**UNIVERSIDADE FEDERAL DO CEARÁ**

**CAMPUS QUIXADÁ**

**CURSO DE GRADUAÇÃO EM REDES DE COMPUTADORES**

**FERNANDO CÉSAR LEMOS JÚNIOR**

# ANÁLISE DE DESEMPENHO DE SIMULADORES BLOCKCHAIN: UM ESTUDO COMPARATIVO ENTRE BITCOIN-SIMULATOR, BLOCKSIM E SIMBLOCK

**QUIXADÁ**

**2023**

FERNANDO CÉSAR LEMOS JÚNIOR

# ANÁLISE DE DESEMPENHO DE SIMULADORES BLOCKCHAIN: UM ESTUDO COMPARATIVO ENTRE BITCOIN-SIMULATOR, BLOCKSIM E SIMBLOCK

Trabalho de Conclusão de Curso apresentado ao Curso de REDES DE COMPUTADORES do CAMPUS QUIXADÁ da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Tecnólogo em REDES DE COMPUTADORES.

Orientador: Prof. Dr. Emanuel Ferreira Coutinho.

QUIXADÁ

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

L577a Lemos Júnior, Fernando César.
Análise de desempenho de simuladores blockchain: um estudo comparativo entre bitcoin-simulator, blocksim e simblock / Fernando César Lemos Júnior. – 2023.
66 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Redes de Computadores, Quixadá, 2023.
Orientação: Prof. Dr. Emanuel Ferreira Coutinho.

1. Blockchains (Base de dados). 2. Bitcoin. 3. Desempenho-Avaliação. . I. Título.
CDD 004.6

---

FERNANDO CÉSAR LEMOS JÚNIOR

# ANÁLISE DE DESEMPENHO DE SIMULADORES BLOCKCHAIN: UM ESTUDO COMPARATIVO ENTRE BITCOIN-SIMULATOR, BLOCKSIM E SIMBLOCK

Trabalho de Conclusão de Curso apresentado ao Curso de REDES DE COMPUTADORES do CAMPUS QUIXADÁ da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Tecnólogo em REDES DE COMPUTADORES.

Aprovada em: 05/12/2023

BANCA EXAMINADORA

---

Prof. Dr. Emanuel Ferreira Coutinho (Orientador)
Universidade Federal do Ceará (UFC)

---

Prof. Dr. João Marcelo Uchôa de Alencar
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Jeandro de Mesquita Bezerra
Universidade Federal do Ceará (UFC)

Dedico este trabalho aos amigos, professores e familiares, que contribuíram de algum modo para minha formação.

## AGRADECIMENTOS

Gostaria de expressar minha mais profunda gratidão a todos que fizeram parte da minha jornada universitária. Aos meus queridos amigos e familiares, que sempre estiveram ao meu lado, oferecendo apoio e encorajamento nos momentos mais desafiadores. Aos professores e funcionários, cuja orientação e dedicação incansáveis foram fundamentais para minha formação acadêmica e pessoal. Cada um de vocês, de alguma maneira, contribuiu para o meu crescimento e aprendizado durante este período. Meu sincero obrigado a todos vocês.

"Se eu vi mais longe, foi por estar sobre os ombros de gigantes."  (Isaac Newton.)

## RESUMO

O emprego da tecnologia blockchain vem experimentando um notável aumento tanto em sua utilização prática quanto nas pesquisas desenvolvidas nesse campo. Diante do crescente interesse de organizações por essa tecnologia, torna-se imperativo contar com ferramentas adequadas para avaliar a viabilidade da implementação de redes blockchain, com o intuito de aprimorar os serviços oferecidos. Nesse contexto, foi conduzido um experimento abrangente envolvendo a construção e a análise de desempenho de uma rede blockchain interligando todas as capitais do Brasil.

O experimento envolveu a implementação de uma rede blockchain que abraçasse todas as capitais do país. Para avaliar o desempenho, foram utilizados três simuladores de redes blockchain distintos: BlockSim, SimBlock e Bitcoin-Simulator. O objetivo central foi conduzir uma análise comparativa do desempenho desses simuladores, buscando determinar qual deles seria mais apropriado para ser empregado no contexto das capitais brasileiras. A análise considerou métricas como a duração da simulação, utilização da CPU, utilização da memória, configuração e flexibilidade do simulador, bem como a diversidade de dados do simulador. Adicionalmente, será realizado testes de cargas para acompanhar os resultados gerados pelos simuladores.

A análise abordou tanto os aspectos positivos quanto negativos de cada simulador blockchain, proporcionando uma visão abrangente das capacidades e limitações de cada ferramenta. Este estudo visa fornecer informações cruciais para a tomada de decisões acerca da escolha do simulador mais adequado para implementar uma rede blockchain interconectando as capitais brasileiras, contribuindo assim para otimizar os serviços e maximizar os benefícios dessa tecnologia inovadora. Ao término do experimento, será sugerido qual simulador obteve os melhores resultados com base nas métricas analisadas.



**Palavras-chave**: blockchain; desempenho de redes blockchain; simuladores de blockchain.

## ABSTRACT

The adoption of blockchain technology has experienced a notable increase both in practical usage and in research within this field. Given the growing interest of organizations in this technology, it becomes imperative to have suitable tools to assess the feasibility of implementing blockchain networks with the aim of enhancing the services offered. In this context, a comprehensive experiment was conducted involving the construction and performance analysis of a blockchain network connecting all the capitals of Brazil.

The experiment entailed the implementation of a blockchain network encompassing all the capitals of the country. To evaluate performance, three distinct blockchain network simulators were employed: BlockSim, SimBlock, and Bitcoin-Simulator. The primary objective was to conduct a comparative analysis of the performance of these simulators, aiming to determine which one would be more suitable for deployment in the context of Brazilian capitals. The analysis considered metrics such as simulation duration, CPU usage, memory usage, simulator configuration and flexibility, as well as the diversity of simulator data. Additionally, load tests were conducted to monitor the results generated by the simulators.

The analysis addressed both the positive and negative aspects of each blockchain simulator, providing a comprehensive view of the capabilities and limitations of each tool. This study aims to provide crucial information for decision-making regarding the selection of the most appropriate simulator for implementing a blockchain network interconnecting Brazilian capitals, thereby contributing to optimizing services and maximizing the benefits of this innovative technology. At the end of the experiment, a suggestion will be made regarding which simulator achieved the best results based on the analyzed metrics.



**Keywords**: blockchain; blockchain network performance; blockchain simulators.

**LISTA DE FIGURAS**

## LISTA DE TABELAS

**SUMÁRIO**

# 1 INTRODUÇÃO

A cada dia, com o surgimento de novas tecnologias como Internet das Coisas, *Big Data*, Inteligência Artificial, o mundo real passa a ter uma integração maior com o mundo digital. Essas tecnologias buscam oferecer serviços personalizados que facilitam o trabalho e a interação no dia a dia das pessoas, entre esses serviços podem ser citados casas inteligentes, assistentes virtuais, recomendações de *streaming*, compras personalizadas, rastreamento de veículos, estimativa de tempo de chegada e acompanhamento em tempo real de um produto. A construção dessa integração que potencializa a interação entre homem e máquina com base na revolução digital que está acontecendo vem recebendo o nome de Revolução Industrial 4.0 (NGUYEN; DANG, 2018).

Apesar de no início a sociedade ver com certa relutância essa revolução, as organizações, sejam elas empresas de serviços financeiros, entidades governamentais ou instituições de pesquisa, mudaram de ideia e hoje investem em pesquisas para gerar inovações, buscando a diminuição de custos junto com o aumento da produtividade de seus serviços ofertados (NGUYEN; DANG, 2018).

Entre as novas tecnologias promissoras uma citada é a *blockchain* (NGUYEN; DANG, 2018). Esta tecnologia ganhou notoriedade após um pseudônimo chamado Satoshi Nakamoto lançar o artigo "*Bitcoin: A Peer-to-Peer Eletronic Cash System*" no ano de 2008, em que Nakamoto afirma que conseguiu resolver o problema do gasto duplo nas transações *online* sem o intermédio de terceiros, mostrando sua criptomoeda, o *Bitcoin*, como solução. O conceito de *blockchain* surgiu do livro-razão, que é essencialmente um bloco contendo registros de transações e eventos. Este livro-razão, acessível a todos e confiável, armazena todos os registros de transações realizadas por meio de *Bitcoins*.

Desde 2009 o *Bitcoin* vem causando impactos relevantes na economia global na forma de realizar negócios a ponto de começar a influenciar inclusive o surgimento de outras criptomoedas (SICHEL; CALIXTO, 2018).

Além das criptomoedas as organizações observaram que a tecnologia *blockchain* pode ser usada para desenvolver novos serviços e melhorar os que já existem. Devido a essa possibilidade foi iniciado um processo de investimento em pesquisas na área dessa tecnologia, de modo a descobrir benefícios e melhores formas de usá-la, além de criar sua própria rede *blockchain* para ofertar serviços da melhor forma. Como exemplo pode ser citado o investimento da empresa Telegram no desenvolvimento de sua própria *blockchain* e logo após criar uma rede

em que possa utilizá-la e com isso fazer a integração de seus serviços (BOVéRIO; SILVA, 2018).

Google, Amazon, Facebook, IBM, Goldman Sachs são algumas empresas que estão fazendo pesquisas e experimentos com *blockchain*. Entre os serviços que a *blockchain* pode oferecer para uma organização de acordo com sua necessidade existem os contratos inteligentes, rastreabilidade da cadeia de suprimentos, votação, sistemas de pagamentos, registro de propriedades, integração com internet das coisas, gerenciamento de energia, registros de transferências e pagamentos entre outros que são descobertos conforme o aumento de pesquisas na área (NGUYEN; DANG, 2018).

A seguir alguns exemplos da tecnologia *blockchain* que foram aplicadas em empresas para melhorar os serviços ofertados.

- Logística: A IBM e a Maersk, uma das maiores empresas de transporte marítimo do mundo, colaboraram para criar uma plataforma que utiliza a tecnologia *blockchain* para rastrear e gerenciar o transporte de *contêineres*. Conseguindo assim melhorar a eficiência, segurança e transparência das operações de transporte marítimo (GROENFELDT, 2017).
- Rastreabilidade da Cadeia de Suprimentos: O Walmart emprega a tecnologia *blockchain* para monitorar a procedência dos alimentos, desde a fase de produção até a entrega ao consumidor. Isso possibilita a identificação ágil de questões relacionadas à segurança alimentar e à contaminação de produtos (BTC Soul, 2023).
- Auditoria: A PriceWaterhouseCoopers uma das quatro principais empresas de auditoria do mundo, conhecidas como as "*BIG FOUR*", está implementando a utilização dá *blockchain* para auditorias em tempo real e também oferecendo suporte a vários projetos relacionados à esta tecnologia (InfoMoney, 2023).
- NFT: Azuki Doodles é um modelo de arte colecionável NFT (*Non-Fungible Token*) que recorre à tecnologia *blockchain* para atestar a legitimidade do item colecionável (OpenSea, 2023).
- Propriedade Intelectual: A IBM registrou uma patente intitulada "*A Blockchain for Program Code Execution*". Que aborda o desafio de executar com segurança o código de programas em uma rede *blockchain*, mostrando uma solução que assegura a autenticidade e integridade do código em execução (Forbes, 2023).
- Cartório: No Brasil, o Cartório de Registro de Imóveis da cidade de Pelotas (RS) realizou o primeiro registro de imóvel utilizando a tecnologia *blockchain* do país (BRASILEIRO, 2017).

– Votação: A plataforma de votação online conhecida como Voatz, que utiliza a tecnologia *blockchain*, foi usada no estado da Virgínia Ocidental nas eleições federais dos EUA de 2018 (Forbes, 2020).
– Registros de Transferências e Pagamentos: O Banco Industrial e Comercial da China implementou a tecnologia *blockchain* para agilizar operações comerciais, ampliar a transparência e possivelmente gerar economias substanciais de bilhões de dólares (Forbes Brasil, 2021).

Alguns dos motivos da popularidade e crescimento de pesquisas sobre a tecnologia *blockchain* está ligado à sua forma de registrar informações. A *blockchain* realiza o registro de informações usando uma estrutura descentralizada e distribuída que geram segurança e transparência no armazenamento de dados (NAKAMOTO, 2009). Devido à transparência e segurança que a *blockchain* oferece no registro de dados, é notório que ela seja bastante útil no dia a dia, já que a segurança é um fator importante para serviços que manipulam dados (SOARES *et al.*, 2021).

Com o aumento de aplicações e pesquisas na utilização de *blockchain* e a importância de manter a segurança e eficiência desses serviços, as organizações necessitam ainda de ferramentas apropriadas para realizar a avaliação desses sistemas. Muitos desses sistemas são projetados para casos de uso específicos (FARIA; CORREIA, 2019). O desenvolvimento de sistemas *blockchain* mais complexos, sem o uso de ferramentas adequadas para a avaliação de desempenho antes da implementação, pode gerar um impacto negativo no processo de implantação (SMETANIN *et al.*, 2020).

O impacto negativo causado pela falta de uma avaliação de desempenho pode acabar fazendo com que organizações desistam de implementar seu sistema, pois o custo de recurso computacional, financeiro e tempo de desenvolvimento podem tornar o sistema inviável de ser mantido, gerando assim prejuízo para as organizações (THOMAZ *et al.*, 2021). Sendo assim, é preciso pesquisar bem a ferramenta que será usada para uma avaliação de desempenho e analisar se ela se encaixa de forma adequada ao sistema. Escolher a ferramenta ideal tende a ser uma tarefa bem complexa devido ao grande número de ferramentas disponíveis no mercado (THOMAZ *et al.*, 2021).

Uma alternativa eficiente para a avaliação de desempenho de um sistema é a simulação (MEDEIROS *et al.*, 1). Ferramentas baseadas em simulação se destacam em termos de escalabilidade, logística, custo-benefício e fácil configuração (MEDEIROS *et al.*, 1). Além disso,

a simulação oferece um ambiente propício para a criação de vários cenários pré-estabelecidos (MEDEIROS *et al.*, 1). Entre as ferramentas de simulação de *blockchain* podem ser citadas: Shadow-bitcoin, VIBES, SIMBA, IGreen, SimBlock, BlockPerf, BlockSim:Alharby Paulavičius *et al.* (2021).

Portanto, observando que a avaliação de desempenho de um sistema *blockchain* é essencial para o sucesso deste em uma organização, que ferramentas baseadas em simulação são indicadas para a avaliação e que existem dúvidas relacionadas a qual ferramenta seria ideal usar devido à complexidade de alguns simuladores. Este trabalho pretende fazer uma análise de desempenho de três simuladores *blockchain* ao construir um cenário que simula uma rede *blockchain* que passe por todas as capitais do Brasil. Alguns dados da *blockchain* simulada serão construídas com base na *blockchain* da data atual deste trabalho. Após a implementação do cenário será realizado testes de cargas de nós para avaliar como cada simulador se comporta ao realizar a simulação de criação da rede *blockchain*.

O cenário do experimento foi escolhido ao levar em consideração organizações que queiram distribuir seus serviços ou realizarem pesquisas sobre uma rede *blockchain* que liga todos os estados do Brasil e o Distrito Federal. Cada estado é simbolizado por sua capital, uma vez que essas cidades são centros políticos, culturais e econômicos proeminentes, tornando-as escolhas naturais para representar seus estados na rede *blockchain*. Os testes de cargas de nós são para avaliar como provavelmente o simulador irá se comportar com o crescimento de números de nós na rede.

Neste estudo, foram selecionadas três ferramentas de simulação de redes *blockchain*: SimBlock (Tokyo Institute of Technology, 2023), BlockSim (FARIA; CORREIA, 2019) e Bitcoin-Simulator (GERVAIS *et al.*, 2016). A escolha desses simuladores seguiu uma pesquisa abrangente sobre as opções disponíveis. Dentre os simuladores analisados, esses três se destacaram devido ao seu código aberto, que é um tipo de licença de software que permite a qualquer pessoa ver, usar e modificar o código, serem desenvolvidos por institutos de pesquisa de renome, o número de recomendações relevantes que seus repositórios no Github possuem e às múltiplas referências encontradas em trabalhos acadêmicos, especialmente no que diz respeito à utilização do código-fonte original dessas ferramentas para criar novos simuladores.

Após a implementação do cenário descrito em cada simulador e a coleta dos dados resultantes, pretende-se realizar uma análise de desempenho com base nesses dados. Em seguida, será discutido os resultados dos experimentos para responder à seguinte questão deste trabalho:

Qual seria o melhor simulador de rede *blockchain* para uma organização que deseja implantar uma rede interligando todas as capitais do Brasil?

Para analisar a eficiência dos simuladores de rede *blockchain* usados para desenvolver o experimento será levado em consideração as seguintes métricas de desempenho: Duração da Simulação, Utilização da CPU, Utilização da Memória, Configuração e Flexibilidade do Simulador, Diversidade de Dados do Simulador.

A avaliação de qual simulador obteve os melhores resultados para o experimento, será realizada a partir de tabelas e gráficos que conterá os resultados obtidos das avaliações de desempenho de cada simulador, e através desse resultado será realizado uma análise empírica para avaliar qual simulador de rede *blockchain* se encaixa melhor no cenário de uma rede *blockchain* que interligue todas as capitais do Brasil.

Por fim, este trabalho tem a intenção de poder ajudar possíveis futuros desenvolvedores de sistema *blockchain* a escolherem o simulador ideal para avaliar o desempenho do seu sistema para atingir um melhor desempenho do sistema na organização.

## 1.1 Objetivo Geral

O objetivo geral deste trabalho é realizar uma análise de desempenho de três simuladores *blockchain* que são o Bitcoin-Simulator, BlockSim e SimBlock, utilizando como caso de uso a simulação de uma rede *blockchain* que interliga todas as capitais do Brasil, e após a análise discutir qual seria o melhor simulador para esse cenário considerando as seguintes métricas de desempenho: Duração da Simulação, Utilização da CPU, Utilização da Memória, Configuração e Flexibilidade do Simulador, Diversidade de Dados do Simulador.

## 1.2 Objetivos Específicos

1. Desenvolver um cenário para o experimento.
2. Implementar o cenário no simulador Bitcoin-Simulator.
3. Implementar o cenário no simulador SimBlock.
4. Implementar o cenário no simulador BlockSim.
5. Conduzir uma análise de desempenho dos dados gerados dos simuladores.

# 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, será apresentado alguns conceitos essências que servirá como base para o desenvolvimento deste trabalho.

## 2.1 Blockchain

Segundo Nakamoto (2009), a palavra *blockchain* é uma junção das palavras *block* (bloco) e *chain* (corrente). É uma tecnologia em que os blocos são conectados por meio de uma corrente, formando uma rede *peer to peer*. Juntos a rede forma um livro razão compartilhado e descentralizado responsável por armazenar dados de transações realizadas.

Praticamente *blockchain* é uma corrente de blocos em que cada bloco desses consegue acessar e armazenar o valor de um registro de transação de um ativo de forma segura. Um ativo pode ser qualquer coisa que tenha algum valor como: *tokens* de segurança, dados, moedas virtuais, etc. Esses registros ficam disponíveis para serem consultados por qualquer outro nó além de que são imutáveis fazendo com que se tornem uma ótima maneira de rastrear os ativos e garantir sua autenticidade.

Devido a esses benefícios, a *blockchain* foi usada na construção de moedas virtuais como *Bitcoin*, Ethereum, Tether, etc. A moeda em que Nakamoto cita em seu artigo que também é responsável pela explosão de criptomoedas no mercado, além de influenciar no investimento em pesquisas de *blockchain* é chamada de *Bitcoin*.

Nakamoto (2009) destaca que muitos sistemas de pagamentos tradicionais e financeiros são centralizados. Isso significa que uma autoridade central tem controle de todas as transações de ativos que ocorrem. No entanto, isso pode ser problemático, pois a autoridade central pode ser corrompida ou atacada, levando a falhas no processo de transição de unidades de valor.

Para evitar esse problema da autoridade central, Nakamoto (2009) sugere uma solução descentralizada para os sistemas financeiros. Esta solução é baseada na apresentação de sua criptomoeda que carrega dentro de si a tecnologia *blockchain*. Com isso é resolvido o problema da autoridade central de uma transação passando a eliminá-lo, e também eliminando o problema do gasto duplo em sistemas descentralizados.

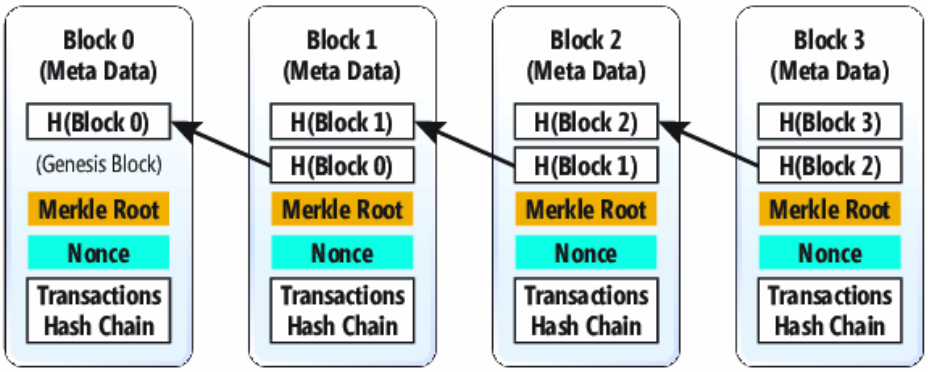
Para resolver o problema do gasto duplo a *blockchain* registra cada transação em blocos de transações. Cada bloco contém um *hash* do bloco anterior, um *timestamp*, que é a hora

que o bloco foi criado, um *nonce*, que é um número arbitrário usado para garantir a autenticidade e validade do bloco.

Além disso, cada bloco contém informações sobre as transações validadas e inseridas no bloco, bem como outras informações importantes. Uma dessas informações é a *Merkle tree root*, que é uma estrutura responsável para criar um tipo de *hash* para facilitar a localização do bloco em uma árvore *Merkle*, gerando uma maneira simples de verificar a integridade do bloco criado. Esses blocos se conectam mutuamente em ordem cronológica sendo gerado assim uma corrente de blocos.

Os mineradores de rede, responsáveis por ajudar a processar e validar transações na rede, usam seu poder computacional para resolverem complexos problemas computacionais. O minerador que encontrar a solução primeiro pode adicionar o próximo bloco a corrente, após um bloco ser adicionado, todos os nós verificam e validam a transação e o novo bloco criado. Se um minerador tentar gastar a mesma unidade de valor novamente, a própria rede irá detectar a tentativa e rejeitara a transação. A Figura 1 um conjunto de blocos vinculados que juntos formam uma *blockchain*.

Figura 1 – Blockchain




Fonte:(Microsoft, 2023)

Nakamoto (2009) mostrou que o sistema de *blockchain* do *Bitcoin* garante que cada unidade de valor só possa ser gasta uma única vez. Conseguindo assim eliminar o problema do gasto duplo, além da remoção da necessidade de uma autoridade central como mediador de transações. Através desta solução a *blockchain* mostrou que pode ser uma tecnologia resistente à fraude e confiável.

Batubara *et al.* (2019) cita possíveis tipos de classificação de *blockchain*, em que

cada tipo de classificação tem influência na maneira em que é autorizada a permissão do acesso e controle de dados. A aplicação do tipo correto de *blockchain* vai depender da finalidade da rede e do nível de autorização de usuário que será disponibilizado pela organização que controla a rede.

Por meio de cada tipo de *blockchain* é possível restringir quais usuários têm permissão para escrever e ler dados, além de quem pode participar do mecanismo de consenso da rede. Essa classificação pode ser dividida em dois tipos principais de *blockchain*: pública e privada.

Esse tipo de classificação é responsável por definir quem tem acesso para ler os dados no livro razão. Além disso, existem *blockchain* com permissão e sem permissão que regula quem pode iniciar transações e participar do mecanismo de consenso da rede para validar novos blocos.

A seguir é listado quais os principais tipos de *blockchain* que são utilizados.

- *Blockchain* público e sem permissão: Permite que qualquer usuário participe da rede *blockchain*, autorizando o usuário a poder ler os dados no livro-razão, iniciar transações e participar do mecanismo de consenso. Como exemplo desse tipo de *blockchain* existe o tipo usado no *Bitcoin*.
- *Blockchain* público com permissão: Permite que qualquer usuário participe da rede *blockchain*, desde que esse seja aprovado pelo processo de autorização, apenas os participantes autorizados têm permissão para ler os dados no livro razão, iniciar transações e participar do mecanismo de consenso.
- *Blockchain* privado sem permissão: Conhecido como *blockchain* de consórcio, permite que qualquer usuário participe da rede, mas apenas os participantes autorizados por um grupo de participantes podem ler dados no livro razão, iniciar transações e participar do mecanismo de consenso.
- *Blockchain* privado com permissão: Permite que qualquer usuário participe da rede *blockchain* apenas após o processo de autorização, e apenas o operador da rede pode iniciar transações e participa do mecanismo de consenso, enquanto autorizado os participantes são capazes de ler dados.

Batubara *et al.* (2019) também afirma que o sucesso da *blockchain* gerou uma influência no crescimento de implementações de contratos inteligentes, já que os contratos agora poderiam ser implementados e aplicados na *blockchain* de forma que garantisse a sua transparência e imutabilidade.

Contratos inteligentes são programas de computadores que armazenam regras para

negociar os termos de um acordo na *blockchain*. Esses acordos não precisam de intermediários para negociar a transação, precisa apenas que as partes envolvidas cumpram os pré-requisitos estabelecidos nos contratos armazenados na *blockchain*. Caso as regras do contrato seja comprida por ambas as partes, o contrato irá executar a programação que foi definida baseada nas condições em que as partes envolvidas concordaram.

## 2.2 Simulação

Segundo Banks (1999), simulação pode ser definida como a imitação da operação de um processo ou sistema do mundo real ao longo do tempo. É uma técnica computacional que envolve a geração de uma história artificial do sistema. Através dessa história artificial gerada é possível deduzir as características operacionais do sistema real representado.

Através da sua metodologia, a simulação se tornou uma ferramenta indispensável para a resolução de problemas do mundo real. Ela pode ser usada para descrever e analisar o comportamento de um sistema a partir de questionamentos feitos no sistema real. Com base nas necessidades encontradas no sistema, são elaborados modelos que descrevem as relações entre as variáveis do sistema que será simulado.

Após a elaboração do modelo, um *software* de simulação é utilizado para simular o sistema. Essa simulação conterá um cenário do sistema com todos os principais componentes descritos no modelo. Permitindo que o sistema possa ser analisado da melhor forma possível.

Carson (1993) afirma que um modelo pode ser descrito simplesmente como uma representação de um sistema real, com limites claros entre a parte do mundo simulado e o mundo exterior. Os componentes do sistema são geralmente representados em diferentes níveis de detalhe na simulação, considerando sua relevância para os objetivos de modelagem. A complexidade do modelo deve ser adequada de forma que possa responder os questionamentos levantados do sistema.

Existem vários tipos de modelos de sistemas, entre eles podem ser citados: modelos de simulação de eventos discretos, modelos matemáticos, modelos descritivos, modelos estatísticos e modelos de entrada e saída.

Entre esses modelos, o modelo de eventos discretos ganha destaque por ser dinâmico e reutilizável. Ele permite que o usuário crie e simule várias estratégias e cenários. Esse modelo é responsável por criar e simular sistemas onde as mudanças ocorrem em pontos discretos no tempo. Nesses tipos de sistema, as mudanças ocorrem em ‘eventos’ localizados em pontos

precisos no tempo. Cada mudança no sistema é representado por um evento. Após a ocorrência desses eventos, uma análise dos resultados gerados pela simulação é realizada.

A escolha do melhor modelo irá depender do tipo de simulação que será implementado e dos objetivos que devem ser alcançados. Dependendo do objetivo final podem ser criados vários modelos diferentes ou semelhantes para serem usados em um simulador a fim de analisar o comportamento do sistema.

Banks (1999) descreve algumas vantagens da simulação entre estas podem ser destacadas:

- A possibilidade de testar cada adição, alteração de funcionalidade ou novas possibilidades de mudança do sistema sem gastar grande parte de recursos destinado a este e analisar se aquela mudança gerará um impacto positivo ou negativo no sistema, avaliando se vale a pena gastar recursos.
- Ter o controle do tempo de simulação a fim de comprimir ou expandir o tempo da análise conforme a investigação do fenômeno que será realizada, importante para analisar minuciosamente processos que no sistema levaria, horas ou segundos.
- Entender os motivos que podem levar o sistema a ter certos comportamentos, após realizar uma análise detalhada da cena em questão reconstruída do sistema real, e a partir de análise, se torna possível diagnosticar erros. Esses que podem surgir descobertos erros vão desde o mais simples aos mais críticos, além dos possíveis impactos que esses erros poderiam causar no funcionamento do sistema.
- Identificar gargalos de produção de forma mais fácil simulando possíveis causas de atrasos nos processos do sistema.

A simulação também contém algumas desvantagens, como elaboração de modelos complexos e demorados, elaboração de modelos incertos que não geram resultados de valor para serem analisados, dificuldade de realizar certas análises de resultados para validá-los, custos dependendo da tecnologia que será simulada.

Pensando em como diminuir os impactos da simulação nos sistemas e aproveitar todos os benefícios que esta oferece, foram escritos softwares de simuladores a fim de tornar o processo de simulação viável vantajoso.

## 2.3 Analise de Desempenho

Segundo Jain (1990), o processo de analisar, medir e entender o comportamento de um sistema de computador sobre diferentes cargas de trabalho e condições pode ser definido como análise de desempenho. Esse processo é essencial para o projeto, avaliação e otimização de sistemas de computacionais que vão desde sistemas simples como computadores pessoais até sistemas mais complexos como data centers e redes de computadores.

Através da análise de desempenho é possível identificar pontos criticos do sistema ou gargalos que limitam a capacidade do sistema para lidar com determinada demanda ou limite de tráfego, a análise pretende maximizar o desempenho de um sistema conforme a quantidade de recursos fornecidos, sendo assim a análise é essencial para o que projeto, desenvolvimento e operações de sistemas de computadores que se tornem confiáveis e eficientes.

Jain (1990) aborda três técnicas adequadas para serem aplicadas no processo de avaliação de desempenho: simulação, modelagem analítica e medição. Para escolher qual técnica será a melhor técnica para o processo de avaliação é necessário considerar o estágio do ciclo de vida de um sistema.

Conhecer bem um sistema e a técnica adequada para realizar o processo de avaliação de desempenho é essencial já que alguns erros de avaliação acontecem por justamente as técnicas de avaliação não serem aplicadas corretamente. Alguns desses erros citados por Jain (1990) podem ser: falta de objetivo definido para análise; cargas de trabalho não representativas; métricas, fatores definidos de forma incorreta; análises errôneas e muito complexas; falta de clareza; falta de clareza nos resultados; entre outros erros que utilizam o processo de análise.

Jain (1990) afirma que a maioria dos problemas de um processo de avaliação de desempenho está normalmente ligado ao mal uso das métricas, cargas de trabalho e técnicas de avaliação, já que após serem usadas para resolver um problema normalmente essas abordagens não podem ser usadas para resolver outros devido muitos desses problemas serem únicos. Entretanto, apesar disso existem algumas etapas que são comuns e ajudam a evitar erros na implantação de projetos de avaliação de desempenho:

1. Definir o sistema: Definir o que parte do sistema deverá ser alvo de estudo e quais os objetivos devem ser alcançados.
2. Listar serviços e resultados: Listar serviços e resultados úteis do sistema que devem ser incluídos na avaliação.
3. Selecionar métricas: Selecionar as métricas que serão usadas para o processo de avaliação

do sistema.

4. Listar parâmetros: Listar todos os parâmetros que afetam o desempenho do sistema.
5. Selecionar Fatores de Estudo: Selecionar o nível de detalhamento do estudo de alguns parâmetros da avaliação.
6. Selecionar a técnica de avaliação: Selecionar a técnica adequada que será usada na avaliação.
7. Selecionar a carga de trabalho: Selecionar a carga de trabalho baseada nas solicitações dos serviços que serão avaliados do sistema.
8. Projetar experimentos: Realizar uma sequência de experimentos para conseguir o máximo de informação possível com o menor gasto de recursos.
9. Analisar e interpretar dados: Organizar os dados gerados na avaliação para ser possível analisar e interpretá-los a fim de realizar uma dedução.
10. Apresentar os resultados: Apresentar os resultados descobertos na análise da melhor forma possível.

Caso necessário, após o fim desses passos, se for preciso revisitar um passo para mudar algo que se adéque melhor ao objetivo final, os passos podem ser revisitados e modificados para ser possível começar o processo de avaliação novamente até obter os resultados relevantes necessários.

## 2.4 Simuladores Blockchain

### *2.4.1 Bitcoin-Simulator*

O Biticoin-Simulator é um simulador de rede *blockchain* baseado no simulador NS-3 (Network Simulator-3). NS-3 é um simulador de redes de eventos discretos desenvolvido para trabalhar com pesquisas, desenvolvimento e uso educacional.

O objetivo do Bitcoin-Simulator é estudar como características de redes, parâmetros de consenso e modificações de protocolo afetam a segurança, escalabilidade e eficiência de *blockchains*. Como mecanismo de consenso o Bitcoin-Simulador usa o Proof of Work e consegue simular redes *blockchain* utilizando as criptomoedas *Bitcoin*, *Litecoin* e *Dogecoin*.

É um módulo implementado a partir da linguagem de programação C++, de código aberto, cujo código é aberto e livre para acesso, modificação e distribuição dos usuários. Permite a definição de alguns parâmetros de entrada para a construção da simulação de forma dinâmica.

Ao final da simulação do Bitcoin-Simulator os dados gerados pelo simulador sobre a rede de *blockchain* simulada são exibidos pelo terminal para serem analisados.

O Biticoin-Simulator foi implementada por Arthur Gervais, professor na *Imperial College London*, que tem contribuições na área de segurança da informação e no estudo de *blockchain*.

A Figura 2 mostra a página inicial do site que fica a documentação do Biticoin-Simulator.

Figura 2 – Página inicial Bitcoin - Simulator


Fonte: (GERVAIS, 2023)

### *2.4.2 BlockSim*

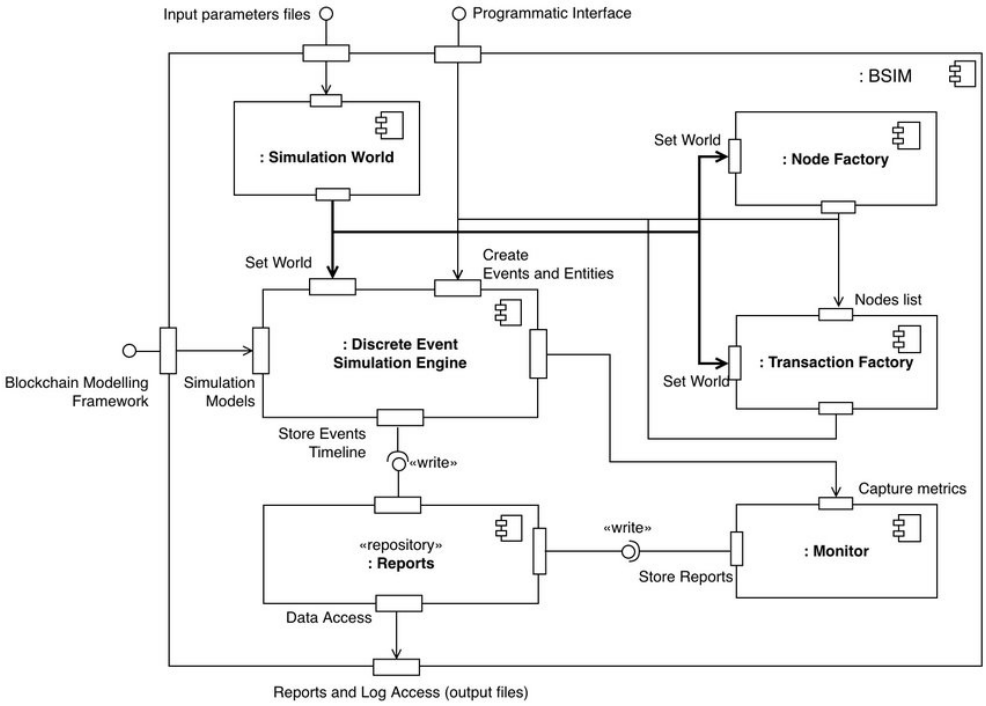
BlockSim é um *framework* de simulação criado para auxiliar no projeto, implementação e avaliação de novos ou existentes *blockchains*. O BlockSim consegue fornecer informações de como um sistema *blockchain* opera a partir modelos abstratos de simulação estabelecido, simulando vários *blockchains* em uma rede de larga escala, mas sem a sobrecarga de uma implementação e implantação de uma rede real.

É implementado através da linguagem de programação Python assim como seus modelos, como mecanismo de consenso o Blocksim usa o Proof of Work, é um software de código aberto cujo o código é aberto e livre para acesso, modificação e distribuição dos usuários, que está sob licença MIT.

O BlockSim é um simulador de eventos discretos capaz de representar fenômenos aleatórios por introdução de distribuições de probabilidade para certos eventos, seguindo o modelo de simulação estocástica (FARIA; CORREIA, 2019).

A Figura 3 mostra o modelo da arquitetura do BlockSim retirado do artigo localizado no repositório oficial do projeto no Github.

Figura 3 – Arquitetura do BlockSim




Fonte: (FARIA; CORREIA, 2019)

### *2.4.3 SimBlock*

SimBlock é um simulador de rede *blockchain* orientado a eventos, que considera a geração de blocos e a mensagem transmissão / recepção como eventos. O objetivo do desenvolvimento do SimBlock é auxiliar no uso em testes de desempenho e segurança *blockchain*. Ele é um software de código aberto, cujo o código é aberto e livre para acesso, modificação e distribuição dos usuários. É implementado na linguagem de programação Java. Este simulador foi desenvolvido pelo *Distributed Systems Group, Tokyo Institute of Tecnology*.

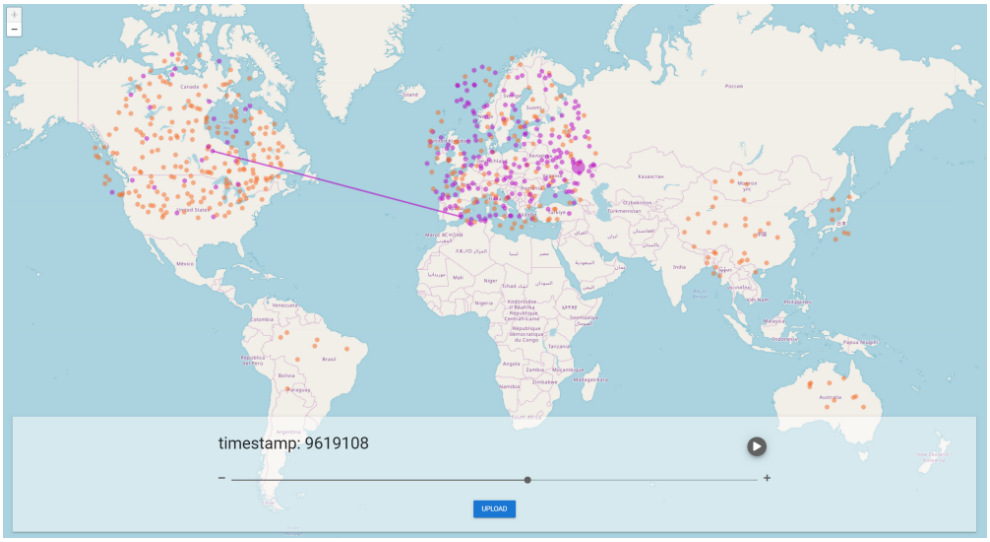
Um dos recursos que o SimBlock possui é uma ferramenta de visualização, pela qual é possível ver a transição da propagação do bloco e desenhar nós de comportamento. Por meio de arquivos JSON é possível o usuário visualizar informações sobre criação, transição e propagação de blocos para toda a rede *blockchain*. Este simulador pode criar um nó, gerar o bloco, transmitir o bloco para outro nó como evento adequado.Os mecanismos de consenso utilizado pelo Simblock são o Proof of Stake e Proof of Work.

O SimBlock é considerado uma maneira fácil de implementar o algoritmo de seleção de nós vizinhos, se tornando assim uma ferramenta adequada para pesquisa de rede *blockchain*.

O simulador está disponível para sistemas operacionais como Windows, macOS, Linux ou qualquer plataforma Unix que suporte Java com no mínimo JDK versão 1.8.0 e *Cradle*

versão 5.1.1 (AOKI *et al.*, 2019). A Figura 4 mostra a interface gráfica gerada após o resultado de uma simulação de uma rede *blockchain* retirada do repositório oficial do SimBlock no Github, através dela é possível ver como os nós se comunicam ao longo do tempo.

Figura 4 – Tela de saida do SimBlock




Fonte: (Tokyo Institute of Technology, 2023)

# 3 TRABALHOS RELACIONADOS

Neste capítulo, será apresentado alguns trabalhos relacionados que correlacionam com o tema abordado neste trabalho.

## 3.1 A Systematic Review and Empirical Analysis of Blockchain Simulators

Em Paulavičius *et al.* (2021), é realizado uma revisão sistemática e análise empírica de simuladores de *blockchain*, o artigo começa mostrando sobre a importância que a tecnologia *blockchain* vem adquirindo impulsionada pelo surgimento do *Bitcoin*, mostrando que a *blockchain* virou uma tecnologia inovadora que se tornou alvo de pesquisa do governo, industria e academia. Logo após é destacado que sistemas que implementam *blockchain* sem passar por uma avaliação de desempenho podem desenvolver resultados negativos na fase de implantação.

São citadas algumas técnicas para realizar essas avaliações, entre as técnicas é destacada a simulação de sistemas. Logo após é mostrado a importância dos simuladores de *blockchain* na pesquisa e desenvolvimento de sistemas *blockchain*, já que esses sistemas são complexos e caros de se desenvolver e testar em ambiente de produção.

O trabalho apresenta uma revisão sistemática dos simuladores de *blockchain* disponíveis, analisando suas funcionalidades, características e limitações. São selecionados os principais simuladores de *blockchain* e destacadas suas principais características e funcionalidades. Após a seleção dos simuladores, é realizada uma análise empírica, que inclui a execução de experimentos para comparar a precisão e o desempenho de cada simulador. Utilizando uma série de métricas, como consumo de recursos, tempo de execução e precisão de resultados, é realizada uma avaliação de desempenho dos simuladores.

Como resultado é mostrado as vantagens e desvantagens dos simuladores quais são mais indicados para testes de desempenho e possíveis futuras pesquisas que podem ser abordadas com simuladores para o uso de pesquisa e desenvolvimento de sistemas *blockchain*.

A principal diferença entre o trabalho de (PAULAVIčIUS *et al.*, 2021) e esse são a carga de trabalho para o experimento de análise de desempenho, cenário estabelecido e a mudança de simulador com o acréscimo do BlockSim, já que ele foi citado mas não foi usado no experimento. Para o experimento desse trabalho será usado o Bitoin-Simulator, BlockSim e SimBlock.

## 3.2 Performance Comparison of SimBlock to NS-3 Blockchain Simulators

Em Hanggoro e Sari (2021), é feita uma análise de desempenho entre dois simuladores: o SimBlock, um simulador de *blockchain*, e o NS-3, um simulador de redes usado para simular diferentes protocolos de rede, incluindo uma rede de *blockchain*. Cada simulador possui suas próprias características e o processo de simulação é descrito. Posteriormente, é realizado um estudo comparativo entre esses dois simuladores, a fim de analisar a precisão e eficiência na simulação de *blockchains*.

É utilizado o mesmo cenário para ambos os simuladores NS3 e Simblock. Após o experimento de simulação, os dados coletados foram analisados e interpretados usando o software Sysstat. Dentre as métricas utilizadas para a análise de desempenho do experimento, foram considerados a porcentagem de uso de memória, a porcentagem de utilização da CPU e o tempo de construção em segundos.

Além de ser analisado o desempenho também é feita uma comparação de seis categorias entre os simuladores gerando uma tabela de resultados para serem descritos logo após, no final é apresentado que apesar dos simuladores serem capazes de similares redes com vários nós os resultados demonstram que cada simulador tem pontos negativos e positivos como o SimBlock tem um tempo de simulação mais curto e o NS-3 um uso mais eficiente de recursos.

A principal diferença do trabalho de (HANGGORO; SARI, 2021) para esse será a mudança de cenário, configuração e simuladores que serão utilizados na análise de desempenho, utilizando somente o simulador SimBlock que será comparado com os simuladores BlockSim e Bitoin-Simulator ambos não são usados neste trabalho.

## 3.3 Evaluating Quality-of-Service in Blockchains Using Modelling and Simulation Tools

Em Lone e Mir (2020), são mencionadas várias características da tecnologia *blockchain* que despertaram interesse em pesquisas na área. Além disso, são abordados os diferentes tipos de aplicações descentralizadas que estão ganhando força com o crescimento do *blockchain*. Nesse contexto, torna-se essencial o desenvolvimento de ferramentas para avaliação de desempenho de sistemas que utilizam essa tecnologia.

Entre essas ferramentas, os simuladores de redes *blockchain* desempenham um papel fundamental. Eles demonstram os benefícios que podem ser alcançados por meio da simulação, auxiliando no processo de construção e implementação de sistemas baseados em *blockchain*.

Lone e Mir (2020) também destaca a importância de utilizar métricas adequadas durante o processo de simulação, a fim de validar os experimentos realizados em redes *blockchain*.

Lone e Mir (2020) cita e descreve algumas métricas de qualidade de desempenho responsáveis por auxiliar no processo de análise de desempenho de um sistema *blockchain*. Após isso é apresentado dois simuladores de *blockchain*, o VIBES e o BlockSim, descrevendo seus pontos fortes e algumas limitações para logo depois realizar uma análise de desempenho desses dois simuladores e estudar as métricas de qualidade do serviço do sistema.

O VIBES foi utilizado para simular uma rede de *Bitcoin* e o BlockSim uma rede de Ethereum, através dessas simulações algumas métricas foram aplicadas de acordo com cada simulador para o BlockSim podem ser citadas: Limite de Gás por Transação, Limite de Gás por Bloco, Transações por Bloco, Tamanho do Bloco, Tempo Médio de Propagação do Bloco.

Já para o VIBES foram: Número de Nós, Número de Vizinhos, Período de Bloco, Latência do Tamanho da Transação, Atraso na Propagação da Transação, Limite de Tamanho do Bloco, Largura de Banda da Rede. Estas métricas foram adaptadas para considerar os resultados para cada caso de uso.

As diferenças desse trabalho com Lone e Mir (2020) será a mudança de cenário em que será realizado a simulação, a mudança dos simuladores de *blockchain*, serão utilizados três simuladores para o experimento, SimBlock, BlockSim e VIBES. A mudança de métricas que possam atender o experimento dos três simuladores e gerarem os dados para serem analisados posteriormente na avaliação de desempenho.

# 4 METODOLOGIA

Neste capítulo, será apresentada a metodologia que será aplicada neste trabalho

## 4.1 Elaboração do cenário

O primeiro passo a ser realizado neste projeto será a elaboração de um cenário para ser usado pelos simuladores de *blockchain* definidos no experimento: BlockSim, SimBlock e Bitcoin-Simulator.

A escolha desses três simuladores de rede *blockchain* foi baseada em diversos motivos. Primeiramente, eles são simuladores de código aberto, cujo código é aberto e livre para acesso, modificação e distribuição dos usuários. O código-fonte se encontra de fácil acesso no repositório oficial de cada projeto localizado no GitHub. Outro fator importante foi o fato de muitos dos simuladores *blockchain* disponíveis atualmente serem uma expansão de um desses três simuladores escolhidos, mostrando assim seu impacto na área de simulação de *blockchain*.

O cenário elaborado será uma rede de *blockchain* que através dos nós consegue interligar todas as capitais do Brasil. Serão realizados testes de carga com acréscimo de número de nós para observar como o desempenho de cada simulador é afetado. Após a validação do cenário proposto será realizado um levantamento de métricas necessárias para serem avaliadas no processo de simulação de cada simulador.

## 4.2 Definição de Métricas e Modelos

Abaixo as métricas que serão utilizadas para a avaliação de desempenho dos simuladores BlockSim, SimBlock e Bitcoin-Simulator.

- Duração da Simulação
- Utilização da CPU.
- Utilização da Memória.
- Configuração e Flexibilidade do Simulador.
- Diversidade de Dados do Simulador.

Os modelos são configurações e arquivos JSON usados pelos simuladores para criar a simulação da rede *blockchain*. Cada simulador possui um formato de modelo diferente, o que significa que é necessário definir um modelo de entrada de dados específico para cada um, os modelos receberão entradas que se adéquem ao cenário proposto e a documentação de

cada simulador, a fim de obter a melhor forma de gerar modelos que impacte positivamente no desempenho das simulações.

Algumas dessas entradas possuem valores idênticos para todos os modelos, buscando uma aproximação máxima entre os simuladores para permitir avaliações consistentes. A seguir estão alguns parâmetros e entradas que devem ser aplicadas a todos os modelos.

- Tempo de simulação: 24 horas.
- Tamanho do bloco: 3 MB.
- Intervalo entre blocos: 10 minutos.
- Número de nós: 27, 54, 108, 216, 432, 864, 1728, 3456
- Nós mineradores: 8.

Os valores dos parâmetros Tempo de simulação, Tamanho do bloco e Intervalo entre blocos foram escolhidos com base nas pesquisas dos últimos estudos sobre a *blockchain* do *Bitcoin*. O número de nós é um valor crescente que será incrementado a cada novo teste de carga, o valor inicial 27 representa o número de capitais no Brasil sendo que cada nó representa uma capital, a cada novo teste de carga a quantidade de nós será dobrada até atingir o valor de 3456. A entrada Nós mineradores são a quantidade de nós que serão mineradores aplicada aos Simuladores BlockSim e Bitcoin-Simulator.

Será realizado o *download* do código-fonte de cada simulador a partir do repositório oficial no GitHub, através do código de simulador será aplicado as configurações estabelecidas no modelo de configuração, para posteriormente os simuladores serem executados e iniciarem a construção da rede *blockchain*. Após a criação da rede *blockchain* os dados gerados desta serão recolhidos para análise.

Para realizar as simulações, será utilizado um notebook Lenovo com processador Core i5, 8 GB de RAM e 240 GB de SSD. O sistema operacional utilizado para a implementação, execução do simulador e posterior avaliação de desempenho, será o Linux, mais especificamente a distribuição Ubuntu 20.04. O Ubuntu foi escolhido por ser um sistema de código aberto que permite a instalação de ferramentas que podem facilitar o processo de implementação e execução das simulações.

## 4.3 Implementação de simulação usando o simulador Bitcoin-Simulator

Nesta etapa, será implementada a simulação de uma rede *blockchain* usando o simulador Bitcoin-Simulator. O modelo que será utilizado para este experimento é composto

pela classe bitcoin-topology-helper.cc e bitcoin.cc.

Essas classes devem ser modificadas de acordo com o experimento, a classe bitcoin-topology-helper.cc é responsável por inserir os dados sobre latência, download e upload das regiões da rede, além da quantidade de nós mineradores que será 8. A classe bitcoin.cc é importante para modelar parte da estrutura de dados do *Bitcoin* na aplicação como por exemplo as regiões geográficas, tipo de minerador,tipo de bloco de transmissão.

Os outros parâmetros de entrada da simulação podem ser inseridos de forma dinâmica na hora da construção da rede. Abaixo alguns parâmetros que serão modificados dinamicamente nessa simulação na construção da rede.

- nodes: 27, números de nós que representam as capitais do Brasil.
- blockSize: 3000000, representa o tamanho do bloco da rede.
- noBlocks: 144, número de blocos gerados.
- blockIntervalMinutes: 10, intervalo médio em minutos para geração de bloco.

Com essas configurações em vigor, a simulação estará pronta para representar de forma precisa a dinâmica de uma rede *blockchain*.

## 4.4 Implementação de simulação usando o simulador BlockSim

Nesta etapa, será implementada a simulação de uma rede *blockchain* usando o simulador BlockSim. O modelo utilizado para este experimento é composto pelas classes main, config, latency, delays, throughput-sent e throughput-received.

Na classe main, algumas propriedades importantes que devem ser definidas são:

- duration = 86.400, que é o tempo total de simulação em segundos.
- miners = {"Distrito Federal", "São Paulo", "Rio de Janeiro", "Rio Grande do Sul", "Ceará", "Pernambuco", "Bahia", "Amazonas"}, classificação de nós mineradores.
- non_miners = {"Acre", "Alagoas", "Amapá", "Espírito Santo", "Goiás", "Maranhão", "Mato Grosso", "Mato Grosso do Sul", "Minas Gerais", "Pará", "Paraíba", "Paraná", "Piauí", "Rio Grande do Norte", "Rondônia", "Roraima", "Santa Catarina", "Sergipe", "Tocantins"}, classificação de nós não mineradores

A classificação dos estados em mineradores e não mineradores foi definida com base em sua importância simbólica e econômica, bem como em sua localização.

Durante a simulação, serão criados 4 lotes de transações, com cada lote contendo 3 transações. Esses lotes serão enviados a cada 15 segundos.

Dentro da classe config, alguns pontos são essenciais devem ser configurados.

- blockchain = "bitcoin", o tipo de *blockchain* utilizado na simulação.
- block_size_limit_mb = 3, tamanho da *blockchain*.
- locations = ["Acre", "Alagoas", "Amapá", "Amazonas", "Bahia", "Ceará", "Distrito Federal", "Espírito Santo", "Goiás", "Maranhão", "Mato Grosso", "Mato Grosso do Sul", "Minas Gerais", "Pará", "Paraíba", "Paraná", "Pernambuco", "Piauí", "Rio de Janeiro", "Rio Grande do Norte", "Rio Grande do Sul", "Rondônia", "Roraima", "Santa Catarina", "São Paulo", "Sergipe", "Tocantins"], configuração de localização de todos os estados do brasil.

É necessário configurar os arquivos latency, delays, throughput-sent e throughput-received, definindo valores específicos para cada estado, bem como os valores que ocorrerão quando esses estados interagirem na rede. Com essas configurações em vigor, a simulação estará pronta para representar de forma precisa a dinâmica de uma rede *blockchain*, com distinção entre mineradores e não mineradores, taxas variáveis e fluxos constantes de transações.

## 4.5 Implementação de simulação usando o simulador SimBlock

Nesta etapa, será implementada a simulação de uma rede *blockchain* usando o simulador Simblock. O modelo utilizado para este experimento é composto pelas classes NetworkConfiguration e SimulationConfiguration.

A classe NetworkConfiguration é responsável por configurar e definir parâmetros relacionados à rede da simulação, as propriedades dessa classe que devem ser configuradas para a realização do experimento são:

- regionList = ["Acre", "Alagoas", "Amapá", "Amazonas", "Bahia", "Ceará", "Distrito Federal", "Espírito Santo", "Goiás", "Maranhão", "Mato Grosso", "Mato Grosso do Sul", "Minas Gerais", "Pará", "Paraíba", "Paraná", "Pernambuco", "Piauí", "Rio de Janeiro", "Rio Grande do Norte", "Rio Grande do Sul", "Rondônia", "Roraima", "Santa Catarina", "São Paulo", "Sergipe", "Tocantins"], configuração de localização de todos os estados do brasil.

Outros parâmetros como LATENCY, DOWNLOADBANDWIDTH, UPLOADBANDWIDTH e REGIONDISTRIBUTION, receberão valores para cada nó que irá representar na rede com base nos valores informados em Region List, no caso cada propriedade dessas receberá 27 valores baseados em cada estado que representa, o parâmetro REGIONDISTRIBUTION

além de receber um valor com o estado que representa irá utilizar a distribuição do *Bitcoin*.

A classe SimulationConfiguration é responsável por configurar alguns comportamentos que serão adotados na simulação. Entre algumas propriedades que devem ser configuradas para o sistema são destaque:

- NUM_OF_NODES = 27, que representa o número de nós na simulação, cada estado e o Distrito Federal representam um nó.
- AVERAGE_MINING_POWER = 190539325, a potência média de mineração.
- STDEV_OF_MINING_POWER = 50000000, intervalo de mineração.
- END_BLOCK_HEIGHT = 144, a altura do bloco em que a simulação.
- BLOCK_SIZE = 3000000 B, tamanho médio do bloco
- COMPACT_BLOCK_SIZE = 53 * 1024, tamanho dos blocos compactos

Com essas configurações em vigor, a simulação estará pronta para representar de forma precisa a dinâmica de uma rede *blockchain*.

## 4.6 Coleta de dados e Análise de desempenho

Nesse passo será realizada a coleta de dados de cada experimento, levando em consideração o modelo de saída de dados de cada simulador. Após a coleta será realizada uma filtragem dos dados relevantes gerados para o experimento, a filtragem ocorrerá através de *scripts* de programação que utilizam a biblioteca psutil da linguagem Python, construídos para auxiliar nesse projeto, os valores dos dados de desempenho serão recolhidos de acordo com o maior valor demostrado pelos *scripts* ao executar os experimentos.

Após a filtragem dos dados relevantes será construído gráficos com os resultados obtidos e o processo de análise de desempenho terá início. Os dados serão analisados e as métricas definidas no primeiro passo serão aplicadas, logo após as tabelas, gráficos e resultados individuais gerados para cada simulador serão usados para comparar resultados.

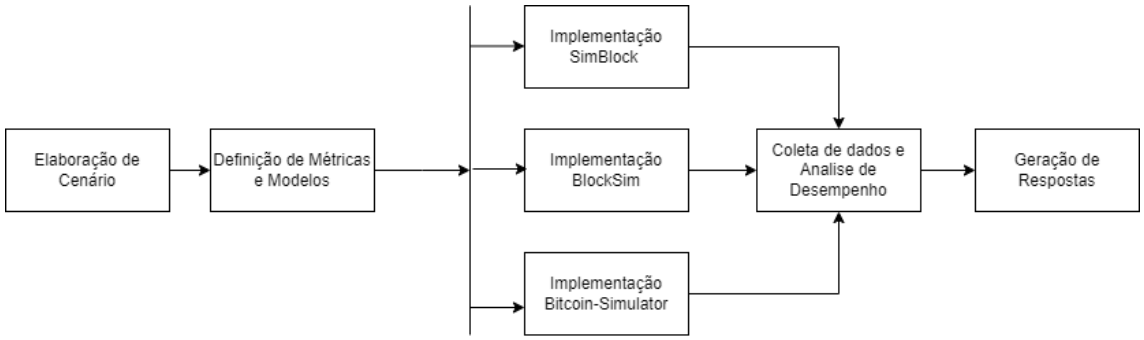
## 4.7 Geração de Respostas

Com base nos resultados apresentados nas tabelas após o processo de análise, será possível avaliar qual simulador se adaptou melhor ao cenário em questão. Além disso, será avaliado o impacto das métricas na avaliação de desempenho dos simuladores.

## 4.8 Diagrama de Atividades

A figura 5 apresenta a sequência das atividades planejadas para a execução deta pesquisa.

Figura 5 – Diagrama de Atividades




Fonte: Autor

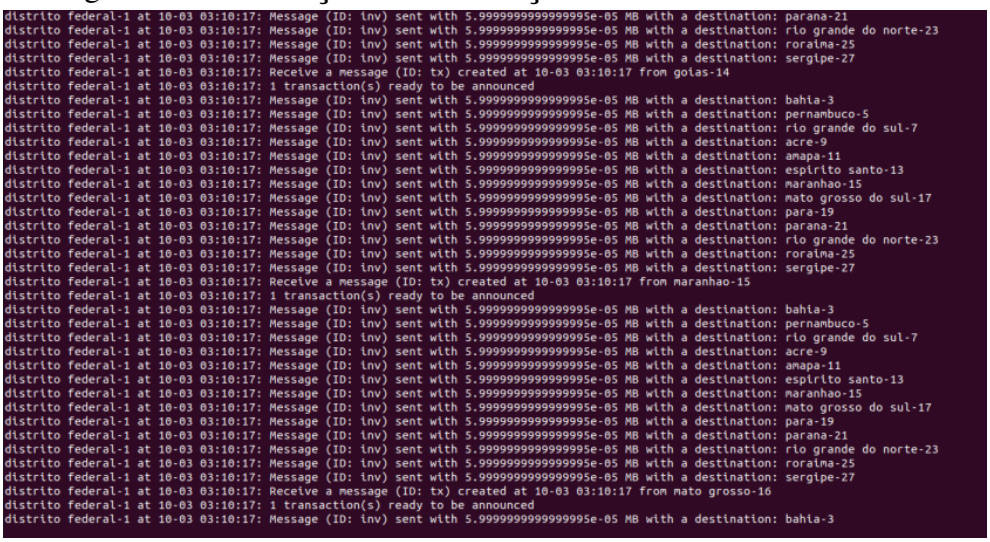
# 5 RESULTADOS

Neste capítulo, será apresentado os resultados obtidos das simulações geradas pelos simuladores *blockchain*: BlockSim, Biticoin-Simulator e SimBlock.

## 5.1 Resultados BlockSim

Após o desenvolvimento do modelo de configuração do simulador BlockSim, foram iniciados os experimentos para a implementação da simulação da rede *blockchain*. Através da inserção dos comandos no terminal do Linux é possível acompanhar diretamente da tela do terminal a construção da rede, as comunicações e transferências entre os nós. Por meio da figura 6 é possível ver a construção da rede *blockchain* no terminal do Linux, nela, Distrito Federal troca mensagens com outros estados além de anunciar novas transações.

Figura 6 – Construção de simulação no Linux do BlockSim



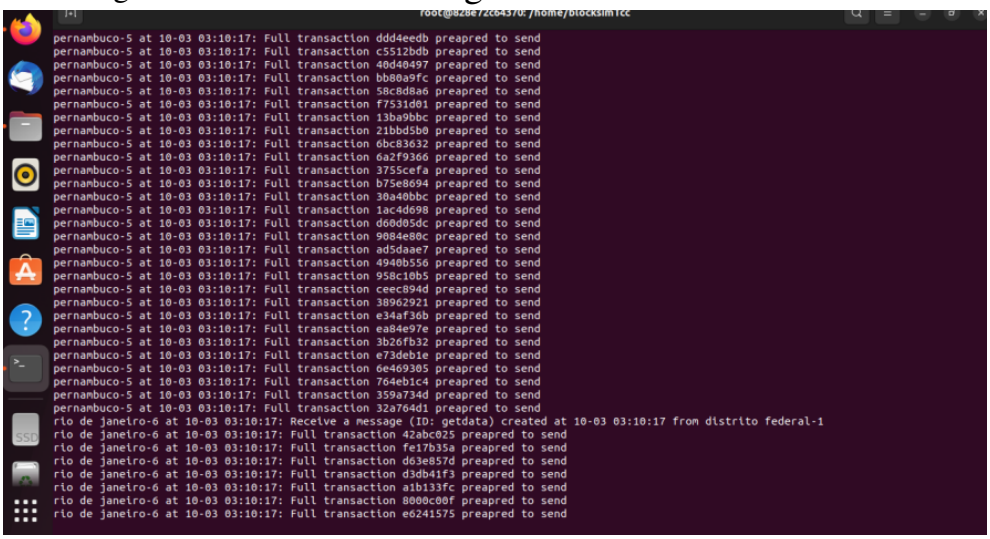

Fonte: Autor

Já a figura 7 mostra os estados Pernambuco e Rio de Janeiro preparando transações para serem enviadas para os outros estados, a hora na simulação que essas transações estão prontas, o número de hash para cada uma, além de mostrar uma troca de mensagens do Rio de Janeiro com o Distrito Federal.

Ao finalizar a simulação foi gerado um arquivo JSON localizado no diretório output denominado "report", que contém os dados gerados pelo simulador de forma organizada.

A figura 8 mostra parte do arquivo report com os dados em formato JSON gerados, após o experimento da construção da rede *blockchain* ser finalizado.

Figura 7 – Troca de mensagens de nós com o BlockSim




Fonte: Autor

Figura 8 – Arquivo output em JSON gerados da simulação BlockSim


Fonte: Autor

Passado a etapa de simulação da rede de *blockchain* foram colhidos resultados dos testes de cargas aplicados ao simulador BlockSim. A tabela 1 contém o tempo em milissegundos em que cada simulador levou para criar o cenário com o número de nós aplicados. A tabela 2 contém a porcentagem de CPU utilizada ao realizar o experimento do simulador para construir a rede *blockchain*. Através da tabela 3 é possível conferir a porcentagem de memória gasta pelo computador para conseguir replicar o experimento da rede *blockchain*. As entradas 1728 e 3456 estão zeradas já que ao aplicar a simulação com essa configuração de entrada o BlockSim teve seu processo interrompido ao exigir mais memória do que o disponibilizado pela máquina do experimento.

Tabela 1 – Tempo de Simulação BlockSim

| Nós | 27 | 54 | 108 | 216 | 432 | 864 | 1728 | 3456 |
|---|---|---|---|---|---|---|---|---|
| Milissegundos | 3.338 | 8.598 | 24.618 | 106.47 | 396.316 | 1034.984 | 0 | 0 |

Fonte: Autor

Tabela 2 – Desempenho CPU BlockSim

| Nós | 27 | 54 | 108 | 216 | 432 | 864 | 1728 | 3456 |
|---|---|---|---|---|---|---|---|---|
| CPU % | 3.2 | 22.7 | 32.4 | 34.9 | 51.6 | 61.5 | 0 | 0 |

Fonte: Autor

Tabela 3 – Desempenho Memória BlockSim

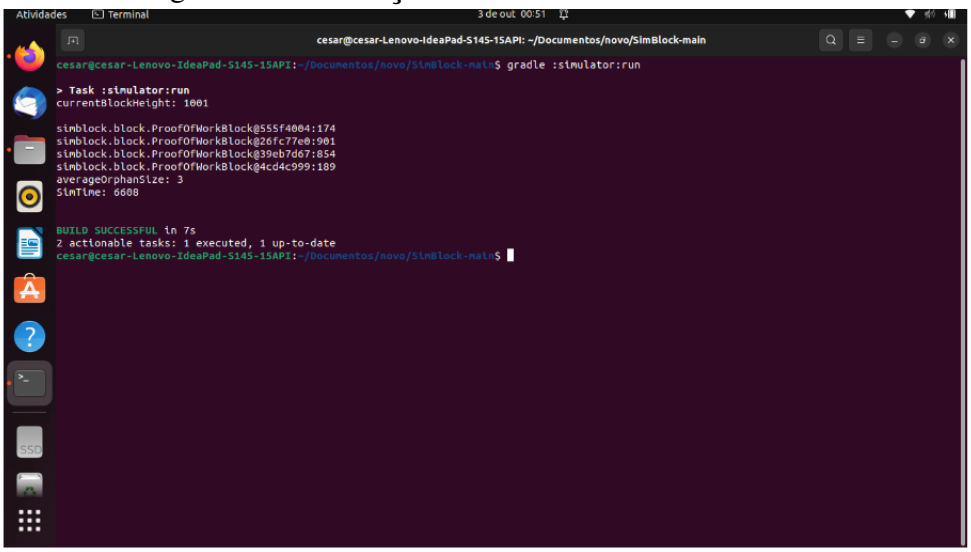
| Nós | 27 | 54 | 108 | 216 | 432 | 864 | 1728 | 3456 |
|---|---|---|---|---|---|---|---|---|
| Memória % | 32.2 | 35.7 | 26.9 | 31.4 | 47.2 | 99.0 | 0 | 0 |

Fonte: Autor

## 5.2 Resultados SimBlock

A partir da criação e configuração do modelo de desenvolvimento do SimBlock, a simulação de uma rede de *blockchain* é iniciada por meio de comandos no terminal do Linux, na figura 9 é possível visualizar o exemplo da simulação de uma rede *blockchain* sendo criada, o terminal mostra alguns dados após o término da simulação.

Figura 9 – Simulação do SimBlock no terminal

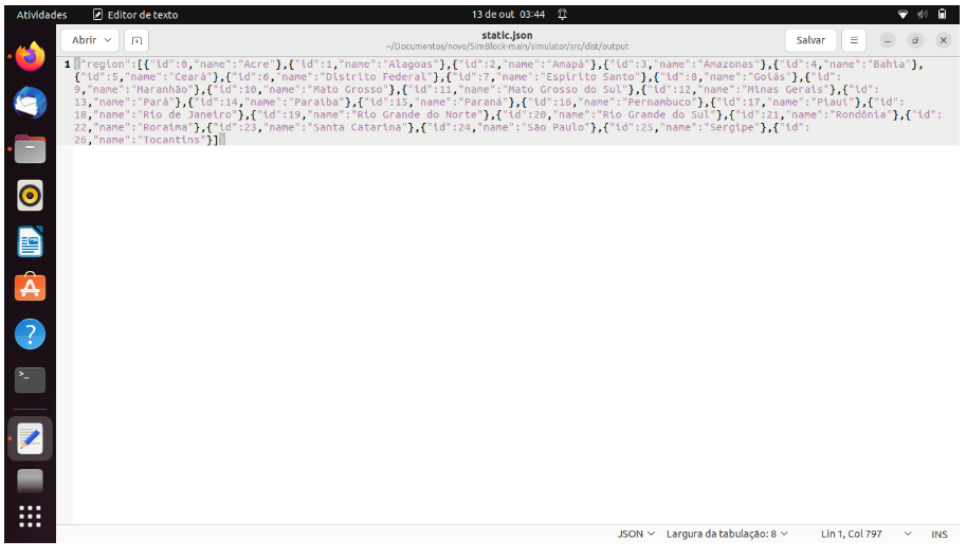



Fonte: Autor

Após a criação da simulação o SimBlock irá gerar arquivos no diretório output contendo dados que são relevantes sobre as características da rede *blockchain* criada, entre esses arquivos existem dois que se destacam e serão usadas para a análise da rede, esses arquivos são criados em formato JSON facilitando assim a futura análise dos dados, eles são chamados de static e output.

O arquivo static representa os nós que serão utilizados no desenvolvimento da simulação *blockchain*, neste caso será um arquivo com todas os estados brasileiros e Distrito Federal como é possível ver na figura 10.

Figura 10 – Arquivo static.json gerado da simulação do SimBlock




Fonte: Autor

O arquivo chamado output contém os dados como transmissões, transações, criação de nós e blocos que foram gerados após a construção da simulação da rede *blockchain*, na figura 11 é possível ver parte desse arquivo.

Passado a etapa de simulação da rede de *blockchain* foram colhidos resultados dos testes de cargas aplicados ao simulador SimBlock utilizando o exemplo da rede *blockchain* que passa pelas capitais. A tabela 4 contém o tempo em milissegundos em que cada simulador levou para criar o cenário com o número de nós aplicados. Através da tabela 5 é possível conferir a porcentagem de CPU utilizada ao realizar o experimento do simulador para construir a rede *blockchain*. A tabela 6 mostra a porcentagem de memória gasta pelo computador para conseguir replicar o experimento da rede *blockchain*.

Figura 11 – Arquivo static.json gerado da simulação do SimBlock


Fonte: Autor

Tabela 4 – Tempo de Simulação SimBlock

| Nós | 27 | 54 | 108 | 216 | 432 | 864 | 1728 | 3456 |
|---|---|---|---|---|---|---|---|---|
| Milissegundos | 2.995 | 4.814 | 8.119 | 13.675 | 22.802 | 55.088 | 94.65 | 214.411 |

Fonte: Autor

Tabela 5 – Desempenho CPU SimBlock

| Nós | 27 | 54 | 108 | 216 | 432 | 864 | 1728 | 3456 |
|---|---|---|---|---|---|---|---|---|
| CPU % | 9.6 | 29.8 | 23.1 | 24.6 | 26.8 | 28.8 | 24.0 | 29.2 |

Fonte: Autor

Tabela 6 – Desempenho Memória SimBlock

| Nós | 27 | 54 | 108 | 216 | 432 | 864 | 1728 | 3456 |
|---|---|---|---|---|---|---|---|---|
| Memória % | 31.9 | 33.7 | 26.9 | 33.1 | 33.8 | 34.4 | 34.4 | 36.5 |

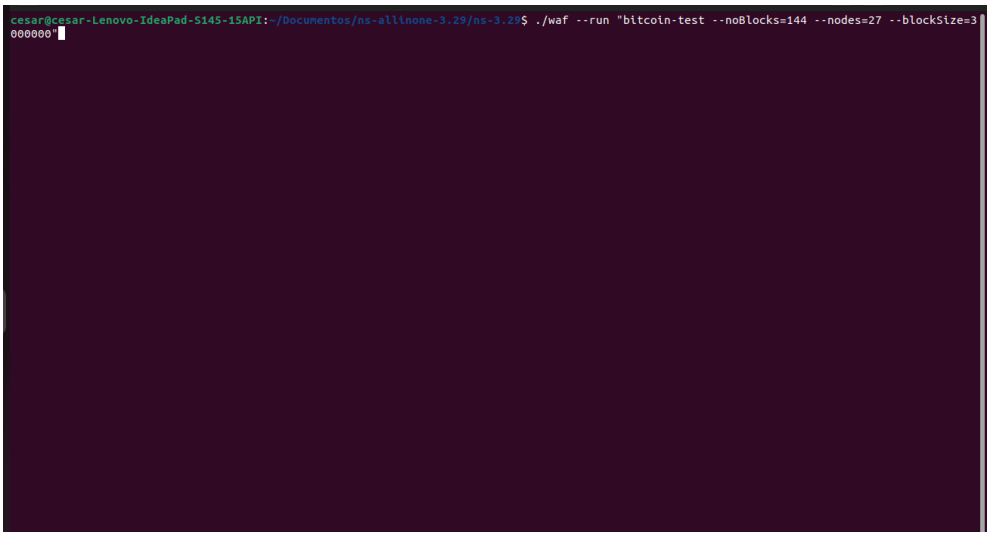
Fonte: Autor

## 5.3 Resultados Bitcoin-Simulator

Após o desenvolvimento do modelo de configuração de simulação do Bitcoin-Simulator para a construção da simulação da rede *blockchain*, é necessário iniciar a aplicação utilizando comandos através do terminal no Linux, neste exemplo alguns parâmetros também serão configurados através do terminal de forma dinâmica, os valores padrões desses parâmetros podem ser modificados antes do sistema ser iniciado configurando o modelo implementado, mas a abordagem dinâmica é a melhor opção devido às vantagens de modificar o cenário sem a necessidade de alterar o modelo completamente.

Na figura 12 é mostrado os valores dos parâmetros que foram definidos para que o experimento possa ser realizado da forma ideal, esses valores são os mesmos descritos na criação do modelo, eles podem ser modificados dependendo da necessidade do experimento, mas os valores descritos na criação do modelo irão ser preservados. Depois da descrição das características que o simulador necessita para criar a rede *blockchain* a simulação poderá ser iniciada.

Figura 12 – Tela de inserção de parâmetros do Bitcoin-Simulator

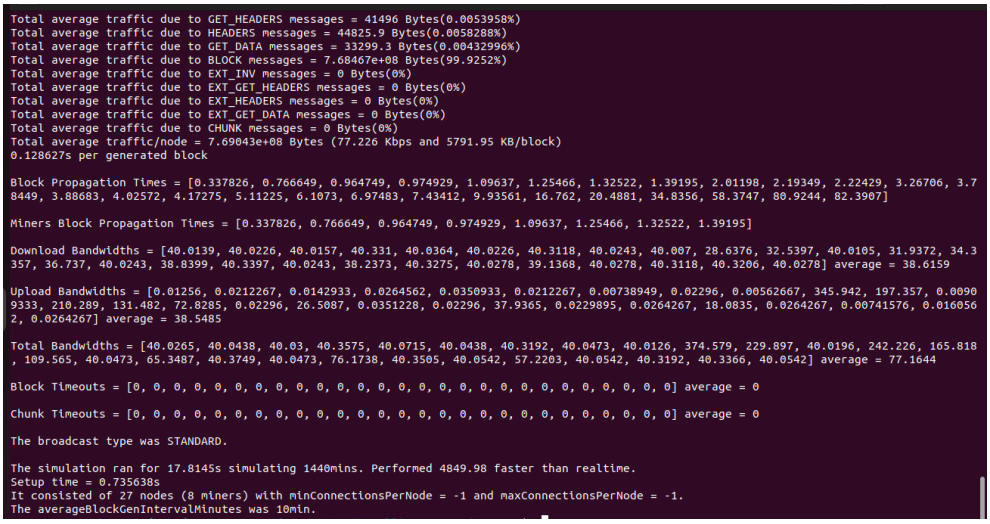



Fonte: Autor

Com o processo de simulação iniciado o simulador irá gerar resultados que podem ser analisados através do terminal, a figura 13 mostra alguns dos valores que são criados e podem ser analisados sobre a rede de *blockchain*.

Passado a etapa de simulação da rede de *blockchain* foram colhidos resultados dos testes de cargas aplicados ao simulador Bitcoin-Simulator utilizando o exemplo da rede

Figura 13 – Tela de resultados da simulação do Bitcoin-Simulator




Fonte: Autor

*blockchain* que passa pelas capitais. A tabela 7 contém o tempo em milissegundos em que cada simulador levou para criar o cenário com o número de nós aplicados. Através da tabela 8 contém a porcentagem de CPU utilizada ao realizar o experimento do simulador para construir a rede *blockchain*. A tabela 9 mostra a porcentagem de memória gasta pelo computador para conseguir replicar o experimento da rede *blockchain*.

Tabela 7 – Tempo de Simulação Bitcoin-Simulator

| Nós | 27 | 54 | 108 | 216 | 432 | 864 | 1728 | 3456 |
|---|---|---|---|---|---|---|---|---|
| Milissegundos | 13.14 | 25.09 | 85.82 | 249.58 | 599.01 | 1390.40 | 2278.56 | 3392.24 |

Fonte: Autor

Tabela 8 – Desempenho CPU Bitcoin-Simulator

| Nós | 27 | 54 | 108 | 216 | 432 | 864 | 1728 | 3456 |
|---|---|---|---|---|---|---|---|---|
| CPU % | 13.8 | 13.0 | 14.1 | 24.6 | 26.8 | 14.4 | 15.4 | 17.9 |

Fonte: Autor

Tabela 9 – Desempenho Memória Bitcoin-Simulator

| Nós | 27 | 54 | 108 | 216 | 432 | 864 | 1728 | 3456 |
|---|---|---|---|---|---|---|---|---|
| Memória % | 36.8 | 25.9 | 26.2 | 26.8 | 27.7 | 29.8 | 32.5 | 37.6 |

Fonte: Autor

## 5.4 Análise de Desempenho

Os dados coletados dos simuladores após os testes de carga em uma rede *blockchain* que passa por todas as capitais do Brasil serão utilizados para realizar uma análise de desempenho dos simuladores, com foco nas métricas: Duração da Simulação, Utilização da CPU, Utilização da Memória, Configuração e Flexibilidade do Simulador, Diversidade de Dados do Simulador.

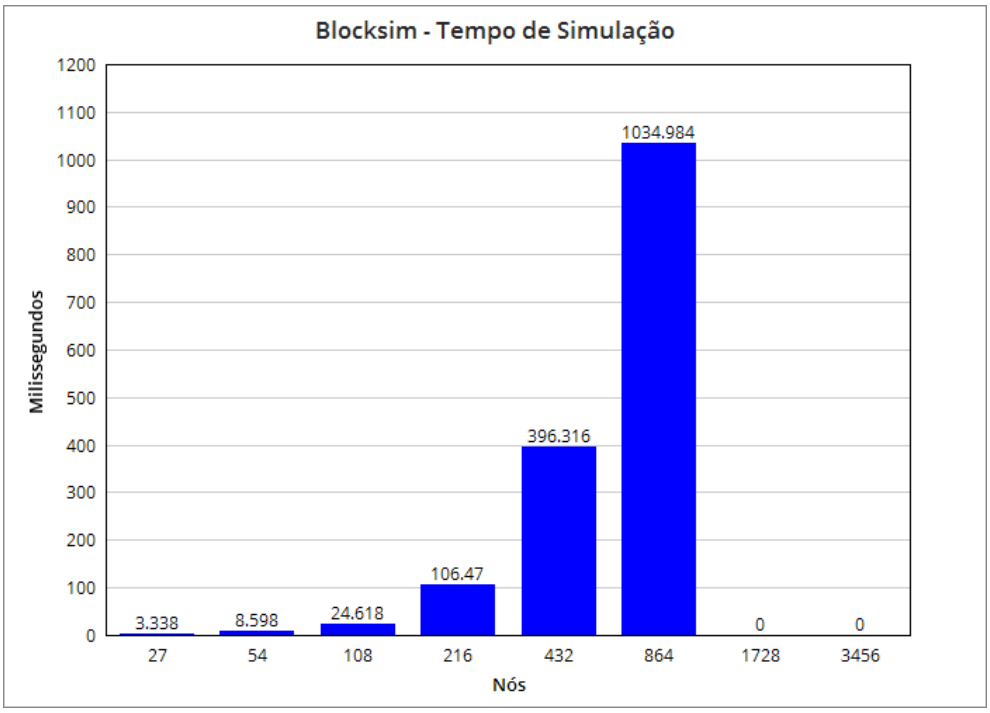
### *5.4.1 Duração da Simulação*

Essa métrica tem o objetivo de avaliar a duração de cada simulação ao executar experimentos com cargas diferentes. Com base no tempo de desempenho de cada simulação, após a implementação dos testes de carga, é possível gerar os seguintes gráficos.

A figura 14 mostra o tempo de duração que a simulação do BlockSim levou para ser concluída com cada carga de nós. É possível notar uma tendência no gráfico de maneira ascendente, com um acréscimo entre trés a quatro vezes no tempo de simulação toda vez que o valor do número de nós é dobrado em cada nova interação, a medida usada são milissegundos. Devido ao problema com esgotamento de memória da máquina, as duas ultimas interações com 1728 e 3456 nós não foram registradas, uma vez que seus processos foram interrompidos.

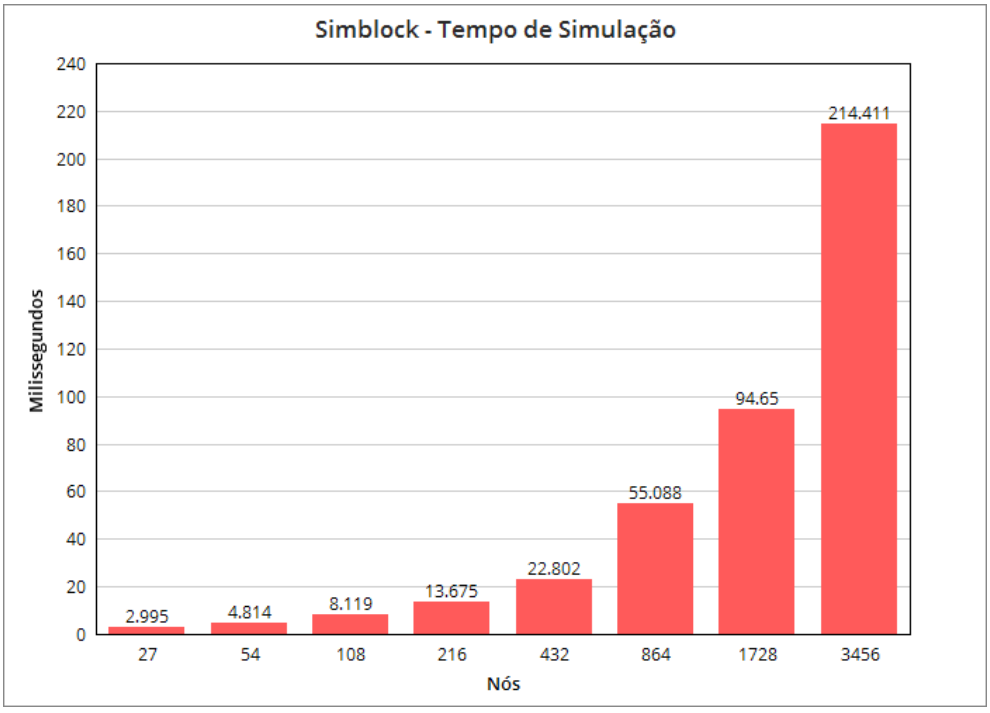
A figura 15 mostra o tempo de duração para a realização de cada simulação com cargas diferentes do SimBlock, é possível notar um gráfico de maneira ascendente com um aumento no tempo de simulação que fica variando a cada nova interação em que a carga de nós é dobrada.

Figura 14 – Duração de Simulação BlockSim




Fonte: Autor

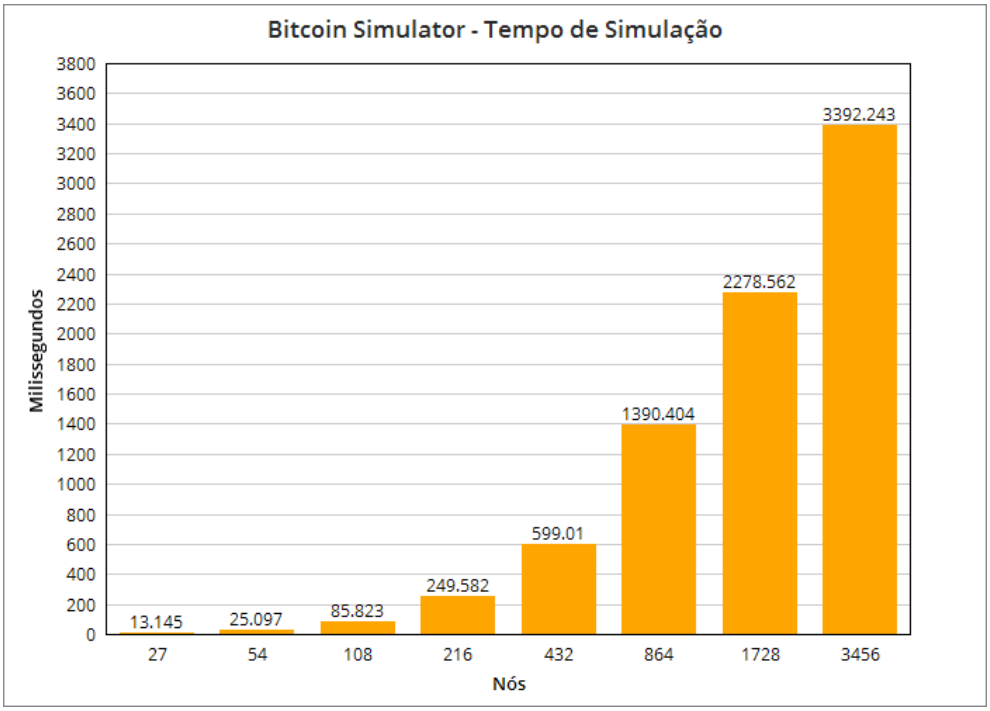
Figura 15 – Duração de Simulação SimBlock




Fonte: Autor

A figura 16 exibe o tempo necessário para realizar cada simulação com diferentes cargas no Bitcoin-Simulator. É possível notar um gráfico de tempo ascendente a cada nova simulação com uma carga de nós dobrada, este crescimento fica delimitado a um intervalo entre duas a três vezes acrescentado no tempo de simulação. Nas últimas cargas de nós existe uma queda na diferença do tempo de simulação dos resultados.

Figura 16 – Duração de Simulação Bitcoin-Simulator




Fonte: Autor

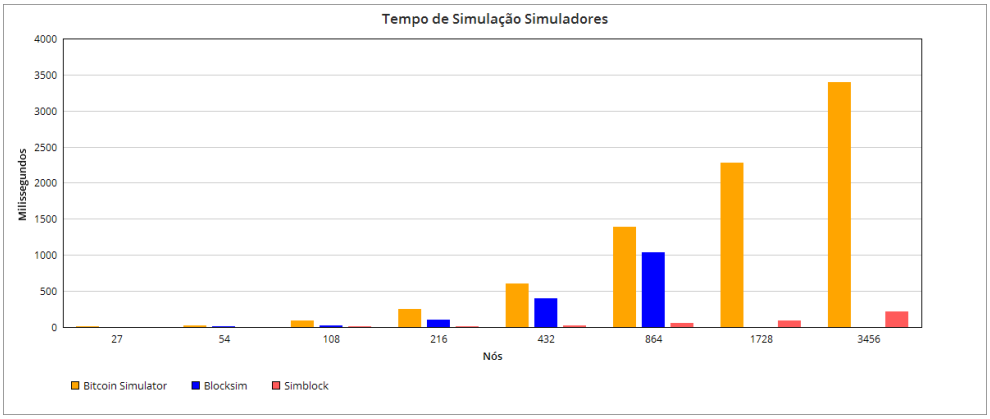
A figura 17 faz um comparativo entre o tempo de duração de simulação de todos os simuladores da rede *blockchain* utilizados nesse experimento: BlockSim, SimBlock e Bitcoin-Simulator. É possível perceber que todos os simuladores apresentam uma tendência de crescimento no tempo de simulação a cada nova interação em que a carga de nós é dobrada no experimento. O simulador SimBlock contém praticamente o menor tempo de simulação em todos os casos, na maioria dos testes de cargas o tempo de duração do SimBlock é bastante inferior gastando até menos da metade do tempo comparado aos outros simuladores com a mesma carga de nós.

Já o simulador BlockSim, que apesar de não conter dados nos dois últimos testes de cargas devido à limitação de *hardware* da memória que ocasionou a interrupção da simulação, sempre se mostrou com um tempo abaixo do Bitcoin-Simulator nos testes de carga.

Cada teste de carga de nós utilizando o Bitcoin-Simulator obteve tempo superior ao

ser comparado com os outros simuladores *blockchain*, inclusive é possível notar que a diferença do tempo de simulação para os outros simuladores aumenta a cada nova carga de nós.

Figura 17 – Duração de Simulação dos simuladores




Fonte: Autor

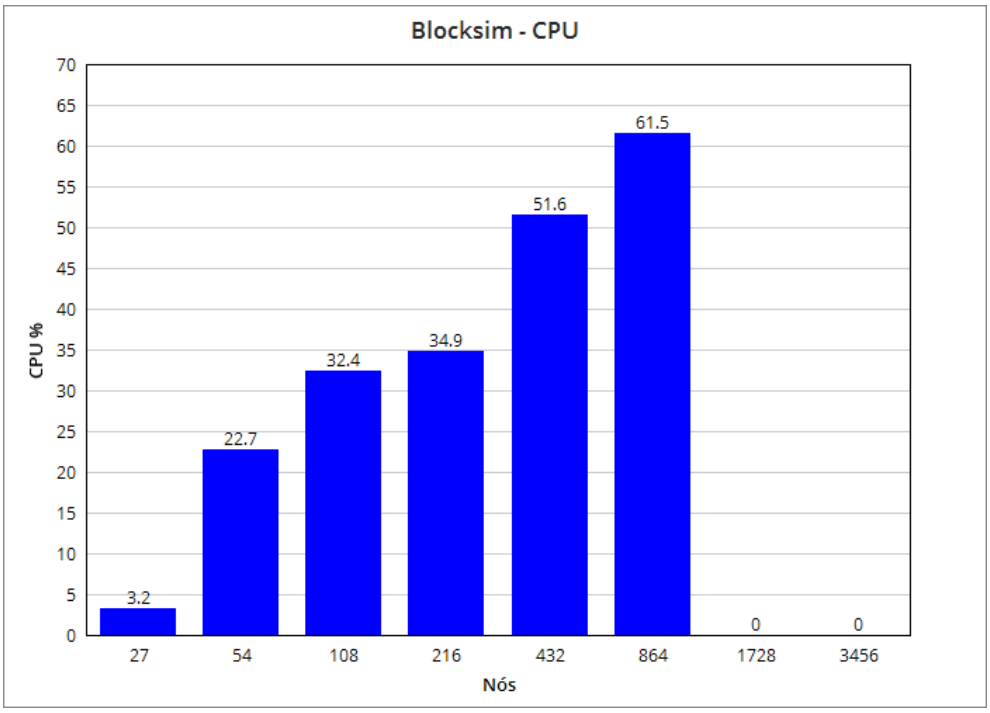
### *5.4.2 Utilização da CPU*

Esta métrica pretende analisar a porcentagem gasta de CPU pelo sistema ao utilizar os simuladores *blockchain*.

A figura 18 mostra a porcentagem de desempenho do processador exigida pelo sistema ao realizar os testes de cargas na simulação de rede do BlockSim, a cada incremento da carga de nós maior o aumento de porcentagem requerido para o processo, na primeira carga existe uma taxa baixa mostrando que o processador tem pouca exigência, a partir da segunda carga o processador teve um aumento que varia entre dois a dezessete por cento a cada vez que a carga é dobrada. Os valores nulos das últimas cargas 1728 e 3456 são reflexos de um erro de memória que evitou a continuação da simulação.

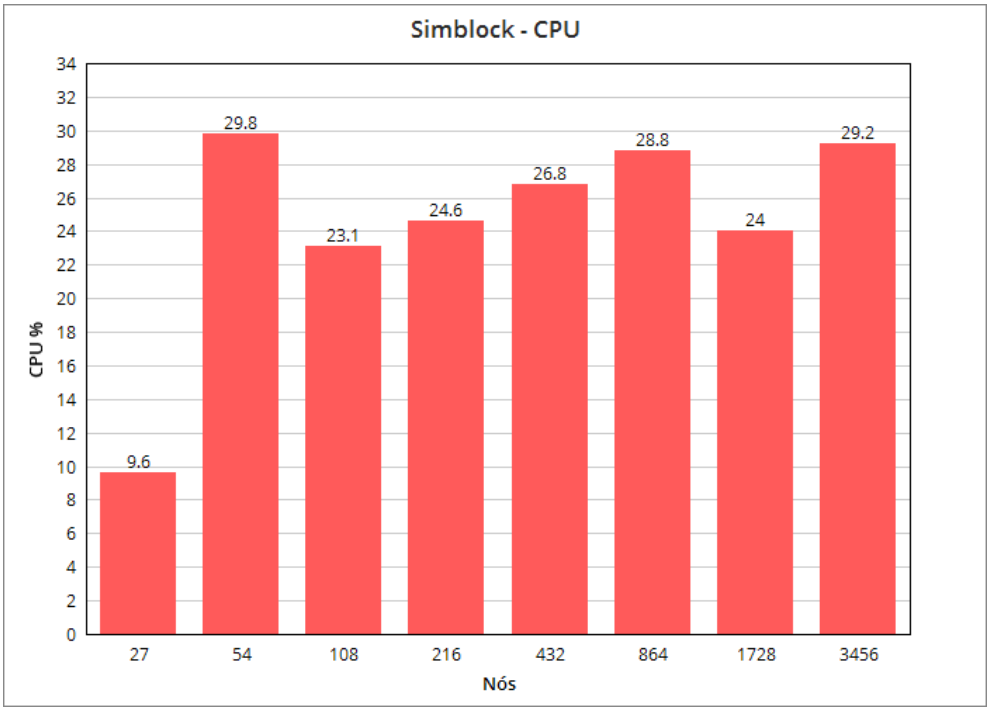
A figura 19 mostra a porcentagem de desempenho do processador exigida pelo sistema ao realizar os testes de cargas na simulação de rede do SimBlock. Existe uma alternância de desempenho da CPU na simulação no decorrer dos testes de cargas, todos os valores ficaram abaixo de trinta por cento, mostrando que o processador não usa sequer metade de sua capacidade de processamento para realizar as simulações. O acréscimo do número de cargas de nós não tem uma grande influência no desempenho da CPU, devido ao realizar a simulação os valores dos resultados são muito próximos e sofrem alternância.

Figura 18 – Desempenho CPU BlockSim




Fonte: Autor

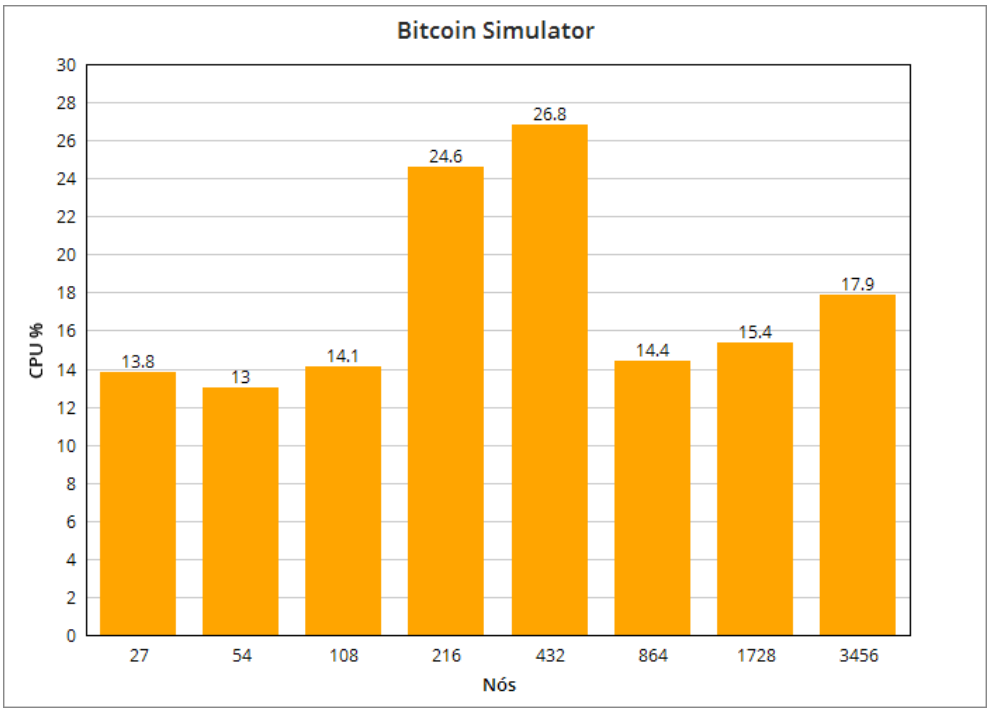
Figura 19 – Desempenho CPU Simblock




Fonte: Autor

A figura 20 mostra a porcentagem de desempenho do processador exigida pelo sistema ao realizar os testes de cargas na simulação de rede do Bitcoin-Simulator. Os valores de desempenho da CPU durante os testes de cargas da simulação apresentam valores com uma pequena variação que ficam abaixo de vinte por cento, dois valores ficaram acima da media geral, mas todos os valores são menores que trinta por cento. Devido a esses valores de desempenho, é possível notar que o processador não usa sequer metade de sua capacidade de processamento para realizar as simulações.

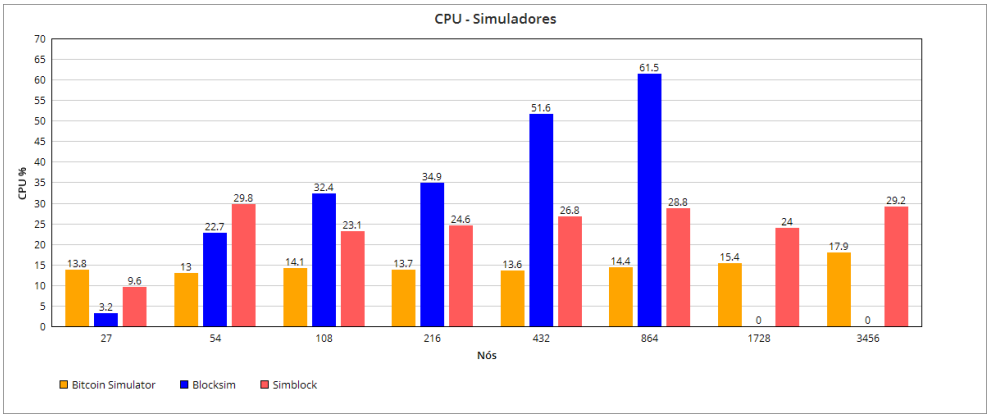
Figura 20 – Desempenho CPU Biticoin-Simulator




Fonte: Autor

A figura 21 faz um comparativo entre o desempenho de CPU de todos os simuladores de rede *blockchain* utilizados nesse experimento para construir a simulação. Apesar do BlockSim começar tendo o menor número de processamento gasto, após os dois primeiros testes de carga ele ultrapassou os outros dois simuladores e a partir do terceiro experimento começou a crescer deixando os outros simuladores para trás. Já o SimBlock foi o segundo simulador que teve as maiores porcentagens de desempenho da CPU, com exceção a primeira carga de nós o SimBlock obteve mais processamento de CPU da rede comparado ao Bitcoin-Simulator.

Figura 21 – Desempenho CPU Simuladores




Fonte: Autor

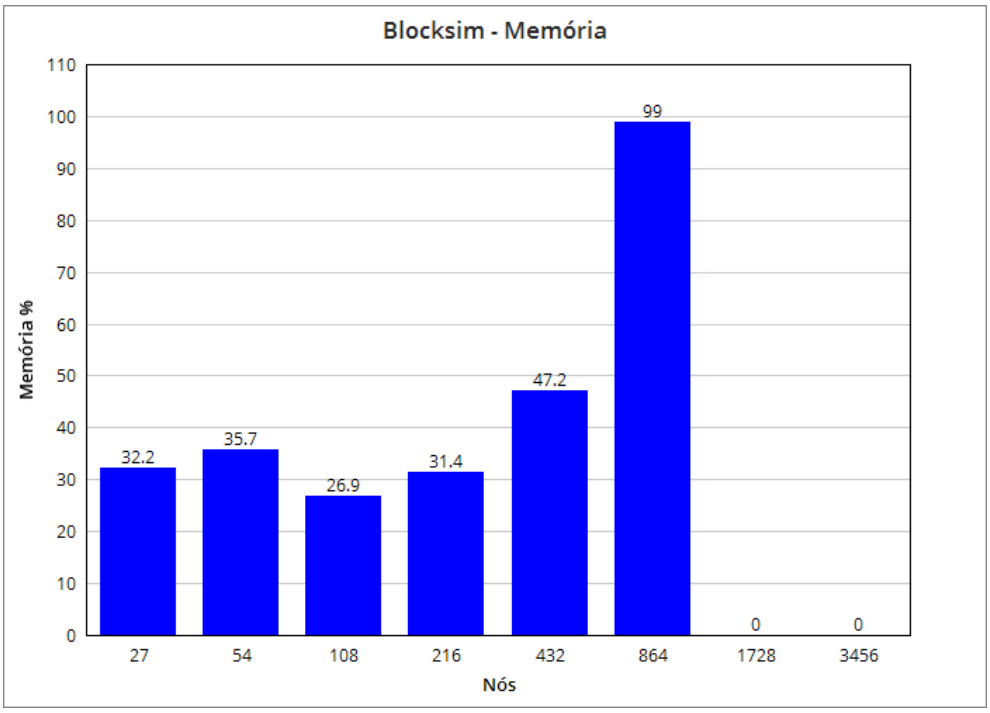
### *5.4.3 Utilização da Memória*

Esta métrica pretende analisar a porcentagem gasta de Memória pelo sistema ao utilizar os simuladores

A figura 22 mostra a desempenho da memória que foi utilizado pelo simulador BlockSim ao realizar os testes de carga da construção da rede. Até a carga de nós 432 existe uma variância de valores sobre a porcentagem de memória gasta na simulação, ao dobrar a próxima carga de nós o valor do uso de memória praticamente dobra também atingindo o seu pico máximo de noventa e nove por cento. Ao realizar outra dobra de carga a memória da máquina não suportou a quantidade exigida pelo simulador, causando assim o encerramento abrupto da simulação alegando erro por falta de memória para realizar o processo.

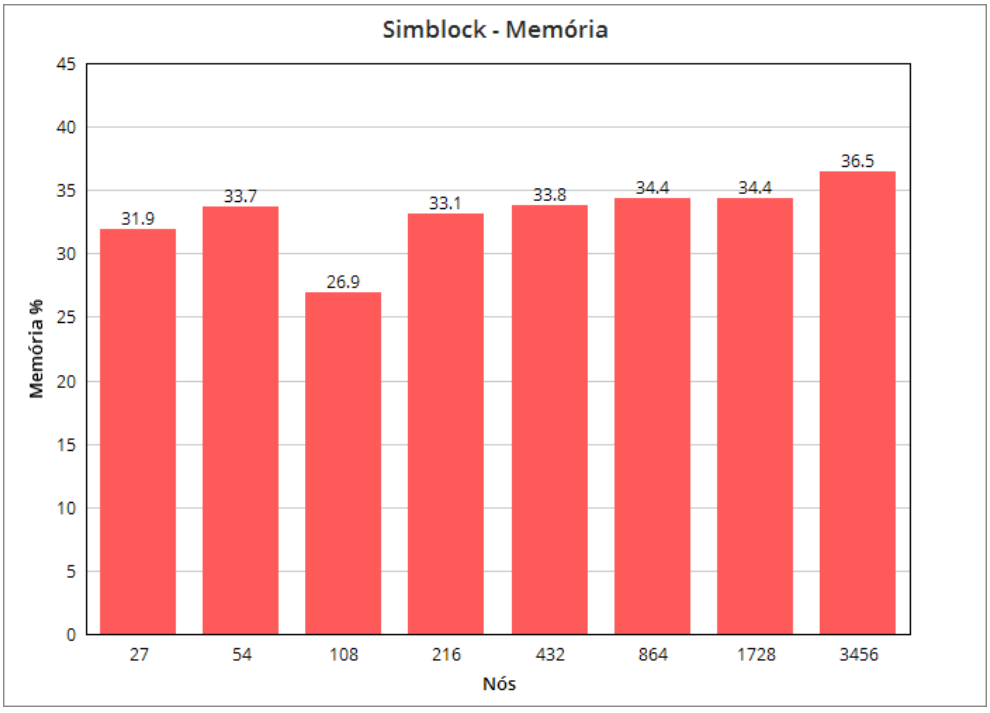
A figura 23 mostra a desempenho da memória que foi utilizado pelo simulador SimBlock ao realizar os testes de carga da construção da rede. Apesar de acontecer uma queda de desempenho na terceira carga de nós, é possível perceber uma tendência de crescimento no consumo de memória. Mesmo com a carga de nós dobrada a cada simulação de rede, o consumo de memória mostra ser estável com pouco crescimento. Não foi atingido nem metade da capacidade de memória oferecida pela máquina, já que o valor de memória da máquina utilizada pela máquina fica sempre abaixo de quarenta por cento da capacidade.

Figura 22 – Desempenho de Memória BlockSim




Fonte: Autor

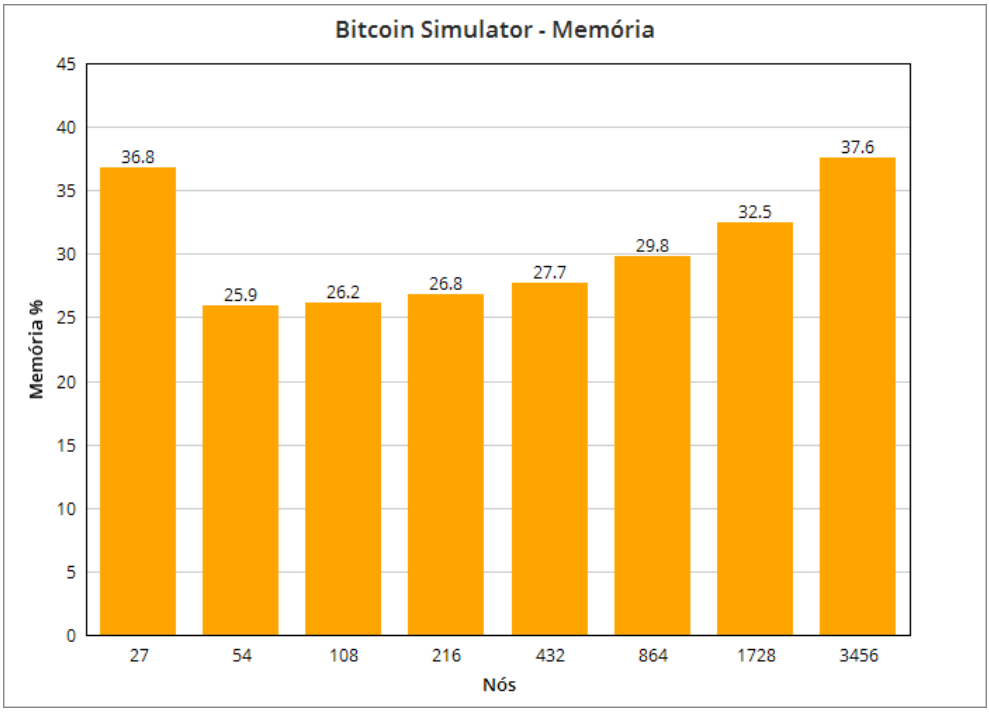
Figura 23 – Desempenho de Memória SimBlock




Fonte: Autor

A figura 24 mostra a desempenho de memória que foi utilizado pelo simulador Bitcoin-Simulator ao realizar os testes de carga da construção da rede. Apesar do valor do primeiro teste de carga de nós ter gerado um alto valor de consumo de memória, a partir da segunda carga de nós é possível ver um crescimento no consumo, mostrando uma tendência ascendente do simulador, mas mesmo assim um crescimento estável e pequeno. O valor de consumo da última carga de nós na simulação é quase igual ao valor da primeira carga.

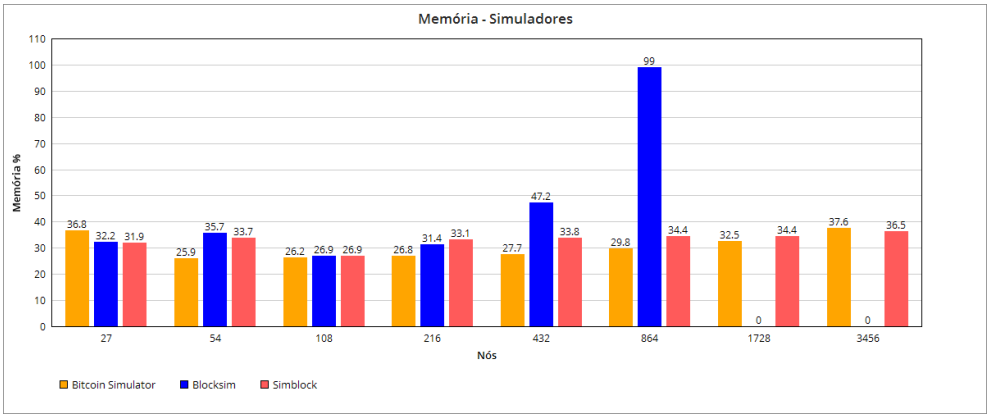
Figura 24 – Desempenho de Memória Bitcoin-Simulator




Fonte: Autor

A figura 25 faz um comparativo entre todos os simuladores sobre o desempenho de memória de cada um. É possível notar que nas primeiras cargas de nós existe uma alternância entre qual o simulador tem uma porcentagem maior de memória exigida para a simulação, a partir da carga 432 o BlockSim realiza um salto no consumo até chegar no ponto em que a máquina não consegue mais suportar o tanto de memória que é requisitado e interrompe o processo de simulação. SimBlock e Bitcoin-Simulator contém os mesmos valores aproximadamente de consumo de memória, existe uma alternância de qual desses dois simuladores utiliza mais recursos, em que o Bitcoin-Simulator leva uma vantagem, pois geralmente ele utiliza menos recursos que todos os outros simuladores.

Figura 25 – Desempenho de Memória




Fonte: Autor

### *5.4.4 Configuração e Flexibilidade do Simulador*

Esta métrica pretende avaliar a complexidade de configuração e flexibilidade de cada simulador.

#### *5.4.4.1 Instalação*

- Bitcoin-Simulator: O processo de instalação do Bitcoin-Simulator é o mais simples, assumindo que na máquina já tenha instalado a versão do NS-3 detalhado na documentação, é necessário apenas copiar as classes já implementadas no repositório oficial e referenciar no arquivo de referências do NS-3. No site oficial do projeto é possível ver um video que contem todo o passo a passo da instalação, bastando apenas seguir os passos mostrados, nesse aspecto o Bitcoin-Simulator se torna o mais fácil de instalar.
- SimBlock: Para a instalação desse simulador é necessário realizar o guia de instalação no repositório oficial, seguir o passo a passo e ficar atento a alguns detalhes para não complicar a instalação. Não chega a ser uma instalação complexa.
- BlockSim: O processo de instalação deste simulador apresenta um desafio específico. Apesar de contar com um guia de instalação fácil e objetivo, juntamente com *scripts* para automatizar o procedimento, alguns módulos do Python não são mais compatíveis entre si. É necessária uma pesquisa mais aprofundando para saber qual modulo instalar de forma que não cause conflitos de compatibilidade entre eles na execução do simulador. Devido a esse problema, o BlockSim é o simulador de maior complexidade na instalação.

#### *5.4.4.2 Documentação*

- Bitcoin-Simulator: A documentação deste simulador é acessível e bem elaborada, ela contém uma variedade de informações sobre o simulador. Nela, é possível encontrar detalhes sobre o processo de instalação, significado dos parâmetros de entrada, estrutura do projeto, resultados de experimentos, exemplos práticos de como realizar as simulações, entre outras coisas.
- BlockSim: A documentação desse simulador é focada nas classes que devem ser modificadas para reproduzir a simulação e no processo de instalação. É descrito algumas classes impostantes e alguns parâmetros para configuração de rede. Apresenta uma descrição do processo de instalação e o resumo com links que mostram os exemplos e resultados obtidos de artigos do sobre o uso do simulador,
- SimBlock: A documentação deste simulador contém o guia de instalação, o nome dos principais parâmetros utilizados para fazer uma simulação, descrição de alguns arquivos de saída gerados pela simulação e alguns requisitos técnicos para operar o simulador.

#### *5.4.4.3 Flexibilidade*

- BlockSim: Este simulador foi o que demostrou maior flexibilidade para a criação de modelos de simulação. Através dele é possível configurar parâmetros de rede em um nível que os outros simuladores não conseguem. É possível definir o tipo de rede ideal para simular como redes *Bitcoin* e Ethereum, outro fato importante é a possibilidade de detalhar o número de transações e quantidade de lote de transações que devem ser utilizadas na rede, além de especificar o tempo em que esses lotes de transações serão utilizados aumentando ainda mais o nível de detalhamento da simulação. É possível atribuir regras de negócios particulares a nós específicos e definir características de redes na interação entre esses nós, dando origens a várias opções de simulações.
- SimBlock: Este simulador tem um bom nível de flexibilidade, permitindo detalhar alguns processos da rede e da simulação de forma rápida. Apenas duas classes necessitam serem modificadas para configuração da rede e da simulação. Parte do código está comentado para facilitar o processo de inserção de dados e estimando possíveis valores que podem ser adicionados para ser criado modelos que serão utilizados nas simulações.
- Bitcoin-Simulator: Este simulador de todos se torna o de maior complexabilidade no

quesito de flexibilidade. Algumas boas práticas de desenvolvimento e código-limpo não foram implantadas no seu código-fonte, fazendo com que simples modificações dos parâmetros que não são dinâmicos para elaborar novos modelos se torne uma tarefa complicada. Para quem não tem experiencia com a configuração do simulador tera que investigar como as classes e métodos trocam informações. Mesmo desenvolvedores que já tem conhecimento da linguagem de programação utilizada no simulador, provavelmente terão um pouco de dificuldade e necessitarão de ferramentas para análise do código ao tentar construir uma versão adaptada do código para utilizar outro modelo de simulação. Os parâmetros de fácil configuração que podem ser adaptados dinamicamente estão escritos na documentação.

### *5.4.5 Diversidade de Dados do Simulador*

Nesta etapa será explorado a diversidade de dados fornecidas pelos simuladores como resultado da simulação com carga de vinte e sete nós.

#### *5.4.5.1 BlockSim*

Os resultados da simulação desenvolvida pelo simulador são armazenados em um arquivo chamado report.json. Ao ler os dados de saída da simulação é possível analisar as seguintes características da rede:

A quantidade de fork dos estados ficou entre 2 a 5. Houve uma fila de transação no estado do rio de janeiro com 2 valores.

Foram gerados 4.104 blocos em cada nó.

Cada bloco contem um cabeçalho com os seguintes atributos preenchidos: Número do bloco, Hash anterior, Timestamp, Coinbase, Dificuldade.

Na figura 26 é possível ver parte dos blocos criados após a simulação.

Esses dados permitem a análise personalizada da rede, como identificar que a cada bloco inserido a dificuldade de mineração aumenta enquanto o tempo de criação do bloco permanece relativamente constante.

Distrito Federal foi o estado que mais minerou blocos nesse no intervalo que a figura 23 mostra.

É possível notar que a integridade e a imutabilidade da *blockchain* foram garantidas, cada bloco contém o *hash* do bloco anterior criando uma ligação de blocos que pode ser

Figura 26 – Resultado simulação BlockSim


federal-1 difficulty:108671)>",
"<BlockHeader(#16 prevhash:fb6b838d timestamp:11-23 09:45:32 coinbase:bahia-3 difficulty:109244)>",
"<BlockHeader(#17 prevhash:e9f26897 timestamp:11-23 09:53:24 coinbase:pernambuco-5 difficulty:109715)>",
"<BlockHeader(#18 prevhash:26d351fe timestamp:11-23 10:01:27 coinbase:amazonas-2 difficulty:110198)>",
"<BlockHeader(#19 prevhash:9a704e7f timestamp:11-23 10:12:29 coinbase:rio de janeiro-6 difficulty:110859)>",
"<BlockHeader(#20 prevhash:c8362495 timestamp:11-23 10:24:49 coinbase:distrito federal-1 difficulty:111599)>",
"<BlockHeader(#21 prevhash:e2c85ee7 timestamp:11-23 10:33:38 coinbase:bahia-3 difficulty:112127)>",
"<BlockHeader(#22 prevhash:d44c5c1b timestamp:11-23 10:43:46 coinbase:distrito federal-1 difficulty:112734)>",
"<BlockHeader(#23 prevhash:40c65959 timestamp:11-23 10:50:07 coinbase:ceara-4 difficulty:113115)>",
"<BlockHeader(#24 prevhash:61666ba9 timestamp:11-23 10:59:24 coinbase:pernambuco-5 difficulty:113671)>",
"<BlockHeader(#25 prevhash:8c9cd338 timestamp:11-23 11:10:37 coinbase:amazonas-2 difficulty:114344)>",
"<BlockHeader(#26 prevhash:5ed5c066 timestamp:11-23 11:20:19 coinbase:rio grande do sul-7 difficulty:114926)>",
"<BlockHeader(#27 prevhash:48ce09ff timestamp:11-23 11:29:27 coinbase:distrito federal-1 difficulty:115474)>",


Fonte: Autor

identificada e auditada.

#### *5.4.5.2 Bitcoin-Simulator*

Os resultados gerados por esse simulador ficam disponíveis através da tela do terminal após a simulação. A seguir alguns resultados sobre a rede *blockchain* disponibilizados pelo simulador após realizar o experimento.

A simulação foi realizada 8300,91 mais rápida que o normal.

Estatísticas de conexões: Em média, cada nó tem 10.3684 conexões e cada minerador tem 23.125 conexões.

Blocos: O tempo médio de recebimento de blocos é de 751.423 segundos.

O tempo médio de propagação de blocos é de 13.1439 segundos.

O tamanho médio do bloco é de 3e+06 Bytes.

Foram gerados um total de 115.852 blocos.

Mensagens: A maioria das mensagens recebidas e enviadas são mensagens de bloco, que representam 49.9626% do total.

Blocos obsoletos: Há 1 bloco obsoleto

Forks: O tamanho do maior fork foi de 1 bloco. No total, houve 2 blocos em forks.

Mensagens: A maioria do tráfego é devido a mensagens de bloco, que representam 99.925.

Largura de banda: A largura de banda média de *download* é de 38.6005 Mbps e a largura de banda média de *upload* é de 38.5485 Mbps. A largura de banda total média por nó é de 77.149 Mbps.

Tempo de propagação do bloco: O tempo médio de propagação do bloco é de 13.1439 segundos.

O tempo de propagação do bloco para os mineradores é de 1.019 segundos em média.

Tempo de geração do bloco: O tempo médio para a geração de um bloco é de 0.0856472 segundos.

Esse são alguns dos dados disponibilizados pelo simulador após a construção da rede, existem ainda mais dados que são disponibilizados e podem ser utilizados conforme a necessidade da rede.

#### *5.4.5.3 SimBlock*

Os resultados gerados por esse simulador são armazenados em arquivos localizadas no diretório output. Entre esses arquivos gerados podem ser retirados as seguintes informações sobre a rede construída.

A Latência individual de cada transação.

Cada bloco contém um id e número de *hash*.

Cada nó recebe é armazenado em um arraylist chamado REGION e recebe um valor de um id.

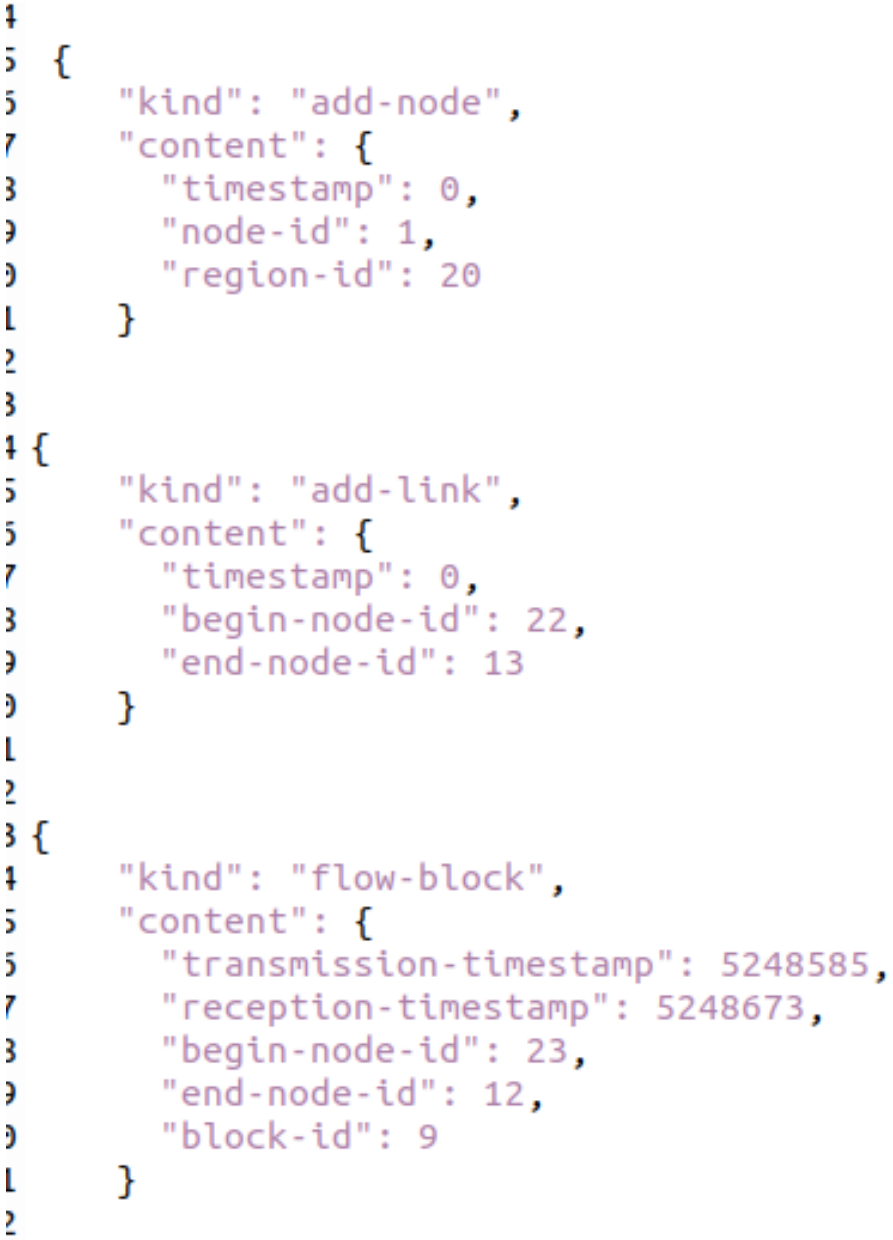
Altura do bloco: 145

Tamanho médio de órfãos: 0

Tempo de simulação: 2995

Na figura 27 é possível ver trés tipos diferentes de transações realizados na rede e salvas no arquivo output.json. Cada transação contém um tipo diferente de atividade que é especificada no cabeçalho. Através dos dados salvos no arquivo é possível criar um histórico de todas as atividades realizadas pelos nós e blocos na simulação.

Figura 27 – Tipos de transações SimBlock



```
{
    "kind": "add-node",
    "content": {
      "timestamp": 0,
      "node-id": 1,
      "region-id": 20
    }
}
{
    "kind": "add-link",
    "content": {
      "timestamp": 0,
      "begin-node-id": 22,
      "end-node-id": 13
    }
}
{
    "kind": "flow-block",
    "content": {
      "transmission-timestamp": 5248585,
      "reception-timestamp": 5248673,
      "begin-node-id": 23,
      "end-node-id": 12,
      "block-id": 9
    }
}
```

Fonte: Autor

# 6 CONCLUSÃO

Neste capítulo será apresentado uma conclusão para este trabalho com base em toda a pesquisa realizada.

## 6.1 Principais Resultados, Dificuldades e Benefícios

### *6.1.1 Principais Resultados*

Após a avaliação de desempenho dos três simuladores de redes *blockchain* BlockSim, SimBlock e Bitcoin-Simulator, em um cenário de simulação envolvendo uma rede *blockchain* conectando todas as capitais do Brasil, e a realização de testes de cargas de nós em cada rede para avaliar como os simuladores se comportavam, foi possível observar os seguintes resultados.

- Duração da Simulação: O SimBlock destacou-se como o simulador com o tempo de simulação mais eficiente durante a execução do experimento, seguido pelo BlockSim que obteve o segundo melhor tempo e o Bitcoin-Simulator ficou com o terceiro melhor tempo. Apesar de o BlockSim ter registrado dois tempos de simulação com valor zero devido o seu processo ter sido interrompido por limitação de *hardware*, ele regularmente demonstrou no experimento ter um tempo superior em relação ao Bitcoin-Simulator.
- Utilização da CPU: O bitcoin-Simulator destacou-se como o simulador que conseguiu realizar o experimento com o melhor desempenho de CPU, seguido pelo SimBlock que obteve o segundo melhor desempenho, o BlockSim obteve o terceiro melhor tempo de desempenho ao longo do crescimento das cargas de trabalho, chegando a alocar o dobro de desempenho de CPU comparado aos outros dois simuladores. Não foi levado em consideração as duas ultimas cargas de nós que não puderam ser concluídas por limitação de *hardware*.
- Utilização da Memória: O simulador Bitcoin-Simulator obteve os melhores resultados em relação a desempenho de memória, ficou um com um valor um pouco melhor que o simulador SimBlock que ficou em segundo lugar, já o simulador BlockSim, que no começo do experimento tinha um desempenho próximo aos outros dois simuladores, teve um salto de alocação de memória a ponto de não conseguir realizar o restante do experimento por limitação de *hardware* que acusava que o simulador já tinha consumido toda a memória disponível interrompendo de forma abrupta a simulação.

– Configuração e Flexibilidade do Simulador: O BlockSim apesar de ter o processo de instalação um pouco mais complexo se mostrou um ótimo simulador quando a questão é flexibilidade, contendo modelos com várias possibilidades de configuração, sua documentação não contém muitos detalhes. O SimBlock demonstrou ser um simulador fácil para ser instalado, flexível a ponto de ser fácil a criação de modelos e uma documentação que consegue descrever alguns parâmetros do simulador a fim de ajudar o experimento. Já o Bitcoin-Simulator é o simulador que contém o processo de instalação e simulação mais fácil, sua documentação tem detalhes que ajudam na instalação e desenvolvimento da simulação. Por outro lado, a flexibilidade para criar modelos é complexa. Depende do tipo de modelo, pois alguns podem ser construídos dinamicamente. Entretanto, para outros cenários e modelos, pode ser necessário recorrer a ferramentas auxiliares para facilitar a implementação e construção.
– Diversidade de Dados do Simulador: O Bitcoin-Simulator é o simulador que apresenta mais dados sobre a rede *blockchain*, apresentando métricas sobre a rede, características da simulação e sobre os nós criados. Os dados são exibidos terminal de execução. O BlockSim apresenta dados sobre a rede, podem ser citados a distribuição de nós, o rastreamento da cadeia de nós, aumento da dificuldade de mineração. Esses dados são armazenados em um arquivo. O SimBlock apresenta mais dados do fluxo de transações de redes, dos valores que são atribuídos aos nós e do seu rastreamento na rede, parte dos resultados são espalhados por aquivos.

.

### *6.1.2 Dificuldades*

Ao realizar o experimento foram descobertas as seguintes dificuldades em cada simulador.

– Bitcoin-Simulator: A principal dificuldade desse simulador está na flexibilidade do seu código-fonte para construir novos modelos de simulação. A ausência de boas práticas de programação no desenvolvimento do código-fonte adiciona complexidade para adaptar modelos e elaborar simulações com cenários diferentes, sendo necessário o uso de ferramentas para auxiliar na construção de modelos de desenvolvimento. A refatoração do código pode também impactar em algumas métricas que não são mostradas devido algumas bibliotecas utilizadas pelo simulador já estarem obsoletas, tornando necessário uma

atualização do código-fonte do simulador. Além disso, há a carência de um mecanismo para rastrear os blocos nas cadeias de nós e verificar a alocação de um bloco específico. Outras desvantagens seriam o tempo de simulação, o qual foi o mais longo entre todos os simuladores e a falta de algum arquivo que armazene as saídas da simulação de rede que podem ser vistas no terminal do Linux.

– BlockSim: A principal dificuldade encontrada nesse simulador foi a incompatibilidade de alguns módulos descritos como necessários no processo de instalação do simulador. Apesar desses módulos serem compatíveis e atualizados na época em que o simulador foi desenvolvido, hoje alguns se encontram obsoletos, ocasionando erros na hora da execução da simulação. Este problema pode passar despercebido durante a execução dos *scripts* de instalação automática, mas ficam evidentes no processo de execução da simulação, sendo necessário recorrer à pesquisa de módulos que sejam compatíveis atualmente para que a simulação seja executada corretamente. Outro fator que pode gerar dificuldade é a documentação, devido à falta de descrições de propriedades-chaves e os módulos que devem ser instalados. O desempenho de memória e CPU ficou abaixo dos outros dois simuladores, inclusive não foi possível realizar os dois últimos testes de carga, pois a memória exigida pelo simulador foi maior do que a oferecida pelo *hardware*, gerando uma interrupção abrupta da simulação. A falta de algumas métricas gerais da rede que poderiam ser calculada e inseridas no documento de saída, facilitando a análise da rede também podem ser citadas como uma dificuldade.
– SimBlock: A principal dificuldade desse simulador é a interpretação de alguns resultados, ocasionado pela falta de algum documento que mostre o que significa cada arquivo e dado gerado após a simulação. A falta de dados e organização da saída de dados gerados após a simulação da a rede que poderiam ser calculados para facilitar o processo de análise.

### *6.1.3 Benefícios*

Após o experimento é possível encontrar os seguintes benefícios em cada simulador.

– BlockSim: Além do nível detalhado de configuração de rede, blocos e controle de transações entre os nós, o BlockSim oferece a capacidade de adicionar mineradores na rede *blockchain*. O simulador permite o controle para verificar a cadeia de blocos gerada de cada nó, realizar auditorias, possibilita a opção de utilizar redes Ethereum. Além disso, oferece a flexibilidade de criar fluxos de transações e controlar quando devem ser ativados

na rede.

- Bitcoin-Simulator: Este simulador permite a possibilidade de configuração de alguns modelos de forma dinâmica ao executar a simulação. Na documentação é possível encontra vídeos para auxiliar no processo de instalação e execução da simulação, exemplos de projetos realizados, descrição de parâmetros para a construção da simulação. Ampla quantidade de dados sobre a rede são gerados e podem ser analisados pela interface do terminal. Foi o simulador que obteve o melhor desempenho de memória e CPU. Possibilidade de simular redes com *Bitcoin*, *Litecoin* e *Digecoin*.
- SimBlock: O tempo de simulação extremamento baixo comparado aos outros simuladores pode ser descrito como uma vantagem. A capacidade de construir experimentos de cenários com milhares de nós enquanto sem gastar muito processamento. Possui fácil instalação e configuração. Código-Fonte com descrição de algumas variáveis, possíveis valores que podem ser atribuídos e mostrando sites de onde pode ser pego referências para futuros valores opcionais. Possibilidade de simular redes com *Bitcoin*, *Litecoin* e *Digecoin*.

## 6.2 Considerações Finais

Após uma análise de desempenho de todas algumas métricas dos simuladores de rede *blockchain*: BlockSim, SimBlock e Bitcoin-Simulator, já é possível responder a seguinte pergunta aplicada nesse trabalho: Qual seria o melhor simulador de rede *blockchain* para uma organização que deseja implantar uma rede interligando todas as capitais do Brasil?

A escolha do simulador ideal requer uma avaliação do projeto de rede que será implantado. Alguns fatores para serem analisados seriam o número de nós em cada capital, a configuração de *hardware* da máquina utilizada para a criar a simulação, qual seria o foco de análise de rede, o tempo necessário para a implementação da simulação, a profundidade da configuração de alguns parâmetros de rede teriam que obter.

Na minha opinião o primeiro simulador que eu tentaria implementar no projeto da rede *blockchain* que passa por todas as capitais do Brasil seria o BlockSim. Embora exija um processamento um pouco mais robusto e a instalação possa ser um tanto complexa em comparação com outros simuladores, suas vantagens são relevantes. Em segundo lugar eu escolheria o Bitcoin-Simulator e em terceiro o SimBlock.

Fica a cargo da organização fazer um levantamento de como será o projeto de rede para escolher o simulador que melhor se adapte para o cenário desenvolvido.

## 6.3 Trabalhos Futuros

Sugestões para trabalhos futuros envolvem a potencial refatoração do código-fonte do Bitcoin-Simulator e BlockSim, juntamente com a proposta de atualização e aprimoramento da documentação no BlockSim e SimBlock. Além disso, sugere-se explorar a implementação de novas métricas de redes para o BlockSim e SimBlock. Estas direções visam aprimorar a eficiência, usabilidade e capacidades analíticas dos simuladores, contribuindo para o contínuo desenvolvimento e avanço da pesquisa nesta área.

# REFERÊNCIAS

AOKI, Y.; OTSUKI, K.; KANEKO, T.; BANNO, R.; SHUDO, K. Simblock: A blockchain network simulator. In: **IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. [*S. l.*: *s. n.*], 2019. p. 325–329.

BANKS, J. Introduction to simulation. In: **Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 1**. [*S. l.*: *s. n.*], 1999. p. 7–13.

BATUBARA, F. R.; UBACHT, J.; JANSSEN, M. Unraveling transparency and accountability in blockchain. In: **Proceedings of the 20th annual international conference on digital government research**. [*S. l.*: *s. n.*], 2019. p. 204–213.

BOVéRIO, M. A.; SILVA, V. A. F. d. Blockchain: uma tecnologia além da criptomoeda virtual. **Revista Interface Tecnológica**, v. 15, n. 1, p. 109–121, jun. 2018.

BRASILEIRO, C. **Registro de imóveis em blockchain entra em fase de testes em dois municípios brasileiros**. 2017. Disponível em: https://anafisco.org.br/registro-de-imoveis-em-blockchain-entra-em-fase-de-testes-em-dois-municipios-brasileiros/. Acesso em: 30 out. 2023.

BTC Soul. **Rodando na blockchain com Walmart e cia**. 2023. Disponível em: https://www.btcsoul.com/noticias/rodando-na-blockchain-com-walmart-e-cia/. Acesso em: 30 out. 2023.

CARSON, J. S. Modeling and simulation worldviews. In: **Proceedings of the 25th conference on Winter simulation**. [*S. l.*: *s. n.*], 1993. p. 18–23.

FARIA, C.; CORREIA, M. Blocksim: Blockchain simulator. In: **2019 IEEE International Conference on Blockchain (Blockchain)**. [*S. l.*: *s. n.*], 2019. p. 439–446.

Forbes. **Blockchain Voting Can Work: Both Republican and Democrats Use Voatz**. 2020. Disponível em: https://www.forbes.com/sites/robertanzalone/2020/09/30/blockchain-voting-can-work-both-republican-and-democrats-use-voatz/?sh=4dbd47ed4ee9. Acesso em: 30 out. 2023.

Forbes. **The Crucial Role of Patents and Intellectual Property in the Blockchain Industry**. 2023. Disponível em: https://www.forbes.com/sites/forbesbooksauthors/2023/08/17/the-crucial-role-of-patents-and-intellectual-property-in-the-blockchain-industry/. Acesso em: 30 out. 2023.

Forbes Brasil. **Blockchain 50: as empresas que mais usaram a tecnologia no último ano**. 2021. Disponível em: https://forbes.com.br/forbes-money/2021/02/blockchain-50-as-empresas-que-mais-usaram-a-tecnologia-no-ultimo-ano/. Acesso em: 30 out. 2023.

GERVAIS, A. **Bitcoin and Blockchain Simulator**. 2023. Disponível em: https://arthurgervais.github.io/Bitcoin-Simulator/. Acesso em: 18 nov. 2023.

GERVAIS, A.; KARAME, G.; WüST, K.; GLYKANTZIS, V.; RITZDORF, H.; CAPKUN, S. **On the Security and Performance of Proof of Work Blockchains**. 2016.

GROENFELDT, T. **IBM and Maersk Apply Blockchain to Container Shipping**. 2017. Disponível em: https://www.forbes.com/sites/tomgroenfeldt/2017/03/05/ibm-and-maersk-apply-blockchain-to-container-shipping/?sh=2c997b9f3f05. Acesso em: 29 out. 2023.

HANGGORO, D.; SARI, R. F. Performance comparison of simblock to ns-3 blockchain simulators. In: **2021 4th International Conference on Circuits, Systems and Simulation (ICCSS)**. [*S. l.*: *s. n.*], 2021. p. 45–50.

InfoMoney. **PwC anuncia serviço de auditoria baseado em blockchain**. 2023. Disponível em: https://www.infomoney.com.br/mercados/pwc-anuncia-servico-de-auditoria-baseado-em-blockchain/. Acesso em: 30 out. 2023.

JAIN, R. **The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling**. [*S. l.*]: john wiley & sons, 1990.

LONE, A.; MIR, R. Evaluating quality-of-service in blockchains using modelling and simulation tools. **International Journal of Computing and Digital Systems**, v. 10, 01 2020.

MEDEIROS, L. F. de; MOSER, A.; SANTOS, N. dos. A simulaÇÃo computacional como tÉcnica de pesquisa na administraÇÃo. **REVISTA INTERSABERES**, v. 9, n. Espec, p. 463–485, 1 1.

Microsoft. **Imagem de uma blockchain**. 2023. Disponível em: https://learn.microsoft.com/pt-br/archive/msdn-magazine/2018/march/blockchain-blockchain-fundamentals. Acesso em: 06 Abr. 2023.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. **Cryptography Mailing list at https://metzdowd.com**, 03 2009.

NGUYEN, Q. K.; DANG, Q. V. Blockchain technology for the advancement of the future. In: **2018 4th International Conference on Green Technology and Sustainable Development (GTSD)**. [*S. l.*: *s. n.*], 2018. p. 483–486.

OpenSea. **Azuki Doodles Collection**. 2023. Disponível em: https://opensea.io/collection/azuki-doodles. Acesso em: 30 out. 2023.

PAULAVIčIUS, R.; GRIGAITIS, S.; FILATOVAS, E. A systematic review and empirical analysis of blockchain simulators. **IEEE Access**, v. 9, p. 38010–38028, 2021.

SICHEL, R. L.; CALIXTO, S. R. Criptomoedas. impactos na economia global. perspectivas. **Revista de Direito da Cidade**, 2018.

SMETANIN, S.; OMETOV, A.; KOMAROV, M.; MASEK, P.; KOUCHERYAVY, Y. Blockchain evaluation approaches: State-of-the-art and future perspective. **Sensors**, v. 20, n. 12, 2020. ISSN 1424-8220.

SOARES, S. P. L.; SOARES, A. C. d. S.; ALVES, A. A. A importância da implementação de uma política de segurança da informação / the importance of implementing an information security policy. **Brazilian Journal of Development**, v. 7, n. 4, p. 37162–37171, Apr. 2021.

THOMAZ, G.; CAMILO, G.; SOUZA, L.; DUARTE, O. Uma análise comparativa da arquitetura e desempenho de plataformas de corrente de blocos permissionadas para contratos inteligentes. In: **Anais do IV Workshop em Blockchain: Teoria, Tecnologias e Aplicações**. Porto Alegre, RS, Brasil: SBC, 2021. p. 114–127. ISSN 0000-0000.

Tokyo Institute of Technology. **SimBlock repositório do github**. 2023. Disponível em: https://github.com/dsg-titech/simblock-visualizer. Acesso em: 06 abr. 2023.