ANÁLISE COMPARATIVA ENTRE A TECNOLOGIA BLOCKCHAIN E BANCO DE DADOS RELACIONAL

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

Orientador: Gustavo Cesar Bruschi
**. Docente na Fatec Bauru
E-mail: gustavo@bruschi.net

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. Através do caso foi possível concluir que a tecnologia blockchain oferece uma abordagem inovadora para o armazenamento dos dados, compartilhando semelhanças com os princípios de bancos de dados tradicionais. Foi possível concluir que seu uso se destaca para cenários que envolvem ativos de valor em que é necessário transparência e consenso entre partes, como demonstrado no caso envolvendo doadores e instituições do terceiro setor.

Palavras-chaves: banco de dados; *blockchain*; transparência; terceiro setor; contratos inteligentes; *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. Dois conceitos diferentes, amplamente discutidos e implementados, 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 as propriedades de ACID (Atomicidade, Consistência, Isolamento e Durabilidade) que garantem a integridade das transações.

Por outro lado, a tecnologia *blockchain*, que obteve sua ascensão a partir da criação das criptomoedas, em destaque o *bitcoin*, representa uma abordagem diferente e inovadora para o armazenamento e a transmissão segura de dados. Sendo baseada em princípios de descentralização, imutabilidade e consenso, a *blockchain* oferece um novo paradigma para a gestão de informações, onde as transações são registradas em blocos conectados em uma cadeia, proporcionando uma auditoria transparente e inviolável, sabendo que uma vez que uma transação é confirmada e registrada em um bloco, ela não pode ser alterada, contribuindo assim para a integridade dos dados.

O presente projeto propõe uma análise comparativa entre essas duas tecnologias de armazenamento de dados, explorando suas principais características, entre elas: segurança, integridade e eficiência. Em especifico a aplicação da arquitetura *blockchain* em um estudo de caso, desenvolvendo uma aplicação dedicada à auditoria de todas as transações ou acordos realizados no âmbito da rede, que envolvam duas partes em conjunto com os contratos inteligentes (*smart-contracts*). Com o intuito de colaborar com a administração pública em um regime de cooperação mútua, aonde se aplica o MROSC (Marco Regulatório das Organizações da Sociedade Civil).

Segundo a Secretaria Nacional de Assistência Social o MROSC tem como objetivo promover a transparência nas relações entre entidades do terceiro setor, entretanto, a regularização segundo as normas propostas para os setores apresenta desvantagens evidentes, como por exemplo a prestação de conta de todo dinheiro que foi arrecadado e aonde o mesmo foi aplicado. Criando um sistema burocrático que pode, em alguns casos, impossibilitar que organizações enquadradas recebam apoio de outras partes ou que a sociedade num todo seja beneficiada com os serviços prestados pela entidade. Adicionalmente, existe o risco de que algumas entidades, de má fé, possam utilizar os recursos recebidos em seu próprio benefício, sem a devida transparência e prestação de contas perante o governo, que realiza o repasse de dinheiro público, especialmente no que diz respeito aos recursos destinados aos municípios ou estados.

A partir desse caso com as instituições do terceiro setor, foi possível concluir que a rede blockchain provou-se ser eficaz na armazenagem e transferência de valores, mantendo a confiabilidade entre as partes de doadores e entidades. Também foi visto

que a tecnologia blockchain oferece benefícios significativos, como transparência, descentralização e consistência, especialmente no contexto de representação de ativos. A capacidade da blockchain de trabalhar em conjunto com bancos de dados relacionais é ressaltada, evidenciando que ela não requer a eliminação completa das funcionalidades desses bancos, mas sim uma integração que resulta em uma complementação valiosa. Foi possível concluir que a blockchain pode ser adotada de forma conjunta, aprimorando a eficiência e a confiabilidade dos sistemas existentes, ao invés de substituí-los integralmente. Essa abordagem destaca a versatilidade da tecnologia e sua capacidade de se adaptar e melhorar processos já estabelecidos, representando um avanço promissor no campo da gestão de ativos e informações.

2 FUNDAMENTOS DA 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 Remoaldo (2022) “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, Segundo Nakamoto (2008) começa com um servidor de carimbo de data/hora. Um servidor de carimbo de data/hora funciona tomando um hash de um bloco de itens a ser carimbado e publicar amplamente o hash, como em um jornal ou postagem. O carimbo de data/hora prova que os dados devem ter existido no tempo, obviamente, para entrar no hash. Cada carimbo de data/hora inclui o carimbo de data/hora anterior em seu hash, formando uma cadeia, com cada timestamp adicional reforçando os anteriores.

Cada transação podendo executar um contrato inteligente ou não, Segundo Remoaldo (2022) cada bloco fechado é distribuído em toda a rede, sendo que cada nó terá uma cópia de todos os blocos registrados.

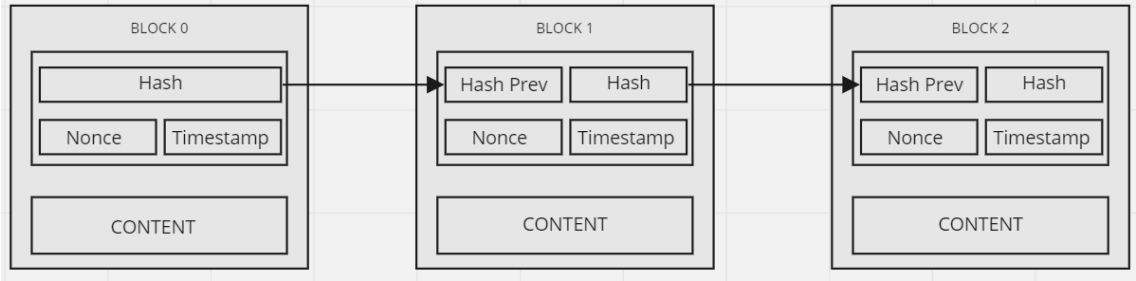
O livro-razão fica armazenado em cada nó mostrando a importância deles na distribuição da blockchain, então a rede segue passos específicos para tratar a transação dentro dos nós. Segundo Nakamoto (2008) novas transações são transmitidas para todos os nós. Cada nó coleta novas transações em um bloco. Cada

nó trabalha para encontrar uma prova de trabalho difícil para seu bloco. Quando um nó encontra uma prova de trabalho, ele transmite o bloco para todos os nós. Os nós aceitam o bloco somente se todas as transações nele contidas forem válidas e ainda não gastas. Os nós expressam sua aceitação do bloco trabalhando na criação do próximo bloco na cadeia, usando o hash do bloco aceito como o hash anterior.

Segundo Dinh (2017), para um bloco fazer parte da rede, quatro camadas são processadas: transações, validação, geração de blocos e distribuição.

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



Fonte: Adaptado dicionário financeiro.

Segundo Tapscott (2016), essa nova maneira de lidar com o armazenamento de dados e de transações financeiras tem a capacidade de ser programada para guardar virtualmente praticamente qualquer coisa que tenha valor e relevância para a humanidade, como escrituras, certidões, diploma, registros médicos, dados contábeis, registros de votos, dados que possibilitem a rastreabilidade de produtos alimentícios, entre as diversas coisas que podem ser expressas através de código computacional.

2.1 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. Vale ressaltar que a criptografia assimétrica é geralmente utilizada para criptografar pequenos blocos de dados, como chaves de criptografia e valores de função hash, utilizadas em assinaturas digitais, afirma Stalling (2011). Estabelecido pela própria função, alguns exemplos de funções utilizadas:

a) 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;
b) 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;
c) SHA-2: Recomendado pelo NIST (*National Institute of Standards and Technology)*, 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.

2.3 Algoritmo De Consenso

Termo da área de ciência da computação o consenso está intrinsicamente relacionado a disciplina de sistemas descentralizado, sendo uma 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.

2.3.1 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.

2.3.2 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).

2.3.3 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.

2.3.4 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”.

2.3.5 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.

2.3.6 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.

2.3.7 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 CARACTÉRISTICAS DA BLOCKCHAIN ANÁLOGAS AO BANCO DE DADOS RELACIONAL

No funcionamento de uma base de dados relacional, principalmente nos Sistemas Gerenciadores de Banco de Dados (SGBD), é de extrema importância o conceito de transações, uma transação pode ser definida, segundo Lóscio (2011), como uma coleção de operações que desempenha uma função lógica dentro de uma aplicação do sistema de banco de dados, em outras palavras uma transação representa um conjunto de operações de leitura ou escrita que são realizadas no banco de dados. A execução de transações em um SGBD deve obedecer a algumas propriedades a fim de garantir o correto funcionamento do sistema e a respectiva consistência dos dados. Estas propriedades são chamadas de propriedade ACID.

Tais propriedades referidas como atomicidade, consistência, isolamento e durabilidade podem ser análogas as características presentes em uma rede *blockchain*, descrito a seguir:

3.1 Atomicidade

Em um banco de dados relacional a atomicidade diz respeito a como tudo está sujeito a falhas e caso isso aconteça o sistema deve retornar para o último estado consistente que existia antes da falha.

A atomicidade é uma propriedade que garante que cada transação seja tratada como uma entidade única, a qual deve ser executada por completo ou falhar completamente. Desta forma, todas as operações da transação devem ser executadas com sucesso para que a transação tenha sucesso. (REIS, 2018).

Em uma rede *blockchain* essa característica é alcançada de maneira diferente, devido a sua arquitetura descentralizada e à natureza de registros imutáveis, as transações são agrupadas em blocos e registradas de forma sequencial. Em cada bloco contém um conjunto de transações que são tratados como uma unidade indivisível. Isso significa que todas as transações de um bloco são executadas em

conjunto ou não, isso ocorre através do mecanismo de consenso, que desempenha um papel crucial na garantia de atomicidade.

3.2 Consistência

A propriedade de consistência permite assegurar que uma transação somente faça um procedimento a partir de um estado válido para um outro, sem que haja a corrupção de dados, isso mantém a estabilidade do banco de dados em qualquer tipo de operação que seja realizada.

Para estabelecer a consistência nos dados na rede *blockchain,* são utilizados os protocolos de consenso para garantir que todas as cópias do livro-razão na rede estejam em conformidade com a mesma versão do registro. Esses protocolos de consenso garantem que todos os participantes da rede concordem sobre quais transações são válidas e quais não são. Garantindo a consistência dos dados que estão gravados, representando sempre um estado válido. Este estado também pode ser determinado pelas regras e lógica de negócios programadas em um contrato inteligente.

3.3 Isolamento

No caso de duas transações serem executadas simultaneamente seus efeitos devem ser produzidos de maneira isolada.

Segundo *Reis* (2018), com a propriedade do isolamento a execução concorrente permite deixar o banco de dados no mesmo estado em que ele estaria caso as transações fossem executadas em sequência.

Se duas transações concorrentes forem executas na *blockchain* tentando alterar os mesmos dados, o protocolo de consenso trata de garantir que apenas uma delas seja confirmada, dessa forma evita conflitos diretos entre as transações concorrentes. Isso não implica que as transações estão isoladas umas das outras, mas sim que a validação é feita de forma a evitar conflitos. As execuções podem ser tratadas em contratos inteligentes, que podem ser projetados para garantir que as transações sejam isoladas quando necessário.

3.4 Durabilidade

Uma vez que uma transação ocorreu com sucesso, seu efeito não poderá mais ser desfeito, mesmo em caso de falha. Esta propriedade está relacionada a capacidade de recuperação de falhas do SGBD. (LÓSCIO, 2011).

Essa característica define que uma vez que a transação tenha sido efetivada, permanecerá neste estado mesmo que ocorra um erro no servidor ou no sistema. Para isso as transações de um banco de dados relacional são gravadas em uma memória permanente, como em discos rígidos, de forma que os dados estejam sempre disponíveis mesmo após a instância do banco ser reiniciada.

Uma das propriedades mais bem definidas em sua forma de tratamento na *blockchain* é a durabilidade. Sabendo que todas as transações são registradas em um

livro-razão e espalhado entre todos os nós da rede, gravando todas as transações que ocorreram desde o início, criando um histórico completo de todas as atividades. Isso contribui para a durabilidade ao permitir que as partes interessadas verifiquem as transações passadas, mesmo que já tenham ocorrido há muito tempo.

Se um dos nós operantes na rede for desligado, os demais continuam operando normalmente, preservando os dados contidos no livro-razão, demonstrando a imutabilidade dos dados que estão.

Um nó que possui as informações não pode ser apagado e caso ele seja desconectado ou adulterado, os outros nós irão lidar com isso e manter a disponibilidade do sistema, pois ele não é centralizado, se possui nós com o banco de dados intacto, são eles quem mandam na decisão, ou seja, o sistema continua em pé como se nada tivesse acontecido (JUNIOR, 2019).

4 COMPARAÇÃO DA ARQUITETURA DE REDES

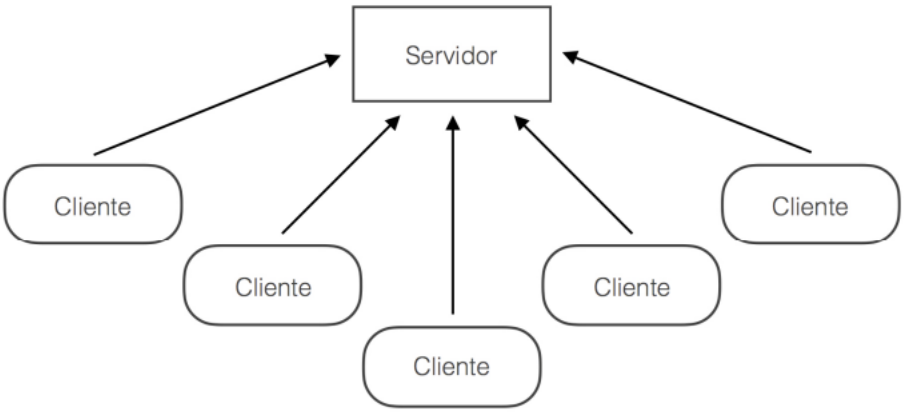
A principal diferença entre as duas abordagens de armazenamento de dados é a arquitetura disposta em suas redes. A característica de um banco de dados relacional é a sua arquitetura centralizada no servidor, entretanto em uma *blockchain* é utilizado o formato da rede *peer-to-peer* para seu funcionamento, sendo essa uma das suas principais propriedades que influenciam em sua disponibilidade e imutabilidade de dados.

4.1 Client-Server

Em uma rede cliente/servidor as funções de ambos são bem definidas. O servidor tem a função de fornecer algum serviço ou recurso para os seus clientes da rede, enquanto que o cliente tem a única função de utilizar os serviços e recursos oferecidos pelo servidor. (SENAI, 2012).

Este é o modelo padrão adotado pelos bancos de dados convencionais, transformando-se em um servidor de dados acessível para seus clientes. Esses clientes podem se enquadrar como aplicativos da web, programas desktop, aplicativos móveis ou qualquer outro dispositivo capaz de estabelecer uma conexão com o banco de dados.

Figura 2 – Arquitetura cliente-servidor

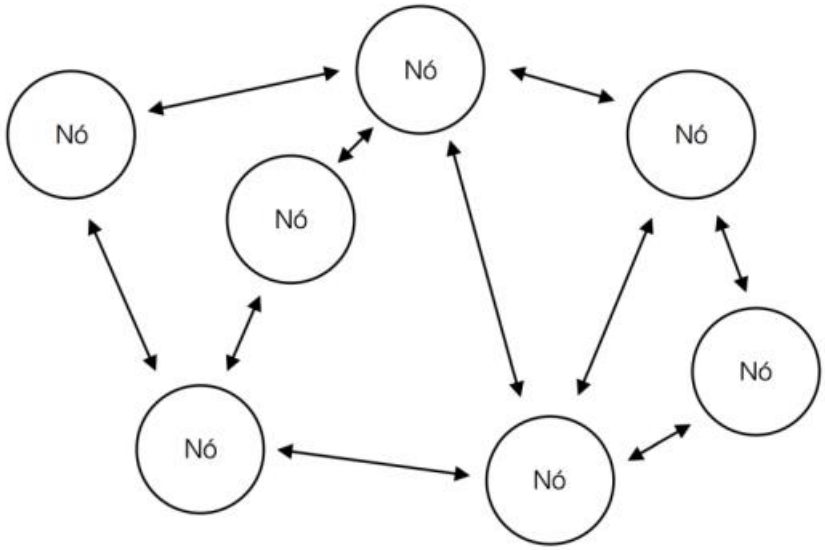


Fonte: adaptado Wikipédia.

4.2 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 3 – Rede Peer-to-Peer



Fonte: Adaptado Wikipédia

5 ESTRUTURAÇÃO DOS DADOS

A estruturação de dados em um banco de dados comum é organizada de forma hierárquica e tabular, utilizando tabelas, registros e campos para representar as informações de maneira organizada e eficiente. Por outro lado, no modelo de uma rede *blockchain* a estrutura de dados é organizada em um formato de árvore de merkle, que são estruturas de dados em forma de árvore na qual cada nó não-folha, denominado de "nó-galho," é o resultado de um *hash* de seus respectivos nós filhos. Cada nó-folha da árvore é o resultado de um *hash* de um conjunto de dados que, no caso da corrente de blocos, representam as transações da rede.

5.1 Tabular

Os dados estruturados, conforme mencionado anteriormente, apresentam formatos tabulares, onde as informações são organizadas em linhas (registros) e colunas (atributos). Esse formato, frequentemente encontrado em bancos de dados relacionais, é denominado de "tabela". Dentro de uma tabela, é possível armazenar uma variedade de subconjuntos e tipos de dados, abrangendo diferentes níveis de detalhes, além da opção de armazenamento de procedimentos, conhecidos como funções e procedures, que são um conjunto de comandos, que podem receber diferentes parâmetros, sendo destinado a uma ação programada.

Portanto, um banco de dados relacional é composto por um conjunto de procedimento e de tabelas interconectadas, nas quais os dados são organizados de maneira estruturada e relacionados entre si.

5.2 Árvore De Merkle

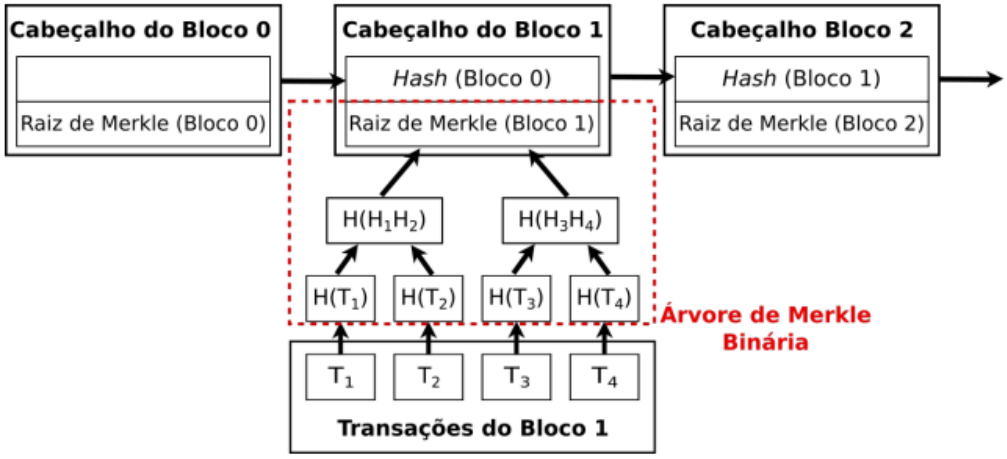
Nomeadas em homenagem a Ralph Merkle, que patenteou o conceito em 1979, às árvores de merkle, são divididas em várias camadas, cuja finalidade dentro da cadeia de blocos é a ligação criptográfica de todos os principais componentes do livro razão. Em seu conceito que relaciona os nós com uma única raiz 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.

O uso de árvores Merkle permite obter uma prova de Merkle (Merkle proof), que verifica a presença e a corretude de uma transação em um bloco de maneira eficiente. Para verificar uma transação, basta fornecer os hashes necessários para reconstruir o caminho da árvore correspondente à transação verificada

As árvores de Merkle são a principal estrutura auxiliar utilizada em correntes de blocos para verificar a integridade e não-repúdio de uma transação de maneira eficiente (Nakamoto, 2008).

Podemos analisar melhor visualmente como a estrutura é utilizada dentro da cadeia dos blocos, segundo a figura 4, exposta a seguir:

Figura 4 – Cadeia de blocos com árvore de merkle



Fonte: UFRJ

O fato de ser organizado 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.

6 CONTRATOS INTELIGENTES (SMART CONTRACTS)

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.

De acordo com Luciano (2018) o *smart contract*, é criado dentro de uma plataforma de aplicações digital. A mais conhecida é a plataforma digital descentralizada criada por Buterin (2014) e Wood (2014) chamada *ethereum*, que segundo seus criadores, é como um computador mundial que se aproxima de uma máquina virtual, denominada como Ethereum Virtual Machine (EVM), com uma linguagem de computação completa, também conhecida como linguagem de Turing, capaz de resolver diversos problemas usando linguagem computacional de script universal conhecida como solidity, contudo passível de futuro desenvolvimento e atualizações. Ainda segundo Buterin (2014), *ethereum* é utilizada para literalmente construir quaisquer atributos matematicamente descritíveis através do mecanismo de contratos. Referida linguagem permite aos desenvolvedores de todas partes do mundo escrever seus próprios contratos, distribuindo-os na rede descentralizada do *ethereum*. Por estarem na *blockchain*, eles se tornam imutáveis, motivo pela qual devem ser muito bem homologados antes de serem utilizados em um ambiente de produção, evitando falhas na execução.

Uma das principais vantagens dos contratos inteligentes é a capacidade de incluir variáveis que contêm informações relevantes das partes envolvidas em um processo. Imagine uma transportadora que utiliza a tecnologia *blockchain* para rastrear suas entregas. Isso possibilita a inclusão de variáveis essenciais, como número do produto, nome do motorista, endereço de entrega, tipo de produto e dados do cliente, entre outros.

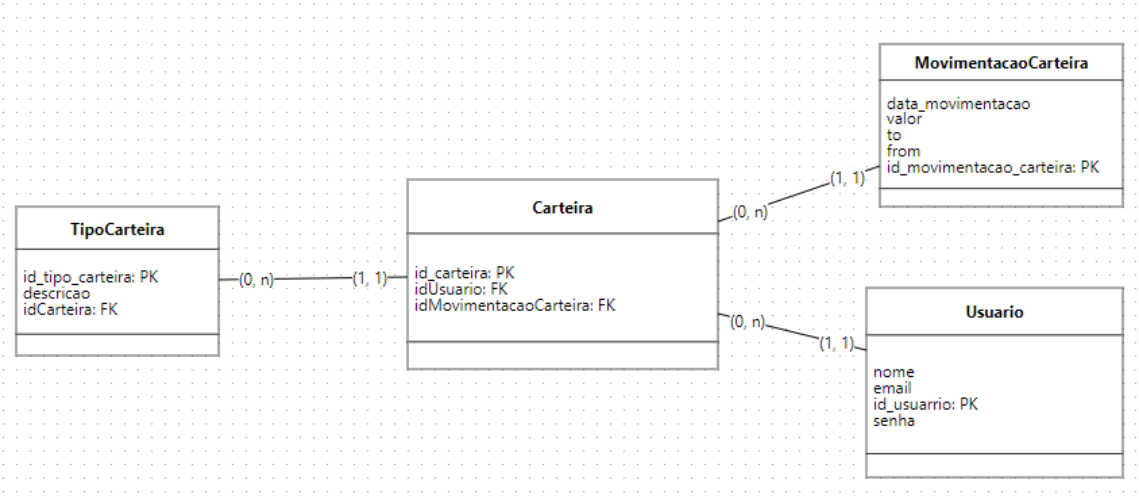
Essa flexibilidade oferece inúmeras utilidades que podem ser pré-programadas no contrato inteligente. Por exemplo, o contrato pode ser configurado para se executar automaticamente assim que o cliente receber o produto, removendo-o dos ativos de estoque e validando a transação. Essa abordagem proporciona um alto nível de segurança quando o contrato é bem elaborado e é visível para todas as partes envolvidas.

O aspecto transparente da *blockchain* permite que qualquer pessoa acompanhe o processo em tempo real e verifique o que está ocorrendo a qualquer momento. Isso promove a confiança e a transparência no processo, melhorando a eficiência e a segurança das transações.

7 MATERIAIS E MÉTODOS

Para o desenvolvimento do caso aplicado para o terceiro setor, foi utilizado uma comparação com um modelo conceitual de um banco de dados relacional, que tem a finalidade de registrar as transações entre as entidades do terceiro setor (usuário) e os doadores, como demonstra a figura 5.

Figura 5 – Modelo de comparação conceitual



Fonte: Elaboração própria

A execução deste modelo conceitual realizada na *blockchain* requer apenas a conexão com a rede e a biblioteca da web3, pois o propósito do protocolo é efetuar

transações de ativos entre carteiras. Durante o desenvolvimento deste projeto, foi utilizado a uma rede pública do *ethereum* conhecida como *goerli*, que pode ser criada através da infraestrutura provida pela *Infura*, com o auxílio do ambiente do *etherscan*, que simplifica a visualização das carteiras e transações que foram efetuadas, incluindo também o gerenciamento de contratos inteligentes. Para a integração com a web3 foi utilizado uma biblioteca com o mesmo nome, permitindo interações diretas com a rede *blockchain*. Para a interface do usuário, foi empregado as tecnologias *react* e Node.js para facilitar a realização de transferências de valores, sendo respectivamente um responsável pelo design e o outro para comunicação com o mongoDB, tendo como papel simular uma carteira na rede.

7.1 Implementação Da Rede Blockchain Com Infura

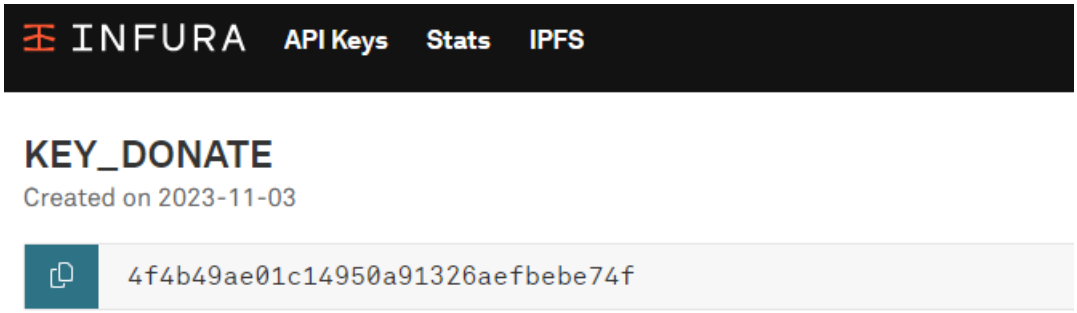
Infura é uma plataforma de infraestrutura *blockchain* que fornece serviços de infraestrutura para desenvolvedores e aplicativos que desejam interagir com redes *blockchain*, especialmente a *ethereum*. A plataforma *Infura* opera nós (nodes) na rede Ethereum e oferece uma interface fácil de usar para que os desenvolvedores possam se conectar à *blockchain* sem a necessidade de configurar e manter seus próprios nós.

Ao utilizar a Infura, os desenvolvedores podem enviar solicitações à rede *ethereum* sem precisar lidar com a complexidade de executar e manter um nó próprio. A Infura gerencia a infraestrutura, garantindo uma conexão confiável à rede *blockchain*. Isso é especialmente valioso para aplicativos e desenvolvedores que desejam se concentrar na lógica do aplicativo em vez de se envolver com detalhes técnicos de execução e manutenção de nós.

Assim, a Infura simplifica o desenvolvimento de aplicativos *blockchain*, oferecendo uma solução pronta para uso para acessar e interagir com a rede *ethereum*, permitindo que os desenvolvedores concentrem seus esforços na criação de funcionalidades e experiências de usuário, sem se preocupar com a operação de nós de *blockchain*.

A implementação de uma rede *blockchain* com *Infura* envolve o uso da *Infura* API, que é um serviço que fornece acesso a várias *blockchains*, para o projeto foi utilizado a rede *ethereum*. Para configurar uma aplicação ou projeto, primeiramente é necessário criar uma conta e obter uma chave de API (figura 6). Em seguida, você pode usar essa chave para se conectar à rede de sua escolha por meio dos end-points. Isso permite que você interaja com as redes, envie transações e acesse dados, sem a necessidade de executar e manter um nó local. A *Infura* simplifica o desenvolvimento de aplicativos *blockchain*, tornando o acesso à rede mais fácil e conveniente para os desenvolvedores.

Figura 6 – Chave infura



Fonte – Site Infura

7.2 Integração Web Com Web3

A biblioteca Web3 é uma ferramenta fundamental no desenvolvimento de aplicativos descentralizados (DApps) que interagem com a *blockchain ethereum*. Ela oferece uma interface programática para facilitar a comunicação entre aplicativos web e a rede *ethereum*, permitindo que desenvolvedores realizem operações como leitura e gravação de dados em contratos inteligentes, transações de *ether* e interações com a *blockchain*. As principais características da web3 são: conexão com a rede *ethereum*, interação com contratos inteligentes, transação e pagamentos, assinatura digital, gerenciamento de contas, eventos e logs, além de compatibilidade com diversas linguagens de programação.

No contexto apresentado, a biblioteca Web3 desempenhou um papel crucial ao facilitar a interação entre uma aplicação web específica e a rede *blockchain ethereum*, mais especificamente a *goerli*. A implementação dessa conexão é visualizada conforme a figura 7. Essa integração se revela fundamental, pois viabiliza a execução de operações como criação, leitura e atualização de registros na *blockchain* por meio de chamadas de função em contratos inteligentes. Esse processo de interação proporciona uma camada de transparência e acessibilidade às informações contidas na rede *blockchain*, tornando-as disponíveis e compreensíveis para as organizações envolvidas. Em essência, a utilização da biblioteca Web3 estabelece uma ponte eficiente e funcional entre a aplicação web e a infraestrutura da *blockchain*, permitindo uma gestão dinâmica e segura de dados na plataforma *ethereum goerli*.

Figura 7 – Conexão provedor web3

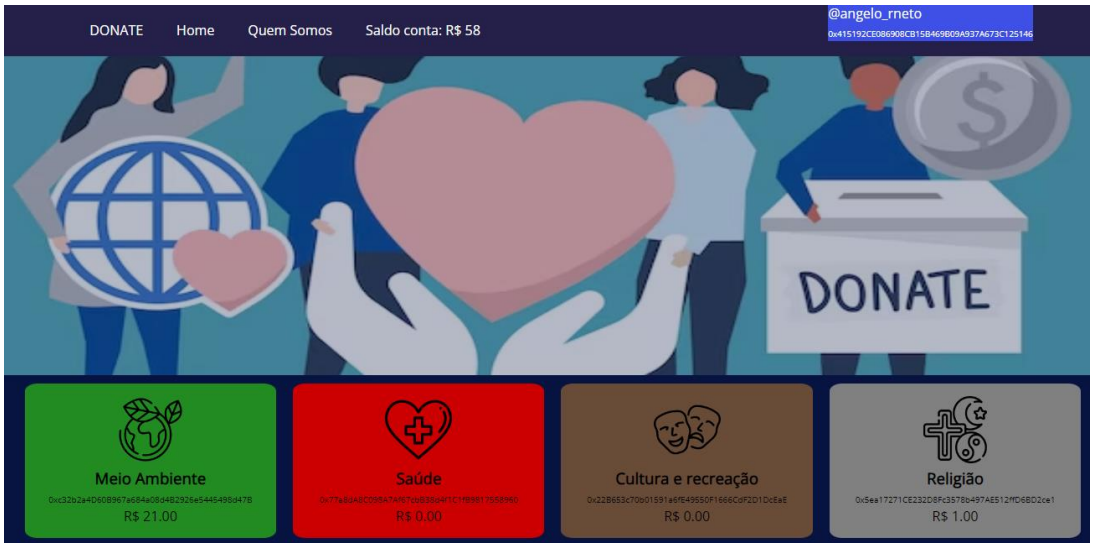
```
import Web3 from 'web3';

export default class CarteiraService {
    constructor() {
        var web3Provider =
        new Web3
        .providers
        .HttpProvider("https://goerli.infura.io/v3/4f4b49ae01c14950a91326aefbebe74f");
        this.web3 = new Web3(web3Provider);
    }
```

Fonte: Adaptado etherscan

7.3 Desenvolvimento da aplicação com React E Nodejs + MongoDB

A tecnologia *react* foi utilizada para criar uma interface de visualização para o usuário de forma amigável e dinâmica (figura 8). O *react* permite a construção de componentes reutilizáveis, proporcionando uma experiência de usuário consistente e intuitiva.

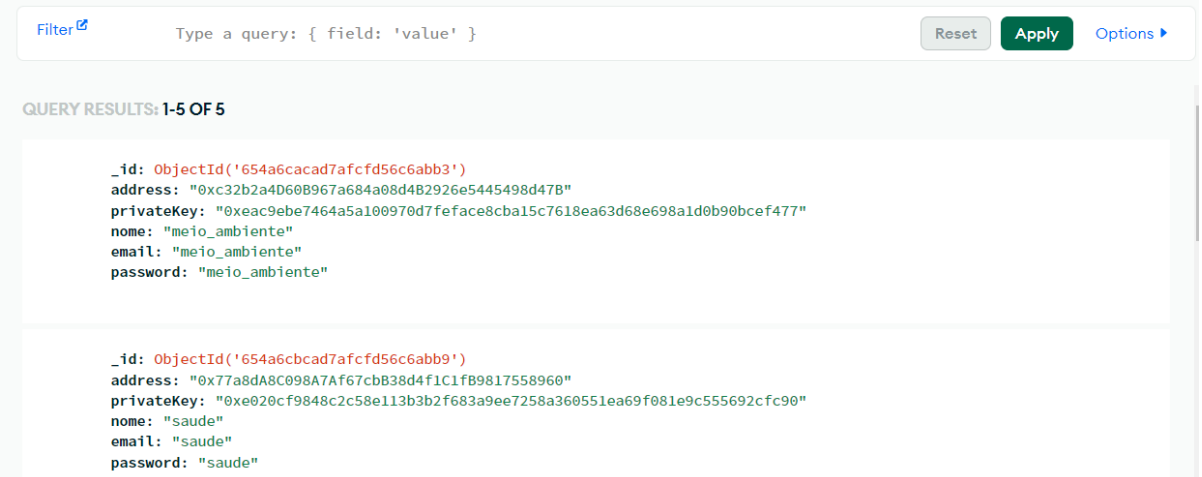
Figura 8 – Interface Donate



Fonte: Elaboração própria

O Node.js foi adotado para o desenvolvimento do lado do servidor, responsável por gerenciar as carteiras que são utilizadas para acesso à rede *blockchain*, os valores como o endereço público, endereço privado, usuário, senha e email são armazenados no MongoDB, para que a interface faça a manipulação das transferências entre os endereços salvos, fornecendo um ambiente de execução eficiente e suportando a comunicação entre o cliente e a rede *blockchain,* como fazem as carteiras dentro da web3, como por exemplo metamask e coinbase.

Figura 9 – Lista de carteiras MongoDB



Fonte: MongoDB

Entretando os valores do saldo salvos para cada carteira, está armazenado dentro da rede ethereum, e para visualização de saldos na rede foi utilizado o *etherscan* que é uma plataforma online que atua como um explorador de *blockchain* para a *ethereum*. Sua principal função é fornecer uma interface de usuário amigável e acessível para explorar, analisar e verificar transações, endereços, contratos inteligentes e outros dados relacionados à *blockchain ethereum*. Assim como demonstra o saldo na figura 10, referente à categoria de meio ambiente do terceiro setor.

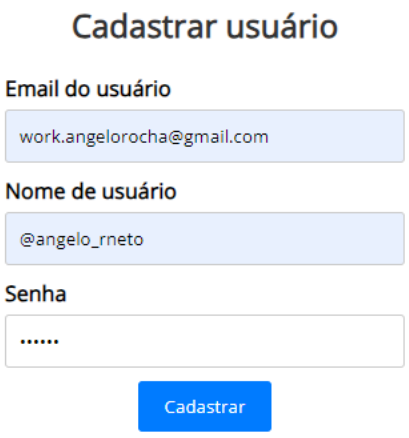
Figura 10 – Carteira meio ambiente

Fonte: Site Etherscan

7.4 Envio de doações para uma categoria do terceiro setor

Para demonstrar as transações na rede, criamos quatro carteiras representando instituições do terceiro setor, cada uma vinculada a uma categoria específica: Meio ambiente, saúde, cultura e recreação e religião. Inicialmente, todas essas carteiras possuíam um saldo de zero ETH.

Figura 10 – Cadastro de usuário



Fonte: Elaboração própria

Conforme a figura 10, foi registrado um usuário com um saldo inicial de 0.9 ETH, tal saldo foi gerado a partir da mineração da criptomoeda em questão foi

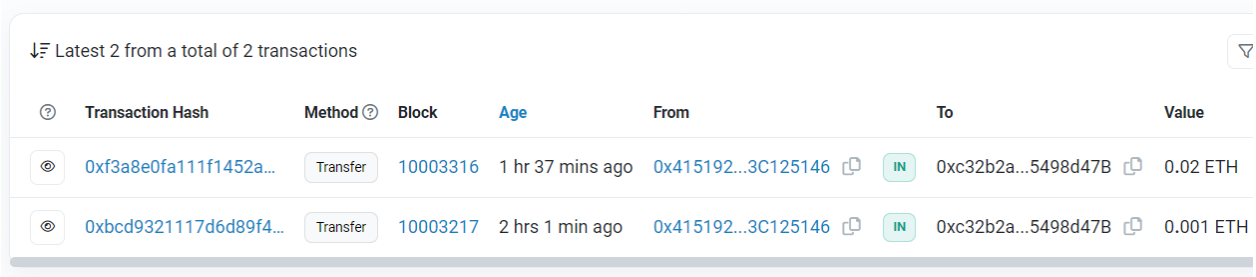
minerada anteriormente através de uma plataforma online chamada faucet, a carteira com o saldo está sendo identificado pela seguinte hash: 0x415192CE086908CB15B469B09A937A673C125146. A transação foi executada através da interface que permitia a seleção de uma das categorias mencionadas anteriormente, para efetuar doações dentro da rede Ethereum. Todos os saldos e valores transacionados foram convertidos em reais para uma melhor compreensão.

Figura 11 – Doação para o meio ambiente

Fonte: Elaboração própria

Foram realizadas duas transações destinadas à categoria "Meio ambiente," identificada pela hash: 0xc32b2a4D60B967a684a08d4B2926e5445498d47B. A primeira transação foi de R$ 1 real, seguida por uma segunda de R$ 20 reais (figura 11), resultando no registro da cadeia de blocos conforme mostrado na figura 12.

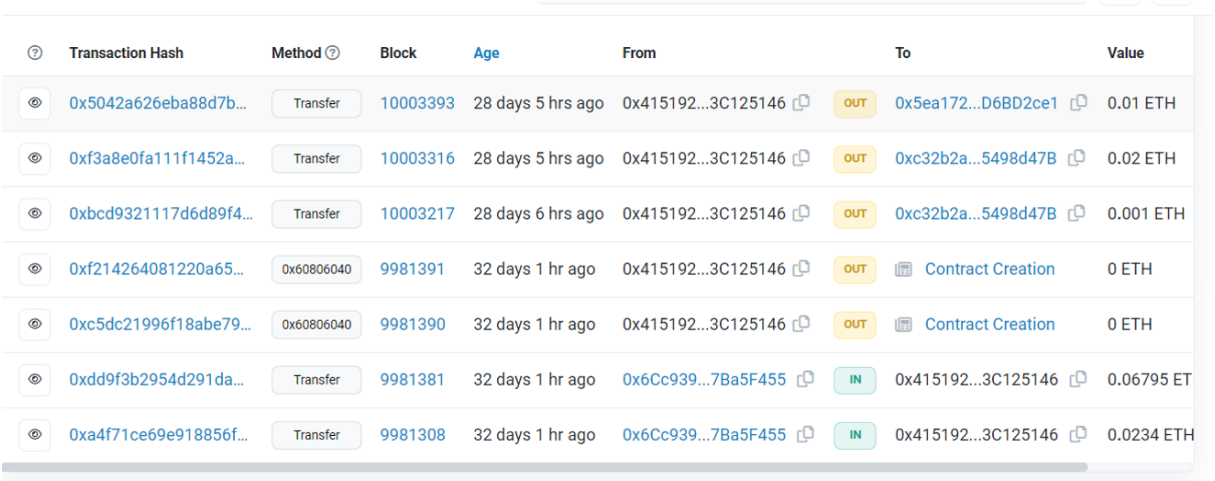
Figura 12 – histórico de transações meio ambiente



Fonte: Site Etherscan

Foi possível validar através da interface que após a efetivação da transação, o saldo da categoria de “Meio ambiente” passou a ser positivo, com o valor de R$ 21 reais, além de analisar com ampla transparência todas as movimentações de entrada e saída de saldo dentro da carteira principal criada para realizar as doações para as entidades conforme demonstra a figura 13.

Figura 13 - histórico de transação carteira de teste



Transaction Hash	Method	Block	Age	From		To	Value
0x5042a626eba88d7b...	Transfer	10003393	28 days 5 hrs ago	0x415192...3C125146	OUT	0x5ea172...D6BD2ce1	0.01 ETH
0xf3a8e0fa111f1452a...	Transfer	10003316	28 days 5 hrs ago	0x415192...3C125146	OUT	0xc32b2a...5498d47B	0.02 ETH
0xbcd9321117d6d89f4...	Transfer	10003217	28 days 6 hrs ago	0x415192...3C125146	OUT	0xc32b2a...5498d47B	0.001 ETH
0xf214264081220a65...	0x60806040	9981391	32 days 1 hr ago	0x415192...3C125146	OUT	Contract Creation	0 ETH
0xc5dc21996f18abe79...	0x60806040	9981390	32 days 1 hr ago	0x415192...3C125146	OUT	Contract Creation	0 ETH
0xdd9f3b2954d291da...	Transfer	9981381	32 days 1 hr ago	0x6Cc939...7Ba5F455	IN	0x415192...3C125146	0.06795 ET
0xa4f71ce69e918856f...	Transfer	9981308	32 days 1 hr ago	0x6Cc939...7Ba5F455	IN	0x415192...3C125146	0.0234 ETH

Fonte: Site Etherscan

A partir disso podemos concluir que foi simulado um caso de doação entre usuário e instituição enquadrada como uma MROSC dentro de uma categoria, aonde é possível identificar cada uma através do seu respectivo endereço, mapeado como carteiras de uma rede blockchain, trazendo um melhor rastreio, segurança e descentralização das informações armazenadas na rede.

Observamos que a blockchain retém as informações essenciais para rastrear as transações efetuadas, armazenando no livro-razão os dados contidos em cada bloco, incluindo a hash da transação, o número do bloco, as carteiras de origem e destino, bem como o valor da transação. Isso confere às transações dentro da rede um alto grau de consistência e rastreabilidade, garantindo que cada movimento de ativos seja transparente e facilmente acompanhado ao longo do tempo.

8 RESULTADO E DISCUSSÃO

Após realizar uma análise comparativa entre as tecnologias *blockchain* e bancos de dados relacionais, destacou-se uma oportunidade para gerir ativos, tais como dinheiro, terrenos, imóveis e outros bens de valor, por meio de uma rede imutável e consistentemente operando de maneira descentralizada, apresentando um contraste notável em relação aos convencionais bancos de dados.

No acesso à rede *blockchain*, torna-se essencial possuir uma carteira que represente um endereço único. Neste projeto específico, essa funcionalidade foi emulada com valores armazenados em um banco de dados não relacional. No entanto, em um cenário real, a conexão com a carteira é estabelecida por meio de um provedor, como o *metamask*, permitindo a realização de transações entre esses endereços.

É importante observar que, embora tenham sido executadas as operações básicas de um banco de dados relacional comum, como a inserção de um novo

registro, e sua visualização, a exclusão e alteração não foi viabilizada neste contexto abordado, devido à natureza de imutabilidade do livro-razão da *blockchain*.

Apesar das vantagens oferecidas pela rede *blockchain,* como sua descentralização, é crucial ressaltar que a mesma requer uma biblioteca intermediária, atuando como facilitador para a execução de procedimentos de transação, listagem de carteiras, busca de blocos e outras operações originadas da web3. Diversas bibliotecas, a exemplo de *ethers* e *crypto*, podem ser exploradas com este propósito, ampliando as possibilidades de desenvolvimento e integração neste inovador ambiente tecnológico.

Em um contexto de banco de dados convencional, estamos vulneráveis a modificações ou eliminações não autorizadas, o que representa uma ameaça à integridade dos dados. Isso contrasta significativamente com os registros armazenados em uma estrutura baseada em *blockchain*, onde a imutabilidade dos dados desempenha um papel crucial na preservação da integridade.

Ao relacionar essa característica ao cenário abordado de doações públicas para uma entidade, torna-se evidente a extrema importância da imutabilidade. Pois existe a possibilidade de pessoas mal-intencionadas que possam tentar desviar os ativos arrecadados, destacando a necessidade crítica de uma infraestrutura que assegure a inviolabilidade dos registros.

Outro aspecto notável é a transparência das transações na blockchain. Todas as operações podem ser rastreadas dentro da rede, sem qualquer possibilidade de censura, graças à distribuição da validação entre os diversos nós da rede. Em contraste, um banco de dados convencional está mais suscetível a interferências externas ou manipulação centralizada, o que compromete a transparência e a confiabilidade das transações. Essas distinções destacam a superioridade da *blockchain* em garantir a integridade, segurança e visibilidade em operações, especialmente em contextos sensíveis, como doações públicas.

Essas vantagens tornam a *blockchain* uma escolha atraente para cenários em que a confiança, segurança e transparência são essenciais, especialmente em setores como finanças. No entanto, é importante notar que cada tecnologia tem seu lugar e aplicação específicos, dependendo dos requisitos do projeto. É possível utiliza-las em conjunto aproveitando o melhor de suas características para atender o mercado.

9 CONCLUSÃO

É evidente que a tecnologia blockchain introduz uma abordagem inovadora para o armazenamento de dados, compartilhando semelhanças com os princípios ACID de um banco de dados relacional tradicional. No entanto, sua aplicação é mais eficaz em cenários que envolvem o armazenamento de ativos de valor, como demonstrado neste projeto.

A arquitetura de blockchain revelou-se altamente eficaz na armazenagem e transferência de valores entre entidades, no contexto doadoras e instituições do terceiro setor. Manteve a consistência dos dados armazenados e proporcionou transparência por meio do registro das atividades entre as carteiras. No entanto, surgiram desafios na gestão de contratos inteligentes, que desempenham um papel

crucial na execução de procedimentos das carteiras. A capacidade de incorporar mais dados diretamente na rede através desses contratos seria vantajosa.

Assim, podemos concluir que a tecnologia blockchain se apresenta como uma nova ferramenta capaz de promover transparência, descentralização e consistência em relação aos valores que representam ativos. Ela pode ser empregada em conjunto com um banco de dados relacional, sem a necessidade de eliminar completamente suas funcionalidades, resultando em uma complementação valiosa.

10 REFERÊNCIAS

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

AZEVEDO, I.C; JUNIOR, C.R.A. **Hyperledger: Descentralizando Informações de Maneira Segura com Blockchain UNISUL.** 2019. Disponível em: https://repositorio.animaeducacao.com.br/bitstream/ANIMA/10950/1/TCC2%20-%20Blockchain%20-%20Cap1-2-3-4-5%20-%206%20-%20Carlos%20e%20Igor%20-%20v27.pdf Acesso em: 24 set. 2023.

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

BUTERIN, V.**A next-generation smart contract and decentralized application platform.** 2014. Disponível em: https://www.weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

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

DINH, T. T. A., Wang, J. et al.: **Blockbench: A framework for analyzing private blockchains. Em Proc. of the ACM International Conference on Management of Data.** Disponível em: https://www.comp.nus.edu.sg/~ooibc/blockbench.pdf Acesso em: 15 set. 2023.

LÓSCIO B.F.; OLIVEIRA, H.R.; PONTES, J.C.S. **NoSql no desenvolvimento de aplicações Web colaborativas.** 2016. Disponível em: https://www.researchgate.net/profile/Bernadette_Loscio/publication/268201466_NoSQL_no_desenvolvimento_de_aplicacoes_Web_colaborativas/links/576aa72008aef2a864d1ef8c/NoSQL-nodesenvolvimento-de-aplicacoes-Web-colaborativas.pdf. Acesso em: 23 set. 2023.

LUCIANO R.B.S**. Aplicação da Smart Contract nos Contratos de Gás Natural: Uma Análise Exploratória.** Instituto Brasileiro de Mercado de Capitais, 2018.

NAKAMOTO, S. **Bitcoin: A peer-to-peer electronic cash system.** 2008. Disponível em: https://www.ussc.gov/sites/default/files/pdf/training/annual-national-training-seminar/2018/Emerging_Tech_Bitcoin_Crypto.pdf

REIS, F. **Conceito de Banco de Dados: O que significa ACID**. 2018. Disponível em: http://www.bosontreinamentos.com.br/bancos-de-dados/conceitos-de-bancos-de-dados-o-que-significa-acid/ Acesso em: 23 set. 2023.

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

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

SECRETARIA NACIONAL DE ASSISTÊNCIA SOCIAL. **Marco Regulatório das Organizações da Sociedade Civil – MROSC.** Disponível em: https://blog.mds.gov.br/redesuas/regulacao/mrosc/. Acesso em: 23 set. 2023.

SENAI. **Série tecnologia da informação – Hardware Arquitetura de Redes.** 2012. Disponível em: https://professorleonardomello.files.wordpress.com/2013/03/arquit_redes.pdf Acesso em: 24 set. 2023.

STALLING, W. **Criptografia e segurança de redes: princípios e práticas.** 6. ed. São Paulo: Pearson Education do Brasil, 2015.

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

TAPSCOTT, D; Tapscott, A. **Blockchain Revolution: como a tecnologia por trás do bitcoin está mudando o dinheiro, os negócios e o mundo.** SENAI - São Paulo, 2017.

WOOD, G. **Ethereum: A secure decentralised generalised transaction ledger. Byzantium Version.** 2017. Disponível em: https://ethereum.github.io/yellowpaper/paper.pdf