




Performance Evaluation of Data Transactions in Blockchain

Antonio Welligton dos Santos Abreu , Emanuel Ferreira Coutinho  and Carla Ilane Moreira Bezerra 

Abstract—Blockchain is an emerging technology, with a decentralized infrastructure avoiding third party dependency. Smart contracts are one of the features of Ethereum blockchain, capable of running distributed applications in unreliable environments, enabling process automation and being one of the most sought technologies due to the high customization added to transactions. However, little is known about predicting the cost and execution time behavior of blockchain-based system transactions. This work aims to evaluate the performance of an Ethereum network through an application designed to analyze the cost and time of transactions that store characters in the blockchain. To meet the proposed objective, we designed an application for performing transactions with data inclusion and query on a blockchain, collecting time and cost data. As main conclusions of this work we have: the Ethereum platform proved to be inconstant in relation to the processing time of transactions on the blockchain and the application developed based on blockchain can provide a mechanism to evaluate text-type operations on Ethereum network.

Index Terms—Blockchain, architectures, smart contracts, performance analysis, application.

I. INTRODUÇÃO

Atualmente, *blockchain* é considerada uma inovação tecnológica que proporciona mudanças, atraindo interesse de pesquisadores no cenário mundial [1]. Tradicionalmente, em nossa sociedade cria-se confiança através de intermediários, entidades centrais que armazenam e protegem dados. *Blockchain* substitui a necessidade de intermediários, transferindo a confiança para sistemas descentralizados, podendo registrar transações entre duas partes de forma eficiente, verificável e permanente, eliminando assim a necessidade de terceiros [2]. Além disso, a disponibilidade de todas as transações concluídas para todos os nós da rede torna um sistema baseado em *blockchain* mais transparente que soluções centralizadas [1]. Além de sua aplicação em criptomoeda, *blockchain* possui aplicações com potencial em outras áreas, como Internet das Coisas (IoT) [3], gerenciamento da cadeia de suprimentos [4] e armazenamento de dados médicos [5].

Um dos mecanismos inovadores recentes da *blockchain* são os contratos inteligentes, que são aplicações de execução automática com termos definidos entre duas partes envolvidas

nas transações, escritos em linhas de código em vez de uma linguagem legal. Os contratos inteligentes permitem realizar transações e acordos confiáveis entre partes diferentes anônimas sem a necessidade de uma autoridade central, sistema legal ou mecanismo de execução externo. Após a implantação do contrato inteligente, obtém-se uma *blockchain* distribuída e descentralizada, rastreável, transparente e irreversível [1].

Como o desempenho da plataforma *blockchain* é uma grande preocupação para aplicações corporativas de uma forma geral, tanto na academia quanto na indústria [6], este trabalho, tem como objetivo analisar o desempenho de transações que armazenam variadas quantidades de caracteres em uma *blockchain*. Assim é possível identificar gargalos de desempenho para esse tipo de transação. O tipo de dado caractere foi escolhido para análise devido a grande presença em rotinas diárias dos mais variados tipos de negócios, pois dados específicos de pessoas e empresas em formato texto são usados constantemente em transações de negócios. Assim, torna-se relevante analisar o comportamento em soluções baseadas em *blockchain* para verificar a viabilidade dessas soluções.

Para o atendimento ao objetivo proposto, projetamos uma aplicação para execução de operações de transações de inclusão e consulta de dados em uma *blockchain*, coletando dados de tempo e custo. Assim pode-se obter uma melhor compreensão da previsão de comportamento de sistemas baseados em *blockchain* que irão trabalhar com cadastros de dados do tipo texto, por exemplo, dados de pessoas e empresas. As principais conclusões deste trabalho foram: a plataforma Ethereum se mostrou inconstante em relação ao tempo de processamento das transações na *blockchain* e a aplicação desenvolvida baseada em *blockchain* pode fornecer um mecanismo para avaliar operações do tipo texto na rede Ethereum. O restante do artigo está dividido nas seguintes seções: na Seção II é apresentada uma descrição sobre *blockchain* e contratos inteligentes; a Seção III descreve os trabalhos relacionados; a Seção IV apresenta o *benchmark* desenvolvido; a Seção V descreve o projeto e execução do experimento; por fim, a Seção VI apresenta as conclusões deste trabalho.

II. REFERENCIAL TEÓRICO

A. Blockchain

Blockchain pode ser considerado um livro compartilhado e distribuído que registra transações, mantido por vários nós em uma rede. Cada nó contém a cópia idêntica desse livro-razão, geralmente representada como uma cadeia de blocos, sendo cada bloco uma sequência lógica de transações, que são registros permanentes, transparentes e imutáveis [6]. Cada

Antonio Welligton dos Santos Abreu is with Graduate Program in Computer Science (PCOMP), Federal University of Ceará (UFC), Quixadá, Brazil, e-mail: siwelligton@alu.ufc.br

Emanuel Ferreira Coutinho is with Graduate Program in Computer Science (PCOMP), Federal University of Ceará (UFC), Quixadá, Brazil, e-mail: emanuel.coutinho@ufc.br

Carla Ilane Moreira Bezerra is with Graduate Program in Computer Science (PCOMP), Federal University of Ceará (UFC), Quixadá, Brazil, e-mail: carlailane@ufc.br

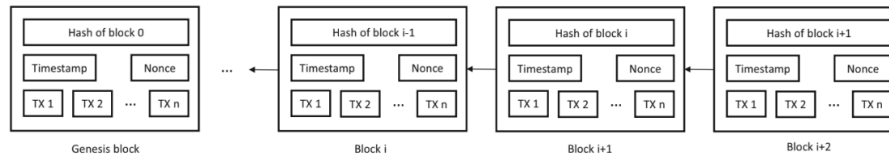


Fig. 1. Exemplo de *blockchain* e seus blocos. [7]

bloco em uma *blockchain* contém seu *hash* computado usando um algoritmo de *hash* ou prova de trabalho conhecido [8] (e.g., *SHA256*, *ethash* e *equihash*) e o *hash* do bloco anterior chamado bloco pai (Fig. 1). O primeiro bloco de uma cadeia é chamado de bloco de gênese, que não possui qualquer pai. O *hash* de cada bloco é calculado com base em seus dados, data e hora atuais e o *hash* do seu bloco pai. Qualquer alteração nos dados de um bloco causa alteração de seu *hash* e invalida todos os blocos subsequentes, e a violação torna-se imediatamente evidente a cada nó membro da cadeia [1].

Não há controle central sobre a operação de uma *blockchain*. A filosofia é que nenhum participante ou grupo possa controlar a infraestrutura da *blockchain*. Todos os participantes da rede têm um papel igual a desempenhar. As transações são validadas pelos nós membros usando um protocolo de consenso, o que garante que todos os nós tenham uma cópia idêntica da *blockchain*. Um novo bloco é considerado verificado somente após a maioria dos nós membros votarem como verdadeiro e confiável usando o protocolo de consenso [1]. As novas transações não são automaticamente adicionadas ao livro-razão. Em vez disso, o processo de consenso garante que essas transações sejam armazenadas em um bloco por um certo tempo (e.g. 10 min no Bitcoin) antes de serem transferidas para o livro-razão. Após este processo, as informações na *blockchain* não podem mais ser alteradas [7].

A implementação ou interação com a *blockchain* pode exigir muitos recursos. Um parâmetro importante para analisar a tecnologia *blockchain* é o próprio bloco, e alguns de seus atributos como tamanho, tempo necessário para mineração e custo. Mineração é o processo de validação de um bloco em *blockchain*, sendo necessário muito poder computacional para se tornar um minerador de *blockchain*. Mineradores são geralmente recompensados, por exemplo, em Bitcoin. Mineração pode afetar todo desempenho do sistema, sendo um conceito crítico que precisa ser considerado [9].

Existem outras plataformas *blockchain* além do Bitcoin. Uma delas, a *Ethereum*, expandiu o conceito de transações de acordo com suas aplicações. Essa plataforma apresentou uma *blockchain* mais genérica, expandindo as transações para operações computacionais chamadas de contratos inteligentes. Um contrato inteligente é uma aplicação autônoma com entradas e saídas pré-definidas que podem ser executadas por um minerador de maneira determinística. Qualquer usuário pode invocar um contrato inteligente, cujo resultado é registrado como uma transação no livro de registro distribuído [10].

B. 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 [11].

No contexto da *blockchain*, os contratos inteligentes são um fluxo de valor baseado em certos termos e condições, sendo como contratos no mundo real. A única diferença é que eles são completamente digitais, o que significa um pequeno código de programação que é armazenado dentro de uma *blockchain*. Existem diferentes plataformas de *blockchain* que podem ser utilizadas para desenvolver contratos inteligentes, sendo a *Ethereum* a mais utilizada [12]. Os contratos inteligentes funcionam como *scripts* armazenados. Como residem na cadeia, eles possuem um endereço exclusivo. 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, de acordo com os dados que foram incluídos no acionamento da transação [13].

Os contratos inteligentes são criados sobre uma plataforma de criptomoeda, por exemplo, a *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 através da publicação de dados assinados que são chamados de transações da rede de criptomoedas. A rede consiste em nós chamados mineradores que propagam informações, armazenam dados, e atualizam os dados aplicando transações. Um esquema de alto nível é exibido na Fig. 2 [14].

Na Fig. 2, o estado do contrato inteligente é armazenado em uma *blockchain*, sendo executado por uma rede de mineradores

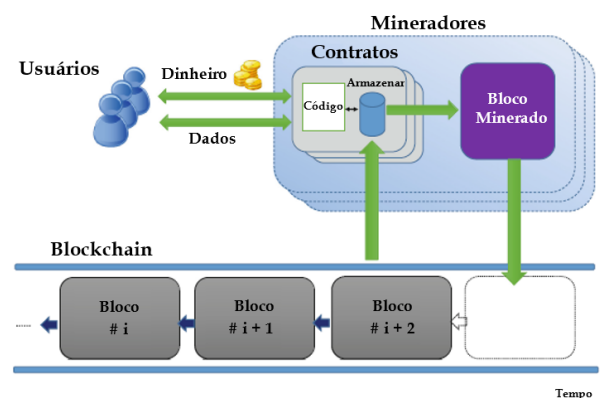


Fig. 2. Sistema de criptomoeda e contratos inteligentes [14]

que chegam a um consenso sobre o resultado da execução, e atualizam o estado do contrato na *blockchain*, podendo os usuários enviar ou receber dinheiro ou dados do contrato. O código do contrato é executado sempre que recebe uma mensagem de um usuário ou de outro contrato. Qualquer usuário pode criar e publicar um contrato através de uma transação na *blockchain*. O código da programação de um contrato é fixado quando o contrato é criado e não pode ser mais alterado [14].

III. TRABALHOS RELACIONADOS

Os trabalhos listados nesta seção estão relacionados à descrição do desempenho de ambientes de *blockchain* em infraestruturas variadas, como o *Ethereum* e o *Hyperledger*.

Thakkar et al. [6] propuseram um estudo empírico para caracterizar o desempenho da plataforma *Hyperledger*, com objetivo de identificar possíveis gargalos de desempenho para obter um melhor entendimento do sistema. Este estudo pretende entender o impacto de vários parâmetros de configuração como tamanho de blocos, política de endosso, alocação de recursos, taxa de transferência da transação e latência, e fornecer várias diretrizes sobre a configuração desses parâmetros. Um dos principais desafios na criação de uma rede eficiente de *blockchain* está relacionado ao conjunto certo de valores para esses parâmetros. As principais contribuições deste trabalho foram fornecer seis diretrizes sobre a configuração desses parâmetros para atingir o desempenho máximo, identificar os três principais gargalos de desempenho e introduzir três otimizações simples para melhorar o desempenho da plataforma.

Aldweesh et al. [15] descreveram uma referência para contratos inteligentes da *Ethereum* que avaliam se a taxa concedida pela execução de contratos inteligentes é proporcional ao esforço computacional necessário. O sucesso da operação do *Ethereum* depende se os incentivos aos mineradores (honorários) para a execução de contratos são proporcionais ao custo, em termos de uso de energia e, portanto, uso da CPU. Em geral, se a taxa recebida não for proporcional ao custo computacional, os mineradores preferem algumas tarefas a outras, assim potencialmente afetando a operação confiável e contínua da *blockchain*. Para demonstrar se os custos e benefícios estão alinhados no *Ethereum*, esse trabalho criou um *benchmark* para usuários e mineradores ajustarem sua confiança na operação da *Ethereum*, podendo comparar tempo de execução de contrato inteligente com o prêmio que um minerador recebe, para determinar se os incentivos se alinham.

Dong et al. [16] apresentaram uma aplicação para avaliação de desempenho de implementações criadas no *Directed Acyclic Graph* (DAG), fornecendo várias cargas de trabalho e adaptadores de amostra que permitem medir o desempenho da implementação do DAG em termos de taxa de transferência, latência e escalabilidade. As comparações de desempenho entre essas implementações ajudam desenvolvedores a avaliar efetivamente diferentes características de desempenho, permitindo identificar gargalos e conseqüentemente, melhorar o desempenho.

A tecnologia *blockchain* ganhou considerável atenção da indústria por seu potencial disruptivo. No caso de mercados

P2P, as *blockchains* permitem não apenas transações mais econômicas, mas aumentam a resiliência compartilhando registros entre os membros da comunidade. Porém, mercados P2P locais em particular exigem a troca de dados entre dispositivos pares, aumentando os requisitos de comunicação devido à natureza descentralizada das redes *blockchain*. Meeuw et al. [17] propuseram uma infraestrutura de comunicação para redes P2P *blockchain*, com uma plataforma de teste que permite comparar a largura de banda usando *benchmarks* e afirmar o tipo de rede de comunicação (3G, LTE, Fiber).

Abreu et al. [18] apresentaram 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.

A Tabela I exhibe os trabalhos relacionados e a relação de semelhanças e diferenças com o trabalho proposto. Alguns critérios para serem respondidos com sim ou não foram definidos para facilitar uma visão geral da relação entre os trabalhos. Percebe-se que todos criaram uma aplicação para analisar uma *blockchain* e a maioria não utiliza um rede *blockchain* pública para análises de *benchmark*. 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*.

TABELA I

COMPARAÇÃO ENTRE TRABALHOS RELACIONADOS.
CRITÉRIOS: APLICAÇÃO = SE CRIOU APLICAÇÃO; PÚBLICA = SE USA *blockchain* PÚBLICA; TEXTO = SE POSSUI FOCO EM OPERAÇÕES DO TIPO TEXTO; MEDIÇÃO = REALIZOU MEDIÇÃO DE *gas* E *ether*

| Estudo | Aplicação | Pública | Texto | Medição |
|----------------------|-----------|---------|-------|---------|
| Thakkar et al. [6] | Sim | Não | Não | Não |
| Aldweesh et al. [15] | Sim | Sim | Não | Sim |
| Dong et al. [16] | Sim | Não | Não | Não |
| Meeuw et al. [17] | Sim | Não | Não | Não |
| Abreu et al. [18] | Não | Sim | Não | Sim |
| Proposta dos autores | Sim | Sim | Sim | Sim |

Embora muitos novos casos de uso e modelos de negócios possam surgir com tecnologias *blockchain*, vários desafios se tornaram aparentes ao implementar um protótipo. A maioria dos trabalhos relacionados realizaram análises de maneira geral em relação às transações, sem focar especificamente em um tipo de operação, diferente do trabalho proposto, que foca nas operações que envolvem dados do tipo texto.

IV. PROPOSIÇÃO DE UMA ARQUITETURA PARA BENCHMARKING EM BLOCKCHAIN

Existe muita preocupação com o desempenho das plataformas *blockchain*, onde as mesmas possuem a missão de conseguir lidar com uma capacidade de enormes volumes de transações com baixa latência [6]. Nas possíveis aplicações baseadas em *blockchain*, muitas delas podem trabalhar com

cadastro de dados do tipo texto, ou seja, informações de identificação e controle como nomes, endereços, contatos e descrições de pessoas e organizações, onde devem ser analisadas e planejadas as transações desses dados, pois geralmente os dados são extensos, podendo requerer espaço e poder de processamento para serem armazenados na *blockchain*.

Para isso, projetou-se uma aplicação descentralizada (DApp) para *benchmarking* (Dapp-Bench), para realizar um estudo empírico das transações com dados variados de número de caracteres na plataforma *Ethereum*, buscando uma relação entre o esforço computacional e tempo de processamento com o número de caracteres usados em uma transação na rede *blockchain*. Destaca-se que esta aplicação não é comercial.

A. Preparação do Ambiente de Experimentação

Existem várias plataformas no mercado atualmente que podem trabalhar com a criação de contratos inteligentes. Além da *Ethereum*, existem outras plataformas com propósito similar, como por exemplo [19]: *Ethereum Classic*, *EOSIO*, *Lisk*, *Tron*, *NEO*, *Stellar*, *Ripple*, *NEM* e *QTUM*. A *Ethereum* foi selecionada para criação da aplicação pois é bastante indicada atualmente para execução de contrato inteligente no contexto da *blockchain* [4]. Ela possui a *Ethereum Virtual Machine* (EVM) [10], onde criou-se a aplicação (contratos inteligentes), que funciona exatamente como programado sem qualquer possibilidade de censura ou fraude, pois o contrato é imutável. A lista a seguir descreve algumas tecnologias associadas ao *Ethereum* utilizadas no desenvolvimento da aplicação:

- **IDE Remix:** IDE online para desenvolvimento de contratos inteligentes.
- **Solidity:** linguagem criada pela própria *Ethereum* para o desenvolvimento de contratos inteligentes.
- **Web3 API:** documentação da web3 JavaScript Dapp API usada para o desenvolvimento da aplicação descentralizada Dapp-Bench.
- **Metamask:** Plugin para o navegador (Chrome ou Firefox) para acessar à plataforma *Ethereum*, funcionando como uma carteira de *ether* e também navega na rede *Ethereum* sem a necessidade de ter uma cópia da *blockchain* instalada no ambiente local.
- **Ropsten:** rede pública de testes da *Ethereum* selecionada para validar a aplicação.
- **Infura:** site para interagir com o contrato inteligente criado, pois precisa-se estar conectado com um nó, que é a porta de entrada para a rede do *Ethereum*.
- **Etherscan:** ferramenta para explorar uma *blockchain*, permitindo analisar transações na plataforma *Ethereum* como forma de ajudar na validação durante o desenvolvimento da aplicação.

B. Implementação do Benchmark

Visualizar a *blockchain* como um componente de software ajuda a entender importantes impactos arquitetônicos sobre o desempenho e a qualidade dos atributos, como segurança, privacidade, escalabilidade e sustentabilidade. *Blockchains* são componentes de software complexos e baseados em rede, que podem fornecer armazenamento de dados, serviços de

computação e serviços de comunicação [20]. Uma referência do modelo arquitetônico para um sistema baseado em *blockchain* está ilustrado na Fig. 3. Este modelo foi tomado como base para entender o contexto envolvendo *blockchain* e o funcionamento da aplicação proposta neste trabalho.

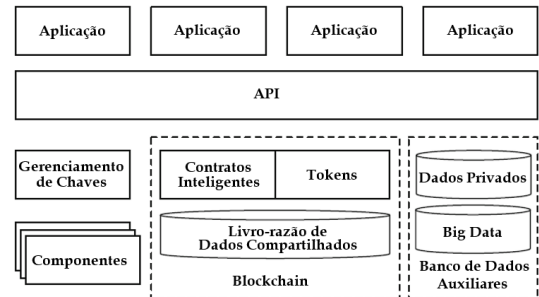


Fig. 3. *Blockchain* em uma arquitetura de software [20].

As informações sobre o mundo externo à *blockchain* podem ser fornecidas por aplicações específicas, que geralmente adicionam essas informações a *blockchain* através de transações. O gerenciamento de chaves e *tokens* permite o controle de permissões e acessos a uma *blockchain*. Parte de uma aplicação pode ser implementada dentro do componente *blockchain* usando o livro-razão *blockchain* e contratos inteligentes. As *blockchains* podem ser usadas como componentes de software, o que pode fornecer o armazenamento de dados, serviços de computação, serviços de comunicação e funções de gerenciamento. Para sistemas baseados em *blockchain*, as principais decisões da arquitetura são sobre quais partes dos dados devem ser colocadas na corrente ou mantidos fora da corrente. No entanto, a quantidade de energia computacional, espaço de armazenamento de dados e controle da leitura dos acessos em uma *blockchain* podem ser limitados. Assim, partes de uma aplicação implementada fora do componente *blockchain* pode hospedar dados *off-line* e lógica da aplicação [20].

Na aplicação criada neste trabalho, para a implantação do contrato inteligente foi utilizado o *plugin Metamask*, que funciona como um intermediário para realizar transações na rede *blockchain*. Foi utilizada a rede pública *Ropsten*, classificada como *testnets*, que utilizam *ethers* fictícios para realizar operações, podendo experimentar diferentes funcionalidades antes de publicar contratos na rede principal (*mainnet*). Através de uma conta criada nesse *plugin* o usuário autoriza ou não as transações na *blockchain*, podendo gerenciar todas transações através dessa conta. Após implantação do contrato inteligente foi desenvolvido a integração do *front-end* da aplicação (Dapp-Bench) com a rede *blockchain*.

A Fig. 4 apresenta uma visão geral do fluxo de execução das tecnologias utilizadas durante as execuções de transações na aplicação criada. Na aplicação, usuários podem executar as transações, recuperar os dados enviados e analisar o desempenho de tempo em segundos, *gas* (unidade de medida do poder computacional necessário para realizar uma operação na rede *Ethereum*) e *ether* (combustível da *Ethereum* cuja finalidade é pagar pelo custo da computação realizada). O objetivo do *Front-end* é prover uma interface para o usuário interagir com o contrato inteligente criado. No *Front-end* se

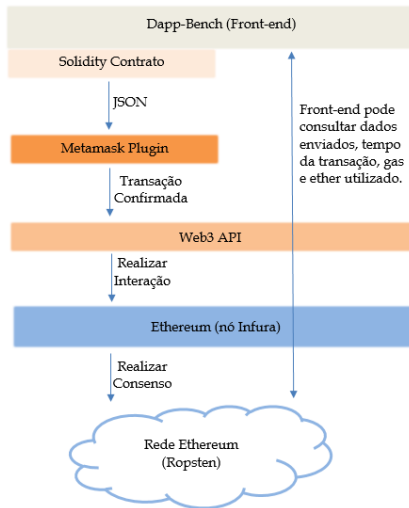


Fig. 4. Fluxo de execução das tecnologias utilizadas nas execuções de transações na aplicação criada.

informa o código do contrato no formato JSON, onde a partir da confirmação da transação através do *plugin Metamask*, a Web3 API reconhece o código Solidity que está no formato JSON e inicia a comunicação com a Rede *testnet Ropsten* da *Ethereum* através do nó criado no *site web* Infura. Para consultar os dados existentes na *blockchain* não é necessário utilizar o *plugin Metamask*, pois após os dados inseridos na *blockchain*, qualquer usuário pode consultar sem a necessidade de instalações extras para acesso aos dados.

V. CONDUÇÃO DO EXPERIMENTO

A. Projeto do Experimento

O objetivo deste experimento é analisar o desempenho de transações que armazenam variadas quantidades de caracteres em uma *blockchain*. Assim é possível identificar gargalos de desempenho para esse tipo de transação.

Para se determinar a média de tempo e custo de uma sequência de transações realizadas na *blockchain Ethereum* através da aplicação *Dapp-Bench* utilizou-se medições conforme projeto. Para isso, 10 categorias de número de caracteres foram criadas, onde para cada categoria realizaram-se 50 transações, contabilizando 500 experimentos no total. As categorias são textos de tamanho 10, 25, 50, 100, 200, 300, 400, 500, 750 e 1000 caracteres. Considerou-se transações com no máximo 1000 caracteres (1KB), pois na literatura há a recomendação do armazenamento de dados de tamanho pequeno na *blockchain* [21], onde o crescente aumento no volume de dados de uma transação faz com que o consumo de tempo e custo aumentem para execução de uma transação, podendo tornar a solução inviável. Por isso, decidiu-se por analisar transações com dados pequenos prevendo um cenário mais adequado para utilizar *blockchain*.

Algumas métricas foram coletadas durante cada transação das categorias: o tempo necessário para seu processamento, o custo de *gas* e *ether* da execução foram coletados. O termo “tempo” é caracterizado como o momento inicial que a transação é aprovada até o momento final em que os dados

gerados pela transação são salvos na *blockchain*. O termo “custo” representa o valor da unidade computacional *gas* e *ether*. Essas são algumas das principais medidas utilizadas na literatura para se avaliar a adoção ou não de uma solução baseada em *blockchain* [15]. Portanto, analisar tempo e custo das transações em um determinado domínio antes de se implantar por definitivo uma solução baseada em *blockchain* pode evitar desgastes e perdas financeiras desnecessárias.

B. Resultados

O experimento foi executado em uma única máquina que envia e analisa todas as transações experimentais (notebook Core i7 de 4.6 GHz com 8GB de memória RAM). A coleta foi realizada entre os dias 09/11/2019 e 21/11/2019, sendo que as operações foram executadas em redes distintas, ou seja, parte das operações foram executadas em uma rede doméstica e outra parte em uma rede de uma instituição pública.

A Fig. 5(a) apresenta a aplicação *Dapp-Bench* onde foram executadas todas as operações na rede *Ethereum*. Para executar uma operação, o texto desejado é informado e enviado. Neste momento, o *plugin Metamask* solicita a confirmação da transação. A Fig. 5(a) apresenta uma das operações com 1000 caracteres realizadas nesse estudo. Esse processo ocorreu em todas transações do estudo realizado. O usuário pode consultar os dados enviados, verificar o tempo gasto, e o custo de *gas* e *ether* utilizados na transação. A Fig. 5(b) exibe a consulta realizada referente à transação da Fig. 5(a).

A Tabela II exibe a média dos resultados obtidos neste trabalho. Nota-se um crescimento de *gas* e *ether* baseado no tamanho do texto utilizado em uma operação. Em contrapartida o tempo para processar as transações não tem o mesmo comportamento, pois se mostrou com uma alta variabilidade, mesmo sendo executadas várias transações com o mesmo tamanho de caracteres. Por este experimento, concluímos que a variável tempo é imprevisível nas operações realizadas na rede *Ethereum*.

TABELA II
VISÃO GERAL DOS RESULTADOS OBTIDOS.

| Qtd Caracteres | Média Tempo | Custo médio GAS | Custo médio Ether |
|----------------|-------------|-----------------|-------------------|
| 10 | 18,04 | 9004,25 | 0,000036017 |
| 25 | 16,76 | 9049,25 | 0,000036197 |
| 50 | 17,80 | 9386,75 | 0,000037547 |
| 100 | 20,02 | 10047,75 | 0,000040191 |
| 200 | 18,46 | 11100,75 | 0,000044403 |
| 300 | 20,80 | 12162,75 | 0,000048651 |
| 400 | 23,14 | 13224,75 | 0,000052899 |
| 500 | 27,70 | 14538,00 | 0,000058152 |
| 750 | 22,46 | 17054,00 | 0,000068216 |
| 1000 | 27,48 | 19827,25 | 0,000079309 |

Esse experimento gerou alguns gráficos, disponíveis pela própria aplicação. Para fim de melhor visualização, os dados foram consolidados em um software estatístico. A Fig. 6 apresenta os gráfico de tempo médio em segundos, de valor médio de *gas* e de valor médio de *ether* das transações realizadas. Observa-se uma variação inconstante do tempo das transações. Por exemplo, a média de tempo das operações com 500 caracteres foi maior que a de 1000 caracteres, tendo uma quebra de padrão de crescimento do tempo baseado no

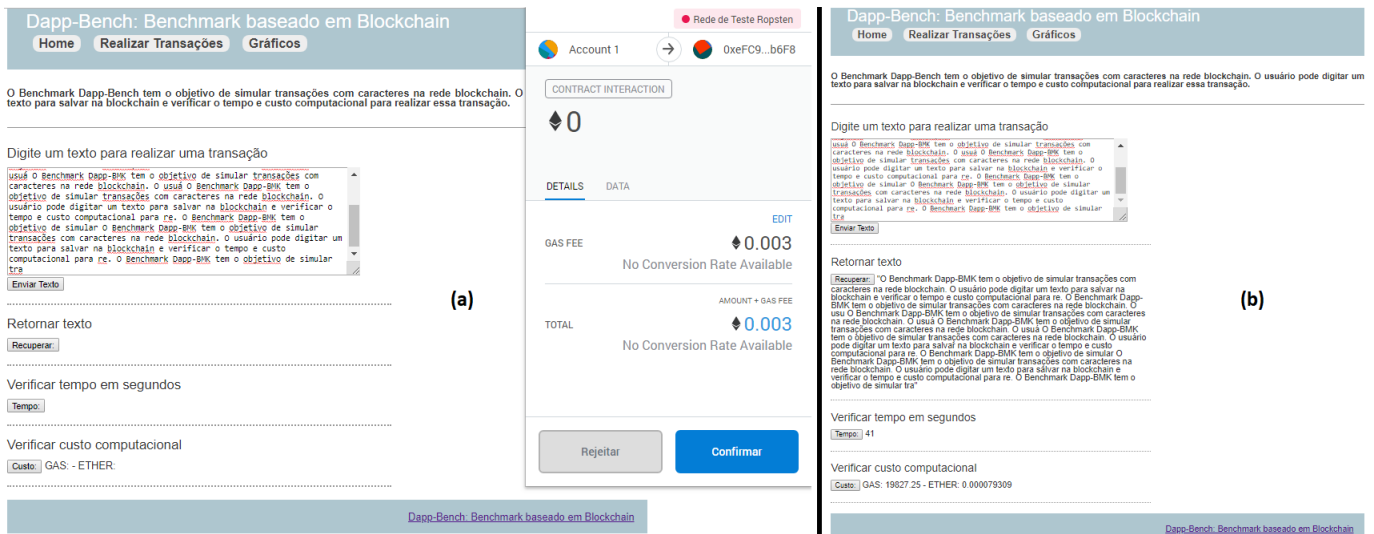


Fig. 5. (a) Operação de salvar texto com 1000 caracteres na *blockchain* e (b) Consulta aos dados enviados, tempo gasto e custo de *gas* e *ether* da transação.

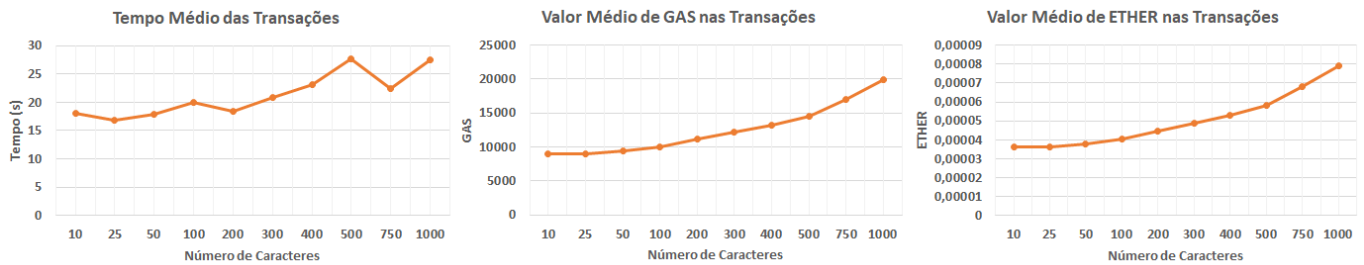


Fig. 6. Gráfico de tempo médio em segundos, de valor médio de *gas* e de valor médio de *ether* das transações realizadas.

tamanho dos dados. Esse mesmo comportamento se repete em outras comparações das categorias. Notou-se no gráfico de valor médio de *gas* usado nas transações um crescimento de custo de *gas* baseado no tamanho do texto usado na transação. Quanto maior o texto, maior o custo computacional para processá-la. Esse mesmo comportamento ocorre no gráfico do *ether*, com crescimento no custo baseado no tamanho do texto usado na transação, ou seja, quanto maior o texto, maior é o valor em criptomoeda para processá-la.

Para complementar a análise dos dados, calculou-se para as operações com 10 e 1000 caracteres a mediana e o desvio padrão. Para 10 caracteres, a mediana dos tempos foi 17 e o desvio padrão 7.07, a mediana do *gas* foi 9004 e o desvio padrão 148.49, e a mediana do *ether* foi 0,00003602 e o desvio padrão 0,00000059. Para 1000 caracteres, a mediana dos tempos foi 19.50 e o desvio padrão 18.12, a mediana do *gas* foi 19827 e o desvio padrão 0, e a mediana do *ether* foi 0,00007931 e o desvio padrão 0.

Para avaliar a distribuição empírica dos dados, histogramas e *boxplots* foram elaborados para analisar o tempo das transações das categorias com menor e maior tamanho (10 e 1000 caracteres), e avaliar os valores discrepantes (*outliers*) (Fig. 7, Fig. 8 e Fig. 9). O losango verde é o desvio padrão, o vermelho é a média, e o azul a mediana.

A Fig. 7 exhibe o *boxplot* de tempo médio em segundos

das transações com 10 caracteres. Observou-se uma variabilidade de tempo considerável, e a ocorrência de dois *outliers* (transações com tempo de 35 e 40 segundos), que apresentam um afastamento dos demais valores. O menor tempo coletado foi 4 segundos e o maior foi 40 segundos, ou seja, os valores de tempo gasto dessas transações estão entre esses intervalos. A mediana foi 17 segundos, ou seja, metade das transações ocorreu com um tempo menor que 17 segundos e a outra metade ocorreu com tempo maior que 17 segundos. O primeiro quartil tem o valor de 13 segundos, ou seja, um quarto das transações foi menor que 13 segundos. Já o terceiro quartil tem valor de 21 segundos, ou seja, 75% das transações ocorreram com tempo menor que 21 segundos. Neste cenário, pode-se afirmar que o alcance da transação com mais tempo e a de menos tempo (40s - 4s) é de 36 segundos. O outro *boxplot* da Fig. 7 apresenta o tempo médio em segundos das transações com 1000 caracteres, onde também observou-se uma variabilidade de tempo considerável, sendo que nesse cenário ocorreu um valor *outlier*, que foi a transação com tempo de 90 segundos. O menor tempo coletado foi 6 segundos e o maior foi 90 segundos. A mediana foi 19,5 segundos, ou seja, metade das transações ocorreu com um tempo menor que 19,5 segundos e a outra metade ocorreu com tempo maior que 19,5 segundos. O primeiro quartil tem o valor de 15 segundos, ou seja, um quarto das transações foram menor que

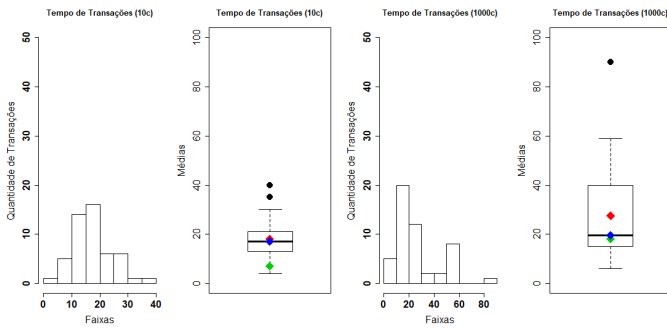


Fig. 7. Histograma e *boxplot* do tempo das transações (10 e 1000 caracteres).

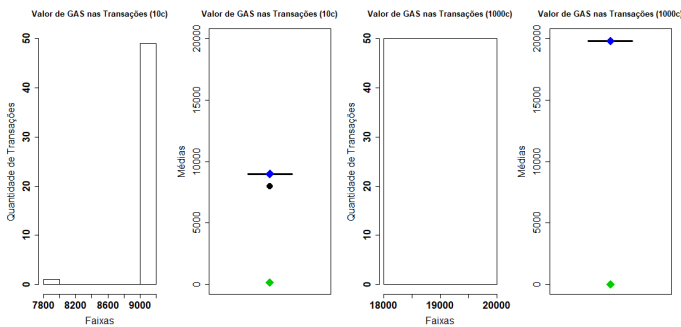


Fig. 8. Histograma e *boxplot* do valor do *gas* (10 e 1000 caracteres).

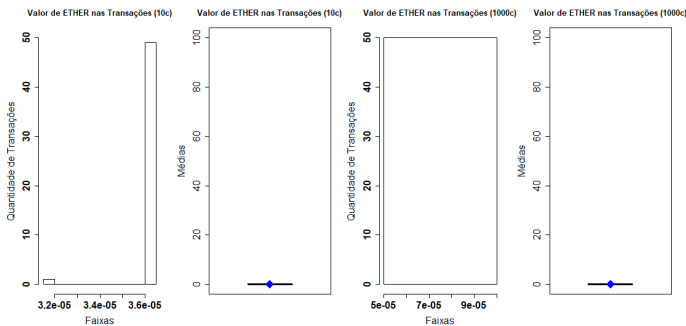


Fig. 9. Histograma e *boxplot* do valor do *ether* (10 e 1000 caracteres).

15 segundos. Já o terceiro quartil tem valor de 40 segundos, ou seja, 75% das transações ocorreram com tempo menor que 40 segundos. Neste cenário, pode-se afirmar que o alcance da transação com mais tempo e a de menos tempo (90s - 6s) é de 84 segundos. Em relação ao desvio padrão, os valores estão baixos, indicando proximidade entre valores de médias ou um valor como a mediana. Mesmo assim, os tempos para transações de 1000 caracteres foram um pouco mais dispersos.

Para fins de análise, Fig. 8 e Fig. 9 exibem os histogramas e boxplots para o *gas* e o *ether*. Como os valores eram praticamente todos iguais (apenas um valor foi diferente, e apenas para uma transação de 10 caracteres), percebe-se nos gráficos uma concentração no mesmo ponto, indicando que todos os valores foram iguais (19827,25 para *gas* e 0,000079309 para *ether*). Isso é reforçado por um desvio padrão baixo, indicando que os dados estão próximos da média ou de um valor.

C. Discussões

Pode-se observar pelos resultados uma relação entre os gráficos de custo de *gas* e *ether*, onde a curva de crescimento dos gráficos são semelhantes. Tem-se uma variação de crescimento do custo em ambos os gráficos, onde o custo aumenta baseado no número de caracteres que é usado na transação. Monitorando o uso dessas duas variáveis pode-se melhorar a compreensão do valor financeiro desejado a ser gasto em possíveis transações do tipo texto, mesmo sabendo que o valor de *gas* e *ether* pode variar ao longo do tempo na rede *Ethereum*, pois basta executar novas simulações na aplicação criada para obter os valores atuais que a plataforma está cobrando nas transações.

Nos gráficos envolvendo tempo, o comportamento se mostrou inconstante durante todo período de simulação do experimento, além de altos valores coletados para executar as transações. Isso leva a acreditar que a plataforma estava instável ou sobrecarregada durante o experimento, podendo ter gerado altos valores de tempo, *gas* e *ether*. Foram encontradas dificuldades em generalizar a forma de executar as transações, ou seja, não foi possível parametrizar a aplicação para executar em lote várias transações em sequência de forma automática, pois foi utilizado o *plugin Metamask* para autorizar cada transação a ser executada de forma manual.

Pelos experimentos há uma relação entre o custo computacional para uma transação na plataforma *Ethereum* (medida em *gas*) e a recompensa pela transação (dada em *ether*). Um aspecto interessante a ser investigado seria identificar qual relação ou proporção que entre ambas valeria mais a pena. Porém, obter este valor iria requerer mais experimentos e pode ser bastante impactado pela rede, que pode influenciar no tempo das transações e consenso. Uma opção poderia ser um limiar para a avaliação de desempenho.

Por fim, em relação a aspectos mais práticos, é importante buscar um alinhamento entre os requisitos de negócio de aplicações que venham a depender de infraestruturas de *blockchain* e os atributos de qualidade requisitados pelas aplicações comerciais. Ao se analisar os resultados do experimento conduzido, que englobam o desempenho do tempo de resposta, *gas* e *ether*, é importante definir ou propor um limite qualidade. Em outras palavras, para aplicações que dependam de determinado limite de desempenho, deve-se projetar um valor para a quantidade de caracteres em transações. Isso impacta na estrutura de dados da aplicação, que para *blockchain* geralmente é pequena devido a questões de desempenho.

VI. CONCLUSÃO

Uma grande variedade de aplicações está surgindo rapidamente com *blockchain*, onde várias organizações da indústria e academia buscam benefícios com essa tecnologia. Conforme resultados obtidos, a plataforma *Ethereum* se mostrou inconstante em relação ao tempo de processamento das transações na *blockchain*, tendo tempos muito distintos em um conjunto de operações semelhantes, ou seja, com o mesmo número de caracteres. A aplicação desenvolvida baseada na tecnologia *blockchain* pode fornecer um mecanismo para avaliar operações do tipo texto na rede *Ethereum*, fornecendo gráficos

na própria aplicação para medir o desempenho das variáveis tempo, *gas* e *ether* das operações realizadas.

O uso da tecnologia *blockchain* não está isento de desafios. Ao avaliar a praticidade de uma solução *blockchain*, os profissionais precisam cuidadosamente avaliar a viabilidade das soluções capazes de atender a diferentes requisitos de negócios. Eles devem considerar os desafios relacionados a segurança, privacidade, custo, escalabilidade e disponibilidade antes de adotar a tecnologia *blockchain*. Como trabalhos futuros, pretende-se realizar um estudo de caso na rede principal da *Ethereum* para a obtenção de uma noção real das transações realizadas na aplicação, observando principalmente a questão de custo das transações no ambiente real. Também tem-se a intenção de identificar causas para as variações nos tempos das transações, assim como identificação de valores mais ideais de custo benefício entre *gas* e *ether*.

REFERENCES

- [1] A. Bosu, A. Iqbal, R. Shahriyar, and P. Chakraborty, "Understanding the motivations, challenges and needs of blockchain software developers: A survey," *Empirical Software Engineering*, vol. 24, no. 4, pp. 2636–2673, 2019.
- [2] D. Macrinici, C. Cartofeanu, and S. Gao, "Smart contract applications within blockchain technology: A systematic mapping study," *Telematics and Informatics*, vol. 35, no. 8, pp. 2337 – 2354, 2018.
- [3] S. Huh, S. Cho, and S. Kim, "Managing iot devices using blockchain platform," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, pp. 464–467, Feb 2017.
- [4] K. Korpela, J. Hallikas, and T. Dahlberg, "Digital supply chain transformation toward blockchain integration," in *50th Hawaii International Conference on System Sciences (HICSS)*, jan 2017.
- [5] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, pp. 25–30, Aug 2016.
- [6] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 264–276, Sep. 2018.
- [7] M. Nofer, P. Gomber, O. Hinze, and D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, pp. 183–187, Jun 2017.
- [8] M. Jakobsson and A. Juels, *Proofs of Work and Bread Pudding Protocols (Extended Abstract)*, pp. 258–272. Boston, MA: Springer US, 1999.
- [9] N. Rifi, E. Rachkidi, N. Agoulmine, and N. C. Taher, "Towards using blockchain technology for ehealth data access management," in *2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME)*, pp. 1–4, Oct 2017.
- [10] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger - v. 3e36772," 2019.
- [11] N. Szabo, "Smart contracts." <http://bit.ly/2Yc9vjb>, 1994. Online; accessed Oct-2019.
- [12] M. Alharby and A. van Moorsel, "Blockchain-based smart contracts: A systematic mapping study," *arXiv preprint arXiv:1710.06372*, 2017.
- [13] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
- [14] K. Delmolino, M. Arnett, A. Kosba, A. Miller, and E. Shi, "Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab," in *Financial Cryptography and Data Security (J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, eds.)*, (Berlin, Heidelberg), pp. 79–94, Springer Berlin Heidelberg, 2016.
- [15] A. Aldweesh, M. Alharby, E. Solaiman, and A. van Moorsel, "Performance benchmarking of smart contracts to assess miner incentives in ethereum," in *2018 14th European Dependable Computing Conference (EDCC)*, pp. 144–149, Sep. 2018.
- [16] Z. Dong, E. Zheng, Y. Choon, and A. Y. Zomaya, "Dagbench: A performance evaluation framework for dag distributed ledgers," in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, pp. 264–271, July 2019.
- [17] A. Meeuw, S. Schopfer, and F. Wortmann, "Experimental bandwidth benchmarking for p2p markets in blockchain managed microgrids," *Energy Procedia*, vol. 159, pp. 370 – 375, 2019. Renewable Energy Integration with Mini/Microgrid.
- [18] E. F. Coutinho, D. J. H. Maia, W. L. B. Bezerra, and A. W. dos Santos Abreu, "Avaliando o custo de contratos inteligentes em aplicações blockchain por meio de ambientes de simulação," in *Anais do II Workshop em Modelagem e Simulação de Sistemas Intensivos em Software*, (Porto Alegre, RS, Brasil), pp. 56–65, SBC, 2020.
- [19] Y. Majuri, "Simply explained: Smart contracts." <https://medium.com/@yakko.majuri/blockchain-definition-of-the-week-smart-contracts-1fbef0d25abf>, 2018. Online; accessed Oct-2019.
- [20] X. Xu, I. Weber, and M. Staples, *Blockchain in Software Architecture*, pp. 83–92. Cham: Springer International Publishing, 2019.
- [21] S. Chen, J. Zhang, R. Shi, J. Yan, and Q. Ke, "A comparative testing on performance of blockchain and relational database: Foundation for applying smart technology into current business systems," in *Distributed, Ambient and Pervasive Interactions: Understanding Humans (N. Streitz and S. Konomi, eds.)*, (Cham), pp. 21–34, Springer International Publishing, 2018.



Antonio Wellington dos Santos Abreu Mestre em Computação pela Universidade Federal do Ceará. Especialização em Redes de Computadores pelo Centro Universitário Católica de Quixadá. Bacharel em Sistemas de Informação pela Universidade Federal do Ceará. Atualmente trabalhando no Centro Universitário Católica de Quixadá como Analista de Tecnologia da Informação, realizando suporte para o Sistema TOTVS RM (ERP). Atua também como Software Developer onde possui as certificações Oracle Certified Professional Java SE 6 Programmer (OCJP) e Oracle Certified Professional Java EE 5 Web Component Developer (OCWCD), tendo interesse nas áreas de Sistemas ERP, Banco de Dados, Engenharia de Software e Desenvolvimento Web.



Emanuel Coutinho Professor Adjunto na Universidade Federal do Ceará (UFC), Campus Quixadá. Graduação em Ciência da Computação pela Universidade Estadual do Ceará (2000). Mestre em Ciência da Computação pela Universidade Estadual do Ceará (2003), trabalhando com grafos, escalonamento e roteamento de veículos. Doutor em Ciência da Computação pela Universidade Federal do Ceará (2014), trabalhando com Computação em Nuvem, métricas e análise de desempenho da elasticidade. Suas áreas de interesse são Computação em Nuvem, Análise de Desempenho, Sistemas de Informação e Engenharia de Software.



Carla Ilane Morerira Bezerra Professora adjunta da Universidade Federal do Ceará (UFC) do Campus Quixadá e é membro do Programa de Pós Graduação em Computação (PCOMP) do Campus UFC Quixadá. Doutorado em Ciência da Computação pela Universidade Federal do Ceará - UFC (2016) e mestrado em Informática Aplicada pela Universidade de Fortaleza - UNIFOR (2009). Atuou como Analista de Sistemas com experiência nas áreas de Melhoria de Processos de Software e Testes de Software. Atuou também como implementadora em diversas empresas do Ceará. Possui interesse em Qualidade de Software, Linhas de Produto de Software, Engenharia de Software Experimental, Refatoração, Manutenção de Software e Sistemas Autoadaptativos.