Mecanismos de Interoperabilidade em *Blockchains*: Um Comparativo de Custo de Transações Cross-chain para Tokens ERC-20

Ronan D. Mendonça[1], Ítallo W. F. Cardoso[1], Rafael Coelho[3], Josué N. Campos[1]
Glauber D. Gonçalves[2], Alex B. Vieira[3], José A. M. Nacif[1]

[1]Universidade Federal de Viçosa (UFV) – Florestal, MG – Brasil
[2]Universidade Federal do Piaui (UFPI) – Picos, PI – Brasil
[3]Universidade Federal de Juiz de Fora (UFJF) – Juiz de Fora, MG – Brasil

{ronan.dutra,itallo.cardoso,josue.campos,jnacif}@ufv.br

ggoncalves@ufpi.edu.br

{alex.borges,rafael.coelho}@ufjf.edu.br

Abstract. *Providing data interoperability between blockchains is a challenge for most applications with this feature as a requirement. The number of existing platforms with different types of protocol implementations makes it difficult to achieve full interoperability. This work presents the importance of interoperability for the blockchain ecosystem and the definitions of the mechanisms used to communicate among them. Therefore, we structured, implemented, and experimented with the Notarial and Hash-Time Lock Contract interoperability mechanisms to obtain and present the costs and complexity of interoperating an ERC-20 token. The results show the complexity of achieving interoperability through the mechanisms listed and demonstrate the fixed costs of the mechanisms.*

Resumo. *Prover a interoperabilidade de dados entre blockchains é um desafio para a grande maioria das aplicações que têm esta característica como requisito. A quantidade de plataformas existentes e com variados tipos de implementações de protocolos dificultam a realização plena da interoperabilidade. O objetivo deste trabalho é apresentar a importância da interoperabilidade para o ecossistema de blockchains e as definições dos mecanismos utilizados para prover a comunicação entre elas. Sendo assim, estruturamos, implementamos e experimentamos os mecanismos de interoperabilidade Notarial e Bloqueio de Hash, de modo a obter e apresentar os custos e a complexidade de interoperar um token ERC-20. Os resultados apresentam o nível de complexidade de se obter a interoperabilidade por meio dos mecanismos elencados e demonstram os custos fixos dos mecanismos.*

1. Introdução

As *Blockchains* se tornaram uma tecnologia extremamente promissora, uma vez que armazenam registros de transações de forma distribuída, podendo ser compostas por um grande número de participantes e sem a necessidade de um controle centralizado. O trabalho de [Nakamoto et al. 2008] sugeriu o termo *blockchain* para sua implementação de um livro-razão da moeda digital Bitcoin. Esta implementação de uma tecnologia para

evitar que uma certa quantidade de moeda fosse gasta mais vezes em situações distintas. Desde então, uma imensa variação deste conceito vem sendo desenvolvida e muitas outras ainda estão sendo idealizadas. Os que se conectam a uma rede *blockchain* obtém uma cópia completa dos dados armazenados e realiza operações de validação e transmissão das transações aos demais nós. Uma vez que a informação é registrada, ela não poderá sofrer alterações, o que resulta em confiabilidade e permite uma fácil auditoria. Todas as informações inseridas em uma *blockchain* ficam organizadas em blocos encadeados de dados por meio de *hashes* criptográficos.

A tecnologia *blockchain* pode ser aplicada em diversas áreas resultando em soluções de problemas como validação, integridade e interoperabilidade de dados. Os registros de dados em *blockchain* pode oferecer maior transparência, controle de acesso e segurança das suas transações [Gordon and Catalini 2018]. Um dos avanços alcançados pelo desenvolvimento da tecnologia *blockchain* foi o conceito de contratos inteligentes. Eles aumentam as possibilidades e diversidades de uso das *blockchain* e incrementam no seu funcionamento o poder de programar uma ação desejada. Os contratos são escritos em uma linguagem de programação específica para a implementação como scripts na rede *blockchain*, como por exemplo o Solidity. As regras dos contratos são executadas pela rede da forma como foram pré-estabelecidas. Devido ao caráter complementar que os contratos possuem em relação à *blockchain*, eles se tornaram parte essencial das *blockchains* apesar de terem surgido após o Bitcoin, proposto por [Nakamoto et al. 2008]. Com a utilização desses contratos, as *blockchains* tiveram sua capacidade aprimorada. Um contrato permite definir um comportamento para um determinado estado e atender necessidades de aplicações diversas. Nesse sentido, os contratos inteligentes são capazes de executar transações muito mais complexas dentro da *blockchain*.

As *blockchains* são classificados basicamente pela maneira em que se apresenta as condições de acesso a ela e de acordo com a forma de participação dos nós da rede no consenso e proteção das transações armazenadas. Os tipos principais de *blockchains* encontradas são as com acesso público (não permissionadas) e as com necessidade de permissão para acesso (permissionadas). As especificações de protocolo de consenso, proteção de acesso às informações e controle da forma de distribuição dos dados em nós também diferenciam estes tipos de *blockchains*. Em uma *blockchain* permissionada, as respostas são mais rápidas e seguras, porém o controle é exercido por proprietários específicos e os nós precisam de permissão para ingressarem na rede. Nas *blockchains* do tipo pública ou não permissionada, por sua vez, são consideradas descentralizadas e pode conter vários nós desconhecidos que armazenam os dados das transações e qualquer nó pode se ingressar à rede por meio de sincronização. Porém, apenas nós sincronizados são utilizados para participar do consenso da rede [Wang and Feng 2018] [Rouhani and Deters 2017].

A partir dessas características, diversas aplicações podem ser desenvolvidas baseadas no uso de contratos inteligentes como, por exemplo, aplicações financeiras (gerenciamento de moeda, serviço de garantia, procedimentos de auditoria, empréstimo), aplicações médicas (gestão de informações de saúde, proteção de dados de pesquisa clínica, monitoramento e tratamento automatizado de pacientes, gerenciamento de identidade e controle de acesso, proteção de dados de identidade), aplicações imobiliárias, aplicações de acordos contratuais, aplicações de Internet das coisas, aplicações de serviços de telecomunicações, aplicações de gestão de logística, além de aplicações entre

diferentes indústrias [Hewa et al. 2021].

Uma importante aplicação de utilização das *blockchains*, por meio dos contratos inteligentes, é a representação digital de um ativo chamada de *token*. O Token simboliza um valor, que também pode pode ser utilizado na representação de objetos e valores monetários. A estrutura de um *token* é realizada fundamentalmente em um contrato inteligente. Esta estrutura contém basicamente um mapa de endereços de contas e seus saldos vinculados. Os padrões de *tokens* provêm uma forma padronizada de questões técnicas a serem seguidas. Dentre estes padrões existe o *Ethereum Request for Comments* - ERC-20, que proporciona recursos para a criação de tokens que podem ser negociados por meio de transferências. Os tokens no padrão ERC-20 são os de maior aplicabilidade em relação às transferências que requerem a interação de mais de uma plataforma *blockchain*. A comunicação entre *blockchains* não acontece de forma efetiva, padronizada e tão pouco nativa das plataformas, fazendo com que tenham que ser criadas soluções específicas para cada aplicação.

Apesar das diversas aplicações da tecnologia *blockchain*, existe vários desafios. Dentre estes desafios está o desafio da interoperabilidade de dados entre *blockchains*, uma vez que há várias plataformas *blockchain* independentes. Além disso, muitas destas plataformas implementam tipos de contrato inteligente e protocolos de consenso diferentes, dificultando assim a interoperabilidade. É importante superar este obstáculo para que o uso das aplicações possa ser realizado em múltiplas plataformas.

Nosso estudo tem o objetivo de implementar e experimentar mecanismos de interoperabilidade para obter e apresentar o custo de interoperar um token ERC-20 entre redes. Inicialmente apresentamos a relevância da interoperabilidade para o ecossistema de *blockchains* e as definições dos mecanismos utilizados para prover interoperabilidade entre *blockchains*, o *cross-chain* ou cadeia cruzada. Além disso, este trabalho apresenta uma avaliação experimental da arquitetura de dois mecanismos *cross-chain*, tendo em vista suas respectivas formas de atuação nas plataformas públicas compatíveis com a rede Ethereum. Esta avaliação evidencia os procedimentos de cada mecanismo para realizar a transferência entre duas redes *blockchain* distintas e o custo para cada transação utilizada em uma das etapas do mecanismo.

As próximas seções são organizadas como segue: a Seção 2, apresenta os *tokens* e suas utilidades. A Seção 3 apresenta a importância de prover a interoperabilidade entre *blockchains*. A Seção 4 aborda os mecanismos *cross-chain*. A Seção 5 mostra a metodologia utilizada, as avaliações experimentais considerando os mecanismos *cross-chain* experimentados. O detalhamento dos resultados alcançados são apresentados na Seção 6. Finalmente, a Seção 7 expõe as considerações finais.

2. Tokens

Os *tokens* representam, na tecnologia, um dispositivo ou sistemas que geram codificações de acesso e autenticação. Em se tratando de *blockchain* são atribuídos à representação digital de um ativo. Portanto é utilizado para especificar a criação de um registro digital de um ativo tangível, como por exemplo o ouro, dinheiro, obras de arte, imóveis ou equipamentos. Um ativo intangível pode ser também software, criptomoedas, artes digitais, etc. Sendo assim, os ativos tangíveis ou intangíveis podem ser protegidos com a ajuda da tecnologia *blockchain* [Mendonça et al. 2022].

Existem, portanto, vários tipos de *tokens* que se distinguem de acordo com a proposta a qual forem designados. Dentre eles os principais tipos são: *tokens* fungíveis e não fungíveis. Um *token* fungível é idêntico a todos os outros *tokens* em valor, para a mesma classe de *tokens* que desempenha a função de moeda digital para pagamentos e transferências. Os principais exemplos são o Bitcoin e o Ether. Já os *tokens não fungíveis* representam algo único, isto é, não pode ser trocado por outro de mesmo valor porque é exclusivo dentro de uma determinada classe de *token*. Este token é uma representação de algo por meio de registro do objeto e seu proprietário [Mendonça et al. 2022].

A estrutura de um *token* é definida por um contrato inteligente que contém um mapa de endereços de contas e seus saldos vinculados. As operações sobre esta estrutura discorrem em transferências da propriedade destes *tokens*. Ao realizar esta transferência, os contratos atualizam o saldo das duas contas envolvidas, creditando-o em uma das contas e debitando na outra.

Os padrões de *tokens* são documentos submetidos como Solicitação de Comentários Ethereum ou *Ethereum Request for Comments* (ERC) que é a forma de definir padrões de uso na Ethereum. Existem vários ERCs aprovados como padrões e dentre eles temos o padrão ERC-20 que trata dos *tokens* fungíveis e que são utilizados nos experimentos deste trabalho. Existem vários outros padrões de *tokens* que também poderiam ser utilizados nos experimentos e são apresentados na Tabela 1.

Tabela 1. Padrões ERC para *tokens* fungíveis e não fungíveis.

Padrão de Token	Definições
ERC-20	Baseado em oferta de criptomoedas e contabilizados em valores de saldos. Tem valor fungível.
ERC-721	Cada elemento é único, fazendo registro de propriedade de ativos colecionáveis e virtuais. Tem valor não fungível.
ERC-1155	Permite que diferentes *tokens* sejam configurados de um único ponto. Conjunto de ativos heterogêneos. Combinação dos padrões ERC-20 e ERC-721.

Vários estudos tratam das pesquisas relacionadas à segurança nos ambientes voltados aos contratos inteligentes e padrões, discutindo a importância e constante necessidade de evolução deste aspecto [Ndiaye and Konate 2021]. Os prejuízos provocados por erros e falhas nos contratos inteligentes podem ser enormes. Portanto, eles têm que ser testados para garantir a precisão de suas funções, além de passar por varredura de possíveis vulnerabilidades. A criação dos contratos devem observar as boas práticas para que não sejam inseridos problemas de seguranças comumente encontrados e explorados por usuários mal intencionados. Alguns exemplos destes problemas são dependência de ordem de transação, *timestamp* e de terceiros [Esquivel et al. 2023].

2.1. Padrão ERC-20

O ERC-20 é um padrão de contratos inteligentes proposto por [Vogelsteller and Buterin 2015] e que permite a criação de *tokens*. Deve ser implementado a interface do padrão para ser possível utilizar as funcionalidades definidas como as de transferência de *tokens* entre contas, verificação do saldo da conta, verificação

do total de *tokens* criados e aprovação de *tokens*. A Listagem 1 apresenta a interface do padrão ERC-20 com alguns dos métodos utilizados por Aplicações Descentralizadas (Dapps).

```
1  // SPDX-License-Identifier: UNLICENSED
2  pragma solidity ^0.4.20;
3  interface ERC20{
4  event Transfer(address, address, uint256)
5  event Approval(address, address, uint256)
6  function name() public view returns (string)
7  function symbol() public view returns (string)
8  function decimals() public view returns (uint8)
9  function totalSupply() public view returns (uint256)
10 function balanceOf(address) public view returns (uint256)
11 function transfer(address, uint256) public returns (bool)
12 function transferFrom(address, address, uint256) public returns (bool)
13 function approve(address, uint256) public returns (bool)
14 function allowance(address, address) public view returns (uint256)
15 }
```

Listagem 1. Padrão para *token* fungível: ERC-20

Os *tokens* ERC-20 são negociados de forma similar às criptomoedas e outros tipos de *tokens*. O valor monetário de um token depende geralmente de sua relação fora da *blockchain* e a forma que é negociado. [Rogers et al. 2022]. Quando ocorre uma transferência de *tokens*, por exemplo do tipo ERC-20, são submetidos ao consenso da rede o débito e o crédito nos saldos das contas envolvidas na transação. A Figura 1 apresenta a função principal executada na transferência de valores entre contas.

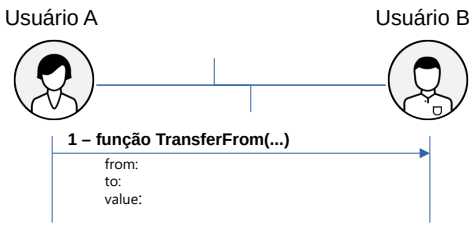

Figura 1. Sequência para transação de transferência de *token* ERC-20.

Neste exemplo, os usuários A e B realizam uma transferência de valor de um *token* entre eles. O usuário A é detentor de saldo do *token* e o usuário B é quem receberá o valor transferido do *token*. A transferência do *token* só se dá por meio da execução da função ***transferFrom*** acionado pelo proprietário do *token*. Esta função recebe os dados de identificação do atual proprietário, identificação do usuário para quem vai ser transferido o *token*, juntamente com o valor a ser transferido do *token* no contrato.

3. Interoperabilidade

As plataformas *blockchain* executam diferentes conjuntos de transações por meio de implementações distintas e normalmente isoladas. Devido a esta heterogeneidade, há uma grande dificuldade em compartilhar informações através de várias redes *blockchain*. Estas características trouxeram em evidência o estudo de tecnologia para prover a interoperabilidade da *blockchain*. Interoperabilidade é definida como a capacidade de haver cooperação entre partes envolvidas em uma solução, ainda que utilizem diferentes tecnologias [Wegner 1996].

A tecnologia *cross-chain* é o envolvimento de um par de *blockchains* em que um tipo de aplicação descentralizada facilita a transferência de ativos de uma *blockchain* para outra, provendo a interoperabilidade. Isso significa que é possível mover ativos, como criptomoedas e *tokens*, de uma *blockchain* para outra sem a necessidade de intermediários centralizados. Uma das maneiras de implementar *cross-chain* envolve a criação de pontos de conexão entre as *blockchains*, conhecidos como "pontes". As pontes permitem que as transações sejam validadas em ambas as *blockchains*, garantindo a segurança e a integridade dos ativos transferidos. De maneira geral, as transferências de ativos entre cadeias seguem um procedimento atômico, baseado no protocolo *Cross-chain communication protocol (CCCP)* em que há o bloqueio de um ativo na origem, responsabilidade de transferência e a criação do ativo no destino [Belchior et al. 2021].

[Qasse et al. 2019] apresenta diversas aplicações de interoperabilidade em redes *blockchain*. Dentre elas estão as transferências de ativos de uma aplicação ou cadeia para outra, troca de ativos entre dois usuários em diferentes redes *blockchain* de maneira segura e atômica, bloqueio de ativos e liberação em outra cadeia mediante condições, uso de oráculos entre cadeias quando é necessário utilizar dados para executar uma ação em cadeia diferente. Contratos em que há dependência de dados de várias cadeias para que uma ação seja acionada. Existem ainda alguns desafios para atingir a interoperabilidade. Dentre eles estão a necessidade de garantia de atomicidade, melhoria na eficiência de manutenção da segurança, tolerância à diversificação de plataformas, bem como facilidade para desenvolvedores de aplicações [Jin et al. 2018].

Os tipos de interoperabilidade em *blockchain* mais comuns encontrados são: interoperabilidade entre *blockchains* (homogêneas), interoperabilidade entre dApps usando diferentes *blockchains* e interoperabilidade de *blockchains* com outras tecnologias *blockchains*(heterogêneas) [Besançon et al. 2019]. Assim, a Figura 2 demonstra um diagrama de fragmentação desses tipos e apresenta como são definidas as transações entre os tipos. As transações de *blockchains* (homogêneas), são nomeadas como uma transação *cross-chain* (CC-Tx), onde "CC"significa *cross-chain* e "Tx", transação. Uma transação *cross-blockchain* (CB-Tx) é uma transação entre diferentes *blockchains* (heterogêneos) e por fim uma aplicação descentralizada *cross-chain* (CC-dApp) é um dApp que utiliza as transações *cross-blockchain* para implementar seus requisitos [Belchior et al. 2021].

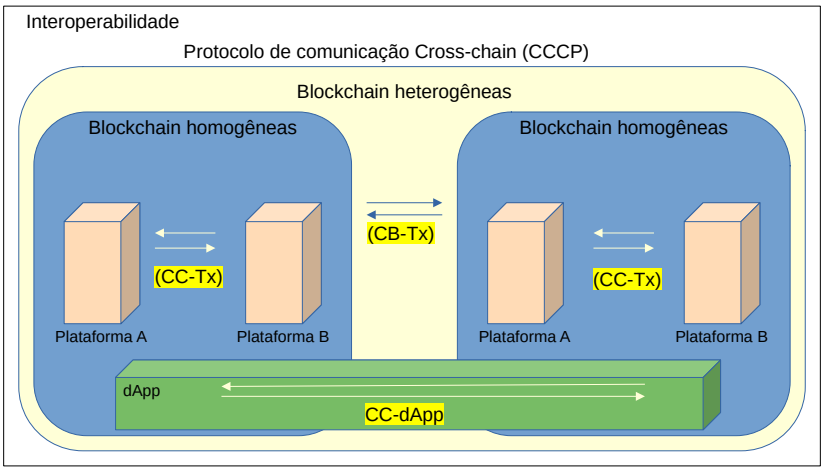

Figura 2. Tipos de interoperabilidade e transações em *blockchain*.

No contexto dos *tokens*, a interoperabilidade entre plataformas pode contribuir para a usabilidade, ao implicar a capacidade de transferir um ativo entre cadeias distintas, mantendo o estado e histórico consistentes. A interoperabilidade de cadeias deve atingir a eficiência de dois tipos, cada um dos quais trazendo considerações distintas porém contribuindo para a usabilidade. A troca de ativos digitais entre cadeias é um dos tipos de interoperabilidade. Ele deveria conter a capacidade de transferir e trocar ativos originários de diferentes cadeias sem intermediários confiáveis, como trocas centralizadas. Um exemplo disso seria tornar um *token*, originário de uma cadeia, válido em qualquer outra cadeia disponível. Outro tipo de interoperabilidade desejada se diz respeito a troca de informações que mantém a capacidade de fazer algo em uma cadeia que reflete em outra cadeia. Esta troca deve permitir o rastreamento não só de ativos ou itens negociáveis, mas também as operações executadas. Como exemplo, o compartilhamento do histórico de transações de um determinado item contendo negociações e proprietários.

4. Mecanismos *cross-chain*

Com tantas oportunidades de aplicativos de negócios com requisitos em interoperar *blockchains*, prover soluções com mecanismos genéricos de *cross-chain* para conectar redes *blockchain* homogêneas e heterogêneas, amplia o espaço de desenvolvimento da tecnologia *blockchain* [Buterin 2016]. Algumas soluções foram propostas para interoperar acesso a dados e transações entre *blockchains*, como o mecanismo notarial (*Notary mechanism*) e bloqueio de hash (*Hash-locking* ou *Hash time lock*) [Belchior et al. 2021]. Estes mecanismos propõem soluções que podem abranger um número maior de variedades de aplicações e distintas soluções de *blockchains*.

4.1. Mecanismo Notarial

O mecanismo notarial é uma forma de implementar a interoperabilidade entre cadeias de forma relativamente simples. Ele consiste em verificar e encaminhar mensagens entre cadeias por meio de uma entidade confiável intermediária chamada de notário. Quando há troca e transferência de ativos entre diferentes sistemas *blockchain*, uma ou mais organizações são designadas como notários para monitorar eventos entre as cadeias, e alcançar um consenso sobre a ocorrência do evento por meio de um algoritmo de consenso específico, e, por fim, responder de forma tempestiva [Belchior et al. 2021]. O mecanismo notarial se divide em mecanismo notarial de assinatura única e de múltiplas assinaturas [Ou et al. 2022].

O mecanismo notarial de assinatura única, também denominado mecanismo notarial centralizado, consiste em designar um único nó ou instituição independente para atuar como notário, e o notário assume as tarefas de coleta de dados, verificação e confirmação de transações no processo de interação entre cadeias. O notário é composto por, pelo menos, uma conta nas cadeias de origem e de destino. Este mecanismo consegue ter um processamento rápido de transações e é bastante adaptável, apesar do escopo restrito, limitando-se a troca de ativos.

No mecanismo notarial de múltiplas assinaturas o notário é geralmente composto por vários nós, onde cada nó possui uma chave e somente quando uma determinada porcentagem destes nós assinam em conjunto é que há um consenso e as transações entre cadeias podem ser confirmadas. Durante a verificação da transação, uma parte dos notários

é selecionada aleatoriamente do grupo notarial, diminuindo o grau de dependência da confiabilidade dos notários.

4.2. Bloqueio de *Hash*

O Mecanismo de Bloqueio de *Hash ou Hash Time Lock Contract* (HTLC) representa um marco significativo na evolução dos mecanismos de troca entre *blockchains*, proporcionando uma solução inovadora para realizar transações entre redes sem depender de intermediários confiáveis [Ou et al. 2022]. Ao implementar contratos HTLC nas *blockchains* envolvidas na negociação, o processo de troca de ativos é seguro e confiável. Esse contrato atua como uma garantia, bloqueando os ativos envolvidos até que as condições acordadas sejam atendidas. A utilização do conceito de *hash* adiciona uma camada adicional de segurança, criando uma trava com uma palavra secreta que deve ser correspondente em ambas as extremidades da transação. Além disso, a imposição de um limite de tempo para a conclusão da troca aumenta a eficiência e a segurança do processo. Em caso de não cumprimento dentro do prazo estipulado, o contrato automaticamente cancela a transação, revertendo os *tokens* para suas respectivas carteiras de origem. Esse mecanismo desempenha um papel fundamental na facilitação de trocas descentralizadas, promovendo a confiança e a segurança nas transações *blockchain* [Belchior et al. 2021].

5. Metodologia

A metodologia proposta para estabelecer o comparativo entre os métodos e custo de transações de *tokens* ERC-20 baseia-se em avaliar principalmente os mecanismos de interoperabilidade determinados, são eles: O Mecanismo Notarial e o Bloqueio de *Hash*. Nesse sentido, nas experimentações de levantamento dos custos são considerados métricas como o *Gas* necessário para realizar as transações e o tempo gasto para efetivá-las nas redes *blockchains* de origem e destino em cada uma das fases dos mecanismos.

5.1. Arquitetura do Mecanismo Notarial

Para a experimentação por meio do uso do Mecanismo Notarial como ferramenta de interoperabilidade entre *blockchains*, se faz necessário um terceiro confiável que gerencie as transações entre a *blockchain* de origem e destino. Dessa maneira, para abstração deste terceiro confiável, foi empregado um contrato inteligente com funcionalidades intrínsecas ao mecanismo.

Este contrato inteligente atua como o notarial do mecanismo, ou seja, ele possui o papel de recebedor do *token* do remetente da *blockchain* A, assumindo a responsabilidade de transferi-lo para o destinatário na *blockchain* B, e registrar informações acerca das transações realizadas. Ao final das etapas, o contrato também deve assegurar a entrega dos recursos ao destinatário designado.

Vale ressaltar que, embora o mecanismo seja eficaz, ele ainda possui limitações. Dentre as possíveis limitações temos como exemplo a segurança, que é de suma importância em qualquer aplicação *blockchain*. No caso desta arquitetura em específico, o mecanismo notarial atua de maneira centralizada, sendo assim a segurança da transação depende da integridade do notário, visto que ele é o responsável por receber os fundos na *blockchain* de origem e transferi-los para a *blockchain* de destino. Se o notário for comprometido de alguma forma, isso pode acarretar em perdas financeiras para os usuários

das redes. Porém, uma vez que o notário seja exaustivamente testado e reconhecido como confiável o mecanismo se torna extremamente eficiente.

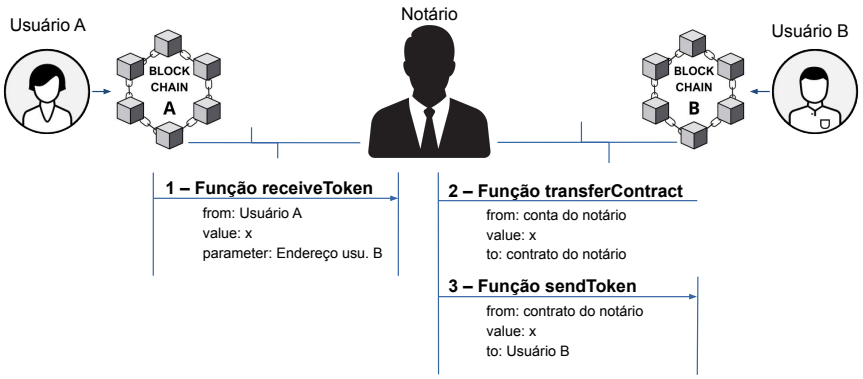

Figura 3. Arquitetura implementada para o Mecanismo Notarial.

No contexto da arquitetura desenvolvida para o Mecanismo Notarial, a Figura 3 demonstra o *Usuário A*, que pertence à *Blockchain A*, que realiza o envio de uma quantia específica do seu *token*, denotada por x, para o *Usuário B*, que é membro da *Blockchain B*. Para viabilizar essa transação entre as duas *blockchains*, o notário atua como um intermediário confiável. Inicialmente, o *Usuário A* transfere os fundos para o contrato inteligente do notário, utilizando a função *receiveFunds()*, e fornece o endereço do destinatário como parâmetro. Após o notário receber os fundos na *Blockchain A*, ele, então, na *Blockchain B*, emprega a função *transferContract()* para enviar *tokens* para o contrato. Finalmente, com a utilização da função *sendToken()*, o notário transfere o valor armazenado no contrato para o destinatário. Os códigos utilizados para os experimentos do Mecanismo Notarial estão disponíveis no repositório *Cross-chain-notarial* [1].

5.2. Arquitetura do Mecanismo Bloqueio de *Hash*

Semelhante ao Mecanismo Notarial, para a realização dos experimentos com o mecanismo Bloqueio de *Hash* também é necessário a implementação de contratos inteligentes. Neste caso, o contrato é responsável pela troca segura dos ativos, ou seja, ele é implantado nas duas redes e possui a tarefa de conectá-las.

Diferentemente do Mecanismo Notarial, não há um terceiro confiável no Mecanismo Bloqueio de *Hash*, o contrato inteligente atua sincronizando as redes no que diz respeito à verificação das transações, da palavra secreta e a devolução dos valores, caso necessário. Sendo assim, o contrato HTLC possui as funcionalidades de bloquear os fundos que serão transferidos, registrar o horário da transação e exigir uma palavra secreta no momento da retirada dos fundos para o destinatário da segunda rede. Caso o tempo tenha ultrapassado o período de bloqueio, o contrato é capaz de devolver a quantia para o remetente, conforme a definição do mecanismo.

Apesar disso, questões referentes à segurança do mecanismo também devem ser consideradas. O comprometimento do contrato inteligente que realiza a sincronização

[1] https://github.com/lesc-ufv/mecanismos

pode levar ao comportamento desordenado das chamadas de retirada e bloqueio dos fundos, causando possíveis perdas ou resultados inesperados.

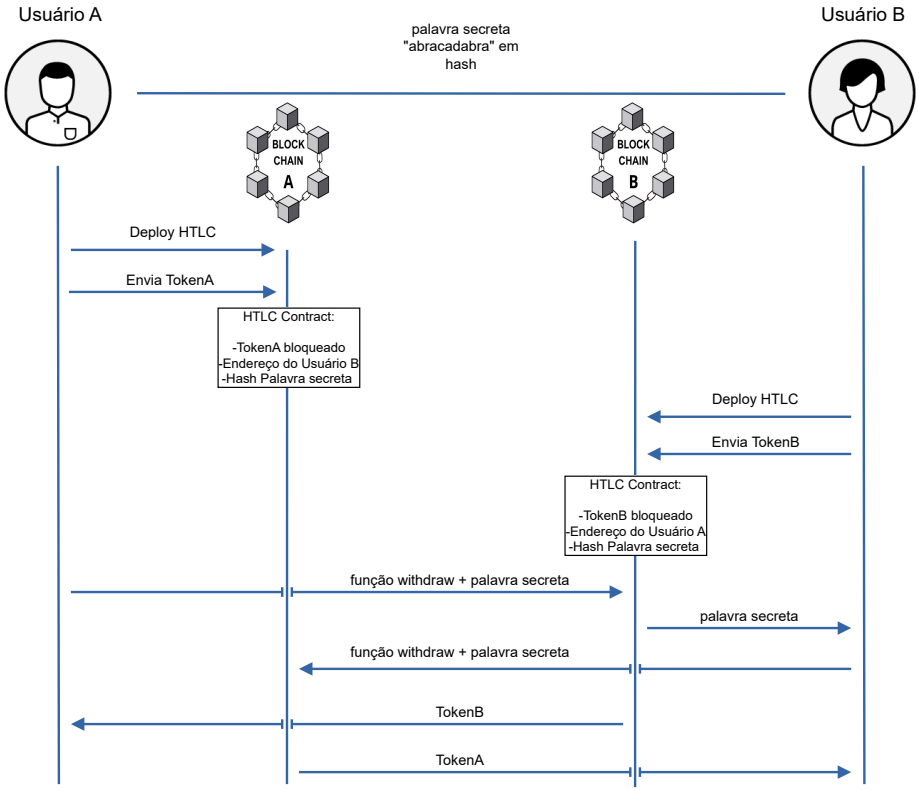

Figura 4. Arquitetura implementada do Mecanismo *Hash-Time Lock*.

No contexto do Mecanismo Bloqueio de *Hash*, conforme a Figura 4, o *Usuário A* deseja fazer a transferir seu *Token* para uma conta do *Usuário B* em outra rede. Para fazer essa transferência, ele escolhe uma "palavra secreta", e utiliza o *Hash* juntamente com o endereço de B para criar o contrato HTLC. Por meio do contrato criado na *Blockchain A*, ele bloqueia o *Token A* para a transferência ser realizada. De maneira similar, o *Usuário B* implanta o HTLC na *Blockchain B* com o endereço de A e a palavra secreta em *Hash*. Sendo assim, o *Usuário A* faz a retirada (*withdraw*) do *Token B* na *Blockchain B* com sua palavra secreta. Ao fazer isso, a palavra secreta pode ser usada pelo *Usuário B* para retirar o *Token A* da *Blockchain A*. Os códigos utilizados para os experimentos do Mecanismo HTLC estão disponíveis no repositório *Cross-chain-htlc* [2].

6. Resultados e Avaliações

Para a obtenção dos resultados, o ambiente de simulação foi composto estruturalmente utilizando redes locais com padrão Ethereum, por meio do software Ganache[3], que é utilizado para simular, executar testes e inspecionar o estado da rede. Foi utilizado também a

[2]https://github.com/lesc-ufv/mecanismos
[3]https://github.com/trufflesuite/ganache-ui

biblioteca Web3.js[4] para acesso aos contratos inteligentes. Os testes foram automatizados de maneira que as informações acerca das transações, as métricas de avaliação e o custo de *Gas* pudessem ser coletados, processados e apresentados para análise.

Para medir os custos intrínsecos aos mecanismos em relação às variações de quantidades de transações, levou-se em conta as diferentes fases dos mecanismos que se diferem em: custo gerado inicialmente para implantação do mecanismo e custo de cada transferência realizada. A coleta dos dados de custos de cada fase são apresentadas por valores em *Gas*, que é o valor cobrado por cada transação ou contrato inteligente executados nas *blockchains* com padrão Ethereum. A contabilização do *Gas* é realizada pela unidade de medida *Gwei*, onde 1 gwei é igual a 0,000000001 *Ethers*[5].

O Mecanismo Notarial foi segmentado em quatro fases executadas separadamente para obter as métricas de custo da transação e por fim obter os valores para comparação entre os mecanismos. Conforme a Tabela 2, ao analisar os valores constatou-se que o Mecanismo Notarial apresenta um comportamento quase linear. À medida que o número de transações aumenta, o tempo de execução pode ser prolongado e o preço do *Gas* pode sofrer alterações, pois ambos são proporcionais ao esforço despendido pela *blockchain*.

Tabela 2. Métricas de Desempenho do Mecanismo Notarial.

	1 Transação		10 Transações		100 Transações	
Fases	**Gas (Gwei)**	**Tempo (ms)**	**Gas (Gwei)**	**Tempo (ms)**	**Gas (Gwei)**	**Tempo (ms)**
Deploy	1.128.302	272	1.128.302	272	1.128.302	272
receiveFunds	112.960	174	514.000	1.760	4.524.400	15.827
transferContract	21.184	96	211.840	451	2.118.400	6.173
sendEther	30.703	120	307.030	783	3.070.300	14.675
Total	1.293.149	662	2.161.172	2.966	10.841.402	36.947

Como esses resultados são simulados em um ambiente local e controlado, sem altas demandas relacionadas ao custo e à capacidade da *blockchain*, eles representam o melhor cenário possível. Os custos de *Gas* englobam tanto a implantação na *blockchain* de origem quanto na de destino. Além disso, são computados os custos de *Gas* e o tempo para cada função individualmente. Cada tabela apresenta os valores correspondentes a diferentes quantidades de transações de interoperabilidade.

Vale ressaltar que, as métricas não variam em função da quantidade de Ether utilizado na transação, portanto todas as operações foram conduzidas com 1 Ether por padrão. Ao analisar a evolução do gráfico da Figura 5 de acordo com o número de operações, pode-se observar o comportamento do mecanismo. Nesse sentido, tanto o custo quanto o tempo dependem estritamente da quantidade de transações solicitadas.

O Mecanismo Bloqueio de *Hash* contempla fases diferentes do mecanismo Notarial. Foi necessário segmentar em duas fases de igual peso nos custos, uma vez que estas fases são executadas todas as vezes que se faz necessário realizar uma transação *cross-chain*. Neste mecanismo, os custos de *Gas* e tempo estão intimamente ligados ao fator humano, visto que a funcionalidade do mecanismo de retirada do valor transferido depende da palavra-chave do usuário e, portanto, podem ocorrer variações no tempo. Por tal motivo o tempo não é calculado para a função *withdraw* e que pode ser observado

[4]https://web3js.org/

[5]Criptomoeda ou Moeda digital descentralizada que é utilizada na rede Ethereum.

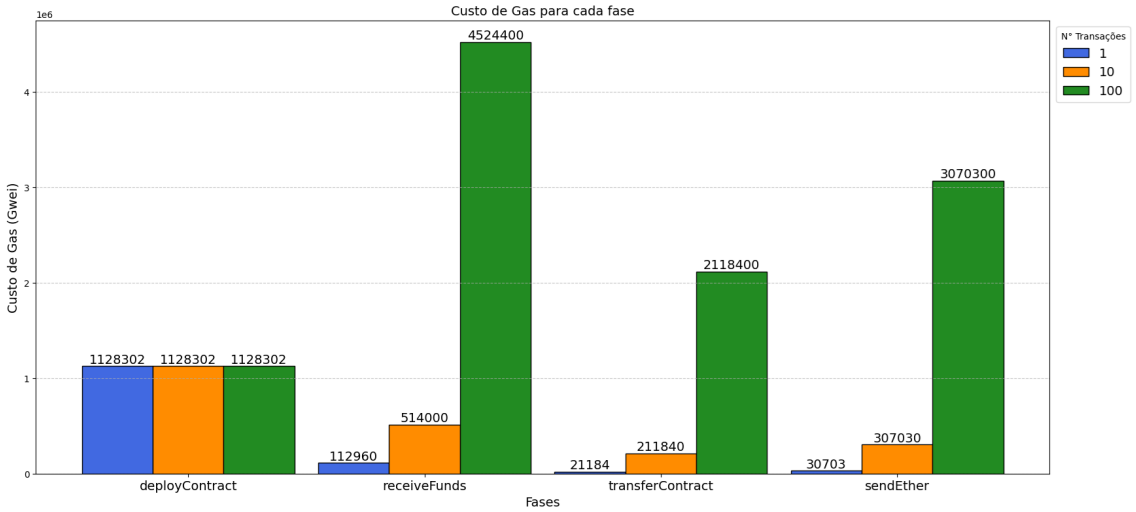

Figura 5. Custo em *Gas* por fase do Mecanismo Notarial.

na Tabela 3. Este comportamento apresenta uma diferença com o Mecanismo Notarial, pelo fato da retirada ser controlada pelos próprios usuários das duas redes. Ainda podemos observar na Tabela 3 os valores para as Fases *deployHTLC* e *transferToken* e que são destacados pelo gráfico da Figura 6. Onde é possível observar também que os valores da implantação do contrato (deployHTLC) impactam significativamente na utilização do mecanismo.

Tabela 3. Métricas de Desempenho do Mecanismo Bloqueio de *Hash*.

	1 Transação		10 Transações		100 Transações	
Fases	**Gas (Gwei)**	**Tempo (ms)**	**Gas (Gwei)**	**Tempo (ms)**	**Gas (Gwei)**	**Tempo (ms)**
deployHTLC	900.243	532	9.002.430	5.335	90.024.300	50.836
transferToken	83.177	-	831.770	-	8.317.700	-
Total	983.420	-	9.834.200	-	98.342.000	-

É importante notar também que, o custo de implantação do contrato mantém-se em valor padrão pelo fato do ambiente ser em local controlado, e portanto apresenta o melhor cenário possível, assim como no Mecanismo Notarial. Dessa maneira, o comportamento dos mecanismos de interoperabilidade possui um padrão linear, porém no Mecanismo Bloqueio de *Hash*, o custo de retirada varia conforme o tempo despendido pelos usuários para inserir a palavra-chave, enquanto o custo de implantação do contrato no Mecanismo Notarial se dá apenas a primeira vez que se realiza a transferência *cross-chain*.

Ao levar em consideração os custos totais por número de transações entre os Mecanismo Notarial e Bloqueio de *Hash*, conforme gráfico da Figura 7, podemos observar que além do mecanismo por Bloqueio de *Hash* ter um custo superior ao Notarial pelo motivo de ter que implantar um contrato a cada transferência realizada, ele se torna inviável quando se tem um grande número de transferências de baixo valor. Isto se deve aos custos fixos impostos pelo próprio método.

7. Considerações finais e trabalhos futuros

A interoperabilidade entre *blockchains* habilita a capacidade de haver cooperação e maior adesão das diferentes tecnologias por aplicações descentralizadas. Estruturamos arqui-

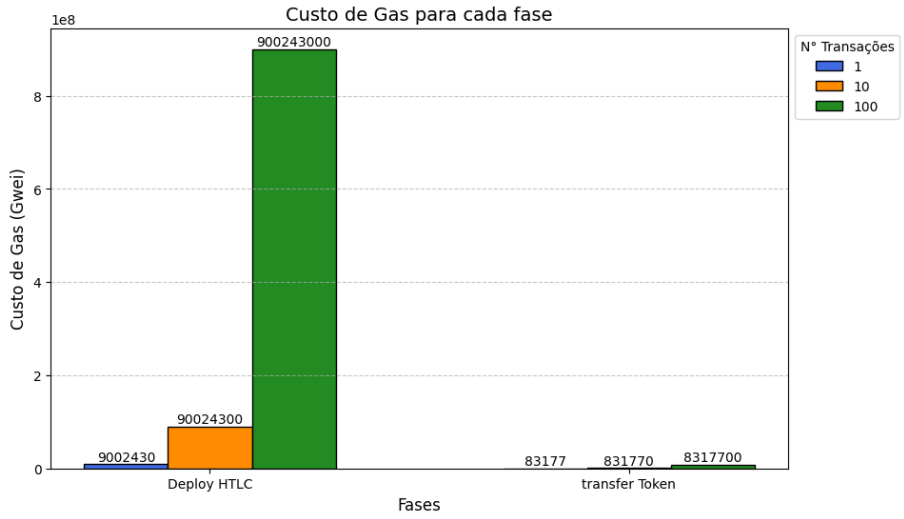

Figura 6. Custo em *Gas* por fase do Mecanismo Bloqueio de *Hash*.

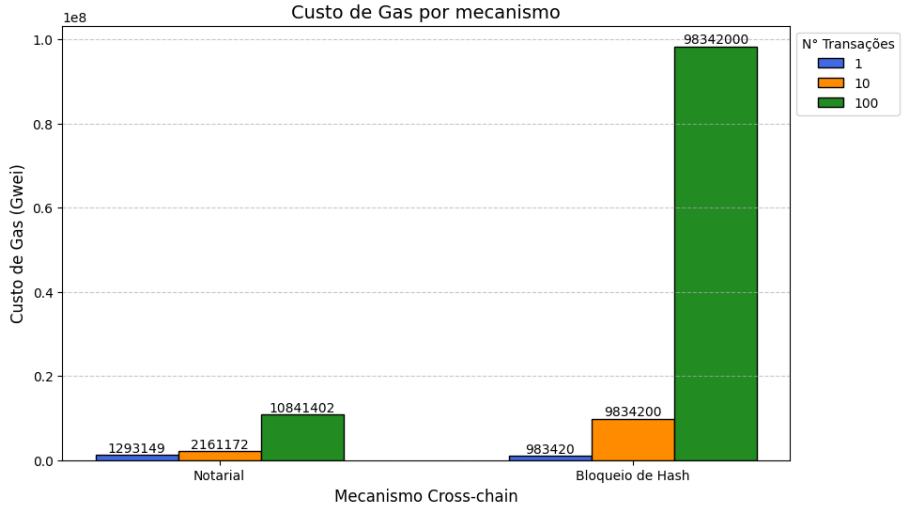

Figura 7. Custo total por número de transações entre os mecanismos.

teturas para implementar e experimentar os mecanismos de interoperabilidade, a fim de avaliar os seus custos. Pudemos observar que os custos fixos do Mecanismo Notarial diminuem significantemente quando o número de transações aumentam, enquanto na arquitetura apresentada do mecanismo Bloqueio de *Hash*, o custo fixo do método pode se tornar inviável para transações de baixo valor, uma vez que para cada transação é acarretado o custo de todo o método.

Além de avaliar os custos dos mecanismos de interoperabilidade apresentados, pretendemos ampliar ainda mais as pesquisas sobre interoperabilidade implementando os mecanismos em redes de teste fora de um ambiente controlado, para mensurar seu desempenho em ambientes reais. Além disso, expandir os teste em diferentes padrões de *tokens* e outros mecanismos cross-chain.

Referências

Belchior, R., Vasconcelos, A., Guerreiro, S., and Correia, M. (2021). A survey on blockchain interoperability: Past, present, and future trends. *ACM Computing Surveys (CSUR)*, 54(8):1–41.

Besançon, L., Silva, C. F. D., and Ghodous, P. (2019). Towards blockchain interoperability: Improving video games data exchange. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 81–85.

Buterin, V. (2016). Chain interoperability. *R3 research paper*, 9:1–25.

Esquivel, E. V., Campos, J. N., Mendonça, R. D., Vieira, A. B., and Nacif, J. A. M. (2023). Detecção de vulnerabilidades em contratos inteligentes utilizando arvore sintática abstrata. In *Anais do XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 335–348. SBC.

Gordon, W. J. and Catalini, C. (2018). Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability. *Computational and Structural Biotechnology Journal*, 16:224 – 230.

Hewa, T., Ylianttila, M., and Liyanage, M. (2021). Survey on blockchain based smart contracts: Applications, opportunities and challenges. *Journal of Network and Computer Applications*, 177:102857.

Jin, H., Dai, X., and Xiao, J. (2018). Towards a novel architecture for enabling interoperability amongst multiple blockchains. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1203–1211. IEEE.

Mendonça, R. D., Campos, J. N., Vieira, L. F., Vieira, M. A., Vieira, A. B., and Nacif, J. A. (2022). Tokens não fungíveis (nfts): Conceitos, aplicações e desafios. *SBC*.

Nakamoto, S. et al. (2008). Bitcoin: A peer-to-peer electronic cash system. *Bitcoin*.

Ndiaye, M. and Konate, P. K. (2021). Cryptocurrency crime: Behaviors of malicious smart contracts in blockchain. In *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–8. IEEE.

Ou, W., Huang, S., Zheng, J., Zhang, Q., Zeng, G., and Han, W. (2022). An overview on cross-chain: Mechanism, platforms, challenges and advances. *Computer Networks*.

Qasse, I. A., Abu Talib, M., and Nasir, Q. (2019). Inter blockchain communication: A survey. In *Proceedings Annual International Conference Research Track*, pages 1–6.

Rogers, I., Carter, D., Morgan, B., and Edgington, A. (2022). Diminishing dreams: The scoping down of the music nft. *M/C Journal*, 25(2).

Rouhani, S. and Deters, R. (2017). Performance analysis of ethereum transactions in private blockchain. In *2017 8th IEEE(ICSESS)*, pages 70–74. IEEE.

Vogelsteller, F. and Buterin, V. (2015). Eip-20: Token standard, ethereum improvement proposals. `https://eips.ethereum.org/EIPS/eip-20`.

Wang, X. and Feng, J. (2018). The research of consortium blockchain dynamic consensus based on data transaction evaluation. In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pages 214–217. IEEE.

Wegner, P. (1996). Interoperability. *ACM Computing Surveys (CSUR)*, 28(1):285–287.