Avaliando o Custo de Contratos Inteligentes em Aplicações Blockchain por meio de Ambientes de Simulação

Emanuel F. Coutinho[1], **Delano José Holanda Maia**[1]
Wagner L. Braga Bezerra[1], **Antonio Welligton dos Santos Abreu**[1]

[1]Programa de Pós-Graduação em Computação (PCOMP)
Universidade Federal do Ceará (UFC) – Quixadá – CE – Brasil

emanuel.coutinho@ufc.br, delanomaia@ufc.br

wagnerbragacrf@alu.ufc.br, siwelligton@gmail.com

Abstract. *Blockchain is a sequence of blocks containing a complete record of transactions as a public ledger, informing the order in which transactions occurred, and which has become an option for application development. Smart contracts are a flow of value based on terms and conditions, similar to contracts in the real world, but digital and in code. The objective of this work is to present a simulation of the smart contracts use in blockchain, to have a view of the resources consumption in the execution of operations, specifically financial costs. For this, a smart contract was designed to simulate an environment of financial donations. As a result, it was possible to evaluate the costs of smart contract methods and their impact on the number of calls in the application, reinforcing the importance of simulating environments.*

Resumo. *Blockchain é uma sequência de blocos contendo um registro completo de transações como um livro público, informando a ordem na qual transações ocorreram, e que vem se tornando uma opção para desenvolvimento de aplicações. Os contratos inteligentes são um fluxo de valor baseado em termos e condições, semelhantes a contratos no mundo real, porém digitais e em código. O objetivo deste trabalho é apresentar uma simulação do uso de contratos inteligentes em blockchain para se ter uma visão do consumo dos recursos na execução das operações, especificamente custos financeiros. Para isso, projetou-se um contrato inteligente para simular um ambiente de doações financeiras. Como resultado foi possível avaliar os custos dos métodos do contrato inteligente e seu impacto na quantidade de chamadas da aplicação, reforçando a importância de simular ambientes.*

1. Introdução

Muitas vezes, aplicações necessitam de operações simples, porém que garantam integridade dos dados e possibilidade de rastreabilidade. A manutenção de um histórico das ações executadas por aplicações é algo muito atrativo, principalmente se não houver a possibilidade de alterações e fraudes. Após o sucesso do Bitcoin, Ethereum e Hyperledger, *blockchain* está ganhando ampla adoção em uma variedade de aplicações, usando uma diversidade de sistemas distribuídos com características variadas [Stoykov et al. 2017].

Blockchain é uma sequência de blocos que contém o registro completo de transações como um livro público, apontando a ordem na qual transações ocorreram [Bhaskar e Chuen 2015]. Cada bloco na cadeia confirma a integridade do anterior, e

todo o caminho de volta ao primeiro bloco. A *blockchain* consiste em um conjunto de dados compostos por uma cadeia de pacotes de dados (blocos) onde um bloco compreende transações múltiplas [Nofer et al. 2017], podendo ser estendida por blocos adicionais e, representando um registro completo do histórico de transações. Estes blocos podem ser validados pela rede usando mecanismos criptográficos [Nofer et al. 2017]. Sua aplicação atualmente está se expandindo em diversas áreas, como: finanças, saúde e agricultura.

No contexto da *blockchain*, os contratos inteligentes são um fluxo de valor baseado em certos termos e condições, semelhantes a contratos no mundo real. A única diferença é que eles são completamente digitais, o que significa um pequeno código de programação é armazenado em uma *blockchain*. Existem diferentes plataformas de *blockchain* que podem ser utilizadas para desenvolver contratos inteligentes, sendo a *Ethereum* a mais utilizada [Alharby e van Moorsel 2017]. Pode-se acionar um contrato inteligente endereçando uma transação para ele, onde em seguida, ele executa de forma independente da forma que foi escrito, em qualquer nó da rede, conforme os dados que foram incluídos no acionamento da transação [Christidis e Devetsikiotis 2016].

Adicionalmente, desenvolvedores estão começando a criar diversas aplicações e empregar tecnologias variadas para soluções de problemas com *blockchain* para as mais variadas áreas, como saúde, agricultura e educação. Assim, esta tecnologia começa a ser incorporada na sociedade, seja para a construção de aplicações, seja para sua utilização de maneira transparente por usuários, gerando demandas para diversas soluções.

Com a ampla utilização da internet, um serviço que surgiu foi o das *lives*. Uma *live* é uma transmissão ao vivo, geralmente em vídeo, na qual existem diversas maneiras de quem assiste poder participar, seja com textos em redes sociais ou acessando aplicativos para fins diversos. Uma das aplicações que em tempos de isolamento social, como o que se destacou devido à pandemia do Coronavirus, foi a de doações para instituições de caridade, seja por meio de transferências bancárias ou por uma aplicativo de doações. Porém, quem doa normalmente não sabe de forma transparente e segura, se o arrecadado foi de fato para uma instituição ou para quais instituições, caso haja alguma divisão. Nesse contexto, surgiu a ideia aplicar *blockchain* com contratos inteligentes, e assim quem acessar uma aplicação baseada em *blockchain* pode ter um histórico do registro da doação, quanto foi doado, quantas doações ocorreram, para quem foi doado, e quando.

Este trabalho tem como objetivo apresentar uma simulação do uso de contratos inteligentes em um ambiente de *blockchain*, para se ter uma visão do consumo dos recursos na execução das operações, especificamente custos financeiros. Para atender este objetivo, projetou-se um contrato inteligente para simular um ambiente de doações financeiras, no contexto das *lives*. Este artigo está dividido nas seguintes seções: a Seção 2 apresenta um breve referencial teórico de *blockchain*; na Seção 3 alguns trabalhos relacionados são descritos; a Seção 4 contextualiza o cenário das doações em *lives*; na Seção 5 a estratégia de condução do trabalho é apresentada; a Seção 6 exibe os resultados e análises; e na Seção 7 conclusões e trabalhos futuros são apresentados.

2. Referencial Teórico

2.1. *Blockchain*

Blockchain é uma sequência de blocos que contém o registro completo de transações como um livro público, indicando a ordem na qual as transações ocorre-

ram [Bhaskar e Chuen 2015]. A Figura 1 representa uma *blockchain* com o bloco recém validado apontando para o bloco imediatamente anterior gerado. Cada bloco na cadeia confirma a integridade do anterior, e todo o caminho de volta ao primeiro bloco (bloco de gênese). Uma *blockchain* consiste em um conjunto de dados compostos por uma cadeia de pacotes de dados (blocos) onde um bloco compreende transações múltiplas [Nofer et al. 2017]. Ela é estendida por cada bloco adicional e, portanto, representa um registro geral completo do histórico de transações. Estes blocos podem ser validados pela rede usando mecanismos criptográficos. Além das transações, cada bloco contém um carimbo de data/hora (*timestamp*), o valor de *hash* do bloco anterior (pai), e um "*nonce*", que é um número aleatório para verificar o *hash*. Este conceito garante a integridade de toda a *blockchain* até o primeiro bloco.

Os valores do *hash* são únicos e fraudes podem ser efetivamente prevenidas, uma vez que as mudanças em um bloco na cadeia mudariam imediatamente o respectivo valor do *hash* [Nofer et al. 2017]. Se a maioria dos nós da rede concordarem por meio de um mecanismo de consenso sobre a validade das transações em um bloco e sobre a validade do próprio bloco, então este bloco pode ser adicionado à cadeia. Portanto, novas transações não são automaticamente adicionadas ao registro. Em vez disso, o processo de consenso garante que essas transações sejam armazenadas em um bloco por certo tempo (e.g. 10 minutos na *blockchain Bitcoin*) antes de serem transferidas para o livro-razão. Após este processo, as informações na *blockchain* não podem mais ser alteradas. No caso do *Bitcoin*, os blocos são criados pelos chamados "mineradores", que são recompensados com *Bitcoins* pela validação dos blocos.

2.2. Contratos Inteligentes

O conceito de contrato inteligente foi introduzido por Nick Szabo em 1994, definido como um protocolo de transação computadorizado que executa os termos de um contrato. Szabo sugeriu traduzir cláusulas contratuais (e.g., garantias e títulos) em código e incorporá-las em propriedades (hardware ou software) que possam se autoaplicar, de modo a minimizar a necessidade de intermediários confiáveis entre as partes envolvidas na transação, e a ocorrência de exceções maliciosas ou acidentais [Szabo 1994].

Dentro do contexto da *blockchain*, os contratos inteligentes são um fluxo de valor baseado em certos termos e condições, como contratos no mundo real. A única diferença é que eles são completamente digitais, significando que um pequeno código é armazenado na *blockchain*. Existem diferentes plataformas de *blockchain* que podem ser utilizadas para desenvolver contratos inteligentes, sendo a *Ethereum* a mais utilizada [Alharby e van Moorsel 2017].

Os contratos inteligentes funcionam como *scripts* armazenados. Como residem na cadeia, eles possuem um endereço exclusivo. Pode-se acionar um contrato inteligente

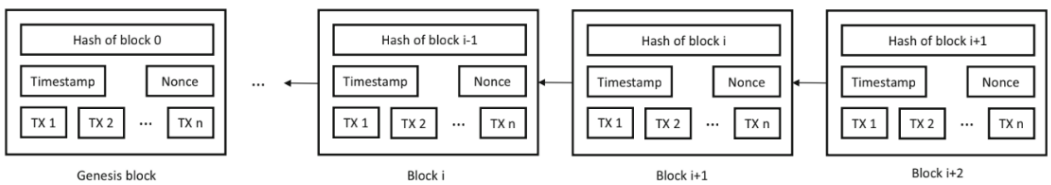

Figura 1. *Blockchain* e seus blocos [Nofer et al. 2017][Zheng et al. 2017]

endereçando uma transação para ele, onde em seguida, ele executa de forma independente da forma que foi escrito, em qualquer nó da rede, de acordo com os dados que foram incluídos no acionamento da transação [Christidis e Devetsikiotis 2016].

Contratos inteligentes são criados sobre uma plataforma de criptomoeda (e.g. *Ethereum*). Uma criptomoeda é um sistema descentralizado para interagir com dinheiro virtual em um livro compartilhado de forma global. Os usuários transferem dinheiro e interagem com contratos por meio da publicação de dados assinados, denominados de transações da rede de criptomoedas. A rede consiste em nós chamados mineradores que propagam informações, armazenam dados, e atualizam os dados aplicando transações.

3. Trabalhos Relacionados

Alguns trabalhos na literatura simularam infraestruturas de *blockchain*, com análises variadas [Stoykov et al. 2017] [Aoki et al. 2019] [Banno e Shudo 2019] [Faria e Correia 2019].

Segurança e confiabilidade em um sistema com *blockchain* são propriedades que necessitam ser analisadas. Para isso, [Stoykov et al. 2017] propuseram o VIBES, um simulador de *blockchain* configurável. Com o VIBES, os usuários podem explorar características e métricas importantes da rede, argumentar sobre interações entre nós e comparar diferentes cenários de maneira intuitiva. Os autores apresentaram uma arquitetura, a possibilidade de configuração de diversos parâmetros da rede, métricas de saída, e um exemplo simples de aplicação. Em relação aos contratos inteligentes, é possível configurar o tempo, intensidade e densidade.

A realização de experimentos em *blockchain* reais é difícil porque é necessário um grande número de nós em grandes áreas. [Aoki et al. 2019] desenvolveram o simulador de rede *blockchain* SimBlock para experimentos. O SimBlock pode alterar facilmente o comportamento dos nós, para permitir investigar a influência do comportamento dos nós na *blockchain*. Um experimento foi apresentado com o tempo de propagação dos blocos.

Uma das dificuldades que os pesquisadores de *blockchain* estão enfrentando é a falta de maneiras de verificar comportamentos de redes *blockchain* de grande escala. [Banno e Shudo 2019] utilizaram o SimBlock para visualizar o comportamento dos nós e sua propagação. Alguns exemplos de uso foram discutidos, como explorar melhores estratégias de seleção de vizinhos e avaliar a influência das redes de retransmissão,

A falta de ferramentas para avaliar as decisões de projeto e implementação em blockchain é uma carência da literatura. [Faria e Correia 2019] apresentaram um simulador de eventos discretos flexível para avaliar diferentes implementações de blockchain, o BlockSim. A *blockchain* pode ser modelada e simulada rapidamente, estendendo os modelos existentes. Um modelo para a arquitetura do simulador e para o framework foi apresentado, assim como exemplos de código para uso. Um modelo exemplo de simulações da Ethereum com diferentes valores de *gas* possibilitou um estudo sobre o desempenho, como o impacto de dobrar o número de transações por bloco, impacto no atraso de propagação do bloco, e impacto da criptografia.

A diferença entre os trabalhos relacionados e o trabalho proposto é o foco, pois este propõe-se a simular aplicações *blockchain*, não com foco na infraestrutura, mas no código e em métricas para sua avaliação. Os trabalhos relacionados focam em simulações

da rede, com apenas configurações de parâmetros de contratos inteligentes. Nenhuma delas focou com mais profundidade no código em si, que é o caso deste trabalho, ao analisar os efeitos das funções dos contratos inteligentes, focando nos custos.

4. Contexto da Aplicação e Cenários

No cenário atual mundial de pandemia do Coronavírus, uma das soluções para se evitar contágio é o isolamento social. Nesse contexto, muitas pessoas acabam permanecendo em suas residências, tendo que adaptar sua rotina. Uma das consequências desse momento foi a aumento considerável na quantidade de *lives* transmitidas. Por *live* entende-se a transmissão *online* de algum evento, onde quem assiste pode participar de alguma forma (via aplicativo, telefone, redes sociais, etc). E com essa capacidade de comunicação e alcance com o público, surgiram também serviços secundários, como o de doações financeiras. Uma das aplicações resultantes foi a de doação financeira para instituições de caridade, que pode ocorrer por meio de transferências bancárias ou por um aplicativo de doações.

Entretanto, um fato que ocorre é que pessoas que realizam a doação no intuito de ajudar acabam não sabendo, de forma transparente e segura, de fato a quais instituições ajudaram, e se essas instituições estão atendendo ao que foi apresentado na *live*. Por exemplo, instituições que recebem doações para manter ações de combate ao Coronavírus.

Dado uma aplicação de doações *online*, algumas operações simples podem ser consideradas, automatizadas ou não: definição do valor de doação, seleção do meio de transferência do valor, execução da transferência do valor, consulta dos valores totais doados, cadastro e consulta das instituições participantes, consulta dos valores doados para as instituições participantes, definição do período de tempo de arrecadação de doações, etc. A forma como essas operações são definidas e executadas depende da regra de negócio da aplicação, pois as variações são muitas. Além disso, diversas outras funcionalidades podem surgir, e como em qualquer sistema, há manutenção e evolução.

5. Estratégia de Condução do Trabalho

5.1. Projeto do Experimento

O objetivo deste experimento é analisar os custos de um contrato inteligente e seus métodos. Neste trabalho, utilizaremos o contexto apresentado previamente na Seção 4 das doações em *lives* e como base a *blockchain* Ethereum.

Para essa análise do custo, para cada cenário o valor do *gas* será coletado. Para fins de explicação, o *gas* é a unidade de medida do poder computacional na Ethereum, e o *ether* é para a medição e pagamento pelo custo computacional no Ethereum. O *ether* é o combustível da *Ethereum* cuja finalidade é pagar pelo custo da computação realizada. O termo "custo" representará os valores das unidades computacionais *gas* e *ether*.

Para isto, as seguintes etapas foram projetadas: (i) Definição de cenários para simulação em alto nível de requisições de uma aplicação real a uma *blockchain*; (ii) Desenvolvimento de um contrato inteligente com funções de escrita e leitura na *blockchain*; (iii) Execução dos cenários definidos e coleta dos dados resultantes da simulação; (iv) Consolidação dos resultados; e (v) Análise dos dados.

O *gas* total para cada cenário será coletado, convertido para uma unidade monetária e analisado. Existem dois tipos de custos, medidos em *gas*: custos de transação

e custos de execução. Normalmente para o cliente ele não tem essa informação, tendo apenas um valor a ser debitado em sua carteira. Os custos de transação são baseados no envio de dados e contrato inteligente para a *blockchain*, e dependem do tamanho do contrato. Os custos de execução são baseados no custo das operações computacionais executadas como resultado da transação, relacionados também com custos de armazenamento de variáveis globais e o tempo de execução das chamadas de método. A rigor, o custo total seria a soma desses dois valores.

5.2. Projeto das Simulações

Existem várias redes no mercado atualmente que podem trabalhar com a criação de contratos inteligentes. A *Ethereum* foi selecionada para o trabalho pois é bastante indicada atualmente para execução de contratos inteligentes em *blockchain* [Korpela et al. 2017].

Para esta simulação, basicamente duas ferramentas serão utilizadas: Solidity e Remix. A Solidity é a linguagem de programação criada pela própria *Ethereum* para o desenvolvimento de contratos inteligentes. O Remix é um ambiente *online* para desenvolvimento de contratos inteligentes. Neste trabalho, como o foco é simular uma aplicação real, a rede oficial de Ethereum não será utilizada. Esse é um dos motivos pelo qual o Remix será utilizado, pois essa é uma de suas funções.

Para este trabalho, três cenários foram projetados. Todos eles utilizam os mesmos métodos do contrato inteligente, apenas variando a quantidade e sequência. Todos serão basicamente chamadas de funções do contrato inteligente, executados no Remix.

O primeiro cenário consiste em 5 doações de valor fixo, consulta ao número de doações, consulta ao valor total arrecadado e envio do valor total para uma instituição. O segundo cenário realiza 10 doações de valor variável, consulta ao número de doações, consulta ao valor total arrecadado e envia o valor total para uma instituição. O terceiro cenário executa 5 doações de valor variável, consulta ao número de doações, consulta ao valor total arrecadado, mais 5 doações de valor variável, consulta ao número de doações, consulta ao valor total arrecadado e envio do valor total para uma instituição.

6. Resultados e Análises

Para a contabilização dos custos, algumas conversões devem ser realizadas. O valor do *gas* é variável, e medido em *gwei* (10^{-9} de 1 *ether*). Então, na data da realização dos experimentos (30/07/2020), o valor do *gas* estava 52 *gwei*. Também nesta data, 1 *ether* valia R$ 1722,05. Para a obtenção do valor do *gas* dos cenários em reais, basta multiplicá-lo por todos esses valores.

Para a implementação dos cenários, quatro métodos foram projetados em um contrato inteligente, desenvolvidos na IDE Remix com a linguagem de programação Solidity. Os métodos do contrato inteligente foram: **enviarDoacao()**, responsável por enviar uma quantia como doação para a *blockchain*; **contarDoacoes()**, responsável por listar a quantidade de doações realizadas; **obterValorTotal()**, responsável por retornar o valor total doado; e **doarParaInstituicao()**, responsável por executar a doação do valor total arrecadado para uma instituição. Os cenários basicamente vão executar chamadas a esses métodos, conforme o projeto de cada um.

Todas as doações no Cenário 1 foram de valor fixo, iguais a 1 *ether*. A Tabela 1

exibe os valores de *gas* para este cenário, com os valores consolidados ao final. A Figura 2 apresenta os valores de *gas* para as transações do Cenário 1.

Tabela 1. Métricas para o Cenário 1

Operação	Custo da Transação	Custo da Execução
DEPLOY	524131	362775
DOAÇÃO 1	164210	142938
DOAÇÃO 2	114410	93138
DOAÇÃO 3	114410	93138
DOAÇÃO 4	114410	93138
DOAÇÃO 5	114410	93138
CONTA DOAÇÃO	21682	410
VALOR TOTAL	21945	673
DOAR INSTITUIÇÃO	72403	123534
Total *gas*	1262011	1002882
Total *ether*	0,065624572	0,052149864
Total R$	113,0087942	89,8046733

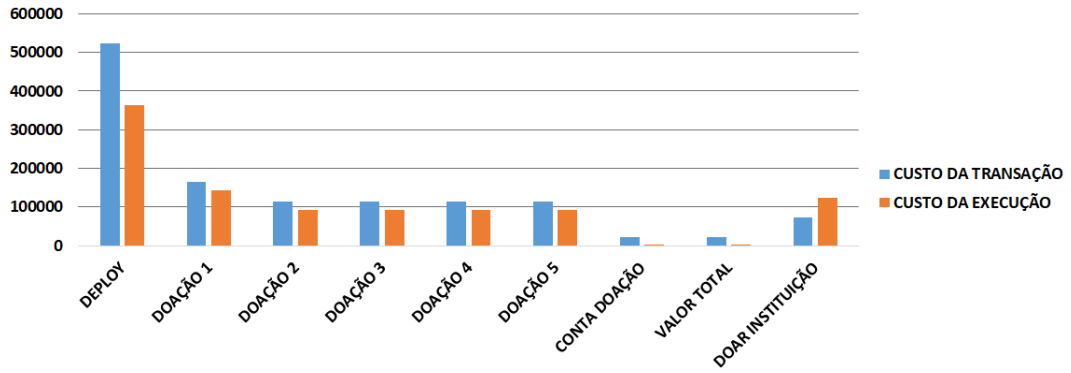

Figura 2. Métricas para o Cenário 1 - 5 doações

Os valores das doações no Cenário 2 variaram de 1 a 10 *ether*. A Figura 3 apresenta os valores de *gas* para as transações do Cenário 2. A Tabela 2 exibe os valores de *gas* para este cenário, com os valores consolidados ao final.

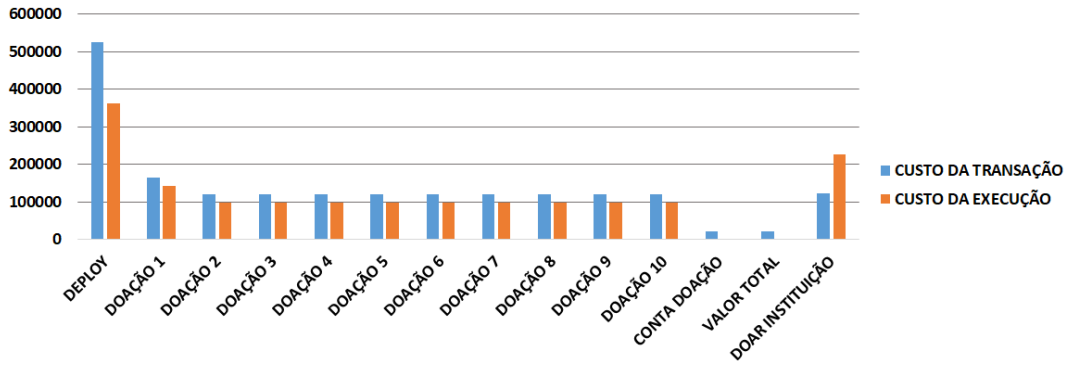

Figura 3. Métricas para o Cenário 2 - 10 doações

Os valores das doações no Cenário 3 também variaram de 1 a 10 *ether*. A diferença entre este cenário e o Cenário 2 é o resultado parcial da contagem de doações e valor arrecadado após 5 doações. A Figura 4 apresenta os valores de *gas* para as transações do Cenário 3. A Tabela 3 exibe os valores de *gas* para este cenário, com os valores consolidados ao final.

Tabela 2. Métricas para o Cenário 2

Operação	Custo da Transação	Custo da Execução
DEPLOY	524131	362775
DOAÇÃO 1	164210	142938
DOAÇÃO 2	114410	93138
DOAÇÃO 3	114410	93138
DOAÇÃO 4	114410	93138
DOAÇÃO 5	114410	93138
DOAÇÃO 6	114410	93138
DOAÇÃO 7	114410	93138
DOAÇÃO 8	114410	93138
DOAÇÃO 9	114410	93138
DOAÇÃO 10	114410	93138
CONTA DOAÇÃO	21682	410
VALOR TOTAL	21945	673
DOAR INSTITUIÇÃO	72403	123534
Total *gas*	1928219	1613687
Total *ether*	0,100267388	0,083911724
Total R$	172,6654555	144,5001843

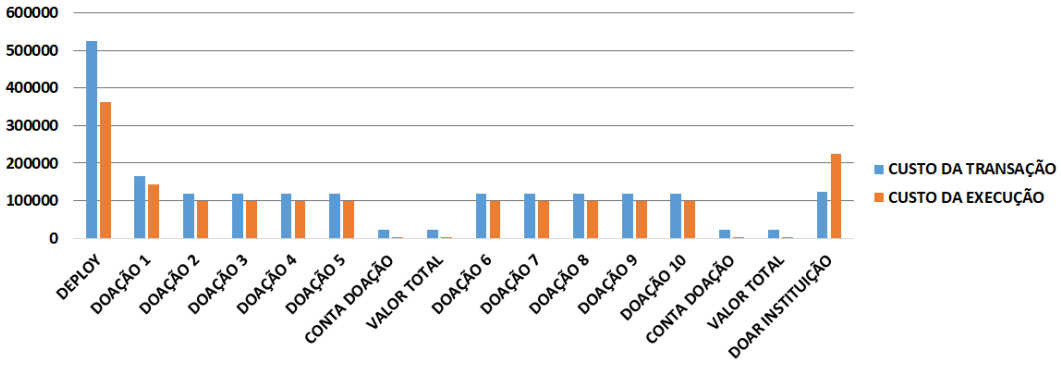

Figura 4. Métricas para o Cenário 3 - 10 doações com parciais

Tabela 3. Métricas para o Cenário 3

Operação	Custo da Transação	Custo da Execução
DEPLOY	524131	362775
DOAÇÃO 1	164210	142938
DOAÇÃO 2	114410	93138
DOAÇÃO 3	114410	93138
DOAÇÃO 4	114410	93138
DOAÇÃO 5	114410	93138
CONTA DOAÇÃO	21682	410
VALOR TOTAL	21945	673
DOAÇÃO 6	114410	93138
DOAÇÃO 7	114410	93138
DOAÇÃO 8	114410	93138
DOAÇÃO 9	114410	93138
DOAÇÃO 10	114410	93138
CONTA DOAÇÃO	21682	410
VALOR TOTAL	21945	673
DOAR INSTITUIÇÃO	72403	123534
Total *gas*	1971846	1614770
Total *ether*	0,102535992	0,08396804
Total R$	176,572105	144,5971633

6.1. Análises e Discussões

A implantação do contrato inteligente na *blockchain* requer um custo. Esse custo ocorre apenas na implantação, não sendo mais efetivado. Para o cliente da aplicação, esse custo não é cobrado, pois é algo que quem detém o serviço arca para disponibilizar o serviço. Conforme indicado previamente, há custo de *gas* para transações e execução, que são transparentes para o cliente. Esse valores são especificados na documentação da Ethereum [Wood 2020], e possuem muitos desdobramentos para se chegar no valor final. Porém, ao analisar os resultados, percebe-se que os valores para os métodos são sempre iguais. O valor final de *gas* é proporcional à quantidade de chamadas de métodos, independente dos

valores doados e arrecadados. Também se verificou que o tamanho dos métodos influencia na quantidade de gas. Assim, métodos mais "enxutos" custam menos.

Analisando o quanto a simulação está perto ou longe da realidade, do ponto de vista de valores na unidade monetária real, utilizamos valores do momento da coleta dos dados e realizamos as devidas conversões, para a obtenção do valor de custo final. É uma estimativa, pois frequentemente os valores de referência para *gas*, *gwei* e *ether* estão variando. Assim, para fins de estimativa, deve-se considerar os resultados adequados em períodos próximos da data da coleta dos dados.

Identificamos algumas limitações neste trabalho. O Remix não permite que inclusões na *blockchain* sejam executadas de maneira automática, como uma função *main*. Então as simulações foram manuais, por meio do pressionamento do botão de chamada de cada método do contrato. Além disso, as coletas dos *logs* também foram manuais. Assim, uma análise da escalabilidade não foi possível, sendo outra limitação do trabalho. Há então a necessidade do estudo de outras ferramentas de apoio ao desenvolvimento.

Muitos simuladores atualmente focam na simulação da infraestrutura e redes. Porém, a simulação dos custos não é comum, assim como quanto custariam os métodos do contrato inteligente. Esses resultados são importantes para se estimar custos de desenvolvimento, que muitas vezes são repassados para o cliente. Isso reforça a importância da seleção de ferramentas para simulação adequadas para *blockchain*. Como existe uma grande diversidade de ferramentas, cada uma com suas características e métricas variadas, há uma complexidade para a plena utilização na simulação de aplicações *blockchain*.

Uma questão que ficou fora do escopo deste trabalho é como testar. Uma vez implantado, não há como corrigir o código, tendo que haver uma nova implantação de código. O próprio Remix possui um módulo para testes unitários dos contratos inteligentes, mas como em todo desenvolvimento de software com boas práticas, todo código deve ser bem projetado e testado antes de ser implantado. Há uma variedade grande de ferramentas para suportar o desenvolvimento de aplicações em *blockchain*. Também há a necessidade de se definir quais serão mais adequadas para o domínio da aplicação a ser desenvolvida e os custos envolvidos no processo de desenvolvimento.

Uma proposta para ampliar este trabalho é criar um contrato inteligente que receba uma lista de instituições, junto com uma data de inicio e fim para as doações e também definindo a porcentagem que irá para cada uma, uma espécie de peso. Antes de iniciar uma *live*, a lista deve ser divulgada mostrando as entidades que serão beneficiadas com as doações recebidas no período estipulado. Os espectadores poderão realizar doações de qualquer valor, que serão acumuladas no contrato. A qualquer momento doadores podem verificar quanto já foi arrecadado e ao chegar ao tempo determinado como fim, o contrato divide os valores e envia de forma automática para os beneficiários. Uma situação funcional que não foi considerada neste trabalho é como garantir que a doação foi efetivamente utilizada no destino, pois apenas se verificou o registro da doação.

7. Conclusão

Blockchain atualmente está sendo bastante utilizada em aplicações que necessitam de um diferencial no armazenamento e compartilhamento de dados, além de sua capacidade de imutabilidade. Este trabalho apresentou uma avaliação dos custos da execução de

métodos de um contrato inteligente em uma rede de testes da Ethereum por meio da IDE Remix e linguagem de programação Solidity. Os principais resultados foram a avaliação dos custos de chamadas dos métodos de um contrato inteligente, e a necessidade de identificar ferramentas adequadas para as simulações, reforçando a importância de simular eventos antes de partir para ambientes de produção.

Como contribuições científicas desse trabalho, tem-se uma análise dos custos da utilização de contratos inteligentes em uma situação real. Ainda que o contexto de uso da aplicação tenha sido simulado, a ideia por trás dessa avaliação auxilia no projeto de aplicações que utilizam recursos de *blockchain*.

Este trabalho auxilia desenvolvedores de aplicações *blockchain* e gerentes de projeto, pois ajuda na previsão de custos e melhor redimensionamento do código. Como trabalhos futuros, pretende-se explorar outras ferramentas para simulação de aplicações *blockchain* do ponto de vista de código e testes. Adicionalmente, há a necessidade de se estudar simulações em larga escala, com muitas requisições à *blockchain*.

Referências

Alharby, M. e van Moorsel, A. (2017). Blockchain-based smart contracts: A systematic mapping study. *arXiv preprint arXiv:1710.06372*.

Aoki, Y., Otsuki, K., Kaneko, T., Banno, R., e Shudo, K. (2019). Simblock: A blockchain network simulator. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 325–329.

Banno, R. e Shudo, K. (2019). Simulating a blockchain network with simblock. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 3–4.

Bhaskar, N. D. e Chuen, D. L. K. (2015). Chapter 3 - bitcoin mining technology. In Chuen, D. L. K., editor, *Handbook of Digital Currency*, pages 45 – 65. Academic Press, San Diego.

Christidis, K. e Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303.

Faria, C. e Correia, M. (2019). Blocksim: Blockchain simulator. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 439–446.

Korpela, K., Hallikas, J., e Dahlberg, T. (2017). Digital supply chain transformation toward blockchain integration. In *50th Hawaii International Conference on System Sciences (HICSS)*.

Nofer, M., Gomber, P., Hinz, O., e Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering*, 59(3):183–187.

Stoykov, L., Zhang, K., e Jacobsen, H. (2017). Vibes: Fast blockchain simulations for large-scale peer-to-peer networks: Demo. In *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos*, Middleware '17.

Szabo, N. (1994). Smart contracts. http://bit.ly/2Yc9vjb. Online; accessed Oct-2019.

Wood, G. (2020). Ethereum: A secure decentralised generalised transaction ledger.

Zheng, Z., Xie, S., Dai, H.-N., Chen, X., e Wang, H. (2017). Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services (IJWGS)*.