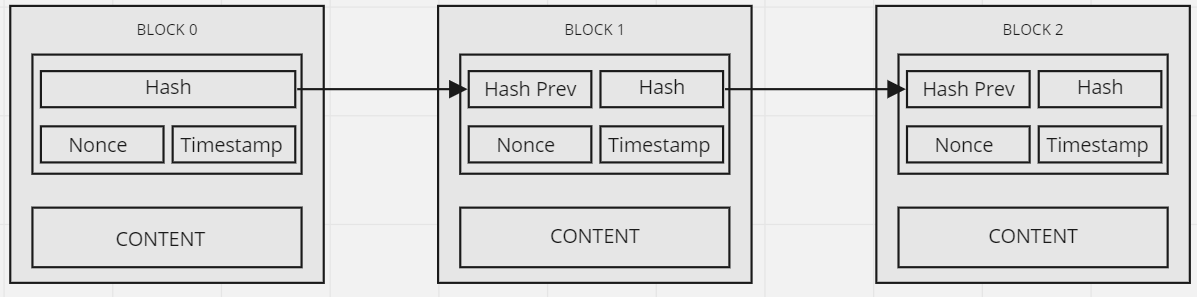
Com o intuito de evitar um gasto duplo (*double speding*), após uma grande crise econômica mundial gerada pelo setor imobiliário dos EUA, em 2009 foi criada publicamente uma rede *blockchain*, para utilização de uma moeda virtual, o *bitcoin*. Constituída pelo pseudônimo Satoshi Nakamoto, representando uma pessoa ou talvez um grupo de pessoas, sendo desconhecido o real autor, que realizou o lançamento do *white paper bitcoin*: um sistema de dinheiro eletrônico *peer-to-peer*.

Essa rede consiste em ser descentralizada, armazenando as informações em blocos interligados em uma cadeia. Os dados permanecem cronologicamente consistentes e salvos em um livro-razão que está disponível para todo participante, se mantendo integro, pois não é possível excluir nem modificar nenhum dado na cadeia, sem que haja um controle feito pelo algoritmo de consenso, pré-estabelecido para toda a rede. Como resultado disso, qualquer transação pode ser mantida inalterável ou imutável para monitorar qualquer documento, moeda, pagamento, pedido ou bens que estão sendo negociados.

Possuindo inúmeras evidências das transações, segundo Pegoraro (Blockchain, p. 8) “Nessa tecnologia disruptiva é possível efetuar vários registros – como em um banco de dados distribuído pelo mundo inteiro – de forma imutável, onde as pessoas se comunicam diretamente, havendo uma transparência nas informações”.

O processo de uma nova operação que será encadeada aos outros blocos, segue algumas etapas, a primeira delas é o evento de inicialização do bloco da transação, que será enviado para todos os nós de rede, que com a cópia do livro-razão, através do consenso estabelecido pela maioria da rede, irão validar se cada parte corresponde corretamente a transação corrente. A partir disso o bloco é adicionado ao final da cadeia, contendo as informações no seu cabeçalho como *hash* de identificação do bloco anterior, o *timestamp* representando a data de criação do bloco, campo *nonce* que é utilizado para provar que um bloco foi validado e que atende aos critérios estabelecidos, a *hash* de identificação do próprio bloco, além das informações da transação do bloco.

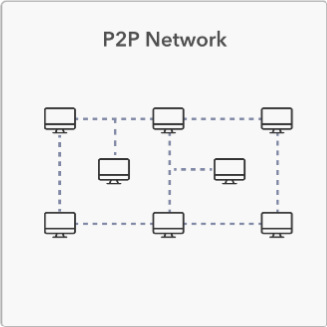
A interligação dos blocos ocorre através da dependência de envolver no seu cabeçalho, a *hash* de identificação do bloco anterior do encadeamento, com a exceção do primeiro bloco da cadeia, conhecido como o bloco gênesis.

 Figura 1 – Encadeamento de blocos

**2.2 PEER-TO-PEER (P2P)**

A arquitetura de banco de dados *blockchain* tem como a sua principal característica a descentralização da comunicação entre dois ou mais nós (computadores) que compõem uma rede peer-to-peer, sendo um conceito de redes de computadores que segundo Tanenbaum (2010), os nós agem como clientes e servidores para os outros nós da rede, dessa forma compartilhando serviços e dados sem a necessidade de um servidor centralizado.

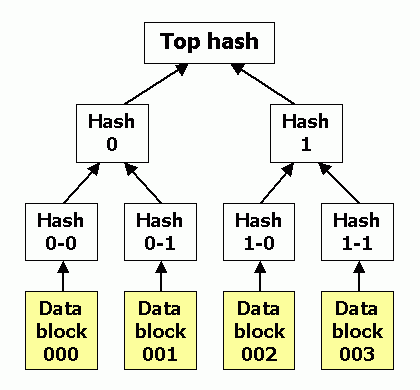
Figura 2 – Rede Peer-to-Peer

****

**2.3 ÁRVORE DE MERKLE**

A estrutura de dados, denominada como árvore de merkle, é dividida em várias camadas, cuja finalidade dentro da cadeia de blocos é a ligação criptográfica de todos os principais componentes do livro razão. Em seu conceito que relaciona os nós com uma única raiz associada a eles. Cada nó deve possuir um identificador exclusivo, conhecido como hash, os chamados nós filhos (folha), estão associados a um nó superior chamado nó pai (ramificação), dessa forma o nó superior contém o resultado da soma das informações presentes em nós inferiores, ou seja, à medida que você continua a escalar, a mesma estrutura se repete, pois, todos os blocos estão conectados a um bloco raiz.

Figura 3 – Rede Peer-to-Peer



O fato de ser estruturado dessa forma, faz com que na alteração dos dados de uma parte, acabe invalidando todas as demais, possibilitando uma verificação no caso de mudanças de um nó, servindo como um mecanismo para evitar adulteração nas informações, permitindo um alto nível de confiabilidade na transmissão de dados em redes descentralizadas, pois, se tornam computacionalmente eficientes na criação, processamento e verificação das informações.

**2.4 FUNÇÃO HASH**

Para realização da criptografia de dados na *blockchain,* cada bloco necessita de um identificador conhecido como *hash*, obtido através de uma função, a partir da entrada dos dados contidos no próprio bloco, com o tamanho de caracteres variável, criando após a sua execução uma *string* de tamanho fixo, estabelecido pela própria função, alguns exemplos de funções utilizadas:

1. MD5: Essa função possui um algoritmo de hash de 128 bits desenvolvido pela RSA Data Security, muito utilizado com o protocolo *p2p,* porém com alguns métodos de ataques conhecidos;
2. SHA-1: Projetada para ser um padrão de processamento de informações, essa função produz um valor de dispersão de 160 bits, tratado como um número hexadecimal de 40 dígitos;
3. SHA-2: Recomendado pelo *National Institute of Standards and Technology* (NIST), este algoritmo retorna um valor de hash de 224 ou 256 bits de acordo com a versão em questão, que gera respectivamente 56 ou 64 caracteres hexadecimais, sendo os mais utilizados para o algoritmo de consenso.

As funções são executadas de forma unidirecional, isso representa que não é possível recuperar os dados originais a partir do resultado gerado após a execução, sendo que qualquer pequena mudança no conteúdo de entrada gera uma sequência totalmente diferente da anterior. Outra propriedade importante das funções hash é a colisão, quando se tem dois dados originais que geram o mesmo conteúdo, tem-se uma colisão, quanto mais uniforme e dispersa ao se utilizar de ferramentas de ajustes da distribuição do resultado, menor probabilidade da colisão na função.

As características contidas na função que será executada, faz com que cada bloco da cadeia possua um identificador totalmente único dentro da sequência, o que favorece o conceito de árvore merkle, utilizado pela arquitetura *blockchain* para encadear e arquivar as transações entre os nós descentralizados, que de certa forma facilita a validação de cada parte sabendo que segundo Braga (2021), a verificação do *hash* de uma transação só usa o ramo da árvore em que a transação está localizada, que é necessário para verificar o *hash* da transação. Ou seja, o ramo em que não possui o identificador da transação não será verificado.

**2.5 ALGORITMO DE CONSENSO**

Termo da área de ciência da computação o consenso está intrinsicamente relacionado a disciplina de sistemas descentralizado e área fundamental para o funcionamento de redes P2P *blockchain*. Consenso não significa unanimidade, mas sim que quase todos concordam, buscando resolver o problema comum entre os participantes. Segundo Revoredo (2019) nas *blockchains*, um protocolo ou mecanismo de consenso é um conjunto de regras que descreve como funciona a comunicação e o registro da transmissão de dados entre dispositivos eletrônicos, conhecidos por nós ou nodes.

Para flexibilidade da rede existem diversos tipos de algoritmos de consenso. A seguir alguns desses tipos:

**2.5.1 PROVA DE TRABALHO – PROOF OF WORK (PoW)**

Sendo utilizado pela criptomoeda *bitcoin*, esse algoritmo corresponde a um desafio de trabalho para validação do bloco, após ter passado pela verificação da função SHA-256, deve-se garantir que o próximo bloco tenha todos os *n* primeiros números do resultado do hash, para isso ele altera o nonce criptográfico ao final do bloco, não havendo mudanças da transação em si, um nonce é um valor numérico que pode ser modificado milhares de vezes até que se leve ao resultado com n primeiros números sendo zero. Para todos os outros usuários verificarem a validade desse próximo bloco, eles só precisam verificar se ele tem os primeiros *n* dígitos zero. E após todo o processo estar feito o nó minerador recebe uma recompensa pelo seu trabalho.

Esse algoritmo de consenso se torna seguro, por seu formato, porém acaba consumindo mais poder computacional, por conta da dificuldade de resolução estabelecida, consequentemente acaba consumindo mais energia para funcionamento de acordo com o crescimento de nós na rede.

**2.5.2 PROVA DE PARTICIPAÇÃO – PROOF OF STAKE (PoS)**

Em vez de gastar recursos computacionais, um nó minerador deve apostar parte de seus ativos para receber uma chance de minerar um bloco. Sua chance é proporcional à quantia de ativos apostados, ganha o direito de fechar o bloco quem tiver apostado mais. A prova de participação tem se destacado devido ao *fork* efetuado pela rede Ethereum que abandonou o método de prova de trabalho para implementar PoS, sendo considerado o Ethereum.

Segundo Remoaldo (2022), a principal vantagem é que o procedimento é mais rápido e com baixo custo de energia. A desvantagem é a possibilidade de haver centralização no fechamento dos blocos (de quem tiver mais dinheiro).

**2.5.3 PROVA DE PARTICIPAÇÃO DELEGADA – DELEGATED PROOF OF STAKE (DPoS)**

Segundo descrito por Kiayias (2017), os nós mineradores utilizam de seus ativos para eleger delegados em um quórum que define o bloco a ser adicionado. A quantidade de votos de um minerador é proporcional aos seus ativos. Embora PoS e DPoS sejam semelhantes no sentido de ambos se basearem em "participação", DPoS apresenta um novo sistema de votação democrática, pelo qual aqueles que detém maiores quantidades da moeda são eleitos. Uma vez que um sistema DPoS é mantido pelos eleitores, os delegados são motivados a serem honestos e eficientes, caso contrário não são eleitos. Além disso, as *blockchains* baseadas em DPoS tendem a ser mais rápidas em termos de transações por segundo.

**2.5.4 PROVA DE QUEIMADURA – PROOF OF BURN (PoB)**

Os ativos apostados são queimados intencionalmente como uma maneira de “investir” recursos na *blockchain*, de modo que os candidatos a mineradores não necessitam investir com recursos físicos (hardware).

Segundo Remoaldo (2022) “queimar” uma criptomoeda significa enviá-la paraum endereço inexistente. Como sabemos que a blockchain é irreversível, ao fazer isso você perde essa criptomoeda – ou no jargão da comunidade, “queima”.

**2.5.6 PROVA DE AUTORIDADE – PROOF OF AUTHORITY (PoA)**

Similar ao DPoS, porém o conjunto de delegados (autoridades) é pré-determinado em acordo e suas identidades são públicas e verificáveis por qualquer participante da rede.

O modelo de PoA é dependente de um número limitado dos validadores de bloco, tornando-o um algoritmo altamente escalável. Blocos e transações são verificados por participantes pré-aprovados, que agem como moderadores do sistema, sendo considerado uma opção valiosa para aplicações logísticas.

**2.5.7 PROVA DE CAPACIDADE – PROOF OF CAPACITY (PoC)**

A probabilidade de propor um bloco é proporcional ao espaço de armazenamento cedido à rede por um nó minerador. Quanto maior a capacidade de armazenamento em disco, maior o domínio sobre o consenso.

**2.5.8 PROVA DE TEMPO DECORRIDO – PROOF OF ELAPSED TIME (PoET)**

Cada nó minerador recebe um temporizador aleatório decrescente e o nó cujo temporizador terminar primeiro propõe o próximo bloco. Este protocolo de consenso funciona exclusivamente em hardware que suportam a tecnologia Intel software guard extensions (SGX). O Intel SGX garante a distribuição aleatória de temporizadores e que nenhuma entidade tem acesso a mais de um nó minerador, geralmente utilizada nas *blockchains* privadas como a *Hyperledger – Sawtooth Lake Project*

**3 CONTRATOS INTELIGENTES (SMART CONTRACTS)**

Contratos inteligentes, ou *smart contracts*, como também são chamados, são programas de computação executados em uma máquina virtual chamada VM e que fazem parte do protocolo da *blockchain*.

Por estarem na *blockchain*, eles são imutáveis, motivo pela qual devem ser muito bem testados antes de ser utilizado em produção, evitando falhas.

Uma das grandes vantagens dos contratos inteligentes, é que pode ser incluído variáveis que possuem informações das partes envolvidas no processo, pense que uma transportadora, rastreia sua entrega através da cadeia de blocos, sendo assim é possível obter variáveis como: número do produto, motorista, endereço de entrega, tipo de produto ou cliente, dando assim inúmeras utilidades para serem pré-programas no contrato, como por exemplo executar um smart contract a partir do momento que o cliente, recebeu o produto, retirando o mesmo dos ativos de estoque, validando a transação. Isso gera uma segurança quando o contrato é bem programado, sabendo que o mesmo permanece visível a todos, que podem acompanhar o processo e checar em tempo real o que está acontecendo a qualquer momento.

Segundo Alves (2020), um contrato inteligente pode ser entendido como um agente autônomo armazenado em uma *blockchain*, onde o contrato é enviado da mesma forma que uma transação. Assim, ele deve ser aprovado pelos nós da rede de acordo com o seu mecanismo de consenso. Uma vez criado, o contrato inteligente é identificado por um endereço para que possa ser chamado por outros sistemas, usuários e até mesmo por outros contratos inteligentes.

**4 IMPLEMENTAÇÃO**

A rede blockchain é construída utilizando o framework Truffle, que facilita a criação e gerenciamento de contratos inteligentes. A integração com a web é realizada através da biblioteca Web3, que permite a interação direta com a rede blockchain. Para a interface do usuário, as tecnologias React e Node.js são empregadas, garantindo uma experiência de usuário moderna e responsiva.

**4.1 PLATAFORMA DE INFRAESTRUTURA: GOOGLE CLOUD**

A plataforma Google Cloud foi selecionada devido às suas características de escalabilidade, flexibilidade e confiabilidade. O Google Cloud Platform (GCP) oferece uma variedade de serviços de hospedagem de máquinas virtuais, proporcionando um ambiente adequado para a execução das aplicações blockchain.

**4.2 IMPLEMENTAÇÃO DA REDE BLOCKCHAIN COM TRUFFLE**

O framework Truffle foi escolhido para a criação da rede blockchain devido à sua ampla aceitação na comunidade de desenvolvimento blockchain. Foram definidos contratos inteligentes que especificam as regras de colaboração entre as organizações do terceiro setor. Esses contratos são programados em linguagem de programação Solidity, permitindo a automatização de processos e transações na rede.

**4.3 INTEGRAÇÃO WEB COM WEB3**

A biblioteca Web3 foi empregada para possibilitar a interação entre as aplicações web e a rede blockchain. Essa integração permite a criação, leitura e atualização de registros na blockchain por meio de chamadas de função nos contratos inteligentes. Dessa forma, as informações na rede blockchain são acessíveis e transparentes para as organizações envolvidas.

**4.4 DESENVOLVIMENTO DE INTERFACES COM REACT E NODEJS**

As tecnologias React e Node.js foram utilizadas para criar interfaces de usuário amigáveis e dinâmicas. O React permite a construção de componentes reutilizáveis, proporcionando uma experiência de usuário consistente e intuitiva. O Node.js foi adotado para o desenvolvimento do lado do servidor, fornecendo um ambiente de execução eficiente e suportando a comunicação entre o cliente e a rede blockchain.

**5 REFERÊNCIAS**

A.S. TANENBAUM, “**Computer Networks**”, 5th Edition, Pearson Education, 2010.

BRAGA, A.M. **Tecnologia blockchain:** *Fundamentos, tecnologias de segurança e desenvolvimento de software***.** ed. Centro de pesquisa e desenvolvimento de telecomunicação 2021.

P.H. ALVES, R. LAIGNER, R. NASSER, G. ROBICHEZ, H. LOPEZ, M. KALINOWSKI, “**Desmistificando Blockchain: Conceitos e Aplicações**”, Em: C. Maciel, J. Viterbo (Orgs). “Computação e Sociedade”, Sociedade Brasileira de Computação, 2020.

REMOALDO, D.P. **Blockchain.** 1. ed. Senac São Paulo: (Série Universitária), 2022.

REVOREDO, T. **Blockchain**: tudo o que você precisa saber. São Paulo: Independently Published, 2019.

KIATIAS, A., RUSSELL, A., DAVID, B., OLIYNYKOV, R. **Ouroboros**:   
Um protocolo blockchain de proof-of-stake comprovadamente seguro. In Katz, J. and Shacham, H., editors, Advances in Cryptology – CRYPTO 2017.