isec
Engenharia

MESTRADO EM INFORMÁTICA E SISTEMAS

Sistema transparente de monitorização da qualidade do ar usando Blockchain e LoRaWan

Autor

Marco Aurélio Lopes Sequeira

Orientadores

João Carlos Costa Faria da Cunha

José Manuel Meireles Marinho

Coimbra, abril de 2021

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA

isec
Engenharia

DEPARTAMENTO DE INFORMÁTICA E SISTEMAS

Sistema transparente de monitorização da qualidade do ar usando Blockchain e LoRaWan

Relatório de Trabalho de Projeto para a obtenção do grau de Mestre em Informática e Sistemas

Especialização em Desenvolvimento de Software

Autor

Marco Aurélio Lopes Sequeira

Orientadores

João Carlos Costa Faria da Cunha

José Manuel Meireles Marinho

Coimbra, abril de 2021

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

AGRADECIMENTOS

Concluída mais uma fase na minha vida académica, é necessário agradecer a todos aqueles que, de uma ou de outra forma, me ajudaram a chegar até aqui. Desta forma gostaria de deixar um agradecimento especial:

Aos meus orientadores da unidade curricular de Projeto Industrial - Doutor João Cunha e Doutor José Marinho – por toda a disponibilidade e ajuda prestada ao longo deste trabalho.

Ao Instituto Superior de Engenharia de Coimbra, a todos os professores e ao FikaLab ISEC por me terem oferecido uma formação superior de elevada qualidade, bem como excelentes condições de aprendizagem.

Aos colegas de curso por toda a motivação, companhia e disponibilidade para trabalhar em grupo.

A todos os meus amigos por toda a motivação, amizade e companhia ao longo de todo este tempo.

Aos meus pais por todo apoio dado e pelo esforço que tiveram para me ajudarem a chegar até aqui.

À minha namorada por todo o apoio, paciência, companhia e motivação prestadas ao longo desta fase da minha vida.

RESUMO

Internet of Things (IoT) corresponde à interligação de objetos que utilizamos no quotidiano através de redes de comunicação de dados. Nos últimos anos estes objetos passaram a ser “inteligentes”, ou seja, dotados de capacidade de processamento e interligação. Atualmente existe um grande investimento em IoT, com tendência para aumentar, expandindo-se por várias áreas, tais como a saúde, a agricultura, cadeias de fornecimento, habitações ou cidades. Muitas das aplicações de IoT baseiam-se em sensores que, de um modo autónomo, recolhem e transmitem dados. Por esta razão, existem preocupações relacionada com a privacidade e a segurança dos dados – nomeadamente adulteração dos dados recolhidos e armazenados.

A tecnologia Blockchain, pelo seu lado, conhecida por ter possibilitado a generalização da criptomoeda, ao aplicar técnicas de descentralização, consenso e imutabilidade, garantiu a transparência, privacidade e confiança das transações realizadas. O seu uso em outras áreas passou então a ser estudado, nomeadamente a sua integração em redes IoT para tratar questões inerentes de privacidade e segurança dos dados.

Neste trabalho é realizado um estudo sobre IoT e Blockchain e proposta uma *framework* (BC4IoT) que integra uma rede de sensores baseada em LoRaWAN (protocolo de rede de baixa potência) com uma Blockchain baseada em Ethereum. Esta *framework* tem como principal objetivo proporcionar um ambiente onde seja possível manter os dados recolhidos numa rede de sensores num local seguro, inviolável e público. BC4IoT pode ser adaptada a diversos casos de uso, bem como permitir que diversos serviços externos se liguem à mesma para obter informação.

Palavras-Chave: Blockchain, Internet of Things, IoT, LoRa, LoRaWAN, Smart Contracts, Ethereum

ABSTRACT

Internet of Things (IoT) corresponds to the interconnection of objects that we use daily through data communication networks. These objects have become "smart" - they have the ability to process and interconnect. Investment in IoT has increased and is expanding in several areas. Many of the IoT applications are based on sensors that autonomously collect and transmit data. Thus, there are concerns related to data privacy and security - namely adulteration with the data collected and stored.

Blockchain technology is known for cryptocurrency where it applies techniques of decentralization, consensus and immutability. Thus, it manages to guarantee the transparency, privacy and trust of the transactions carried out. Its use started to be studied for other areas, namely its integration in IoT networks to address privacy and data security issues.

In this work, a study on IoT and Blockchain is made and a framework (BC4IoT) is proposed that integrates a sensor network based on LoRaWAN (low power network protocol) with a Blockchain based on Ethereum. This framework aims to create an environment where it is possible to keep *the* data collected in a network of sensors in a safe, inviolable and public place. BC4IoT can be adapted to different use cases and allow several external services to connect to it to obtain data.

Palavras-Chave: Blockchain, Internet of Things, IoT, LoRa, LoRaWAN, Smart Contracts, Ethereum

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

ACRÓNIMOS E SIGLAS

IoT – Internet of Things

LoRa – Long Range

LoraWAN – Long Range Wide Area Network

LPWAN – Low Power Wide Area Network

ISM – Industrial, Scientific and Medical

LPWA – Low Power Wide Area

FOTA – Firmware Over The Air

TTN – The Things Network

ICS – Industrial Control System

SPV – Simplified Payment Verification

PoW – Proof-of-Work

PoS – Proof-of-Stake

DPoS – Delegated Proof-of-Stake

ETH – Ether

EVM – Ethereum Virtual Machine

RFID – Radio Frequency Identification

BC4IoT – BlockChain for IoT

1 Introdução

A IoT (Internet of Things) consiste numa rede global de dispositivos autónomos e frequentemente inteligentes com capacidade de comunicação. Atualmente existe um grande investimento em IoT e o mesmo tende a aumentar, havendo uma expansão para diversas áreas de aplicação tais como a saúde, a agricultura, cadeias de fornecimento, habitações ou cidades. Devido ao seu crescimento, as ameaças também tendem a aumentar e a confiança nos dados recolhidos pode ser colocada em questão.

Por outro lado, existe uma tecnologia que tem estado a expandir-se em áreas onde existe a necessidade de garantir a confiança – a tecnologia Blockckchain. Esta tem tido um grande impacto no domínio das criptomoedas e apresenta características de relevo como descentralização, segurança, privacidade e imutabilidade. A integração da tecnologia Blockchain em redes IoT tem sido vista como uma possível solução para alguns problemas relacionados com as redes IoT - como o caso da confiança nos dados apresentados, transparência e privacidade. Com esta integração, é possível armazenar os dados recolhidos pelos dispositivos IoT de forma confiável.

O trabalho descrito neste relatório teve como objetivo investigar a tecnologia Blockchain, explorando a possibilidade da sua integração com IoT, por forma a abordar os desafios da IoT. Neste sentido, foi ainda definida uma *framework* que integra ambas as tecnologias, tendo sido aplicada a um caso de estudo.

No presente capítulo é apresentado o enquadramento do tema, os objetivos e metodologia utilizada, bem como a estrutura do relatório.

1.1 Descrição do problema

A IoT é constituída por duas palavras principais: “Internet” e “Things”. A Internet é uma rede de dados global baseada em protocolos *standards* (pilha protocolar TCP/IP) e que permite a interligação de milhões de dispositivos em todo mundo [1]. De acordo com as estatísticas globais, no primeiro trimestre de 2020, estimava-se que esta tinha 4.6 mil milhões de utilizadores, o que corresponde a quase 60% da população mundial [2].

As “coisas” podem ser qualquer dispositivo ou objeto do quotidiano. Estes objetos, ao ligarem-se à internet e ao comunicarem dados que tenham recolhido de forma autónoma, acabam por poder ser consideradas “inteligentes” [3]. Estima-se que em 2025 estejam conectados 75 mil milhões de dispositivos IoT [1].

O desenvolvimento das áreas de IoT e Big Data tem vindo a acelerar ao longo dos últimos anos e, desta forma, tem afetado todas as áreas das tecnologias e negócios, aumentando também os benefícios tanto para as organizações como para os indivíduos [3]. Dado o

número elevado de dispositivos “inteligentes” que estão ligados às redes IoT, a análise de dados começou a ter outros desafios, tais como o volume de dados a processar, a sua heterogeneidade e a velocidade a que os mesmos são processados [3].

Os dados recolhidos pelos diversos sensores IoT, após serem processados e analisados, fornecem um conjunto de informações que são do interesse de várias entidades e, muitas das vezes, ajudam à tomada de decisões dentro das mesmas. Desta forma, os dados recolhidos pelos sensores IoT revestem-se de uma grande importância e, caso sejam de alguma forma adulterados, podem levar a tomadas de decisão incorretas, levando a elevados prejuízos tanto para as entidades envolvidas como para os cidadãos.

A possibilidade de adulteração dos dados antes de os mesmos serem usados para diversos fins retira alguma confiança quanto à sua veracidade, o que resulta num problema que deve ser resolvido. Por esta razão, são necessárias soluções que garantam que estes dados não possam ser adulterados ou que permitam a deteção de situações anómalas.

1.2 Objetivos do trabalho

Para colmatar as questões relacionadas com a confiança nos dados recolhidos em redes IoT, tem vindo a surgiu a ideia de integrar a tecnologia Blockchain com as redes IoT [4] [5].

A tecnologia Blockchain consiste numa base de dados descentralizada que é replicada e partilhada entre os membros de uma rede, onde tudo fica registado e pode ser consultado por qualquer pessoa ou dispositivo. A Blockchain, enquanto base de dados, regista transações, sejam estas referentes a criptomoedas (tal como na sua conceção original) ou a um determinado sistema de votação, entre muitas outras possibilidades. Um registo em Blockchain é imutável, o que significa que, uma vez criado, não pode ser apagado ou alterado.

A integração da tecnologia Blockchain em redes IoT tem sido vista como uma possível solução para garantir a confiança dos utilizadores [6]. Essas soluções são caracterizadas principalmente pela adição de segurança, privacidade, transparência e redução de custos (principalmente porque é possível eliminar intermediários).

O projeto descrito neste documento teve como principal objetivo resolver alguns dos problemas de confiança nos dados recolhidos pelos sensores de uma rede IoT - LoRaWAN[1] - utilizando para o efeito a tecnologia Blockchain que armazenará os dados recolhidos pelos sensores. A arquitetura concebida para a resolução deste problema pretendeu ser versátil ao ponto de poder ser adaptada a diversos casos de estudo das mais diversas áreas que

[1] LoRaWAN é um protocolo de rede *Low Power, Wide Area* (LPWA) projetado para conectar ‘things’ que funcionam através de baterias e estão ligadas à internet sem fios, estando organizadas em redes regionais, nacionais ou globais. LoRaWAN visa os principais requisitos de IoT, como serviços de comunicação bidirecionais, segurança *end-to-end*, mobilidade e localização [80].

necessitem da integração de sensores IoT tais como saúde, agricultura, cidades inteligentes, seguradoras, entre outras.

Em concreto, este projeto levou à criação de uma *framework*, designada BC4IoT, sistema que possui as seguintes características:

- **Público** - qualquer pessoa deve pode participar (através da disponibilização de sensores) e visualizar os dados recolhidos;
- **Integro** - através das características de Blockchain, deve ser possível garantir a integridade dos dados recolhidos (i.e., depois de recolhidos e registados, os dados não podem ser adulterados ou removidos);
- **Democrático** - qualquer pessoa deve poder participar, inserindo, de forma transparente novos sensores, ou participando da rede Blockchain.

A *framework* proposta baseia-se na tecnologia de comunicação LoRaWan e na Blockchain Ethereum, tendo sido aplicada a um caso de estudo específico, um sistema público de recolha, armazenamento e consulta de informações ambientais de uma cidade.

1.3 Estrutura do relatório

Além do presente capítulo, onde foi descrito, de uma forma genérica, o problema ao qual se pretendeu dar resposta e apresentados os principais objetivos do trabalho desenvolvido, este relatório é constituído por mais quatro capítulos.

No segundo capítulo é apresentado o estudo teórico realizado acerca das tecnologias que podem resolver os problemas de confiança nos dados recolhidos por sensores em redes IoT. Das tecnologias e projetos estudados foram retiradas as devidas conclusões e foram selecionadas as tecnologias que melhor davam resposta ao problema em análise.

No terceiro capítulo é proposta e descrita uma solução genérica para o problema da potencial falta de confiança nos dados recolhidos em redes IoT (a *framework* BC4IoT).

No quarto capítulo é apresentado um caso de estudo específico de aumento da confiança nos dados recolhidos por sensores em redes IoT através da aplicação da solução proposta. Este consiste numa hipotética comunidade em que os seus membros colaboram através da recolha de dados meteorológicos. Também é descrito o ambiente de simulação usado. São ainda descritos os testes efetuados e apresentados os resultados obtidos, sendo feita uma análise crítica e tiradas conclusões sobre a viabilidade da solução proposta.

No quinto capítulo tecem-se conclusões sobre o trabalho apresentado e apresentam-se algumas diretrizes para trabalho futuro.

2 IoT e Blockchain

No presente capítulo é feita uma análise e escolha de tecnologias suscetíveis de serem usadas no projeto apresentado.

2.1 Internet of Things

De uma forma genérica, a noção de Internet of Things corresponde à interligação (em rede) de objetos que utilizamos no quotidiano. Os objetos passaram a ser “inteligentes” (com capacidade de processamento e interligação) e começaram a comunicar e a interagir em rede.

O conceito de IoT pode ser definido como a junção de dois termos [7]. O primeiro termo é a Internet, que possui a capacidade de conectar milhares de milhões de utilizadores [1]. O outro termo é "Coisas", que designa os dispositivos e objetos que utilizamos diariamente e que se transformaram em objetos inteligentes ao ligarem-se à Internet.

A IoT visa tornar a Internet ainda mais imersiva e difundida. Além disso, ao facilitar o acesso e a interação com uma ampla variedade de dispositivos como, por exemplo, eletrodomésticos, câmaras de vigilância, sensores de monitorização, *displays* e veículos, promoverá o desenvolvimento de diversas aplicações. Estas aplicações fazem uso da enorme quantidade e variedade de dados gerados por esses objetos para fornecer novos serviços a cidadãos, empresas e administração pública. Este paradigma encontra aplicação em diversos domínios, como automação residencial, automação industrial, equipamentos médicos, assistência médica móvel, assistência a idosos, gestão inteligente de energia e redes inteligentes, indústria automóvel e gestão de tráfego, entre muitos outros [8]. Estima-se que, em 2025, a IoT possa conectar cerca de 75 mil milhões de dispositivos (Figura 2-1).

De acordo com [9], a IoT é a interação entre o mundo digital e físico (Figura 2-2). Este paradigma estende a comunicação, ou seja, interação e troca de dados, entre humanos e aplicações para integrar as “coisas” [10] [7]. A IoT também pode ser considerada como uma rede global que permite a comunicação de humano para humano, de humano para “coisas” e “coisas” para “coisas”, fornecendo identidade única a todos e a cada um [11].

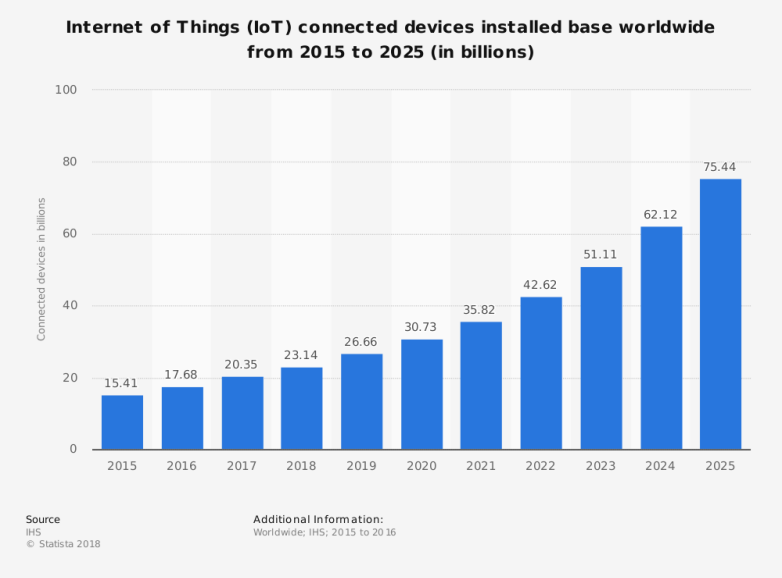

Figura 2-1 - Estimativa de dispositivos ligados a IoT [90]

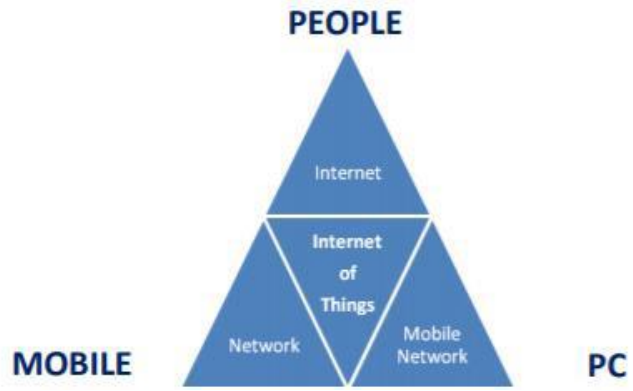

Figura 2-2 - Interação entre o mundo físico e o mundo digital [7]

O principal objetivo da IoT é tornar tudo mais eficiente e mais eficaz convergindo para a redução de custos e redução do desperdício – água e luz, entre outros bens. A IoT está a ter um impacto cada vez maior na sociedade. Isto deve-se à capacidade das tecnologias envolvidas nos permitirem analisar e estudar os dados recolhidos, bem como nos trazerem comodidade e simplicidade na execução nas tarefas do dia-a-dia. A Figura 2-3 apresenta o resultado de um estudo que data de 2018 e que analisou 1600 projetos da área de IoT [12].

As áreas com mais projetos em IoT identificados são as *Smart Cities*, seguida de *Connected Industry* e *Connected Building*. É o continente Americano que possui um maior número de projetos em IoT (45%), seguido da Europa (35%) e da Ásia (16%). Na Europa está localizada a maioria dos projetos de *Smart Cities* (45%), sendo que as Américas possuem mais projetos ligados a *Connected Cars* (55%) e *Connected Health* (54%). No caso da Ásia, o seu ponto forte é a *Smart Agriculture*, onde possui 31% dos projetos na área.

As *Smart Cities* subiram no ranking muito devido a iniciativas governamentais que apostam bastante nesta área. *Connected Building* também teve uma subida em relação ao estudo de 2016 [13], pois esta é uma área que tem obtido bons resultados, principalmente no aumento da eficiência operacional e redução de custos.

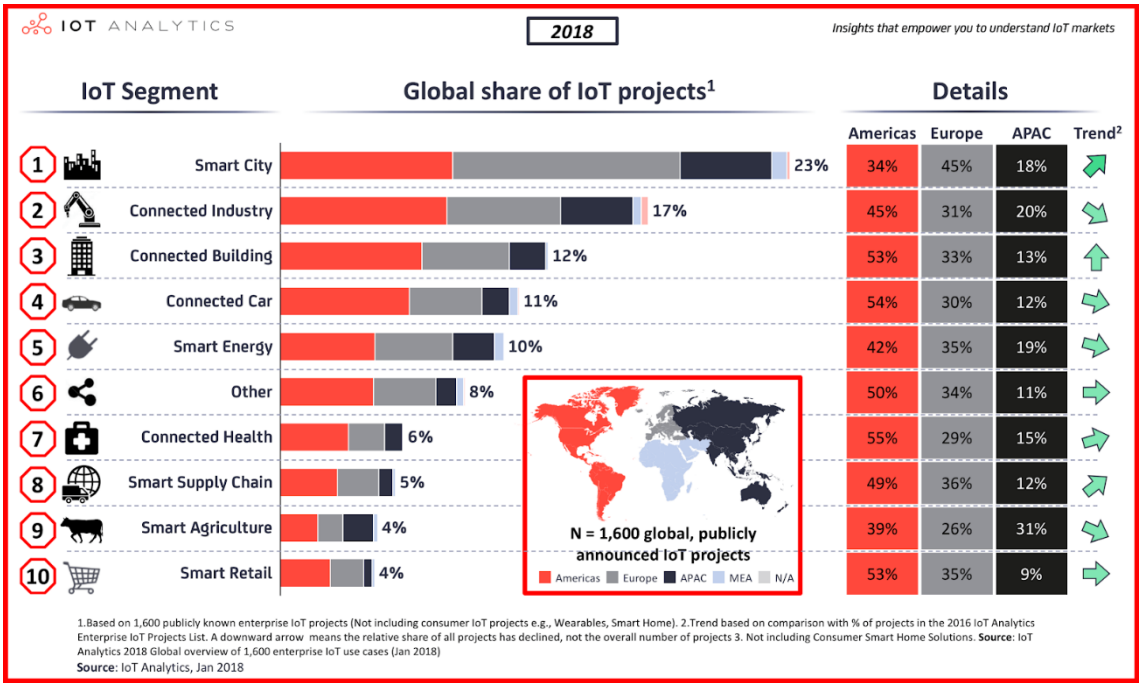

Figura 2-3 - Interação entre o mundo físico e o mundo digital [12]

A aplicação mais utilizada na área das *Smart Cities* é *Smart Traffic*, onde é possível, por exemplo, controlar o trânsito numa cidade, a partilha de bicicletas, as vias de autocarros autónimos, o carregamento de veículos elétricos e o controlo de estacionamentos. Nesta área ainda existe um controlo sobre outros serviços públicos como é o caso da iluminação, as questões ambientais e a segurança (Figura 2-4).

A área de *Connected Industry* envolve um largo conjunto de "coisas" que estão ligadas no interior e no exterior de uma fábrica. As aplicações mais populares estão relacionadas com o controlo de ativos localizados no exterior de uma fábrica como, por exemplo, guindastes e empilhadoras, entre outros, que necessitam de um controlo e de uma atenção especial. Outra área de ação são as *Smart Factories* onde está, por exemplo, incluída a monitorização do chão de uma fábrica e o controlo remoto de algumas máquinas, entre outros fatores presentes em ambientes fabris [12].

A área de *Connected Building* está a ter uma boa evolução, principalmente pelos resultados que tem vindo a obter. A maioria dos projetos desta área inclui a automação de instalações para reduzir custos energéticos (existem alguns casos em que a redução vai de 10% a 15%). Existem outros projetos que estão relacionados com a segurança dos edifícios e com a temperatura dos mesmos [12].

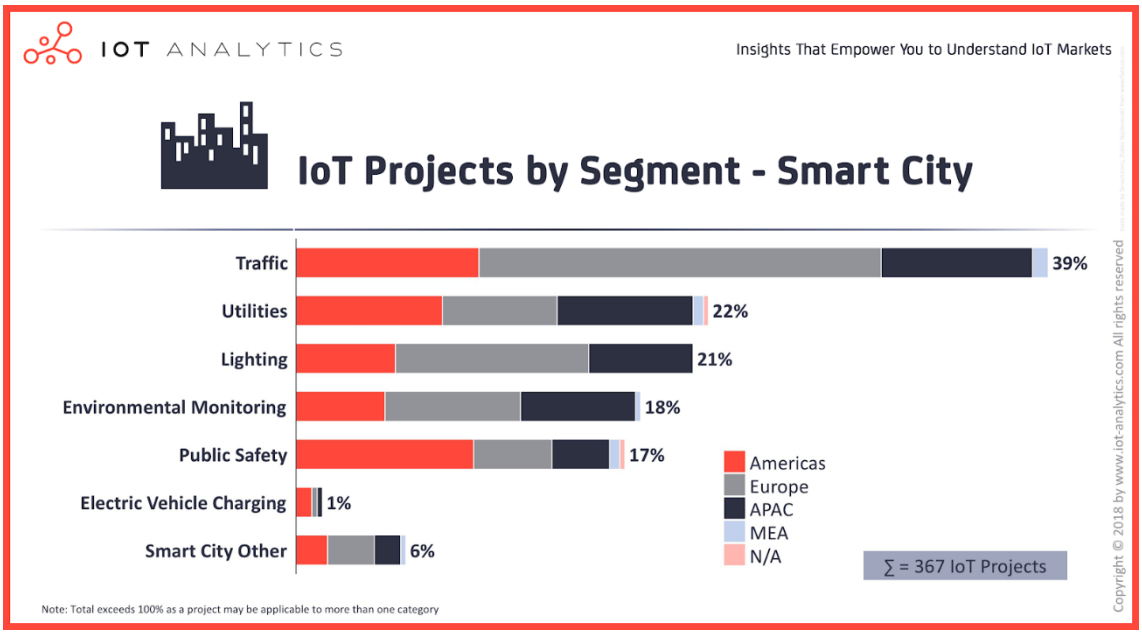

Figura 2-4 - Áreas de atuação numa Smart City [12]

A área de *Connected Car*, também em expansão e em desenvolvimento, tem já alguns projetos de bastante qualidade como é o caso da *TracknStop* - uma empresa de software de controlo remoto na Irlanda que oferece soluções de diagnóstico de veículos, incluindo acompanhamento em tempo real, controlo de leituras de sensores e controlo remoto de veículos [12].

Com o aumento da necessidade de alimentos para a população, os governos e as empresas estão a ajudar os agricultores a apostar em tecnologias avançadas para atender às suas necessidades (área de *Smart Agriculture*). Através destas tecnologias, os agricultores podem controlar diversos fatores que são importantes no seu dia-a-dia como é o caso do controlo da saúde do seu gado e produtos alimentares, evitando, assim, que os mesmos fiquem danificados ou doentes. É possível recolher informação sobre a localização do gado, quais as condições do solo e controlar a utilização de água e fertilizantes, entre outros fatores importantes para a área. Desta forma, os agricultores conseguem ter um maior controlo sobre a sua quinta e conseguem reduzir custos nas suas tarefas do quotidiano [14].

Desta forma, devido às suas caraterísticas, as implementações de casos de estudo semelhantes aos anteriores utilizando as tecnologias LoRa e LoRaWAN têm vindo a crescer.

2.1.1 LoRa/LoRaWAN

A tecnologia LoRa (Long Range), destinada a comunicações de longo alcance e baixo consumo de energia, utiliza o espectro eletromagnético não licenciado e este facto tem duas consequências principais [15]. A primeira é a coexistência com outras tecnologias - os dispositivos devem ser capazes de comunicar e ser resilientes à interferência causada

por outros elementos que partilham o mesmo espectro rádio. A segunda consequência é a limitação de transmissão imposta pelas autoridades reguladoras. Por exemplo, na Europa, existe uma limitação de *transmission duty cycle* (ciclo de trabalho de transmissão) de 1%, para tentar ter um uso responsável do espectro por cada utilizador [15].

LoraWAN (Long Range Wide Area Network) é uma abordagem LPWAN (Low Power Wide Area Network) baseada na tecnologia LoRa. Foi projetada para conectar as "coisas" sem-fios que são alimentadas por bateria. As "coisas" estão ligadas à Internet em redes regionais, nacionais ou globais. O protocolo LoRaWAN aproveita o espectro de rádio não licenciado na banda ISM (Industrial, Scientific and Medical) [16]. LoRaWAN define o protocolo de comunicação e a arquitetura do sistema para a rede, enquanto a camada física LoRa possibilita a comunicação de longo alcance [17]. Enquanto a empresa Semtech [18] fornece os chips de rádio com a LoRa Technology, a LoRa Alliance [19], uma associação sem fins lucrativos e a aliança de tecnologia de crescimento mais rápido no mundo (contanto com 148 operadores LoRaWAN em 162 países [19]), impulsiona a padronização e a harmonização global do protocolo LoRaWAN [16]. A Figura 2-5 apesenta a estrutura de LoRa e LoRaWAN.

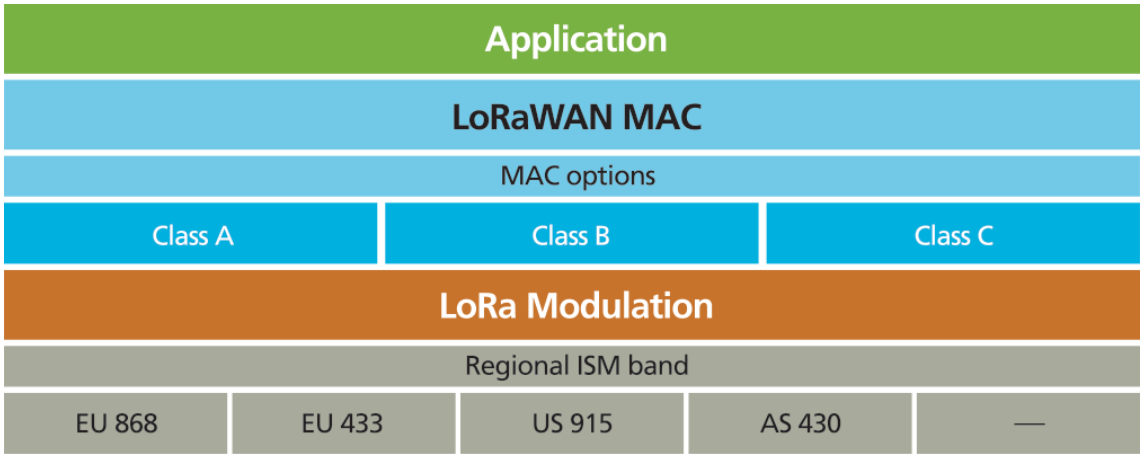

Figura 2-5 - Estrutura do LoRa e LoRaWAN [16]

2.1.1.1 Funcionamento da tecnologia LoRaWAN

Numa rede LoRaWAN (Figura 2-6 e Tabela 2-1), os nós não estão associados a um *gateway* específico - os dados transmitidos podem ser recebidos por um ou por vários *gateways*. Cada *gateway* encaminha os pacotes recebidos para o servidor de rede que é responsável por filtrar os pacotes redundantes e realizar verificações de segurança, entre outras operações [17].

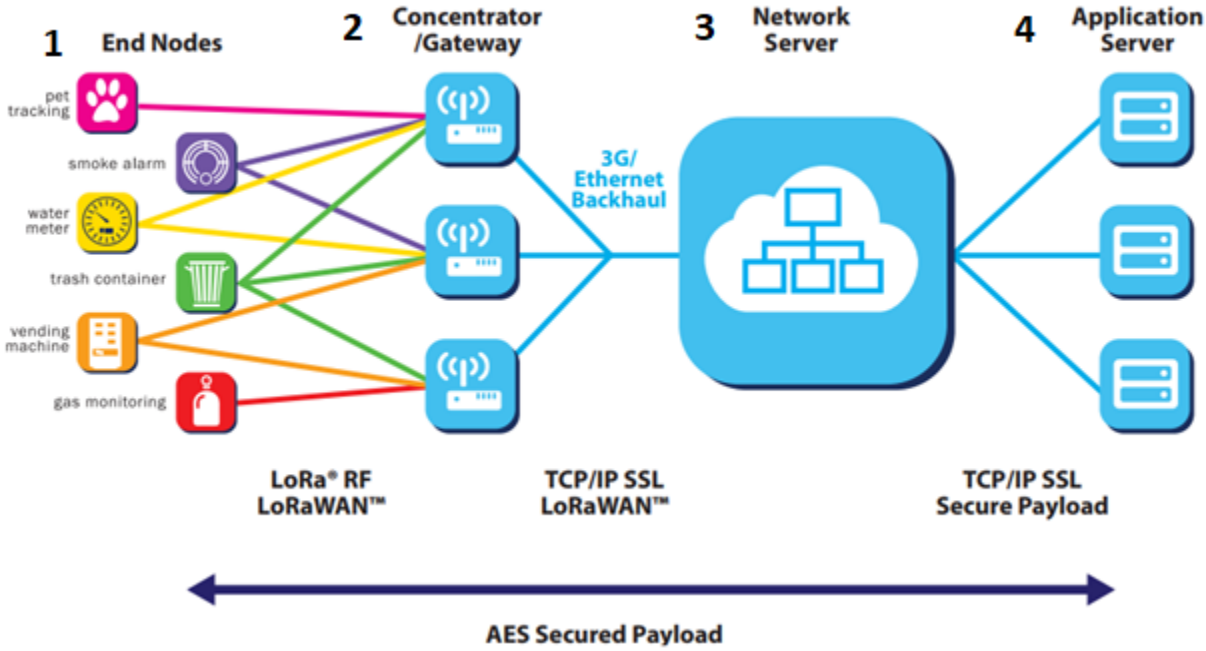

Figura 2-6 - Funcionamento de LoRaWAN (adaptado de [17]**)**

A Tabela 2-1 descreve os componentes existentes na LoRaWAN e que estão apresentados na Figura 2-6.

2.1.1.2 Principais características

As principais características da tecnologia LoRaWAN são as seguintes [16]:

- **Longo alcance**: conecta dispositivos a até cerca de 48 quilómetros de distância em áreas rurais e penetra em densos ambientes internos ou urbanos;
- **Baixa potência**: requer energia mínima, com vida útil prolongada da bateria de até 10 anos, minimizando os custos de substituição;
- **Segurança**: possui criptografia AES128 *end-to-end*, autenticação mútua, proteção de integridade e confidencialidade;
- ***Standardized***: oferece interoperabilidade de dispositivos e disponibilidade global de redes LoRaWAN para rápida implementação de aplicações IoT em qualquer lugar;
- **Geolocalização**: permite aplicações de *tracking* sem GPS, oferecendo benefícios exclusivos de baixa potência;
- **Móvel**: mantém a comunicação com dispositivos em movimento sem sobrecarregar o consumo de energia;
- **Alta capacidade**: suporta milhões de mensagens por estação base, atendendo às necessidades das operadoras de redes públicas que servem grandes mercados;
- **Baixo custo**: reduz o investimento em infraestruturas, a despesa de substituição de bateria e, em última instância, as despesas operacionais.

Tabela 2-1 - Dispositivos em LoRaWAN

Devices	
End-Nodes (1)	Os *end-node* são sensores de baixa potência ou alimentados por bateria que enviam dados para aplicações através de *gateways* LoRa. Normalmente são as "coisas" - sensores que recolhem dados dos mais diversos elementos (humidade, temperatura, etc.) [16].
***Gateways* (2)**	Os sensores recolhem e transmitem dados para os *gateways* em distâncias próximas e distantes, internas e externas, com requisitos mínimos de energia. Os *gateways* aceitam dados dos *end-nodes* e retransmitem-nos para o network server. A comunicação entre os *gateways* e o *network server* é muitas vezes através de *backhaul* 3G ou Ethernet que permite uma melhor performance na transferência de dados. Uma rede LoRaWAN normalmente contém uma infinidade de *gateways* [16].
Redes	
***Network Server* (3)**	Os *gateways* enviam informações para o *network server* que é responsável pela gestão da rede efetuando tarefas como: filtragem e rejeição de dados duplicados, *dynamic frame routing*, *adaptive rate control*, gestão de tráfego, e administração [16].
Aplicações	
***Application Servers* (4)**	As aplicações recebem e interpretam os dados recolhidos pelos dispositivos LoRa, aplicando técnicas como *machine learning* e inteligência artificial para resolver problemas de negócios [16].

2.1.1.3 Aplicações da tecnologia LoRa

A tecnologia LoRa permite que aplicações inteligentes de IoT tenham em conta alguns dos principais desafios da atualidade: gestão de energia, redução do consumo de recursos naturais, controlo de poluição, eficiência de infraestruturas e prevenção de desastres, entre outros.

A LoRa tem cerca de 87 milhões de dispositivos conectados a redes em 100 países, com tendência de crescimento. Esta tecnologia também acumulou mais de 600 casos de uso conhecidos em diversas áreas, algumas das quais mencionadas no início da presente secção, tais como:

- **Smart Agriculture**: *Smart Farming Networks*, *Smart Cattle Ranching*, *Ingestible Cattle Health Tracker*, *Soil Moisture Monitoring*, *Cattle Health Monitoring*, *Smart Soil Sensors*, *Autonomous Irrigation*, *Rural IoT Network for High-Tech Ag*;

- **Smart Cities**: *Smart Metering & Lighting Solutions*, *Structural Health Monitoring System*, *Smart Water Leakage Detection*, *Smart Flood Sensors*, *Smart Bus Schedule Signs, Smart Street Lighting*, *Latin America Network Coverage*, *Street Light Gateway*, *Smart Waste Management*, *Connecting the Cook Islands*;
- **Smart Environment**: *Natural Disaster Communication*, *Endangered Species Protection*, *Water System Monitoring*;
- **Smart Healthcare**: *IoT-Connected Hearing Aids*, *Alzheimer Patient Tracking*, *Dementia Patient Location*, *Smart Wearables for Kids & Seniors*, *Skier Safety Monitoring*;
- **Smart Homes**: *Water Leakage Detection and Damage Prevention*, *Structural Health Monitoring System*, *Commercial Smart Building Applications*, *Smart Fire Evacuation*, *Smart Door Locks*, *Smart Home Management*;
- **Smart Industrial Control**: *Smart Construction Machine Usage*, *Smart Mining Solutions*, *Smart Industrial Management*, *Remote Sensors in Harsh Environments*;
- **Smart Metering**: *Smart Metering & Lighting Solutions*, *Convenient Utility Metering*, *Real-Time Utility Monitoring*, *Smart Water Leakage Detection*, *Real-Time Water Metering*, *Smart Water & Gas Management*, *Urban Smart Grid*, *Gas Level Monitoring*;
- **Smart Supply Chain & Logistics**: *Asset Tracking for Aerospace Manufacturing*, *Cargo Tracking*, *Bike Share Location Tracking*, *Airport Asset Tracking*, *Stolen Vehicle Recovery*, *Shipping Container Tracking*.

2.1.2 Comparação entre Lora/LoRaWan e outras abordagens de interligação para IoT

Nesta secção é feita uma análise de diferentes tecnologias de interligação em rede aplicáveis à área de IoT (LoRa, NB-IoT e Sigfox), onde são descritas as principais características de cada uma e são identificadas as suas vantagens e desvantagens. Finalmente, existe uma análise comparativa entre LoRa/LoRaWan e outras abordagens de interligação para IoT.

2.1.2.1 LoRa/LoRaWAN

Na secção 2.1.1 foi descrita a tecnologia LoRa de uma forma detalhada. As suas vantagens e desvantagens estão sintetizadas na Tabela 2-2.

Tabela 2-2 - Vantagens e desvantagens de LoRa/LoRaWAN [20]

LoRa/LoRaWAN	
Vantagens	**Desvantagens**
É apropriado para *single building applications*	Tem taxas de transmissão mais baixas do que o NB-IoT, no entanto mais baixo que Sigfox
É possível configurar e gerir a própria rede	Tem um tempo de latência maior que o NB-IoT, no entanto mais reduzido que Sigfox
LoRa é uma boa opção se for necessário bidirecionalidade	Requer um *gateway* (o que, em muitos casos, também pode ser uma vantagem)
Os dispositivos LoRa funcionam bem quando estão em movimento, o que os torna úteis para fazer o *tracking* em movimento, como encomendas	
Os dispositivos LoRa têm uma duração de bateria mais longa que os dispositivos NB-IoT, sendo muito semelhante a Sigfox	

2.1.2.2 NB-IoT

NB-IoT é um tipo de conectividade que permite massificar a utilização de IoT e torná-la acessível ao maior número possível de empresas, organizações e setores. É uma tecnologia LPWA (*Low Power Wide Area*) assente na rede móvel que, como complemento das redes 2G/3G e 4G, permite assegurar a conectividade das "coisas" com um baixo consumo de energia, aumentando a vida útil das baterias com toda a segurança e robustez da rede móvel [21]. Na Tabela 2-3 são apresentadas as vantagens e desvantagens desta rede IoT.

Tabela 2-3 - Vantagens e desvantagens de NB-IoT [20]

NB-IoT	
Vantagens	**Desvantagens**
A cobertura é muito boa. Os dispositivos NB-IoT dependem da cobertura 4G para que funcionem bem em ambientes internos e em áreas urbanas densas	É difícil implementar transferências *firmware-over-the-air* (FOTA) ou de ficheiros. Algumas das especificações de projeto para o NB-IoT fazem com que enviar grandes quantidades de dados para um dispositivo seja difícil
Tem tempos de resposta mais curtos do que LoRa e pode garantir uma melhor qualidade de serviço	Os *handoffs* de rede e torre são um problema. Portanto, o NB-IoT é mais adequado para ativos estáticos, como sensores em um local fixo, em vez de ativos de roaming

2.1.2.3 Sigfox

O Sigfox é um padrão proprietário de origem francesa. A sua principal vantagem é operar em frequências não licenciadas, com baixo consumo de energia e baixo preço - a assinatura anual para envio de duas mensagens por dia por dispositivo custa US$ 0,50. Cada mensagem pode ter até 12 bytes, o que limita o Sigfox a aplicações com baixo consumo de largura de banda, tal como dados captados por medidores em geral, como GPS, humidade, etc. [22]. Na Tabela 2-4 são enumeradas as vantagens e desvantagens da utilização desta rede IoT.

2.1.2.4 Análise comparativa

A Tabela 2-5 e a Figura 2-7 resumem as comparações realizadas entre as diferentes redes IoT analisadas. Após análise, podemos concluir que a utilização de LoRa é a mais recomendada quando se pretende um *deployment* muito flexível, pois permite ter instalações em vários tipos de rede e consegue cobrir tanto ambientes rurais como urbanos. É ainda vantajosa a utilização de LoRa em casos onde se pretenda que as baterias dos dispositivos tenham um elevado tempo de vida e onde haja a necessidade de ter uma boa eficiência de custos.

Para o caso de se pretender uma rede onde seja importante ter um tempo de resposta rápido, uma boa escalabilidade e os *payload* serem maiores, é aconselhável a utilização da rede NB-IoT.

No caso das necessidades se prenderem com uma maior cobertura e as mensagens serem enviadas com pouca frequência e com um *payload* muito reduzido, o melhor será optar pela rede Sigfox.

Tabela 2-4 - Vantagens e desvantagens de Sigfox [20]

Sigfox	
Vantagens	**Desvantagens**
Consome uma baixa quantidade de energia.	Não está implementado em todos os lugares, por isso não funciona para um grande número de casos de uso atualmente.
Funciona bem para dispositivos simples que transmitem com pouca frequência, porque envia quantidades muito pequenas de dados muito lentamente.	A comunicação é mais bem direcionada do *end-point* para a base station. Tem funcionalidade bidirecional, mas a sua capacidade da base station de volta ao ponto de extremidade é restrita.
Suporta uma ampla área de cobertura nas áreas em que está localizada.	A mobilidade é difícil com dispositivos Sigfox

Tabela 2-5 - Comparação entre LoRa, NB-IoT e Sigfox (adaptado de [23]**)**

	LoRa	NB-IoT	Sigfox
Tempo de latência	Maior que NB-IoT	Baixo	Alto
Bateria - tempo de vida	Elevada	Inferior a LoRa	Elevada
Necessidade de cobertura 4G	Não	Sim	Não
Deployment	Altamente flexível - podem ser instalados em redes públicas, privadas ou híbridas, internas ou externas.	Necessita de uma boa cobertura 4G para funcionar bem em ambientes fechados	Se bem localizada, permite uma boa área de cobertura
Bidirecional	Sim	Sim	Sim, mas com algumas limitações
Espectro	ISM	*Cellular Spectrum*	*Ultra Narrow Band* (UNB) - ISM
Mobilidade	Os dispositivos funcionam bem em movimento - muito bom para fazer o *tracking* em movimento	Possui algumas limitações - não é otimizada para efetuar o *tracking* de ativos	Difícil
Cobertura	5 km (urbano), 20 km (rural) No entanto a Semtech diz refere que pode cobrir até cerca de 48 km em ambientes rurais, consegue ter boa cobertura em ambientes urbanos e em ambientes internos [16].	1 km (urbano), 10 km (rural)	10 km (urbano), 40 km (rural)

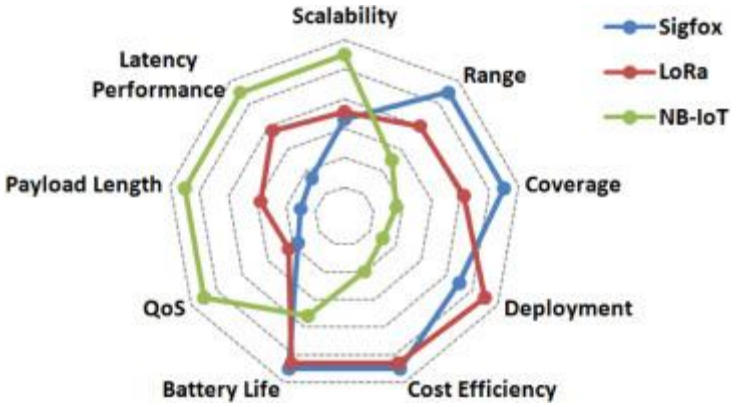

Figura 2-7 - Comparação entre LoRa, NB-IoT e Sigfox [23]

2.1.3 The Things Network

A The Things Network (TTN) é uma infraestrutura IoT aberta, sendo apoiada pelos seus membros, que contribuem colocando *gateways* ou executando servidores de rede. Esta é uma rede colaborativa segura e redundante. A TTN utiliza LoRaWAN (secção 2.1) e está a crescer em direção a uma rede global robusta e estável, fornecendo conectividade onde é necessária [24].

O *routing* dos dados IoT entre os dispositivos e as aplicações são realizados pelos sistemas de *backend* da TTN. Quando os próprios dispositivos suportam IP *stack*, os *gateways* apenas necessitam de encaminhar os pacotes para a internet [25]. No caso dos protocolos Non-IP – como é o caso de LoRaWAN - necessitam da existência de uma forma que permita o *routing* e processamento antes do envio e entrega das mensagens a uma aplicação [25]. A The Things Network está posicionada entre os *gateways* e as aplicações (Figura 2-8), tratando do processo de *routing* e processamento [25].

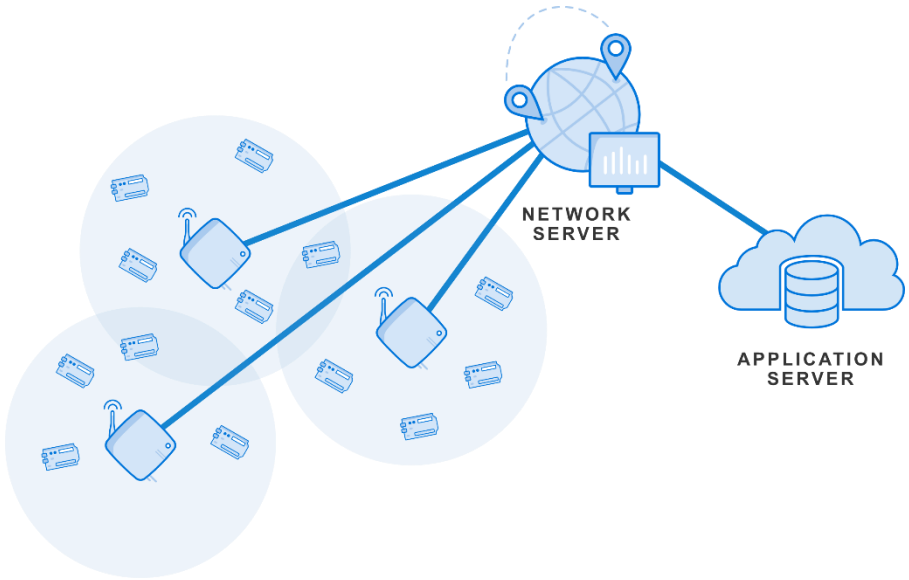

Figura 2-8 - Arquitetura genérica de rede da TTN [25]

A Figura 2-9 apresenta a arquitetura da rede TTN. De uma forma geral a TTN funciona da seguinte forma [25]:

- Os sensores transmitem mensagens LoRaWAN pelo protocolo de rádio LoRa;
- As mensagens são recebidas por vários *Gateways* que encaminha para o *backend*;
- Cada *Gateway* está ligado a um router que é responsável pela gestão do status e *schedule* das transmissões;
- Cada *Router* está ligado a um ou mais *Brokers* (são parte central da TTN);
- Os *Brokers* têm a responsabilidade de mapear um dispositivo para uma aplica, encaminhar mensagens de *uplink* para a aplicação correta e encaminhar mensagens de *downlink* para o *Router* correto;

- O *Network Server* é responsável pela parte da LoRaWAN;
- O *Handler* é responsável pela gestão de dados de uma ou mais aplicações e é onde os dados são encriptados ou desencriptados;
- O *Discovery Server* é parte fundamental na arquitetura descentralizada da TTN, sendo aqui que os *Routers*, os *Brokers* e os *Handlers* se "registam" [26].

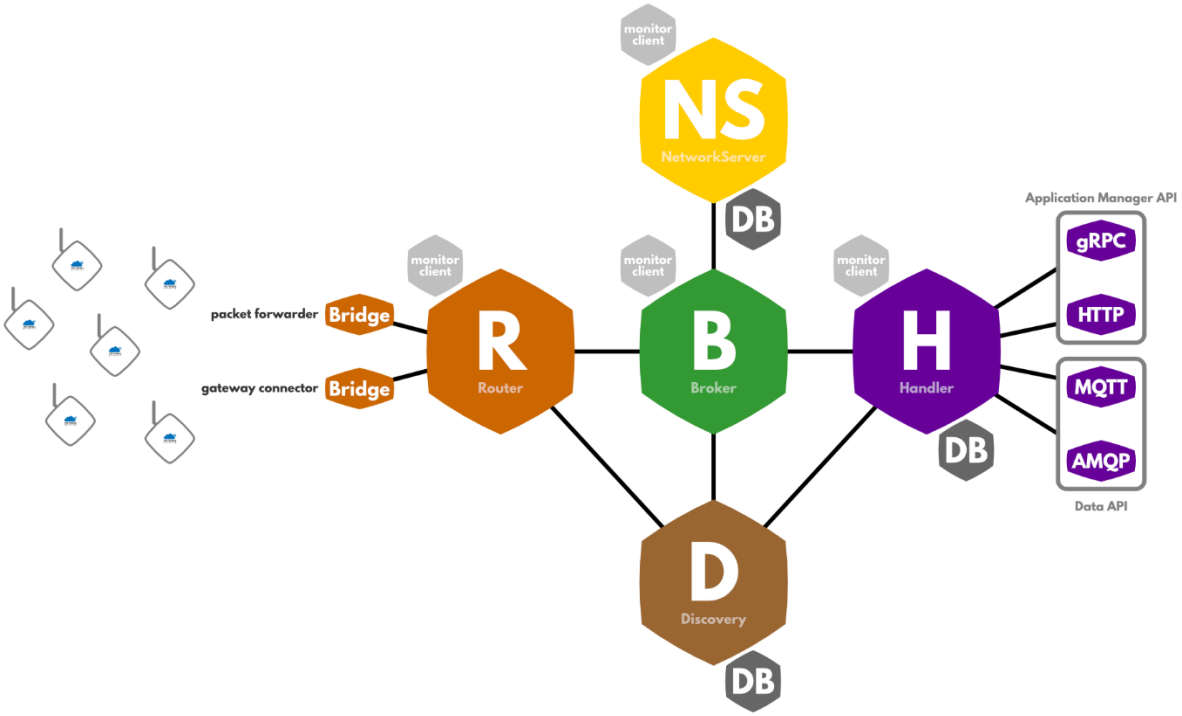

Figura 2-9 - Arquitetura da rede TTN [25]

2.1.4 Ameaças em ambientes IoT

Com o aumento de "coisas" ligadas a sistemas IoT (estima-se que em 2025 possam estar ligados cerca de 75 mil milhões de dispositivos – secção 2.1), as ameaças também podem aumentar [27] [28]. Foram lançados num passado recente alguns ataques cibernéticos sofisticados - como é o caso de Mirai (2016), Ransomware (2016), Shamoon-2 (2017) e DuQu-2 (2015) - na Industria Control System e em IoT que tornaram os protocolos IoT existentes ineficazes e provaram que os sistemas/dispositivos IoT são vulneráveis a este tipo de ataques [27]. Muitas das vezes este tipo de ataques terminam em pagamento de resgates, roubo de dados, falsificação de dados, entre outros [27].

De acordo com Jessica Groopman [29], é necessário fazer – de forma segura e eficiente - o *tracking* de todas as entidades, interações e atividades de cada "coisa" na rede. É ainda necessário a criação de uma arquitetura universal partilhada.

De acordo com N. Petracek [30], Gartner refere que três quartos de todos os projetos IoT levarão o dobro do tempo para serem implementados tal como planeado – existindo muitas falhas de segurança, o que poderá levar a um cenário muito complicado. Atualmente, há muitos ataques em dispositivos IoT e não tendem a reduzir, antes pelo contrário. Ameaças associadas aos dispositivos podem incluir:

- Ataques a dispositivos físicos (controlo de dispositivos não autorizados);
- Ataques de software (*malware*[2], como vírus ou *worms*[3]);
- Ataques de rede (assaltos *Denial of Services*[4], exploração de vulnerabilidades sem fio);
- Ataques criptográficos (*brute-force password cracking*[5], ataques *Man in the Middle*[6]).

Os dados devem ser recolhidos, transferidos e entregues com segurança entre os *stakeholders*. No entanto, em algumas arquiteturas de redes IoT não há confiança em algumas tecnologias e alguns desses problemas podem ser resolvidos através de uma solução descentralizada onde não existem essas vulnerabilidades.

A tecnologia Blockchain, descrita na próxima secção, surge como uma possível solução para resolver ou, pelo menos, mitigar alguns dos problemas de segurança acabados de mencionar. A sua integração com IoT, um dos focos do trabalho apresentado neste relatório, é descrita na Secção 2.3.

2.2 Blockchain

A tecnologia Blockchain consiste numa base de dados transparente e descentralizada que é replicada e partilhada entre os membros de uma rede, onde qualquer tipo de informação pode ficar registada e consultada (dependendo das características e permissões da rede). Esta tecnologia foi introduzida por Satoshi Nakamoto aquando da publicação de “Bitcoin: A Peer-to-Peer Electronic Cash System” [31],. Também foi apresentada a primeira rede de pagamento totalmente descentralizada - Bitcoin - baseada em Blockchain.

[2] *Malware* é um *software* malicioso que tenta invadir, danificar ou incapacitar computadores, sistemas, redes ou outros dispositivos – assumindo o controlo parcial das operações de um dispositivo [81].
[3] *Worms* é um tipo de *malware* que se replica com o intuito de se espalhar para outros computadores. A sua propagação ocorre através da rede a que o dispositivo esteja ligado, sites infetados, anexos de e-mail, entre outros [82].
[4] *Deniel of Service* é um tipo de ataque que tenta tornar os recursos de um sistema indisponíveis para os utilizadores através da sua sobrecarga [83].
[5] *Brute-force password cracking* é um método utilizado para tentar de adivinhar informações de *login*, chaves de encriptação ou encontrar uma página web oculta através de tentativa erro (conjugação de várias combinações possíveis até encontrar a combinação certa) [84].
[6] *Man in the Middle* é uma forma de ataque onde os dados trocados entre o utilizador e outro serviço são intercetados (por exemplo entre o utilizador e o seu banco online) [85].

O Blockchain, enquanto base de dados, regista transações num registo imutável, o que significa que, uma vez criado, não pode ser apagado ou alterado. Note-se que, apesar de atualmente existirem centenas de redes Blockchain, ainda não existem standards tecnológicos, havendo ainda um baixo nível de interoperabilidade entre elas. A maioria dos países não possui uma legislação para regulamentar esta tecnologia, o que seria fundamental para legitimá-la, construir confiança nos *stakeholders* e, assim, contribuir para a sua aceitação e desenvolvimento [32].

2.2.1 Principais fundamentos

A tecnologia Blockchain funciona com um conjunto de blocos encadeados seguros, sendo os mesmos imutáveis, ou seja, não podem ser apagados ou alterados. Cada bloco possui uma lista de transações e um apontador (um código de *hash* – secção 2.2.2) para o bloco anterior. Isto apenas não acontece no primeiro bloco que é chamado de *génesis* e que é comum a todos os blocos da rede.

Um conjunto de nós opera na mesma rede Blockchain (formando uma rede *peer-to-peer*) através da cópia da cadeia de blocos que cada um possui. A Figura 2-10 ilustra um pedido de transação a uma rede Blockchain e que é genericamente composta pelos seguintes processos:

- Os utilizadores fazem um pedido de transação à Blockchain. Uma transação poderá envolver criptomoedas, contratos, registos ou outro tipo de informação;
- O pedido de transação é enviado para todos os nós da rede;
- As transações são validadas de forma a verificar os recursos utilizados (por exemplo garantir que um utilizador não está a tentar transferir criptomoedas que não tem na sua carteira), a identidade e o estado do utilizador – ver secção 2.2.3;
- As transações recolhidas e validadas pela rede, durante um intervalo de tempo acordado (no caso da Bitcoin são cerca de dez minutos), são ordenadas e empacotadas num bloco candidato com registo de data e hora (chamado processo de *mining*[7]). O nó de *mining* que "pensa" ter resolvido o problema inerente ao processo de *mining* volta a transmiti-lo de volta à rede;
- Após a validação da solução apresentada pelo *miner*[8] é adicionado um novo bloco à cadeia, criando assim uma corrente de blocos. Todos os blocos estão ligados através de uma ligação criptográfica o que garante que esta corrente seja inalterável.

[7] *Minning* é o processo utilizado para a criação de novos blocos na Blockchain através da resolução de um "puzzle criptográfico" (ver secção 2.2.3).
[8] *Miner* é o tipo de nó da rede Blockchain responsável pelo processo de *minning* (ver secção 2.2.3 e 2.2.4).

O processo é mais detalhado nas restantes secções deste capítulo, nomeadamente na secção 2.2.2 (Proof-of-Work) e 2.2.3 (*Mining*).

2.2.2 Proof-of-Work

Proof-of-Work (PoW) consiste na resolução do "*puzzle* criptográfico" que os *miners* necessitam de realizar para poderem registar um novo bloco de transações. Para efetuarem este registo é necessário os *miners* apresentarem aos restantes nós da rede que efetuaram um esforço computacional suficiente (através da "prova de trabalho" que é a resolução do *puzzle* criptográfico), isto é, que consumiram recursos económicos próprios - por exemplo, energia elétrica. Na maioria dos casos, os *miners* são recompensados - por exemplo, no caso da Bitcoin, são atribuídos novos bitcoins [32].

Para garantir a eficácia da PoW, é necessário que a mesma tenha algumas propriedades, tais como [33]:

1. *Puzzle* criptográfico que obrigue à execução de uma tarefa computacional exigente (descoberta do *puzzle* criptográfico) - obrigar a consumo de recursos económicos;
2. Grau de dificuldade parametrizável - aumentar quando a capacidade computacional aumentar e reduzir quando a capacidade computacional também reduzir;
3. A validação do *puzzle* criptográfico tem de ser simples – por exemplo, tem de ser semelhante ao Sudoku, ou seja, difícil de resolver e fácil de verificar se está resolvido

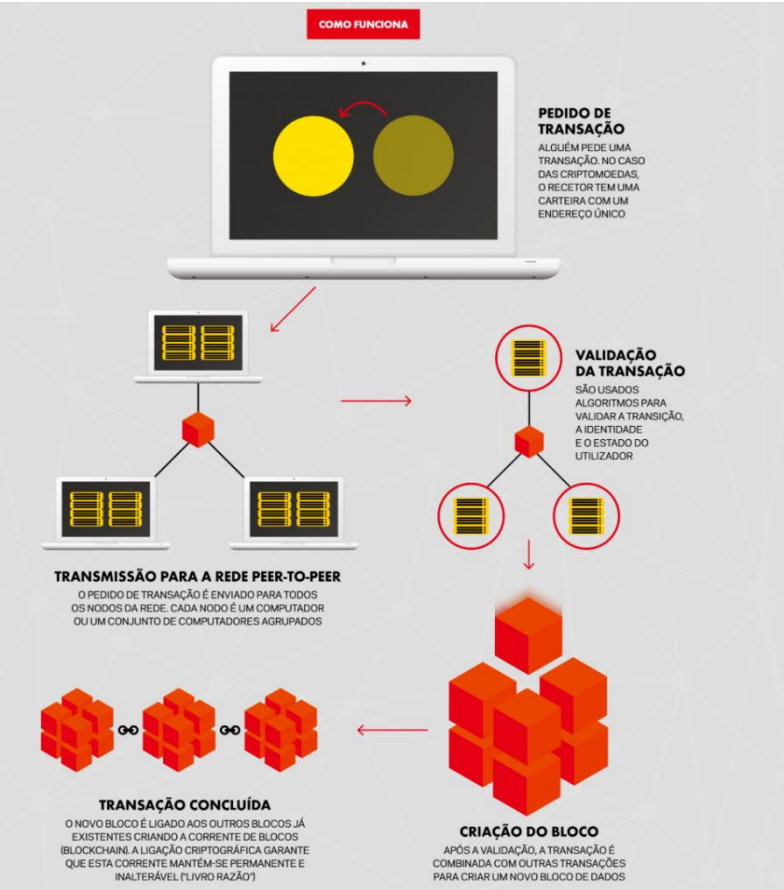

Figura 2-10 - Pedido de transação a rede Blockchain [91]

Na *memory pool* local (ver secção 2.2.3), os *miners* constroem um novo bloco que é composto por:

- Índice do novo bloco;
- Código de *hash* para o bloco anterior;
- *Timestamp*;
- Conjunto de transações que pretendem registar na base de dados de transações;
- *Nonce*[9] - valor de 32 bits que pode variar.

A Figura 2-11 ilustra a composição de um conjunto de blocos na Blockchain.

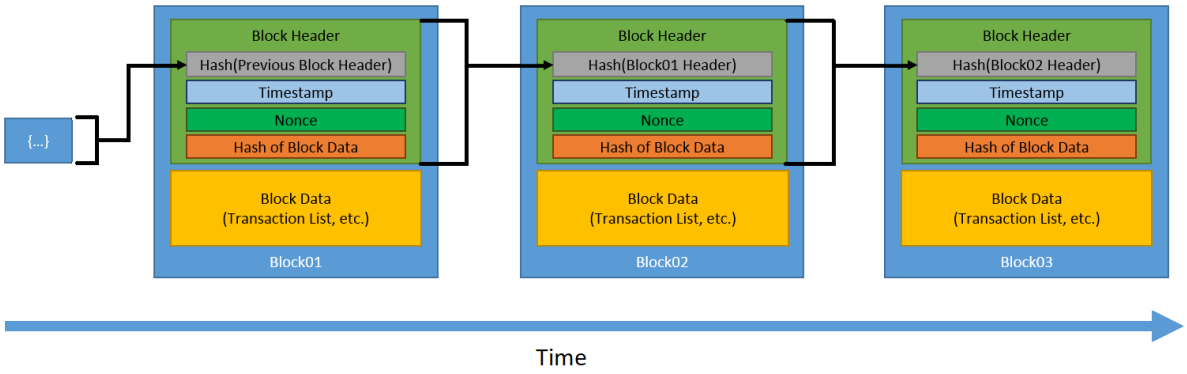

Figura 2-11 - Composição de um bloco na cadeia de Blockchain [92]

O *miner* para descobrir o *puzzle* criptográfico tem de calcular uma *hash* (por exemplo na Bitcoin é utilizado SHA-256) que cumpra um conjunto de restrições impostas pelas regras da rede (de forma a criar uma competição entre os *miners* para determinar quem adiciona o próximo bloco de transações) – normalmente são um conjunto '0' pelos quais a *hash* tem de começar [34]. Para conseguir descobrir o *puzzle* criptográfico/calcular uma *hash* válida, o *miner* tem de adicionar um número arbitrário ao bloco (*nonce*) de forma a que este inicie com um conjunto de '0' definidos pela Blockchain. A dificuldade para a criação de um novo bloco poderá variar, quantos menos '0' a *hash* tiver que ter no início, mais fácil será. Desta forma, poderá ainda existir um conjunto diferente de *nonces* que sejam válidos, pois a restrição é o número de '0' com que começa a *hash* calculada.

[9] Abreviatura para "number only used once" [86]. Valor incluído num bloco em Blockchain e pelo qual os *miners* tentam encontrar uma solução para concluírem o processo de *mining* e publicarem um bloco na Blockchain.

O *miner* apenas consegue encontrar a solução através de um processo tentativa-erro, o que leva a que o mesmo tenha um elevado esforço computacional e, desta forma, consuma recursos económicos.

A Figura 2-12 apresenta um caso onde um bloco que tem o seu número ("Block N"), um conjunto de transações, a *hash* para o bloco anterior ("Hash of Block M") e um *nonce*, gerou

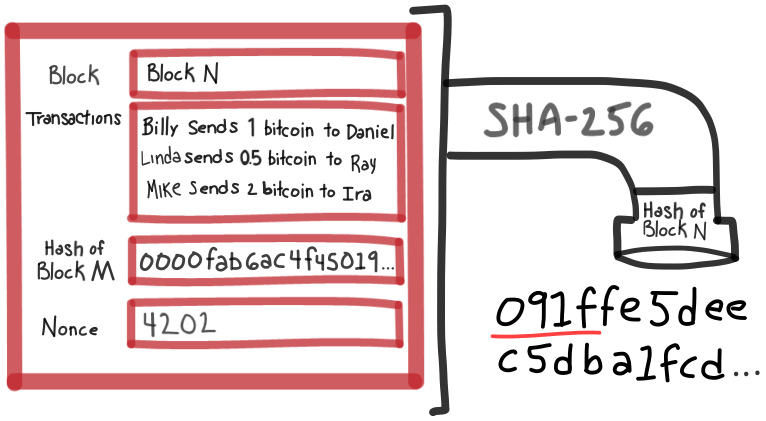

Figura 2-12 - Tentativa de descoberta de *nonce* inválida [34]

uma *hash* inválida, pois não cumpre com o requisito do número de '0' iniciais (neste caso 4 zeros em hexadecimal). Este resultado significa que o *nonce* é inválido, tendo assim que continuar a tentar descobri-lo [34]. Neste exemplo, o *miner* pegou no bloco de novas transações, mais a *hash* do bloco anterior, e adicionou o *nonce* 4202 a esses dados (essa suposição de *nonces* costuma ser chamada de *hashing*) [34]. Em seguida, ele fez o *hash* de todos os dados através de SHA-256 e chegou ao *hash* resultante.

A Figura 2-13 apresenta um caso onde o *miner* conseguiu identificar o *nonce* correto para o bloco anteriormente ilustrado. Tal como referido anteriormente este é um processo complexo, e um *miner* pode necessitar de fazer milhões de tentativas até descobrir o *nonce* correto.

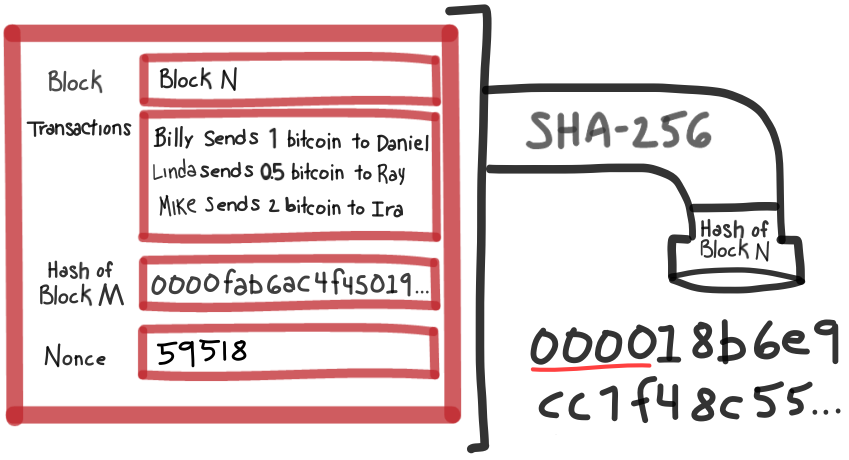

Figura 2-13 - Descoberta de um *nonce* válido para um bloco [34]

Quando é encontrada uma solução válida para o *nonce*, o *miner* regista o novo bloco na sua base de dados de transações local e envia-o para a restante rede juntamente com o *nonce* calculado. De seguida, os restantes nós validam a solução proposta. Esta solução

não obriga a um elevado esforço computacional, pois apenas têm de verificar se aquele *nonce* respeita as características impostas pela rede. Após a validação do novo bloco, o mesmo é registado nas bases de dados locais dos *miners*, confirmando desta forma as novas transações na rede.

Através deste processo, os utilizadores de uma rede Blockchain podem confiar no conteúdo da base de dados de transações e, por exemplo, fazerem pagamentos sem uma entidade central e desconhecendo-se mutuamente. Por esta razão, a tecnologia Blockchain pode ser vista como uma inovação tecnológica para a mecanização da confiança.

2.2.3 Mining

De entre os *full nodes*[10] existe um tipo especial de nós, os *miners*. que têm um papel importante nos processos de *mining*, validação e registo de novas transações na base de dados Blockchain.

Cada bloco possui um código de *hash* denominado *nonce*. O processo de *mining* corresponde à tentativa de os *miners* encontrarem um código de *hash* que dê resposta àquele *nonce* (o designado enigma criptográfico) – processo computacionalmente complexo e que requer um elevado consumo de recursos. Esta é uma tarefa computacionalmente muito complexa e exige um grande esforço computacional por parte dos *miners*.

Os *miners* ao terem conhecimento da existência de uma nova transação, realizam um conjunto de validações tais como: verificar se todas os *inputs* especificados não foram consumidos em outras transações anteriores e verificam ainda se as assinaturas digitais são válidas [32]. Caso a transação seja válida, a mesma é enviada para os restantes nós da rede e é publicada na rede.

Após a sua publicação na rede, os *miners* que validam a nova transação adicionam-na a uma *memory pool* local de novas transações. A partir dessa *memory pool*, os *miners* juntam um conjunto de novas transações num bloco de transações e "competem" entre si pelo registo desse bloco na base de dados de transações (Figura 2-14) [32]. A "competição" consiste na tentativa de resolução do *puzzle* criptográfico – *nonce* [32].

O primeiro *miner* a resolvê-lo publica o novo bloco e a solução do problema na rede. Os outros *miners* depois de validarem e aceitarem a solução (verificando se o *nonce* apresentado por aquele *miner* dá resposta às exigências da rede – ver secção 2.2.2), registam esse bloco na sua base de dados de transações – caso mais de 50% dos *miners* aceitem a solução, há portanto, um consenso [32].

[10] Full nodes são tipos de nós da rede Blockchain que entram no processo de verificação de blocos e transações seguindo as regras de um algoritmo de consenso (ver secção 2.2.4).

Se a solução apresentada for inválida - ataque ou anomalia no funcionamento do *miner* - os restantes *miners* rejeitam o novo bloco e o mesmo não é guardado na sua base de dados de transações. Este *miner* terá de reverter a sua base de dados para o estado imediatamente anterior ao registo do novo bloco e, caso não obedeça e continue a registar blocos na sua base de dados, todos os seus blocos serão rejeitados por não obedecerem ao encadeamento criptográfico correto. Desta forma, é criada uma visão única para todos os participantes da rede e existe uma sincronização das bases de dados que contêm registo de transações [32].

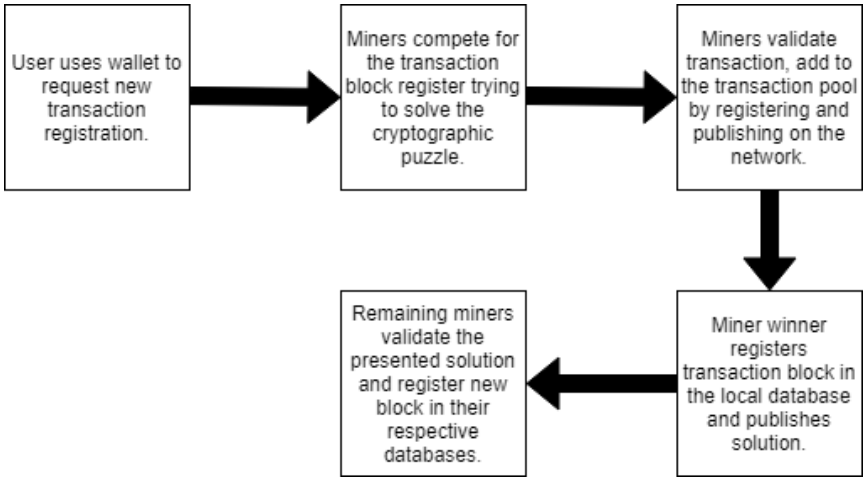

Figura 2-14 - Processo de *mining* – adaptado [32]

2.2.4 Tipos de nós nas redes Blockchain

Os nós de uma rede Blockchain são responsáveis por agir como um ponto de comunicação e podem ter diferentes funções.

Quaisquer computadores ou dispositivos que se conectem a uma rede Blockchain podem ser considerados nós no sentido em que comunicam entre si de alguma forma. Esses nós também são capazes de transmitir informações a respeito das transações e blocos dentro de uma rede de computadores ao usar o protocolo *peer-to-peer* da Blockchain. Cada computador é definido de acordo com sua função particular, pelo que existem diferentes tipos de nós dentro de uma Blockchain [35]:

- **Full nodes (Full Validating Nodes)** - São indispensáveis à rede porque são os responsáveis pela segurança. Estes nós entram no processo de verificação de blocos e transações seguindo as regras de um algoritmo de consenso. Também são capazes de retransmitir novos blocos e transações para a Blockchain. Este tipo de nó ao juntar-se à rede, por norma faz o download de uma cópia da Blockchain (onde estão presentes os blocos e transações). No entanto, este não é um requisito para ser *full node*, pois pode ser utilizada uma cópia reduzida da Blockchain, tendo

apenas os cabeçalhos de blocos durante o processo de sincronização inicial e, em seguida, solicitam transações conforme necessário.

- **Super Nodes (Listening Nodes)** - É um *full node* que é visível publicamente e que comunica com qualquer outro nó que entre em contacto com ele. É um ponto de redistribuição que pode funcionar como fonte de informação e ponte de comunicação em simultâneo. Este tem várias conexões estabelecidas transmitindo as informações e transações da Blockchain para vários outros nós por forma a sincronizar estes dados com todos os elementos.
- **Miners** - É o tipo de nó da rede Blockchain responsável pelo processo de *minning*. Um *miner* pode escolher trabalhar sozinho (*solo miner*) ou em grupos conhecidos como *miner pools*. Enquanto os *solo miners* usam uma cópia própria da Blockchain, os *pool miners* trabalham juntos, cada um contribuindo com seu poder computacional individual (*hashpower*). Numa *mining pool* [11], só o respetivo administrador é que precisa de atender ao requisito de correr um *full node* – conhecido como *pool miner's full node*.
- **Lightweight ou SPV (Simplified Payment Verification)** – Por exemplo no caso da Bitcoin muitos clientes são projetados para funcionar em *smartphones* e máquinas com poucos recursos e armazenamento. Para esses dispositivos, um sistema SPV é executado para permitir que operem sem o conteúdo total da Blockchain armazenado localmente [36].
  Os SPV fazem o uso da Blockchain, mas não agem como *full nodes*. Portanto, SPV não contribuem para a segurança da rede simplesmente porque não têm uma cópia da Blockchain e não fazem parte do processo de verificação e validação de transações. De uma forma geral, estes nós têm uma cadeia de blocos sem transações e isso torna possível ter uma quantidade de dados resultante armazenada localmente significativamente menor [36].

  Um SPV exige que seja obtida apenas uma parte da Blockchain. Conecta-se a *full nodes* e usa filtros para garantir que recebe apenas as transações necessárias e relevantes para as operações que pretende realizar. Estes obtêm apenas os cabeçalhos de blocos durante o processo de sincronização inicial e, em seguida, solicita transações aos *full nodes* conforme necessário [37].

  Resumidamente, um SPV é a forma pela qual determinado participante consegue verificar se algumas transações foram ou não incluídas num bloco sem precisar de ter todas as informações do mesmo. Logo, clientes SPV dependem da informação

[11] Uma *minning pool* funciona como um coordenador de *pool members*. Entre as funções a desempenhar está a gestão de *hashes* dos *pool members,* procurando recompensas através de esforços conjuntos de poder de processamento disponível. O trabalho de cada *pool member* é registado e são atribuídas quotas a cada membro em proporção ao trabalho realizado após uma verificação correta.

fornecida por outros *full nodes* (*super nodes*), funcionando como terminais de comunicação, sendo usados, por exemplo, por várias carteiras de *criptomoedas*.

2.2.5 Proof-of-Stake

Proof-of-Stake (PoS) é um dos protocolos de consenso utilizados na tecnologia Blockchain e é uma alternativa ao Proof-of-Work (PoW). Este é utilizado por redes Blockchain para validar os blocos. O sistema foi sugerido pela primeira vez em 2011 e a primeira criptomoeda a implementá-lo foi a Peercoin em 2012. As principais vantagens de PoS em relação a PoW são a eficiência energética e a segurança. Este protocolo de consenso terá de ter sempre uma criptomoeda associada (mesmo que não tenha qualquer valor real), pois o poder de *mining* de um *miner* é proporcional ao número de moedas que este tem em sua posse.

Ao contrário da PoW, onde o algoritmo recompensa os *miners* que resolvem problemas matemáticos com o objetivo de validar transações e criar novos blocos, com PoS o criador de um novo bloco é escolhido de forma determinística – o poder de mineração é proporcional ao número de moedas que um *miner* tem em seu poder. Desta forma, em vez de utilizar energia para responder aos enigmas de PoW, um *miner* de PoS está limitado a minerar uma percentagem das transações que reflita sua participação acionária [38]. Por exemplo, um *miner* que possui 3% de Bitcoin disponível pode teoricamente minerar apenas 3% dos blocos [38].

Em PoS os blocos são denominados por *forged* ou *minted*, não são *mined*. Quem valida as transações cria novos blocos são chamados de *forgers* [39].

Através de PoS, as criptomoedas são criadas no lançamento da moeda e o seu número é fixo. Desta forma, os *forgers* são recompensados em taxas de transação em vez de serem em criptomoedas (como acontece com os *miners* na Bitcoin, por exemplo) [38].

Este sistema não permite lidar com a distribuição inicial de moedas na fase de fundação da criptomoeda, assim, as criptomoedas iniciais podem ser vendidas ou a rede começa com o sistema de PoW e depois muda para o PoS.

Os *forgers* para entrarem na “discussão” de quem irá ser o *forger* da próxima transação têm de arriscar uma determinada quantia de criptomoedas que serão perdidas caso validem transações maliciosas [40].

Existem dois métodos para a seleção de quem irá ser o forger da próxima transação [41]:

- Randomized block selection – É utilizada uma fórmula que procura o *forger* com a combinação do menor valor de hash e o tamanho da sua aposta. Como o valor das apostas é público, cada nó geralmente é capaz de prever quem será selecionado;
- Coin Age based selection – O próximo *forger* é selecionado com base na 'idade da moeda' da aposta que o potencial forger apostou. A idade da moeda é calculada por: “número de dias que as moedas foram mantidas como aposta” a multiplicar por “número de moedas que estão a ser apostadas”. As moedas devem ter sido

guardadas no por mínimo 30 dias antes de poderem competir por um bloco. Quem apostou conjuntos de moedas maiores e mais antigos têm uma probabilidade maior de serem designados como *forgers* do próximo bloco. Depois *forger* ter utilizado as moedas num *forged* de um bloco, a idade da moeda volta a zero e deve esperar pelo menos 30 dias novamente antes de assinar outro bloco.

A principal vantagem de PoS em relação a PoW é que a solução PoS consome consideravelmente menos energia e, como resultado, é mais económica. Por exemplo, a Bitcoin, que utiliza PoW, consome mais energia anualmente do que a Suíça ou a República Checa (Figura 2-15) – ou seja, é insustentável e ineficaz.

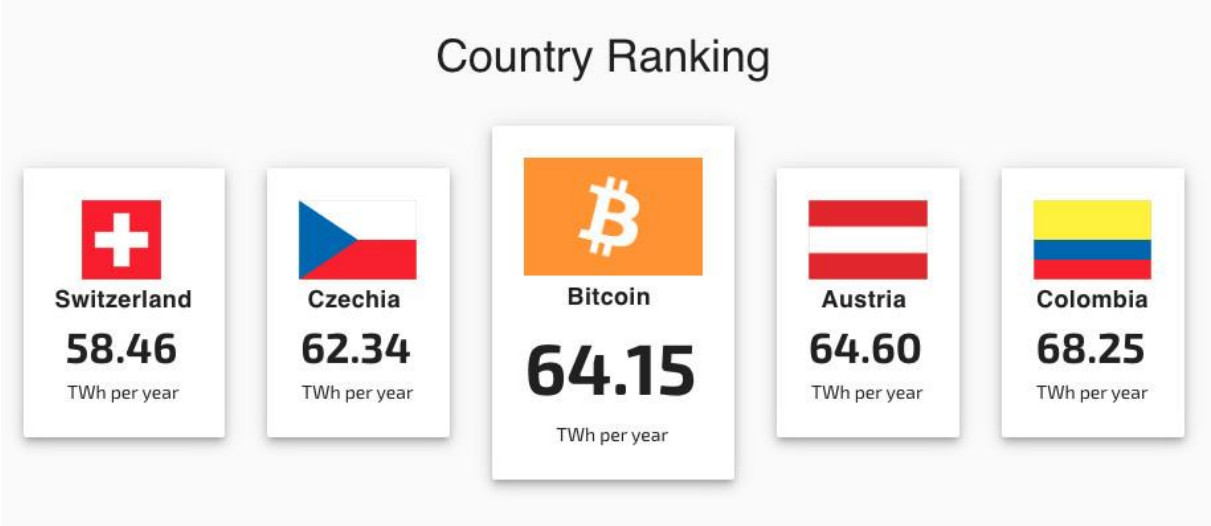

Figura 2-15 - Consumo energético anual de Bitcoin em comparação com alguns países [93]

A este nível, o protocolo de consenso PoS pode ser considerado superior, pois necessita de muito menos eletricidade. Este também é mais económico, pois não é necessário despender tantas moedas como forma de incentivo aos *miners* - logo permite que o preço de uma moeda seja mais estável.

Ao nível de segurança, é semelhante à Proof-of-Work, pois no caso de PoW é preciso ter pelo menos 51% da rede a seu favor, em PoS é necessário possuir pelo menos 51% das moedas da rede para obter o consenso e, assim, poder registar uma nova transação. Para além de ficar extremamente caro e difícil possuir este valor, os participantes da rede que possuem elevadas quantidades da moeda também não querem que a mesma seja atacada, pois o seu valor iria reduzir bastante, sendo do interesse destes utilizadores preservar a segurança na rede [42].

A Figura 2-16 ilustra as diferenças existentes entre os métodos de consenso PoS e PoW.

De uma forma geral, pode afirmar-se que o mecanismo de consenso PoS cumpre com o que é pretendido e é um sistema robusto. No entanto, as empresas ainda o melhoraram e avançaram para o Delegated Proof of Stake (DPoS).

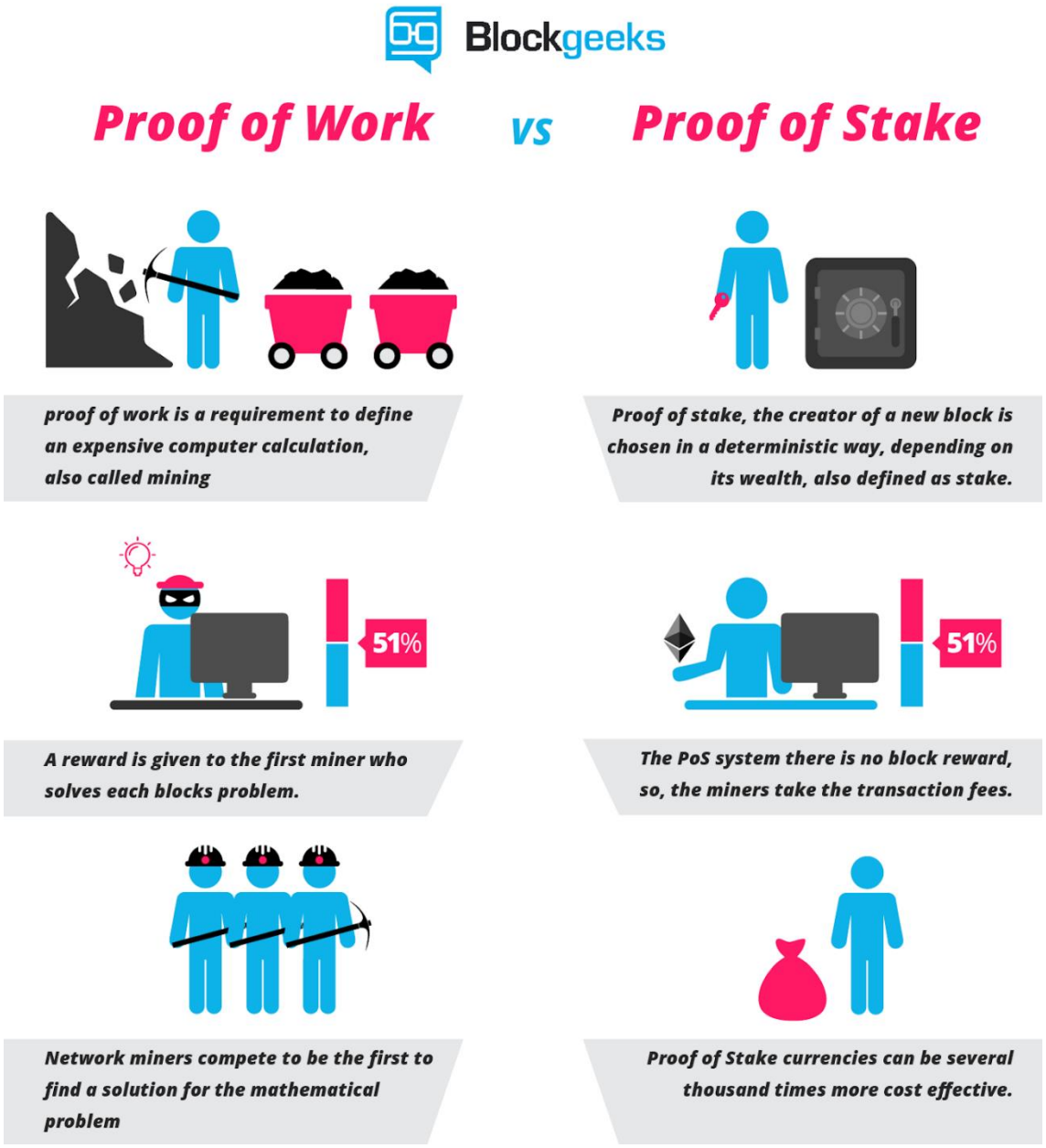

Figura 2-16 - Diferenças entre Proof-of-Work e Proof-of-Stake [39]

2.2.6 Delegated Proof-of-Stake

Delegated Proof-of-Stake (DPoS) é uma evolução do conceito de PoS, em que os utilizadores da rede votam e elegem delegados para validar o próximo bloco [43]. Este é um mecanismo onde um número fixo de entidades (chamados *block producers* ou *witnesses*) eleitas são selecionadas para criar blocos [44].

Uma rede Blockchain baseada em DPoS conta com um sistema de votação onde as partes envolvidas podem votar em alguns delegados que protegerão a rede em seu nome [45]. Os delegados são responsáveis por obter consenso durante o processo de gerar e validar

novos blocos. O poder do voto é proporcional ao número de moedas que cada utilizador detém. O sistema de votação varia de projeto para projeto, mas em geral cada delegado apresenta uma proposta individual ao solicitar votos. Geralmente as recompensas recolhidas pelos delegados são partilhadas com os seus eleitores [45].

Este processo tem dois tipos de entidades importantes para o funcionamento da rede Blockchain [46]:

- ***Block producers* ou *witnesses*** são os responsáveis pela criação e assinatura de novos blocos. Eles são limitados em número e são eleitos pelos eleitores (pelos *stakeholders* da rede - Figura 2-17).
- **Validadores de blocos** são os *full nodes* que verificam se os blocos criados pelos produtores de bloco seguem as regras de consenso. Qualquer utilizador pode executar um validador de bloco e verificar a rede.

De uma forma geral o processo de votação de *witness* decorre da seguinte forma (Figura 2-17):

- Os nós expressam o seu interesse em se tornar *witness*, fazendo contribuições positivas para a rede e engajando a comunidade;
    - As *witness* têm uma responsabilidade enorme e são cruciais para a integridade da rede Blockchain, portanto, elas devem ter um histórico comprovado antes de serem consideradas. Aqueles que desejam se tornar *witness* podem aumentar sua reputação fazendo contribuições positivas para a rede - isso pode vir na forma de recrutamento de novos membros para a comunidade, etc. – tal como uma a campanha de umas eleições;
- Os *stakeholders* da rede (outros nós que não são candidatos, por exemplo – no entanto um candidato a *witness* pode votar nele próprio) dão seus *tokens* como votos para *witness* – quando mais moedas tiverem, mais tokens terão. E quantos mais *tokens* têm, maior é a sua influência no voto;
- É encontrado o ranking de nós com mais votos (um maior número de *tokens* associados a eles). Os N (geralmente N ronda entre os 21 e os 100 [47]) primeiros tornar-se-ão membros do painel de *witness* eleitas.

Uma ronda em um Blockchain DPoS com N *block producers*/*witnesses* de blocos segue uma ordem *round robin* da seguinte forma [48]:

- N *Block producers* são eleitos a partir do pool de candidatos;
- O k-ésimo *block producer* assina o k-ésimo bloco, até k = N;
- Um bloco é finalizado quando é votado por (2/3 + 1) dos *block producers*.

Este é um algoritmo de consenso que mantém um acordo irrefutável sobre a verdade em toda a rede, validando transações e agindo como uma forma de democracia digital.

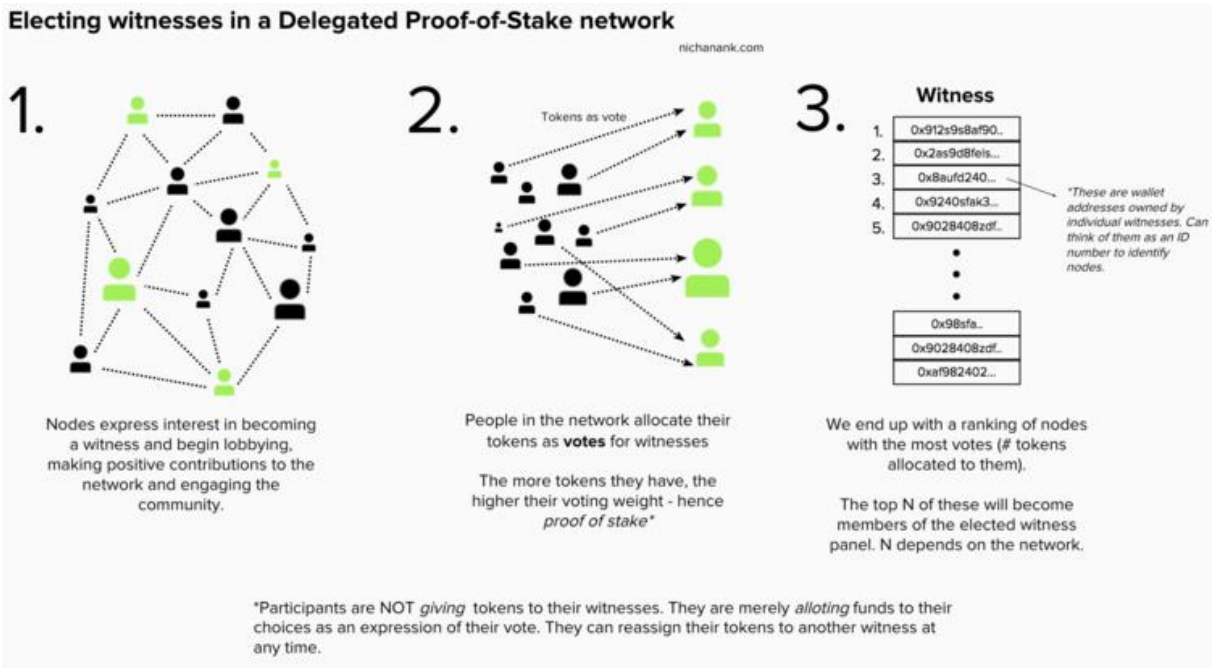

Figura 2-17 - Processo de eleição de *witness* em DPoS [94]

A Figura 2-18 apresenta a comparação entre os mecanismos analisados ao longo do documento (PoW, PoS e DPoS).

Como podemos verificar PoW é o mecanismo de consenso que envolve uma maior complexidade, pois tem de ser descoberto um *puzzle* criptográfico para poder completar o processo de *mining* e adicionar um novo bloco. Desta forma, isto leva a um consumo muito elevado de energia e o tempo gasto no processo também é elevado demorando cerca de 10 minutos.

PoS surgiu de forma a resolver os principais problemas trazidos por PoW, havendo uma redução na dificuldade de acrescentar um novo bloco à rede e assim conseguiu reduzir bastante o consumo de energia e a velocidade com que é acrescentado um bloco.

DPoS veio ainda melhorar aquilo que o PoS tinha conseguido e conseguiu distribuir o consumo de energia, pois todos os Block Producers trabalham na mesma quantidade de blocos num período específico. Dadas as suas características, veio ainda reduzir o tempo de processamento de um novo bloco, pois como eles são votados pela comunidade, eles provam o seu valor mantendo a produção de blocos o mais rápida possível.

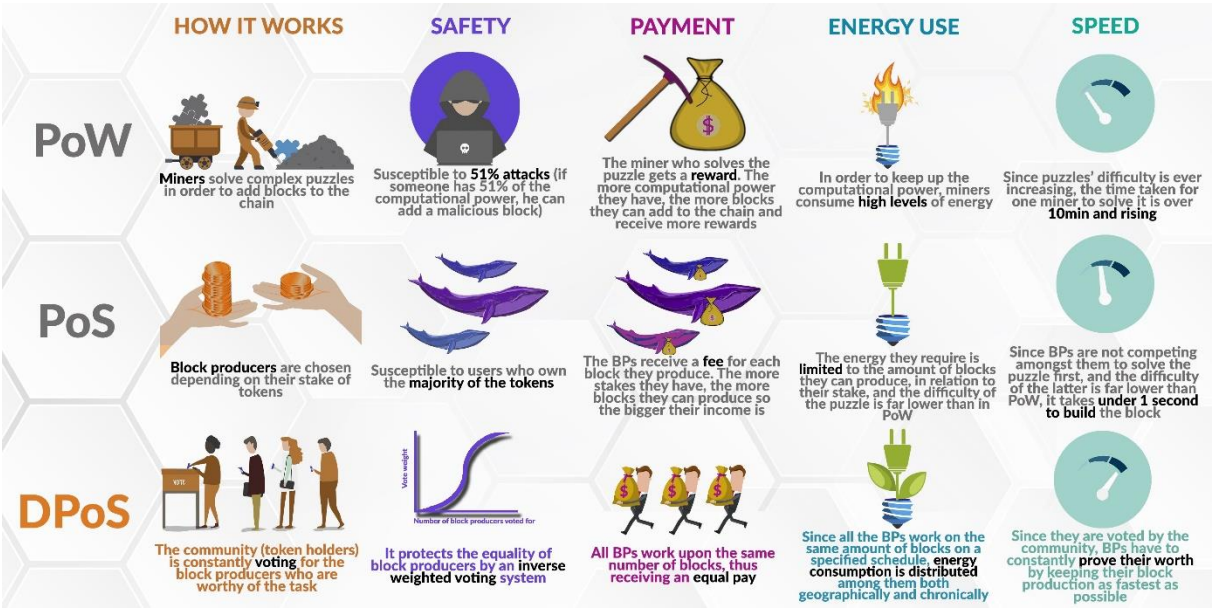

Figura 2-18 - Comparação entre os mecanismos de consenso [95]

2.2.7 Smart contracts

O conceito de *smart contract* foi apresentado por Nick Szabo num artigo intitulado "Smart Contracts: Building Blocks for Digital Free Markets" [49]. Este corresponde a um protocolo computacional que impõe a execução dos termos dos contratos digitalmente especificado pelo *software* quando existem condições pré-acordadas. Dessa forma, é possível reduzir custos de transações e automatizar processos [32] [50] [51]. A tecnologia Blockchain tem vindo a facilitar a implementação de *smart contracts*, pois existe uma visão comum dos eventos entre os participantes da rede. Desta forma, permite um contexto de execução do *smart contract* que é aceite por todas as partes envolvidas no contrato [32].

A rede Bitcoin não suporta *smart contracts* porque a especificação funcional de um contrato não podes ser totalmente suportada pelo tipo de script usado nesta rede (não está baseado numa linguagem computacional universal). Este problema foi resolvido pela segunda geração de redes Blockchain, onde foi adicionado o suporte à uma linguagem computacional universal designada Solidity descrita na Secção 2.2.9.3 [32]. Um *smart contract* é um contrato comparável a um contrato convencional de papel, ou seja, com cláusulas e outorgantes (definidos e aplicados por *software*), mas que é armazenado em Blockchain e, desta forma, é inviolável, auto-executável e automaticamente aplicável.

Os *smart contracts* têm muito potencial, essencialmente em aplicações de pagamento automático (compras, estacionamento, etc.). Também se aplica à gestão de direitos digitais, serviços financeiros e pagamento automático de sinistros [27].

Como podemos verificar na Figura 2-19 existem duas pessoas que pretendem celebrar um contrato (neste caso uma pretende vender a sua casa e outra pretende comprar uma casa). Ambos os intervenientes acordam um em comprar e ou em vender a casa e o *smart contract* recebe os dados necessários para proceder à celebração do contrato. Desta forma, é

automaticamente celebrado o contrato e o comprador fica com provas irrefutáveis que agora a casa é sua, bem como o vendedor também fica com a prova de venda da sua casa. Assim, foram eliminados um conjunto de intermediários que num contrato tradicional são necessários.

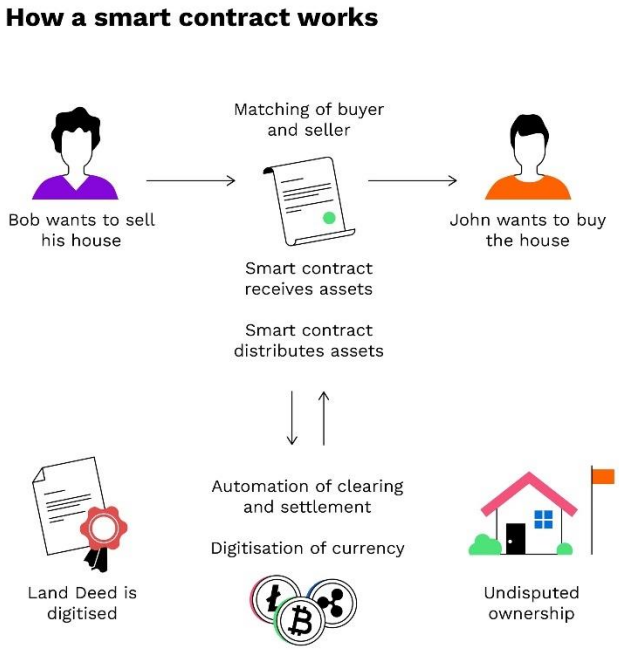

Figura 2-19 - Exemplo de utilização de um *smart contract* [96]

2.2.8 Bitcoin

Bitcoin é uma moeda completamente digital que funciona de forma consensual. Esta é a primeira rede de pagamento descentralizada (*peer-to-peer*), criando desta forma uma nova forma de pagamento. Na bitcoin, são os utilizadores que fazem a gestão do sistema não necessitando de entidades centrais nem de intermediários. O utilizador comum pode visualizar a Bitcoin como dinheiro para a Internet.

Em 31 de outubro de 2008 foi anunciado e disponibilizado por Satoshi Nakamoto, no site bitcoin.org, o artigo “Bitcoin: A Peer-to-Peer Electronic Cash System”, onde foi apresentada a solução para o problema de *double spending*[12].

Contudo, Satoshi Nakamoto não se limitou a apenas apresentar a solução para o problema do *double spending* e trabalhou para a sua implementação prática, tendo ainda, no ano de

[12] *Double spending* é uma falha existente em algumas criptomoedas que ocorre quando um utilizador consegue gastar as mesmas moedas digitais mais de uma vez [87].

2008, registado o projeto Bitcoin em sourceforge.net. A 3 de janeiro de 2009, foi feito o primeiro registo na base de dados da rede Bitcoin – o Genesis block - e, de seguida, foi disponibilizado o primeiro programa cliente onde é possível interagir com a rede e efetuar pagamentos.

A Bitcoin teve uma elevada valorização muito devido a [32]:

- A tecnologia Blockchain, que suporta a Bitcoin, tem demonstrado ser uma tecnologia segura e isso tem contribuído para o aumento da confiança nesta tecnologia e para a sua adoção;
- O investimento crescente que se tem observado nesta tecnologia, por parte de investidores de capital de risco e de grandes empresas dos mais variados setores, tem credibilizado a tecnologia Blockchain;
- A instabilidade, a incerteza e perseguição política que existem em alguns países têm motivado os cidadãos desses países a recorrerem a formas de pagamento alternativas como, por exemplo, a Bitcoin;
- A Bitcoin tem propriedades parecidas com o ouro e outros metais preciosos, sendo encarada por alguns como forma de reserva de valor e esse facto tem também determinado a sua procura.

2.2.9 Ethereum

Vitalik Buterin - jovem programador russo - questionou-se quanto ao facto de se desenvolver uma rede Blockchain para cada finalidade. Em alternativa, poder-se-ia desenvolver uma rede Blockchain de propósito genérico que poderia ser programada caso a caso para resolver problemas específicos. Esta investigação levou à criação de uma nova geração de redes Blockchain [32].

A rede Ethereum é uma rede Blockchain similar à Blockchain 1.0, mas suporta uma execução descentralizada de programas informatizados ou *smart contracts*, escritos numa linguagem computacional universal. Esta propriedade é extremamente importante porque, assim, a tecnologia Blockchain é mais flexível e, através desta propriedade, é possível gerir as ações necessárias para implementar requisitos funcionais complexos [32] [52].

“Ethereum é uma rede Blockchain como a Bitcoin, mas tecnicamente muito mais complexa” [32], permitindo que qualquer utilizador faça o upload de um *smart contract* através de uma transação especial, pagando “um pequeno imposto” em moeda virtual da rede: o ether (ETH). Os *smart contracts* são executados em todos os nós da rede que acordam, através de um mecanismo de consenso semelhante ao usado na rede Bitcoin (Proof-of-Work), uma visão comum sobre o estado de execução do *smart contract*. Atualmente, existem outras redes capazes de executar *smart contracts*, mas a rede Ethereum foi a primeira a suportar esse tipo de recurso.

2.2.9.1 Ether

O *ether* é a unidade monetária da rede Ethereum. A sua criação advém das operações de *mining*, as quais servem como remuneração para os *miners* pelo seu trabalho de verificação/validação de blocos/transações e manutenção da segurança da rede. Pode existir transferência de *ethers* entre utilizadores (à semelhança da Bitcoin), mas estes também são utilizados para efetuar pagamentos pela execução de *smart contracts* - os *miners* são recompensados com cerca de 5 *ethers* por cada novo bloco registado, estando a rede concebida para registar um novo bloco a cada 15 segundos. Desta forma, o *ether* funciona como uma espécie de combustível para a rede Ethereum [32].

2.2.9.2 Tipos de conta Ethereum

Uma conta Ethereum é uma entidade com saldo em *ether* que pode enviar transações para a rede Ethereum. Estas contas tanto podem ser controladas por um utilizador ou pode ter o deployment de um *smart contract* [53].

Na rede Ethereum existem dois tipos de contas que têm diferentes funções e especificações [32]:

- **Externally Owned** - Este tipo de conta possui um saldo que corresponde ao montante em *ether* que o utilizador, detentor dessa conta, tem à sua disposição. Estas contas podem ser utilizadas na rede Ethereum para a realização de pagamentos diretamente entre utilizadores da rede;
- **Contract Account** - São um pouco mais complexas e correspondem aos objetos que representam os *smart contracts*. Estas contas são compostas por:
    - Contador - Garante que cada transação que interage com a conta só pode ser executada uma única vez;
    - Saldo – Saldo da conta medido em ether;
    - Código - Código informático do *smart contract* a ser executado;
    - Armazenamento – Dados armazenados na zona de armazenamento da *contract account* (ver secção 2.2.9.8).

2.2.9.3 Linguagem de programação Solidity

Esta linguagem de alto nível é utilizada para escrever *smart contracts* que são compilados para uma forma que a EVM (Ethereum Virtual Machine) – assunto abordado na secção 2.2.9.3 - os consiga interpretar. Apesar desta ser a linguagem mais utilizada, não é a única, existindo outras tais como a Mutant e a Serpent. No capítulo 4 encontram-se alguns excertos de um *smart contract* programado na linguagem Solidity.

2.2.9.4 Futuro de Ethereum

Está prevista uma nova fase de desenvolvimento, chamada Serenity, que prevê a substituição do atual mecanismo de consenso baseado em Proof-of-Work para um mecanismo baseado em Proof-of-Stake, chamado Casper. Esta mudança deve-se principalmente pelos problemas que têm vindo a ser levantados relativamente à Proof-ofWork – nomeadamente o seu consumo elevado de energia e o elevado tempo no consumido no processo de *mining* (ver secções , 2.2.5 e 2.2.6).

Empresas como a IBM e a Microsoft têm estado atentas ao desenvolvimento da Ethereum, isto porque as suas características são do interesse destas organizações, principalmente pela sua capacidade de execução de *smart contracts* e a utilidade que esta funcionalidade pode ter na automatização de processos, redução de custos e criação de modelos de negócio [32].

2.2.9.5 Transações e mensagens

Com a Ethereum, as transações correspondem à forma de interagir com as contas a partir de uma *externally owned account*. Desta forma, as transações são compostas da seguinte forma, conforme ilustrado na Figura 2-20 [32]:

- Mensagem a transmitir;
- Endereço do destinatário da transação;
- Assinatura digital identificando o ordenante;
- Montante em ether a transferir do ordenante para o destinatário;
- Dados a transferir com a transação;
- Dois dados adicionais:
    - STARTGAS - número máximo de passos computacionais que uma transação está autorizada a executar;
    - GASPRICE - taxa de utilização a pagar à rede pela execução de cada passo computacional (em ether).

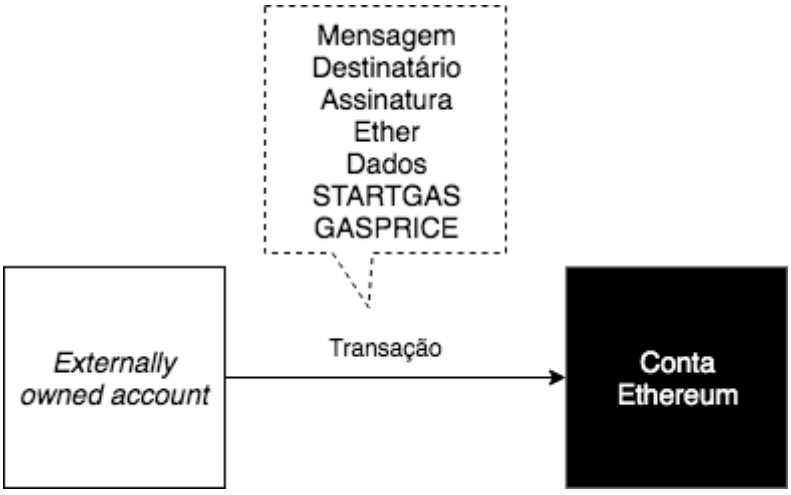

Figura 2-20 - Transação na rede Ethereum – adaptado [32]

2.2.9.6 GAS

É a unidade fundamental de computação da rede Ethereum, correspondendo um gas a um passo computacional [32]. Quanto mais complexo for o *smart contract* (número e tipo de etapas computacionais, memória usada para armazenamento, etc.), mais gas o contrato requer para ser executado e concluído [54].

Considerando que o *gas amount* é para executar um contrato e é fixada para qualquer contrato específico (conforme determinado pela complexidade do contrato), o *gas price* é especificado pela pessoa que deseja que o contrato seja executado, no momento em que o solicita - cada *miner* verificará quão generoso é o *gas price* e determinará se deseja executar o contrato [54].

A execução de um *smart contract*, ao ter este custo, é uma forma de impedir as pessoas de ativar *smart contracts* de forma desnecessária, o que resolve problemas relacionados ao *spam* de transações que aconteceria se a execução fosse gratuita [54]. É ainda uma forma de limitar o desenvolvimento de *smart contracts* computacionalmente “pesados” ou ineficientes [55].

As mensagens correspondem às interações que podem ocorrer com e entre *contract accounts*. Estas mensagens são compostas por, conforme ilustrado na Figura 2-21:

- Mensagem a transmitir;
- Emissor;
- Destinatário;
- Montante em Ether a ser transferido conjuntamente com a mensagem;
- Valor de STARTGAS.

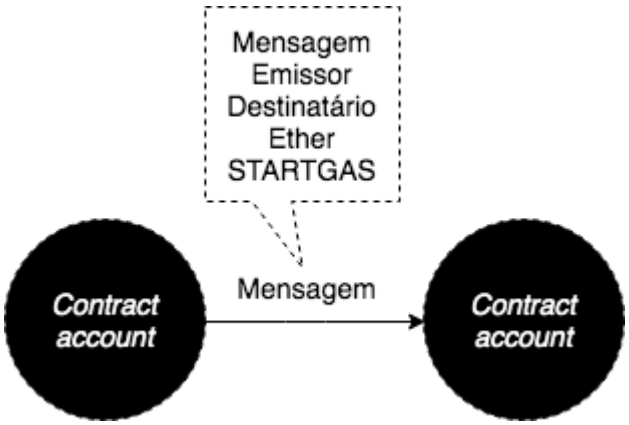

Figura 2-21 - Mensagem na rede Ethereum – adaptado [32]

2.2.9.7 Ethereum Virtual Machine

Para os *smart contracts* serem executados, são utilizadas máquinas virtuais que estão instaladas nos nós da rede: Ethereum Virtual Machine (EVM). Através das EVM, os *smart contracts* podem ser executados em todos os nós da rede independentemente das características dos equipamentos que disponham. A EVM está limitada à quantidade de gas existente. Desta forma, depende do número de passos computacionais que são necessários para executar a operação.

2.2.9.8 *Smart contracts* em Ethereum

Uma rede Ethereum é uma rede Blockchain que funciona como uma máquina de estados baseada em transações. O estado inicial de Ethereum é definido através do seu *genesis* (bloco inicial) e, daí em diante, as atividades como transações, contratos e *mining* mudam continuamente o estado da rede. Um exemplo disso é a alteração de saldo de uma conta que muda após a ocorrência de uma transação [56].

O estado não é armazenado nos blocos da Blockchain. Os blocos permitem fazer o *tracking* das transações (como se fosse um diário). Visto que uma das principais características de Blockchain é a imutabilidade, os blocos após o processo de *mining* não podem ser atualizados. Desta forma, os dados permanentes, como as *mined transactions*, são armazenados separadamente dos dados (como os saldos das contas dos utilizadores da Blockchain) numa camada de base de dados (Figura 2-22). Para o efeito, é usado o LevelDB, uma biblioteca *open-source* de armazenamento do tipo *key-value* da Google que fornece um conjunto de funcionalidades que permitem, entre outros aspetos, o mapeamento e navegação entre os dados, funções de comparação personalizadas e compactação automática. Este mecanismo é importante para a gestão do estado da Blockchain e é utilizado nos clientes Ethereum mais populares (go-ethereum ou Geth, cpp-ethereum e pyethereum) [56].

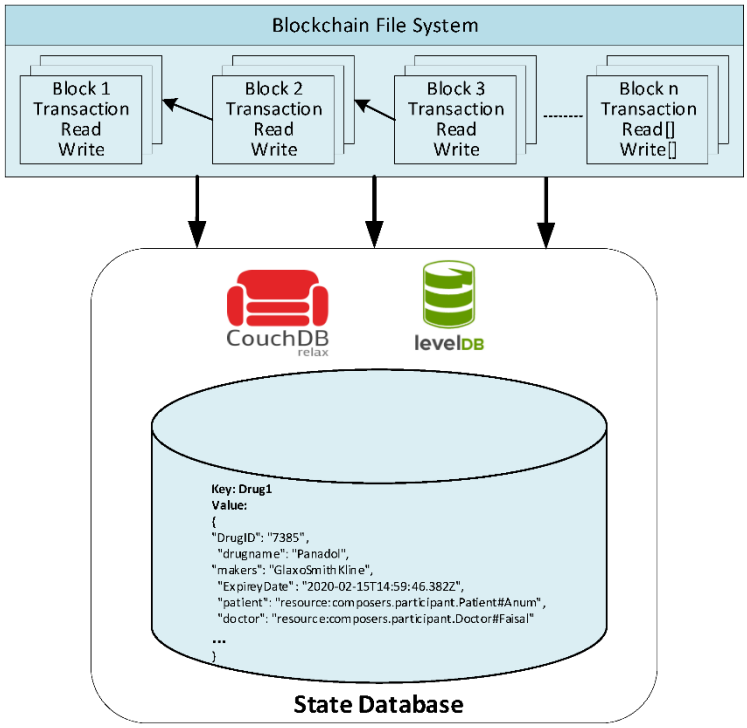

Figura 2-22 - Blockchain e State Database [97]

A Ethereum Virtual Machine pode lidar com uma ampla gama de transações via *smart contracts*. Esta EVM tem uma arquitetura baseada em pilha e pode armazenar diversos dados (por exemplo, operações de *byte code*), na memória (por exemplo, variáveis temporárias dentro de funções) ou no armazenamento (por exemplo, variáveis permanentes contendo entradas de base de dados). Cada *smart contract* só pode ler e escrever no seu próprio armazenamento (Figura 2-22).

O estado de um *smart contract* reside num endereço único específico no estado global da rede Ethereum, ao qual os utilizadores se podem ligar. Caso um utilizador não possua gas suficiente, a chamada do contrato e a transação correspondente não podem ser concluídas [55].

2.2.10 Semelhanças e diferenças entre Bitcoin e Ethereum

Estas redes são diferentes em alguns pontos, mas a base delas é semelhante. O Ethereum é uma atualização da Blockchain e diz-se que o Ethereum pertence à Blockchain 2.0 e o Bitcoin pertence à Blockchain 1.0 [32].

As principais semelhanças entre eles são [32]:

- Ambas têm moeda virtual;
- Base de dados descentralizada que regista o histórico de eventos;
- Mecanismo de consenso descentralizado para sincronizar as réplicas da base de dados;

- *Miners* que suportam a operação e a segurança da rede.

Embora as bases destas redes sejam semelhantes, existem algumas diferenças relacionadas principalmente com a capacidade da Ethereum de suportar uma execução descentralizada de *smart contracts*. A Tabela 2-6 mostra as principais diferenças entre as redes.

Tabela 2-6 - Diferenças entre Bitcoin e Ethereum – adaptado [32]

	Bitcoin	**Ethereum**
Objetivo	Criar rede global de pagamentos. As redes possuem características bastantes distintas.	Criar uma plataforma de computação capaz de executar aplicações informáticas de forma descentralizada - dApps (*Descentralized Apps*) [13]. As redes possuem características bastantes distintas.
Linguagem	É programável - Linguagem Script.	É programável - Linguagem Solidity - Linguagem Computacionalmente Universal. As aplicações desenvolvidas sobre esta rede podem interagir com outras aplicações Web.
Moeda	Bitcoin - principal função realizar pagamentos.	Ether - servir como forma de pagamento pela execução dos *smart contracts* e como forma de garantir que um *smart contract* não fica indefinidamente em execução, consumindo recursos da rede.
Saldo	Não tem saldo, apenas regista transações	Implementa o conceito de saldo
Consenso	Proof-of-Work	*Proof-of-Work*, mas prevê-se que seja substituído por *Proof-Of-Stake*.

2.2.11 Aplicações de Blockchain

Através das suas características, a tecnologia Blockchain permite que duas ou mais entidades que não possuam confiança entre si realizem transações de forma transparente sem a necessidade de um intermediário confiável.

Os casos de estudo onde melhor se aplica a tecnologia Blockchain estão relacionados com a agilização de processos principalmente, quando é imperioso ter intermediários confiáveis.

[13] dApp (Descentralized App) é uma aplicação criada numa rede descentralizada que combina um *smart contract* com uma interface para o utilizador (*front-end*).

Desta forma, estas são as áreas onde existe um maior investimento em Blockchain [57]:

- Serviços financeiros;
- Governo;
- Saúde;
- Identidade;
- Internet of Things;
- Seguros;
- Dinheiro;
- Música;
- Imobiliária;
- Cadeias de fornecimento.

2.3 Integração de IoT com Blockchain

Considerando o impacto que tem na sociedade, a IoT é um tema muito debatido atualmente. Desta forma, há muito trabalho nesta área, tendo um foco maior no conforto que pode proporcionar e no valor agregado que pode trazer para o dia-a-dia. No entanto, em alguns trabalhos mais recentes surgiram questões de segurança, privacidade e confiança às quais os utilizadores estão sujeitos em transações usando tecnologias de IoT. Considerando estas preocupações, têm sido feitas propostas que visam a integração da tecnologia Blockchain em redes de IoT. Desta forma, a tecnologia Blockchain, com potencial para revolucionar o negócio digital, e a junção desta com IoT tornou-se num assunto de maior relevância relacionado com a Internet. Estiva-se que em 2019, 20% de todas as implementações IoT tinham níveis básicos de serviços blockchain (IDC) [58]. A mesma fonte garante que a integração das duas tecnologias é essencial para a construção de confiança (redução do risco de intrusão e alteração de dados), redução de custos (redução no número de intermediários) e aceleração das transações. Aproveitando estes pontos fortes, já existem vários projetos em diversas áreas, a saber, setores de energia, saúde e seguradoras.

Christidis e Devetsikiotis [44] analisam a integração da tecnologia Blockchain com IoT, na atualização de dispositivos. Do lado do fabricante, o atual modelo centralizado tem um alto custo de manutenção - por exemplo a distribuição de atualizações de software para milhões de dispositivos anos após terem sido descontinuados. Do lado dos utilizadores têm alguma falta de confiança em dispositivos que “ligam a casa” em segundo plano e necessitam de uma abordagem segura e transparente. Aplicando *smart contracts*, os dispositivos IoT podem consultar o contrato, descobrir o novo *firmware* e solicitá-lo através de um sistema distribuído. Segundo estes autores, existem mais exemplos tais como a aplicação destas tecnologias no setor energético para que máquinas (por exemplo, painéis solares) possam

registar em Blockchain dados pertinentes e vender autonomamente a energia excedente (configurável pelo utilizador) aos "vizinhos" através de *smart contracts*. Outra aplicação ainda referida é a distribuição de encomendas. Por exemplo, um contentor pode ser transportado para um “porto vizinho”, depois exportado para o "porto de destino" e, finalmente, transportado para o destinatário final. Através das tecnologias mencionadas, é possível acompanhar a encomenda, bem como é possível a utilização de *smart contracts* que vão sendo assinados, o que garante que a encomenda está, de facto, num determinado lugar. Assim, os custos com intermediários são reduzidos, há maior confiança no serviço e o processo é muito mais ágil, o que também pode ser refletido na sua velocidade de execução.

Segundo Petracek [59], com a introdução da Blockchain nas estruturas IoT, a segurança pode melhorar, assim como se pode otimizar recursos de forma autónoma, fornecendo:

- Um sistema de registo distribuído para partilhar dados através de uma rede de *stakeholders*;
- Regras de negócios incorporadas na rede de forma automatizar as interações entre os nós do sistema;
- Segurança baseada em funções de *hash*, verificação de autenticação de identidade e proveniência;
- Modelos de consenso para detetar os maus atores e mitigar as ameaças.

Com esta implementação, a integridade do sistema será aprimorada e, como não há um sistema central, ameaças como ataques *denial of service* podem ser inerentemente dissuadidas em diferentes camadas da arquitetura. Além dos problemas de segurança, a aplicação da tecnologia Blockchain em IoT também pode resolver vários problemas em negócios digitais, como:

- **Acompanhamento de modelo analítico**: permite que o sistema registe os resultados na lógica executada na rede para ter conformidade com as regulamentações e, assim, criar um registo imutável do porquê da tomada de certas "decisões" ao longo do processamento em IoT;
- **Atualizações de software seguras**: capacidade de publicar atualizações de software com um URL na Blockchain, juntamente com um código hash da atualização que pode ser validado por dispositivos IoT conectados à Blockchain durante o processo;
- **Pagamentos e micropagamentos**: pagamentos automatizados a participantes da rede de negócios com base em dados dos sensores (indicando, por exemplo, a conclusão do serviço ou entrega do produto) e micropagamentos entre os próprios dispositivos em determinadas redes para funções e recursos - tudo sem envolvimento humano.

2.3.1 Aplicações

Atualmente, existem algumas áreas muito interessadas na integração da Blockchain em IoT. A integração pode melhorar as transações e as áreas mais interessadas estão no setor industrial e nas cadeias de fornecimento.

2.3.1.1 Setor industrial

De acordo com [60], nos próximos 10 anos Blockchain e IoT terão um enorme impacto para as empresas do setor industrial porque, através de IoT, é possível recolher dados de sensores em tempo real e a Blockchain permite o armazenamento e a partilha da informação mais relevante. Este autor analisa três áreas em que essas tecnologias podem ser aplicadas. A primeira área analisada está relacionada com redes de fornecedores, onde é demonstrada uma forma de reduzir atrasos, eliminando intermediários e burocracias, aumentando a previsão de entrega dos pedidos e, consequentemente, tornando o processo mais ágil. Assim, sugere-se que os pedidos tenham dispositivos de IoT e que a Blockchain armazene essas informações em toda a cadeia, servindo assim como prova de envio e entrega.

A segunda área analisada consiste nas soluções para veículos autónomos, onde um desafio é lançado para que um veículo seja totalmente autónomo, sendo especialmente focado no deslocamento automático para o seu reabastecimento e no agendamento de um conserto numa oficina se uma falha for detetada. Para o efeito, os veículos beneficiariam de uma solução de IoT com Blockchain, utilizando os dados mais adequados recolhidos para a Blockchain dos sensores do veículo. Os fabricantes desse tipo de veículo também teriam acesso oportuno a informações sobre um colapso (através do bloco de sensores de veículos) e poderiam determinar as tendências de falhas que estariam a ocorrer para o componente em questão e poderiam, assim, garantir segurança e satisfação do consumidor. O veículo pagaria os custos de reparação sem intervenção humana e um registo dessas operações seria armazenado na Blockchain e partilhado entre os *stakeholders*.

A última área abordada por [60] é a gestão de recursos em fábricas onde, com o uso de IoT e Blockchain, seria possível prever e prevenir falhas num equipamento da fábrica. Sensores de equipamentos detetariam condições como vibração excessiva ou calor, o que poderia levar a falhas ou lesões ao operador. Os dados recolhidos seriam usados para detetar tendências para tais falhas e facilitar a manutenção proativa e as reparações antes que a falha ocorra. Os reguladores e fornecedores de equipamentos teriam visibilidade dos registos do equipamento e poderiam fornecer inspeções e certificações em tempo útil para garantir a confiabilidade do equipamento.

A Kaleido Insights realizou uma pesquisa com líderes de negócios [29]. Aqui foram analisadas as oportunidades associadas às “coisas” (ou seja, como a Blockchain permite aumentar a confiança num ecossistema). Nessa análise, os autores consideram que a Blockchain suporta a confiança do produto em três domínios (Identidade do produto, transações do produto e iterações do produto - Figura 2-23) e a confiança da rede em cinco

oportunidades do ecossistema (Cadeia de fornecimento, gestão de rede IoT, autenticação do utilizador final, redes *on-demand/asset-sharing* e *smart contracts* & conformidade - Figura 2-24).

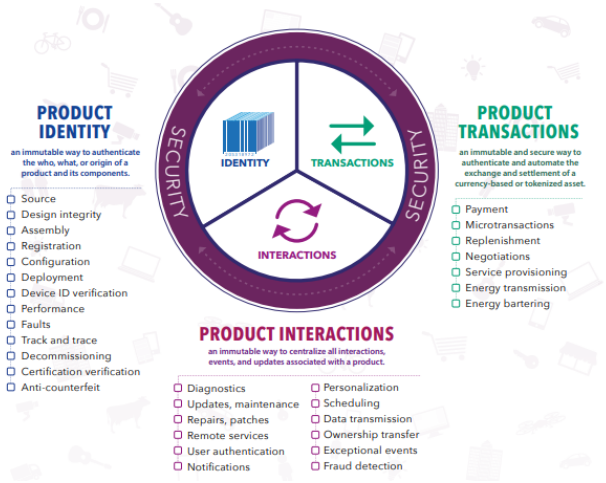

Figura 2-23 - Blockchain suporta produtos IoT em três categorias de casos de uso [29]

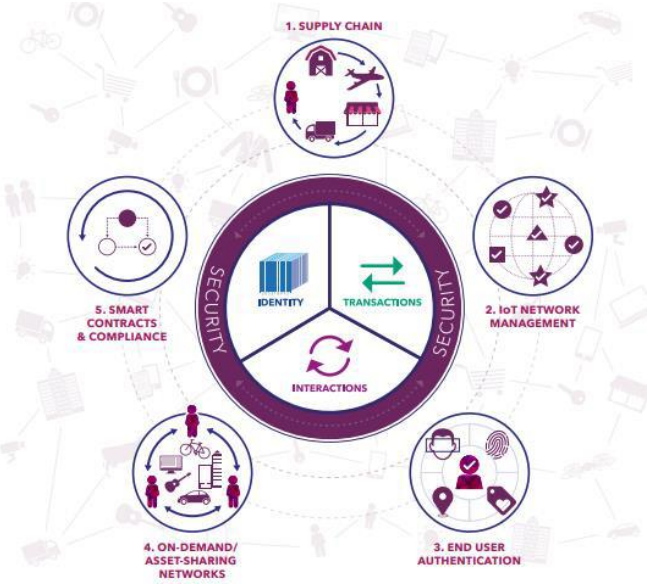

Figura 2-24 - Casos de uso para blockchain - A convergência de IoT envolve tanto o ecossistema como o produto [29]

2.3.1.2 Cadeia de fornecimento

Com o avanço da Indústria 4.0, o potencial da Blockchain na cadeia de fornecimentos e logística tem sido estudado pelas pessoas responsáveis por esta indústria. Nos portos ainda é utilizada a comunicação tradicional, há um elevado número de transações e os *stakeholders* trabalham num ambiente intensivo, o que tem por efeito diminuir a eficiência das operações [61] – é frequente que o transporte de um contentor entre dois ponto envolva mais de 30 partes diferentes, com uma média de 300 interações entre eles [62]. Essas são

as razões para incluir uma solução Blockchain na gestão da cadeia de fornecimento, logística e transporte. Esta área está no topo da lista dos gastos com Blockchain e nas iniciativas de Blockchain, logo após o setor financeiro [61]. Em 2017, a Maersk firmou uma parceria com a IBM para procurar formas de integrar Blockchain nos seus processos – essa parceria resultou numa solução para a cadeia de fornecimento com a utilização de Blockchain [62].

2.3.1.3 Agricultura

Casado-Vara [5] propõe o uso de blockchain em *multiagent system* (MAS) para aumentar a segurança e a privacidade na cadeia de fornecimento alimentar. Na Figura 2-25 é descrita a atual cadeia de fornecimento, o seu funcionamento e a proposta de como funcionaria com a inclusão de Blockchain. O atual modelo da cadeia de fornecimento é um modelo linear de produtores e compradores para retalho e serviços alimentares. Com a introdução de Blockchain as cadeias de fornecimento tornam-se descentralizadas – sendo todas as transações armazenadas em Blockchain. Todos os membros podem escrever as suas transações na Blockchain, no entanto, só têm permissões de leitura para os blocos que estejam diretamente ligados a si.

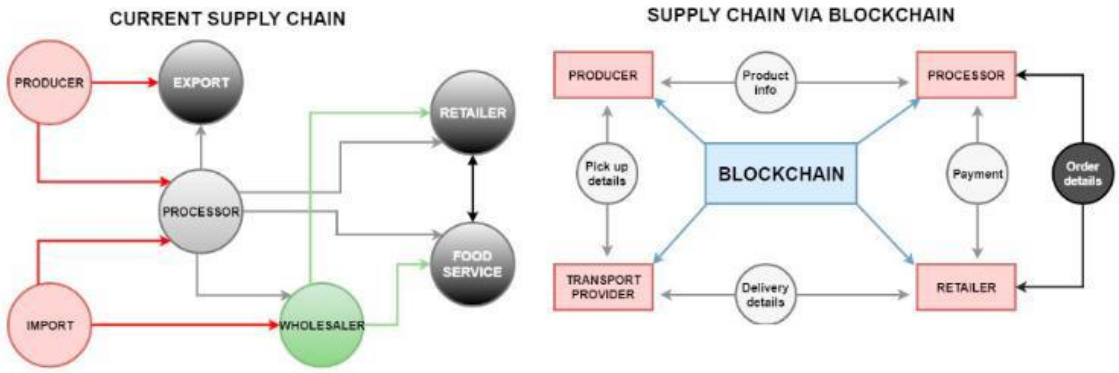

Figura 2-25 - Comparação da cadeia de fornecimento tradicional e da cadeia de fornecimento com Blockchain [5]

No atual modelo da cadeia de fornecimento existe a centralização dos dados em cada um dos elementos da cadeia, o que leva a que os restantes não consigam ver as transações – assim, também o consumidor não consegue verificar a origem do alimento que está a adquirir [5].

Na cadeia de suprimentos via Blockchain “todos os membros da cadeia de fornecimentos armazenam todas as suas transações na blockchain” [5]. Desta forma, esse modelo corrige as desvantagens do modelo atual e há mais segurança nas transações. Esta solução também remove intermediários (com *smart contracts* - Figura 2-26) e os dados são descentralizados, o que permite que cada membro da cadeia de fornecimento visualize dados importantes para as suas operações na Blockchain - “Por exemplo, o produtor pode visualizar as informações do produto do processador e os detalhes do transporte dos produtos” [5]. Nesta solução cada camada regista os dados das suas transações na

Blockchain e as camadas fazem a gestão dos artigos – comunicando entre si através de *smart contracts* (servindo para compra e venda de artigos).

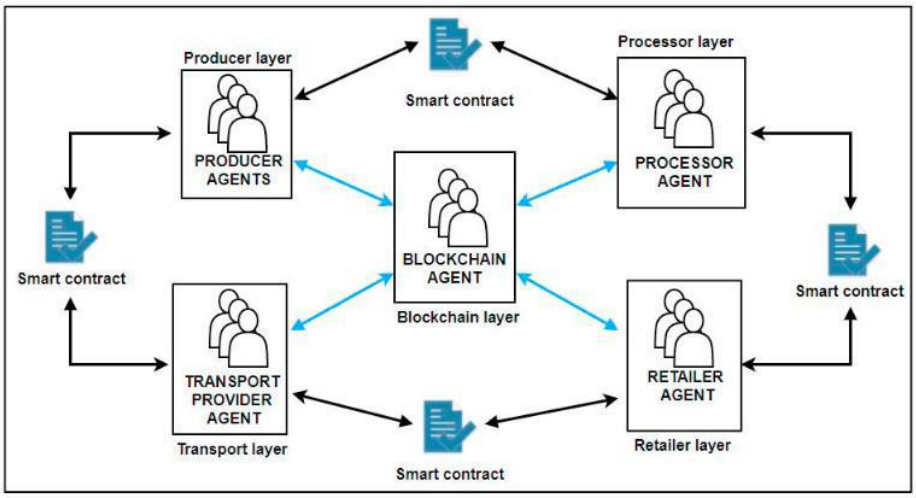

Figura 2-26 - Cadeia de fornecimento através da arquitetura Blockchain MAS [5]

Faye [63] também avalia o uso da tecnologia Blockchain no negócio agrícola. Aqui é referido que Blockchain pode eventualmente substituir todo o “trusted third party” centralizado.

Na cadeia de transformação há trocas estabelecidas entre os seguintes pares de autores: agricultor e a cooperativa, a cooperativa e o transformador, transformador e revendedor. Neste caso, nem todas as partes interessadas podem estar presentes no momento das trocas. Para conhecer o caminho que cada produto seguiu, é necessário registar todas as transações realizadas. Para isso, é necessário existirem testemunhas para que não haja falsificação ou corrupção. Com a tecnologia Blockchain, qualquer pessoa pode aceder às transações sem um intermediário. Dessa forma, é simples e seguro armazenar as transações entre todas as entidades, porque cada transação é validada pelos nós da rede.

Um exemplo é quando existe uma transação entre a cooperativa e um transformador. Aqui, cada nó vai verificar se a cooperativa assinou a transação em questão (na sua cópia local). Verifica ainda, na sua cópia local, se existe uma transação onde identifique que os agricultores venderam as suas propriedades [63].

O sistema atual é um sistema de ineficaz onde facilmente poderá haver uma falsificação[63]. A tecnologia Blockchain pode gerir o *tracking* e fornecer garantias quanto à certificação da origem ou qualidade [63]. A tecnologia RFID (Radio Frequency Identification) [64] permite armazenar e recuperar dados remotamente através de sensores [63]. Assim, é possível armazenar e gerir todas as informações do produto. Relativamente à certificação da qualidade, caso exista um produto que seja identificado como “não conforme” (não poderá ser vendido), os participantes da rede/mercado deverão demonstrar que cumpriram os padrões de qualidade estabelecidos e aquele produto não poderá ser vendido [63].

Além disso, Blockchain permite o uso de *smart contracts*. Um exemplo da utilização destas tecnologias é acionar o pagamento de um seguro assim que os *smart contracts* identifiquem as condições necessárias para o fazer [63].

Blockchain simplificará, portanto, o trabalho de todos os atores na "Agri-chain" e tornará as transações mais transparentes [63].

2.3.1.4 IoT e Blockchain na produção dos produtos alimentares

Os dados recolhidos pelos sensores são importantes para ajudar o agricultor a tomar decisões e a ser avisado em situações de alerta. Estes dados têm de ser tratados antes de serem armazenados. Para além da limpeza dos dados, é necessário acrescentar-lhes valor e estruturá-los – por exemplo, adicionar timestamps e demografia. A utilização de algoritmos de *machine learning* são uma mais-valia, pois através deles é possível obter resultados de alto valor para os produtores: recomendações de qualidade de cultura, identificação de colheita, previsão de rendimento de culturas, GrowScore (fator de crescimento de culturas automatizado) e previsão da melhor data para a colheita dos produtos [65].

Estes dados obtidos através de algoritmos de *machine learning* deverão ser armazenados na Blockchain para que seja possível que os participantes na rede (incluindo participantes do mercado agrícola, produtores, prestadores de serviço, entre outros *stakeholders* da área) possam aceder-lhes de forma transparente. Estes dados, para além de poderem ser utilizados para o negócio da agricultura, também poderão ser utilizados na área dos seguros, pois com a recolha de dados meteorológicos é possível analisar a veracidade de um determinado requerimento relativamente a prejuízos causados por más condições climatéricas [65].

2.3.1.5 Cadeias de fornecimento alimentares

Nesta secção é abordada a integração de Blockchain em IoT com aplicação às cadeias de fornecimento alimentares, dando um exemplo da sua utilização nos vários componentes da mesma (Figura 2-27).

A Tabela 2-7 apresenta os vários processos nas cadeias de fornecimento alimentares onde a integração de IoT e Blockchain podem ser úteis para resolver algumas questões que se colocam nas cadeias de fornecimento alimentares tradicionais.

Tabela 2-7 - Processos das cadeias de fornecimentos alimentares onde IoT e Blockchain podem ajudar – adaptado de [65]

Processo	**Descrição**
Dados recolhidos	• Dispositivos IoT recolhem dados; • Demonstram as condições em que os produtos foram criados; • Podem ser uma ajuda para garantir a qualidade dos produtos em questão.
Distribuição de plantações adultas	• Quando as plantações estão a ser cultivadas, as empresas de processamento vão fazendo propostas de licitação na plataforma; • As licitações são validadas por *smart contracts*; • Os dados recolhidos em todas as fases do processo são armazenados na Blockchain de forma a garantir que tudo foi cumprida em todas as etapas da cadeia de fornecimento alimentar; • No transporte das plantas, dispositivos IoT estão a recolher dados acerca das condições de transporte; • As informações recolhidas ao longo do processo podem ajudar os restantes *stakeholders* a confirmar se o alimento entregue é de boa qualidade ou não.
Fornecimento de alimentos processados a *wholesalers* e *reteilers*	• *Wholesalers* e *reteilers* podem licitar os produtos que desejam através da plataforma de licitação; • Semelhante ao processo de distribuição, os produtos alimentares também são distribuídos para *wholesalers* e *reteilers* onde são recolhidos dados durante o seu transporte através de dispositivos IoT; • Blockchain oferece acompanhamento ao longo da cadeia de fornecimento, ajudando empresas de alimentos a realizar *recalls* de alimentos ou pesquisas de forma rápida e contínua.
Consulta de dados pelo consumidor final	• Os *stakeholders* podem aceder a informações relacionadas à qualidade dos alimentos em todas as etapas, tais como: números de lote, processamento de alimentos, temperatura de armazenamento, entre outras informações que ficam armazenadas na Blockchain ao longo de todo o processo; • Como Blockchain traz transparência para o ecossistema da cadeia de abastecimento alimentar, será mais fácil descobrir quando e como os alimentos foram contaminados.

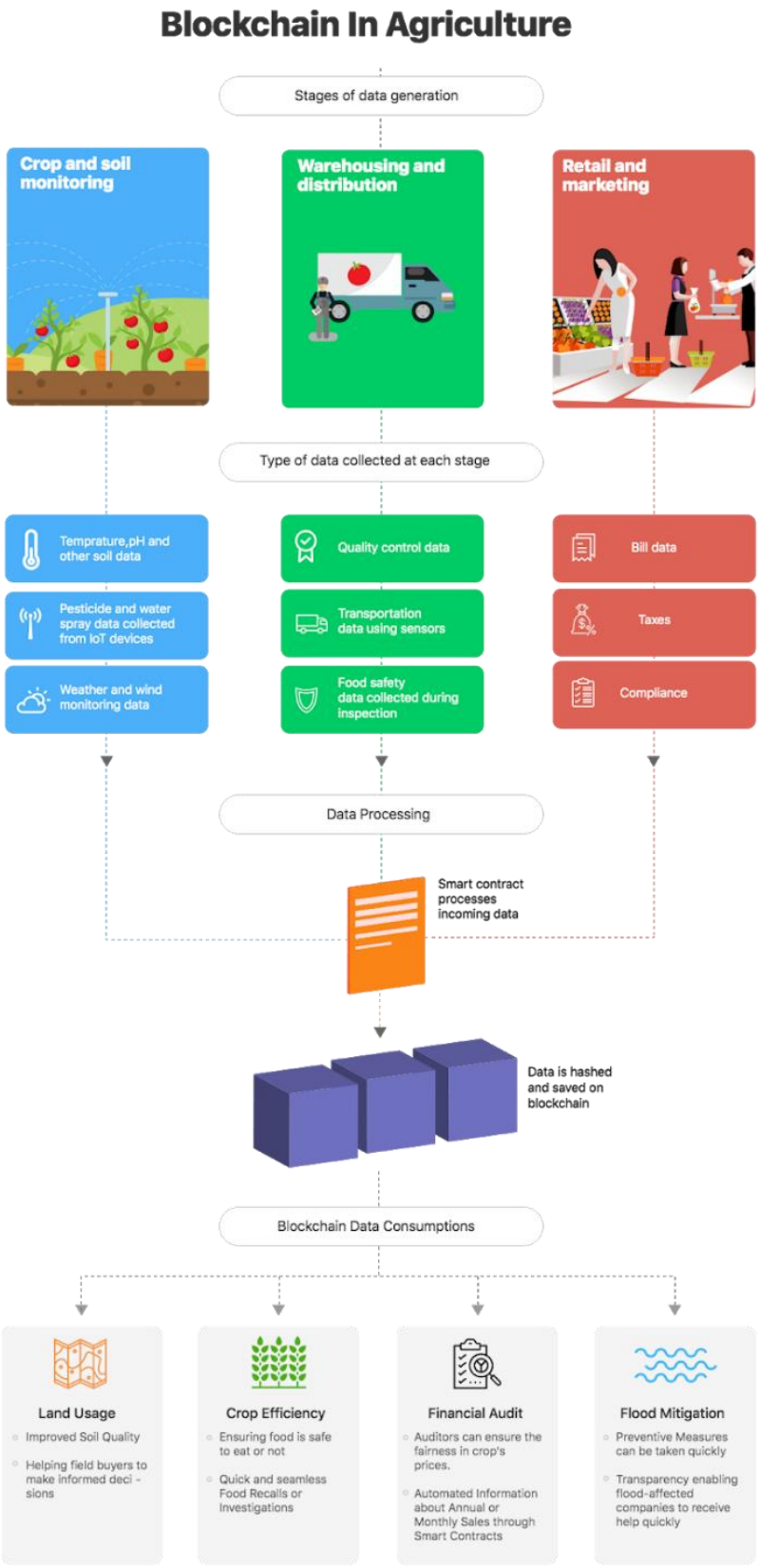

Figura 2-27 - Caso de uso da utilização de Blockchain e IoT na agricultura [65]

Benefícios da utilização de Blockchain na agricultura

A Tabela 2-8 apresenta alguns dos benefícios da utilização de Blockchain no setor da agricultura.

Tabela 2-8 - Benefícios da utilização de Blockchain na agricultura – adaptado de [66]

Benefício	**Descrição**
Acompanhamento dos alimentos	Blockchain pode registar informações e as mesmas são inalteráveis em todas as etapas da cadeia de fornecimento de alimentos – fornecendo informações confiáveis sobre as origens dos alimentos e o caminho exato que os alimentos tiveram desde a quinta à mesa.
Otimização da cadeia de fornecimento	A tecnologia Blockchain pode ajudar a corrigir o desequilíbrio existente nas cadeias de transmissão, pois grava as transações em tempo real e fornece informações atualizadas sobre a oferta e a procura a todos os participantes. O acesso a estas informações pode permitir que os agricultores definam os seus próprios preços e otimizem as quantidades de produtos que disponibilizam no mercado. Para além disso facilita o processo de negociação, removendo intermediários e permitindo que as partes interessadas concluam as suas transações de forma segura.
Melhores preços e opções de pagamento	A tecnologia Blockchain pode resolver as ineficiências das cadeias de fornecimento tradicionais – essencialmente no processo de pagamento que atualmente pode levar alguns dias. A utilização de *smart contracts* é também uma possibilidade de integração onde os pagamentos podem ser acionados automaticamente após o cumprimento de determinada condição (por exemplo, entrega de mercadorias).

2.3.2 Integração de Blockchain em LoRaWAN

Em [67] é apresentada uma proposta de um modelo onde é possível obter o melhor da LoRaWAN e de Blockchain com o intuito de criar um “sistema de rede aberto, confiável, descentralizado e inviolável” [67]. Este trabalho tem como principal objetivo responder à questão: “Como é que um servidor de partilha distribuído pode aumentar a confiança das pessoas numa LoRaWAN pública e ajudar a transportar dados de *gateways* para servidores de aplicações sem haver roubos, adulterações ou enganos?”.Com base nos conceitos de *crowdsourcing* e economia de partilha, é proposta uma arquitetura de Blockchain para servidores LoRaWAN, que pode utilizar as vantagens da tecnologia de Blockchain e da tecnologia LoRaWAN.

A Figura 2-28 apresenta a aplicação de Blockchain ao nível da camada de servidor de rede LoRaWAN. Esta decisão é tomada pelas seguintes razões [67]:

- Normalmente os gateways da LoRaWAN têm recursos restritos por recursos e dispositivos de IoT ao ar livre – ou seja, não são adequados para suportar muitas funções de segurança, verificação e armazenamento de Blockchain;
- Os Join Servers da LoRWAN por norma são fornecidos pelos fabricantes de end-nodes para produzir chaves de sessão – logo, também não são adequados para realizar as funções de Blockchain;
- Os Application Servers da LoRaWAN por norma são fornecidos pelos clientes para processarem os principais dados de negócio – assim, também não adequados para realizar as funções de Blockcain.

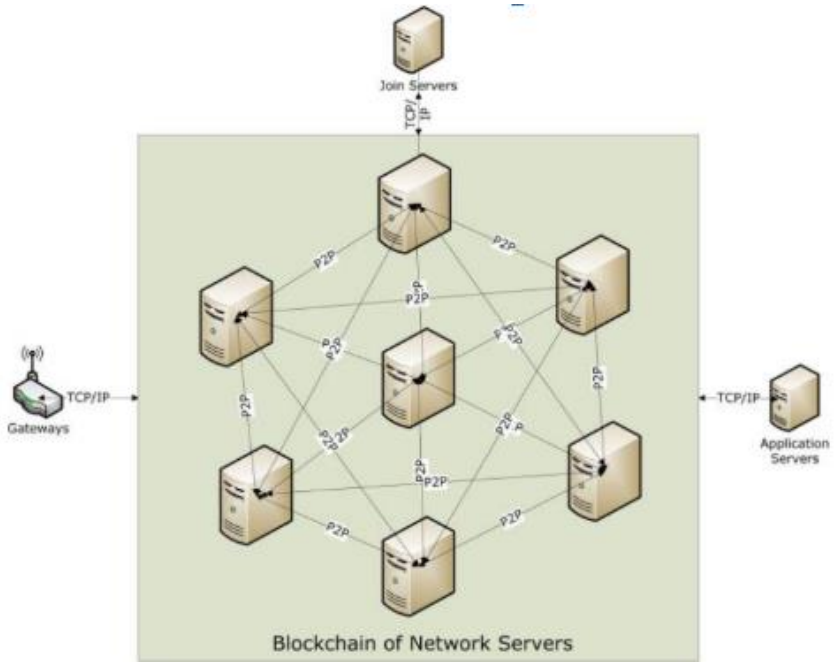

Figura 2-28 - Arquitetura de Blockchain para o servidor LoRaWAN [67]

Em cada Network Server foram adicionados os componentes de gestão da Blockchain.

Na Figura 2-29 é apresenta a estrutura de dados que é armazenada em Blockchain. Neste caso os autores do trabalho optaram por introduzir blocos com 2000 registos cada, contudo, este número é alterável caso assim seja necessário.

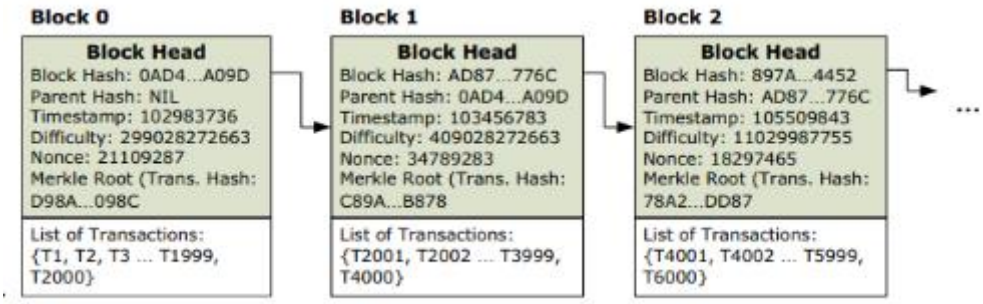

Figura 2-29 - Exemplo da estrutura de dados armazenada em Blockchain [67]

Este trabalho resulta numa solução que utiliza a tecnologia Blockchain de forma a criar um sistema aberto, confiável, descentralizado e inviolável, fornecendo um mecanismo que permite verificar se os dados de uma transação existiram num determinado momento na rede. Como trabalho futuro, os autores deste trabalho falam na inclusão de *smart contracts* de forma a automatizarem alguns processos no modelo da rede IoT. Referem ainda que gostariam de construir redes Blockchain LoRaWAN totalmente dimensionadas para conectar *Gateways* e *Application Servers*.

3 Framework - BC4IoT

Neste capítulo é apresentada uma *framework* definida para integrar uma rede IoT com Blockchain, designada por BC4IoT (*BlockChain for IoT*). Para além da sua arquitetura genérica, também é especificado cada componente da mesma por forma a clarificar a função de cada um.

Esta *framework* tem como principal objetivo proporcionar um ambiente onde seja possível manter os dados recolhidos de uma rede de sensores num local seguro, inviolável e público. A *framework* faz a agregação e tratamento dos dados destes sensores e armazena-os numa rede Blockchain Ethereum.

Entre os vários componentes existentes na arquitetura da *framework* BC4IoT (secção 3.1), inclui-se uma rede de sensores onde são recolhidas diversas informações em vários locais de forma periódica. Outro fator importante é a segurança da comunicação entre os sensores e a rede IoT. Sendo um sistema que é pautado pela segurança e pela confiança dos dados, é necessário utilizar uma rede IoT onde estes requisitos sejam cumpridos.

Apresentado o âmbito geral para o bom funcionamento e cumprimento dos principais objetivos da *framework* desenvolvida, a tecnologia The Things Network/LoRaWAN possui as caraterísticas adequadas aos objetivos pretendidos, tais como:

- **Longo alcance** – existindo uma infraestrutura que assim o permita, é possível adaptar-se a grandes cidades;
- **Escalável** – a rede possui uma grande capacidade de expansão;
- **Baixo custo** – a vida útil das baterias é longa;
- **Segurança –** possui mecanismos de segurança que permitem a integridade e confidencialidade dos dados, pois garante a encriptação *end-to-end* dos dados recolhidos pelos sensores [68].

A cada nova recolha é aplicado um algoritmo – definido num *smart contract* - que, com base nas recolhas feitas anteriormente por aquele sensor, avalia se aquele registo é válido ou inválido.

No capítulo 4 é feito um estudo de aplicação da *framework* BC4IoT para a recolha e validação de dados ambientais em ambiente urbano. No entanto, esta pode ser adaptada a outros casos de estudo/negócios, desde que sejam utilizadas as tecnologias que a mesma suporta - uma rede similar de sensores baseada em TTN e a rede Blockchain Ethereum. Para cada caso de estudo diferente será necessário alterar os *smart contracts* por forma a que as regras de negócio específicas sejam cumpridas.

3.1 Arquitetura

A *framework* está dividida em vários componentes (Figura 3-1), tais como:

- Sensor Application (secção 3.1.1);
- Web Application (secção 3.1.2);
- TTN/LoRaWAN (secções 2.1.1 e 2.1.3);
- Rede Blockchain – Ethereum (secção 2.2.9)

As aplicações Sensor Application e Web Application assentam sobre node.js e fazem a comunicação com a rede Ethereum através de um conjunto de bibliotecas denominada web3.js (secção 3.2.4).

De uma forma geral, os dados recolhidos por um sensor são armazenados nos servidores da rede de sensores TTN. De seguida, a Sensor Application (SA) recolhe os dados do sensor, presente na TTN. Esta SA trata os dados e envia-os para a Blockchain onde são validados e armazenados no *smart contract*. A Web Application faz a ligação à Blockchain por forma a obter os dados armazenados no *smart contract*, trata-os e mostra-os de forma intuitiva ao utilizador.

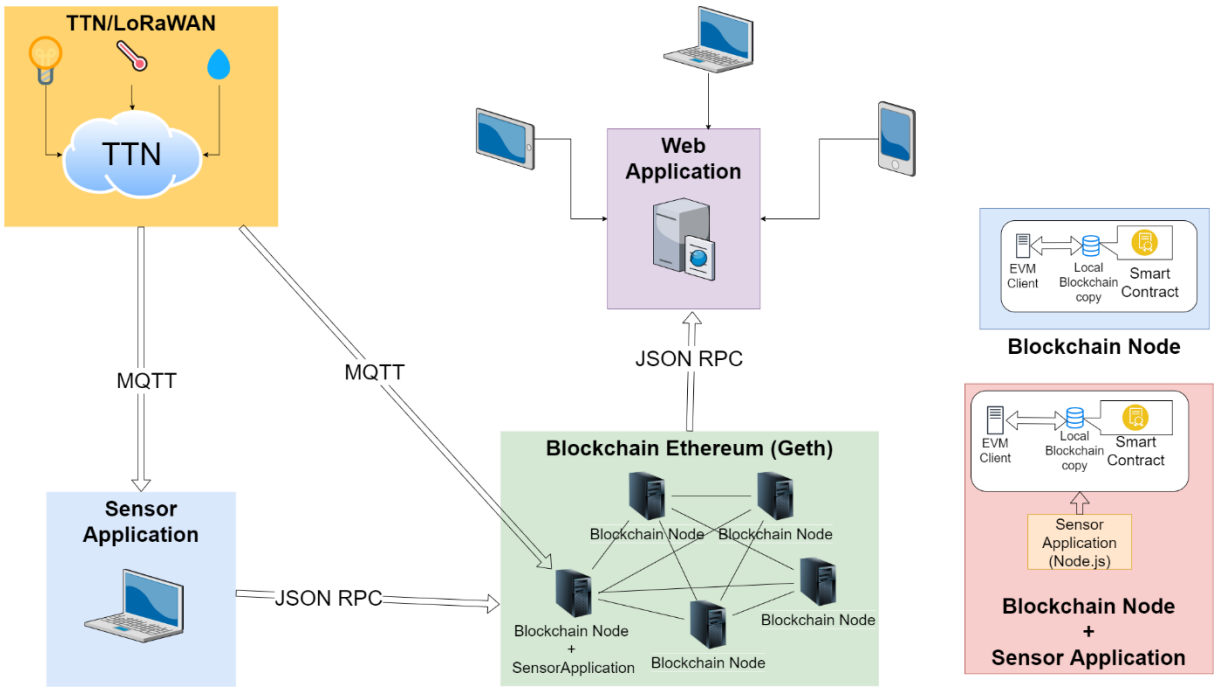

Figura 3-1 - Arquitetura da *framework* BC4IoT

3.1.1 Sensor Application

A Sensor Application é uma aplicação em JavaScript (node.js) que tem como principal objetivo recolher e tratar os dados de um sensor da rede de sensores TTN e enviar o pedido

de armazenamento para uma rede Blockchain Ethereum. Após o registo do sensor na TTN e após configurar a Sensor Application com os seus dados, é possível iniciar a utilização desta aplicação.

Ao existir uma nova recolha de dados do sensor, estes são armazenados nos servidores da rede de sensores, sendo também criada uma entrada (*uplink*) na Sensor Application. De seguida, a Sensor Application recolhe os dados do sensor que estão na TTN e faz o pedido de inserção a um nó da rede Blockchain. Aí, os dados recolhidos são validados através do algoritmo que está presente no *smart contract* onde os mesmos serão armazenados.

A Sensor Application pode ser instalada e configurada pelo utilizador que adquire um sensor ou mais sensores (que transmitem dados através de LoRaWAN) e pretende iniciar o armazenamento destes dados na Blockchain. Os dados recebidos dos sensores são tratados na Sensor Application e são enviados para a Blockchain, onde irão ser validados e armazenados no *smart contract* presente naquela rede (os algoritmos de validação podem variar de caso para caso/de negócio para negócio).

A rede Blockchain é composta por diversos nós. Nestes nós também poderão estar instaladas Sensor Applications que irão comunicar com a Blockchain de igual forma.

O diagrama na Figura 3-2 ilustra a sequência de ações realizadas pela Sensor Application ao longo de todo o processo. No “Anexo A - Sensor Application” encontra-se a documentação técnica relativa a esta aplicação.

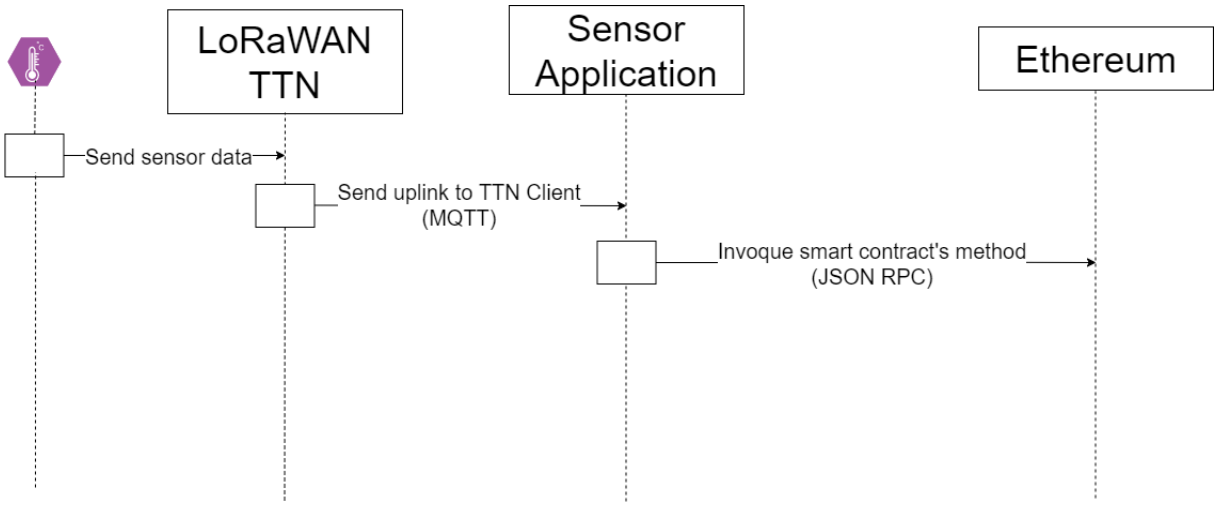

Figura 3-2 - Diagrama de sequência da Sensor Application

3.1.2 Web Application

A Web Application consiste numa aplicação em Vue.js e tem como principal objetivo permitir a visualização e análise da informação tratada que está presente no *smart contract* armazenado na Blockchain. Esta aplicação acede aos dados do *smart contract* através de

Web3.js. Depois, os dados são preparados de forma a que sejam o mais intuitivo possível ao utilizador (gráficos, tabelas, filtros, etc.) e são mostrados ao utilizador.

A Figura 3-3 apresenta a sequência de ações desencadeadas quando a Web Application ao longo de todo o processo.

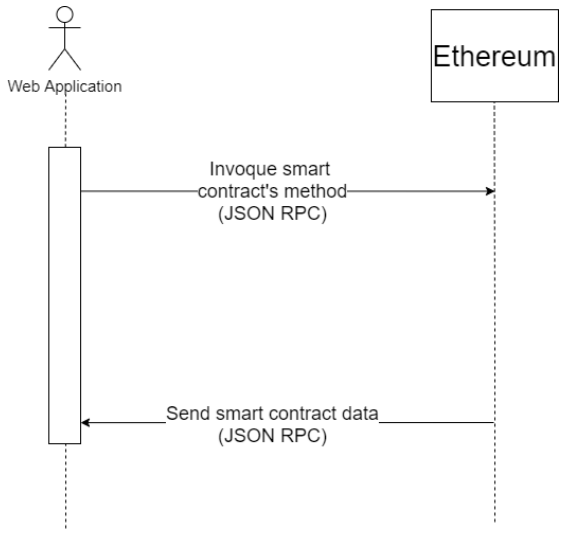

Figura 3-3 - Diagrama de sequência da Web Application

No “Anexo B - Web Application” encontra-se a documentação técnica relativa a esta aplicação.

3.2 Tecnologias

Este capítulo descreve as tecnologias utilizadas para o desenvolvimento da *framework* apresentada, bem como as decisões tomadas na sua seleção.

3.2.1 Truffle

Truffle é um 3-em-1 – um ambiente de desenvolvimento, uma *framework* de testes e um *asset pipeline* para a Ethereum, com o objetivo de tornar a vida de um *developer* Ethereum mais simples [69].

Esta *framework* (Truffle) oferece um conjunto de funcionalidades tais como [69]:

- **Gestão dos Smart Contracts** - compila os contratos que serão executados na EVM (secção 2.2.9.7);
- **Testes Automatizados** – permite a criação de teste tanto em JavaScript como em Solidity (secção 2.2.9.3);
- **Deployment & Migrations** - permite criar scripts de migração e implementação de *smart contracts* em qualquer rede blockchain pública baseada em Ethereum;
- **Gestão da rede** - permite ligar a qualquer outra rede blockchain pessoal;

- **Consola de Desenvolvimento** - permite interagir com os *smart contracts* diretamente através da consola;
- **Script Runner** - corre qualquer script dentro do projeto.

Truffle é a *framework* mais completa para desenvolvimento em Ethereum. Isto deve-se essencialmente às características enumeradas anteriormente, o que a torna essencial para o desenvolvimento de dApps (secção 2.2.10).

3.2.2 Ganache

O Ganache é uma Blockchain Ethereum (secção 2.2.9) pessoal de desenvolvimento local que simula o comportamento de uma Blockchain pública. Este oferece 10 contas com 100 ether (secção 2.2.9.1) fictícios [70] e pode ser utilizado para implementar *smart contracts*, desenvolver aplicações e executar os respetivos testes.

O Ganache está disponível como uma aplicação *desktop* e como uma ferramenta de linha de comando (conhecida como TestRPC).

O Ganache é um simulador de uma Blockchain pública, isto é, uma Blockchain onde qualquer pessoa pode ler, enviar transações e vê-las incluídas na Blockchain (caso as mesmas tenham sido validadas). Também qualquer pessoa pode participar no processo de consenso - processo que determina quais os blocos a serem adicionados à cadeia e qual o estado atual da cadeia [71].

Com o Ganache é possível realizar todos os testes necessários aos *smart contracts* desenvolvidos, armazenar informações/transações, ter *logs* de todas as transações realizadas, bem como aceder à informação que lá está armazenada. Desta forma, esta ferramenta é importante durante a fase de desenvolvimento de *smart contracts*, pois agiliza o processo de testes/implementação dos mesmos. No entanto, não é a tecnologia mais adequada para a simulação real, pois o Ganache não realiza efetivamente o processo de *mining* – permite ter a funcionalidade *automine* (onde os dados são inseridos instantaneamente) e permite também estipular um período (definido pelo utilizador) que seria supostamente a duração do processo de *mining*. Desta forma, esta foi uma tecnologia utilizada apenas durante a fase de desenvolvimento e validação dos *smart contracts* da *framework*.

3.2.3 Node.js

Node.js permite criar aplicações em JavaScript para correr *standalone* numa máquina, não sendo necessário recorrer a um browser para a executar. Assim, é possível afirmar que Node.js é um ambiente de execução JavaScript *server-side* [72].

Na *framework* existia a necessidade de comunicar com o cliente da TTN que está desenvolvido em Node.js e também era necessário utilizar a Web3.js (secção 3.2.4) para fazer a comunicação das aplicações com a Blockchain. Desta forma, surgiu Node.js por forma a realizar estes dois tipos de ligações que as aplicações necessitam.

3.2.4 Web3.js

A Web3.js possui uma coleção de bibliotecas que permite desenvolver clientes que interagem com uma Blockchain Ethereum (web3-eth) e, desta forma, é possível executar ações como, por exemplo, enviar *Ethers* de uma conta para outra ou ler e gravar dados em *smart contracts* [73].

Em desenvolvimento web é usual utilizarem-se pedidos Ajax para ler e escrever dados no servidor web. A Web3.js tem um conceito bastante semelhante, pois é através desta coleção de bibliotecas que poderemos efetuar essas ações na Blockchain Ethereum [73] [71].

A Web3.js e a Blockchain Ethereum comunicam através de JSON RPC (*Remote Procedure Call*) - tecnologia popular para implementar o modelo cliente servidor onde o cliente faz a chamada ao procedimento remoto e envia uma mensagem para um servidor de forma a executar o procedimento específico [73].

Dadas as suas características e o seu conceito, esta tecnologia está a ser utilizada na Sensor Application e na WebApplication da BC4IoT – aplicações que necessitam de fazer a ligação com Ethereum através de JavaScript.

3.2.5 Geth

O Go Ethereum (Geth) é uma implementação de Ethereum na linguagem de programação Go, uma das três implementações originais de Ethereum, juntamente com C++ e Python [74]. O Geth é o cliente CLI autónomo do Go Ethereum e é o cliente de *software* mais popular para executar um nó na rede Ethereum. A execução de um nó permite que os utilizadores realizem transações e interajam com contratos inteligentes na Blockchain Ethereum [74]. O Geth foi utilizado no projeto por forma a realizar os testes de performance a uma rede Blockchain real e de forma descentralizada (secção 4.4.4) – em vez de utilizar uma rede Blockchain simulada (Ganache).

3.3 Características globais da framework BC4IoT

De uma forma geral, a *framework* proposta neste relatório (BC4IoT) tem as seguintes características:

- É um intermediário entre a rede TTN e a Blockchain;
- Não é necessário efetuar alterações na rede TTN;
- Permite alterar a validação de dados de acordo com as regras de negócio pretendidas;
- É facilmente escalável. Ao existir um novo sensor, o mesmo deve ser acrescentado a uma Sensor Application. Pode, eventualmente, ser acrescentada uma nova Sensor Application;

- A Sensor Application poderá estar do lado de quem adquiriu o sensor, mas um sensor também poderá ser associado a outra Sensor Application. Desta forma, quem adquire um sensor tanto poderá configurar e executar a Sensor Application do seu lado, como poderá solicitar para que o seu sensor seja associado a outra Sensor Application que esteja a ser executada;
- As instâncias da Sensor Application poderão também estar presentes nos nós da Blockchain;
- A forma como são distribuídas as instâncias da Sensor Application é uma decisão dos responsáveis pela gestão do sistema, contudo, o presente sistema permite qualquer um destes cenários;
- Através da Web Application é possível verificar os dados presentes na Blockchain, se os mesmos são ou não válidos, bem como analisar o evoluir dos resultados e, dessa forma, auxiliar as tomadas de decisão relativamente ao negócio em questão.

A BC4IoT pode ser adaptada a diferentes casos de estudo/negócios. Para isso, é necessário adaptar o *smart contract* utilizado de forma a dar resposta ao pretendido, tendo de passar pelas seguintes fases:

- Definir a estrutura de dados a utilizar;
- Definir e programar as regras de negócio essenciais para o *smart contract*;
- Definir e programar os métodos necessários para a comunicação com as restantes aplicações (como por exemplo a Sensor Application e a Web Application).

No próximo capítulo é apresentado um caso de estudo onde são realizadas as adaptações necessárias à *framework* BC4IoTpara um sistema de recolha de dados ambientais em ambiente urbano.

4 Caso de estudo - Recolha e validação de dados ambientais em ambiente urbano

As alterações climáticas e a poluição presente nos grandes centros urbanos são assuntos que estão na ordem do dia, pois as condições climatéricas estão a agravar-se cada vez mais. Assim, os estudos climáticos referentes às grandes metrópoles têm-se intensificado. Contudo, muitas das vezes existem algumas dúvidas relativamente à veracidade dos mesmos principalmente porque são normalmente lançados por uma das partes interessadas [75].

É neste contexto que este capítulo apresenta um caso de estudo aplicando a *framework* BC4IoT, descrevendo as suas principais características, os seus objetivos, bem como o ambiente de testes utilizado.

4.1 Objetivo

Tal como referido na secção 1.2, o projeto descrito neste documento teve como principal objetivo a criação de uma arquitetura capaz de integrar os sensores de uma rede IoT e de armazenar os dados recolhidos por estes num local seguro e inviolável onde os utilizadores possam ter confiança nos dados consultados - tendo assim a certeza de que os mesmos não foram alterados após a sua recolha.

Outro objetivo passou por conceber uma arquitetura versátil ao ponto de ser adaptável a diversas aplicações das mais diversas áreas que necessitem da integração de sensores IoT - saúde, agricultura, cidades inteligentes, seguradoras, entre outras.

Partindo do caso de estudo apresentado (recolha de dados ambientais), o projeto desenvolvido resultou na criação de um sistema público, fiável e democrático onde seja possível recolher informações ambientais e armazená-las num local inviolável de forma a acrescentar alguma confiança, conseguindo assim resolver os problemas de confiança dos utilizadores nos dados apresentados.

A ideia concreta passa por ter uma comunidade onde qualquer pessoa pode aderir e, desta forma, após terem adquirido um/vários sensores ambientais (temperatura, pressão atmosférica, humidade, etc.) ou estação meteorológica e de os ter ligado a uma rede de sensores TTN, pode ligar-se à Blockchain onde os dados recolhidos são armazenados. Os dados recolhidos pelos sensores são públicos e podem ser acedidos por qualquer cidadão.

Na próxima secção são apresentados os procedimentos que os utilizadores terão de seguir para se poderem juntar à comunidade criada a partir deste projeto.

4.2 Procedimentos para um utilizador se juntar à comunidade

O objetivo concreto deste caso de estudo passa pela criação de uma comunidade onde qualquer cidadão de uma cidade se pode juntar à mesma e, desta forma, contribuir através da recolha de dados meteorológicos de um sensor adquirido por si. Assim, o utilizador após adquirir o seu sensor poderá iniciar a sua colaboração das seguintes formas:

1. Fornecer os dados dos seus sensores a uma Sensor Application existente:
    - Registar o sensor na The Things Network;
    - Fornecer a *appKey*[14] e *appEui*[15] ao administrador do sistema;
    - A Sensor Application selecionada (podendo estar a correr tanto num nó da rede Blockchain como num outro participante na rede) é configurada e as recolhas passarão a ser transmitidas para a Blockchain onde serão armazenadas.
2. Ligar os seus sensores à sua própria Sensor Application:
    - Adquirir gratuitamente o software responsável pela integração do seu sensor com a rede Blockchain - Sensor Application;
    - Registar o sensor na The Things Network;
    - Configurar a Sensor Application com a *appKey* e *appEui* do seu sensor;
    - Correr a Sensor Application num computador seu - automaticamente as recolhas começarão a ser guardadas na Blockchain.
3. Adicionar um nó à rede Blockchain e correr a sua própria Sensor Application
    - Solicitar aos gestores da rede Blockchain para ser adicionado e configurado um novo nó na rede;
    - Registar o sensor na The Things Network;
    - Configurar a Sensor Application com a *appKey* e *appEui* do seu sensor;
    - Correr a Sensor Application.

Este caso de estudo conta com a validação de dados meteorológicos recorrendo a um *smart contract*. Os métodos de validação desses dados são apresentados na próxima secção.

[14] AppKey é a chave que é utilizada para proteger a comunicação entre o dispositivo e a rede [88].
[15] Identificador único (64 bits) para identificar as aplicações na LoRaWAN e na TTN [89].

4.3 Validação de dados meteorológicos

Os dados ambientais (temperatura, humidade, precipitação, etc.) antes de serem armazenados necessitam de serem validados, pois pode haver erros nos sensores, ou mesmo adulteração dos dados antes de os inserir na Blockchain. Por isso, é necessário verificar, por exemplo, se uma temperatura está correta (ex: se não tem valores impossíveis, como 100ºC, ou uma variação de 20ºC em 30min.)

Para a validação dos dados recolhidos pelos sensores pertencentes à rede IoT, o *smart contract* que está na Blockchain contém um algoritmo que permite verificar se os registos são ou não válidos.

Desta forma, foi realizada uma pesquisa acerca de possíveis algoritmos de validação de dados meteorológicos e foram encontrados os seguintes algoritmos:

1. Guidelines on validation procedures for meteorological data from automatic weather stations [76] – Estudo realizado em Andalusia (sul de Espanha) e analisa a radiação solar, a temperatura do ar, a humidade relativa do ar, a velocidade do vento, a precipitação;
2. Preliminary Data Validation and Reconstruction of Temperature and Precipitation in Central Italy [77] – Estudo realizado em Itália e analisa a temperatura e a precipitação;
3. Guidelines on Quality Control Procedures for Data from Automatic Weather Stations [78] – Estudo realizado na Eslováquia e analisa a temperatura do ar, a temperatura do ponto de orvalho, a temperatura do solo, a temperatura da terra, a humidade relativa, a pressão atmosférica, a direção do vento, a radiação solar e a intensidade da precipitação.

Na Tabela 4-1 é feita a comparação entre os algoritmos apresentados.

Após uma análise destes algoritmos de validação, foi selecionado o algoritmo 1, pois permite validar os principais dados recolhidos para este caso de estudo (temperatura e humidade). Houve ainda outro fator para a sua escolha que foi o facto de os *datasets* utilizados na fase de testes da *framework* BC4IoT incluírem os dados ambientais (temperatura e humidade) que são consideradas neste algoritmo.

No entanto, é de realçar que, apesar de ter sido selecionado este algoritmo, poderá ser utilizado qualquer outro de forma a existir a validação de outros tipos de dados, bem como para se adaptar a outros casos de estudo. A melhor solução para a validação deste tipo de dados passa pela integração de uma equipa especializada na área na fase de especificação e desenvolvimento dos *smart contracts* por forma a obter um algoritmo adaptado para o caso e em conformidade com todas as variáveis em questão (nº de sensores, intervalos de recolha, o clima onde as recolhas estão a ser efetuadas, entre outros fatores que influenciam a análise dos dados).

Tabela 4-1 - Comparação de algoritmos de validação de dados meteorológicos

Algoritmo	Variáveis	Processos de validação	Periodicidade das recolhas
1 - Guidelines on validation procedures for meteorological data from automatic weather stations [76]	- Radiação Solar; - Temperatura do ar; - Humidade relativa do ar - Velocidade do vento - Precipitação	a) Range test b) Step test c) Internal consistency test d) Persistence test e) Spatial consistency test	30 minutos
2 - Preliminary Data Validation and Reconstruction of Temperature and Precipitation in Central Italy [77]	- Temperatura - Precipitação	- Verificação de erros brutos - Verificação de consistência interna - Teste de tolerância - Consistência temporal - Consistência espacial.	Não é indicado
3 - Guidelines on Quality Control Procedures for Data from Automatic Weather Stations [78]	- Temperatura do ar - Temperatura do ponto de orvalho - Temperatura do solo - Temperatura da terra - Humidade relativa - Pressão atmosférica no nível da estação - Direção do vento - Velocidade do vento - Radiação solar - Intensidade de precipitação	**a) Plausible value check** **b) Time consistency check** - Check on a maximum allowed variability of an instantaneous value (a step test) - Check on a minimum required variability of instantaneous values - Internal consistency check	Não é indicado

Na Tabela 4-1 são indicadas as variáveis que são avaliadas nos algoritmos analisados de uma forma genérica. No caso do algoritmo escolhido, a descrição mais pormenorizada das variáveis é a seguinte:

- Radiação solar
    - Diariamente (Rs)
    - Semihourly (Rssh)
- Temperatura do ar
    - Média diária (Tm)
    - Máximo (Tx)
    - Mínimo (Tn)
- Humidade relativa do ar
    - Média diária (RHm)

    - Máximo (RHx)
    - Mínimo (RHn)
    - Semihourly (RHsh)
- Velocidade do vento
    - Média diária (Um)
    - Máximo (Ux)
    - Semihourly (Ush)
- Precipitação
    - Quantidade diária (P)
    - Semihourly (Psh)

Tal como apresentado na Tabela 4-1, o algoritmo selecionado utiliza diferentes processos de avaliação (Tabela 4-3).

O processo a) “range test” compara o valor de uma variável meteorológica com valores extremos estabelecidos. Qualquer observação que ocorra fora do intervalo aceitável é considerada incorreta e não é validada pelos próximos testes. Existem dois tipos de intervalos:

- Fixo - propostos para diferentes variáveis são baseados nos valores apresentados na Tabela 4-2 e nas faixas propostas por alguns autores (Tabela 4-3);
- Dinâmico - os limites dinâmicos para cada variável meteorológica são baseados em valores extremos medidos para cada local ou no extremo teórico possível para cada local e período de tempo - ter referência por exemplo o IPMA (Instituto Português do Mar e da Atmosfera). Os dados rejeitados são identificados como suspeitos e devem ser verificados manualmente (aplicado ao caso de estudo, terá de ser numa fase antes do *deploy* ou fazer-se numa fase de análise dos dados da Blockchain). Se um dado suspeito (potencial discrepante) for verificado corretamente, ele será armazenado como um novo valor extremo para o local correspondente.

O processo b) “step test” tem como base a consistência dos dados ao longo do tempo. Compara a diferença entre valores sucessivos. Se a diferença exceder um valor permitido, o mesmo será marcado como um registo suspeito. Portanto este teste verifica a taxa excessiva de alteração entre dois valores consecutivos. Os valores utilizados estão presentes na Tabela 4-3.

O processo c) “internal consistency test” baseia-se na consistência de cada parâmetro observado ou na relação entre duas variáveis medidas. Várias variáveis meteorológicas medidas no mesmo local e horário devem ser meteorologicamente consistentes entre si. Caso contrário, ambas as observações são consideradas suspeitas. Por exemplo, um valor

médio será sempre menor que o valor instantâneo máximo, ou a quantidade de precipitação ocorrida em 6 h será sempre menor ou igual à quantidade de precipitação em 24h.

O processo d) “persistence test” verifica a variabilidade das medições. Quando um sensor falha, geralmente produz um valor constante, portanto o desvio padrão fica baixo. Quando o sensor estiver desligado durante um o período em que o desvio padrão é analisado, o seu desvio padrão será zero. Se o desvio padrão estiver abaixo de um mínimo aceitável, os valores são indicados como suspeitos. Como o teste de persistência usa estatísticas agregadas, não é possível indicar quais as observações que no período são responsáveis por esse alerta. Portanto, todos os dados desse período são indicados como suspeitos ou com um aviso.

A Tabela 4-2 apresenta a especificação dos sensores utilizados durante a realização dos testes efetuados no artigo [76].

Tabela 4-2 - Especificação dos sensores utilizados durante a realização dos testes efetuados no artigo [76]

Sensor	Variables	Accuracy	Range
Pt1000	Temperature	±0.2 °C (20 °C)	−39.2/60 °C
Humicap 180	Relative Humidity	±2% (0–90%) ±3% (90–100%)	0.8/100%
Young 05103	Wind Speed	±0.3 m s^{-1} (1–60 m s^{-1})	0/60 m s^{-1}
	Wind direction	±3°	0/360°
Skye SP1110	Solar radiation	±5%	350–1100 nm
ARG 100	Precipitation	0.2 mm/tip	

Na próxima secção são descritas as fases de desenvolvimento da *framework* BC4IoT com incidência no caso de estudo apresentado, bem como os testes realizados nas diferentes fases.

4.4 Desenvolvimento e testes do caso de estudo

O desenvolvimento da *framework* BC4IoT (capítulo 3) foi iterativo, testando várias hipóteses e analisando os seus resultados. Assim, foi necessário criar vários ambientes de teste para que se pudesse abranger todos os cenários onde a mesma poderá ser aplicada no caso de estudo selecionado.

As fases 1 (secção 4.4.1), 2 (secção 4.4.2) e 3 (secção 4.4.3) foram essencialmente fases de desenvolvimento e testes nas quais era necessário obter resultados rapidamente. Por forma a ter um processo de desenvolvimento mais fluido, foi utilizada a rede Blockchain Ganache (secção 3.2.2), pois, devido às suas características é muito simples e prático fazer o *deploy* de *smart contracts* e obter resultados rapidamente.

Tabela 4-3 - Processos de validação dos dados mateológicos [76]

Validation procedures	Global radiation daily (MJ m^2 d^{-1}) semihourly (W m^{-2})	Relative humidity (%)	Air temperature (°C)	Wind speed (m s^{-1})	Precipitation (mm)
Range test	$-1 < Rs_{sh} < 1500$ (Shafer et al., 2000); $0.03R_a \leqslant Rs, Rs_{sh} \leqslant R_a$; $Rs, Rs_{sh} < 1.1R_{so}$ (Allen, 1996; Geiger et al., 2002; Moradi, 2007)	$0.8 < RH < 103$ (Table 1) (Shafer et al., 2000)	$-30 < T < 50$ (Shafer et al., 2000); $T_{LOW} < T < T_{HIGH}$ (AEMET, 2008)	$0 < U, U_{sh} < 60.3$ (Table 1); $0 < U_x < 100$ (Shafer et al., 2000)	$0 \leqslant P_{sh} \leqslant 120$ (Zahumensky, 2004); $0 \leqslant P < 508$ (Shafer et al., 2000); $0 \leqslant P, P_{sh} \leqslant P_{MAX}$ (AEMET, 2008)
Step test	$0 \leqslant \lvert Rs_{sh} - Rs_{sh\text{-}2}\rvert \leqslant 555$ (Meek and Hatfield, 1994)	$\lvert RH_{sh} - RH_{sh\text{-}1}\rvert < 45$ (Zahumensky, 2004)	$\lvert T_{sh} - T_{sh\text{-}2}\rvert < 4$; $\lvert T_{sh} - T_{sh\text{-}4}\rvert < 7$; $\lvert T_{sh} - T_{sh\text{-}6}\rvert < 9$; $\lvert T_{sh} - T_{sh\text{-}12}\rvert < 15$; $\lvert T_{sh} - T_{sh\text{-}24}\rvert < 25$; (WMO, 1993)	$\lvert U(d) - U(d-1)\rvert < 10$; $\lvert U_{sh}\text{-}U_{sh\text{-}2}\rvert < 10$; (Meek and Hatfield, 1994)	
Internal consistency test		$RH_x > RH_m > RH_n$ (Reek et al., 1992; Feng et al., 2004) $RH_x > \max(RH_{sh})$; $RH_n < \min(RH_{sh})$ (Vejen et al., 2002)	$T_x > T_m > T_n$; $T_x(d) > T_n(d-1)$; $T_n(d) \leqslant T_x(d-1)$ (Reek et al., 1992; Feng et al., 2004) $T_x > \max(T_{sh})$ $T_n < \min(T_{sh})$ (Vejen et al., 2002)	Speed = 0 and direction = 0; speed ≠ 0 and direction ≠ 0 (DeGaetano, 1997; Zahumensaky, 2004) $U_x(d) > U(d)$; $U_x(d) > \max(U_{sh})$ (Vejen et al., 2002)	Diurnal precipitation events are true if: $K_T^* < 0.5$ *and* $RH^* > 80\%$ (Estévez, 2008) $P_{sh(0-3h)} \leqslant P_{sh(0-6h)}$; $P_{sh(0-12h)} \leqslant P_{sh(0-24h)}$ (Vejen et al., 2002)
Persistence test	$Rs(d) \neq Rs(d-1) \neq Rs(d-2)$; $Rs_{sh} \neq Rs_{sh\text{-}2} \neq Rs_{sh\text{-}4} \neq Rs_{sh\text{-}6}$ (Meek and Hatfield, 1994)	$RH(d) \neq RH(d-1) \neq RH(d-2)$ (Meek and Hatfiled, 1994) $\sigma RH_{sh} > 1$ (Zahumensky, 2004)	$T(d) \neq T(d-1) \neq T(d-2)$; $T_{sh} \neq T_{sh\text{-}2} \neq T_{sh\text{-}4} \neq T_{sh\text{-}6}$ (Meek and Hatfield, 1994)	$U(d) \neq U(d-1) \neq U(d-2)$; $U_x(d) \neq U_x(d-1) \neq U_x(d-2)$; $U_{sh} \neq U_{sh\text{-}2} \neq U_{sh\text{-}4} \neq U_{sh\text{-}6}$ (Meek and Hatfield, 1994)	
Spatial consistency test			$T'\text{-}fs' \leqslant T^* \leqslant T' + fs'$ where $T^* = T_x, T_n, T_m$; $T' = T'_x, T'_n, T'_m$ (reference series); s' = weighting root-mean-square error; f = factor		

Rs_{sh}: semihourly global radiation (W m^{-2}); Rs: daily global radiation (MJ m^2 d^{-1}); R_{so} : daily or semihourly global radiation under clear sky conditions (MJ m^2 d^{-1} or W m^{-2}); R_a: daily or semihourly extraterrestrial radiation (MJ m^2 d^{-1} or W m^{-2}); RH: daily maximum, minimum or mean relative humidity (%); RH_{sh}: semihourly relative humidity (%); T: daily or semihourly temperature (°C); T_{LOW}: minimum temperature value measured in long-term observatory closer to each station (°C); T_{HIGH}: maximum temperature value measured in long-term observatory closer to each station (°C); T_m: daily mean temperature (°C); T_x: daily maximum temperature (°C); T_n: daily minimum temperature (°C); T_{sh}: semihourly temperature (°C); U: daily mean wind speed (m s^{-1}); U_x: daily maximum wind speed (m s^{-1}); U_{sh}: semihourly wind speed (m s^{-1}); P: daily precipitation (mm); P_{sh}: semihourly precipitation (mm); $P_{sh\ (0\text{-}Xh)}$: amount of semihourly precipitation from 0 to Xh (mm); P_{MAX}: daily or semihourly extreme value measured in long-term observatory closer to each station; K_T^*: mean "clearness" index during diurnal precipitation event; RH*: mean relative humidity during diurnal precipitation event; d: day d; d-1: day before day d; d-2: day before day d-1; sh: semihour sh; sh-1: semihour before 30 minutes; sh-2: semihour before one hour; sh-4: semihour before two hours; sh-6: semihour before three hours, etc.

σ_{RHsh}: standard deviation of semihourly relative humidity data.

Na fase 4 (secção 4.4.4), como o objetivo passava por testar a viabilidade da utilização de Blockchain num contexto real, foi necessário utilizar uma infraestrutura real em que pudesse ser feita essa análise.

4.4.1 Primeira fase – infraestrutura inicial para inserção de dados na Blockchain

Numa fase inicial o objetivo era testar a forma como os dados eram tratados de forma a selecionar apenas os dados essenciais vindos do sensor (temperatura, luminosidade, bateria) e posteriormente armazená-los na Blockchain. Assim, o ambiente de testes consistiu em ter:

- um sensor ligado à rede LoRaWAN a recolher dados;
- uma máquina a correr a Sensor Application;
- uma rede Blockchain para testes (Ganache);
- uma máquina a correr a Web Application.

A Figura 4-1 apresenta um exemplo do ambiente de testes montado.

Figura 4-1 - Primeira fase: Ambiente de testes

Nesta fase é utilizado um sensor (The Things Node) que recolhe informação relativamente à temperatura, à humidade e à luminosidade. Este sensor foi registado na TTN. Após o registo, o sensor faz recolhas a cada 30 segundos e os dados são publicados na TTN.

A Sensor Application também já se encontra desenvolvida nesta fase. Esta está à escuta dos dados vindos da TTN (bateria, temperatura, humidade e luminosidade), trata-os e envia-os para a Blockchain. Na Figura 4-2 podemos verificar a mensagem vinda da TTN recebida na Sensor Application com os dados de uma recolha do sensor.

```
DevId ttn_node
payload =  { app_id: 'ttn_application',
  dev_id: 'ttn_node',
  hardware_serial: '0004A30B00200909',
  port: 1,
  counter: 0,
  payload_raw: <Buffer 00 01 02 03 04 05>,
  payload_fields:
   { battery: 1, event: 'setup', light: 515, temperature: 10.29 },
  metadata: { time: '2019-10-01T20:47:40.100992331Z' } }
```

Figura 4-2 - Mensagem enviada pela TTN para a Sensor Application

Na Blockchain está armazenado um *smart contract* desenvolvido na linguagem Solidity. Nesta fase é ainda muito simples e o seu papel é apenas de armazenar na estrutura de dados lá definida, os dados vindos da Sensor Application não havendo qualquer validação antes do armazenamento. Assim, o *smart contract* possui uma estrutura de dados (Figura 4-3) que representa a informação vinda de uma recolha de um sensor (id – interno, valor da temperatura, valor da luminosidade, o valor da bateria que ainda tem disponível, o evento que despoletou a recolha do sensor (se foi forçada – 'button' - ou se foi nos intervalos pré-definidos – 'interval'), o id do sensor e a data em que foi feito o registo).

```
uint public dataId = 0;

struct dataFromSensor {
    uint id;
    uint temperature;
    uint light;
    uint battery;
    string sensorEvent;
    string devId;
    string date;
}
```

Figura 4-3 - *Smart contract*: estrutura de dados

As várias recolhas de um sensor (ou seja, um conjunto de estruturas de dados) encontram-se armazenadas num *mapping* (Figura 4-4) – *array* associativo em que a *key* é um *unsigned int* e o conteúdo do *array* é uma instância da estrutura "dataFromSensor" (Figura 4-3).

```
mapping (int => dataFromSensor) public dataFromSensorArray;
```

Figura 4-4 - *Array* associativo de dataFromSensor

Quando a Sensor Application pede o armazenamento dos dados do sensor, invoca a função "createDataSensor" (Figura 4-5). Esta função recebe os dados da Sensor Application, cria um objeto da estrutura de dados "dataFromSensor" (Figura 4-3) e adiciona-a ao *mapping* "dataFromSensorArray" (Figura 4-4).

```
function createDataSensor (uint _temperature, uint _light, uint _battery, string memory _sensorEvent, string memory _devId, string
    //DATA ID
    dataId ++;

    //add a instance of dataFromSensor to dataFromSensorArray
    dataFromSensorArray[dataId] = dataFromSensor(dataId, _temperature, _light, _battery, _sensorEvent, _devId, _date);

    emit dataFromSensorCreated(dataId, _temperature, _light, _battery, _sensorEvent, _devId, _date);
}
```

Figura 4-5 - *Smart contract*: Criação de instância da estrutura de dados

Ainda nesta fase também se encontra desenvolvida a Web Application que permite visualizar os dados que estão armazenados. A Web Application obtém os dados do *smart contract* – um *array* com as estruturas de dados que estão armazenadas no *mapping* (Figura 4-4) – e trata-os de forma a serem apresentados ao utilizador de uma forma apelativa.

A Web Application nesta fase ainda era muito simples consistindo apenas numa tabela onde eram apresentados os valores e onde era possível pesquisá-los e atualizá-los de acordo com os resultados vindos da Blockchain (Figura 4-6).

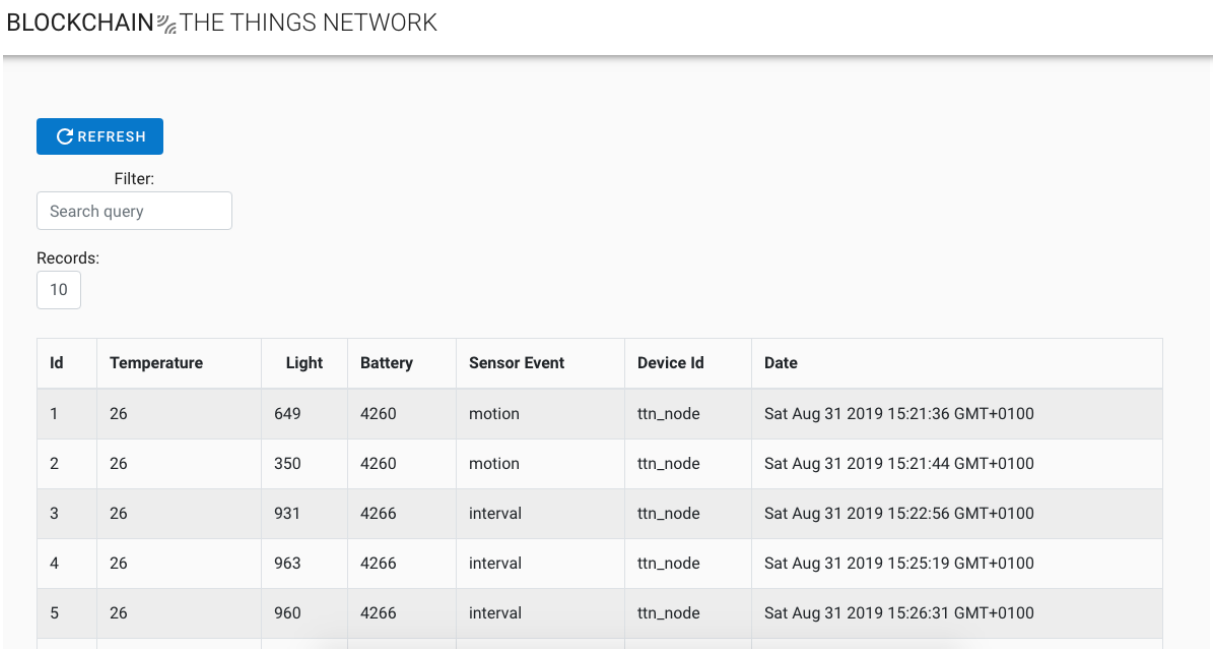

Figura 4-6 - Web Application na primeira fase

No final desta fase, os princípios básicos da *framework* BC4IoT estão implementados, pois existe a integração entre a rede IoT (TTN) e a rede Blockchain (Ganache), existindo também um *smart contract* com uma estrutura de dados para os armazenar os dados recolhidos. Esta integração passava pela utilização da Sensor Application. Nesta fase estava ainda implementada uma primeira versão da Web Application que permitia visualizar

os dados que tinham sido inseridos na Blockchain. Faltava aplicar as validações de dados meteorológicos e aplicar numa rede Blockchain real.

4.4.2 Segunda fase – adição de validações simples de dados meteorológicos

Nesta fase são necessários todos os componentes que foram utilizados na fase anterior – o sensor da TTN, a Sensor Application, a rede Blockchain (Ganache), o *smart contract* e a Web Application. A principal diferença entre estas fases consiste na adição de algumas das validações dos dados meteorológicos no *smart contract*. Por forma a manter o histórico e preparando já a estrutura de dados para uma possível deteção de dispositivos "maliciosos" ou com comportamento erróneo, foi criada mais uma lista no *smart contract* (*mapping* – semelhante à utilizada na fase anterior), existindo agora duas - uma para os dados que foram considerados válidos ("dataFromSensorArray" - Figura 4-4) e outra para os que foram classificados como inválidos ("dataFromSensorErrorArray" - Figura 4-7).

```
mapping (int => dataFromSensor) public dataFromSensorErrorArray;
```

Figura 4-7 - Array associativo de dataFromSensor para dados classificados como inválidos

A Figura 4-8 apresenta o algoritmo básico de validação utilizado nesta fase, sendo possível verificar que, consoante o resultado da validação, os dados são introduzidos na estrutura de dados (*mapping*) com os registos considerados válidos ("dataFromSensor") ou inválidos ("dataFromSensorErrorArray").

Este algoritmo vai verificar se em cada um dos dados recolhidos não há uma variação elevada – tanto para metade em relação ao último registo, como para o dobro também em relação ao último registo. Ou seja, o algoritmo não permite registar como válido um caso onde, por exemplo, a temperatura em 30 minutos (período em que são feitas as recolhas) varie de 20º para 10º ou de 20º para 40º.

A Web Application nesta fase também foi atualizada por forma a permitir filtrar os dados de acordo com as validações (válido ou inválido). São ainda apresentados gráficos de forma ao utilizador poder fazer uma melhor análise visual dos dados recolhidos pelos sensores (Figura 4-9).

No final desta fase, a *framework* BC4IoT já possuia algumas validações básicas dos dados que estão a ser recolhidos pelo sensor da TTN. Ainda nesta fase a Web Application também sofreu melhorias e adaptações nomeadamente na filtragem dos dados consoante as validações implementadas no *smart contract* e foram acrescentados gráficos por forma a ser possível ao utilizador fazer uma análise mais pormenorizada dos mesmos. Faltava aplicar as validações de dados meteorológicos de acordo com o algoritmo selecionado (secção 4.3) e aplicar numa rede Blockchain real.

```
function createDataSensor (uint _temperature, uint _light, uint _battery, string memory _sensorEvent, string memory _devId, string memory _date) public {
    bool validateValues = true;

    if(dataId >= 1) {
        //TEMPERATURE
        uint halfTemperatureSensorData = dataFromSensorArray[dataId].temperature/2;
        uint doubleTemperatureSensorData = dataFromSensorArray[dataId].temperature*2;

        ///LIGHT
        uint halfLightSensorData = dataFromSensorArray[dataId].light/2;
        uint doubleLightSensorData = dataFromSensorArray[dataId].light*2;

        //BATTERY
        uint halfBatterySensorData = dataFromSensorArray[dataId].battery/2;
        uint doubleBatterySensorData = dataFromSensorArray[dataId].battery*2;

        //VALIDATE IF VALUES IS BETWEEN DEFINED VALUES
        if ((_battery > halfBatterySensorData && _battery < doubleBatterySensorData) &&
            (_temperature > halfTemperatureSensorData && _temperature < doubleTemperatureSensorData) &&
            (_light > halfLightSensorData && _light < doubleLightSensorData)) {
            validateValues = true;
        }
        else {
            validateValues = false;
        }
    }

    if(_battery > 0 && validateValues == true) {
        //DATA ID
        dataId ++;

        //add a instance of dataFromSensor to dataFromSensorArray
        dataFromSensorArray[dataId] = dataFromSensor(dataId, _temperature, _light, _battery, _sensorEvent, _devId, _date);

        emit dataFromSensorCreated(dataId, _temperature, _light, _battery, _sensorEvent, _devId, _date);
    }
    else {
        //DATA ID
        dataIdError ++;

        //add a instance of dataFromSensor to dataFromSensorArray
        dataFromSensorErrorArray[dataIdError] = dataFromSensor(dataId, _temperature, _light, _battery, _sensorEvent, _devId, _date);

        emit dataFromSensorCreated(dataId, _temperature, _light, _battery, _sensorEvent, _devId, _date);
    }
```

Figura 4-8 - Segunda fase: Validação de dados no smart contract

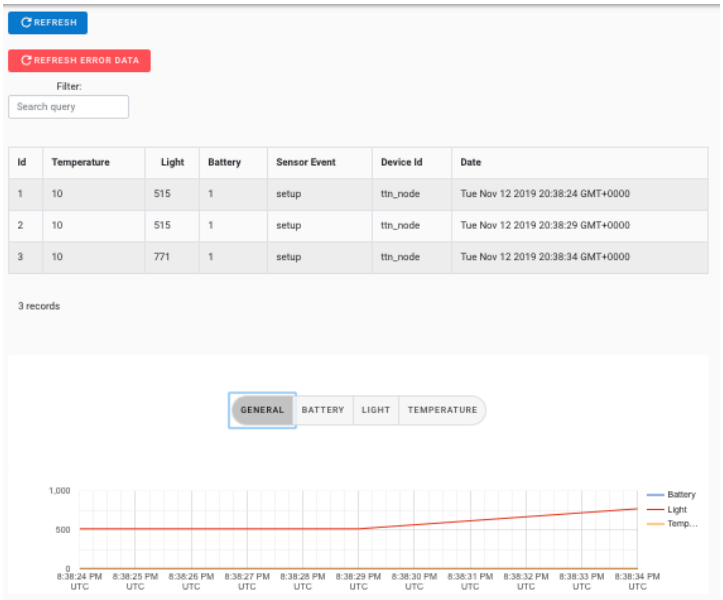

Figura 4-9 - Web Application na segunda fase

4.4.3 Terceira fase – teste com *dataset* real

O objetivo desta fase foi de programar algumas das validações presentes no algoritmo de validação de dados meteorológicos selecionado (secção 4.4.3). Como não era possível ter um grande número de sensores, foi criado um simulador (uma aplicação em Node.js) que substitui a rede de sensores – TTN – e insere valores predefinidos na rede Blockchain (Ganache).

Para testar as validações programadas, foi utilizado um *dataset* real[16] da cidade de Milão constituído por dados recolhidos por 5 sensores dispersos pela cidade (Brera, Filippo Juvara, Lambrate, Marche e Zavattari). Estes dados foram introduzidos pelo simulador referido anteriormente. Este *dataset* apenas tem dados referentes à humidade e à temperatura recolhidos de hora a hora.

A Figura 4-10 apresenta a distribuição geográfica dos sensores do *dataset* na cidade de Milão.

Figura 4-10 - Distribuição geográfica dos sensores do dataset utilizado na fase de testes do algoritmo de validação

É de realçar que nestes testes apenas foram realizadas validações à temperatura e à humidade – as únicas variáveis que estão presentes no *dataset*. Foi ainda necessário fazer algumas adaptações relativamente ao algoritmo escolhido, pois as recolhas presentes no

[16] O *dataset* utilizado e o script de tratamento do mesmo para inserção no simulador encontram-se em: https://drive.google.com/drive/folders/1aguENUG_nDkFmzV1WYl3YcyWpz5p-3cG?usp=sharing

dataset utilizado foram obtidas de hora a hora e o algoritmo previa recolhas de 30 em 30 minutos – o algoritmo foi adaptado neste sentido. Não foi ainda implementada a validação espacial, pois não estavam presentes todos os dados necessários para a sua implementação. Desta forma, estas adaptações podem influenciar os resultados dos testes realizados. Tal como referido na secção 4.3, os *smart contracts* permitem a implementação de qualquer tipo de algoritmos e a melhor forma de avaliar a fiabilidade dos algoritmos passa por ter uma equipa especializada na área de forma a avaliar todas as vertentes que possam influenciar as validações a terem um melhor resultado.

De forma a testar o algoritmo de validação de dados meteorológicos implementado, primeiro foi introduzido o *dataset* real e avaliado se todos os valores eram considerados válidos pelo *smart contract*. Depois foram injetadas falhas no *dataset*, produzidos alguns registos inválidos, para verificar até que ponto os *smart contracts* os conseguiam identificar como tal. Desta forma, para a realização dos testes foram injetados os seguintes tipos de falhas:

- Os registos submetidos para a Blockchain e que são validados pelo *smart contract* são sempre submetidos em pares (temperatura, humidade) por forma a simular a inserção de um sensor real;
- Todos os testes iniciam com um *smart contract* vazio, deixando introduzir os primeiros 36 registos por motivos de validação dos dados históricos – o algoritmo de validação em algumas das suas validações necessita de dados das 36 horas anteriores;
- A **duração** das falhas injetadas nos dados dos sensores pode ser:
    - ***Permanent*** –todos os dados são adulterados de acordo com o tipo de falha;
    - ***Transient*** –os dados são adulterados 50% dos dados de forma aleatória;
- O **tipo** de falha pode ser:
    - ***Scale*** - Este tipo de teste tem como objetivo avaliar a resposta do algoritmo num caso onde – por exemplo – um sensor vai desviando progressivamente os valores medidos. O valor é alterado, aplicando uma escala X ao valor lido no período anterior, ou seja, NovoValor=ValorLido*X. X pode tomar os valores de: 0.5, 0.9, 0.99, 1.01, 1.1, 1.5 – desta forma, o valor inserido é alterado em -50%, -10%, -1%, 1%, 10% ou 50% do valor anterior;
    - ***Offset*** - O valor é alterado, aplicando uma escala X ao valor lido no período anterior, ou seja, NovoValor=ValorLido + X. X pode tomar os valores de: -5, -1, -0.2, 0.2, 1.5, 5 – desta forma, o valor inserido é somado aos valores de X descritos;
    - **Min** – Este teste passa por inserir um valor inferior ao valor mínimo de cada variável (temperatura e humidade) definido no “range test” da Tabela 4-3;
    - **Max** – Este teste passa por superior um valor inferior ao valor máximo cada variável (temperatura e humidade) definido no “range test” da Tabela 4-3.

Cada um dos testes foi realizada apenas uma repetição, pois o principal objetivo do trabalho realizado não passa por justificar a qualidade dos algoritmos, mas por identificar apenas alguns registos que estejam errados (principalmente erros “grosseiros”, onde a discrepância é elevada).

Numa primeira fase, foi inserido o *dataset* com todos os registos de temperatura e humidade vindos dos 5 sensores – num total de 6758 registos - e passaram sem registos inválidos. No caso dos testes *Transient* foram injetadas 3379 falhas e nos testes *Permanent* os dados começaram logo a ser adulterados logo na fase inicial – após a inserção dos primeiros 36 registos.

4.4.3.1 Temperatura

Os testes realizados à variável da temperatura seguem as considerações anteriormente descritas. Podemos observar na Tabela 4-4 e na Tabela 4-5 os resultados dos testes realizados utilizando o mesmo *dataset*, mas com a injeção de falhas.

Tabela 4-4 - Teste *transient* à variável da temperatura

Duração	Tipo	Resultado
Transient	Min	Inválido
Transient	Max	Inválido
Transient	Scale X (0,5)	Inválido
Transient	Scale X (0,9)	Válido
Transient	Scale X (0,99)	Válido
Transient	Scale X (1,01)	Válido
Transient	Scale X (1,1)	Válido
Transient	Scale X (1,5)	Inválido
Transient	Offset X (-5)	Inválido
Transient	Offset X (-1)	Válido
Transient	Offset X (-0,2)	Válido
Transient	Offset X (0,2)	Válido
Transient	Offset X (1)	Válido
Transient	Offset X (5)	Inválido

Tabela 4-5 - Teste *permanent* à variável da temperatura

Duração	Tipo	Resultado
Permanent	Scale X (0,5)	Inválido
Permanent	Scale X (0,9)	Válido
Permanent	Scale X (0,99)	Válido
Permanent	Scale X (1,01)	Válido
Permanent	Scale X (1,1)	Válido
Permanent	Scale X (1,5)	Inválido
Permanent	Offset X (-5)	Inválido
Permanent	Offset X (-1)	Válido
Permanent	Offset X (-0,2)	Válido
Permanent	Offset X (0,2)	Válido
Permanent	Offset X (1)	Válido
Permanent	Offset X (5)	Inválido

Através dos resultados dos testes podemos verificar que em nenhum dos casos o algoritmo permite que exista um “erro grosseiro”, pois quando os dados são adulterados de forma mais significativa é logo detetado que aquele registo é inválido. Contudo, podemos verificar que se as alterações forem feitas de forma gradual, ou seja, se houver alterações mínimas, o algoritmo deixa que os registos sejam registados como válidos.

4.4.3.2 Humidade

Os testes realizados à variável da humidade seguem as considerações anteriormente descritas.

Numa primeira fase, foi inserido o *dataset* com todos os registos de temperatura vindos dos 5 sensores – num total de 6758 registos - e passaram sem registos inválidos.

Como referido anteriormente, os dados são submetidos em pares (temperatura, humidade), assim, um registo, ao ser avaliado, passa sempre primeiro pelo algoritmo de validação da temperatura. Desta forma, existem valores que são logo registados como inválidos na validação da temperatura e não chegam à fase de validação da humidade.

Tabela 4-6 - Teste *transient* à variável da humidade

Duração	Tipo	Resultado
Transient	Min	Inválido
Transient	Max	Inválido
Transient	Scale X (0,5)	Válido
Transient	Scale X (0,9)	Válido
Transient	Scale X (0,99)	Válido
Transient	Scale X (1,01)	Válido
Transient	Scale X (1,1)	Válido
Transient	Scale X (1,5)	Válido
Transient	Offset X (-5)	Inválido
Transient	Offset X (-1)	Válido
Transient	Offset X (-0,2)	Válido
Transient	Offset X (0,2)	Válido
Transient	Offset X (1)	Válido
Transient	Offset X (5)	Inválido

Tabela 4-7 - Teste *permanent* à variável da humidade

Duração	Tipo	Resultado
Permanent	Scale X (0,5)	Válido
Permanent	Scale X (0,9)	Válido
Permanent	Scale X (0,99)	Válido
Permanent	Scale X (1,01)	Válido
Permanent	Scale X (1,1)	Válido
Permanent	Scale X (1,5)	Inválido
Permanent	Offset X (-5)	Inválido
Permanent	Offset X (-1)	Válido
Permanent	Offset X (-0,2)	Válido
Permanent	Offset X (0,2)	Válido
Permanent	Offset X (1)	Válido
Permanent	Offset X (5)	Inválido

À semelhança dos resultados obtidos nos testes realizados à variável da temperatura, aqui também podemos verificar que em nenhum dos casos o algoritmo permite que exista um "erro grosseiro", pois quando os dados são adulterados de forma mais significativa é logo detetado que aquele registo é inválido. Contudo, no caso da variável da humidade – à semelhança da variável da temperatura - também podemos verificar que se as alterações forem feitas de forma gradual, ou seja, se houver alterações mínimas, o algoritmo permite que os registos sejam registados como válidos. Desta forma, podemos concluir que o algoritmo implementado não é infalível e necessita de análises cuidadas caso a caso.

4.4.4 Quarta fase – Blockchain com múltiplos nós - Geth

Nesta fase pretende-se analisar a viabilidade da integração de Blockchain numa rede IoT - de baixo custo, baixa potência e de longo alcance. Para isso é necessário recorrer a um cenário em que seja utilizada uma rede real de Blockchain (Geth – secção 3.2.5), pois devido às limitações que o Ganache apresenta – nomeadamente não realizar efetivamente o processo de *mining* (secção 3.2.2) -, não é viável fazer os testes utilizando esta tecnologia.

Os testes desenvolvidos ao longo desta fase têm como principal intuito verificar a viabilidade da integração da tecnologia Blockchain numa rede IoT. Assim, o objetivo é testar os limites da rede Blockchain, de forma a avaliar o limite de transações que a rede Blockchain permite. Também importante avaliar o comportamento da rede no processamento de transações à medida que a mesma vai crescendo e qual a velocidade no processamento das transações realizadas. O desempenho das máquinas (nós) que compõem a rede também é importante por forma a verificar onde é que possa existir um maior esforço e/ou identificar o local onde o processamento das transações possa estar mais demorado.

4.4.4.1 Ambiente de testes

Para a realização dos testes de desempenho da Blockchain foi montada uma rede de até 5 máquinas e criados 14 cenários de teste, apresentados na Tabela 4-8. Ao longo dos testes a função das máquinas vai variando, ou seja, numa fase inicial começa-se apenas com um *miner*. Nos testes seguintes são acrescentados mais nós e *miners* por forma a avaliar o impacto que o aumento de cada tipo tem no processamento de transações, bem como no desempenho/esforço de cada máquina.

A rede Blockchain utilizada possui ainda *miner threads*. O número de *miner threads* do Geth consiste no número de *threads* a correrem em paralelo para o processo de *mining*.

Os componentes do ambiente de testes estão divididos em dois tipos: os nós que formam a rede Blockchain e uma máquina onde é corrida a Sensor Application. Tal como referido anteriormente, os nós vão sendo selecionados de acordo com o cenário de testes aplicado. Cada um dos nós está a correr numa máquina virtual e num computador diferente, pertencendo todos à mesma rede local.

Os dados inseridos pelas Sensor Applications são excertos do *dataset* utilizado nos testes da terceira fase (secção 4.4.3).

Desta forma, os componentes desde ambiente de testes possuem as seguintes características:

- Nó Blockchain
    - CPU: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz – 4 cores (máquina virtual);
    - RAM: 5214416 kB (máquina virtual);
    - SO: Lubuntu - Ubuntu 18.04.5 LTS.

- Máquina da Sensor Application
    - CPU: Intel Core i7-8850U;
    - RAM: 16GB;
    - SO: Windows 10.

Na próxima secção são apresentados os diversos cenários de teste e as suas características.

4.4.4.2 Cenários de teste

Por forma a avaliar as várias formas de inserção de dados na Blockchain, foram criados os cenários descritos na Tabela 4-8.

Tabela 4-8 - Cenários de teste

ID	Sensor Application Simulator (SAS)	Transactions (per SAS)	*Miners*	Nodes (*full nodes*)	Difficulty	Interval (ms)	Duration (min)	*Miner threads*[17]
Base	0	0	1	0	200000	-	1.5	*default*
Base_Difficulty_0	1	10	1	1	0	-	1.5	*default*
1	1	10	1	0	200000	-	-	*default*
2	1	10	1	1	200000	-	-	*default*
3	1	10	2	2	200000	-	-	*default*
4	1	10	3	2	200000	-	-	*default*
5	2	10	1	0	200000	-	-	*default*
6	2	10	2	2	200000	-	-	*default*
7	1	10	1	1	0	-	-	*default*
8	5	10	1	1	200000	-	-	*default*
9	5	-	1	1	200000	500	2	*default*
10	1	10	1	1	0	-	-	2
11	1	10	4	0	0	-	-	*default*
12	1	10	5	0	0	-	-	*default*

Os componentes da tabela anterior têm os seguintes significados:

- ID: Identificador do teste;
- Sensor Application Simulator (SAS): Número SAS utilizados durante os testes;
- *Miners*: Número de *miners* que compõem a rede Blockchain;
- *Nodes* (*full nodes*): Número de *full* nodes que compõe a rede Blockchain – excluindo os *miners*;
- Difficulty: Valor da dificuldade utilizada na rede Blockchain - corresponde ao nível de dificuldade aplicado durante a descoberta do *nonce* de um bloco;

[17] Por omissão este valor é igual ao número de cores do processador [88] – no caso deste ambiente de testes, 4 cores.

- Interval (ms): Intervalo em milissegundos entre cada pedido de transação;
- Duration (min): Período em minutos em que as SAS estiveram a enviar pedidos de transações (apenas aplicado aos testes que tinham um período de duração associado, os restantes tinham por base o número de transações realizadas. Isto deve-se ao facto de nuns casos o objetivo passar por avaliar o desempenho num número definido de transações, noutros casos pretende-se avaliar valores durante um período de tempo, como é o caso dos testes "Base", "Base_Difficulty_0" e o Teste 9);
- *Miner threads*: Número de *miner threads* utilizadas por cada *miner*.

Especificação dos testes

Os testes realizados no decorrer desta fase têm as seguintes características:

- Teste Base – Este teste tem como objetivo apenas recolher as métricas que estão a ser avaliadas nos testes numa situação onde não estão a ser enviadas transações – ou seja, uma situação onde não haja stress sobre um nó. A rede Blockchain neste teste está configurada com uma *difficulty* a 200000. Este teste serve como *baseline* para todos os cenários de teste que utilizem uma *difficulty* de 200000;
- Teste Base_Difficulty_0 – À semelhança do teste anteriormente descrito ("Teste Base"), o objetivo é o mesmo, sendo apenas alterada a *difficulty* da rede.
- Teste 1 – Este teste tem como objetivo analisar o desempenho da rede Blockchain constituída por um único nó (um *miner*);
- Teste 2 – Este teste tem como objetivo avaliar a influência que a adição de um *full node* (em relação ao Teste 1) tem na capacidade de processamento de transações da rede e no consumo de recursos dos elementos da mesma;
- Teste 3 – Este teste tem como objetivo avaliar a influência que a adição de um miner e um *full node* (em relação ao Teste 2) têm na capacidade de processamento de transações da rede e no consumo de recursos na mesma;
- Teste 4 – Este teste tem como objetivo avaliar a influência que a adição de um *miner* – em relação à rede do Teste 3 - tem na capacidade de processamento de transações da rede e no consumo de recursos;
- Teste 5 - Este teste consiste em estarem 2 SAS em simultâneo a enviar 10 transações cada uma. A rede Blockchain tem 1 nó (um *miner*) – tal como no Teste 1. Este teste visa comparar a performance do processamento de transações da rede em relação ao Teste 1 – onde eram enviadas 10 transações apenas de uma SAS;
- Teste 6 - Este teste visa comparar a performance do processamento de transações da rede em relação ao Teste 3 – onde eram enviadas 10 transações apenas de uma SAS;

- Teste 7 – Este teste possui exatamente o mesmo formato que o Teste 2, no entanto a *difficulty* da rede é alterada para 0. Através desta alteração é pretendido avaliar qual a influência da *difficulty* no tempo de processamento das transações, bem como no consumo de recursos dos nós da rede (essencialmente dos *miners*);
- Teste 8 - Este teste visa comparar a performance do processamento de transações da rede/performance dos nós em relação ao Teste 2, onde eram enviadas 10 transações apenas de uma SAS, existindo neste teste uma maior afluência de pedidos em simultâneo;
- Teste 9 – Este teste visa o envio de transações – vindas de 5 SAS a correr em simultâneo - a cada 500 ms durante 1 min. A rede Blockchain é igual à rede utilizada nos testes 8 e 2 – 2 nós (um *miner* e um *full node*). Neste teste o objetivo é a sobrecarga da rede Blockchain no processamento de transações;
- Teste 10 – O objetivo deste teste passa por avaliar a influência que tem a alteração das *miner threads* do Geth em comparação com o Teste 7 em que as redes Blockchain têm as mesmas características variando apenas o número de *miner threads*;
- Teste 11 – O objetivo deste teste passa por analisar o impacto de ter mais *miners* numa rede Blockchain, sendo toda a rede composta por *miners*;
- Teste 12 – Este teste inclui uma rede Blockchain com mais um *miner* em relação ao Teste 11 de forma a avaliar qual o impacto que este terá no desempenho entre estes dois testes.

Considerações gerais sobre os testes

Cada um dos cenários de teste foram repetidos três vezes por forma a garantir e verificar a existência de *outliers* – à exceção do Teste 9, Base e Base_Difficulty_0. O Teste 9 tinha como objetivo sobrecarregar o sistema e esse objetivo foi alcançado logo na primeira repetição. Os Testes Base e Base_Difficulty_9 tinham como objetivo apenas recolher as métricas que estão a ser avaliadas nos testes numa situação onde não estão a ser enviadas transações (durante 1 minuto) – ou seja, uma situação onde não haja stress sobre um nó.

Os testes consistiam no envio de uma transação, aguardar o sucesso, enviar outra transação – à exceção do Teste 9, Base e Base_Difficulty_0. Dadas as suas características e o seu objetivo, no caso dos testes Base e Base_Difficulty_0 não era enviado qualquer tipo de transação. O Teste 9 como o objetivo era sobrecarregar a Blockchain e eram enviados pedidos a cada 500ms, também não aguardava pela conclusão da transação anterior.

Foram definidos nos testes valores iniciais de *difficulty* diferentes - corresponde ao nível de dificuldade aplicado durante a descoberta do *nonce* de um bloco. Quanto maior a dificuldade, estatisticamente mais cálculos um *miner* deve realizar para descobrir a "solução" para um bloco válido. A *difficulty* é ainda utilizada de forma a controlar o tempo que demora um bloco a ser gerado – mantendo o processamento de um bloco dentro de

um intervalo alvo. A rede Blockchain utiliza o algoritmo de consenso Ethash (Proof-of-work – secção) e este algoritmo adapta rapidamente a *difficulty* à quantidade de recursos disponíveis na rede.

Ao longo da realização dos testes é necessário ter em conta as limitações fixadas pelo Geth, tais como [79]:

- AccountSlots - Número de *slots* de transação executáveis[18] garantidos por conta: 16;
- GlobalSlots - Número máximo de *slots* de transação executáveis para todas as contas: 4096;
- AccountQueue - Número máximo de *slots* de transação não executáveis[19] permitidos por conta: 64;
- GlobalQueue - Número máximo de *slots* de transação não executáveis para todas as contas: 1024.

Na secção seguinte são apresentadas as métricas avaliadas no decorrer dos testes realizados.

4.4.4.3 Métricas avaliadas

Por forma a avaliar o desempenho dos nós da rede Blockchain, em cada cenário de testes são avaliadas as seguintes métricas:

- CPU – Percentagem de CPU utilizada (medida a cada segundo);
- RAM – Memória em bytes que está a ser utilizada (medida a cada segundo);
- DISK – Memória utilizada de disco em bytes (medida a cada segundo);
- Tempo de resposta – Tempo (em milissegundos) desde que é enviado o pedido da Sensor Application até receber (também na Sensor Application) uma resposta em como o pedido foi concluído com sucesso.

Na próxima secção são apresentados os resultados experimentais dos cenários de teste descritos na secção anterior.

4.4.4.4 Análise dos resultados experimentais

Como podemos verificar através da especificação do ambiente de testes, este é um ambiente controlado e homogéneo – rede local, máquinas virtuais iguais e *hardware*

[18] As transações executáveis são aquelas em que a conta tem saldo suficiente para cobrir os custos e o *nonce* é o próximo da fila ou há uma sequência de transações válidas com *nonces* que levam ao atual.
[19] As transações não executáveis são aquelas em que não há saldo suficiente ou há uma lacuna única nas transações, tornando a atual inexequível, mesmo se o bloco estiver vazio.

equivalente -, logo, poderá levar a que o processamento das transações também seja semelhante em vários cenários diferentes.

Foi avaliada a situação dos *miners* das redes Blockchain num estado onde não estão a ser enviadas transações por forma a obter qual o comportamento dos *miners* neste estado (Teste Base e Teste Base_Difficulty_0). Desta forma, podemos verificar que mesmo nos testes de obtenção de uma *baseline* (onde não estão a ser recebidas na Blockchain quaisquer transações vindas das SAS), os *miners* estão a ter um grande esforço ao nível do CPU tanto com a *difficulty* a 0 (Figura 4-11) como a 200000 (Figura 4-12). No entanto, conseguimos verificar no teste com a diffuculty a 0 (Figura 4-11), há uma oscilação muito mais elevada do que no teste com difficulty a 200000 (Figura 4-12) – isto deve-se ao facto de quando a *difficulty* é menor, o *miner* consegue encontrar de uma forma muito mais rápida a resposta para o *puzzle* criptográfico (ver secção 2.2.2), pois não necessita de tantos cálculos para encontrar o *nonce*. No caso da *difficulty* mais elevada, o *miner* está em esforço contínuo, pois a tarefa de encontrar uma resposta para o *puzzle* criptográfica é computacionalmente muito mais exigente.

Figura 4-11 - CPU teste base – *difficulty* 0

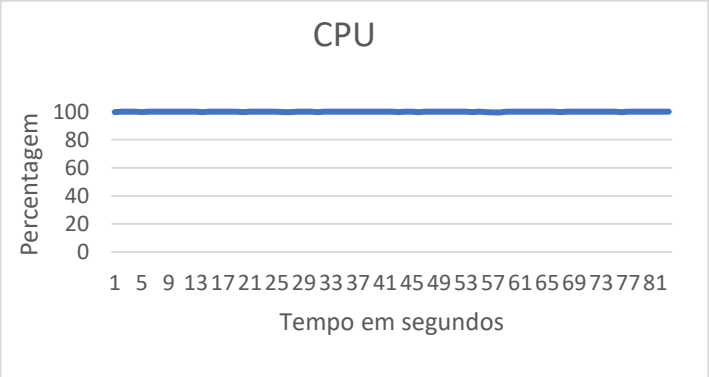

Figura 4-12 - CPU teste base – *difficulty* 200000

Uma das importantes métricas avaliadas é o tempo de processamento de uma transação, pois esta poderá ser uma condicionante para este caso de estudo, mas principalmente para a avaliação da viabilidade da utilização destas tecnologias para sistemas de tempo real. Desta forma, a Figura 4-13 apresenta a média do tempo de cada transação no total de todas as repetições de cada teste. Relativamente à métrica da memória RAM que está a ser utilizada, de uma forma geral, os valores não são muito elevados relativamente à capacidade que as máquinas possuem – não tendo influência na capacidade de processamento de transações da Blockchain.

O Teste 5 visa comparar a performance do processamento de transações da rede em relação ao Teste 1. Como podemos verificar no gráfico da Figura 4-13, o envio de transações por parte de mais uma SAS não tem impacto na performance de processamento das transações, aliás, este tempo ainda baixou ligeiramente.

O Teste 6 tem como objetivo comparar a performance no processamento de transações (enviando 10 transações por cada uma das 2 SAS) em comparação com o Teste 3 (10

transações vindas de 1 SAS). Neste caso, há um aumento significativo relativamente ao tempo de processamento da transação, passando para quase o dobro. Esta alteração no tempo de processamento das transações deve-se ao facto de também estarem a chegar mais transações em simultâneo à Blockchain, visto que no caso do Teste 3 apenas estava uma SAS a enviar dados, enquanto no Teste 6, estavam 2 SAS a enviar dados ao mesmo tempo.

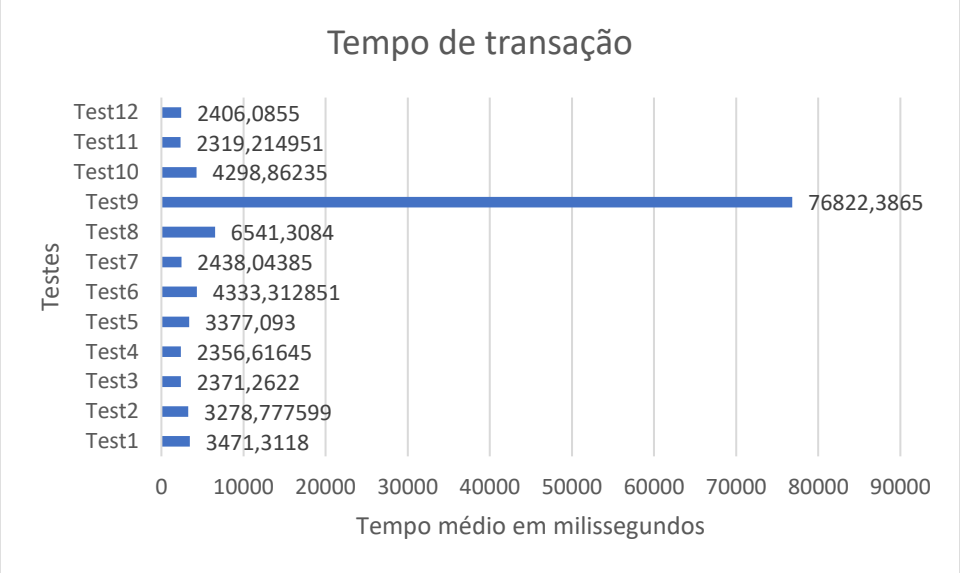

Figura 4-13 - Tempo médio de processamento das transações em cada teste no total de todas as repetições

O Teste 6 tem como objetivo comparar a performance no processamento de transações (enviando 10 transações por cada uma das 2 SAS) em comparação com o Teste 3 (10 transações vindas de 1 SAS). Neste caso, há um aumento significativo relativamente ao tempo de processamento da transação, passando para quase o dobro. Esta alteração no tempo de processamento das transações deve-se ao facto de também estarem a chegar mais transações em simultâneo à Blockchain, visto que no caso do Teste 3 apenas estava uma SAS a enviar dados, enquanto no Teste 6, estavam 2 SAS a enviar dados ao mesmo tempo.

No caso do Teste 8, o objetivo passa pela comparação do tempo de processamento de transações com o Teste 2 – sendo enviado, no Teste 8, 10 transações por cada uma das 5 SAS em teste. Aqui houve uma também uma grande variação no tempo de processamento das transações em relação ao Teste 2, elevando quase o dobro do tempo. A variação no tempo de processamento das transações deve-se ao facto de estar a chegar um maior número de transações em simultâneo à Blockchain. No caso do Teste 2 apenas estava a ser enviada uma transação, enquanto no Teste 8 estão a chegar à Blockchain 5 transações vindas de diferentes SAS.

Ao longo dos testes a capacidade de processamento dos *miners* está no seu limite. No caso do Teste 10 o objetivo passava por avaliar qual o impacto da redução da utilização de *miner threads* para metade – em comparação com o Teste 7. Desta forma, podemos verificar que,

no mesmo cenário, a redução de *miner threads* tem um grande impacto, pois elevou o tempo médio de processamento de transações para quase o dobro. No entanto, a utilização de menos *miner threads* – para o cenário de testes utilizado – não traz melhoria significativa nos resultados globais dos testes, pois reduz o esforço computacional dos *miners*, mas o tempo de resposta aumenta significativamente em comparação com outros testes (como podemos verificar no teste 10 - Figura 4-13). A Figura 4-14 apresenta o estado de utilização CPU do *miner* na repetição onde foi obtido o tempo de processamento de transação mais baixo (4293.24784 ms) – repetição nº 3.

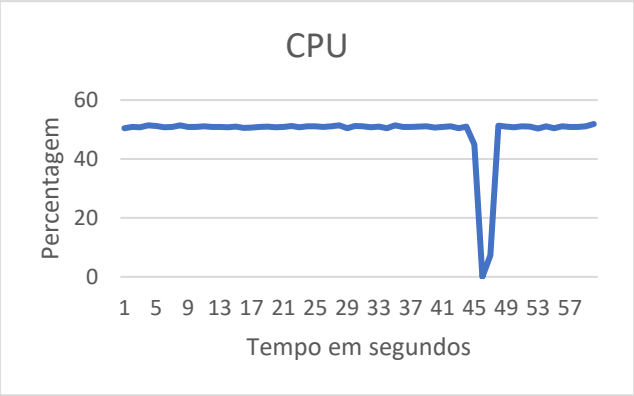

Figura 4-14 - Estado do CPU do *miner* no teste 10

No caso do Teste 12, o objetivo passa pela comparação da performance do processamento de transações em comparação com o Teste 11, pois ambos apenas possuem uma rede Blockchain compostas unicamente por *miners*, mas o Teste 12 tem mais um *miner* e pretende-se analisar o impacto que este tem. No entanto, como podemos verificar nos tempos de processamento, a adição de mais um *miner* não trouxe vantagens, pois a média de tempo de processamento de transações até foi superior com mais um *miner*. Este resultado demonstra que nem sempre ter mais *miners* em PoW (secção 2.2.2) significa ter melhores resultados. Ter mais *miners* pode ajudar a ter melhores resultados, pois existem muitas mais tentativas para descobrir o *puzzle* criptográfico, no entanto, num ambiente homogéneo como é este ambiente de testes, mais um *miner* não tem impacto no tempo de processamento das transações e neste caso até teve piores resultados.

Para o caso de estudo em análise, definir uma dificuldade e elevada não é um fator preponderante, visto que o objetivo da mesma passa por controlar a frequência e o tempo para um novo bloco ser gerado e, assim, controlar também as recompensas atribuídas aos *miners* pelo esforço computacional exigido (que também não é relevante para o caso de estudo). No entanto, este valor é ajustado automaticamente pelo algoritmo de consenso com base nos recursos disponíveis na rede, tal como referido anteriormente. Desta forma, a melhor opção para casos de estudo como este passa por ter uma dificuldade baixa – tal como está definido no teste 7 e no teste 11.

No teste 9, algumas transações falharam, visto que existia uma grande quantidade de solicitações para transações e o Geth tem algumas limitações impostas – nomeadamente AccountSlots, GlobalSlots, AccountQueue e GlobalQueue (ver a secção 4.4.4.2). Desta

forma, não podemos ter mais sensores a enviar dados no período de tempo de processamento de uma transação (leia-se em intervalos inferiores aqueles que a rede demora a fazer o processamento – estes períodos irão sempre depender das configurações e características da Blockchain) superior a esses limites estabelecidos.

No que diz respeito aos *full nodes*, a sobrecarga não é em nada comparável com a dos *miners*, devido às características e funções deste tipo de nós.

Os resultados detalhados dos testes encontram no “Anexo C – Testes de desempenho a uma Blockchain com múltiplos nós”.

4.5 Análise

Na primeira fase foi realizada a integração da rede IoT com a rede Blockchain, sendo implementados os princípios e os componentes básicos da *framework* BC4IoT. Também nesta fase foi criado um *smart contract* que possui uma estrutura de dados que permite armazenar alguns dados fornecidos pelo sensor da TTN – temperatura, humidade, luminosidade e bateria. A Sensor Application já estava desenvolvida por forma a fazer a ligação entre a TTN e a Blockchain (Ganache). Por último, também já estava implementada nesta fase uma primeira versão da Web Application onde era possível visualizar os dados dos sensores inseridos na Blockchain.

Na segunda fase foram implementadas umas validações básicas dos dados que são recolhidos pelos sensores da TTN. Para estas validações foi adaptado o *smart contract* desenvolvido na primeira, onde – para além da programação dessas validações - também foi criada uma nova estrutura de dados para guardar aqueles que são identificados como inválidos. Desta forma, é possível verificar quais os dados válidos e inválidos, bem como identificar qual o sensor que - por um ou outro motivo - está a enviar os dados inválidos. Ainda nesta fase, a Web Application também sofreu algumas atualizações nomeada a adaptação dos filtros à nova estrutura de dados e a inclusão de gráficos por forma a ser possível fazer uma análise mais detalhada acerca dos valores recolhidos pelos sensores.

Na terceira fase foram implementadas algumas validações de acordo com o algoritmo de validação de dados meteorológicos selecionado. Para isso, o *smart contract* teve de ser adaptado de forma a incluir estas validações. Ainda nesta fase foi utilizado e inserido na Blockchain um *dataset* real de dados meteorológicos de forma a testar o algoritmo de validação implementado. Para além disto, foram ainda realizados alguns testes com a introdução de “falhas” de forma verificar até que ponto o algoritmo implementado permite que sejam introduzidos dados inválidos. Ficou concluído que este algoritmo não permite que exista um “erro grosseiro”, pois as “falhas” que o mesmo deixava introduzir como válidas, eram alterações pequenas aos valores válidos

Por último, na quarta fase, foi criado um ambiente de testes com uma Blockchain real com múltiplos nós (Geth). Nesta fase concluiu-se que o ambiente de testes escolhido dá respostas para casos onde não seja necessário obter uma resposta imediata do sistema – essencialmente no armazenamento dos dados recolhidos pelos sensores. O caso de

estudo escolhido enquadra-se no tipo de aplicações onde a *framework* pode ser utilizada, pois aqui o importante é a informação ser guardada num local seguro e inviolável, no entanto, não é necessária ser dada uma resposta de imediato ao utilizador após a inserção – uma discrepância de alguns segundos ou até minutos não coloca em causa o bom funcionamento do sistema. Podemos também verificar através dos testes realizados que se a comunidade começar a crescer muito será necessário optar por uma melhor solução para os componentes da rede Blockchain, pois como é possível ver existem casos onde o consumo de recursos é muito elevado e que isto leva a que o tempo no processamento das transações também seja mais elevado e por sua vez que sejam atingidos os limites da Blockchain de uma forma mais rápida.

5 Conclusões

O trabalho descrito neste documento resultou na proposta de uma *framework* (BC4IoT) que integra redes IoT com redes Blockchain com o objetivo de resolver o problema de falta de confiança nos dados recolhidos por sensores.

A *framework* BC4IoT foi aplicada a um caso de estudo onde são recolhidos e validados dados ambientais em meio urbano. Aqui, foi possível utilizar todas as vertentes da *framework*. Para a rede de sensores é utilizada a LoRaWAN e para a Blockchain a plataforma Ethereum. Também é utilizado um *smart contract* onde são armazenados os dados ambientais e onde estão as validações dos mesmos. Foi ainda utilizado um outro serviço (aplicação web) que permite a ligação à Blockchain para obter os dados ambientais registados e mostrá-los de forma tratada ao utilizador.

A *framework* BC4IoT é versátil e pode ser adaptada a vários tipos de redes de sensores, bem como também a diferentes tipos de redes Blockchain (desde que permitam a execução de *smart contracts*). Esta *framework* permite ainda a adaptação a diversos casos de uso, pois a lógica está presente em *smart contracts* e, desta forma, pode variar de acordo com as necessidades de cada caso em concreto. É ainda possível criar diversos serviços que façam a ligação à Blockchain para obterem os dados armazenados – no caso apresentado é uma aplicação web, no entanto, poderá ser outro serviço que faça o mesmo tipo de comunicação com a Blockchain.

Relativamente aos testes realizados à *framework* aplicada ao caso de estudo escolhido, o ambiente de testes utilizado é apropriado a casos onde não seja necessário ter resultados em tempo real, pois existe sempre algum atraso no processamento e armazenamento das transações em relação ao momento de recolha dos dados pelos sensores. Para casos onde seja necessário ter os resultados disponíveis em tempo real é necessário optar por outras tecnologias (em especial no armazenamento da informação). Assim, é possível afirmar que, com aplicação ao caso de estudo analisado, a *framework* BC4IoT cumpriu com os principais objetivos deste trabalho: resolver alguns dos problemas de confiança nos dados recolhidos de uma rede IoT, armazenando a informação num local inviolável.

O presente trabalho apresentou ainda um estudo acerca da tecnologia Blockchain onde são abordados os principais fundamentos, as evoluções que a mesma foi sofrendo e a evolução que ainda irá ter a curto prazo. Foi também realizado um estudo acerca de IoT com principal foco na LoRaWAN/TTN. Para além disto, foram ainda apresentados exemplos onde esta tecnologia está a ser utilizada juntamente com as redes IoT.

Como trabalho futuro é importante permitir aos cidadãos de uma cidade que possam fazer o registo nesta comunidade (que contribuem através da recolha de dados ambientais e/ou disponibilização de recursos para a rede Blockchain) de forma simples onde o processo

passe por um registo *online*. Aqui, o cidadão indica qual a forma de como prefere contribuir para a comunidade. Independentemente de qual a forma como irá contribuir, deve ser possível ter acesso a todas as informações necessárias, nomeadamente:

- Inserção de dados do sensor com que irá contribuir para a comunidade: a introdução dos dados dos seus sensores para serem adicionados a uma Sensor Application de outro membro da comunidade que já esteja a ser executada;
- Acesso à aplicação (Sensor Application) responsável pela interligação dos sensores com a Blockchain: ter acesso à Sensor Application para que a possa executar de forma simples e intuitiva;
- Procedimentos para adicionar um nó à rede Blockchain: ter as informações necessárias para poder contribuir com um nó na rede Blockchain.

É ainda interessante analisar o comportamento da *framework* (essencialmente da componente Blockchain) para casos onde exista uma necessidade de haver uma maior complexidade nas “regras de negócio” definidas no *smart contract*, pois poderá haver impacto no desempenho dos nós da rede Blockchain, nomeadamente no tempo de processamento das transações.

Por último, o desenvolvimento deste trabalho no âmbito do MIS permitiu-me aprofundar bastante conhecimentos em áreas nas quais tinha algum interesse. Foi com muito agrado que consegui aprofundar as minhas capacidades de investigação e de aplicação dos resultados obtidos a um caso de estudo concreto. Foi ainda no âmbito do desenvolvimento deste projeto que tive oportunidade de participar no FIKALAB ISEC – uma experiência muito positiva que me permitiu sair da minha área de conforto em vários momentos e onde foi possível fazer uma partilha de ideias com especialistas de várias áreas de forma a evoluir no sentido correto. Em suma, foi uma experiência muito desafiante e exigente, no entanto, também foi muito enriquecedora em todas as vertentes.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A Literature Review,” *J. Comput. Commun.*, vol. 03, no. 05, pp. 164–173, 2015, doi: 10.4236/jcc.2015.35021.

[2] I. W. Stats, “No Title.” https://www.internetworldstats.com/stats.htm (accessed Jul. 03, 2020).

[3] M. Marjani *et al.*, “Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges,” *IEEE Access*, vol. 5, pp. 5247–5261, 2017, doi: 10.1109/ACCESS.2017.2689040.

[4] K. Korpela, J. Hallikas, and T. Dahlberg, “Digital Supply Chain Transformation toward Blockchain Integration,” *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, no. January, 2017, doi: 10.24251/hicss.2017.506.

[5] R. Casado-Vara, J. Prieto, F. De La Prieta, and J. M. Corchado, “How blockchain improves the supply chain: Case study alimentary supply chain,” *Procedia Comput. Sci.*, vol. 134, pp. 393–398, 2018, doi: 10.1016/j.procs.2018.07.193.

[6] “Blockchain and the Internet of Things: the IoT blockchain picture,” 2016. https://www.i-scoop.eu/internet-of-things-guide/blockchain-iot/ (accessed Jan. 13, 2019).

[7] K. Goyal, A. Garg, A. Tastogi, and S. Singhal, “A Survey on Internet of Things,” *Int. J. Comput. Sci. Eng.*, vol. 6, no. 12, pp. 492–496, 2018, doi: 10.26438/ijcse/v6i12.492496.

[8] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014, doi: 10.1109/JIOT.2014.2306328.

[9] M. Wu, T. J. Lu, F. Y. Ling, J. Sun, and H. Y. Du, “Research on the architecture of Internet of Things,” *ICACTE 2010 - 2010 3rd Int. Conf. Adv. Comput. Theory Eng. Proc.*, vol. 5, no. August, 2010, doi: 10.1109/ICACTE.2010.5579493.

[10] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “Internet of things: Vision, applications and research challenges,” *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012, doi: 10.1016/j.adhoc.2012.02.016.

[11] R. Anggarwal and M. Lal Das, “Aggarwal_RFID Security inthe Context of ‘Internet of Things’_2012.pdf,” pp. 51–56, 2012.

[12] P. Scully, “The Top 10 IoT Segments in 2018 – based on 1,600 real IoT projects - IoT Analytics,” 2018. https://iot-analytics.com/top-10-iot-segments-2018-real-iot-projects/ (accessed Dec. 10, 2018).

[13] J. Bartje, “The top 10 IoT application areas – based on real IoT projects,” 2016. https://iot-analytics.com/top-10-iot-project-application-areas-q3-2016/ (accessed Oct. 13, 2019).

[14] Y. Chauhan, “Application Areas of IoT !. The Internet of Things (IoT) is defined… | by

Yogesh Chauhan | Medium," Feb. 23, 2018. https://yogeshchauhan09.medium.com/application-areas-of-iot-24c784223b00 (accessed Jan. 13, 2021).

[15] "Choosing between NB-IoT and LoRaWAN - What factors to consider | LinkedIn," 2017. https://www.linkedin.com/pulse/choosing-between-nb-iot-lorawan-what-factors-ortiz-pasamontes/ (accessed Feb. 16, 2020).

[16] "What is LoRa? | Semtech LoRa Technology | Semtech." https://www.semtech.com/lora/what-is-lora (accessed Jan. 13, 2020).

[17] "A technical overview of LoRa ® and LoRaWAN ™ What is it?," 2015.

[18] "Semtech," 2020. https://www.semtech.com/ (accessed Dec. 14, 2020).

[19] "LoRa Alliance®," 2020. https://lora-alliance.org/ (accessed Dec. 14, 2020).

[20] B. Ray, "NB-IoT vs. LoRa vs. Sigfox," Jun. 25, 2018. https://www.link-labs.com/blog/nb-iot-vs-lora-vs-sigfox (accessed May 10, 2019).

[21] F. Fernandes, "Expresso | NB-IoT, a tecnologia inovadora que será alicerce do futuro," 2018. https://expresso.pt/iniciativaseprodutos/pub/2018-10-24-NB-IoT-a-tecnologia-inovadora-que-sera-alicerce-do-futuro-1#gs.963pfm (accessed Apr. 04, 2020).

[22] "Brasil terá 20 milhões de dispositivos conectados à rede Sigfox em 2019, projeta WND | TELETIME News," Feb. 06, 2018. https://teletime.com.br/06/02/2018/brasil-tera-20-milhoes-de-dispositivos-conectados-rede-sigfox-em-2019-projeta-wnd/ (accessed Feb. 16, 2020).

[23] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019, doi: 10.1016/j.icte.2017.12.005.

[24] "Community - The Things Network." https://www.thethingsnetwork.org/community (accessed Nov. 05, 2019).

[25] "Network Architecture | The Things Network." https://www.thethingsnetwork.org/docs/network/architecture.html (accessed Jan. 23, 2021).

[26] "Discovery Server | The Things Network." https://www.thethingsnetwork.org/docs/network/discovery/index.html (accessed Feb. 20, 2021).

[27] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in IoT: The challenges, and a way forward," *J. Netw. Comput. Appl.*, vol. 125, no. March 2018, pp. 251–279, 2019, doi: 10.1016/j.jnca.2018.10.019.

[28] M. Ahlmeyer, "Issues in Information Systems SECURING THE INTERNET OF THINGS: A REVIEW," vol. 17, no. Iv, pp. 21–28, 2016, [Online]. Available: http://www.iacis.org/iis/2016/4_iis_2016_21-28.pdf.

[29] J. Groopman and J. Owyang, "The Internet of Trusted Things Table of Contents," 2018.

[30] N. Petracek, "Is Blockchain The Way To Save IoT?," 2018. https://www.forbes.com/sites/forbestech-council/2018/07/18/is-blockchain-the-way-to-save-iot/#9ba71df5a741 (accessed Dec. 05, 2018).

[31] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *Cryptogr. Mail. List*

https//metzdowd.com, 2008.

[32] P. Martins, *Introdução à Blockchain*, 03–2018th ed. 2018.

[33] “O que é blockchain: indo além do bitcoin,” 2017. https://www.ibm.com/blogs/robertoa/2017/11/o-que-e-blockchain-indo-alem-do-bitcoin/ (accessed Jan. 19, 2019).

[34] “A Technical Explanation of Bitcoin for Everyone | Billy Bitcoin.” http://billybitco.in/ (accessed Jan. 26, 2021).

[35] “What Are Nodes? | Binance Academy.” https://academy.binance.com/en/articles/what-are-nodes (accessed Oct. 14, 2020).

[36] M. Musumeci, “Bitcoin Full nodes vs SPV nodes | Bitcoin privacy and security - Massimo Musumeci,” Nov. 24, 2019. https://www.massmux.com/bitcoin-full-nodes-vs-spv-nodes/ (accessed Jan. 25, 2021).

[37] R. Alves, “Guia completo do bitcoin (Parte 3),” Aug. 27, 2018. https://www.blockmaster.com.br/artigos/guia-completo-do-bitcoin-parte-3/ (accessed Jan. 16, 2021).

[38] J. Frankenfield, “Proof of Stake (PoS) Definition,” Aug. 11, 2019. https://www.investopedia.com/terms/p/proof-stake-pos.asp (accessed Jan. 27, 2021).

[39] A. Rosic, “Proof of Work vs Proof of Stake: Basic Mining Guide - Blockgeeks.” https://blockgeeks.com/guides/proof-of-work-vs-proof-of-stake/ (accessed Nov. 16, 2019).

[40] A. Gouveia, “Criptomoedas: proof-of-work ou proof-of-stake, quais as diferenças? por PROTESTE INVESTE,” 2021. https://www.deco.proteste.pt/investe/investimentos/mercados-moedas/dossie/bitcoin/criptomoedas-proof-of-work-proof-of-stake (accessed Feb. 20, 2021).

[41] S. Ray, “What is Proof of Stake? | Hacker Noon.” https://hackernoon.com/what-is-proof-of-stake-8e0433018256 (accessed Feb. 20, 2021).

[42] “How does blockchain work? | Lisk.” https://lisk.io/academy/blockchain-basics/how-does-blockchain-work/proof-of-stake (accessed Jan. 16, 2020).

[43] “What Are Proof of Stake and Delegated Proof of Stake?,” Jan. 15, 2021. https://www.gemini.com/cryptopedia/proof-of-stake-delegated-pos-dpos (accessed Jan. 28, 2021).

[44] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016, doi: 10.1109/ACCESS.2016.2566339.

[45] “O que é Delegated Proof of Stake? | Binance Academy.” https://academy.binance.com/pt/articles/delegated-proof-of-stake-explained (accessed Jan. 28, 2021).

[46] G. Konstantopoulos, “Noções básicas sobre os fundamentos do Blockchain, Parte 3: Prova de participação delegada | por Georgios Konstantopoulos | Rede Loom | Médio,” Jun. 11, 2018. https://medium.com/loom-network/understanding-blockchain-fundamentals-part-3-delegated-proof-of-stake-b385a6b92ef (accessed Jan. 28, 2021).

[47] “Delegated Proof of Stake (DPoS) - consensus.” https://tokens-

economy.gitbook.io/consensus/chain-based-proof-of-stake/delegated-proof-of-stake-dpos (accessed Feb. 20, 2021).

[48] “Delegated Proof Of Stake (DPoS) - GeeksforGeeks,” Aug. 19, 2020. https://www.geeksforgeeks.org/delegated-proof-of-stake/ (accessed Jan. 28, 2021).

[49] N. Szabo, “Smart Contracts : Building Blocks for Digital Transformation,” vol. 1, no. c, pp. 1–14, 1996.

[50] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, “Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts,” in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 839–858, doi: 10.1109/SP.2016.55.

[51] R. Beck, J. S. Czepluch, N. Lollike, and S. Malone, “BLOCKCHAIN – THE GATEWAY TO TRUST- FREE CRYPTOGRAPHIC TRANSACTIONS,” *Twenty-Fourth Eur. Conf. Inf. Syst. (ECIS), İstanbul,Turkey*, vol. 6, no. May, pp. 4013–4027, 2016, [Online]. Available: http://aisel.aisnet.org/ecis2016_rp%0Ahttp://aisel.aisnet.org/ecis2016_rp.

[52] F. L. Ferreira, “Blockchain e Ethereum Aplicações e Vulnerabilidades,” p. 36, 2017, [Online]. Available: https://linux.ime.usp.br/~fredlage/mac0499/Monografia.pdf.

[53] “Intro to Ethereum | ethereum.org.” https://ethereum.org/en/developers/docs/intro-to-ethereum/ (accessed Feb. 20, 2021).

[54] A. Lewis, “A gentle introduction to Ethereum – Bits on Blocks,” Oct. 02, 2016. https://bitsonblocks.net/2016/10/02/gentle-introduction-ethereum/ (accessed May 12, 2019).

[55] G. Gürsoy, C. M. Brannon, and M. Gerstein, “Using Ethereum blockchain to store and query pharmacogenomics data via smart contracts,” *bioRxiv*, pp. 1–12, 2020, doi: 10.1101/2019.12.16.878488.

[56] T. McCallum, “Ethereum database research and testing,” Jan. 28, 2018. https://github.com/tpmccallum/ethereum_database_research_and_testing/blob/master/leveldb/leveldb.md (accessed Jan. 04, 2021).

[57] “Applications - Blockchain Technologies.” https://www.blockchaintechnologies.com/applications/ (accessed Jan. 18, 2019).

[58] “Blockchain and the Internet of Things: the IoT blockchain picture,” Feb. 2018. https://www.i-scoop.eu/internet-of-things-guide/blockchain-iot/ (accessed May 21, 2019).

[59] Forbes, “Is Blockchain The Way To Save IoT?,” 2018. https://www.forbes.com/sites/forbestechcouncil/2018/07/18/is-blockchain-the-way-to-saveiot/#9ba71df5a741 (accessed Dec. 15, 2018).

[60] D. Miller, “Blockchain and the internet of things in the industrial sector,” *IT Prof.*, vol. 20, no. 3, pp. 15–18, 2018, doi: 10.1109/MITP.2018.032501742.

[61] “Logistics 4.0 and smart supply chain management in Industry 4.0.” https://www.i-scoop.eu/industry-4-0/supply-chain-management-scm-logistics/ (accessed Jan. 26, 2019).

[62] “Blockchain smart port case: container release in the port of Antwerp.” https://www.i-scoop.eu/blockchain-smart-port-project-case-container-release-port-antwerp/ (accessed Jan. 26, 2019).

[63] P. S. Faye, “Use of Blockchain Technology in Agribusiness: Transparency and Monitoring in Agricultural Trade,” vol. 31, no. Msmi, pp. 38–40, 2017, doi: 10.2991/msmi-17.2017.9.

[64] S. Chai, M. Kim, and H. R. Rao, “Firms’ information security investment decisions: Stock market evidence of investors’ behavior,” *Decis. Support Syst.*, vol. 50, no. 4, pp. 651–661, Mar. 2011, doi: 10.1016/j.dss.2010.08.017.

[65] A. Takyar, “Blockchain in Agriculture - Improving Agricultural Techniques.” https://www.leewayhertz.com/blockchain-in-agriculture/ (accessed May 16, 2019).

[66] P.-F. Tétreault, “How the agricultural industry can harvest the benefits of blockchain technology | Canada | Global law firm | Norton Rose Fulbright,” 2008. https://www.nortonrosefulbright.com/en-ca/knowledge/publications/99438e54/how-the-agricultural-industry-can-harvest-the-benefits-of-blockchain-technology (accessed Jun. 07, 2019).

[67] J. Lin, Z. Shen, C. Miao, and S. Liu, “Using blockchain to build trusted LoRaWAN sharing server,” *Int. J. Crowd Sci.*, vol. 1, no. 3, pp. 270–280, 2017, doi: 10.1108/ijcs-08-2017-0010.

[68] A. Gemalto and S. And, “LoRaWAN ™ SECURITY A WHITE PAPER PREPARED FOR THE LoRa ALLIANCE™ FULL END-TO-END ENCRYPTION FOR IoT APPLICATION PROVIDERS,” 2017.

[69] “Truffle | Overview | Documentation | Truffle Suite.” https://www.trufflesuite.com/docs/truffle/overview (accessed Jun. 15, 2020).

[70] “Ganache | Overview | Documentation | Truffle Suite.” https://www.trufflesuite.com/docs/ganache/overview (accessed Jul. 27, 2019).

[71] P. Ramanujam, “Ethereum and Blockchain,” Mar. 10, 2017. https://iotbl.blogspot.com/2017/03/ethereum-and-blockchain-2.html (accessed Jul. 20, 2019).

[72] “Node.js - O que é, como funciona e quais as vantagens,” Sep. 05, 2018. https://www.opus-software.com.br/node-js/ (accessed Nov. 16, 2020).

[73] G. McCubbin, “Intro to Web3.js · Ethereum Blockchain Developer Crash Course | Dapp University,” Nov. 25, 2020. https://www.dappuniversity.com/articles/web3-js-intro (accessed Dec. 15, 2020).

[74] “What is Geth? - ETH Gas Station,” Aug. 22, 2019. https://ethgasstation.info/blog/what-is-geth/ (accessed Nov. 22, 2020).

[75] “On the origins of environmental bullshit,” Jul. 25, 2017. https://theconversation.com/on-the-origins-of-environmental-bullshit-80955 (accessed Feb. 04, 2021).

[76] J. Estévez, P. Gavilán, and J. V. Giráldez, “Guidelines on validation procedures for meteorological data from automatic weather stations,” *J. Hydrol.*, vol. 402, no. 1–2, pp. 144–154, May 2011, doi: 10.1016/j.jhydrol.2011.02.031.

[77] M. Gentilucci, M. Barbieri, P. Burt, and F. D’Aprile, “Preliminary data validation and reconstruction of temperature and precipitation in central Italy,” *Geosci.*, vol. 8, no. 6, pp. 1–16, 2018, doi: 10.3390/geosciences8060202.

[78] I. Zahumenský, “Guidelines on Quality Control Procedures for Data from Automatic Weather Stations The document contains a proposal for the Guidelines on Quality Control Procedures for data from Automatic Weather Stations,” *CBS/OPAG-IOS/ET*

AWS-3/Doc. 4(1), no. 955, p. 9, 2004, [Online]. Available: https://www.wmo.int/pages/prog/www/OSY/Meetings/ET-AWS3/Doc4(1).pdf.

[79] “Github - Go-ethereum.” https://github.com/ethereum/go-ethereum/blob/master/core/tx_pool.go (accessed Jan. 18, 2021).

[80] “About LoRaWAN® | LoRa Alliance®.” https://lora-alliance.org/about-lorawan (accessed Jan. 14, 2021).

[81] “O que é o malware? | Malwarebytes.” https://pt.malwarebytes.com/malware/ (accessed Feb. 20, 2021).

[82] “Vírus de computador x worms de rede | Kaspersky.” https://www.kaspersky.com.br/resource-center/threats/computer-viruses-vs-worms (accessed Feb. 20, 2021).

[83] “What is a denial of service attack (DoS) ? - Palo Alto Networks.” https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos (accessed Feb. 20, 2021).

[84] “Brute Force Attacks: Password Protection | Kaspersky.” https://www.kaspersky.com/resource-center/definitions/brute-force-attack (accessed Feb. 20, 2021).

[85] “O que é um ataque Man-in-the-Middle? | AVG.” https://www.avg.com/pt/signal/man-in-the-middle-attack (accessed Feb. 20, 2021).

[86] J. FRANKENFIELD, “Nonce Definition,” Jul. 27, 2020. https://www.investopedia.com/terms/n/nonce.asp (accessed Jan. 25, 2021).

[87] G. Karame, E. Androulaki, and S. Capkun, *Double-spending fast payments in Bitcoin*. 2012.

[88] “Device Registration | The Things Network.” https://www.thethingsnetwork.org/docs/devices/registration.html (accessed Feb. 21, 2021).

[89] “Addressing & Activation | The Things Network.” https://www.thethingsnetwork.org/docs/lorawan/addressing.html (accessed Feb. 21, 2021).

[90] Statista Research Department, “Number of IoT devices 2015-2025 | Statista,” Nov. 27, 2016. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (accessed Oct. 14, 2019).

[91] “Exame Informática | Blockchain: o que é e como funciona.” https://visao.sapo.pt/exameinformatica/tutoriais/2020-12-10-blockchain-o-que-e-e-como-funciona/ (accessed Feb. 20, 2021).

[92] R. QUINNELL, “Basics of blockchain for the IoT - EDN,” Jul. 15, 2019. https://www.edn.com/basics-of-blockchain-for-the-iot/ (accessed Jan. 29, 2021).

[93] J. Vicent, “Bitcoin consumes more energy than Switzerland, according to new estimate - The Verge,” Jul. 04, 2019. https://www.theverge.com/2019/7/4/20682109/bitcoin-energy-consumption-annual-calculation-cambridge-index-cbeci-country-comparison (accessed Jan. 26, 2021).

[94] “Consensus Algorithms: Proof-of-Stake & Cryptoeconomics — Nichanan Kesonpat.” https://www.nichanank.com/blog/2018/6/4/consensus-algorithms-pos-dpos (accessed Feb. 20, 2021).

[95] T. Feed, “The Blockchain Ecosystem. The blockchain is not only a protocol… | by Telos Feed | Medium,” Jul. 12, 2019. https://medium.com/@Telosfeed_en/the-blockchain-ecosystem-273544192b7c (accessed Jan. 28, 2021).

[96] “What are Smart Contracts and how do they work? — Bitpanda Academy.” https://www.bitpanda.com/academy/en/lessons/what-are-smart-contracts-and-how-do-they-work/ (accessed Feb. 20, 2021).

[97] F. Jamil, L. Hang, K. Kim, and D. Kim, “A Novel Medical Blockchain Model for Drug SupplyChain Integrity Management in a Smart Hospital,” doi: 10.3390/electronics8050505.

ANEXOS

Anexo A - Sensor Application

A Sensor Application (Figura A-1) trata-se de uma aplicação em JavaScript (node.js) e que tem como principal objetivo recolher e tratar os dados de um sensor da The Things Network e armazená-los numa Blockchain.

O projeto relativo a esta aplicação encontra-se no github através do seguinte endereço: https://github.com/marcoaureliosequeira/ttn-to-blockchain/tree/master/ttnNodeApplication.

Figura A-1 - Sensor Application

Dependências e configurações

Caso a máquina não possua todas as dependências, basta estar na pasta do projeto e correr “npm install” e as mesmas serão instaladas automaticamente de acordo com o que está especificado no ficheiro “package.json”.

Relativamente às configurações do sensor TTN (The Things Network) e da Blockchain, existe um ficheiro “configs.js” onde é possível indicar qual o endereço do *host* Blockchain, bem como a appId e accessKey do sensor TTN. Este ficheiro terá sempre de ser alterado acaso se pretenda alterar o sensor pelo qual queremos ficar a aguardar informação, bem como a Blockchain onde pretendemos armazenar a mesma.

Para iniciar o programa é necessário compilar e executar o ficheiro JavaScript correspondente, para isso, é necessário aceder à raiz do projeto e executar “node

store_records.js". Após a compilação o programa ficará à escuta de novos registos para o sensor.

Estrutura do projeto

Neste capítulo é descrita a organização dos ficheiros do projeto, bem como a especificação dos ficheiros mais importantes do mesmo. O projeto está dividido da seguinte forma:

- **ttnNodeApplication**
  - **build**
    - contracts - pasta onde ficam armazenados os ABI (*Application Binary Interface*) dos *smart contracts*;
  - **contracts**
  - **migrations** - pasta onde ficam os ficheiros de configuração para a migração dos *smart contracts*;
  - **config.js** - ficheiro com as configurações do sensor TTN e da Blockchain
  - **store_records.js** - ficheiro principal da aplicação, é o código que será executado;
  - **truffle-config.js** - ficheiro de configuração do Truffle.

Sequência de ações

Esta aplicação possui três ações principais que se subdividem em bastantes outras. As ações são as seguintes:

1. Inicialização e configuração da web3.js- método **initWeb3** - Este método tem como principal objetivo criar uma instância web3 para a comunicação com o *host provider* da Blockchain;
2. Inicialização dos *smart contracts* - método **setContracts** - É neste método que é criado o objecto do smart contract a partir do ABI gerado após a migração do mesmo. A este objeto é também associado a um provider, neste caso o web3Provider que foi inicializado no passo anterior. Estas instruções são executadas através da framework Truffle (uma das dependências necessárias para a execução do projeto e que está presente no ficheiro "package.json");
3. Cliente TTN - método **ttnClient** - Este método tem como objetivo inicializar o cliente TTN através da *appId* e *accessKey* definidas e a cada uplink existente, tratar os dados e invocar o método de armazenamento de dados na Blockchain - **setInformationInBlockchain**.

setInformationInBlockchain - Neste método é criada - através da framework Truffle - uma instância do *smart contract*. Através desta instância criada invocamos o método para adicionar ao array associativo mais um elemento com os dados que lhe são passados.

Anexo B - Web Application

A Web Application (Figura A-2) trata-se de uma aplicação em Vue.js e que tem como principal objetivo permitir uma visualização tratada a informação presente no smart contract armazenado na Blockchain.

O projeto relativo a esta aplicação encontra-se no github através do seguinte endereço: https://github.com/marcoaureliosequeira/ttn-to-blockchain/tree/master/webApplication.

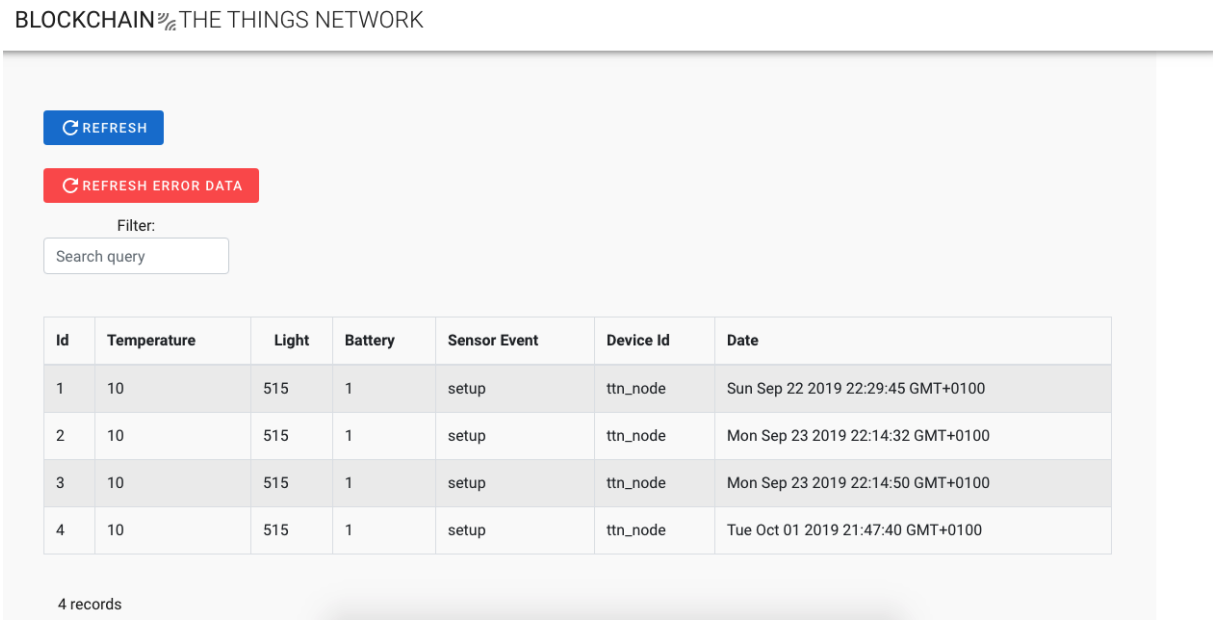

Figura A-2 - Web Application

Dependências e configurações

Caso a máquina não possua todas as dependências, basta estar na pasta do projeto e correr “npm install” e as mesmas serão instaladas automaticamente de acordo com o que está especificado no ficheiro “package.json”.

Relativamente às configurações do sensor TTN (The Things Network) e da Blockchain, existe um ficheiro “configs.js” onde é possível indicar qual o endereço do host Blockchain, bem como a appId e accessKey do sensor TTN. Este ficheiro terá sempre de ser alterado acaso se pretenda alterar o sensor pelo qual queremos ficar a aguardar informação, bem como a Blockchain onde pretendemos armazenar a mesma.

Para iniciar o programa é necessário compilar e executar o projeto, para isso, é necessário aceder à raiz do projeto e executar “npm run serve”. Após a compilação é possível aceder à aplicação no *browser* através de “http://localhost:8080”.

Estrutura do projeto

Neste capítulo é descrita a organização dos ficheiros do projeto, bem como a especificação dos ficheiros mais importantes do mesmo. O projeto está dividido da seguinte forma:

- **ttnWebApplication**
    - **public** - pasta que contem o ficheiro HTML da raiz da aplicação, onde é incluído o componente onde iremos mostrar a informação ao utilizador;
    - **src**
        - **assets**
        - **components** - Componentes que irão ser utilizados na aplicação, como é o caso do componente de “Loading” e da “DataTable”;
        - **store** - Armazena a *store* Vuex para a gestão do *state* da aplicação
            - **store.js** - Ficheiro responsável pela gestão do *state* da aplicação. É neste ficheiro que se fazer as chamadas à Blockchain para obter as informações do *smart contract*;
        - **App.vue** - Componente principal de Vue que irá conter os restantes componentes para a visualização de informação;
        - **main.js** - ficheiro principal de JS onde é inicializado o componente principal de Vue. Irá montar este componente no ficheiro presente na diretoria “public”;
    - **config.js** - ficheiro com as configurações do sensor TTN e da Blockchain.

Sequência de ações

Esta aplicação possui três ações principais que se subdividem em bastantes outras. Estas ações são chamadas nos seus componentes correspondentes e são executadas na *store* da aplicação. As ações são as seguintes:

- Inicialização e configuração da web3.js- método initWeb3 - Este método tem como principal objetivo criar uma instância web3 para a comunicação com o *host provider* da Blockchain;
- Inicialização dos *smart contracts* - método setContracts - É neste método que é criado o objeto do *smart contract* a partir do ABI gerado após a migração do mesmo. A este objeto é também associado a um *provider*, neste caso o web3Provider que foi inicializado no passo anterior. Estas instruções são executadas através da *framework* Truffle (uma das dependências necessárias para a execução do projeto e que está presente no ficheiro “package.json”);
- Obter informação presente no *smart contract* da Blockchain - método getInformationFromBlockchain - Neste método é criada - através da *framework*

Truffle - uma instância do *smart contract*. Através desta instância criada invocamos o getter do número de itens presentes no *array* associativo do *smart contract*. Após isso, é percorrido o *array* associativo e por cada uma das entradas, a informação é tratada de forma a atualizar a variável associada à DataTable onde a informação é visualizada.

Anexo C – Testes de desempenho a uma Blockchain com múltiplos nós

Neste anexo são apresentados os resultados dos testes de desempenho realizados a uma Blockchain com múltiplos nós, aplicado ao caso de estudo de recolha e validação de dados ambientais em ambiente urbano.

Teste Base

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 - Miner

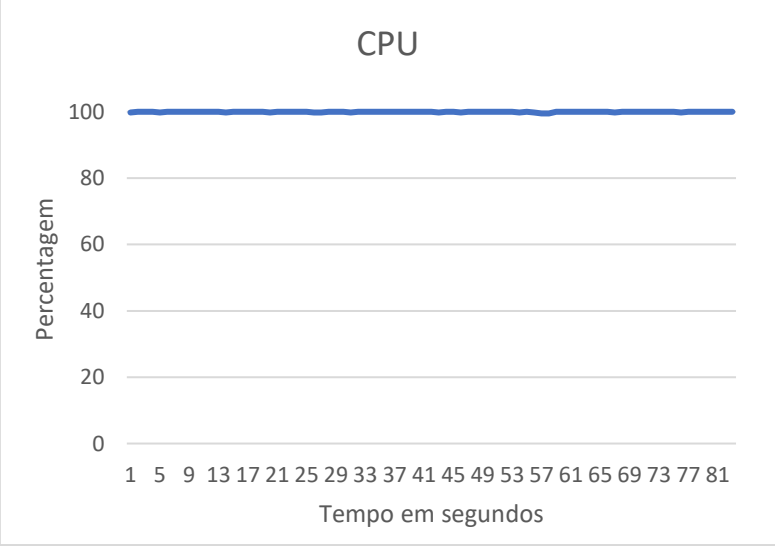

Figura A-3 - Valores de CPU do node 1 para a repetição 1 do teste Base

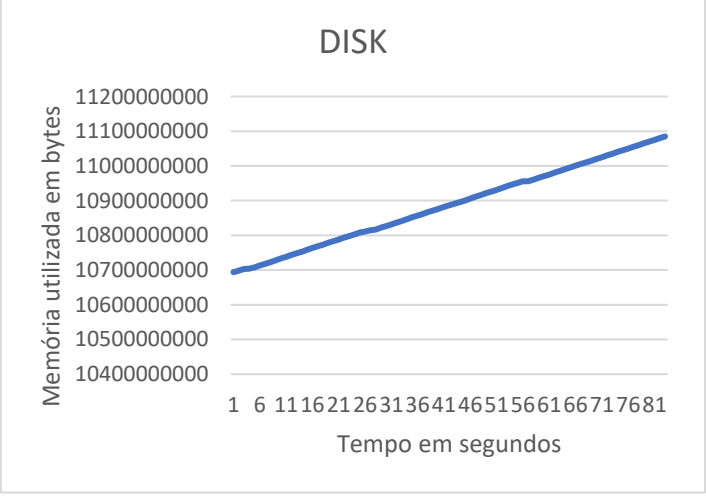

Figura A-4 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste Base

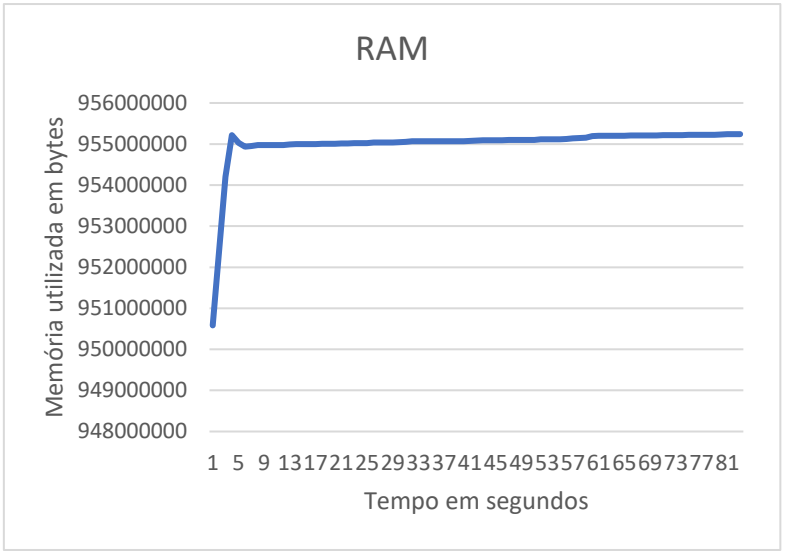

Figura A-5 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste Base

Base_Difficulty_0

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

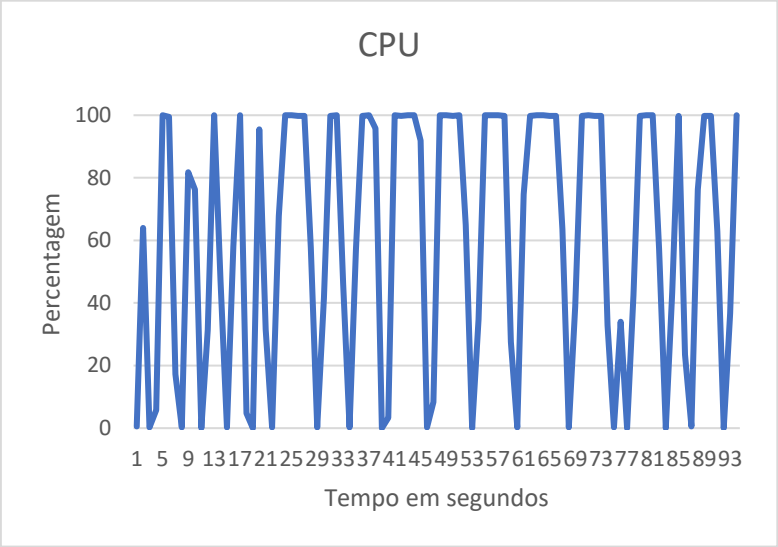

Figura A-6 - Valores de CPU do node 1 para a repetição 1 do teste Base_Difficulty_0

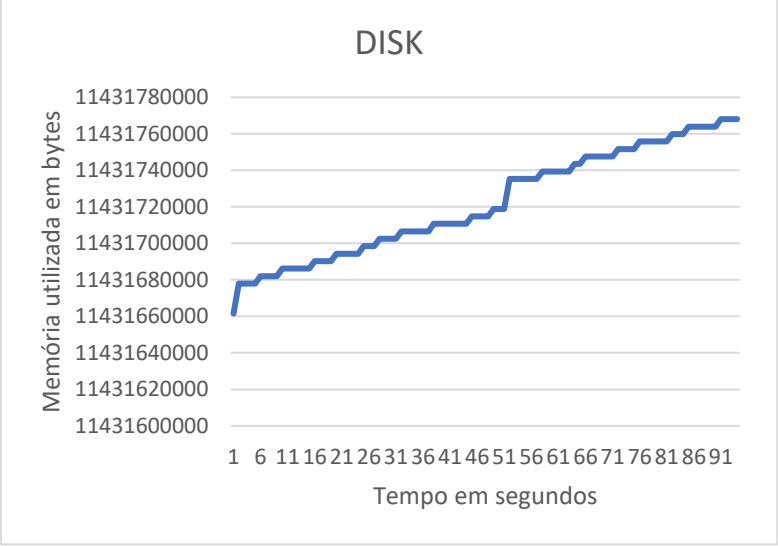

Figura A-7 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste Base_Difficulty_0

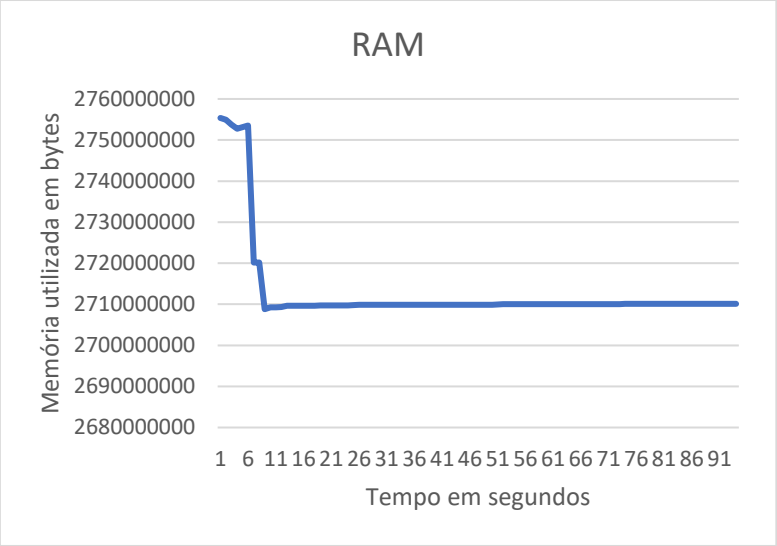

Figura A-8 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste Base_Difficulty_0

Teste 1

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

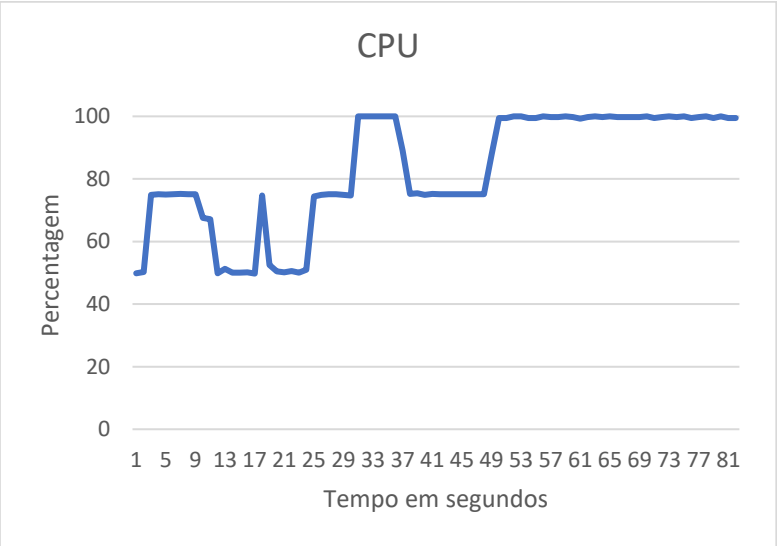

Figura A-9 - Valores de CPU do node 1 para a repetição 1 do teste 1

Figura A-10 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 1

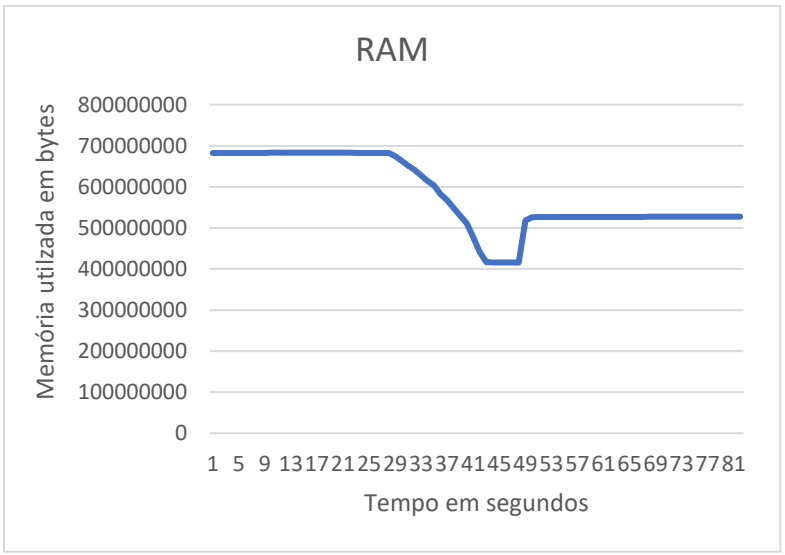

Figura A-11 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 1

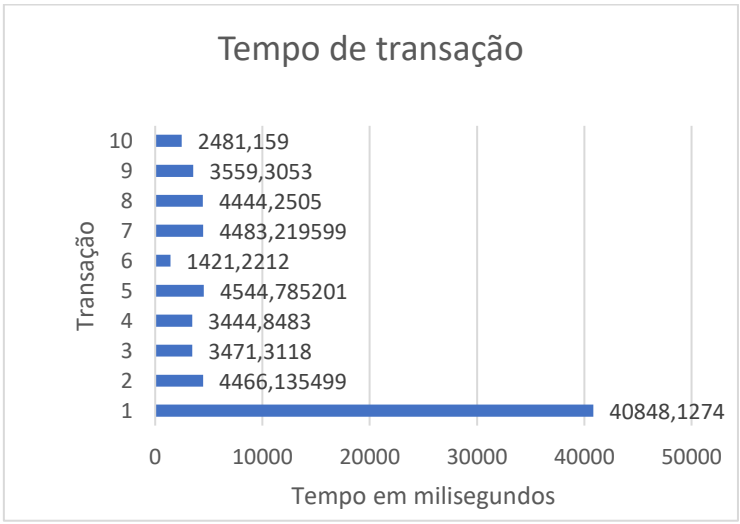

Figura A-12 - Tempo de processamento das transações realizadas na repetição 1 do teste 1

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

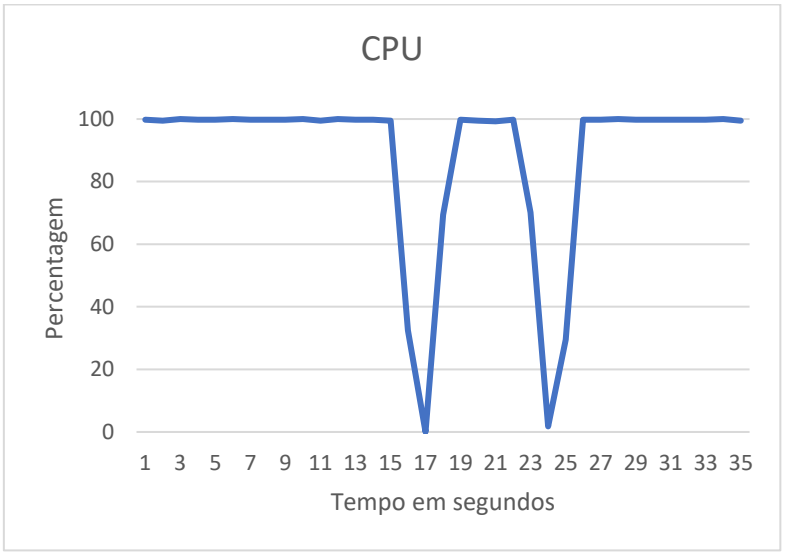

Figura A-13 - Valores de CPU do node 1 para a repetição 2 do teste 1

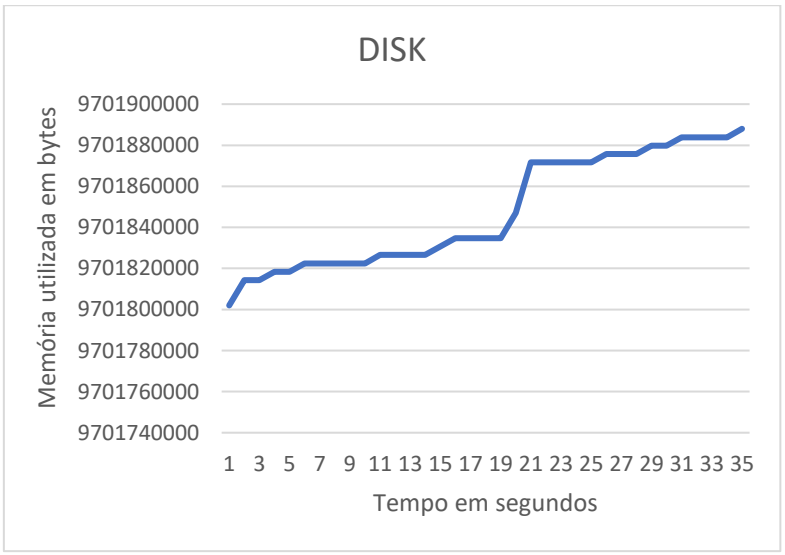

Figura A-14 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 1

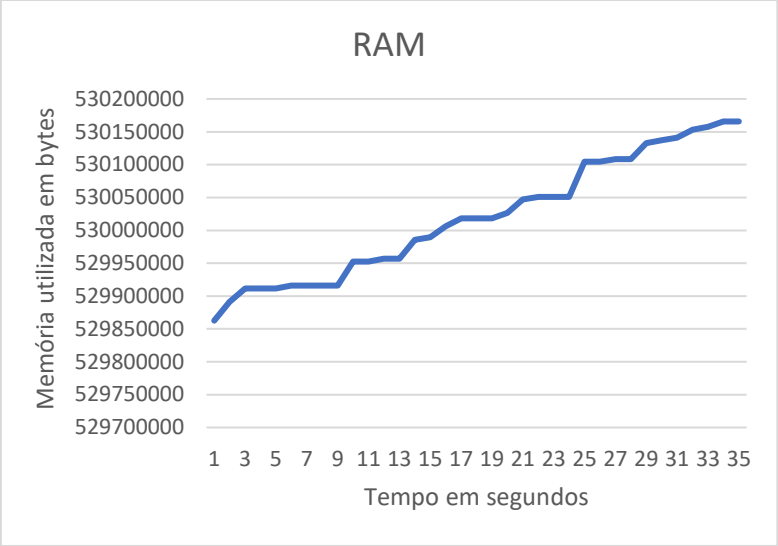

Figura A-15 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 1

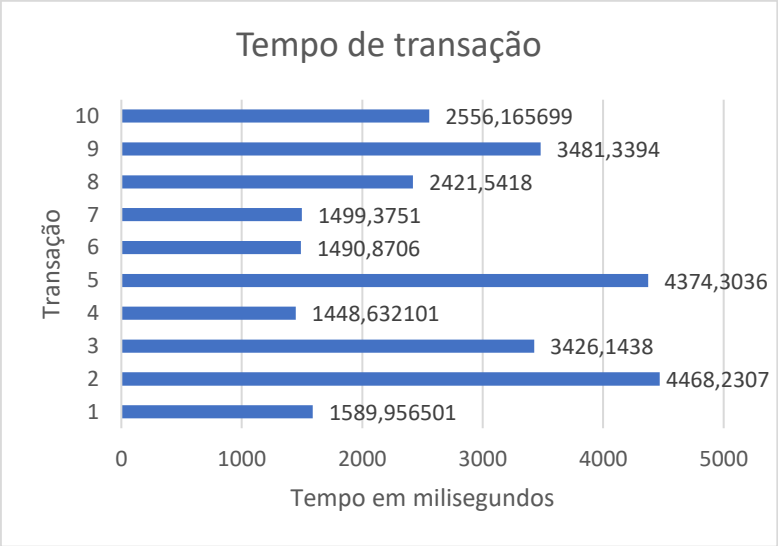

Figura A-16 - Tempo de processamento das transações realizadas na repetição 2 do teste 1

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

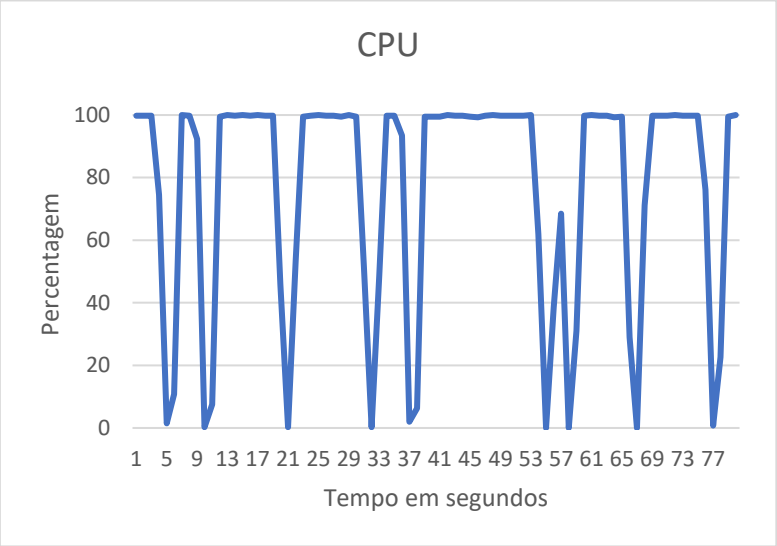

Figura A-17 - Valores de CPU do node 1 para a repetição 3 do teste 1

Figura A-18 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 1

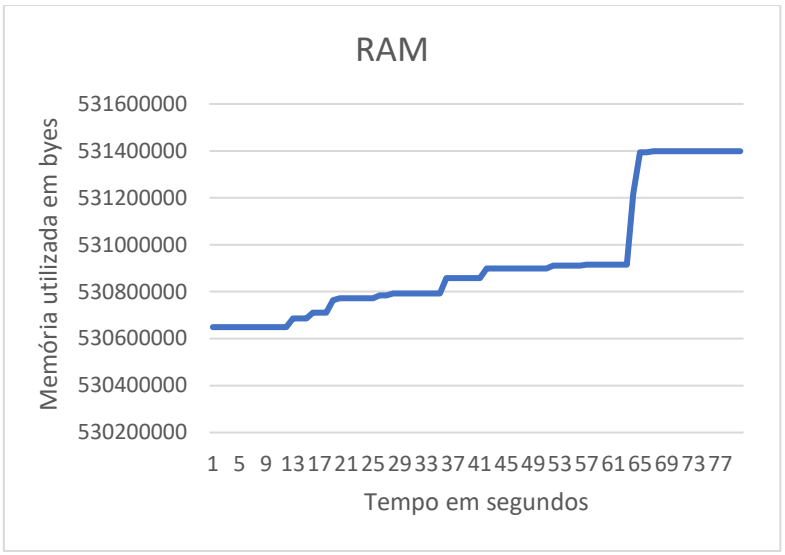

Figura A-19 - Memória em bytes que está a ser utilizada no node 1 para a repetição 3 do teste 1

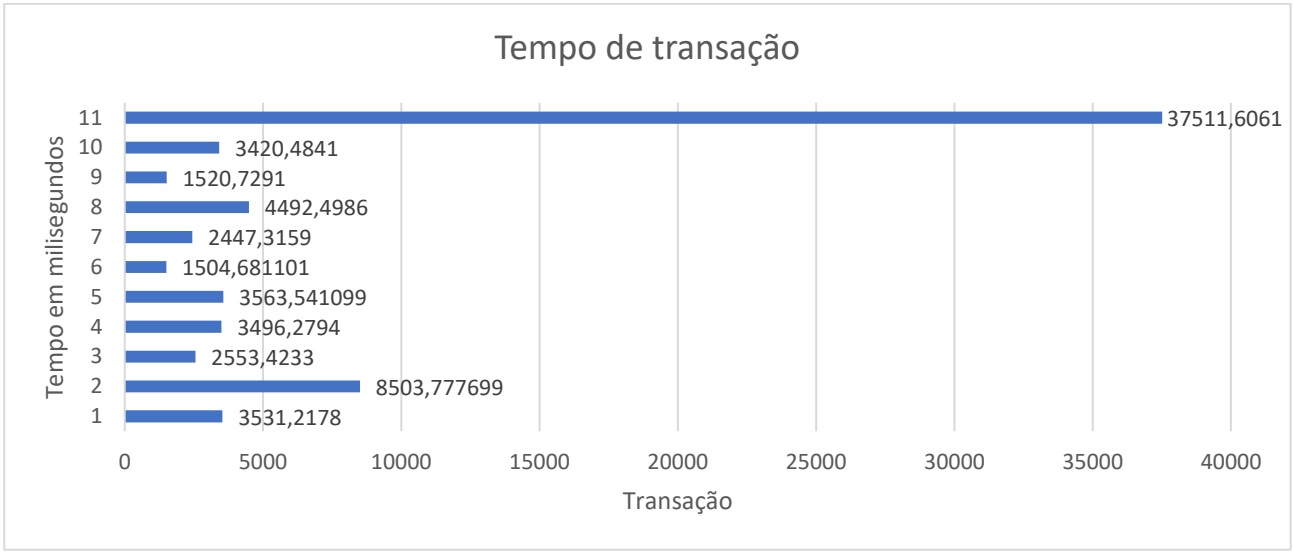

Figura A-20 - Tempo de processamento das transações realizadas na repetição 3 do teste 1

Teste 2

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

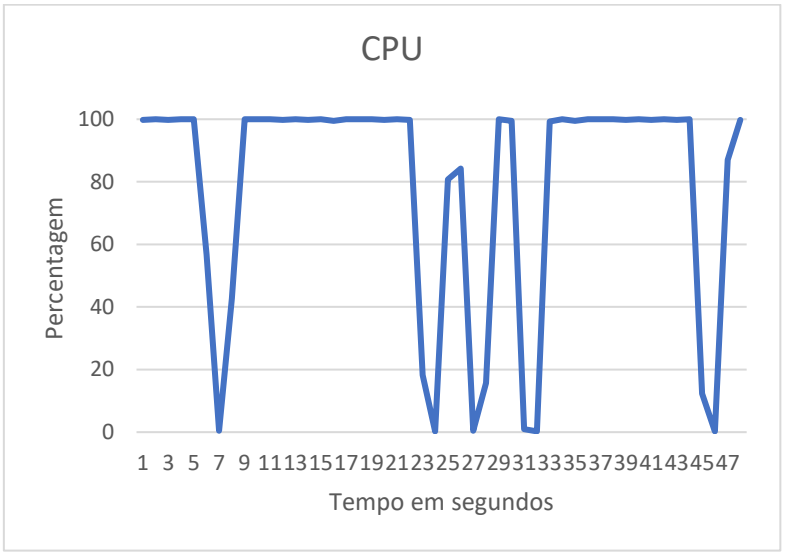

Figura A-21 - Valores de CPU do node 1 para a repetição 1 do teste 2

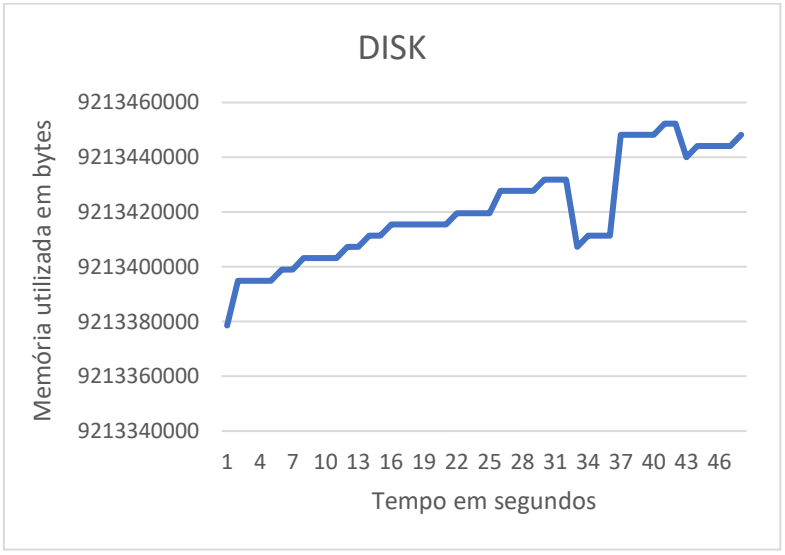

Figura A-22 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 2

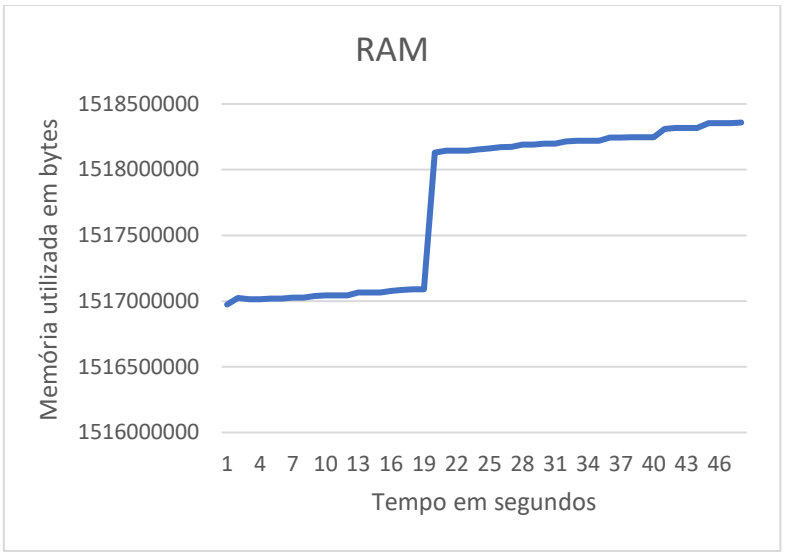

Figura A-23 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 2

Node 2

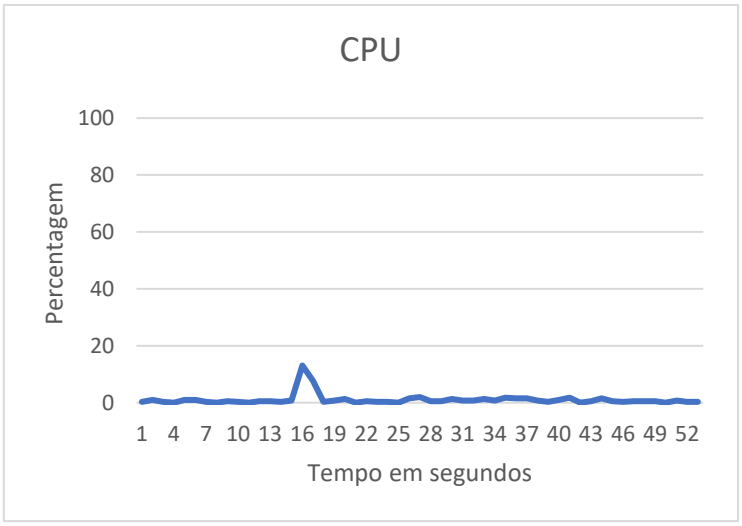

Figura A-24 - Valores de CPU do node 2 para a repetição 1 do teste 2

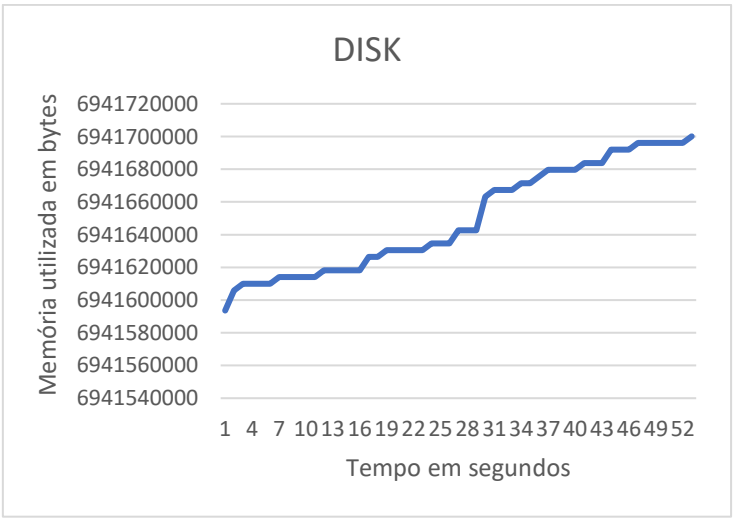

Figura A-25 - Memória utilizada de disco em bytes do node 2 para a repetição 1 do teste 2

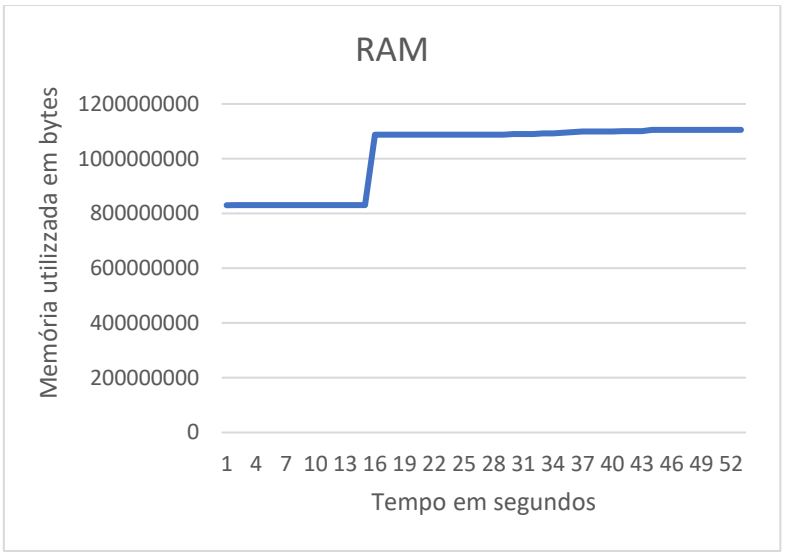

Figura A-26 - Memória em bytes que está a ser utilizada no node 2 para a repetição 1 do teste 2

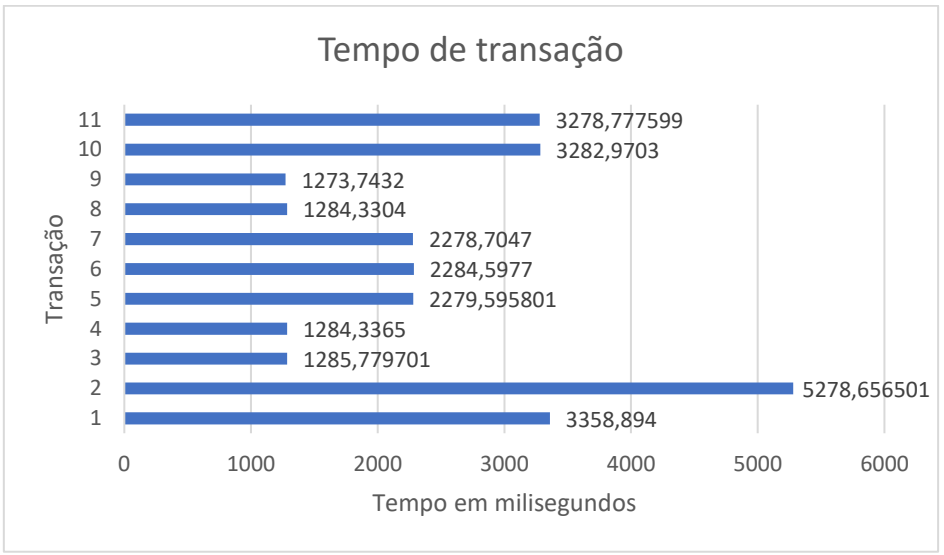

Figura A-27 - Tempo de processamento das transações realizadas na repetição 1 do teste 2

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

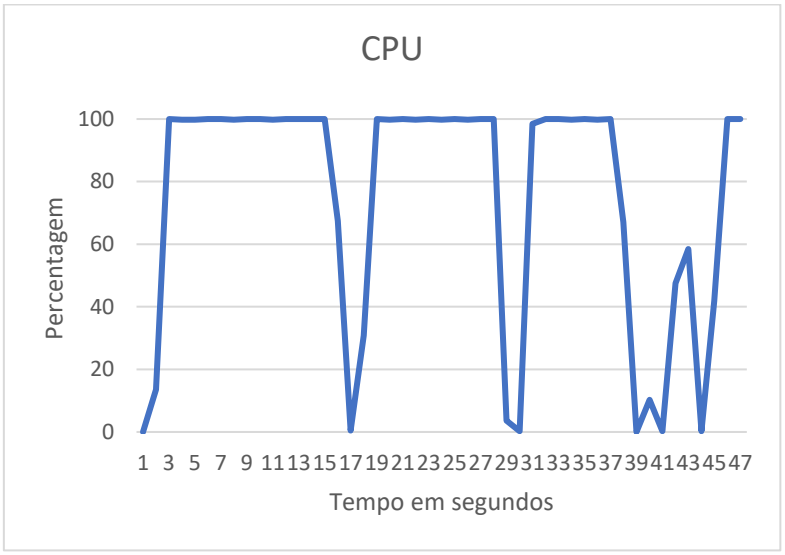

Figura A-28 - Valores de CPU do node 1 para a repetição 2 do teste 2

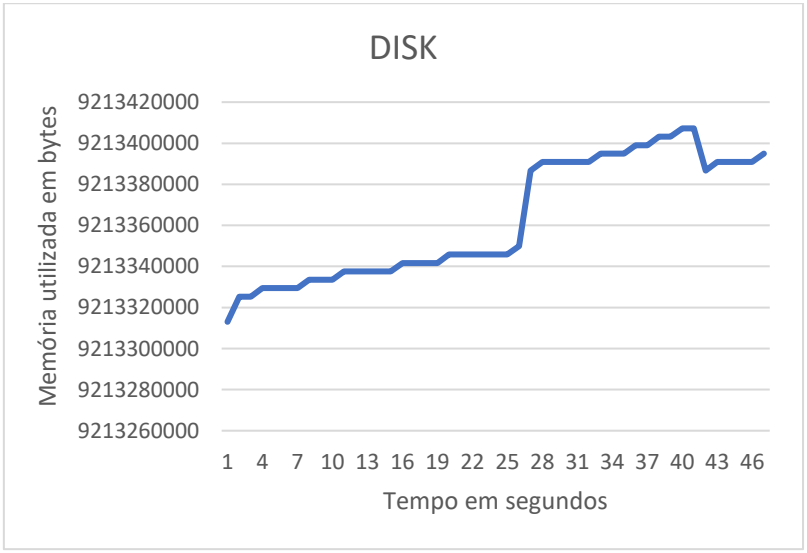

Figura A-29 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 2

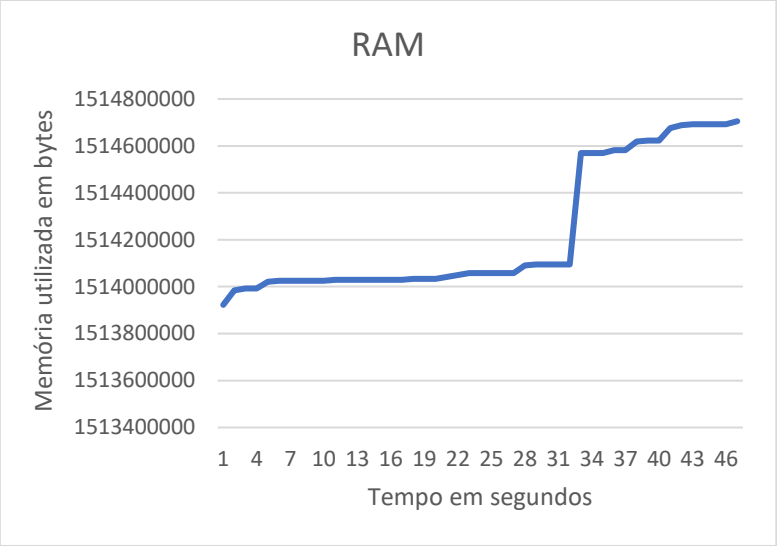

Figura A-30 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 2

Node 2

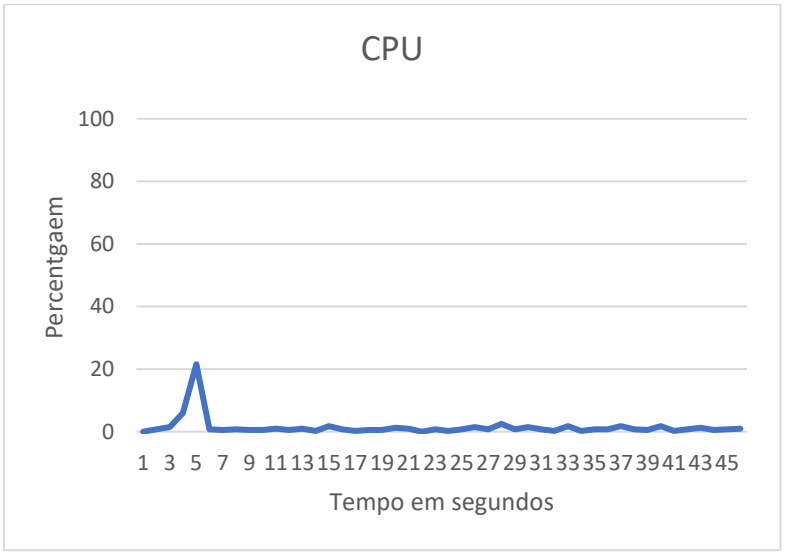

Figura A-31 - Valores de CPU do node 2 para a repetição 2 do teste 2

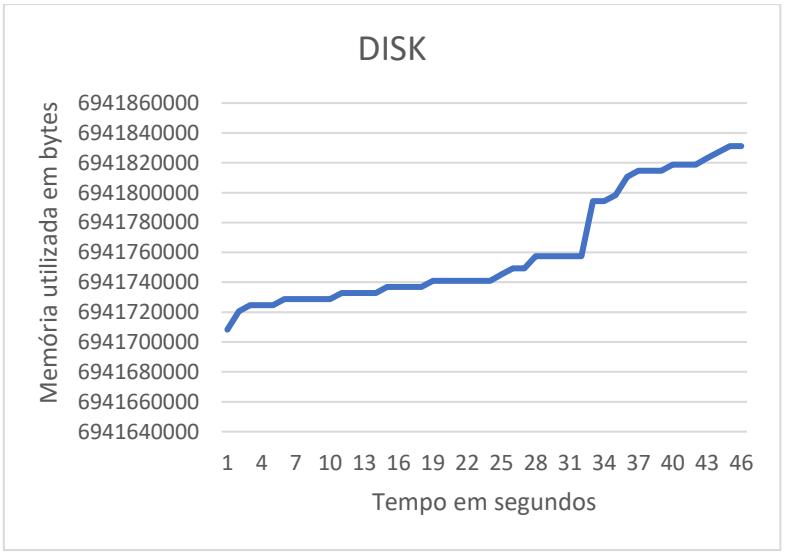

Figura A-32 - Memória utilizada de disco em bytes do node 2 para a repetição 2 do teste 2

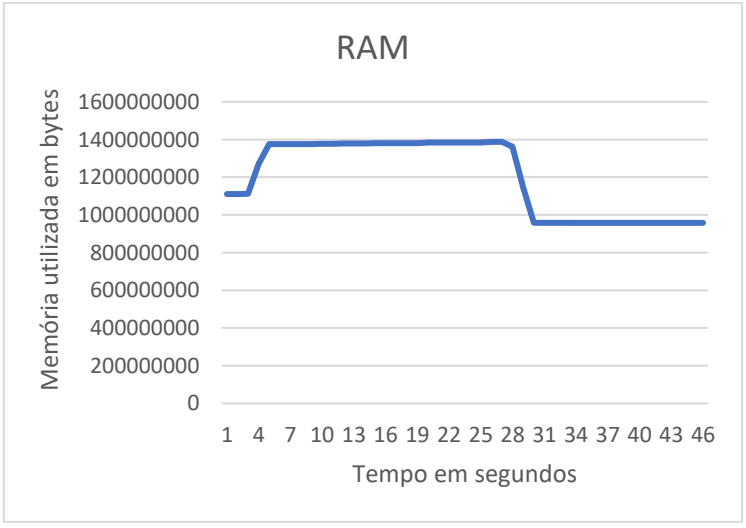

Figura A-33 - Memória em bytes que está a ser utilizada no node 2 para a repetição 2 do teste 2

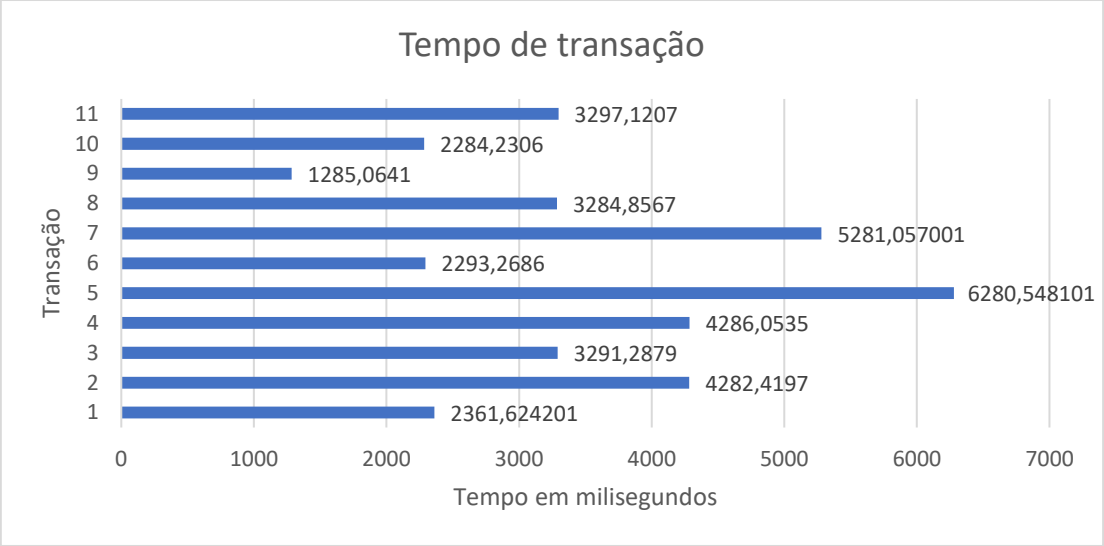

Figura A-34 - Tempo de processamento das transações realizadas na repetição 2 do teste 2

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

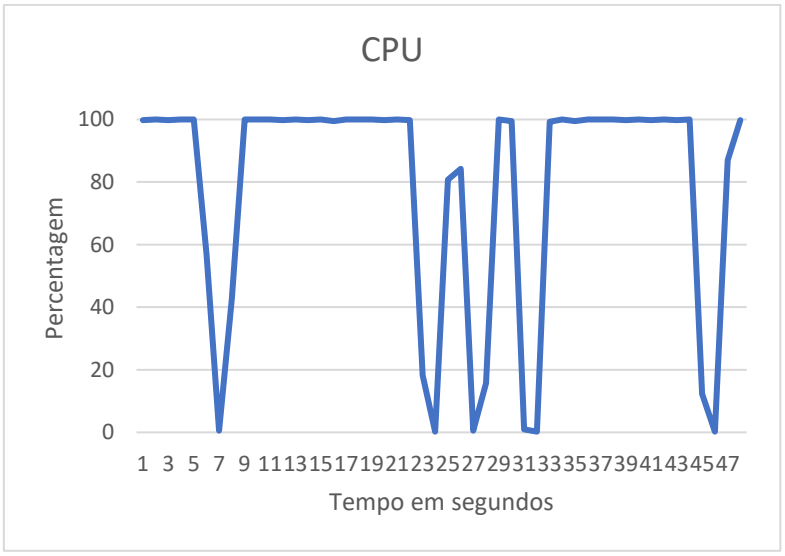

Figura A-35 - Valores de CPU do node 1 para a repetição 3 do teste 2

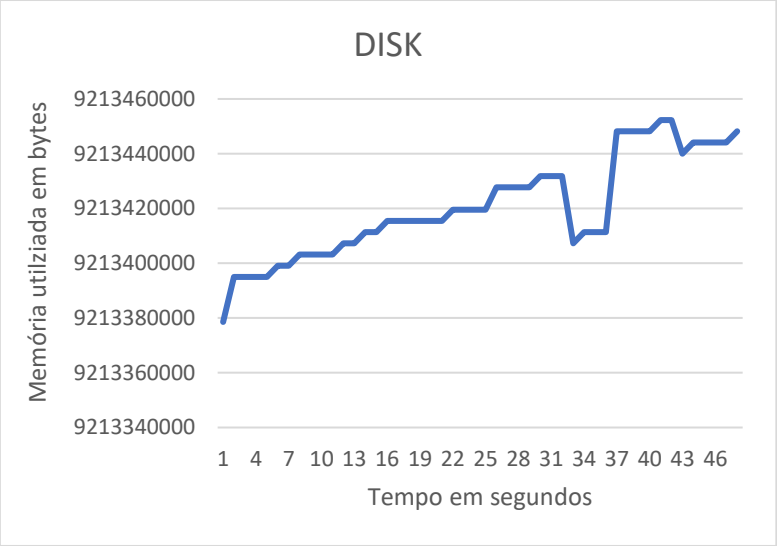

Figura A-36 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 2

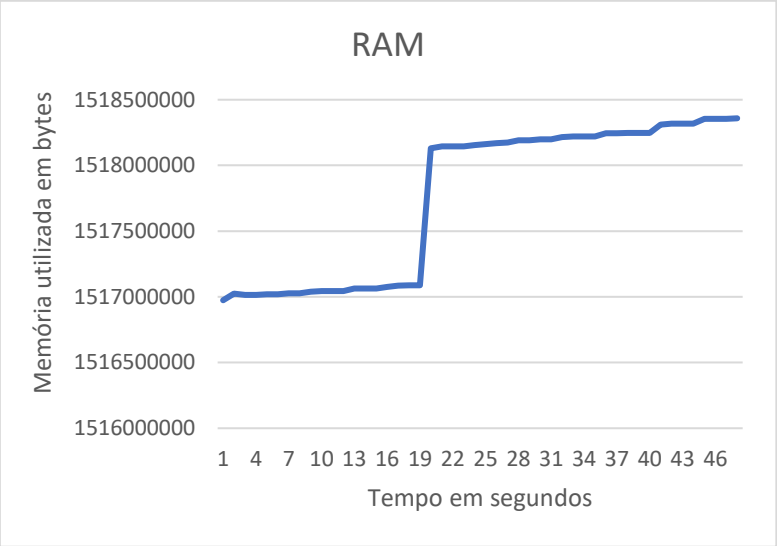

Figura A-37 - Memória em bytes que está a ser utilizada no node 1 para a repetição 3 do teste 2

Node 2

Figura A-38 - Valores de CPU do node 2 para a repetição 3 do teste 2

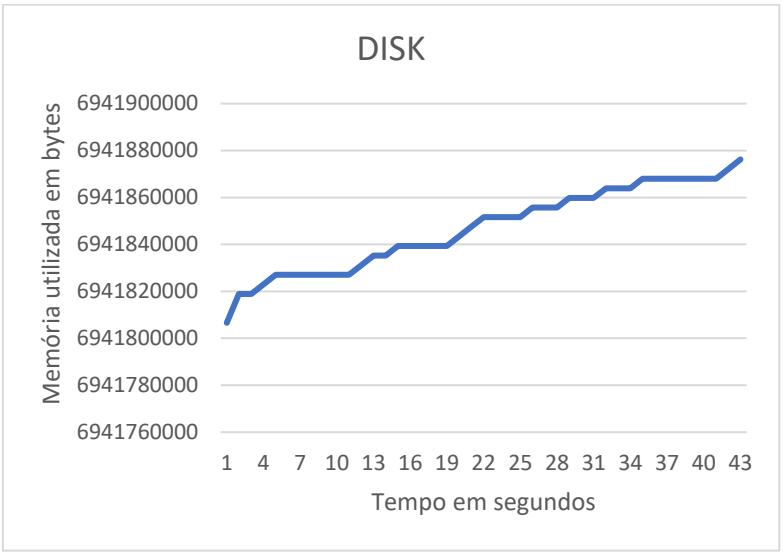

Figura A-39 - Memória utilizada de disco em bytes do node 2 para a repetição 3 do teste 2

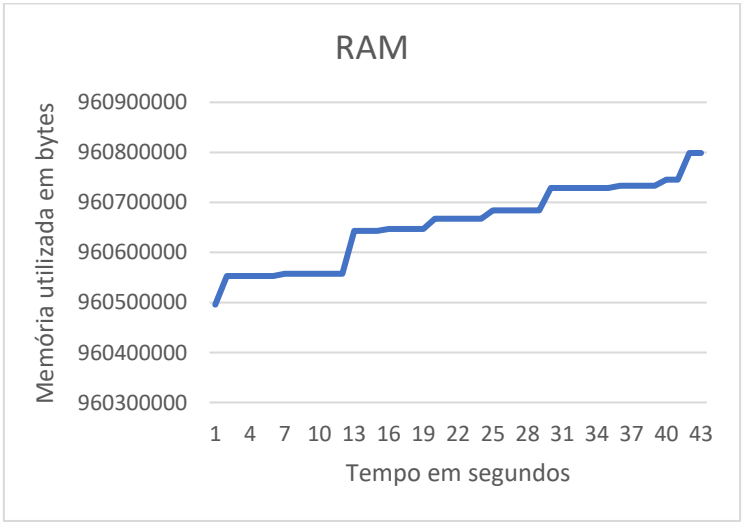

Figura A-40 - Memória em bytes que está a ser utilizada no node 2 para a repetição 3 do teste 2

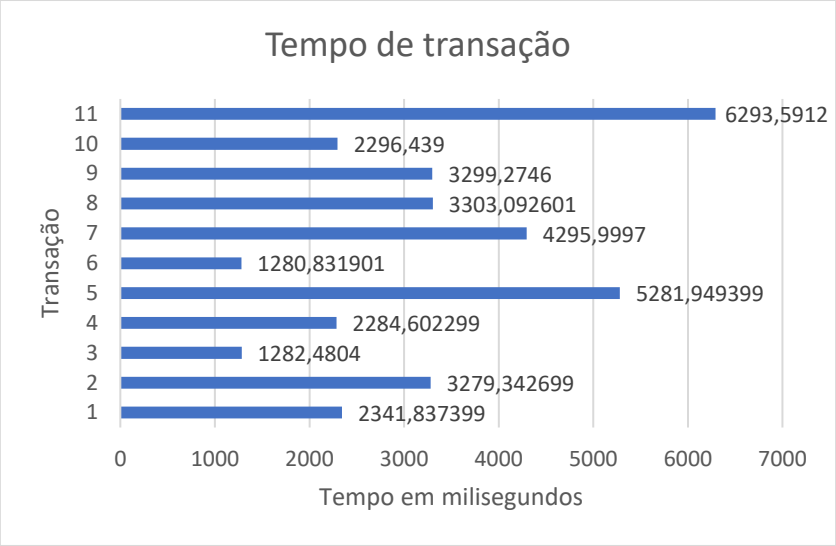

Figura A-41 - Tempo de processamento das transações realizadas na repetição 3 do teste 2

Teste 3

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

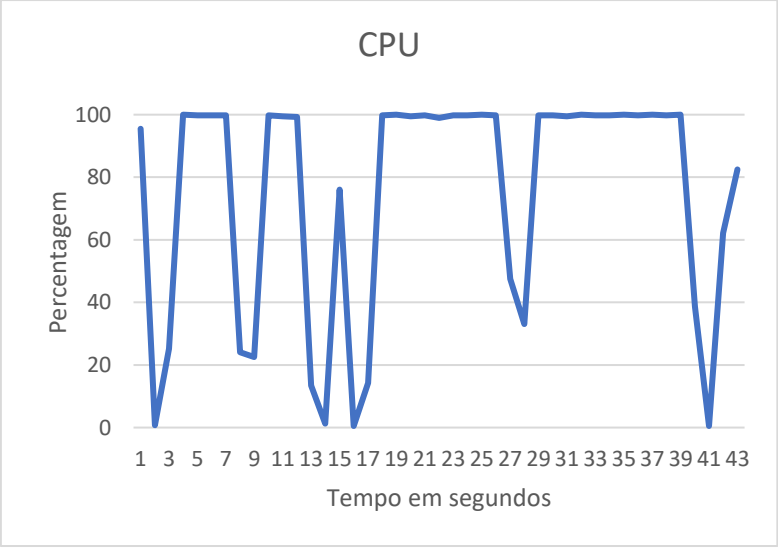

Figura A-42 - Valores de CPU do node 1 para a repetição 1 do teste 3

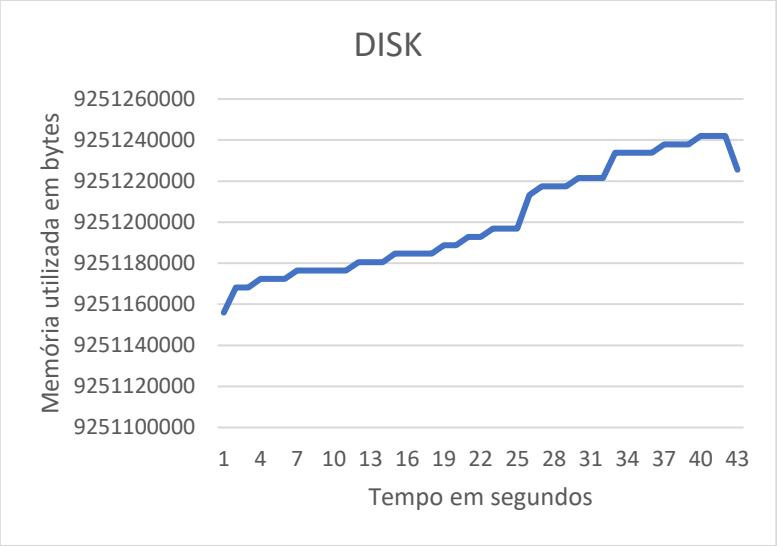

Figura A-43 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 3

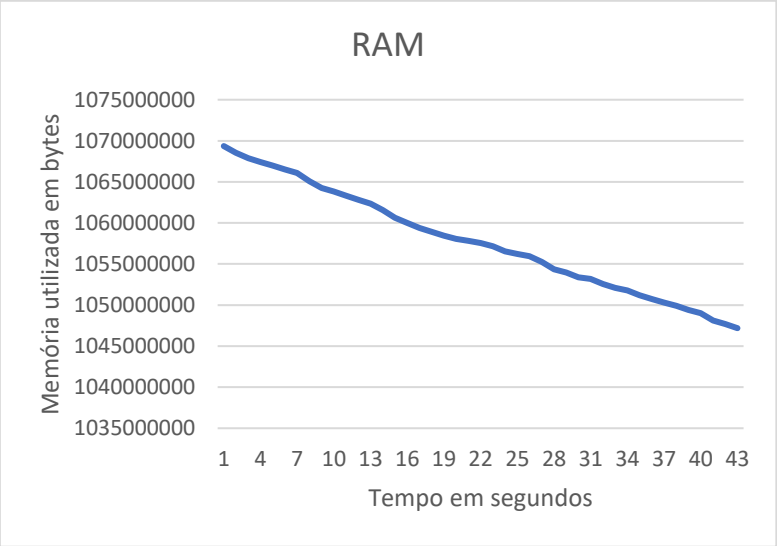

Figura A-44 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 3

Node 2 – Miner2

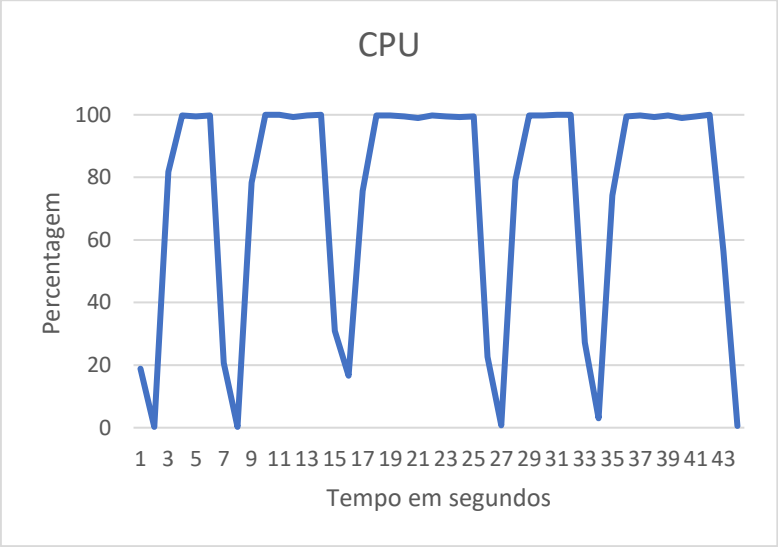

Figura A-45 - Valores de CPU do node 2 para a repetição 1 do teste 3

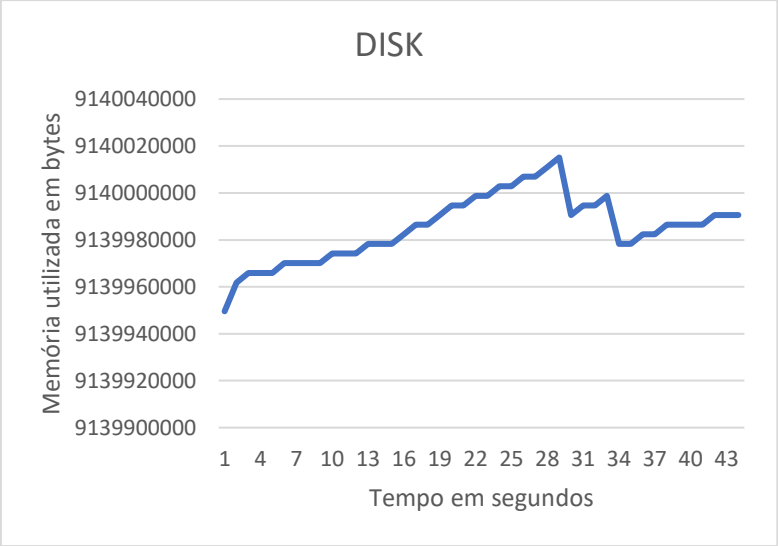

Figura A-46 - Memória utilizada de disco em bytes do node 2 para a repetição 1 do teste 3

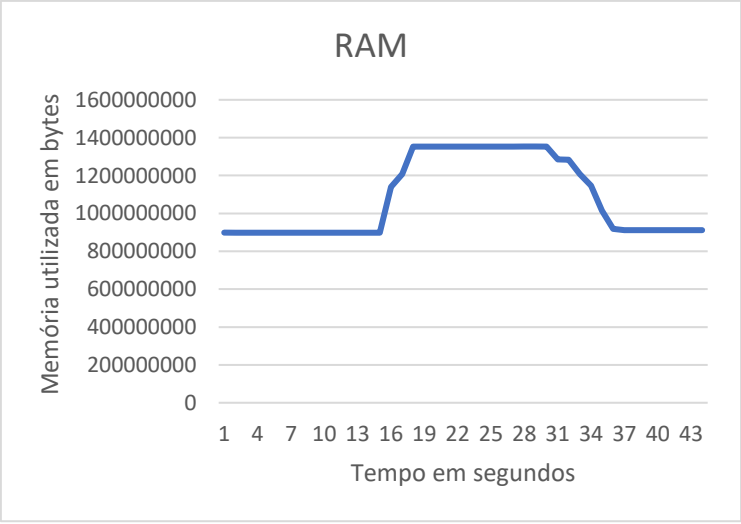

Figura A-47 - Memória em bytes que está a ser utilizada no node 2 para a repetição 1 do teste 3

Node 3

Figura A-48 - Valores de CPU do node 3 para a repetição 1 do teste 3

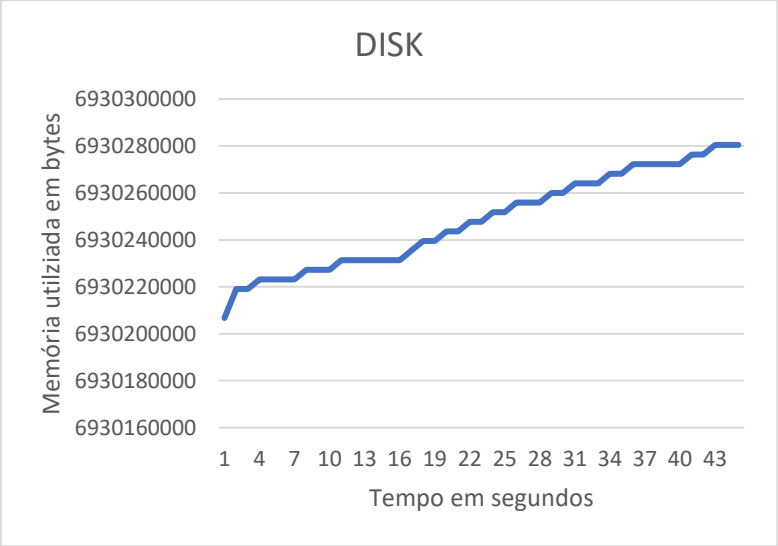

Figura A-49 - Memória utilizada de disco em bytes do node 3 para a repetição 1 do teste 3

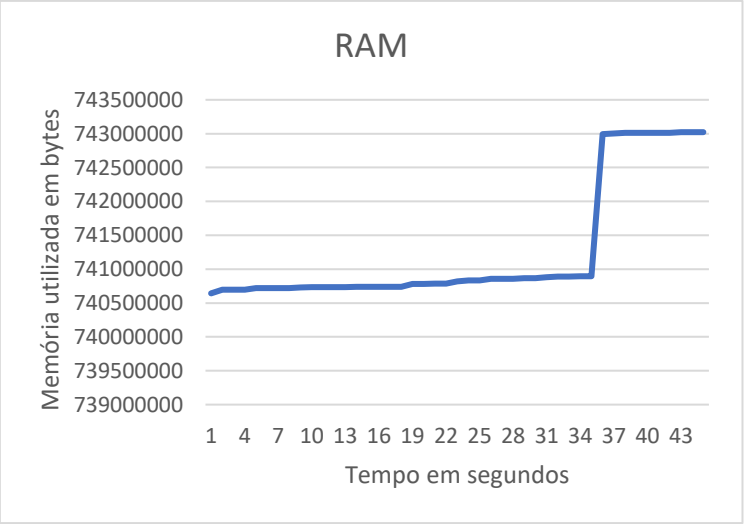

Figura A-50 - Memória em bytes que está a ser utilizada no node 2 para a repetição 1 do teste 3

Node 4

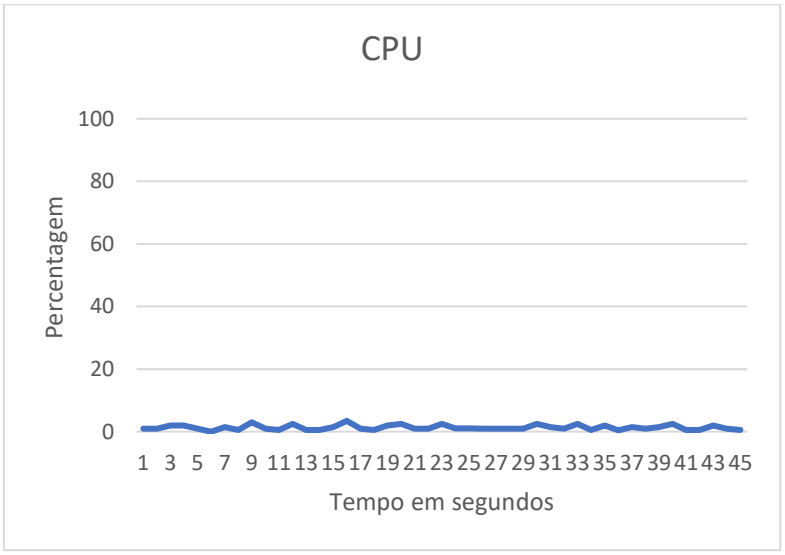

Figura A-51 - Valores de CPU do node 4 para a repetição 1 do teste 3

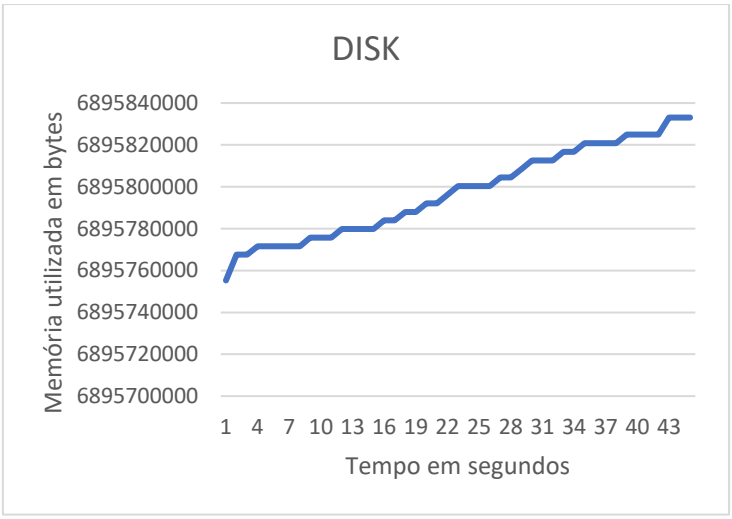

Figura A-52 - Memória utilizada de disco em bytes do node 4 para a repetição 1 do teste 3

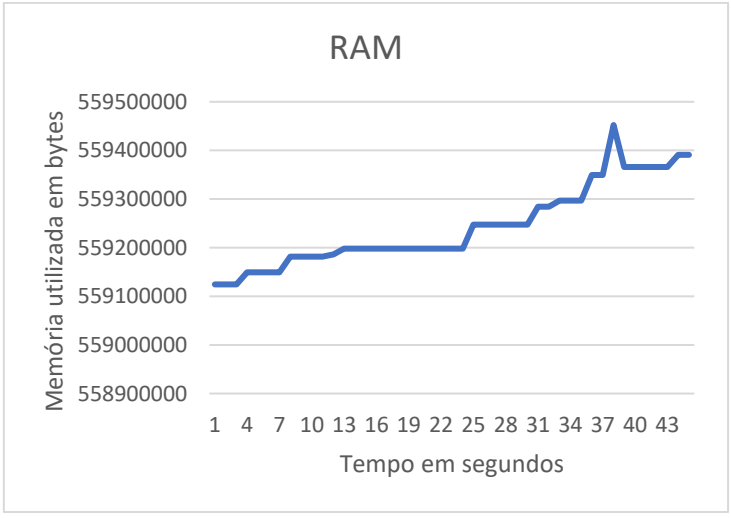

Figura A-53 - Memória em bytes que está a ser utilizada no node 3 para a repetição 1 do teste 3

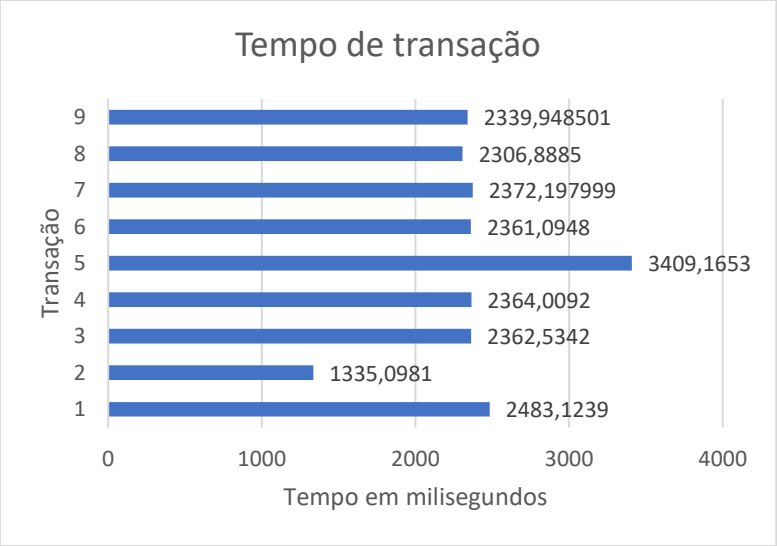

Figura A-54 - Tempo de processamento das transações realizadas na repetição 1 do teste 3

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

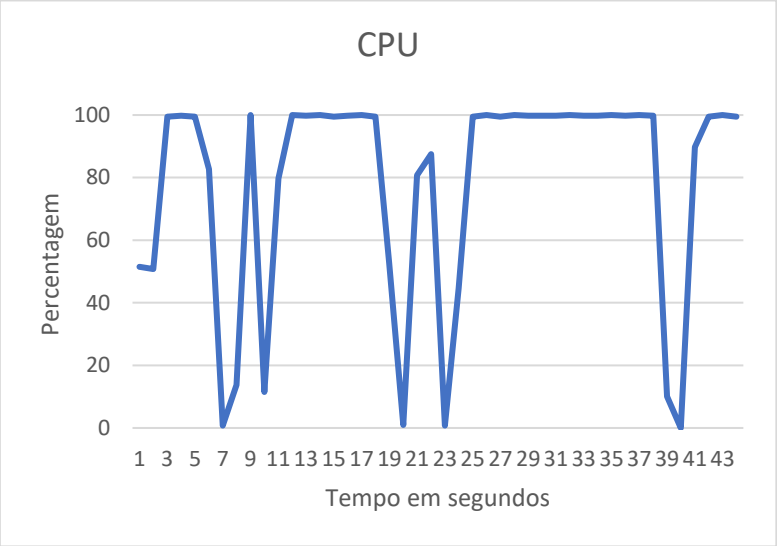

Figura A-55 - Valores de CPU do node 1 para a repetição 2 do teste 3

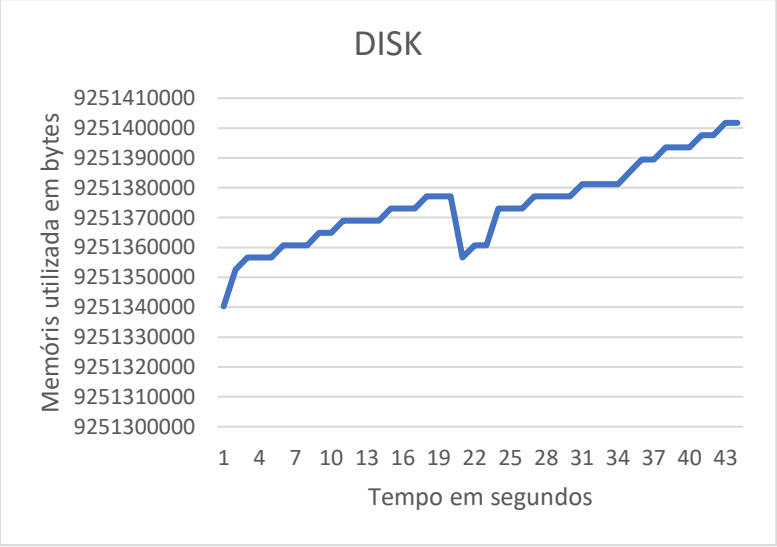

Figura A-56 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 3

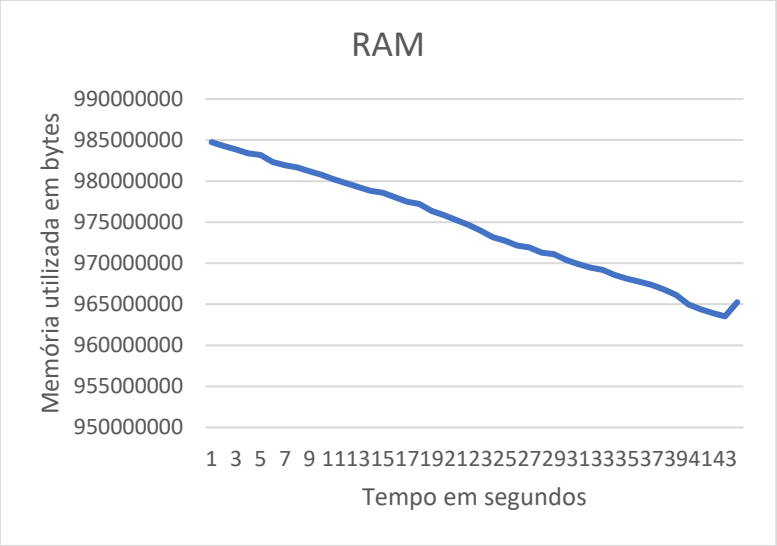

Figura A-57 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 3

Node 2 – Miner2

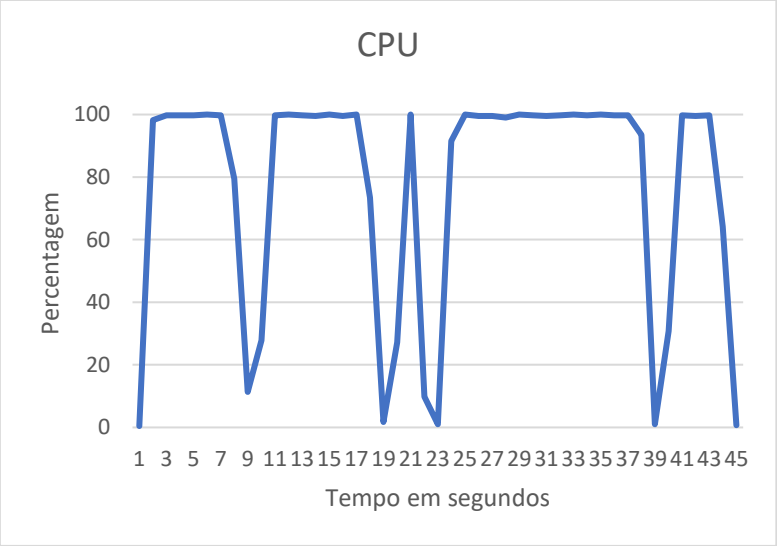

Figura A-58 - Valores de CPU do node 2 para a repetição 2 do teste 3

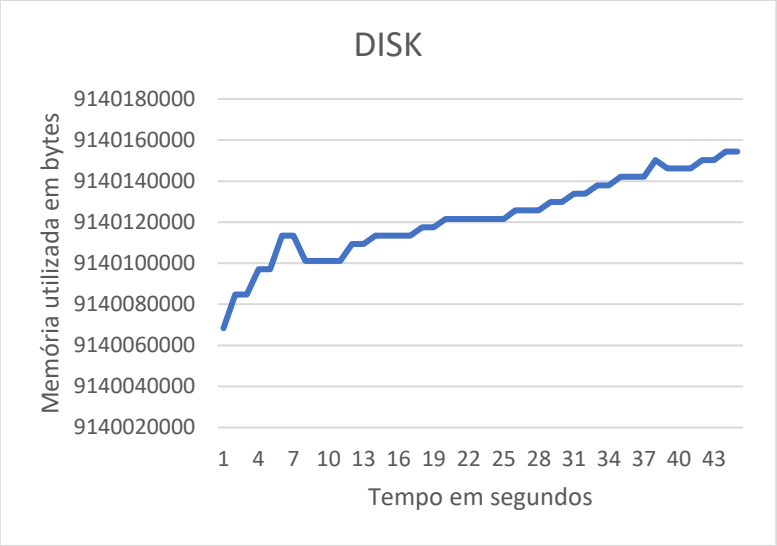

Figura A-59 - Memória utilizada de disco em bytes do node 2 para a repetição 2 do teste 3

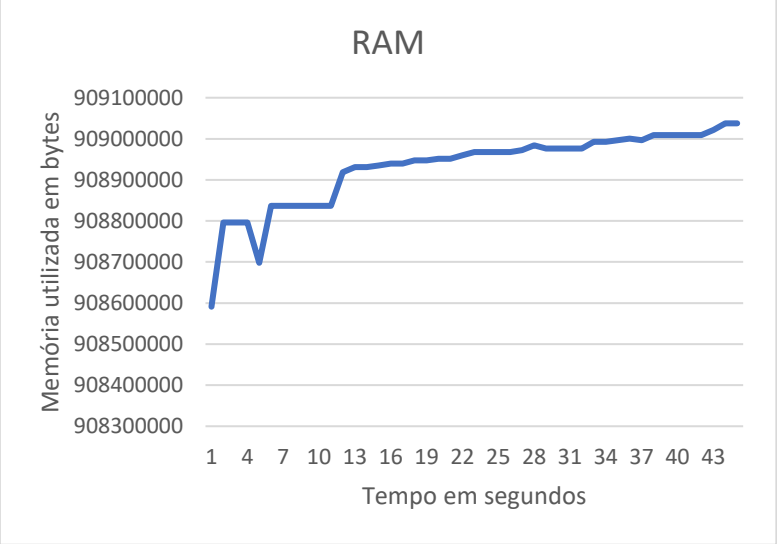

Figura A-60 - Memória em bytes que está a ser utilizada no node 2 para a repetição 2 do teste 3

Node 3

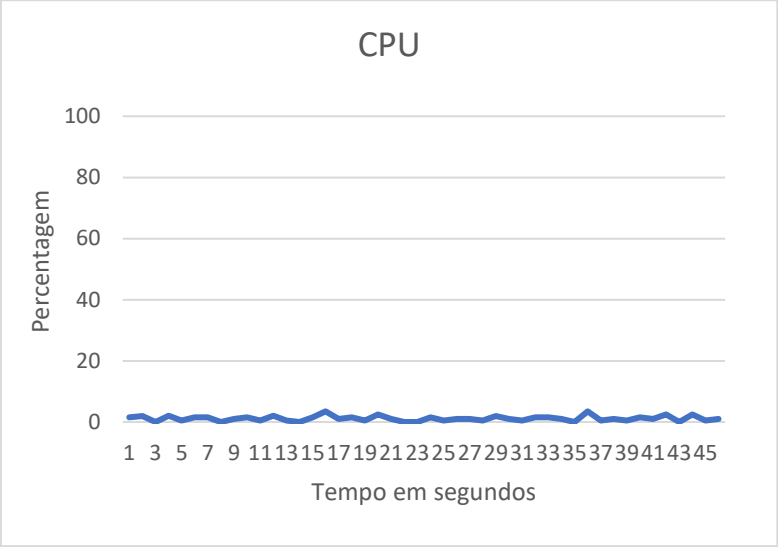

Figura A-61 - Valores de CPU do node 3 para a repetição 2 do teste 3

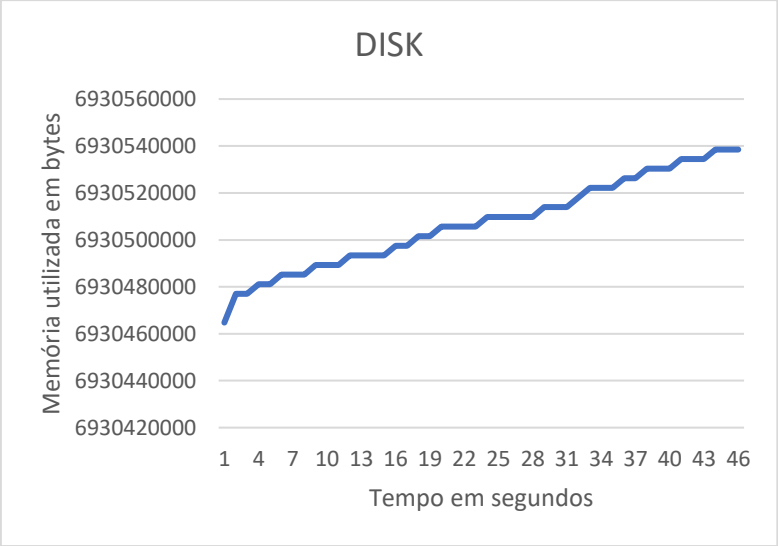

Figura A-62 - Memória utilizada de disco em bytes do node 3 para a repetição 2 do teste 3

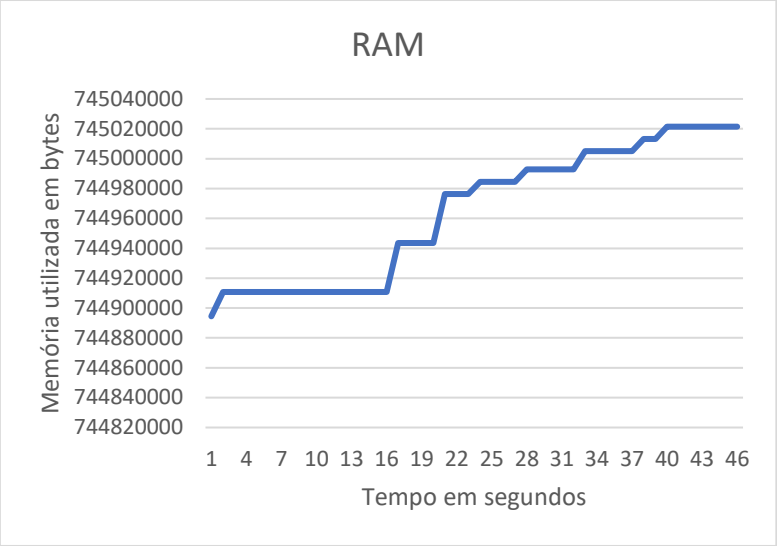

Figura A-63 - Memória em bytes que está a ser utilizada no node 3 para a repetição 2 do teste 3

Node 4

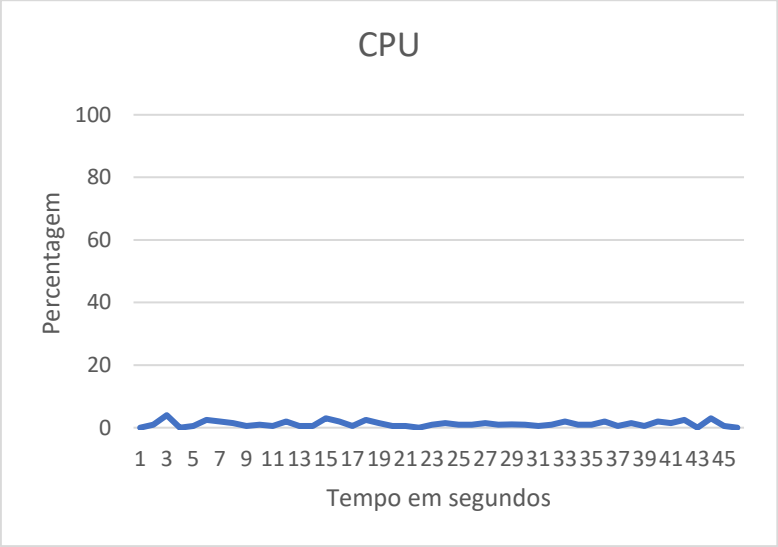

Figura A-64 - Valores de CPU do node 4 para a repetição 2 do teste 3

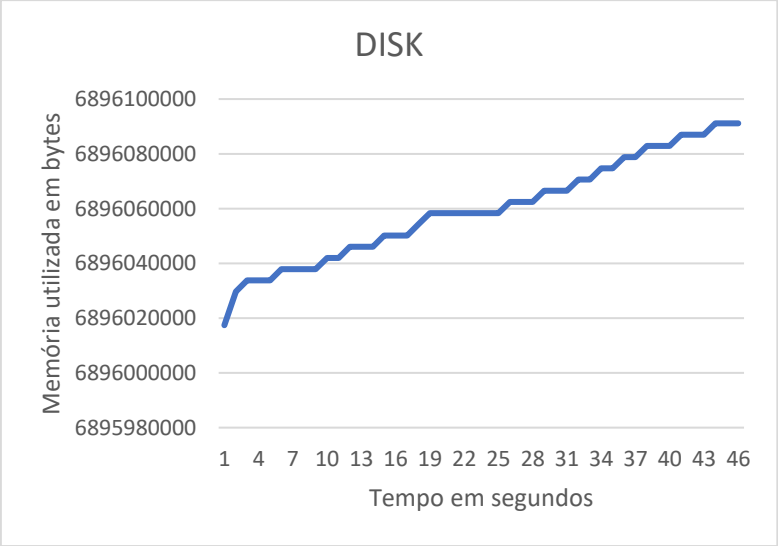

Figura A-65 - Memória utilizada de disco em bytes do node 4 para a repetição 2 do teste 3

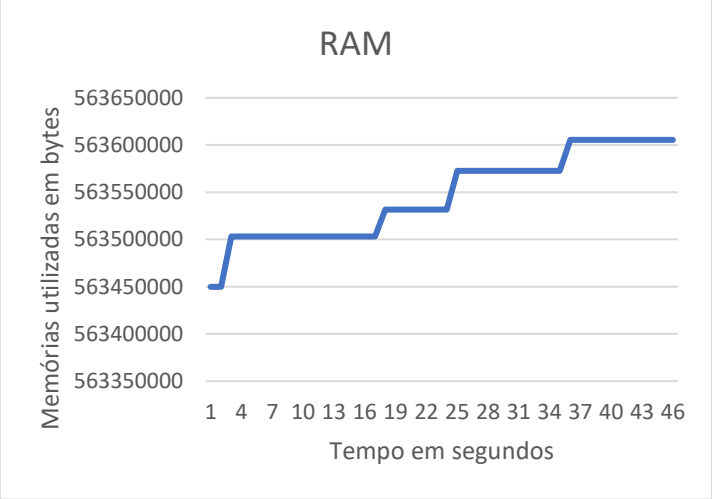

Figura A-66 - Memória em bytes que está a ser utilizada no node4 para a repetição 2 do teste 3

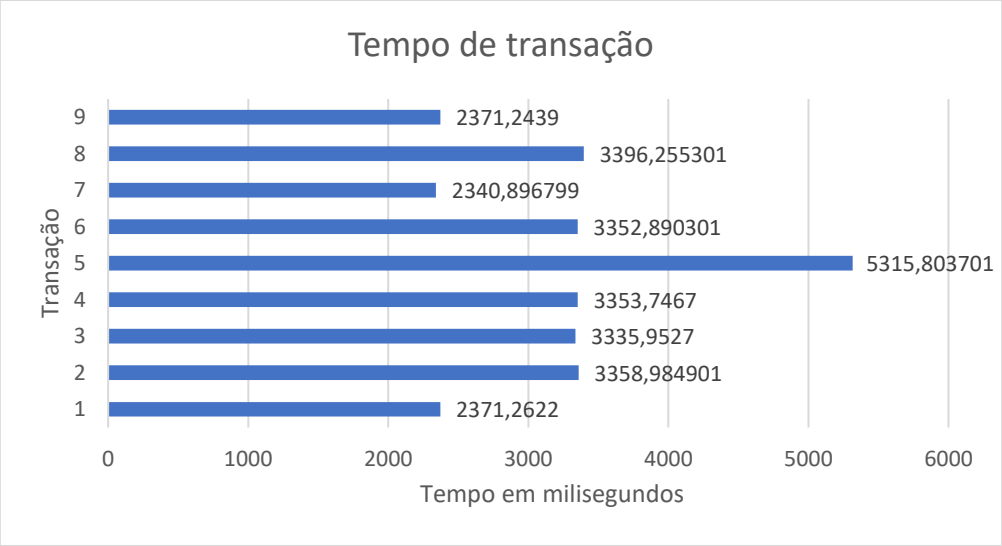

Figura A-67 - Tempo de processamento das transações realizadas na repetição 2 do teste 3

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

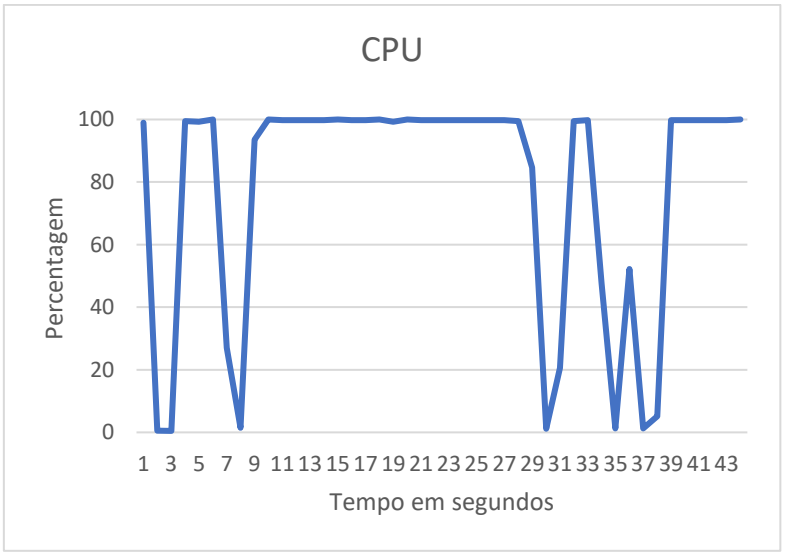

Figura A-68 - Valores de CPU do node 1 para a repetição 3 do teste 3

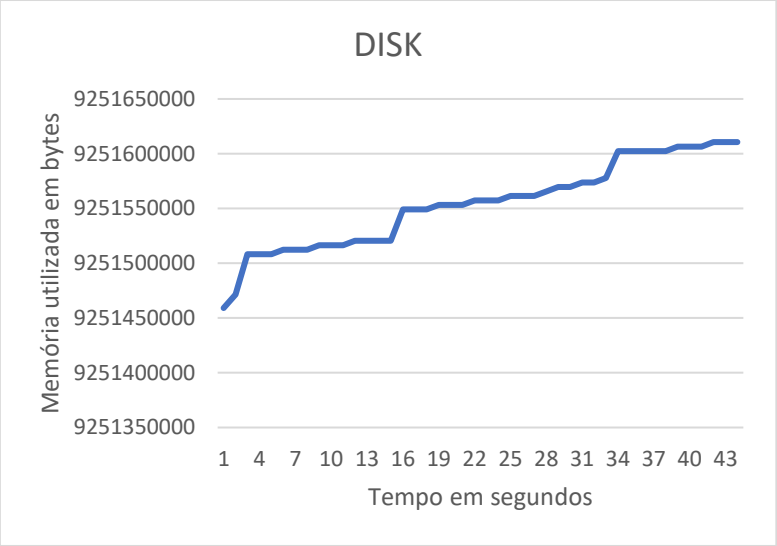

Figura A-69 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 3

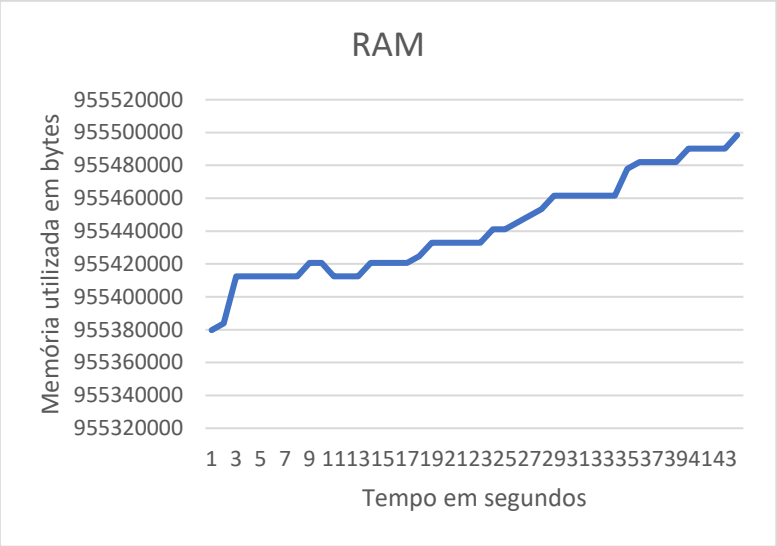

Figura A-70 - Memória em bytes que está a ser utilizada no node1 para a repetição 3 do teste 3

Node 2 – Miner2

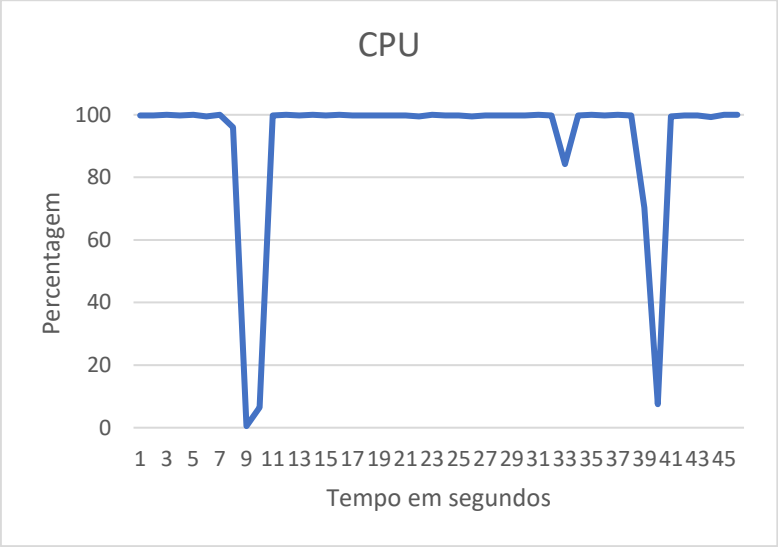

Figura A-71 - Valores de CPU do node 2 para a repetição 3 do teste 3

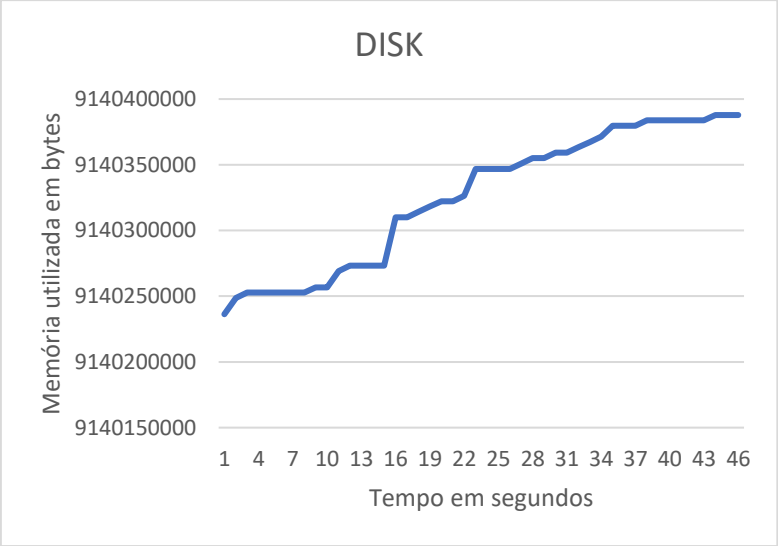

Figura A-72 - Memória utilizada de disco em bytes do node 2 para a repetição 3 do teste 3

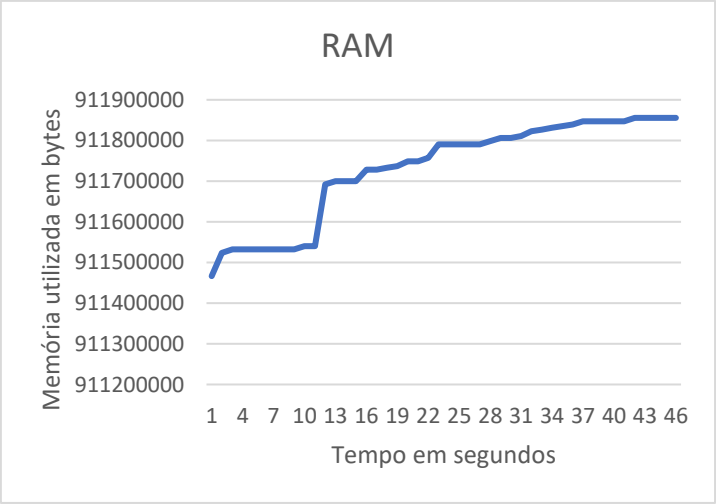

Figura A-73 - Memória em bytes que está a ser utilizada no node2 para a repetição 3 do teste 3

Node 3

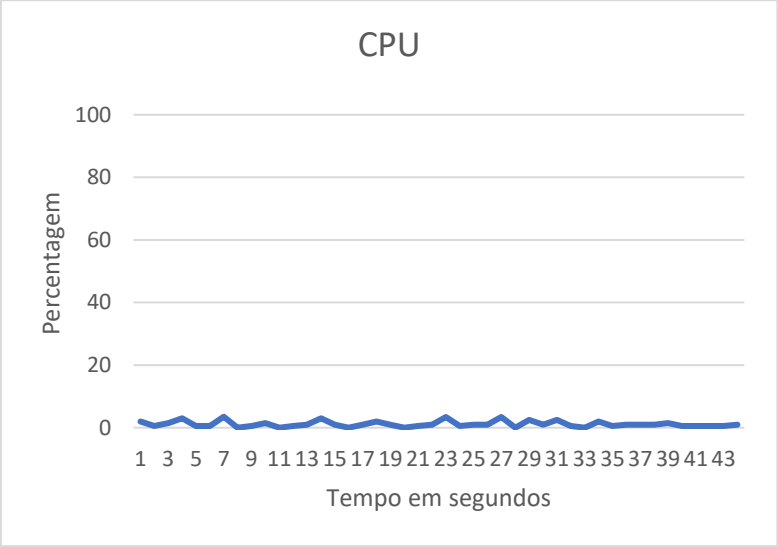

Figura A-74 - Valores de CPU do node3 para a repetição 3 do teste 3

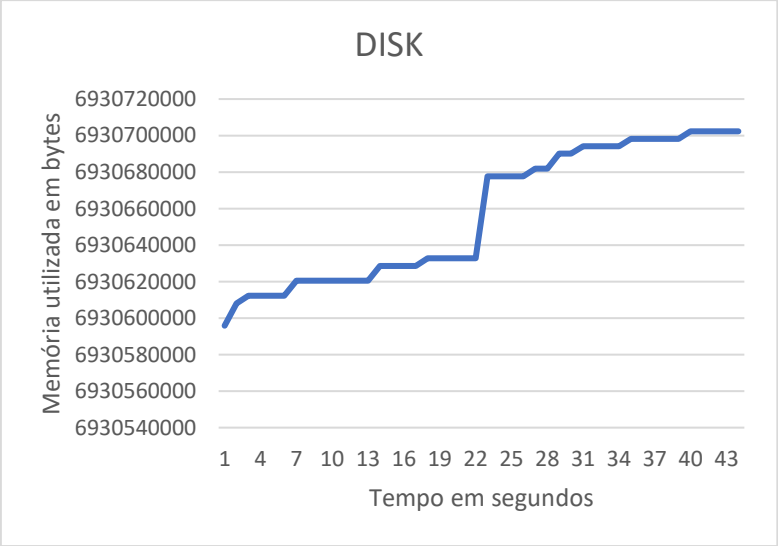

Figura A-75 - Memória utilizada de disco em bytes do node 3 para a repetição 3 do teste 3

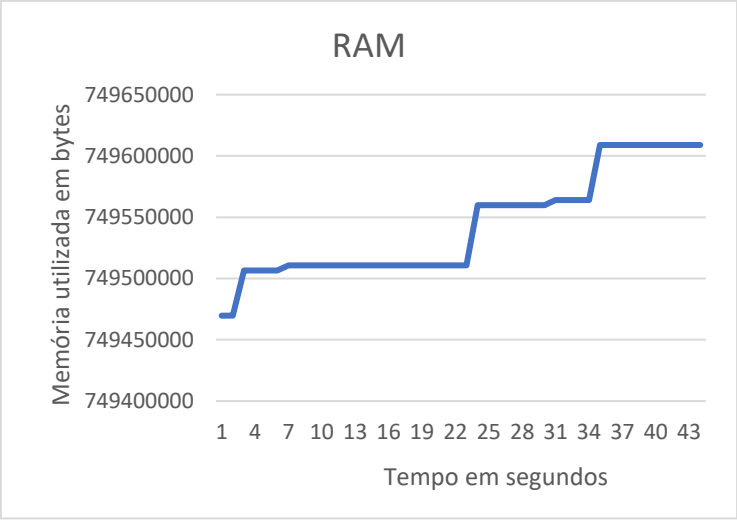

Figura A-76 - Memória em bytes que está a ser utilizada no node3 para a repetição 3 do teste 3

Node 4

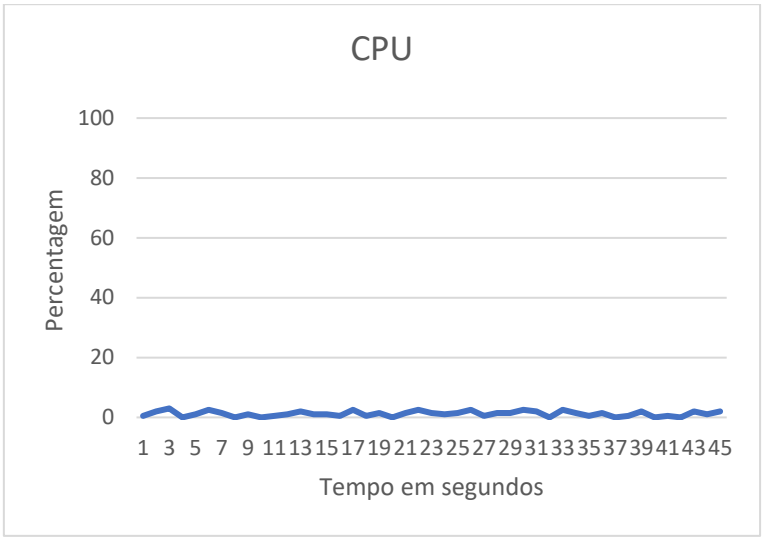

Figura A-77 - Valores de CPU do node 4 para a repetição 3 do teste 3

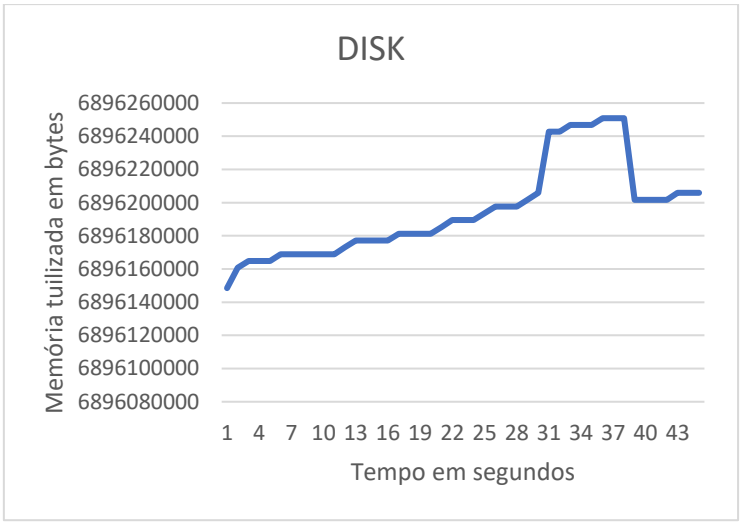

Figura A-78 - Memória utilizada de disco em bytes do node 4 para a repetição 3 do teste 3

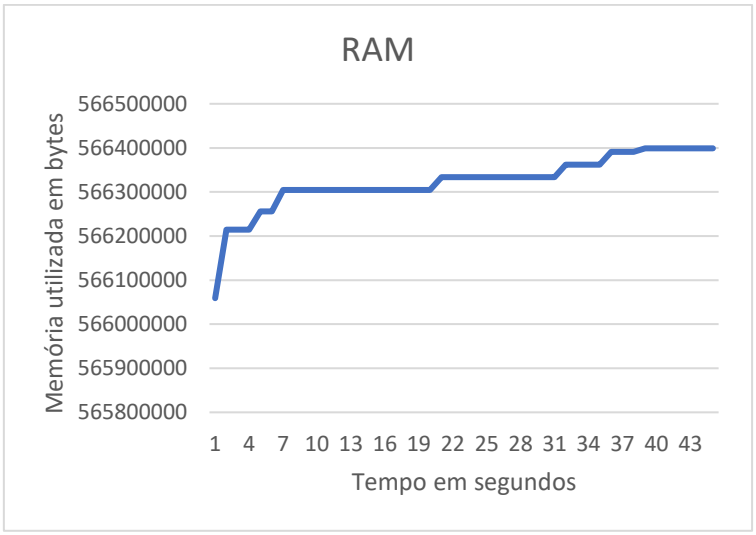

Figura A-79 - Memória em bytes que está a ser utilizada no node4 para a repetição 3 do teste 3

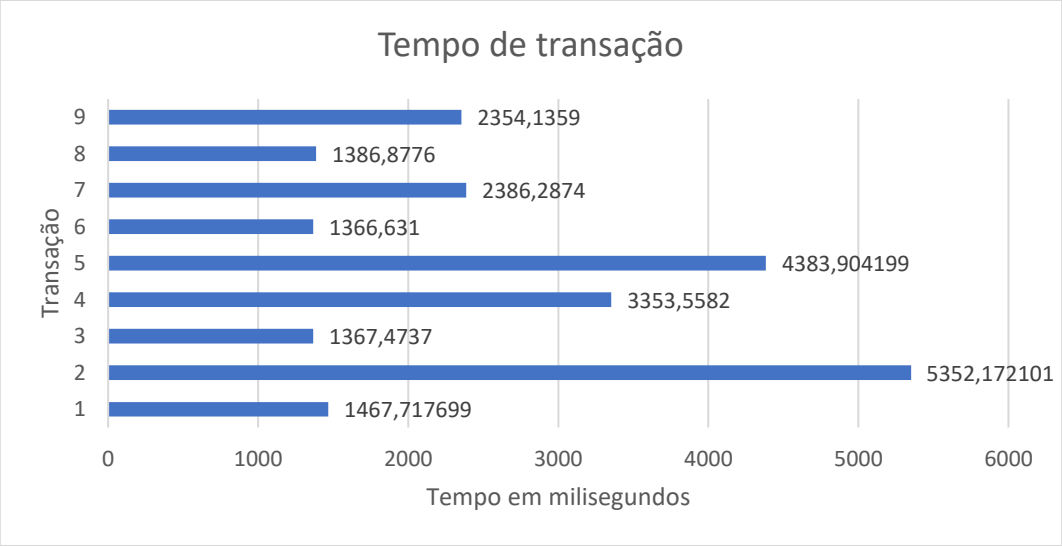

Figura A-80 - Tempo de processamento das transações realizadas na repetição 3 do teste 3

Teste 4

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

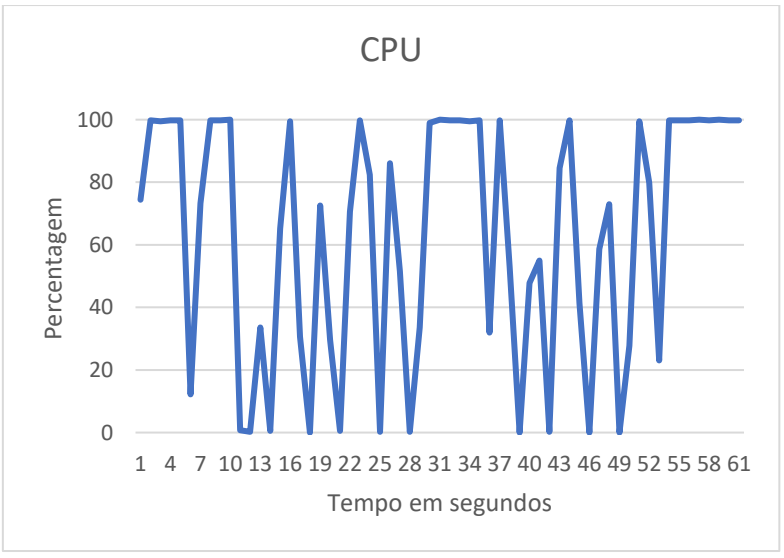

Figura A-81 - Valores de CPU do node 1 para a repetição 1 do teste 4

Figura A-82 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 4

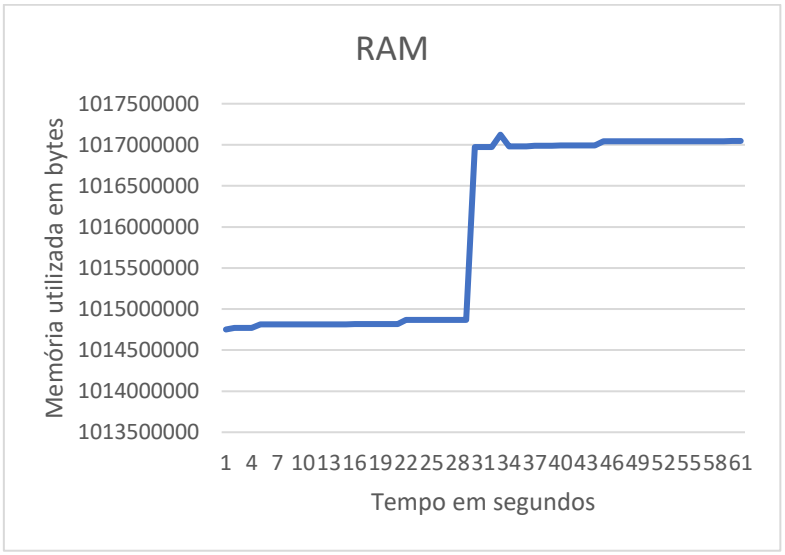

Figura A-83 - Memória em bytes que está a ser utilizada no node1 para a repetição 1 do teste 4

Node 2 – Miner2

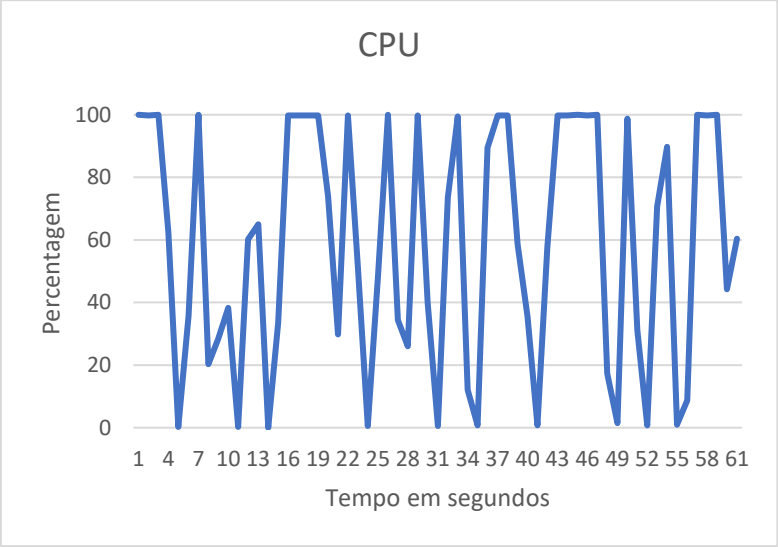

Figura A-84 - Valores de CPU do node 2 para a repetição 1 do teste 4

Figura A-85 - Memória utilizada de disco em bytes do node 2 para a repetição 1 do teste 4

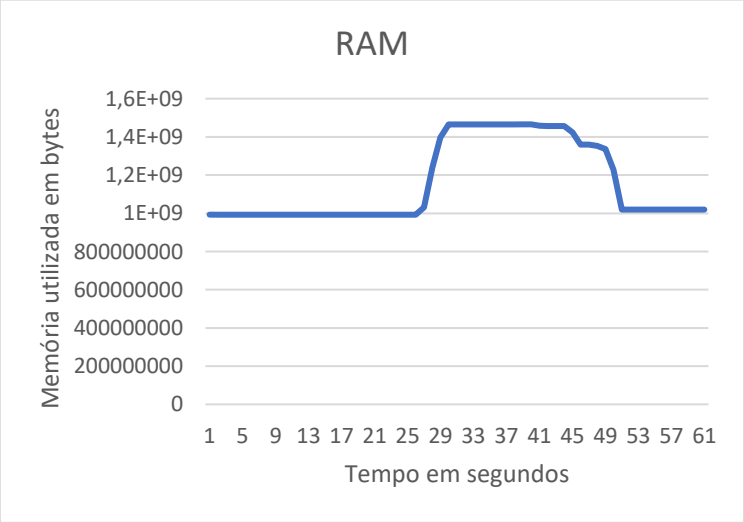

Figura A-86 - Memória em bytes que está a ser utilizada no node2 para a repetição 1 do teste 4

Node 3 – Miner3

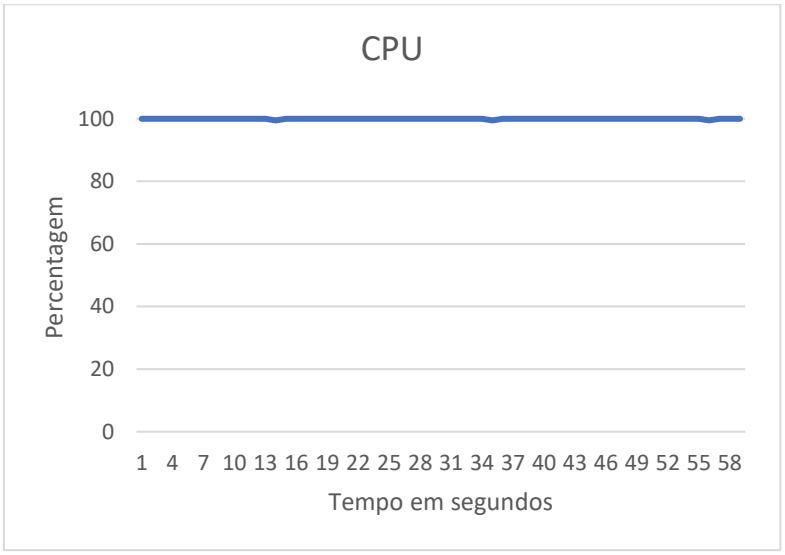

Figura A-87 - Valores de CPU do node 3 para a repetição 1 do teste 4

Figura A-88 - Memória utilizada de disco em bytes do node 3 para a repetição 1 do teste 4

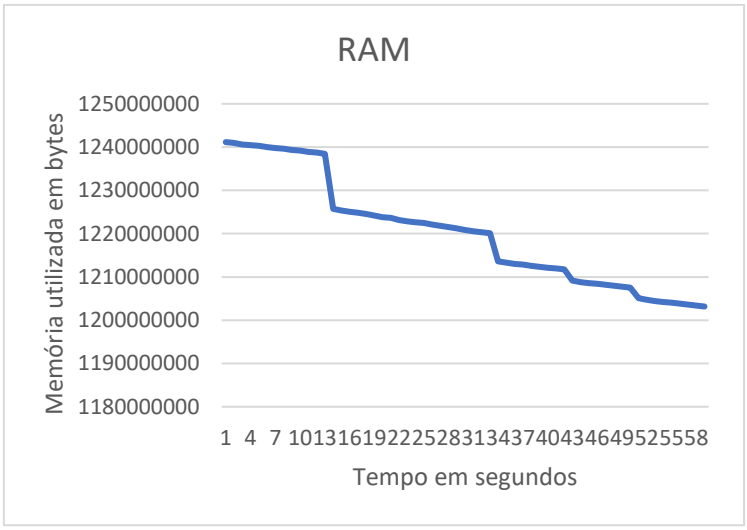

Figura A-89 - Memória em bytes que está a ser utilizada no node 3 para a repetição 1 do teste 4

Node 4

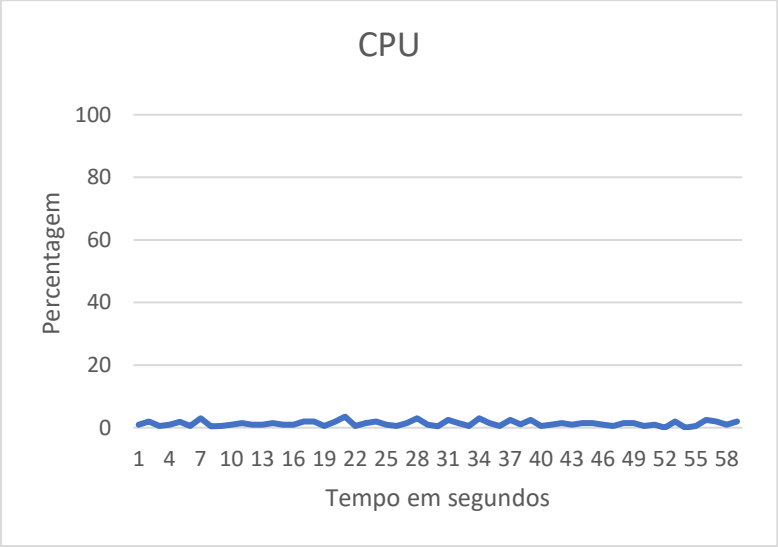

Figura A-90 - Valores de CPU do node 4 para a repetição 1 do teste 4

Figura A-91 - Memória utilizada de disco em bytes do node 4 para a repetição 1 do teste 4

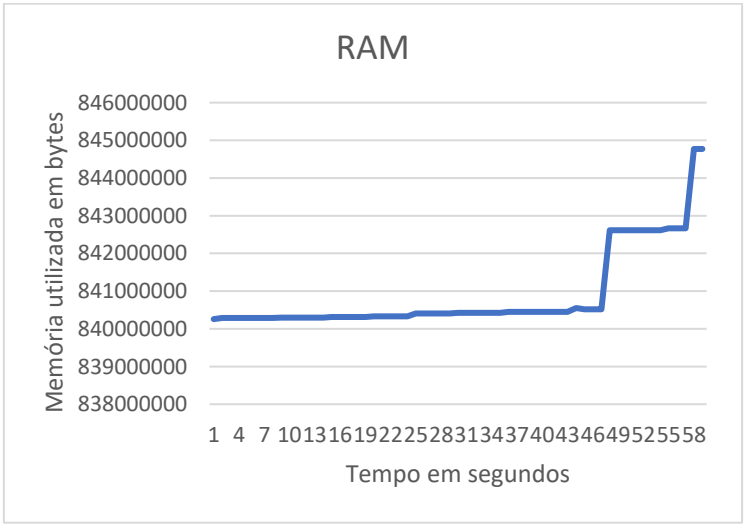

Figura A-92 - Memória em bytes que está a ser utilizada no node 4 para a repetição 1 do teste 4

Node 5

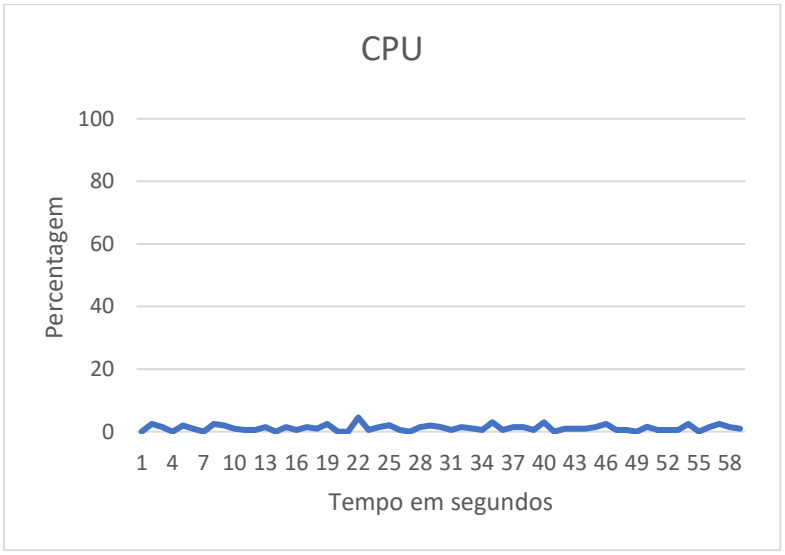

Figura A-93 - Valores de CPU do node 5 para a repetição 1 do teste 4

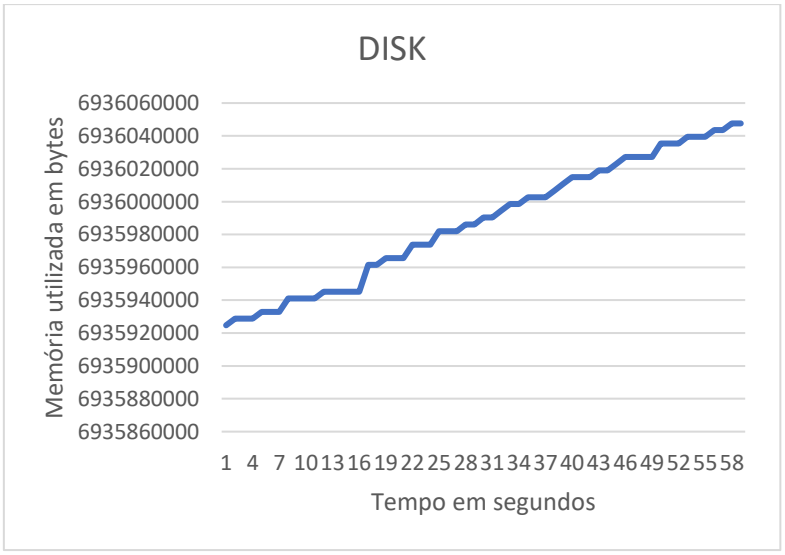

Figura A-94 - Memória utilizada de disco em bytes do node 5 para a repetição 1 do teste 4

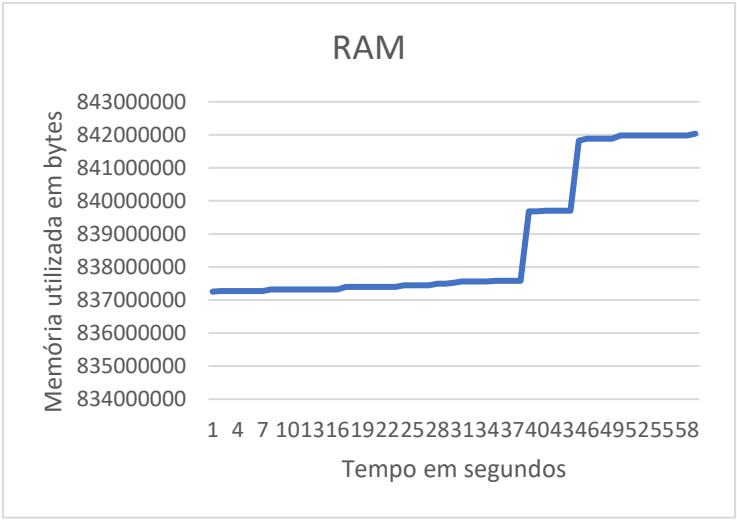

Figura A-95 - Memória em bytes que está a ser utilizada no node 5 para a repetição 1 do teste 4

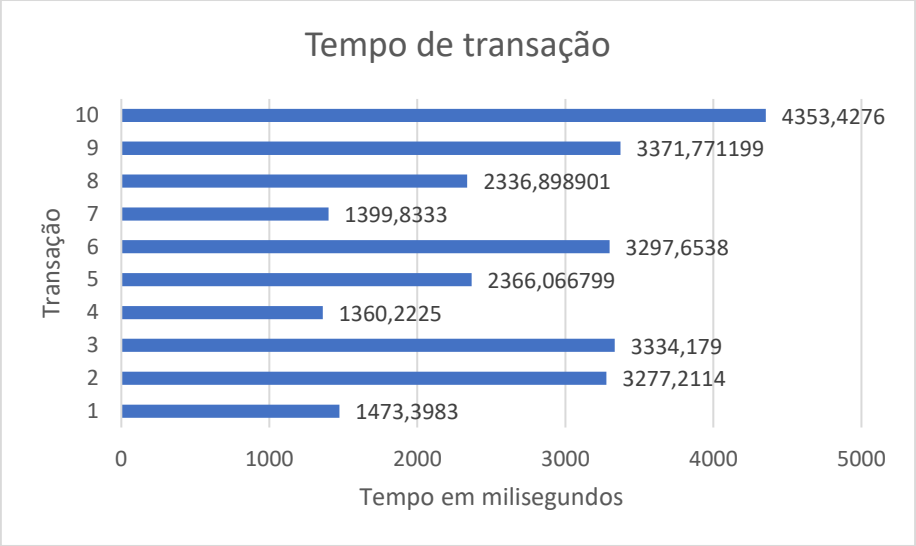

Figura A-96 - Tempo de processamento das transações realizadas na repetição 1 do teste 4

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

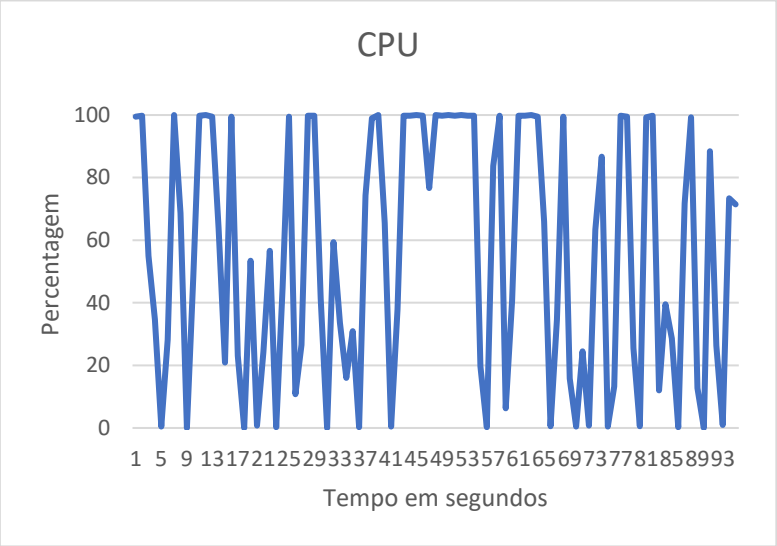

Figura A-97 - Valores de CPU do node 1 para a repetição 2 do teste 4

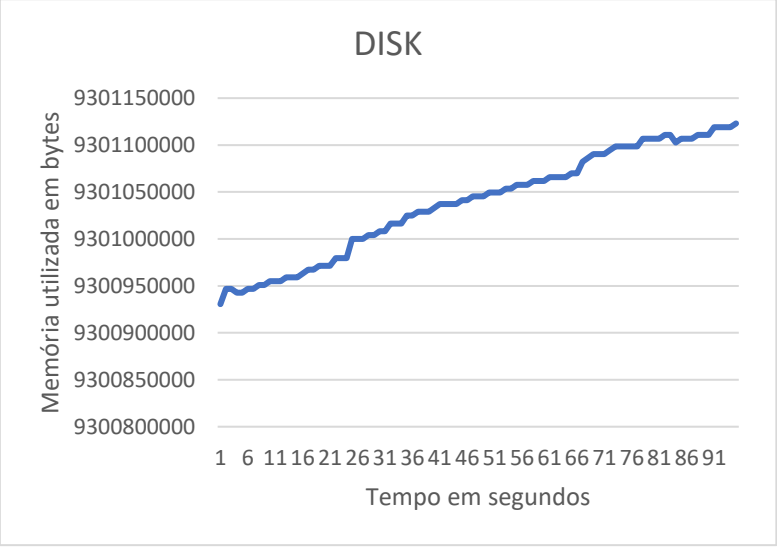

Figura A-98 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 4

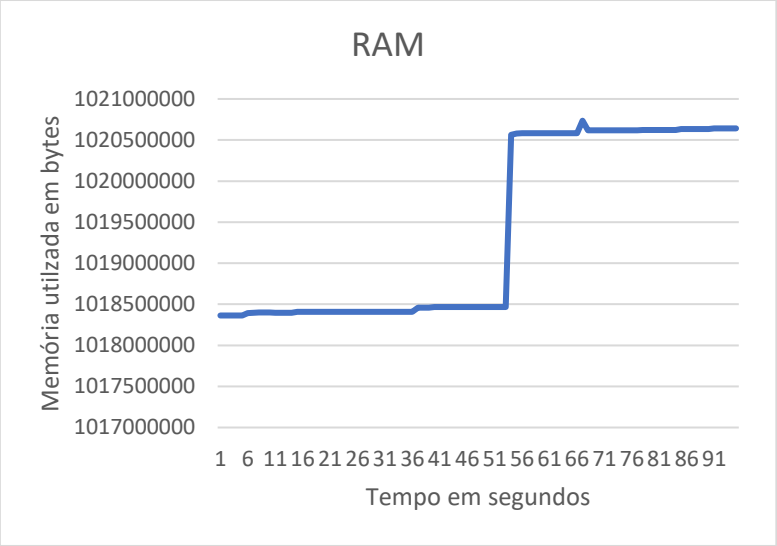

Figura A-99 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 4

Node 2 – Miner2

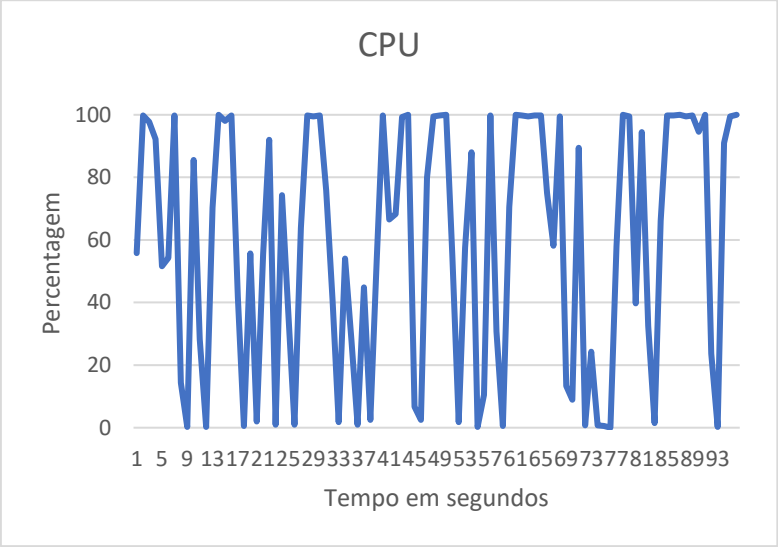

Figura A-100 - Valores de CPU do node 2 para a repetição 2 do teste 4

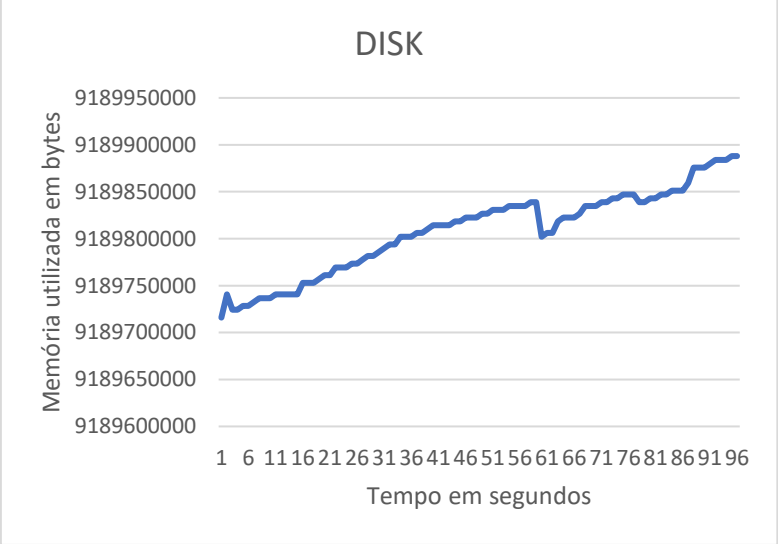

Figura A-101 - Memória utilizada de disco em bytes do node 2 para a repetição 2 do teste 4

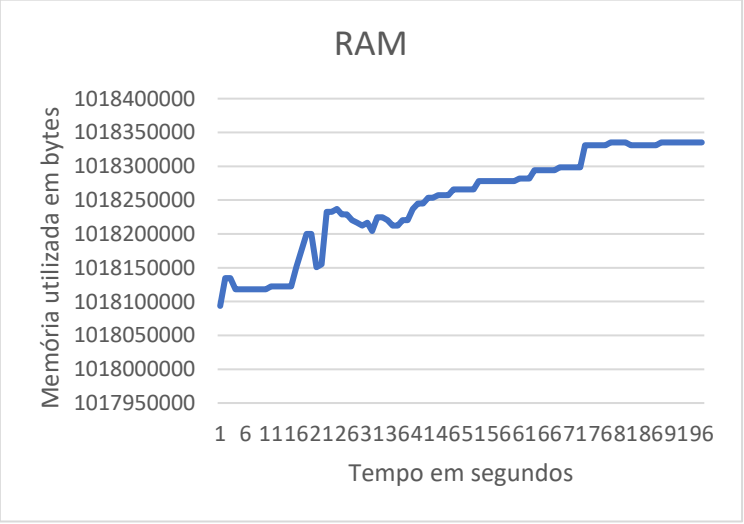

Figura A-102 - Memória em bytes que está a ser utilizada no node 2 para a repetição 2 do teste 4

Node 3 – Miner3

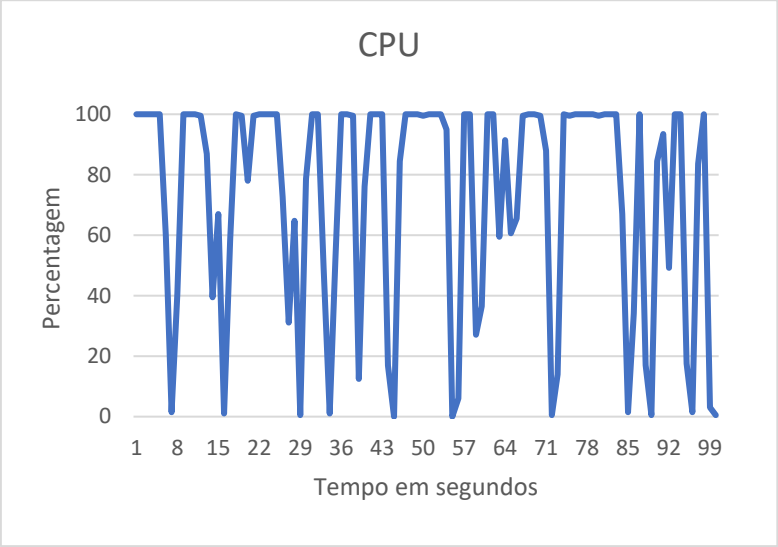

Figura A-103 - Valores de CPU do node 3 para a repetição 2 do teste 4

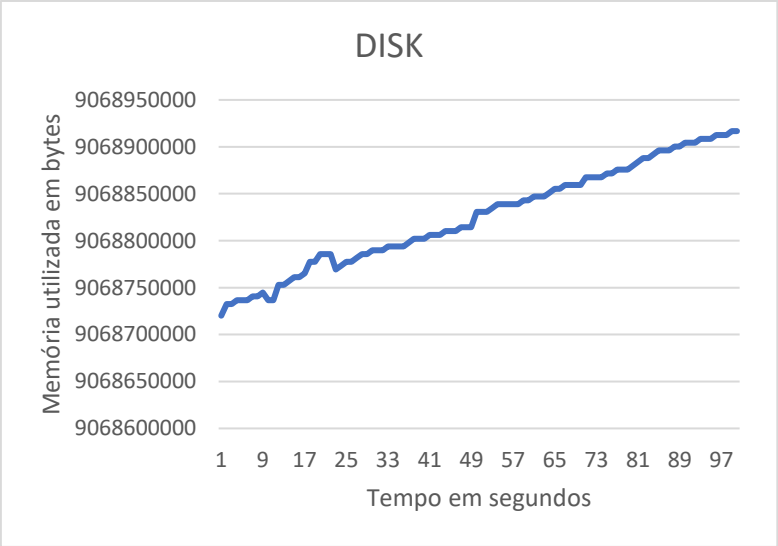

Figura A-104 - Memória utilizada de disco em bytes do node 3 para a repetição 2 do teste 4

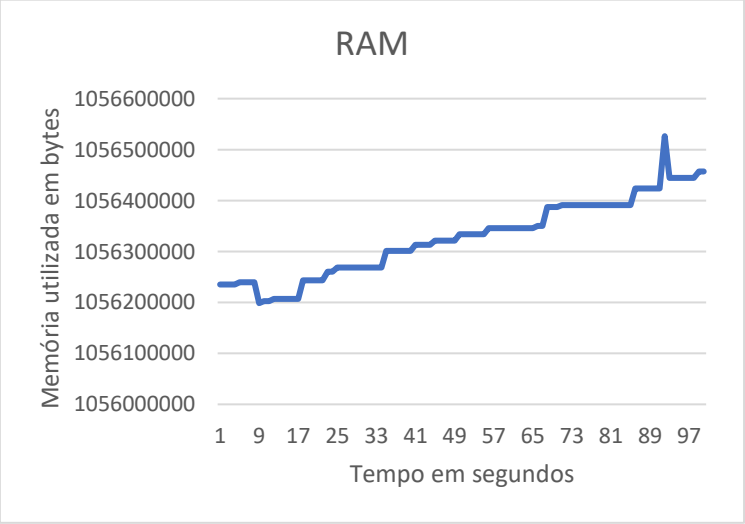

Figura A-105 - Memória em bytes que está a ser utilizada no node 3 para a repetição 2 do teste 4

Node 4

Figura A-106 - Valores de CPU do node 4 para a repetição 2 do teste 4

Figura A-107 - Memória utilizada de disco em bytes do node 4 para a repetição 2 do teste 4

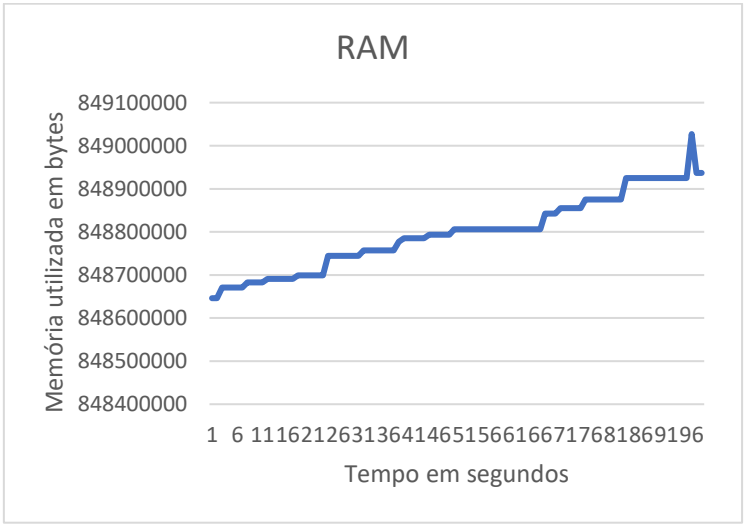

Figura A-108 - Memória em bytes que está a ser utilizada no node 4 para a repetição 2 do teste 4

Node 5

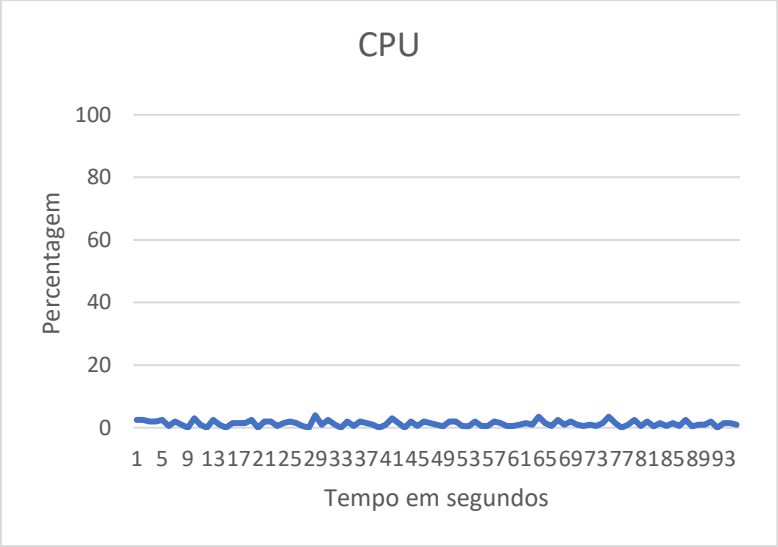

Figura A-109 - Valores de CPU do node 5 para a repetição 2 do teste 4

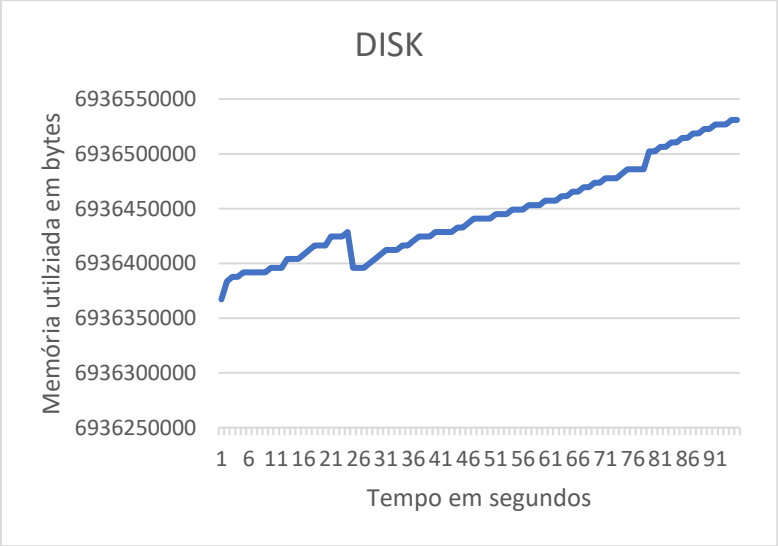

Figura A-110 - Memória utilizada de disco em bytes do node 5 para a repetição 2 do teste 4

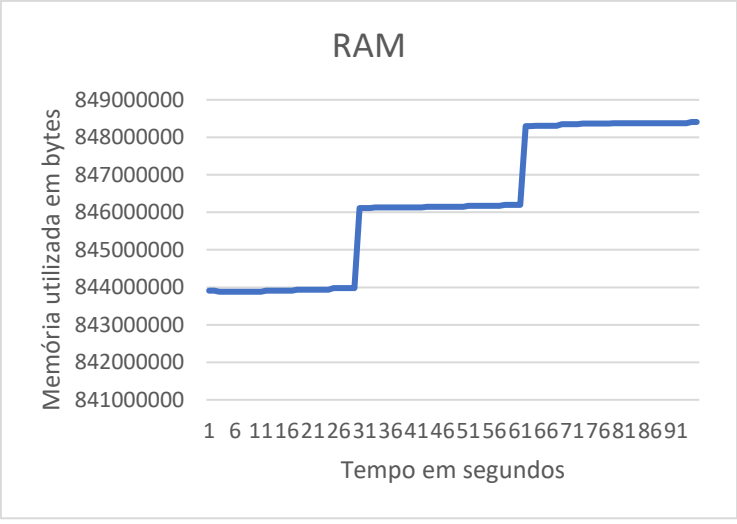

Figura A-111 - Memória em bytes que está a ser utilizada no node 5 para a repetição 2 do teste 4

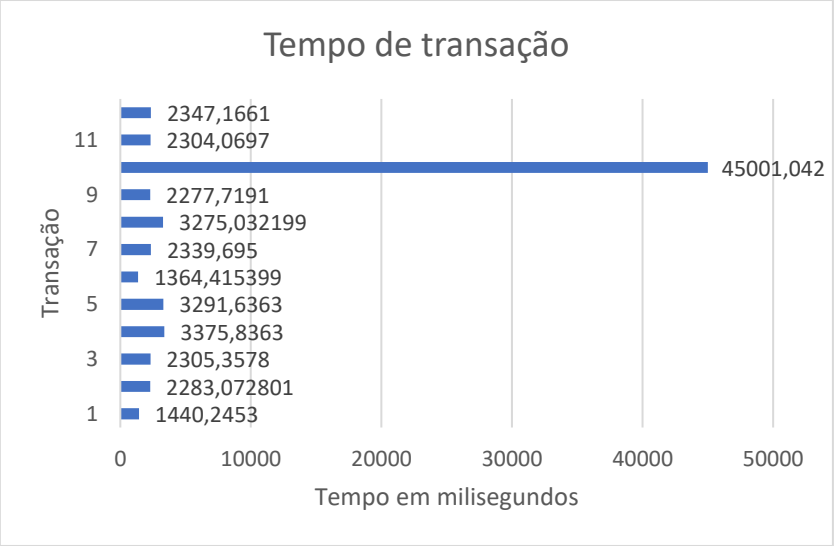

Figura A-112 - Tempo de processamento das transações realizadas na repetição 2 do teste 4

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

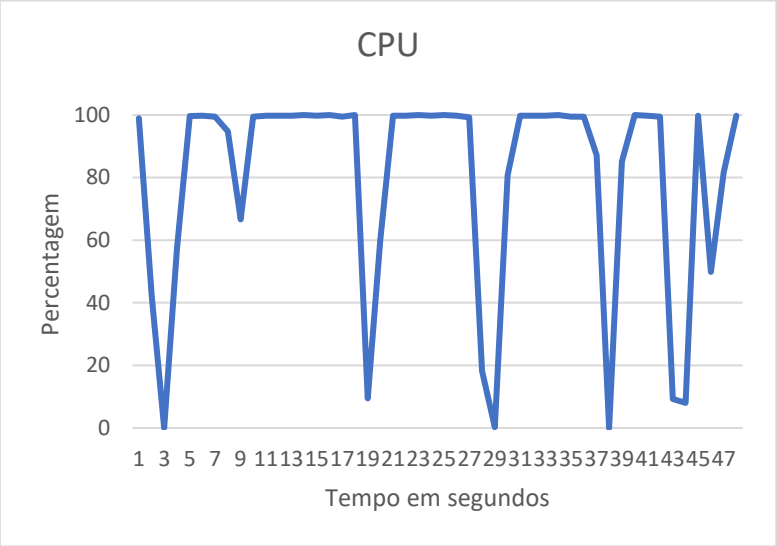

Figura A-113 - Valores de CPU do node 1 para a repetição 3 do teste 4

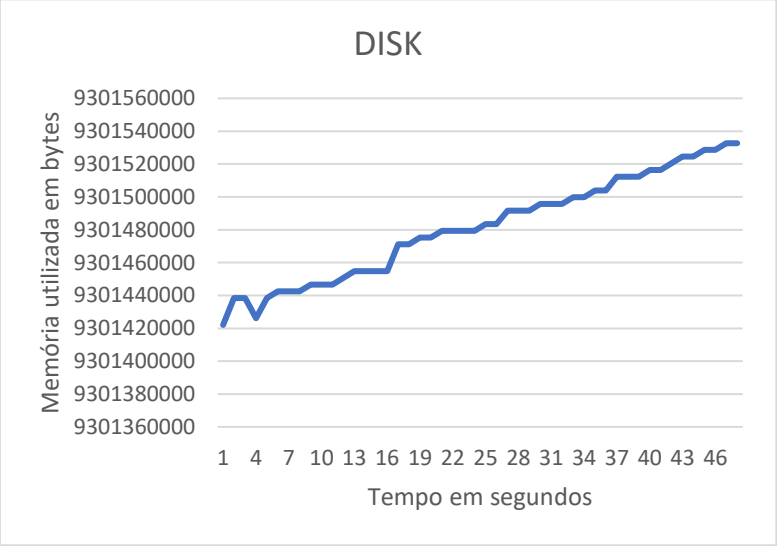

Figura A-114 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 4

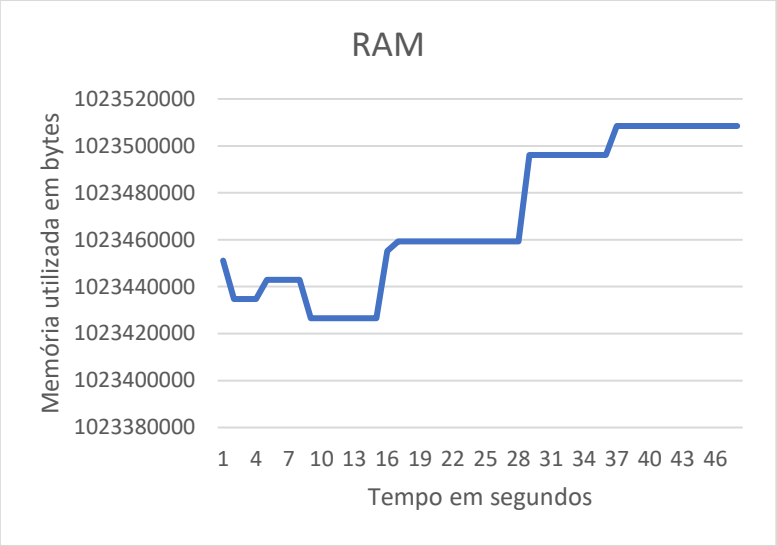

Figura A-115 - Memória em bytes que está a ser utilizada no node 1 para a repetição 3 do teste 4

Node 2 – Miner2

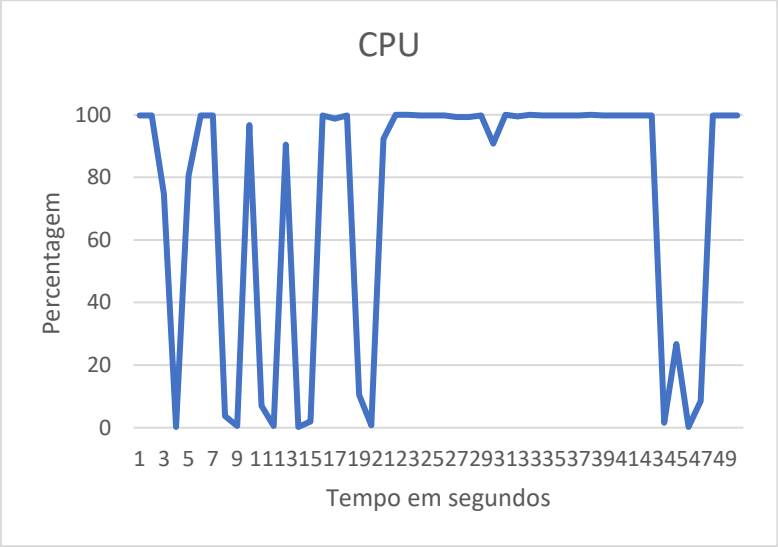

Figura A-116 - Valores de CPU do node 2 para a repetição 3 do teste 4

Figura A-117 - Memória utilizada de disco em bytes do node 2 para a repetição 3 do teste 4

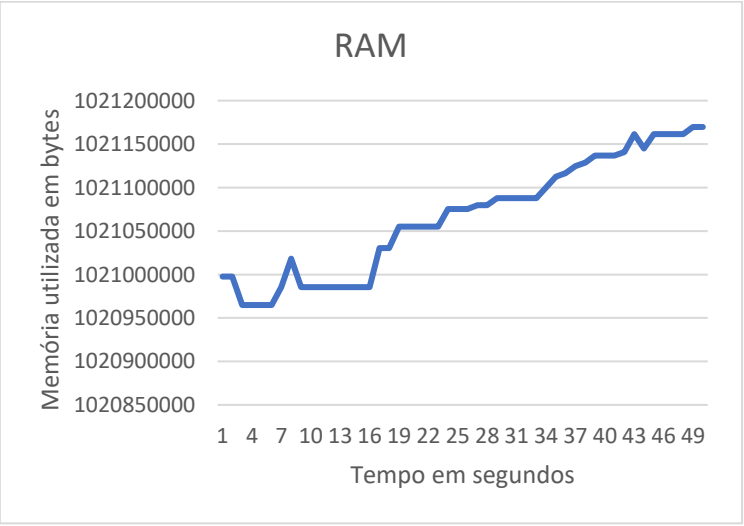

Figura A-118 - Memória em bytes que está a ser utilizada no node 2 para a repetição 3 do teste 4

Node 3 – Miner3

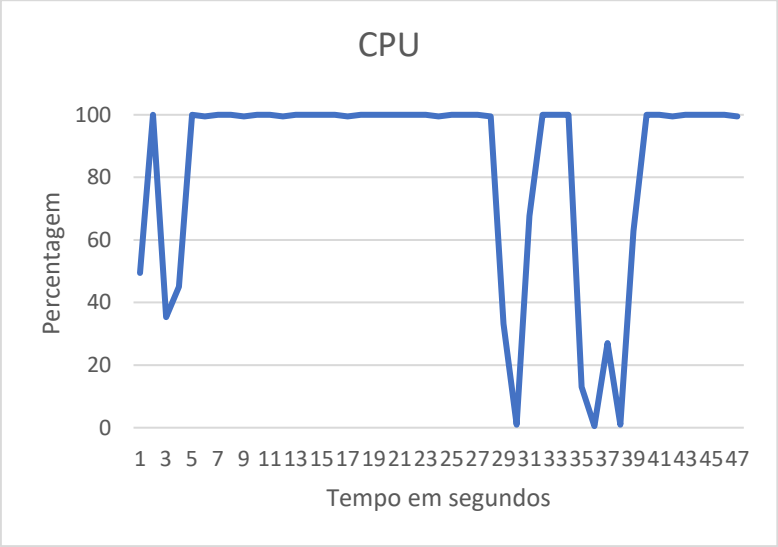

Figura A-119 - Valores de CPU do node 3 para a repetição 3 do teste 4

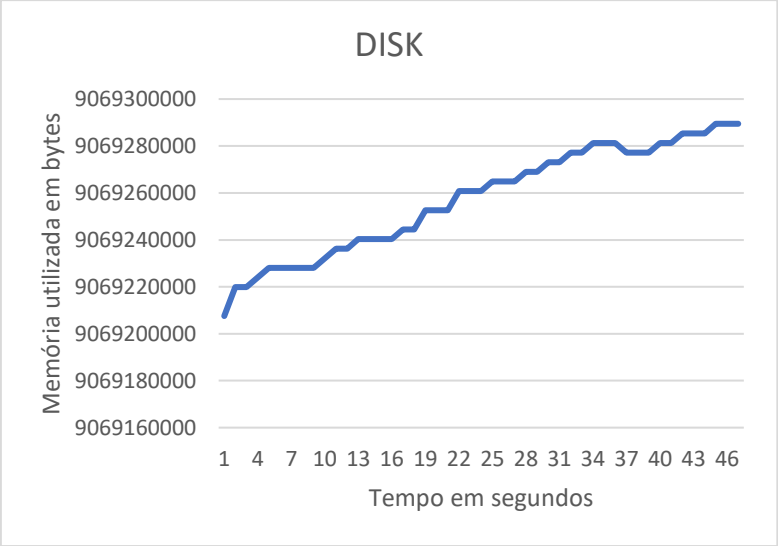

Figura A-120 - Memória utilizada de disco em bytes do node 3 para a repetição 3 do teste 4

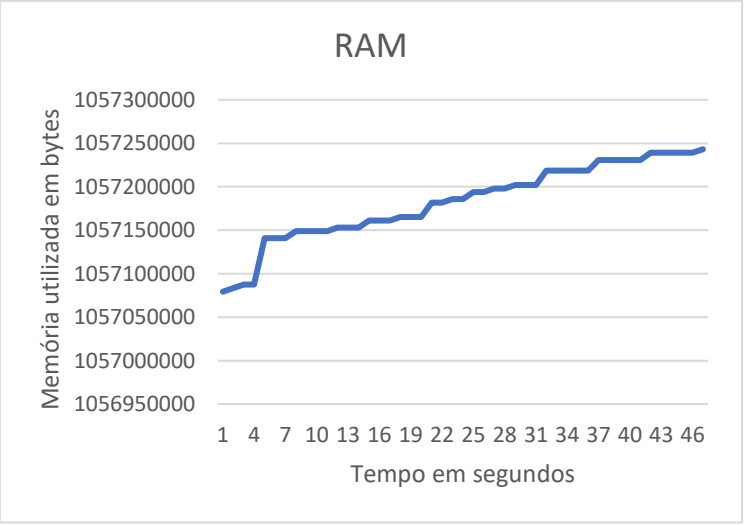

Figura A-121 - Memória em bytes que está a ser utilizada no node 3 para a repetição 3 do teste 4

Node 4

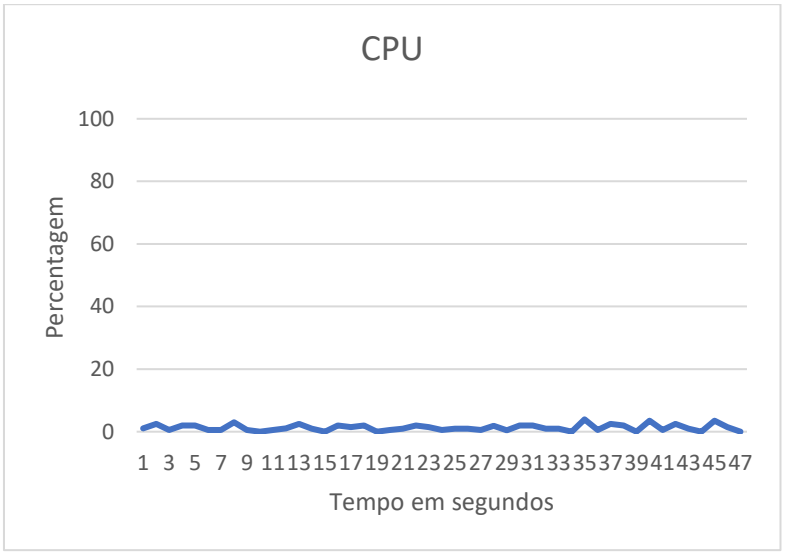

Figura A-122 - Valores de CPU do node 4 para a repetição 3 do teste 4

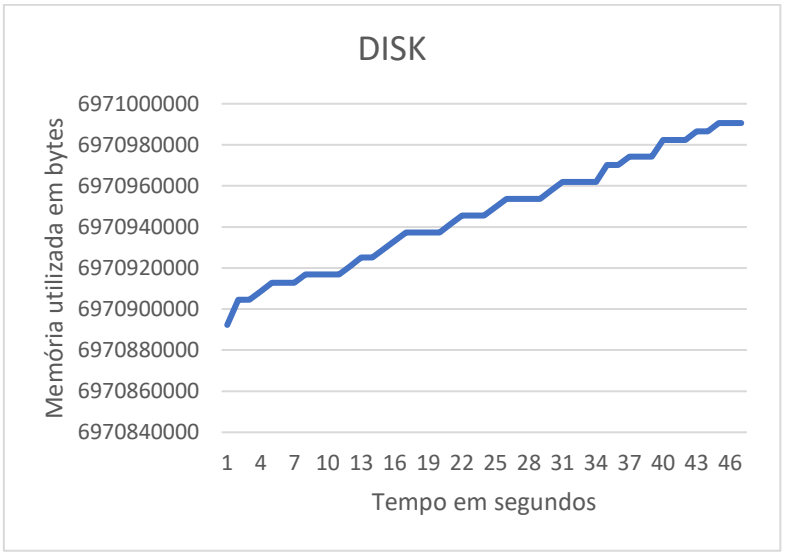

Figura A-123 - Memória utilizada de disco em bytes do node 4 para a repetição 3 do teste 4

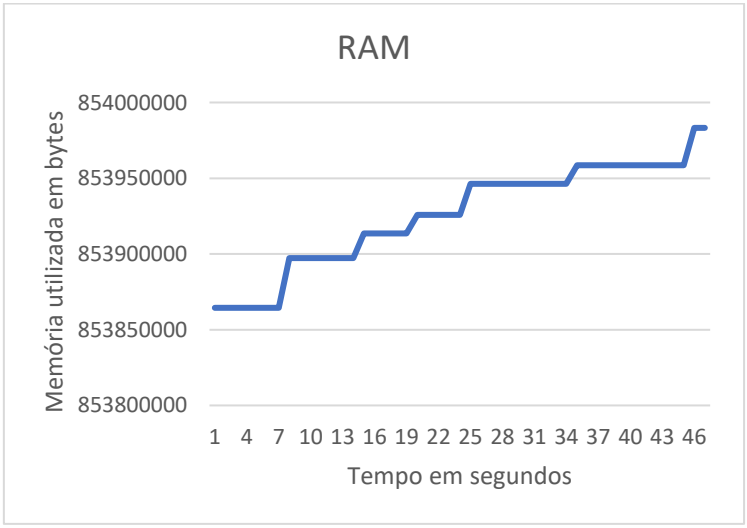

Figura A-124 - Memória em bytes que está a ser utilizada no node 4 para a repetição 3 do teste 4

Node 5

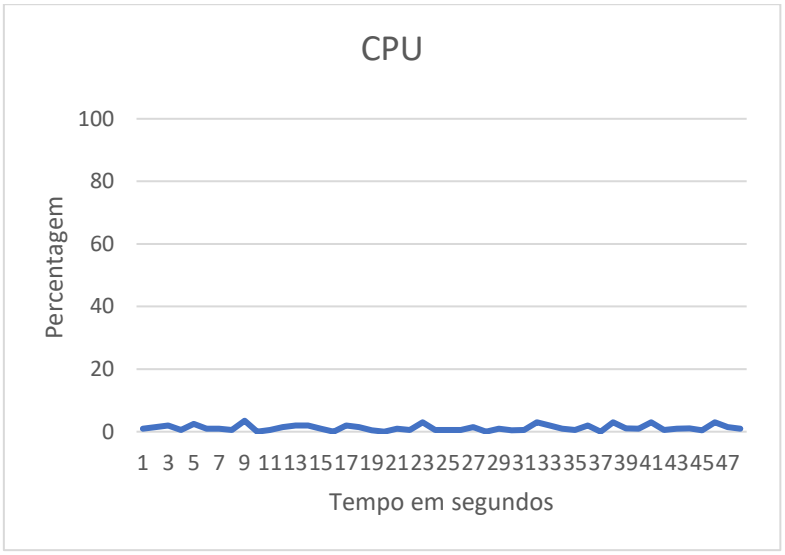

Figura A-125 - Valores de CPU do node 5 para a repetição 3 do teste 4

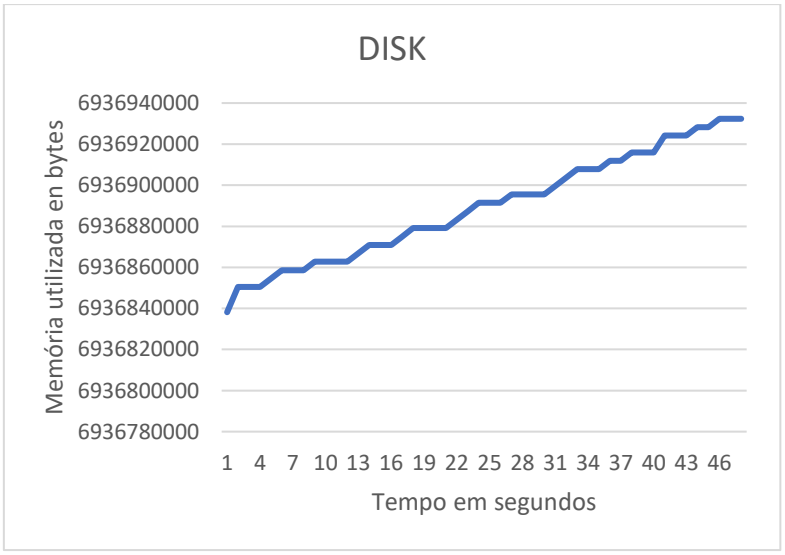

Figura A-126 - Memória utilizada de disco em bytes do node 5 para a repetição 3 do teste 4

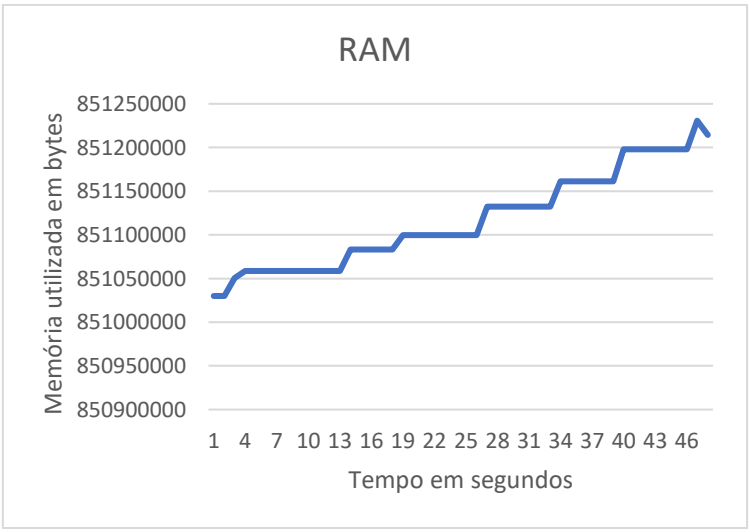

Figura A-127 - Memória em bytes que está a ser utilizada no node 5 para a repetição 3 do teste 4

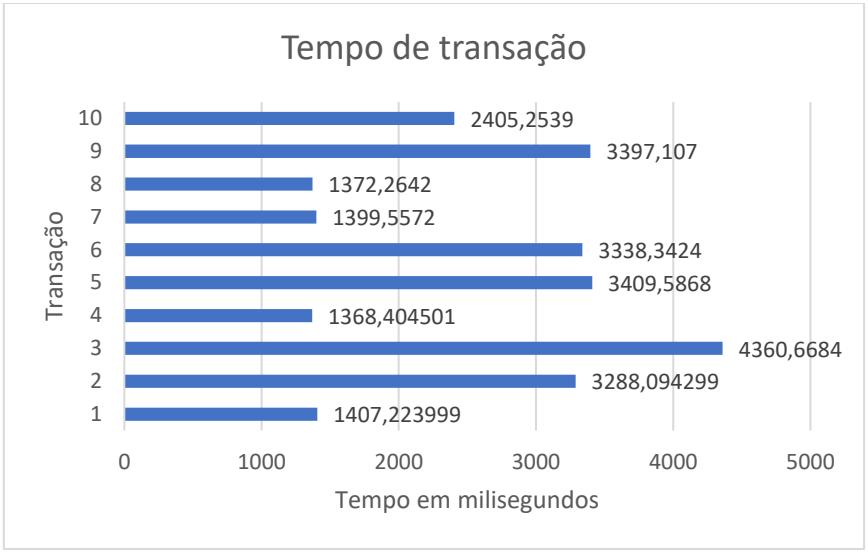

Figura A-128 - Tempo de processamento das transações realizadas na repetição 3 do teste 4

Teste 5

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

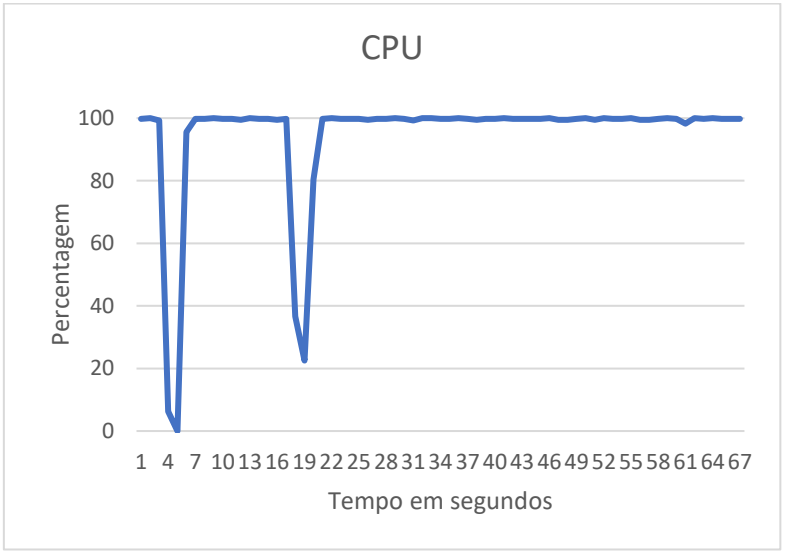

Figura A-129 - Valores de CPU do node 1 para a repetição 1 do teste 5

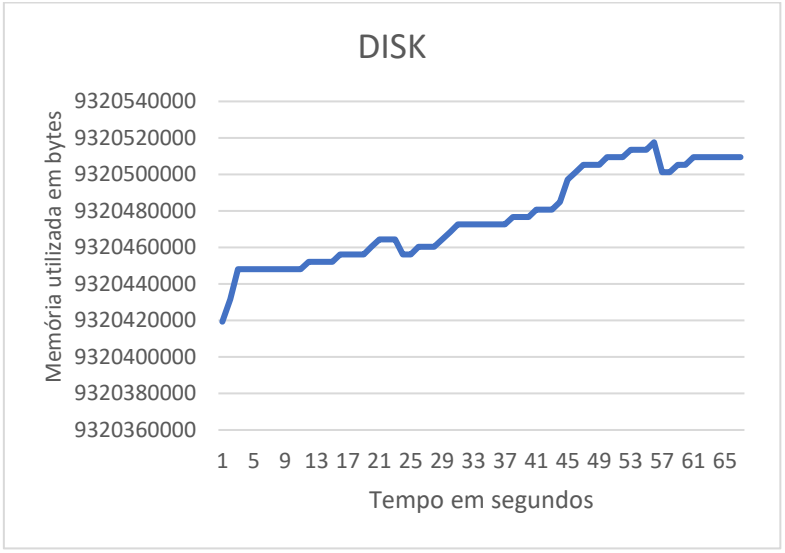

Figura A-130 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 5

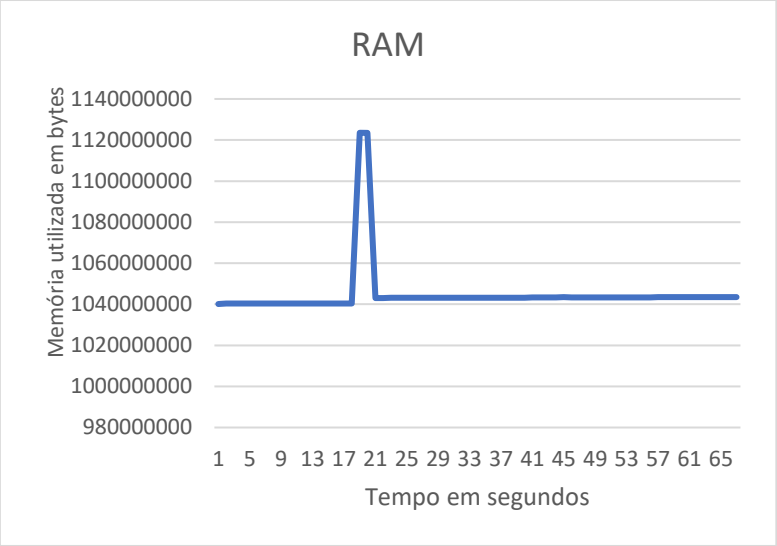

Figura A-131 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 5

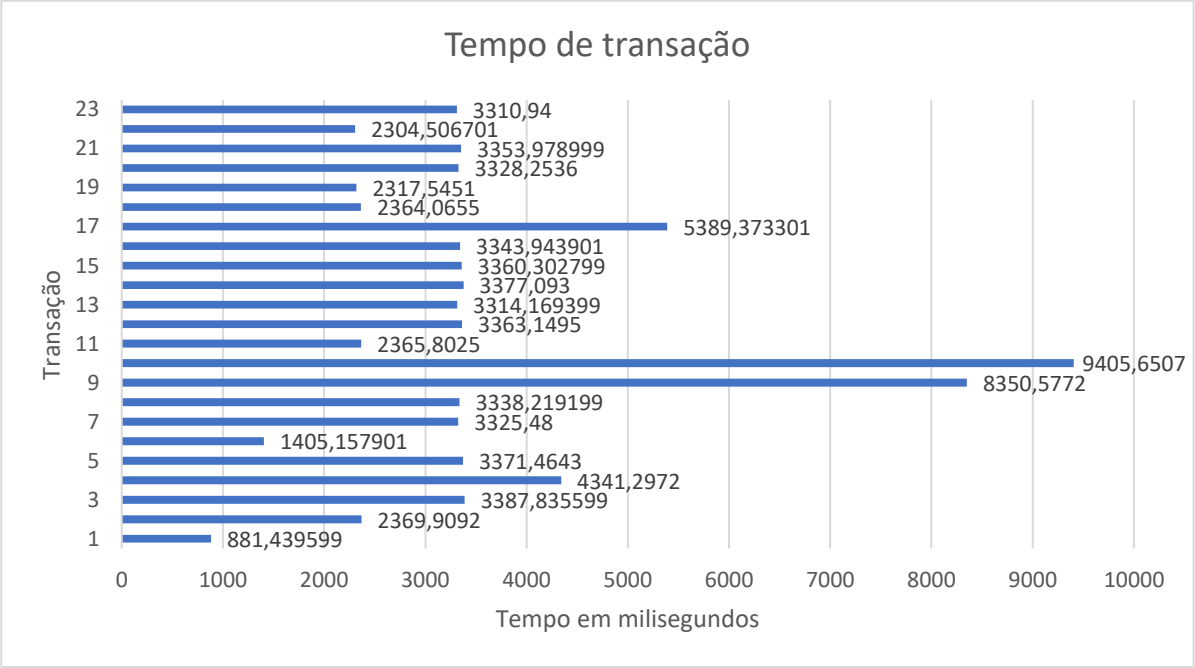

Figura A-132 - Tempo de processamento das transações realizadas na repetição 1 do teste 5

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

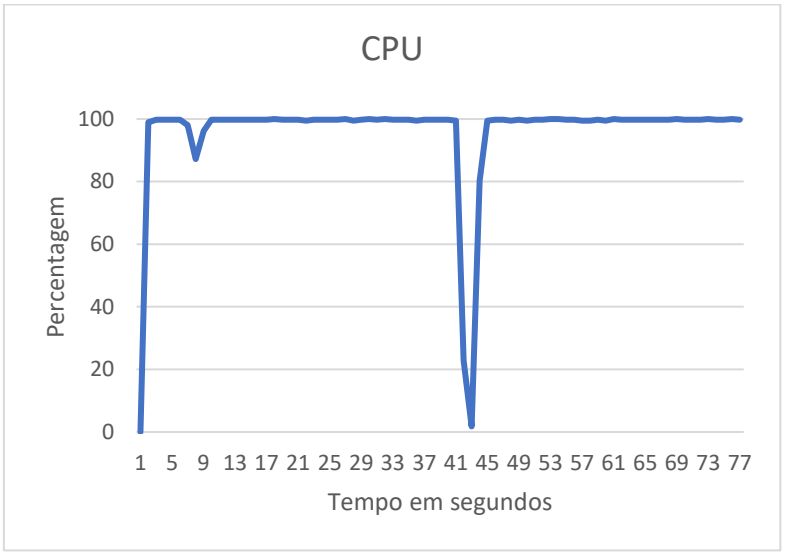

Figura A-133 - Valores de CPU do node 1 para a repetição 2 do teste 5

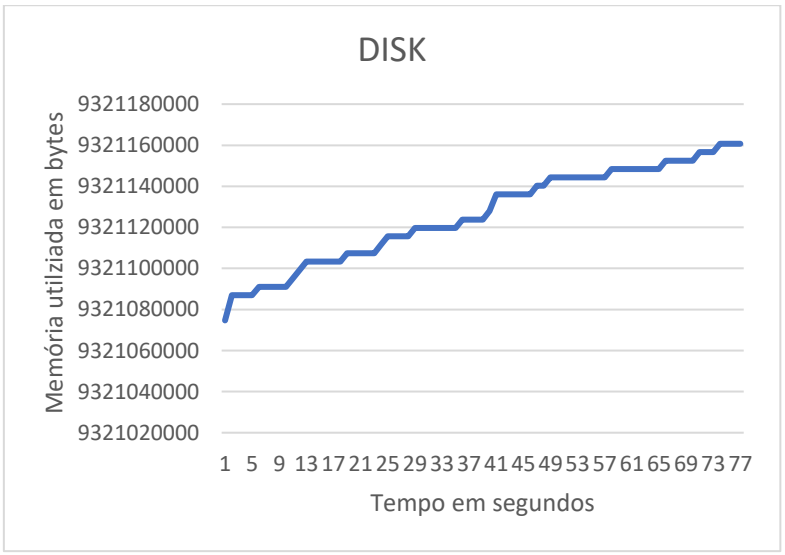

Figura A-134 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 5

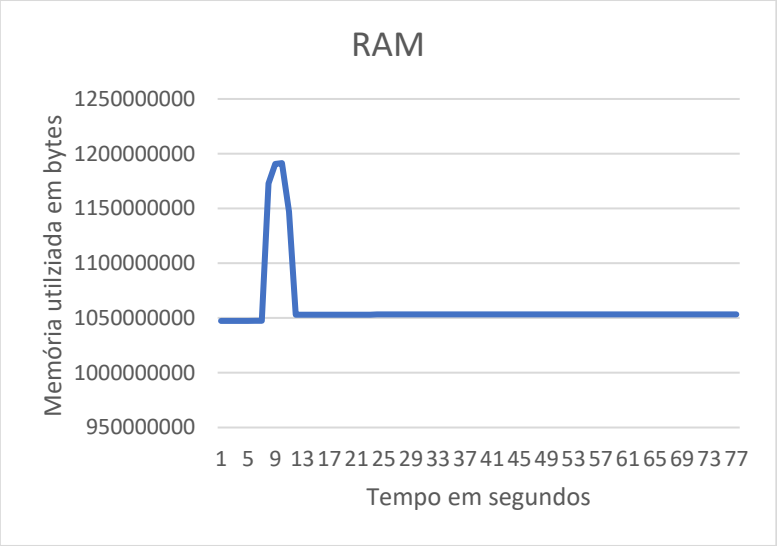

Figura A-135 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 5

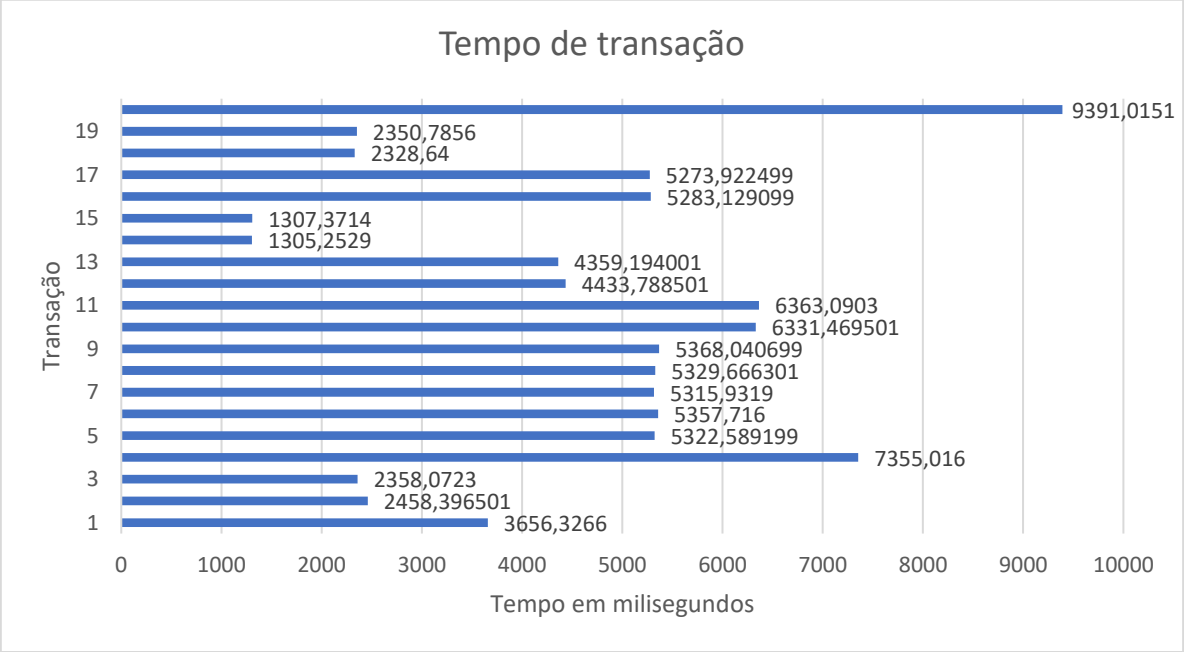

Figura A-136 - Tempo de processamento das transações realizadas na repetição 2 do teste 5

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

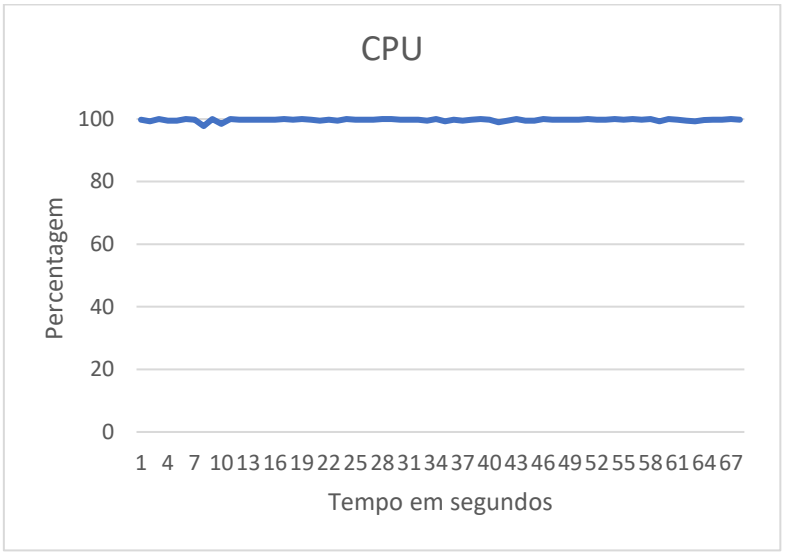

Figura A-137 - Valores de CPU do node 1 para a repetição 3 do teste 5

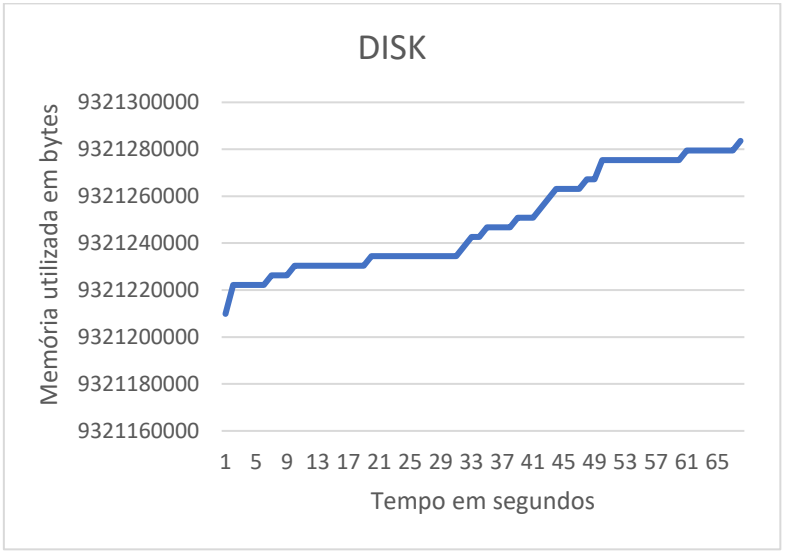

Figura A-138 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 5

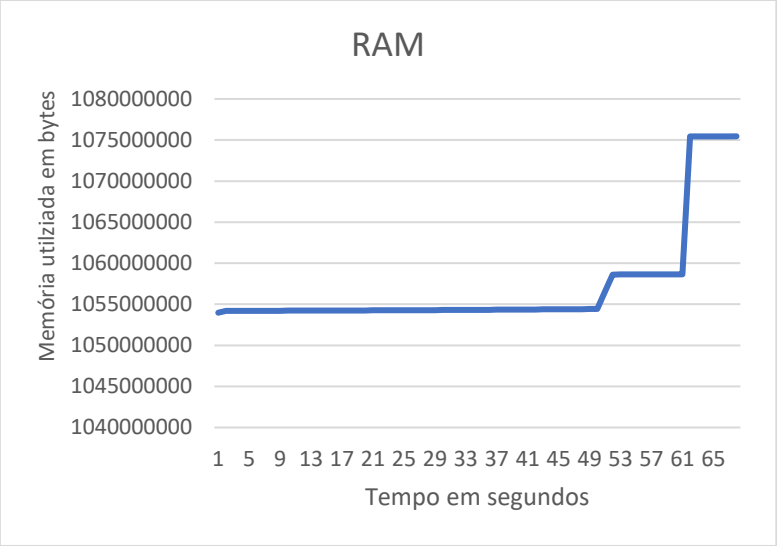

Figura A-139 - Memória em bytes que está a ser utilizada no node 1 para a repetição 3 do teste 5

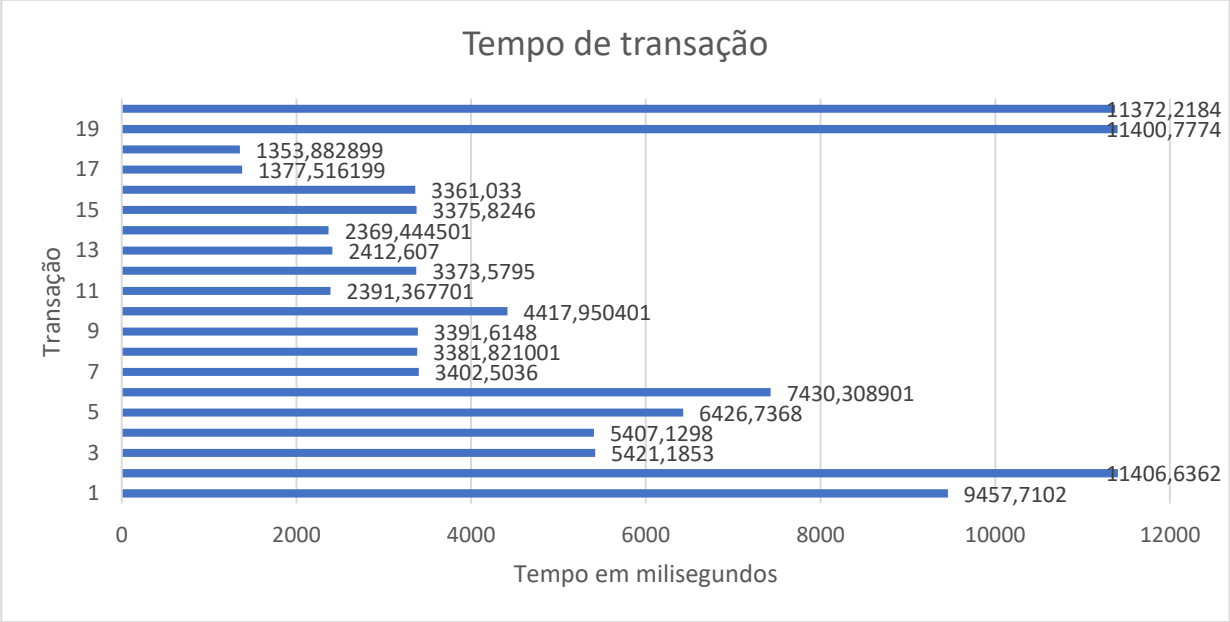

Figura A-140 - Tempo de processamento das transações realizadas na repetição 3 do teste 5

Teste 6

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

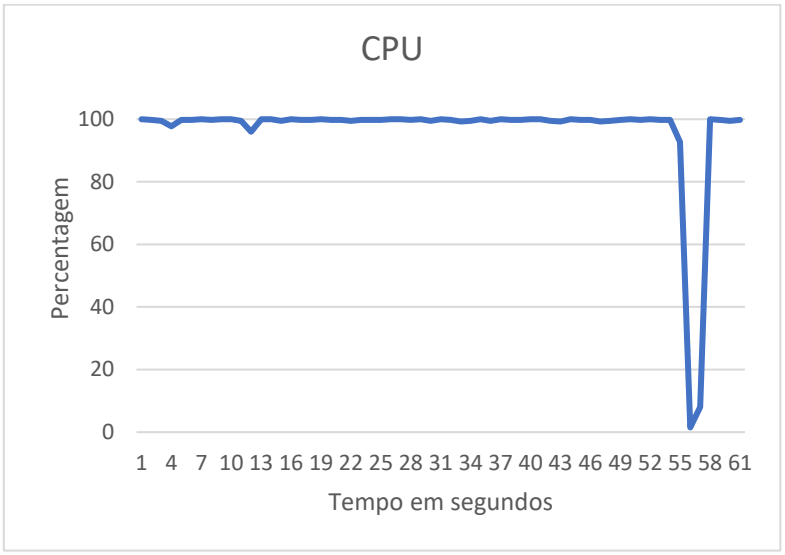

Figura A-141 - Valores de CPU do node 1 para a repetição 1 do teste 6

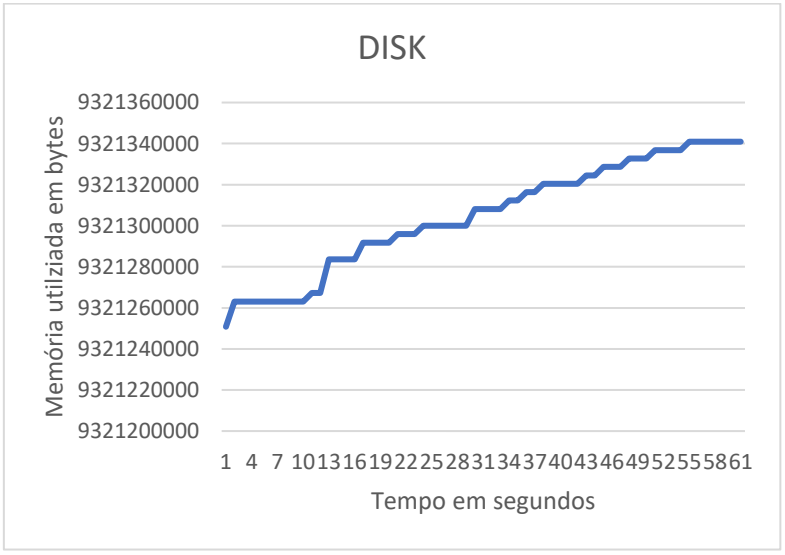

Figura A-142 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 6

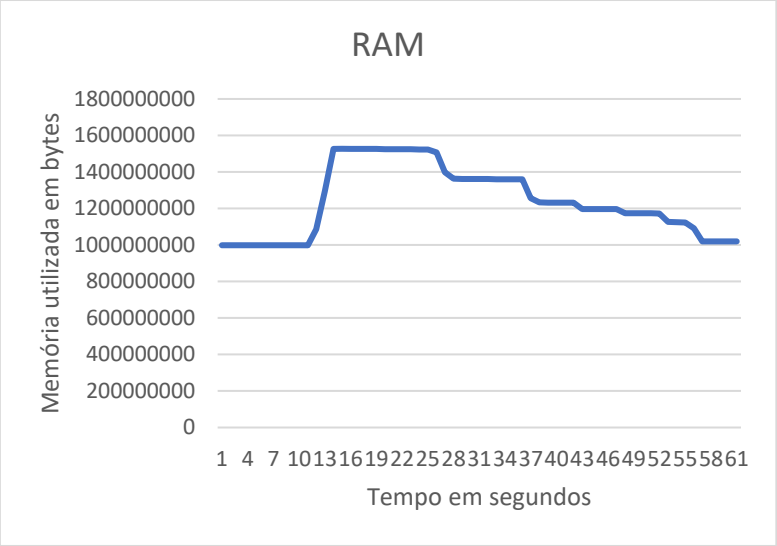

Figura A-143 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 6

Node 2 – Miner2

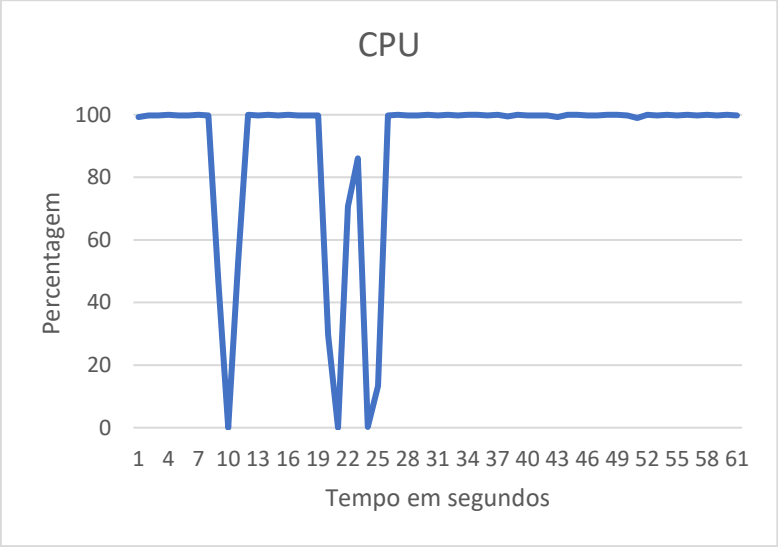

Figura A-144 - Valores de CPU do node 2 para a repetição 1 do teste 6

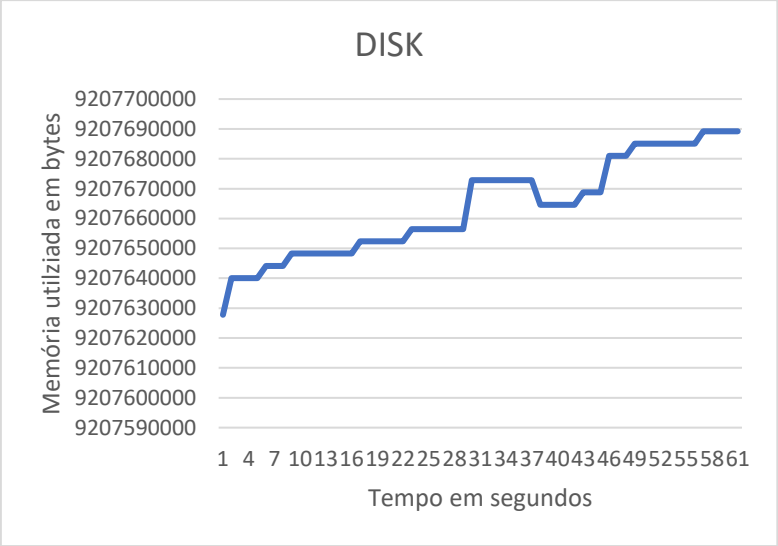

Figura A-145 - Memória utilizada de disco em bytes do node 2 para a repetição 1 do teste 6

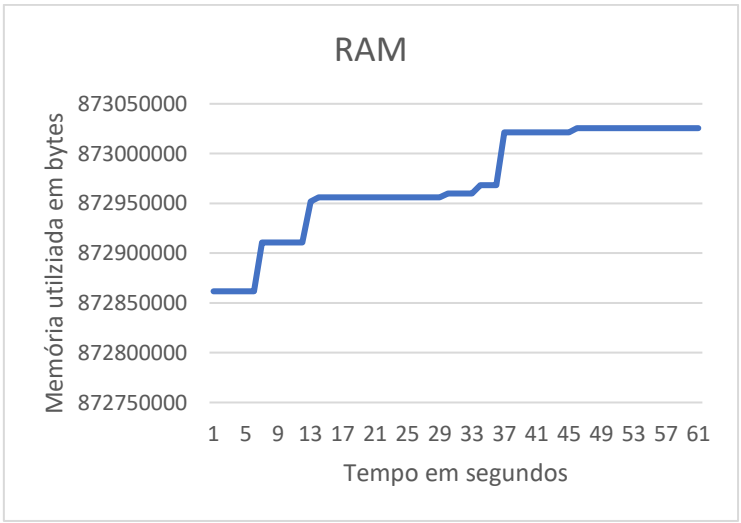

Figura A-146 - Memória em bytes que está a ser utilizada no node 2 para a repetição 1 do teste 6

Node 3

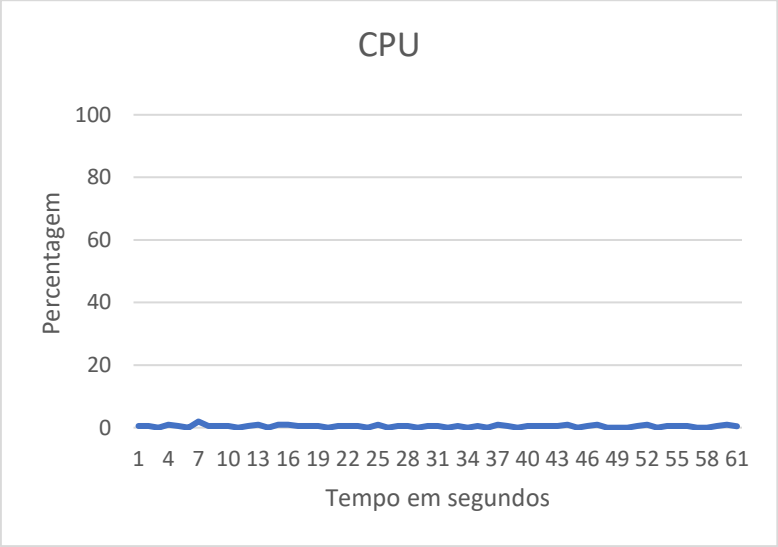

Figura A-147 - Valores de CPU do node 3 para a repetição 1 do teste 6

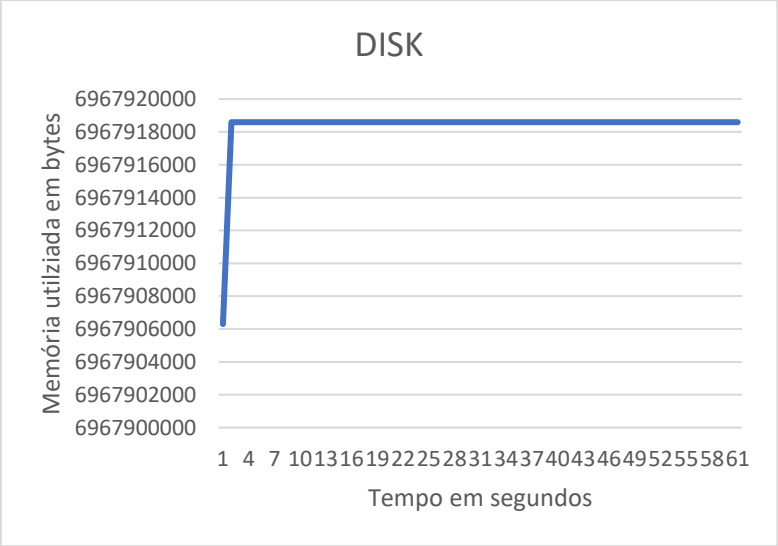

Figura A-148 - Memória utilizada de disco em bytes do node 3 para a repetição 1 do teste 6

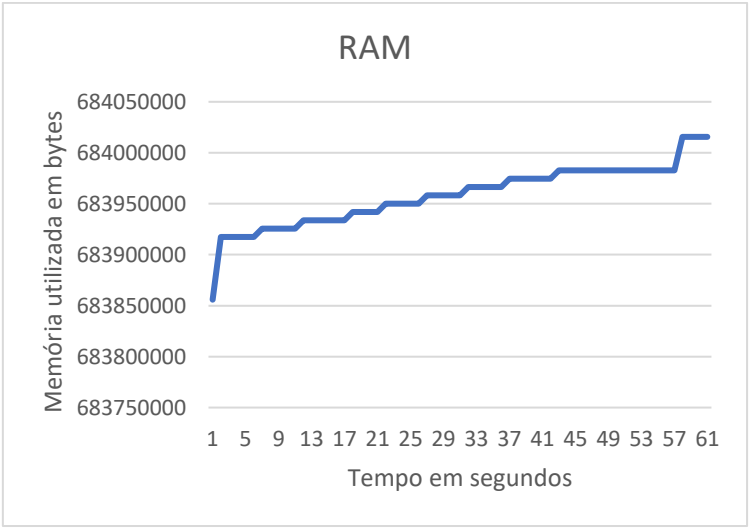

Figura A-149 - Memória em bytes que está a ser utilizada no node 3 para a repetição 1 do teste 6

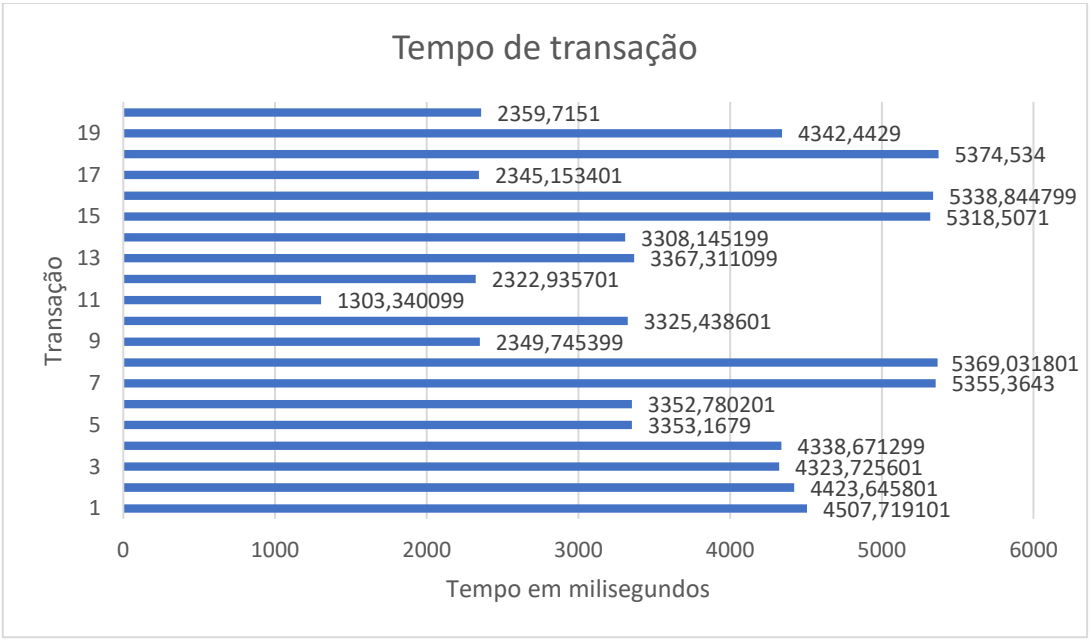

Figura A-150 - Tempo de processamento das transações realizadas na repetição 1 do teste 6

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

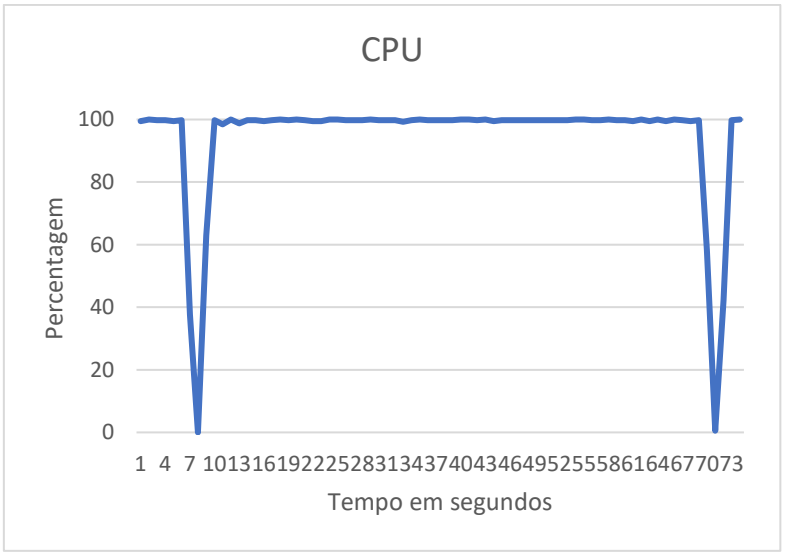

Figura A-151 - Valores de CPU do node 1 para a repetição 2 do teste 6

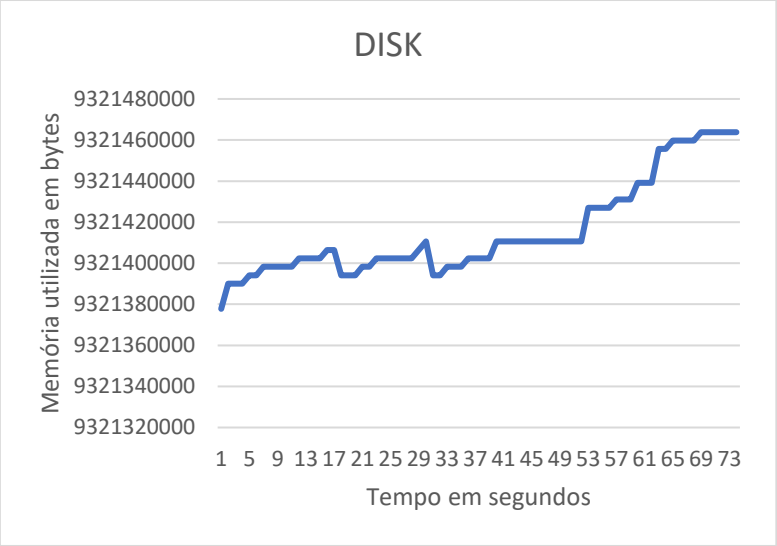

Figura A-152 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 6

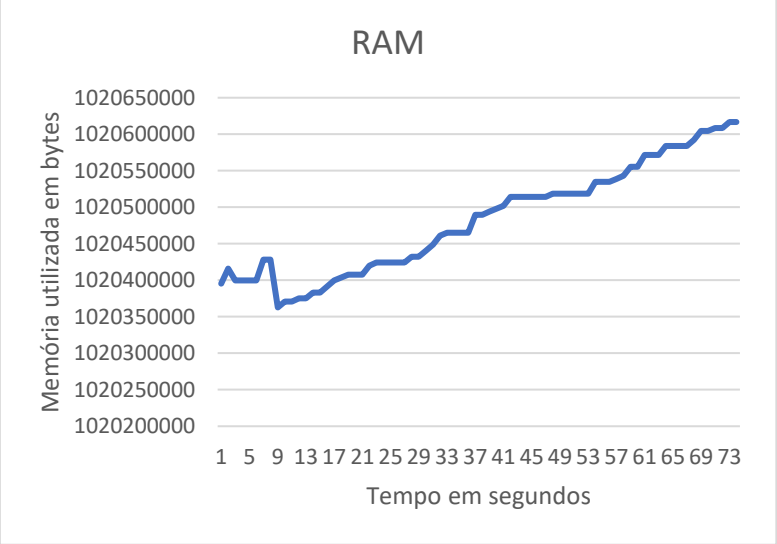

Figura A-153 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 6

Node 2 – Miner2

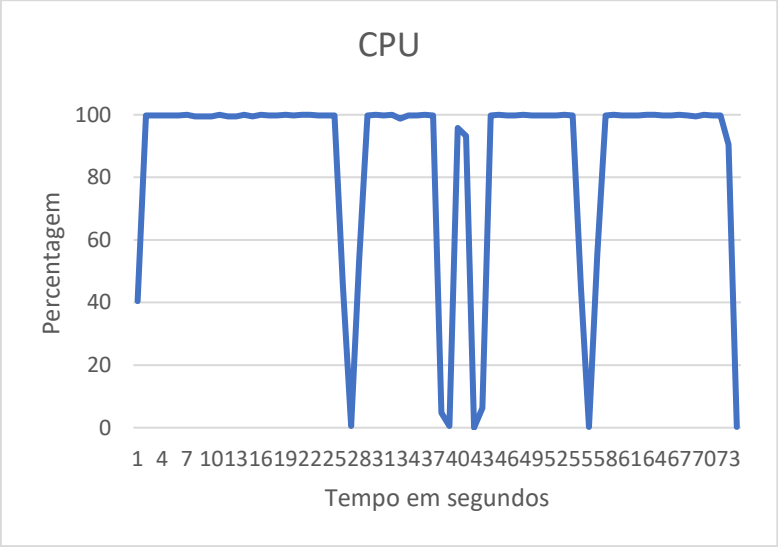

Figura A-154 - Valores de CPU do node 2 para a repetição 2 do teste 6

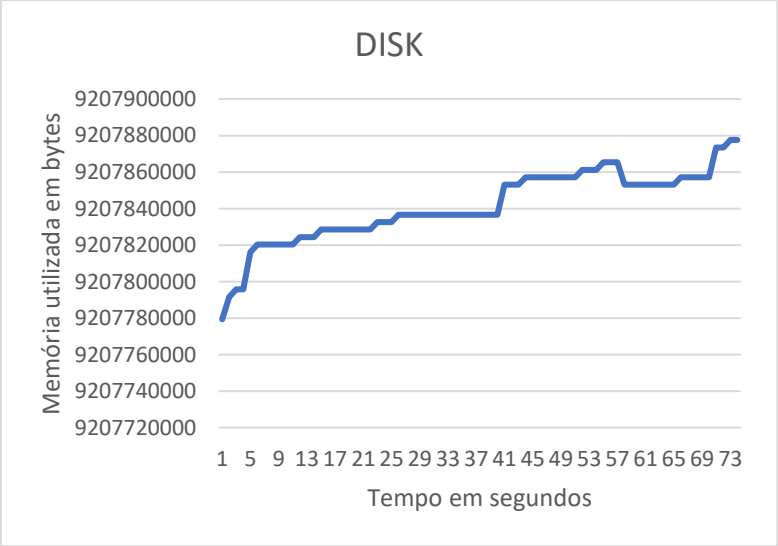

Figura A-155 - Memória utilizada de disco em bytes do node 2 para a repetição 2 do teste 6

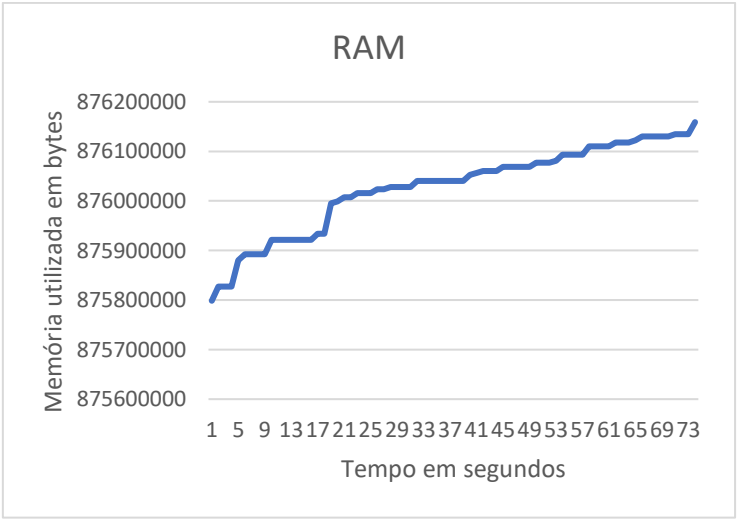

Figura A-156 - Memória em bytes que está a ser utilizada no node 2 para a repetição 2 do teste 6

Node 3

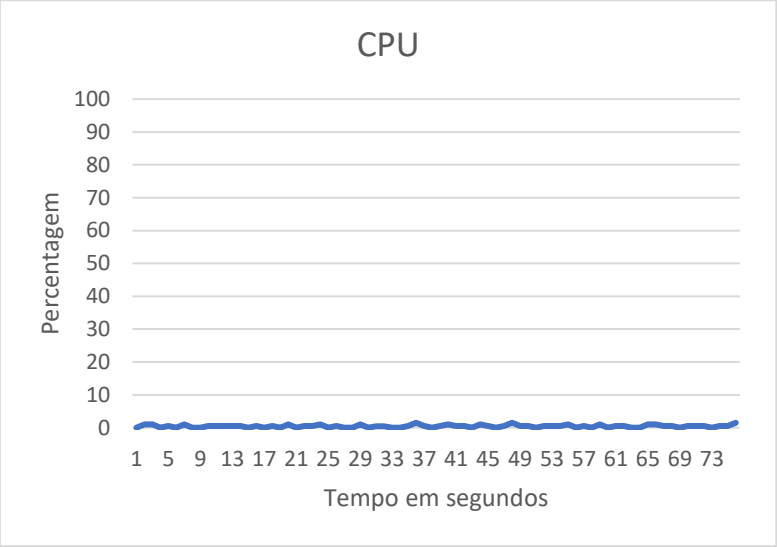

Figura A-157 - Valores de CPU do node 3 para a repetição 2 do teste 6

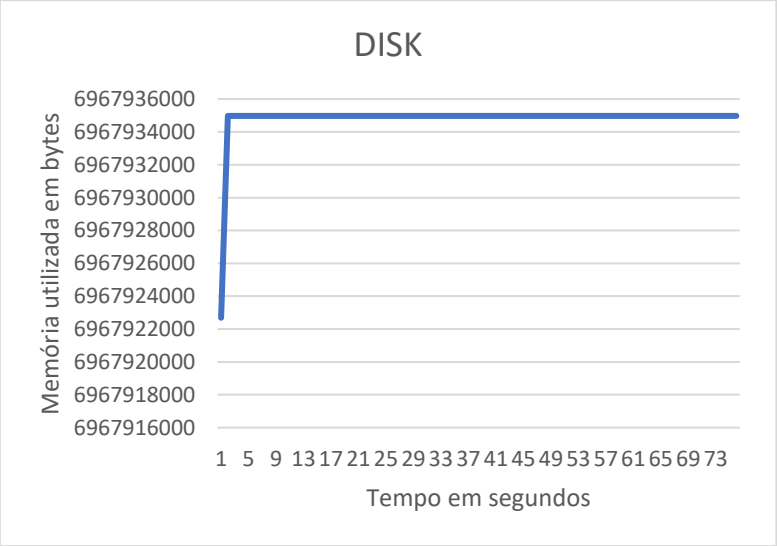

Figura A-158 - Memória utilizada de disco em bytes do node 3 para a repetição 2 do teste 6

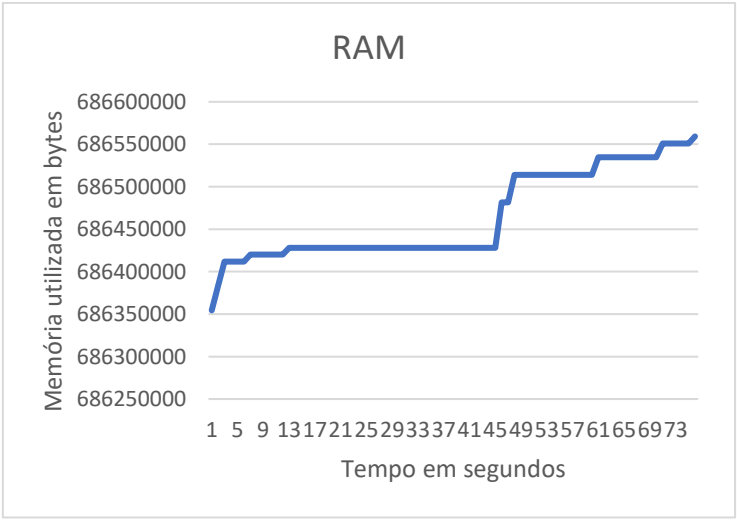

Figura A-159 - Memória em bytes que está a ser utilizada no node 3 para a repetição 2 do teste 6

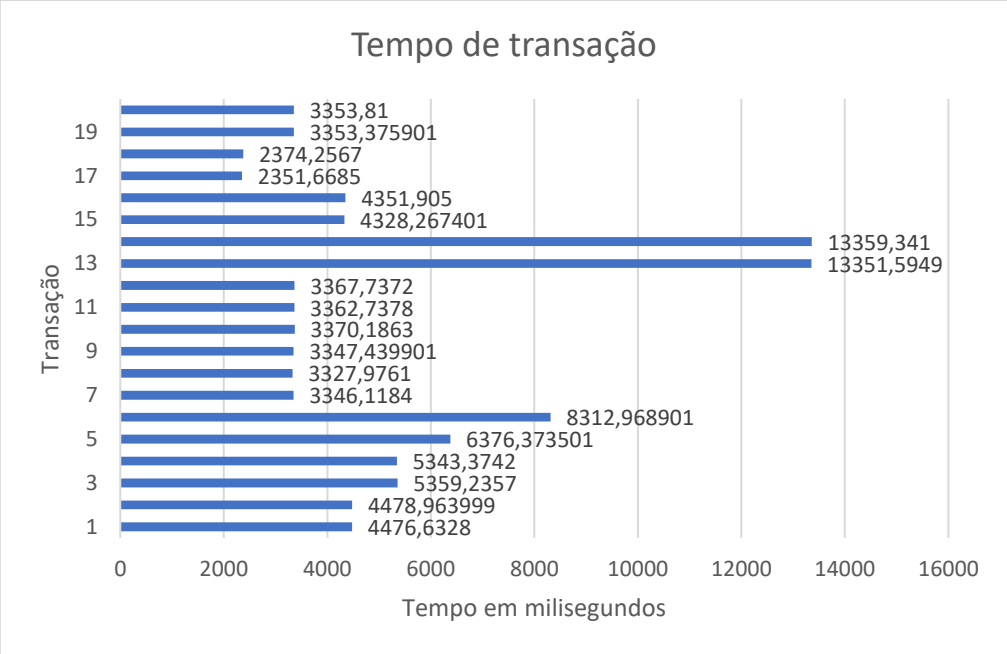

Figura A-160 - Tempo de processamento das transações realizadas na repetição 2 do teste 6

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

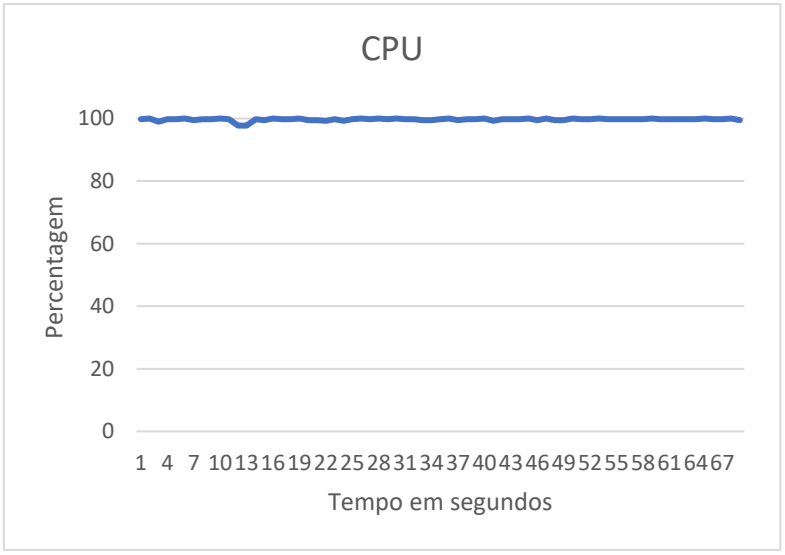

Figura A-161 - Valores de CPU do node 1 para a repetição 3 do teste 6

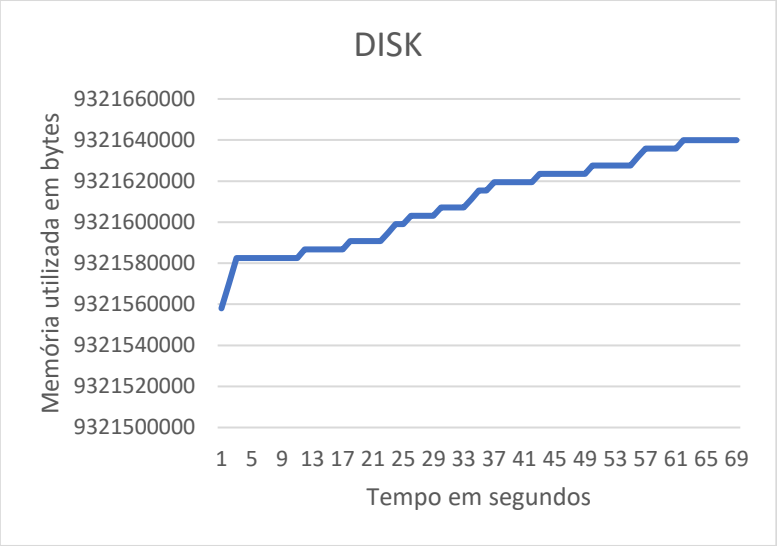

Figura A-162 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 6

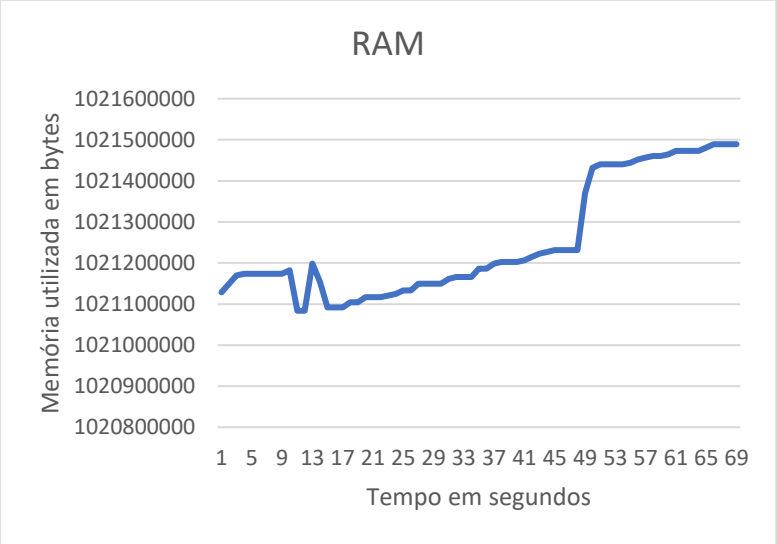

Figura A-163 - Memória em bytes que está a ser utilizada no node 1 para a repetição 3 do teste 6

Node 2 – Miner2

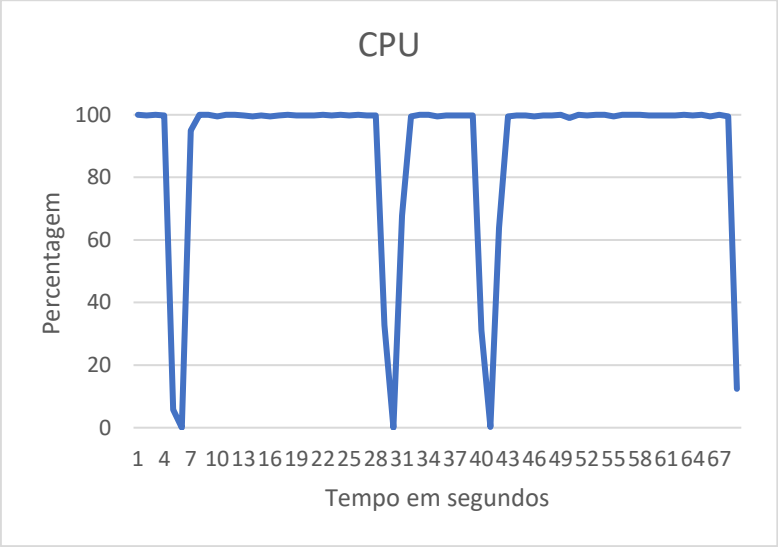

Figura A-164 - Valores de CPU do node 2 para a repetição 3 do teste 6

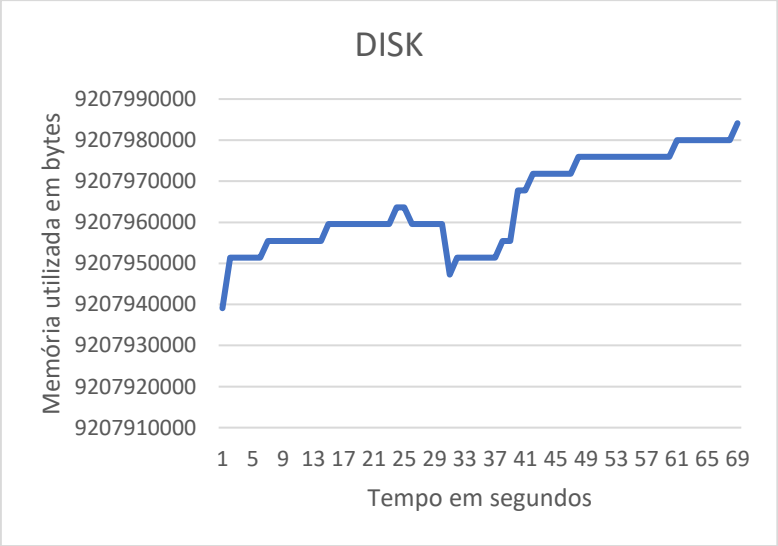

Figura A-165 - Memória utilizada de disco em bytes do node 2 para a repetição 3 do teste 6

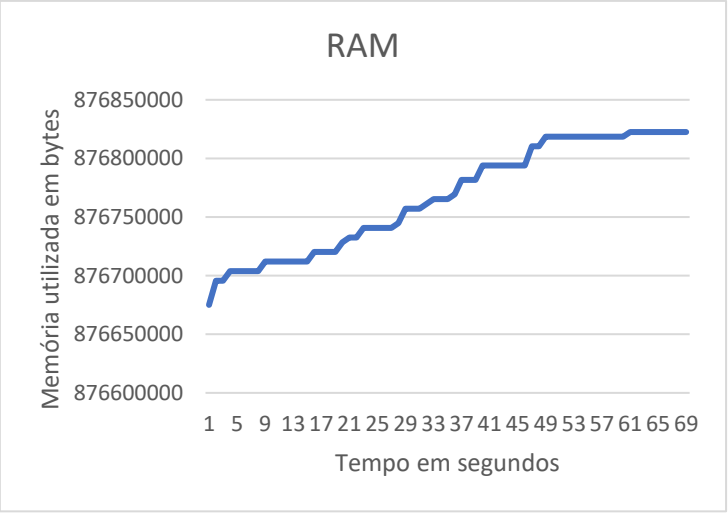

Figura A-166 - Memória em bytes que está a ser utilizada no node 2 para a repetição 3 do teste 6

Node 3

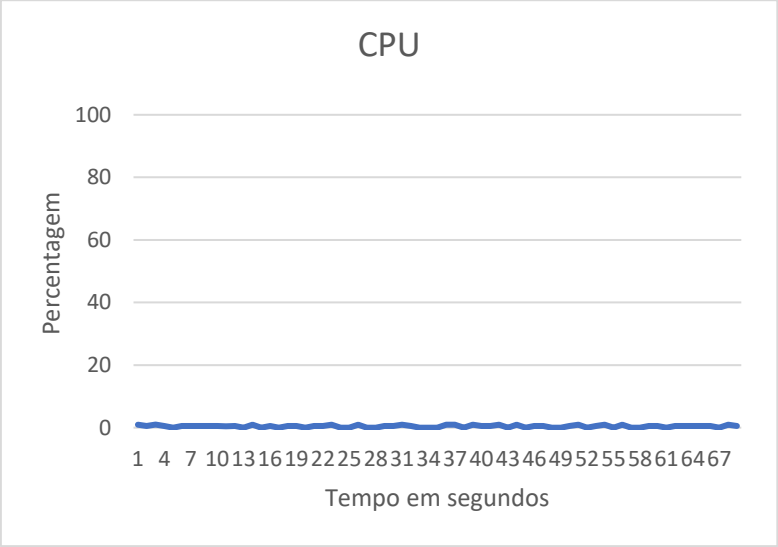

Figura A-167 - Valores de CPU do node 3 para a repetição 3 do teste 6

Figura A-168 - Memória utilizada de disco em bytes do node 3 para a repetição 3 do teste 6

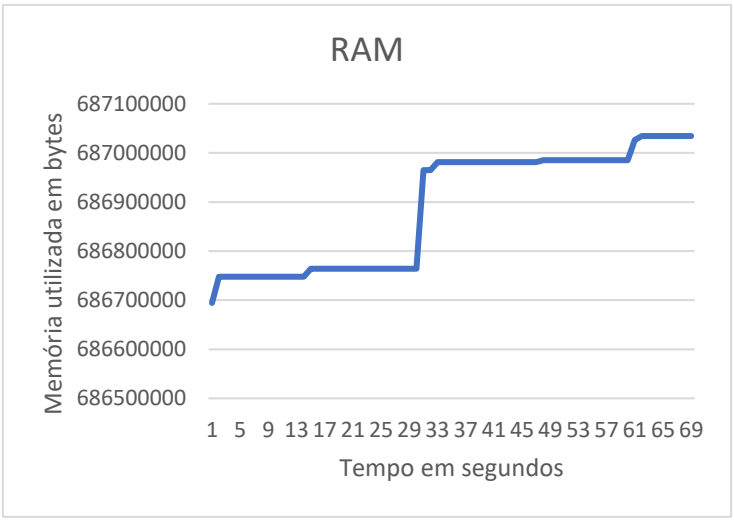

Figura A-169 - Memória em bytes que está a ser utilizada no node 3 para a repetição 3 do teste 6

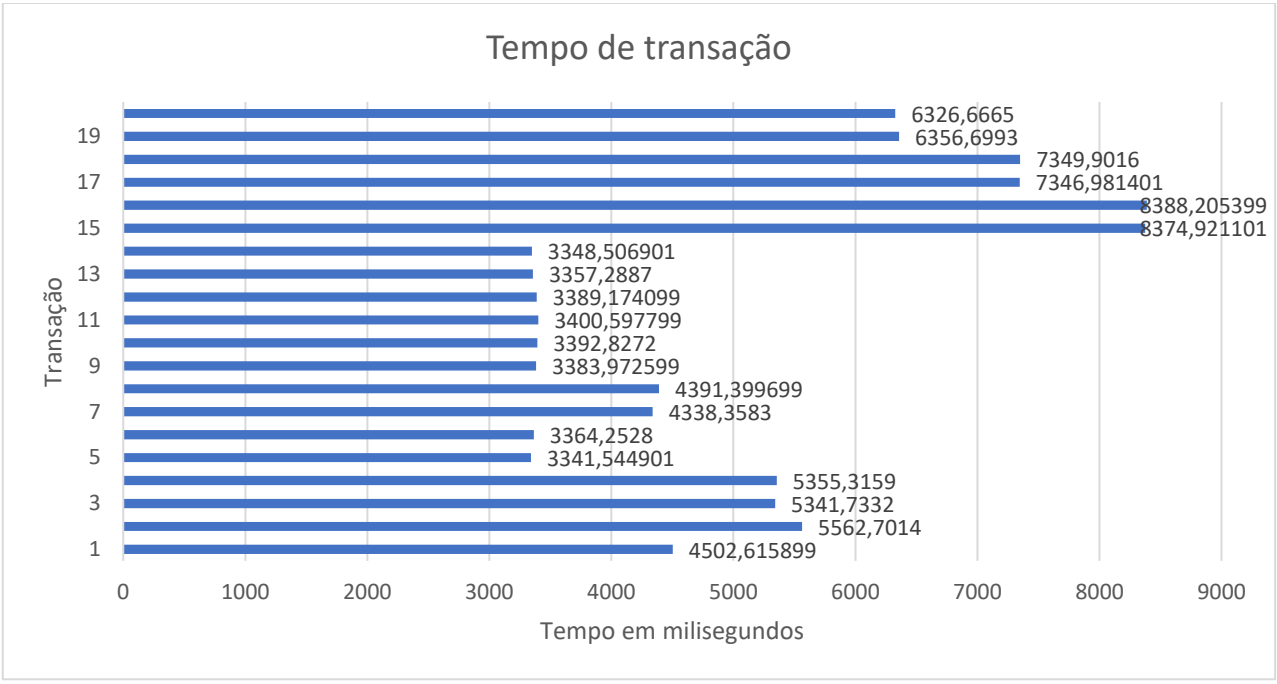

Figura A-170 - Tempo de processamento das transações realizadas na repetição 3 do teste 6

Teste 7

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

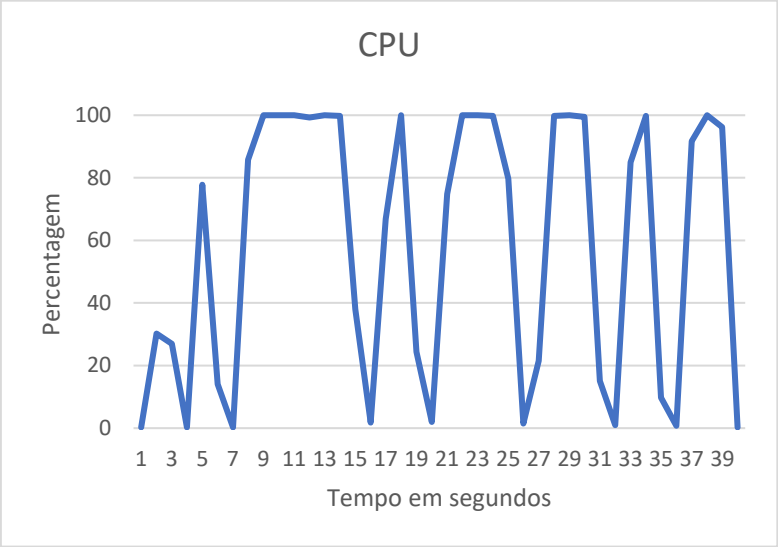

Figura A-171 - Valores de CPU do node 1 para a repetição 1 do teste 7

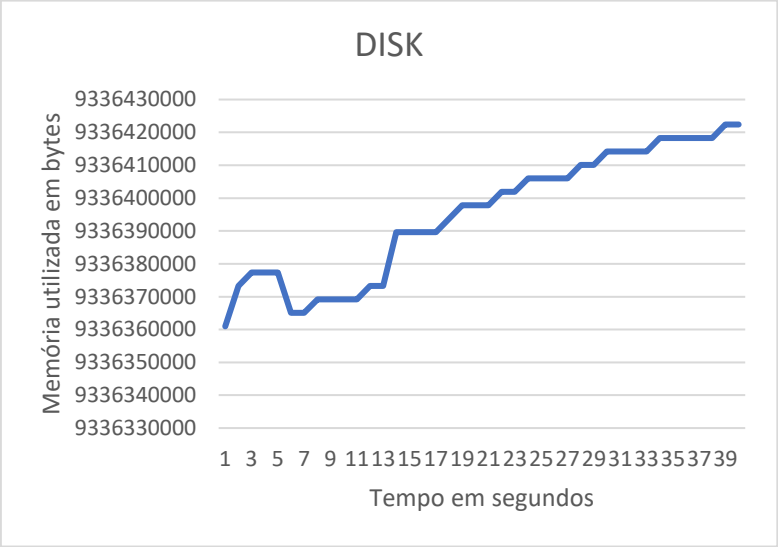

Figura A-172 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 7

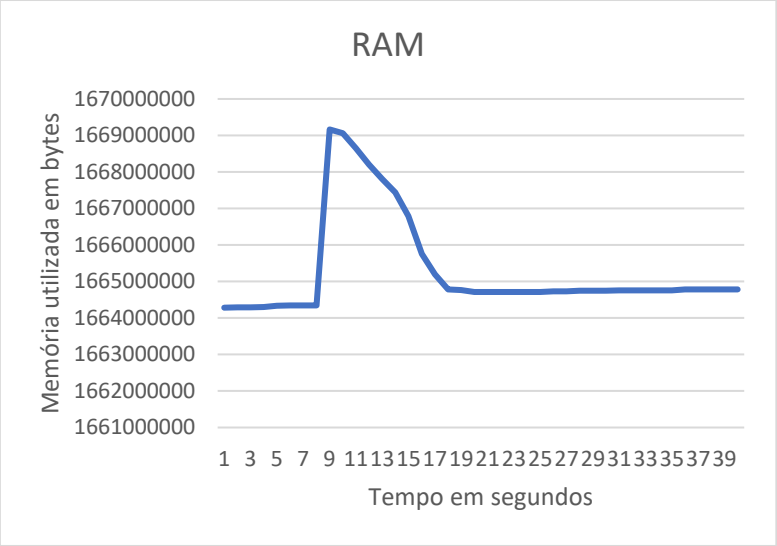

Figura A-173 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 7

Node 2

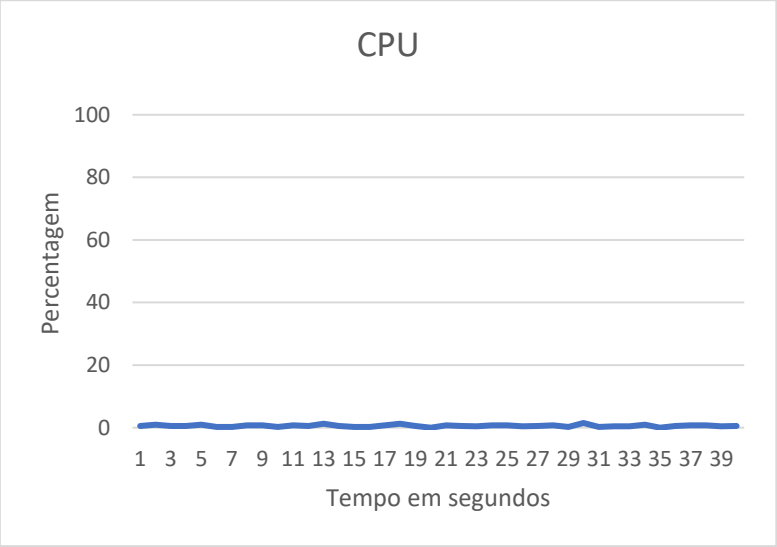

Figura A-174 - Valores de CPU do node 2 para a repetição 1 do teste 7

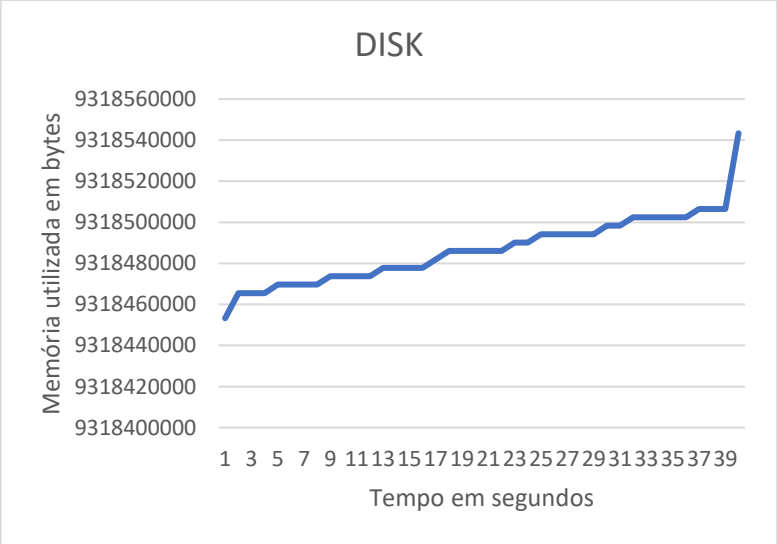

Figura A-175 - Memória utilizada de disco em bytes do node 2 para a repetição 1 do teste 7

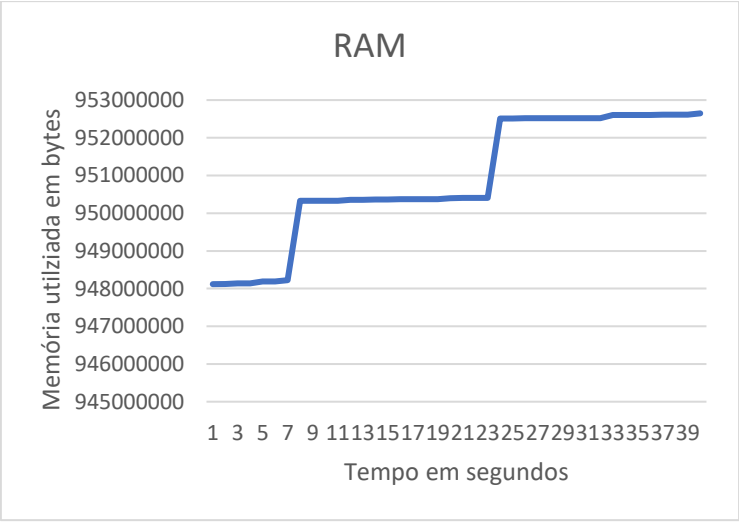

Figura A-176 - Memória em bytes que está a ser utilizada no node 2 para a repetição 1 do teste 7

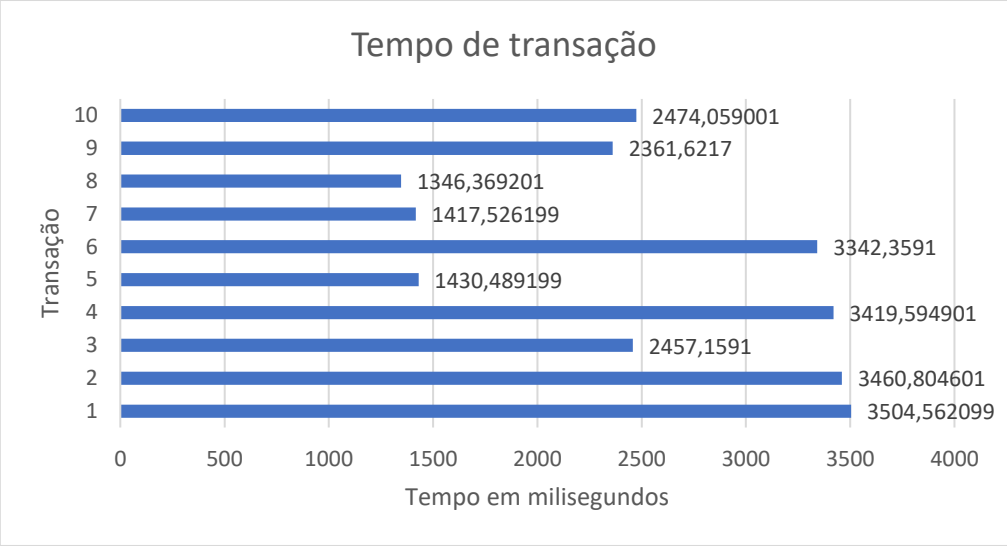

Figura A-177 - Tempo de processamento das transações realizadas na repetição 1 do teste 7

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

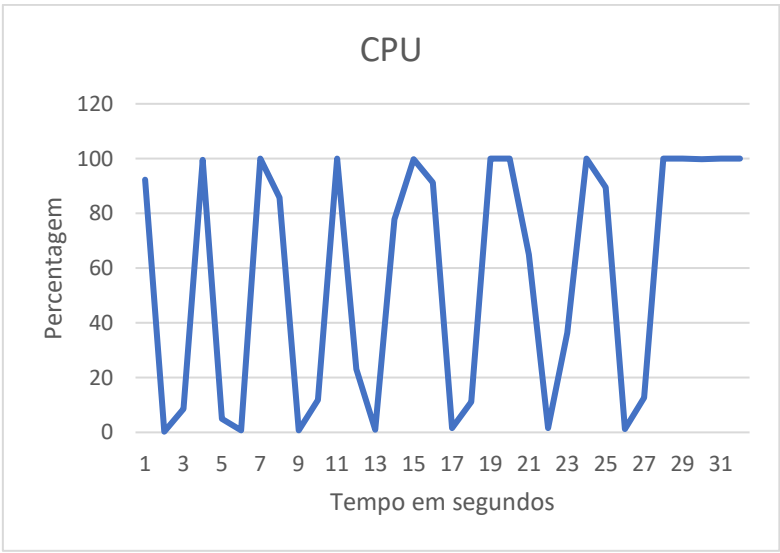

Figura A-178 - Valores de CPU do node 1 para a repetição 2 do teste 7

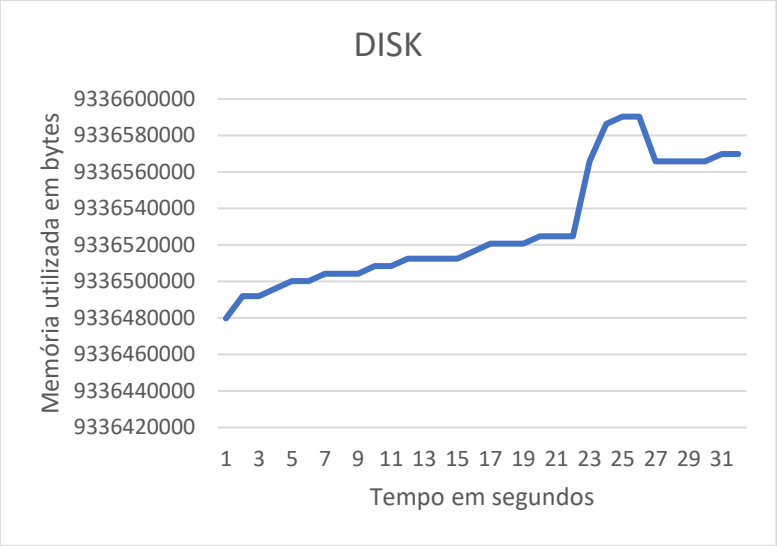

Figura A-179 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 7

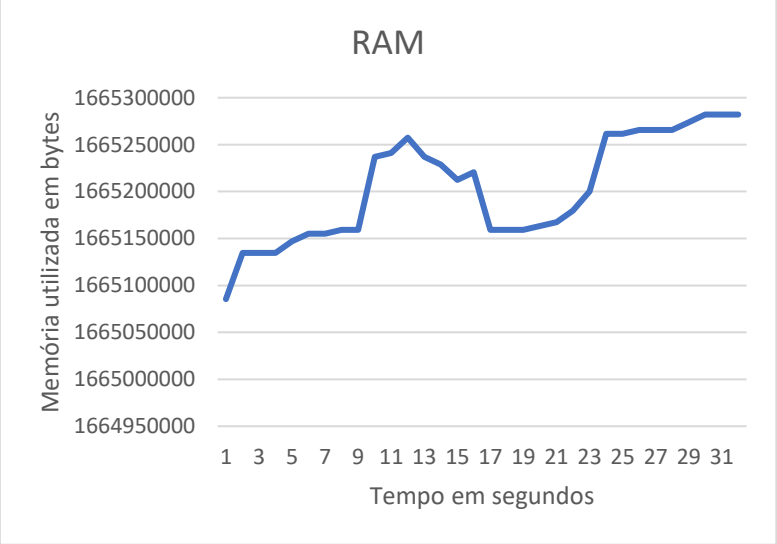

Figura A-180 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 7

Node 2

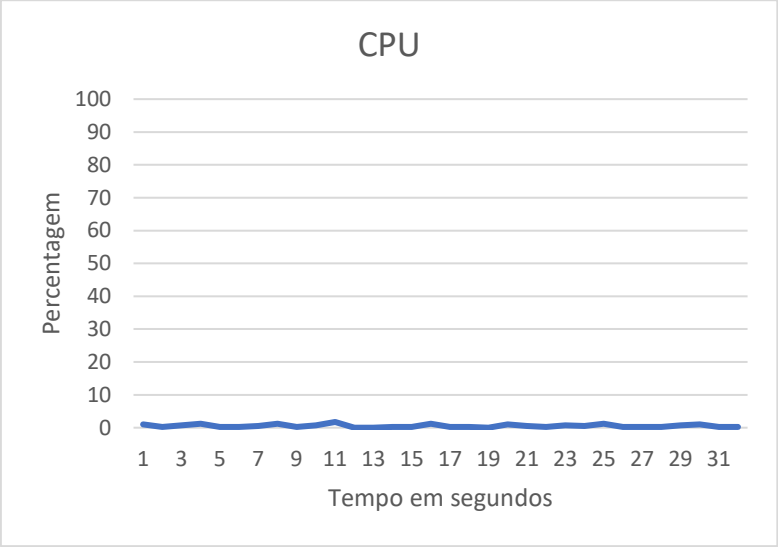

Figura A-181 - Valores de CPU do node 2 para a repetição 2 do teste 7

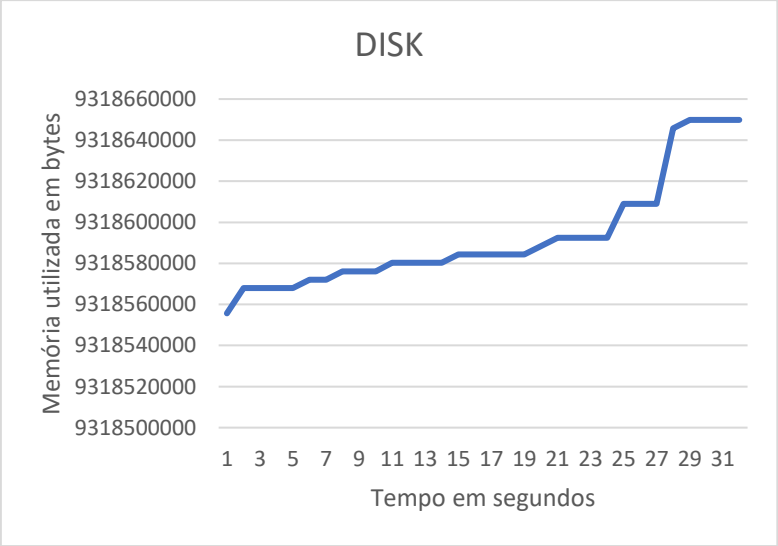

Figura A-182 - Memória utilizada de disco em bytes do node 2 para a repetição 2 do teste 7

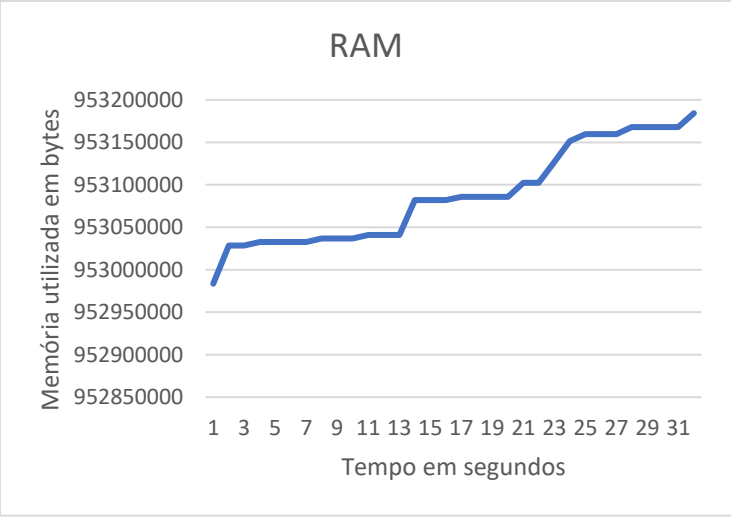

Figura A-183 - Memória em bytes que está a ser utilizada no node 2 para a repetição 2 do teste 7

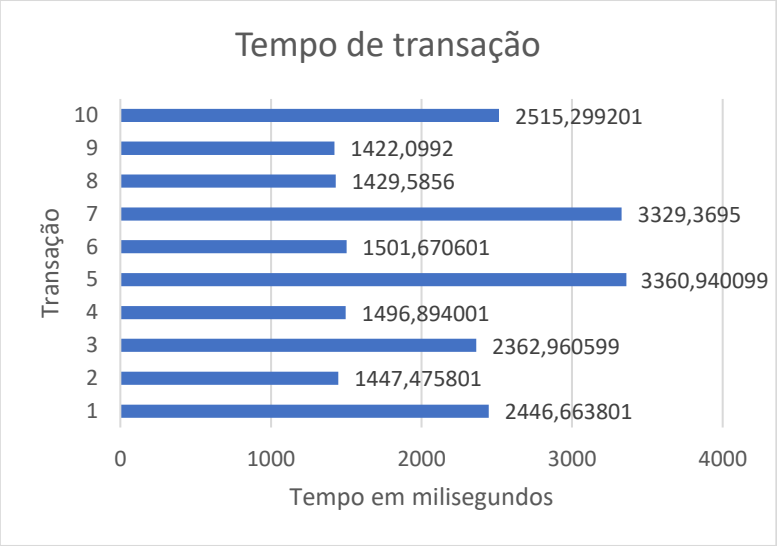

Figura A-184 - Tempo de processamento das transações realizadas na repetição 2 do teste 7

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

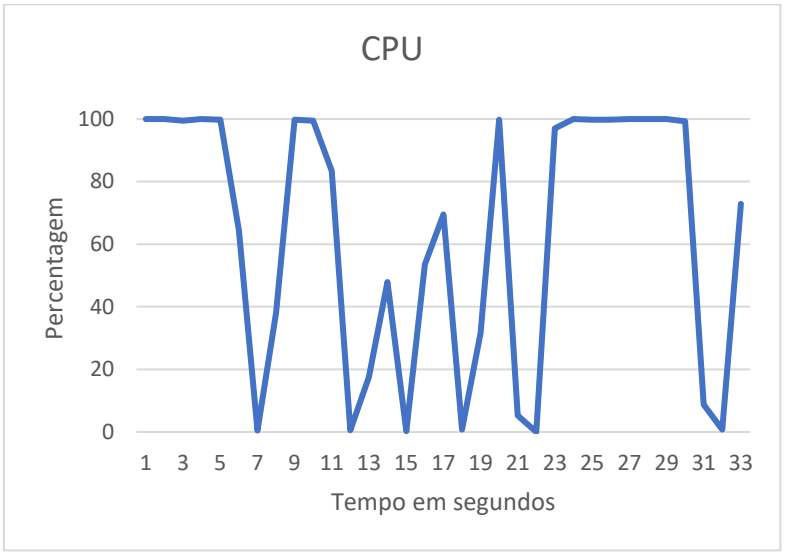

Figura A-185 - Valores de CPU do node 1 para a repetição 3 do teste 7

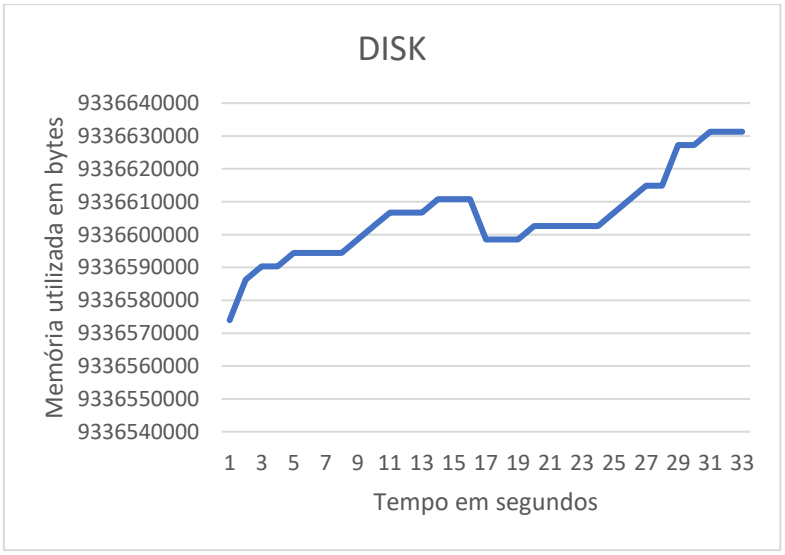

Figura A-186 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 7

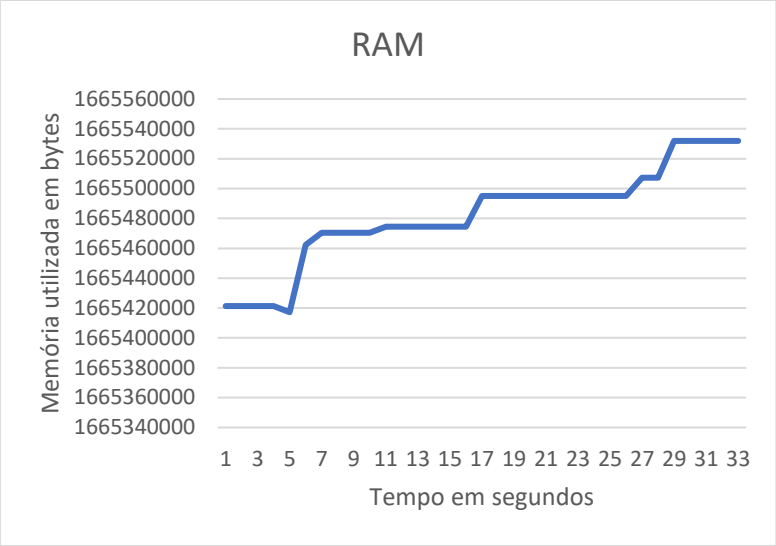

Figura A-187 - Memória em bytes que está a ser utilizada no node 1 para a repetição 3 do teste 7

Node 2

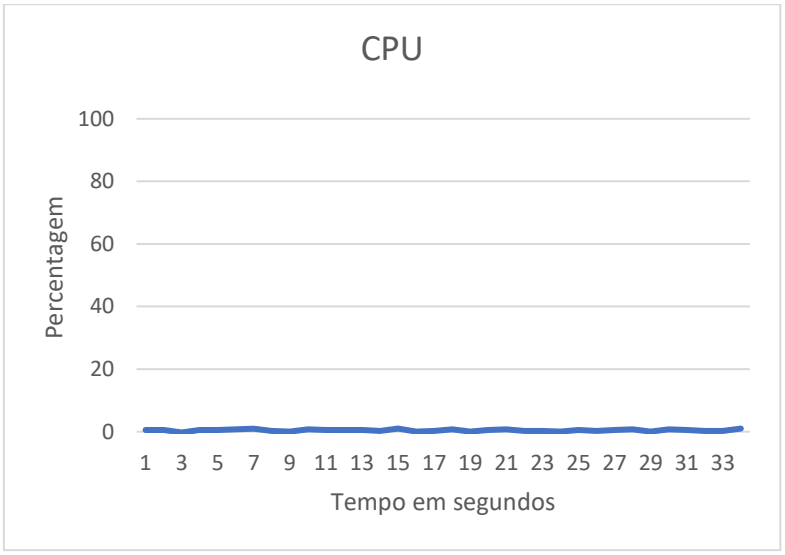

Figura A-188 - Valores de CPU do node 2 para a repetição 3 do teste 7

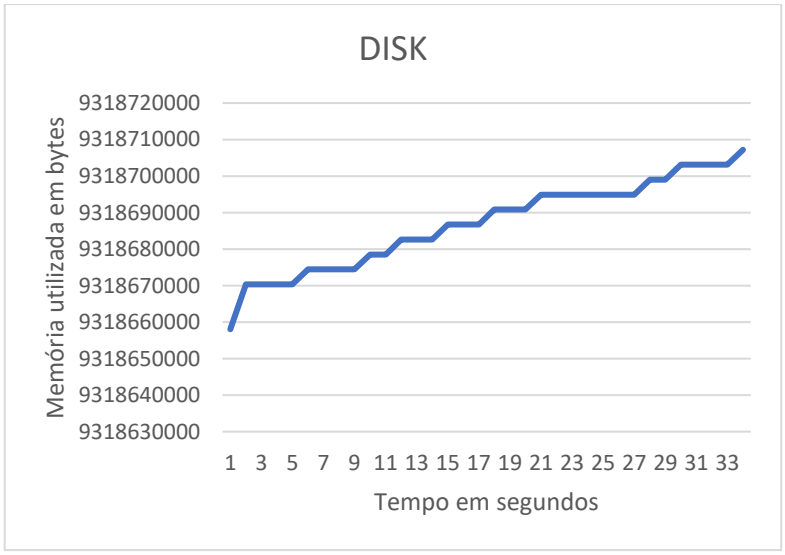

Figura A-189 - Memória utilizada de disco em bytes do node 2 para a repetição 3 do teste 7

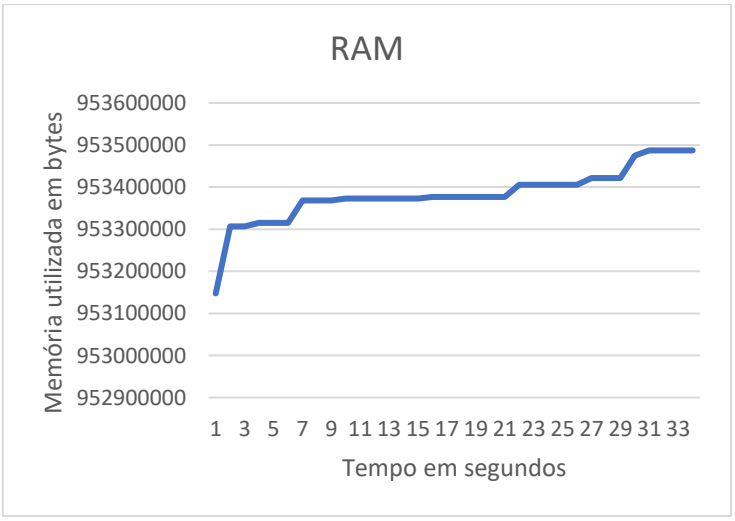

Figura A-190 - Memória em bytes que está a ser utilizada no node 2 para a repetição 3 do teste 7

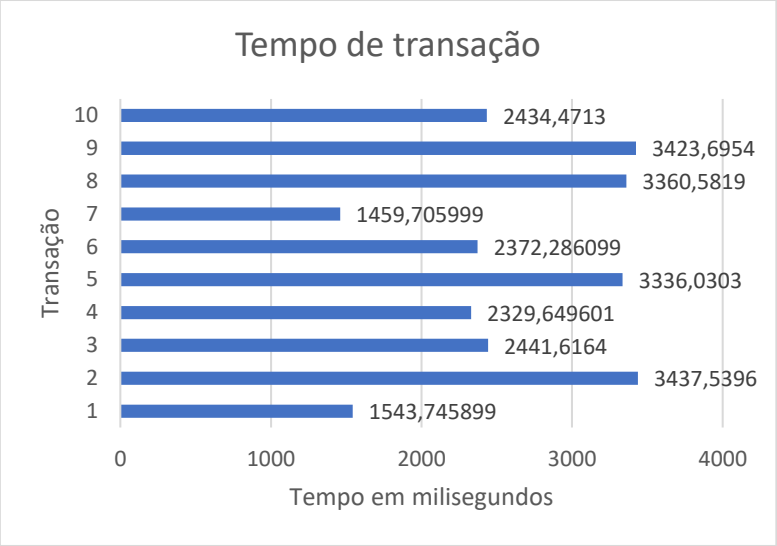

Figura A-191 - Tempo de processamento das transações realizadas na repetição 3 do teste 7

Teste 8

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

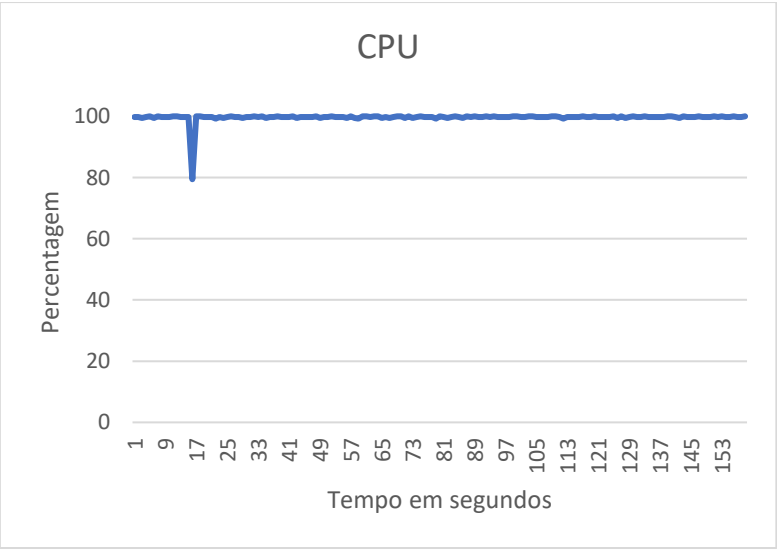

Figura A-192 - Valores de CPU do node 1 para a repetição 1 do teste 8

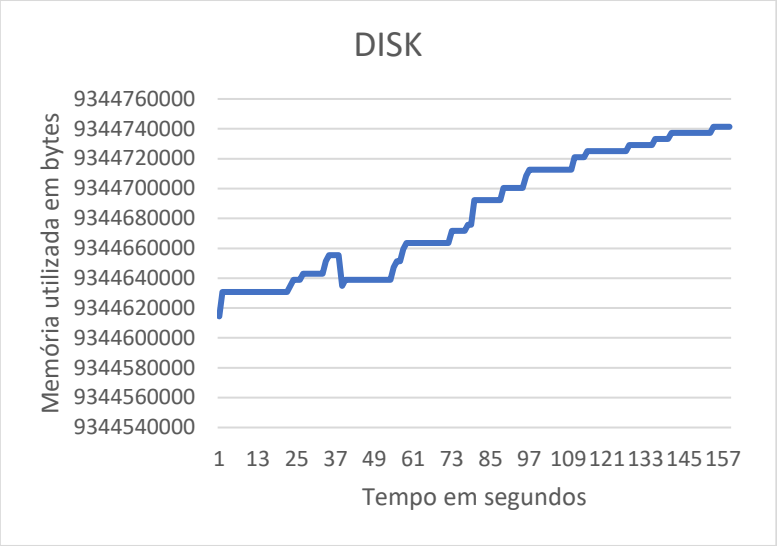

Figura A-193 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 8

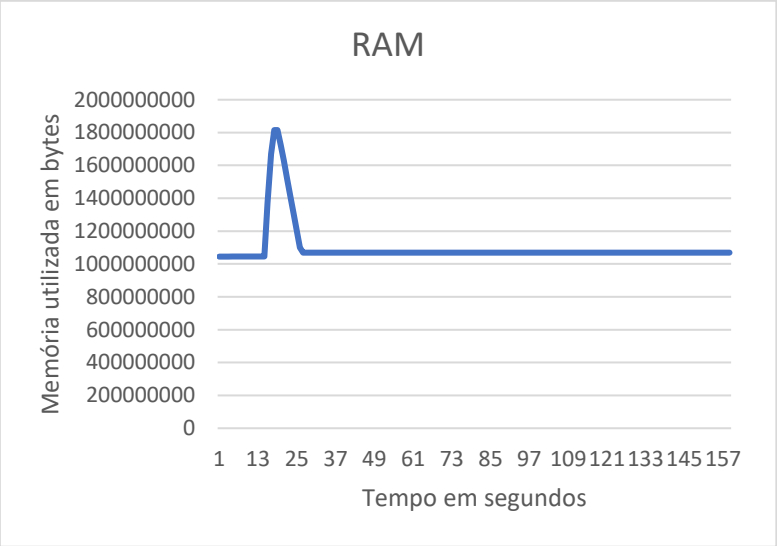

Figura A-194 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 8

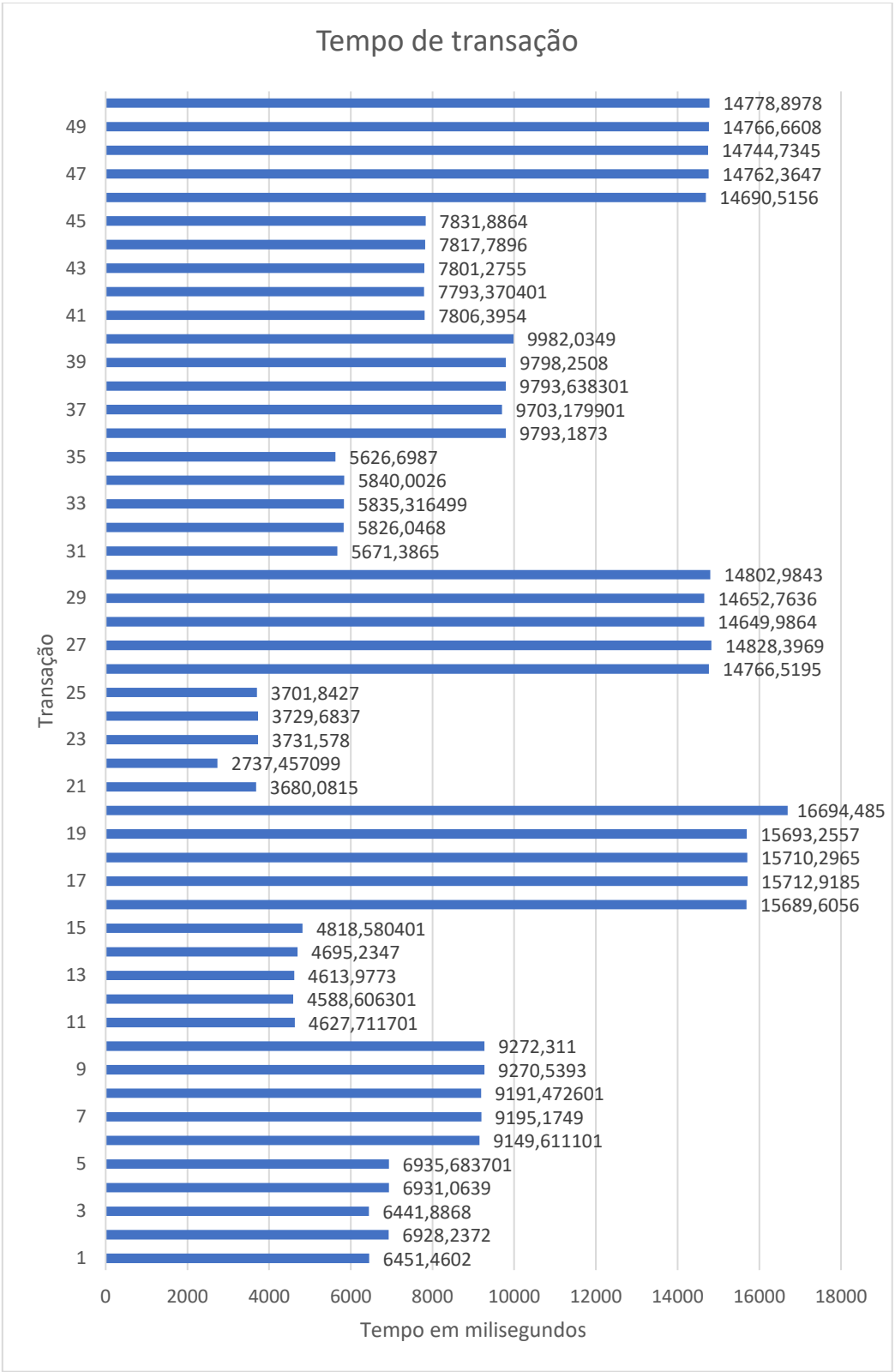

Figura A-195 - Tempo de processamento das transações realizadas na repetição 1 do teste 8

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

Figura A-196 - Valores de CPU do node 1 para a repetição 2 do teste 8

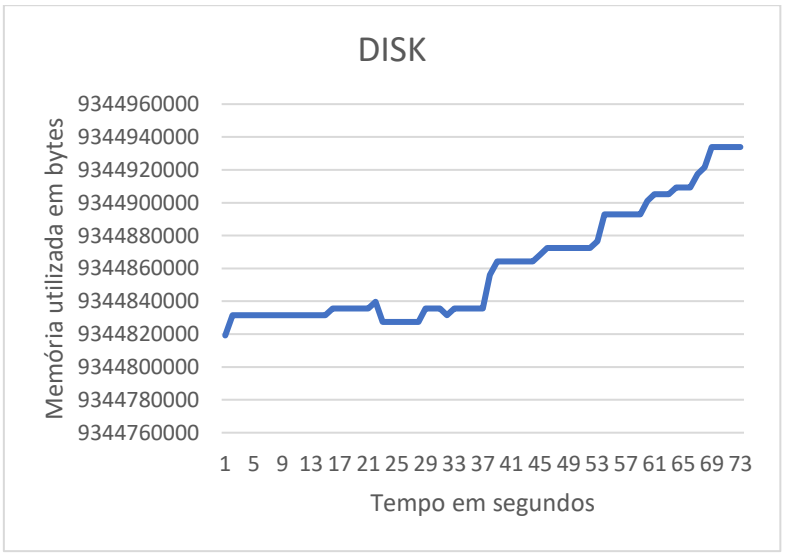

Figura A-197 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 8

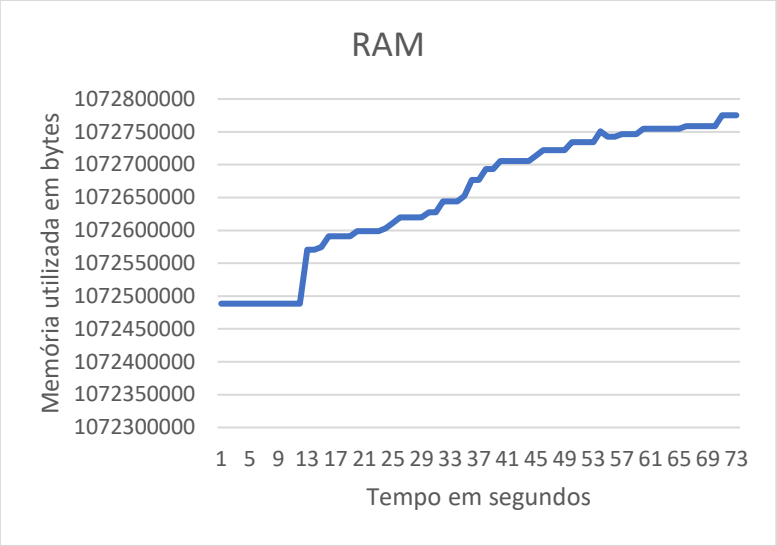

Figura A-198 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 8

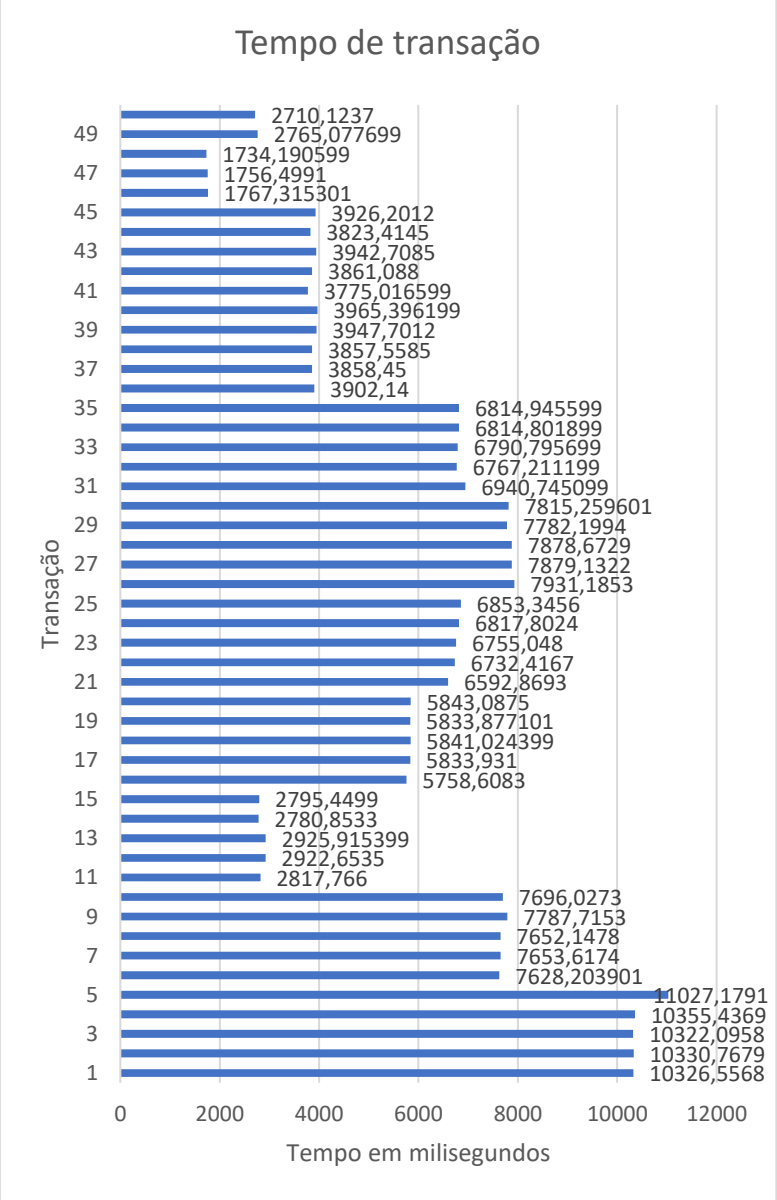

Figura A-199 - Tempo de processamento das transações realizadas na repetição 2 do teste 8

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

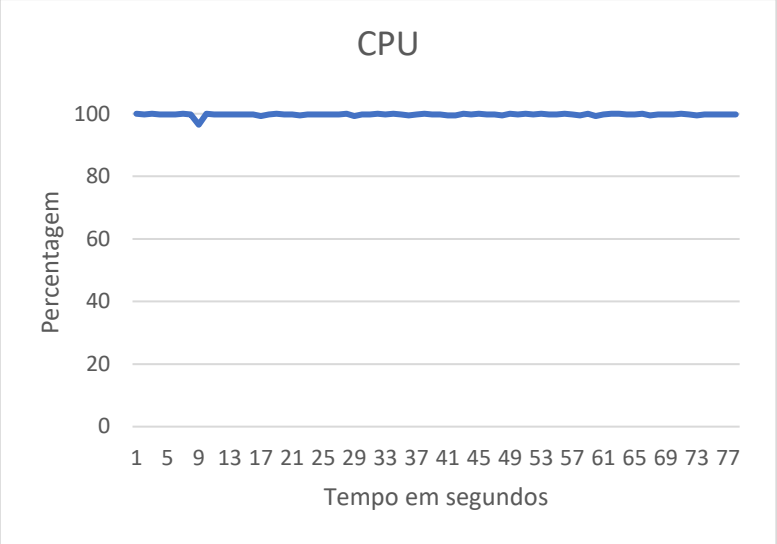

Figura A-200 - Valores de CPU do node 1 para a repetição 3 do teste 8

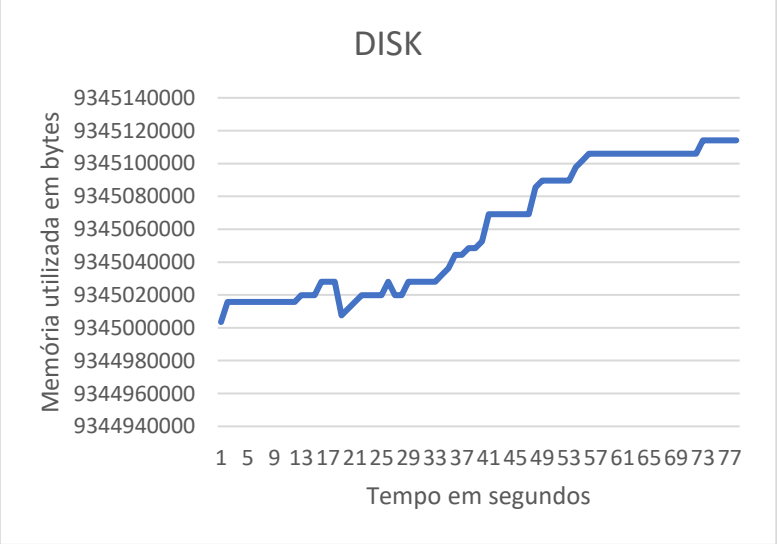

Figura A-201 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 8

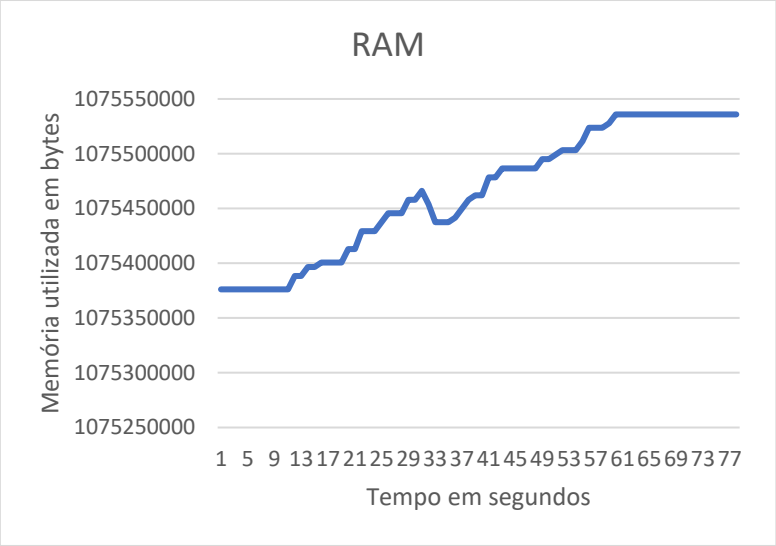

Figura A-202 - Memória em bytes que está a ser utilizada no node 1 para a repetição 3 do teste 8

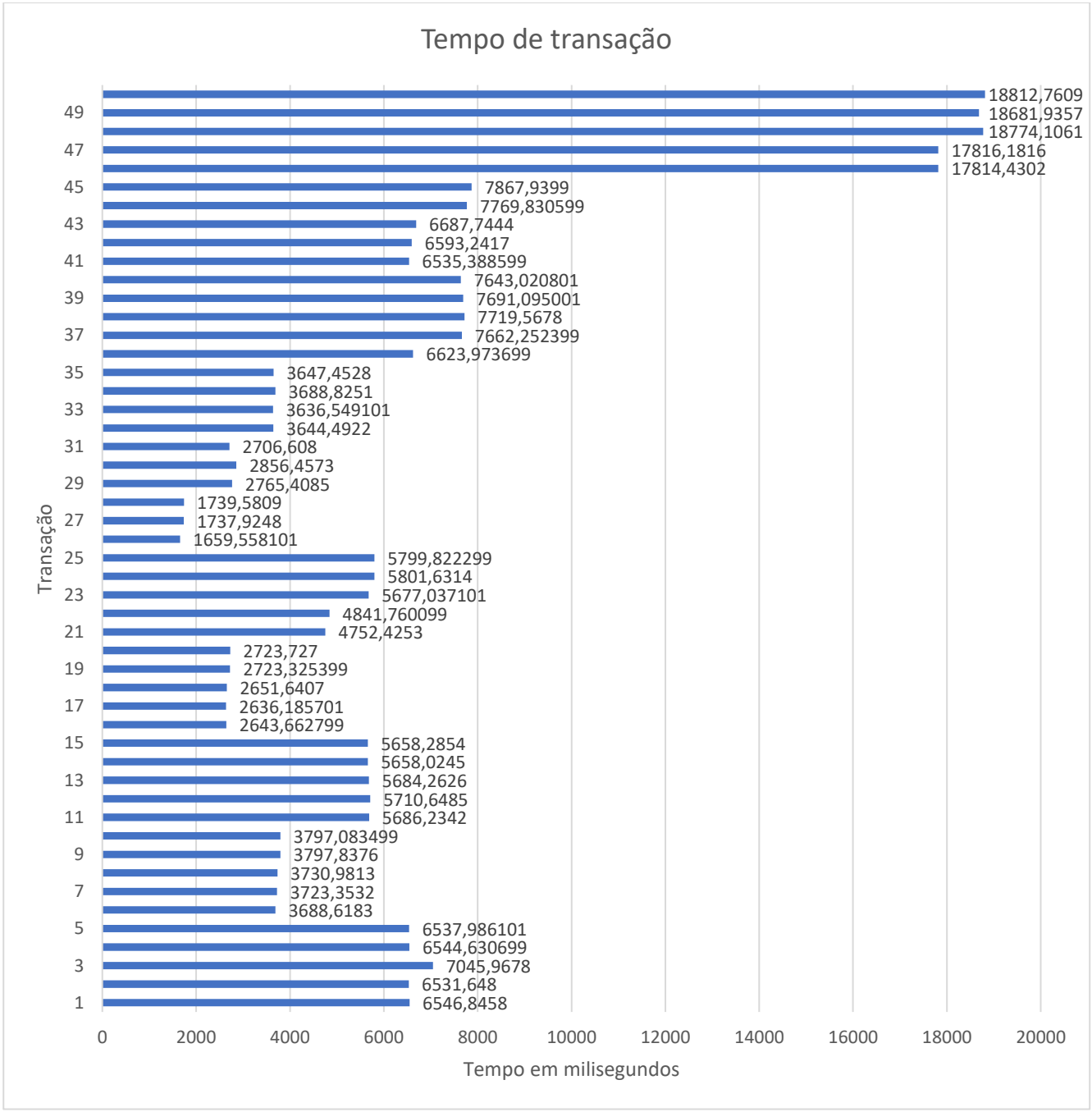

Figura A-203 - Tempo de processamento das transações realizadas na repetição 3 do teste 8

Teste 9

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

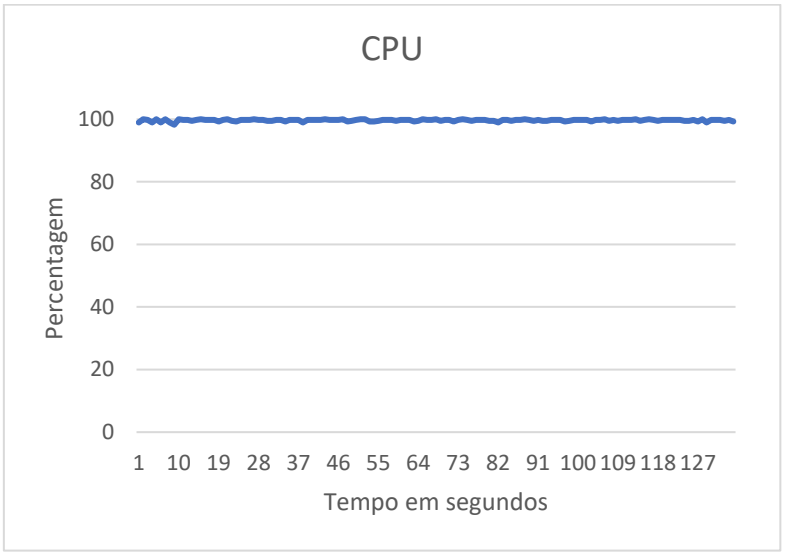

Figura A-204 - Valores de CPU do node 1 para a repetição 1 do teste 9

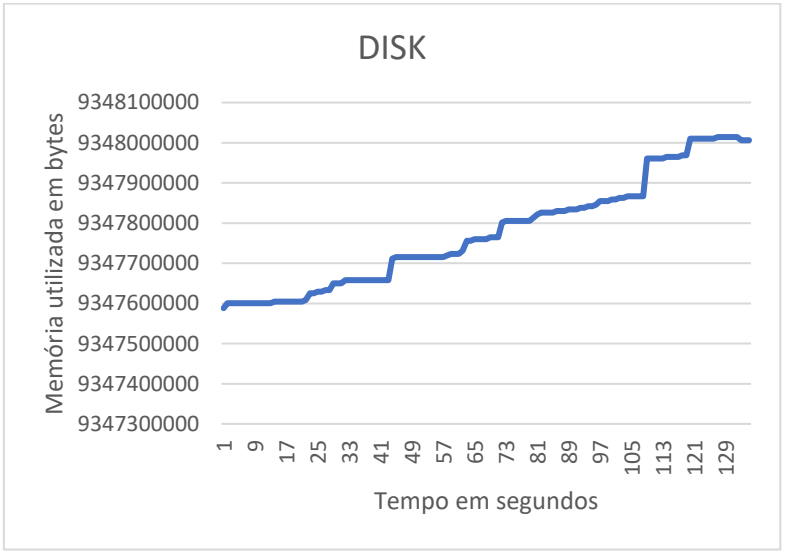

Figura A-205 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 9

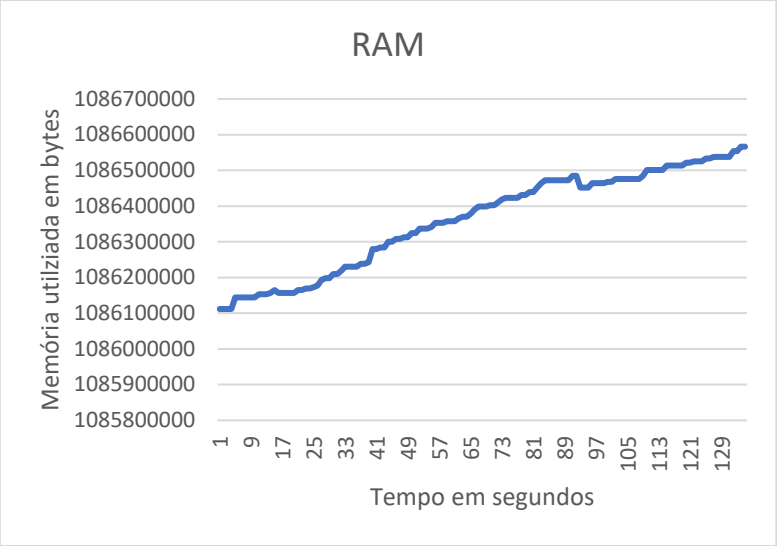

Figura A-206 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 9

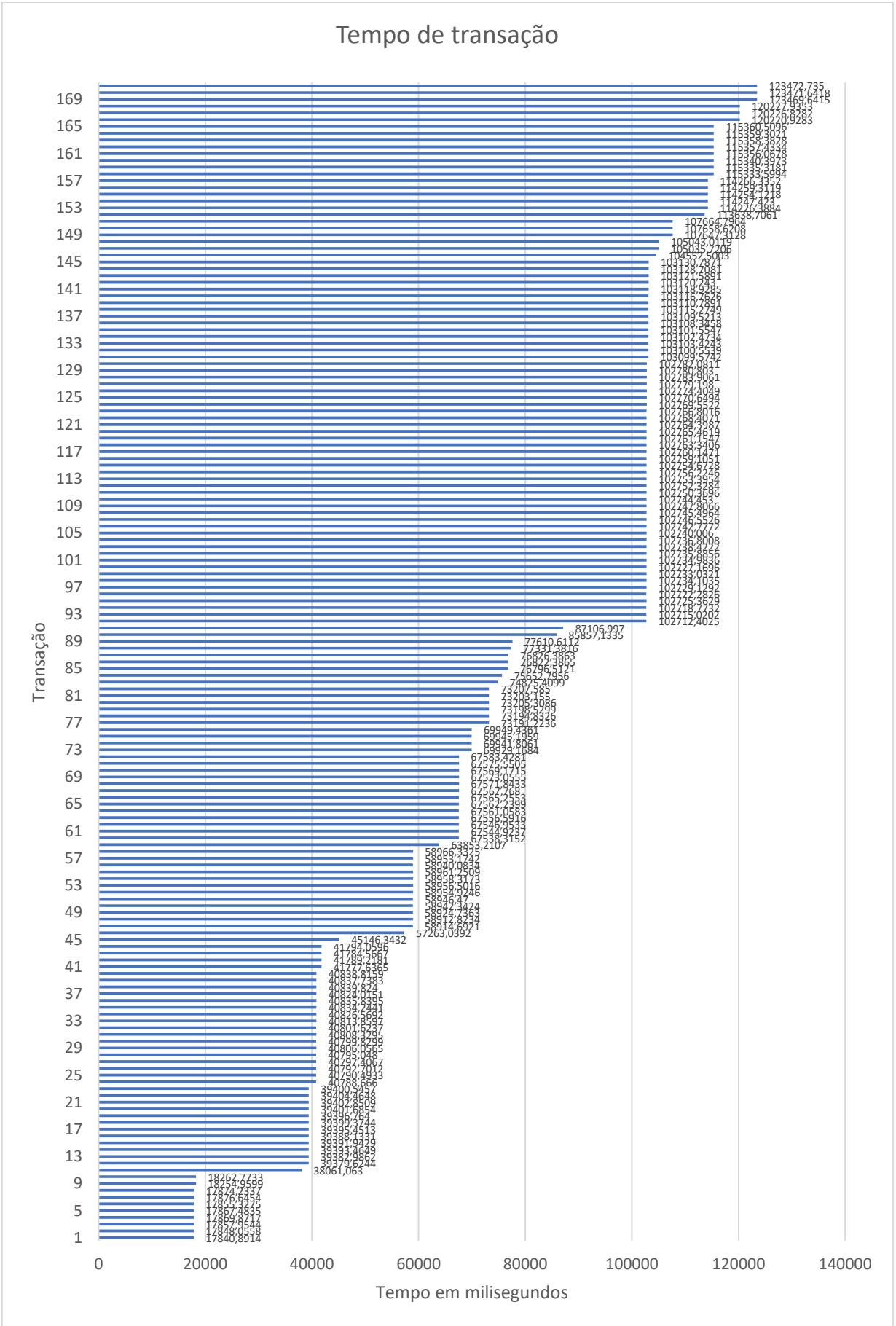

Figura A-207 - Tempo de processamento das transações realizadas na repetição 1 do teste 9

Teste 10

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

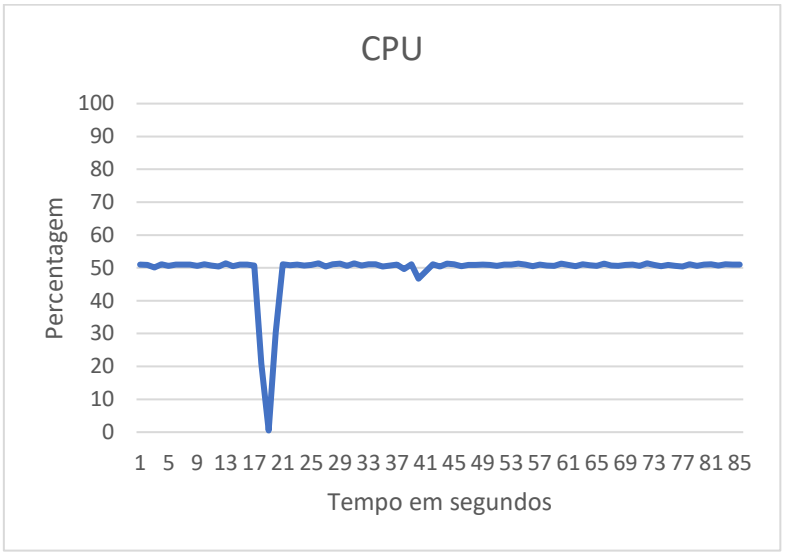

Figura A-208 - Valores de CPU do node 1 para a repetição 1 do teste 10

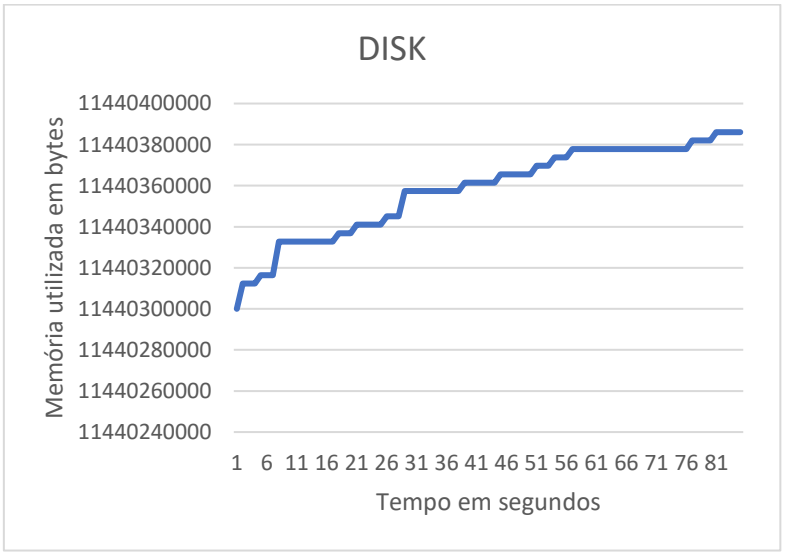

Figura A-209 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 10

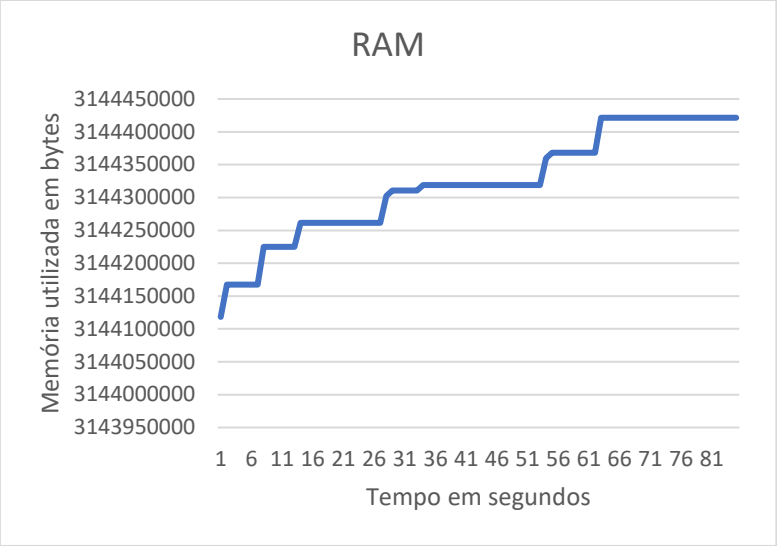

Figura A-210 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 10

Node 2

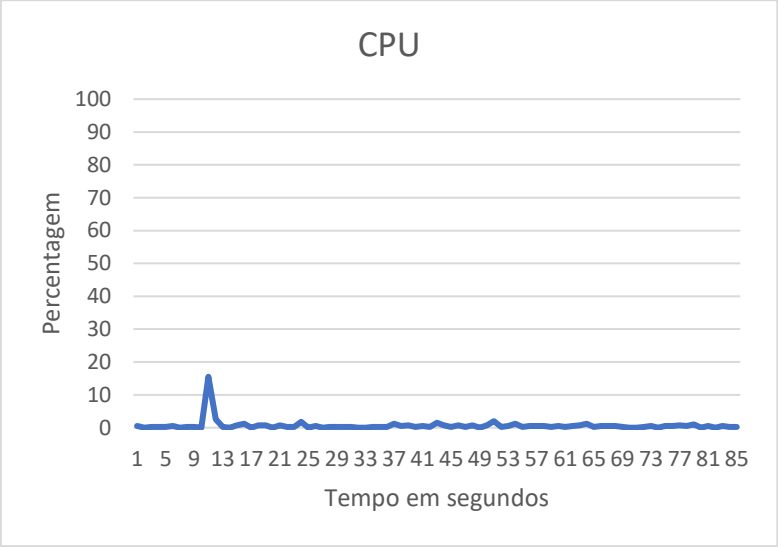

Figura A-211 - Valores de CPU do node 2 para a repetição 1 do teste 10

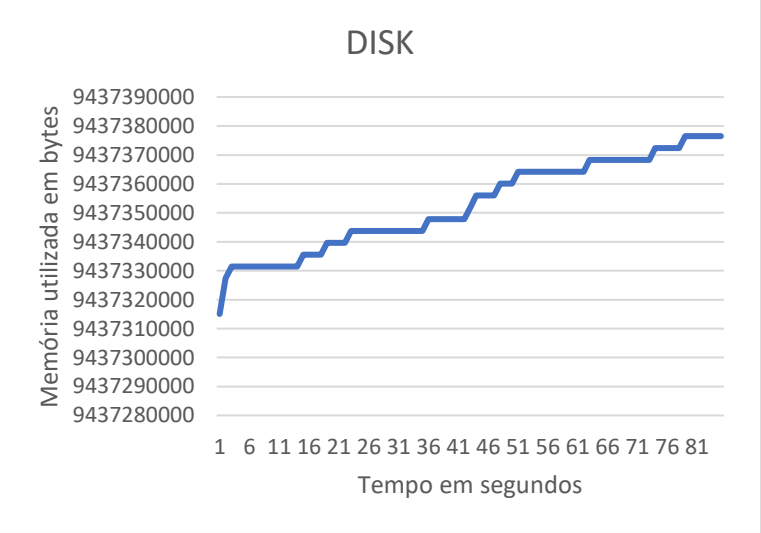

Figura A-212 - Memória utilizada de disco em bytes do node 2 para a repetição 1 do teste 10

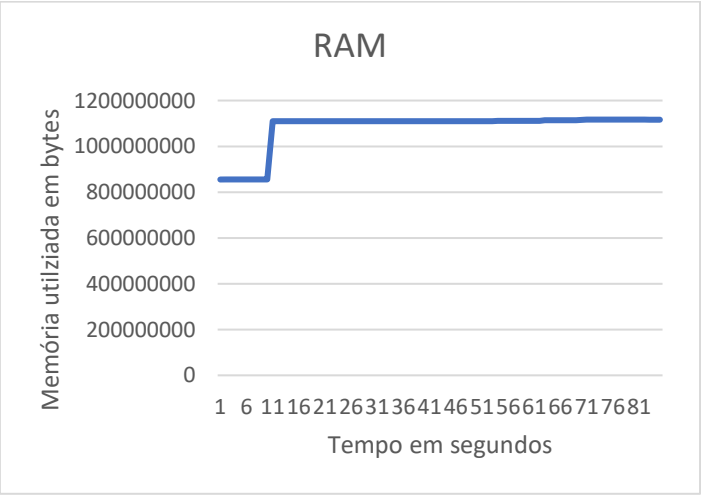

Figura A-213 - Memória em bytes que está a ser utilizada no node 2 para a repetição 1 do teste 10

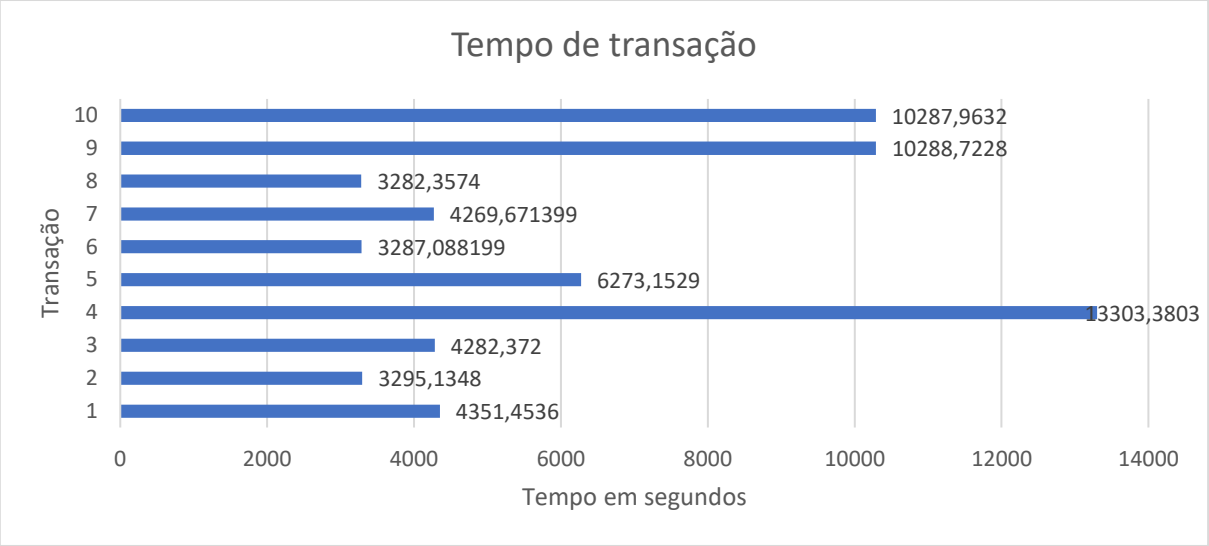

Figura A-214 - Tempo de processamento das transações realizadas na repetição 1 do teste 10

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

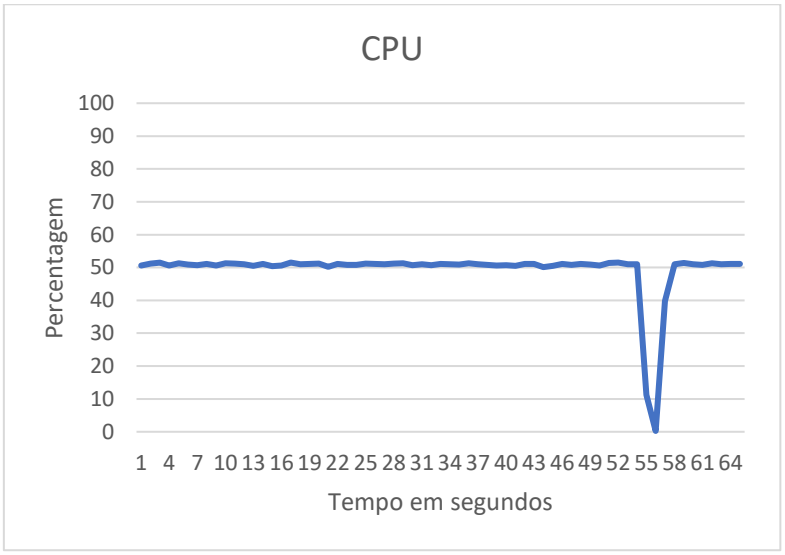

Figura A-215 - Valores de CPU do node 1 para a repetição 2 do teste 10

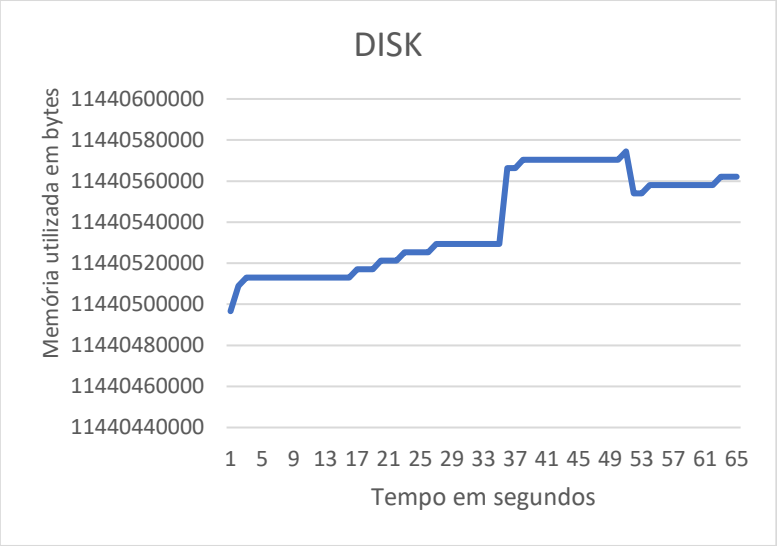

Figura A-216 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 10

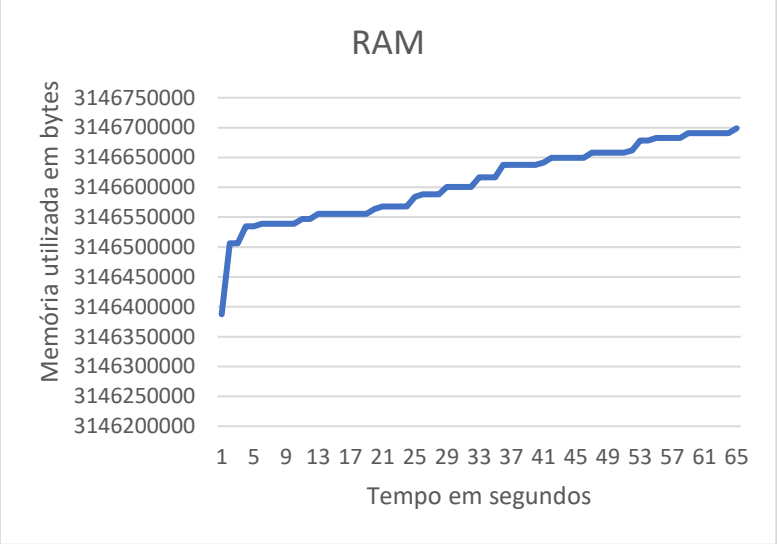

Figura A-217 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 10

Node 2

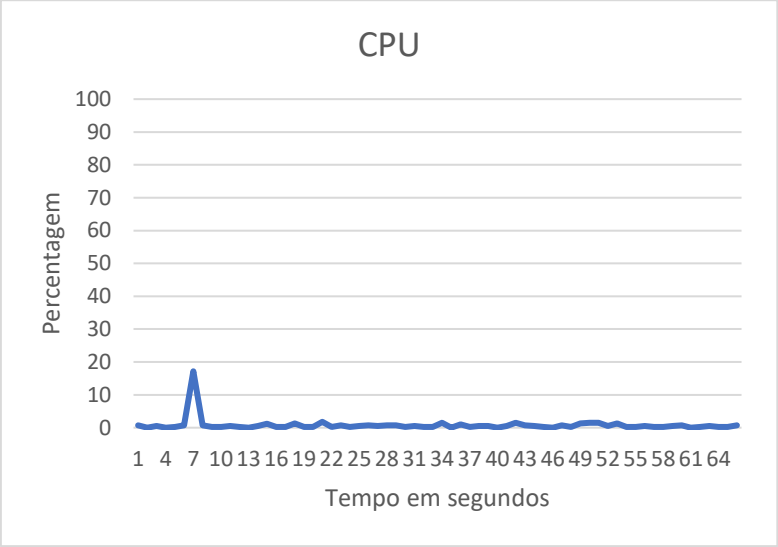

Figura A-218 - Valores de CPU do node 2 para a repetição 2 do teste 10

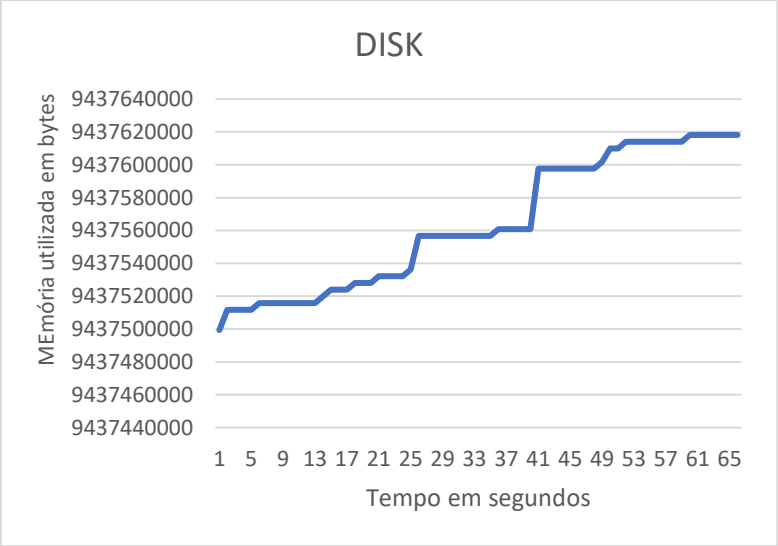

Figura A-219 - Memória utilizada de disco em bytes do node 2 para a repetição 2 do teste 10

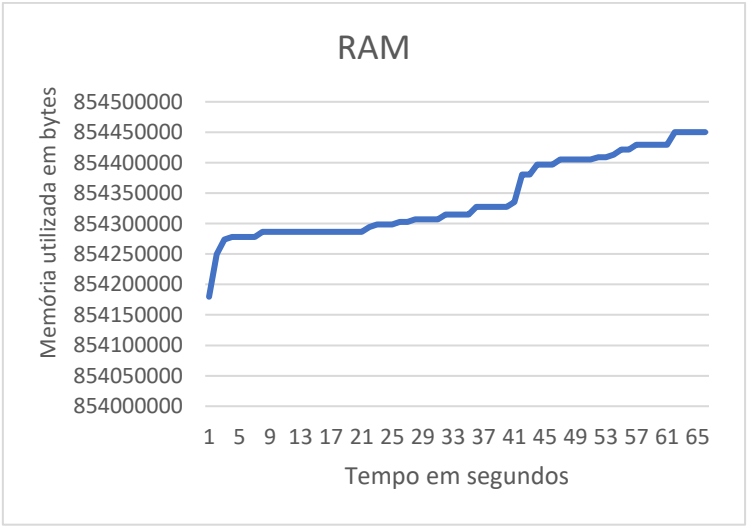

Figura A-220 - Memória em bytes que está a ser utilizada no node 2 para a repetição 2 do teste 10

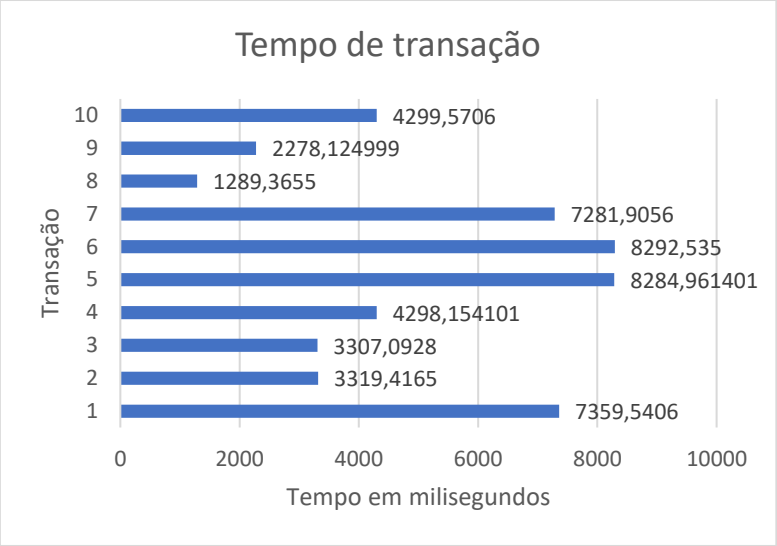

Figura A-221 - Tempo de processamento das transações realizadas na repetição 2 do teste 10

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

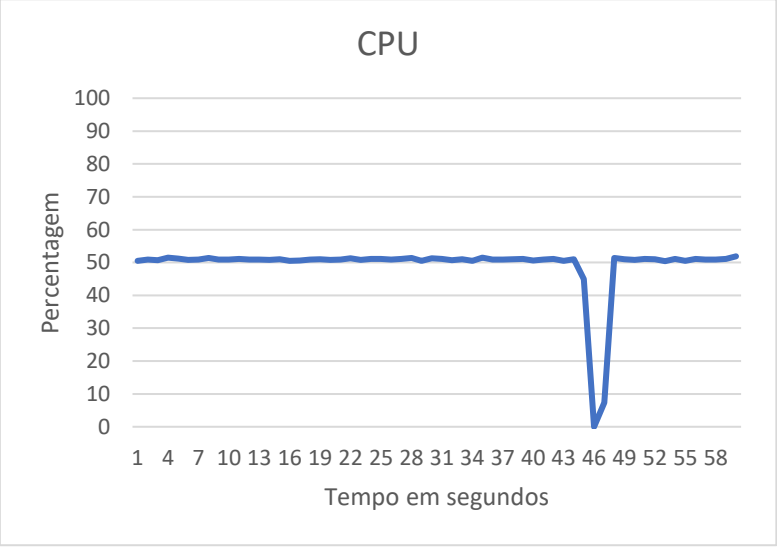

Figura A-222 - Valores de CPU do node 1 para a repetição 3 do teste 10

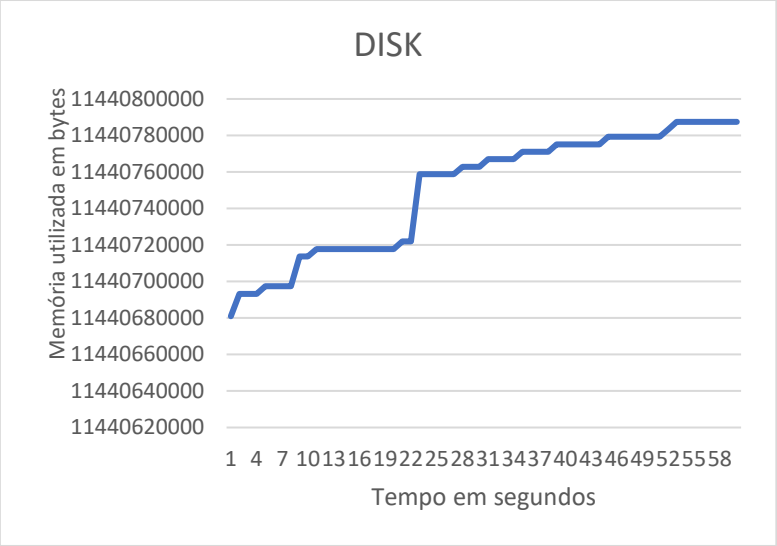

Figura A-223 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 10

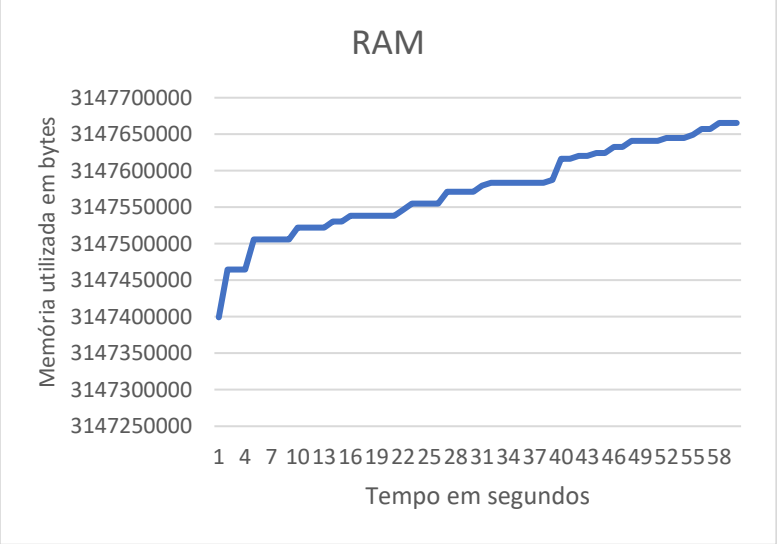

Figura A-224 - Memória em bytes que está a ser utilizada no node 1 para a repetição 3 do teste 10

Node 2

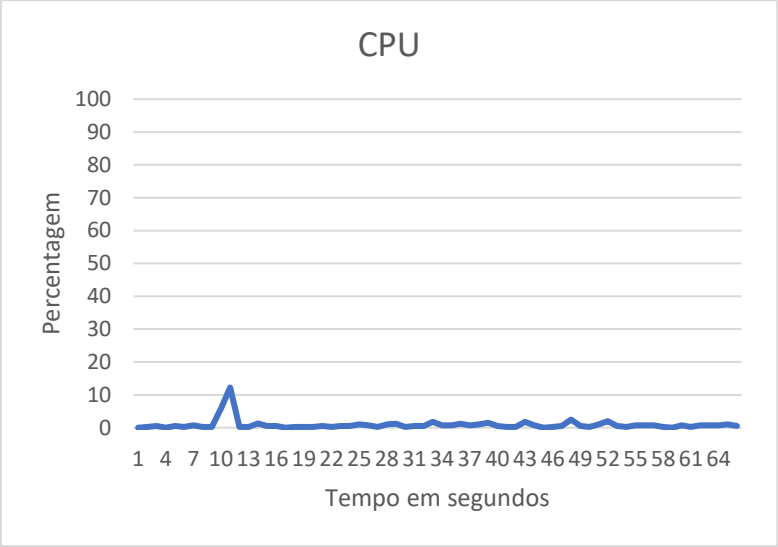

Figura A-225 - Valores de CPU do node 2 para a repetição 3 do teste 10

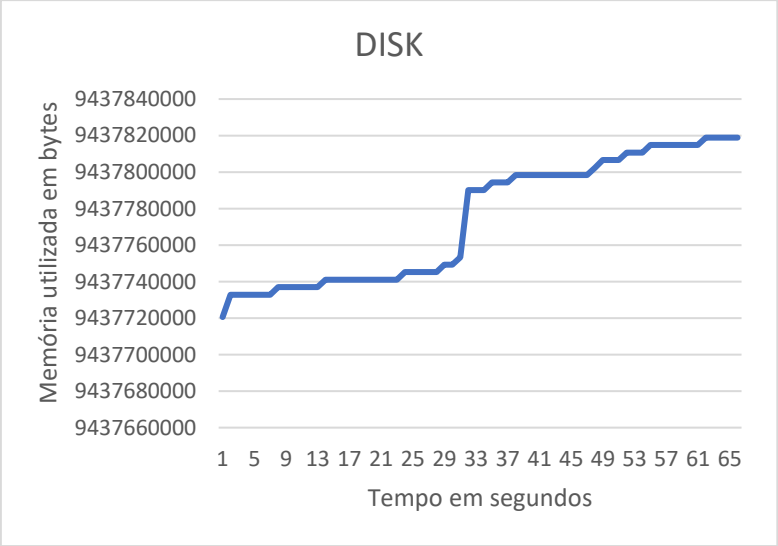

Figura A-226 - Memória utilizada de disco em bytes do node 2 para a repetição 3 do teste 10

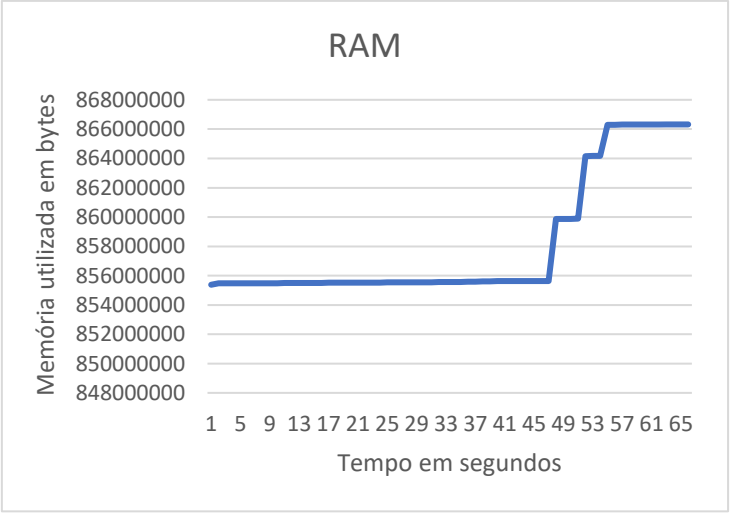

Figura A-227 - Memória em bytes que está a ser utilizada no node 2 para a repetição 3 do teste 10

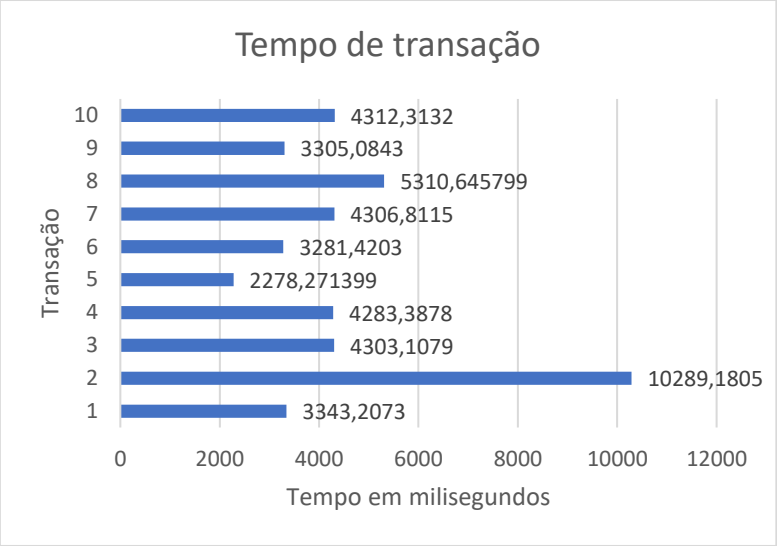

Figura A-228 - Tempo de processamento das transações realizadas na repetição 3 do teste 10

Teste 11

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

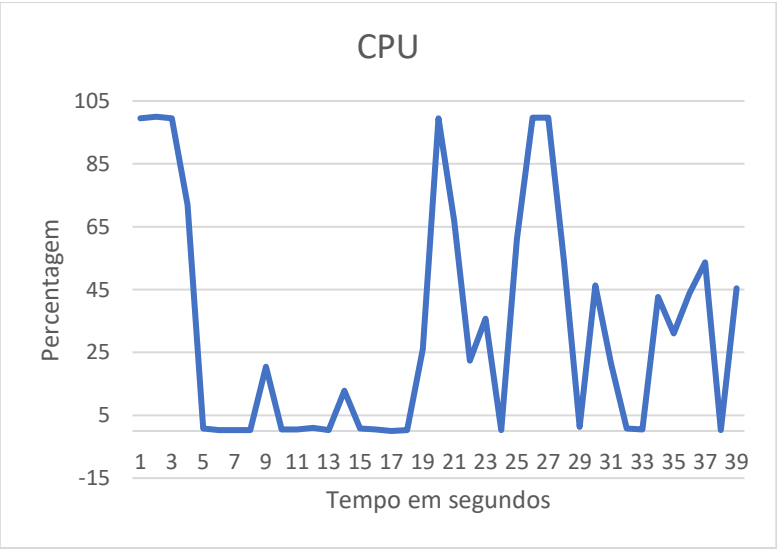

Figura A-229 - Valores de CPU do node 1 para a repetição 1 do teste 11

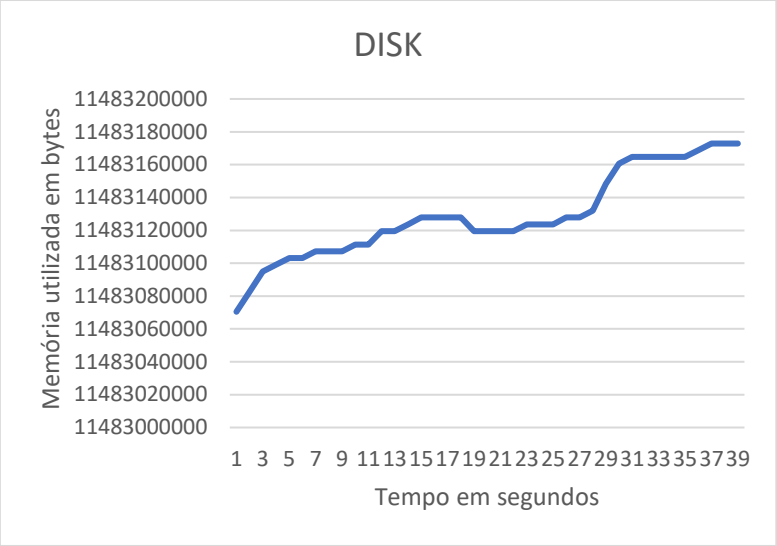

Figura A-230 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 11

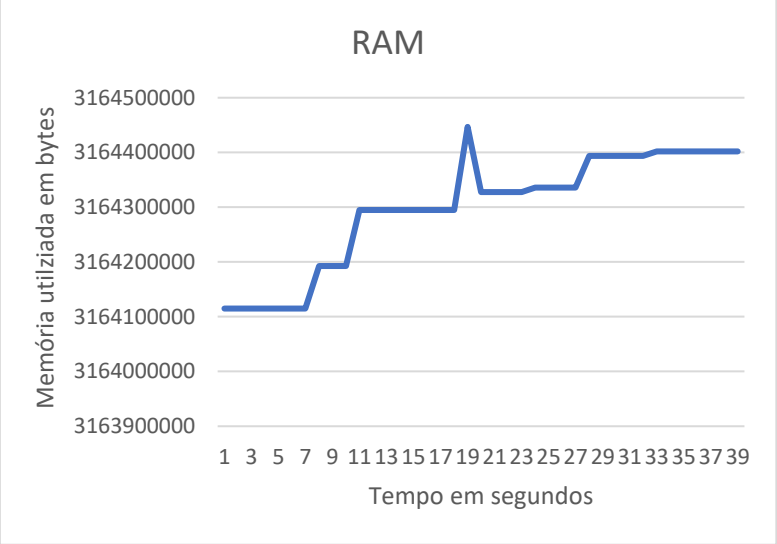

Figura A-231 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 11

Node 2 – Miner2

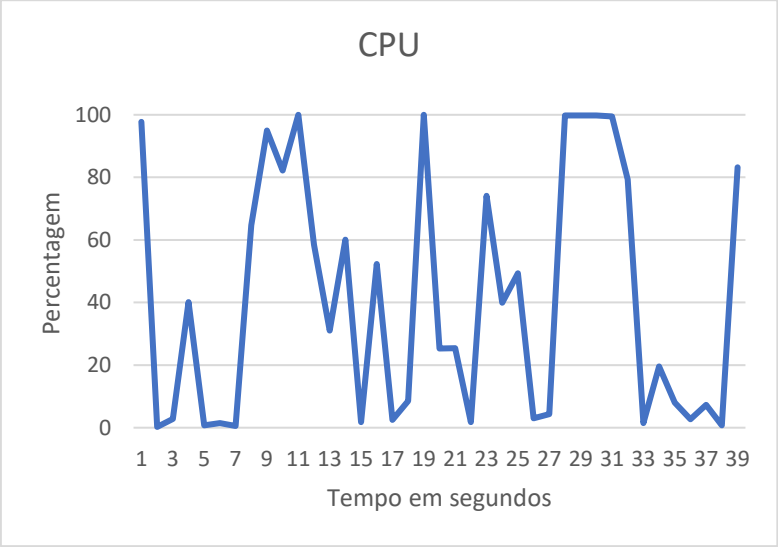

Figura A-232 - Valores de CPU do node 2 para a repetição 1 do teste 11

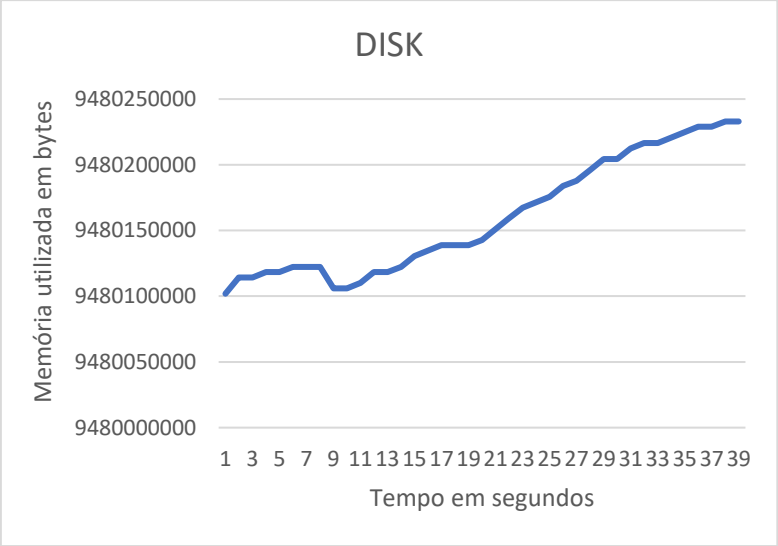

Figura A-233 - Memória utilizada de disco em bytes do node 2 para a repetição 1 do teste 11

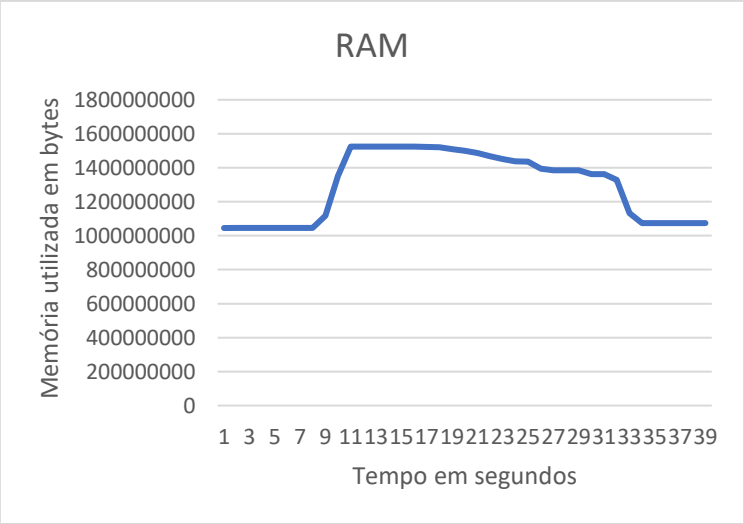

Figura A-234 - Memória em bytes que está a ser utilizada no node 2 para a repetição 1 do teste 11

Node 3

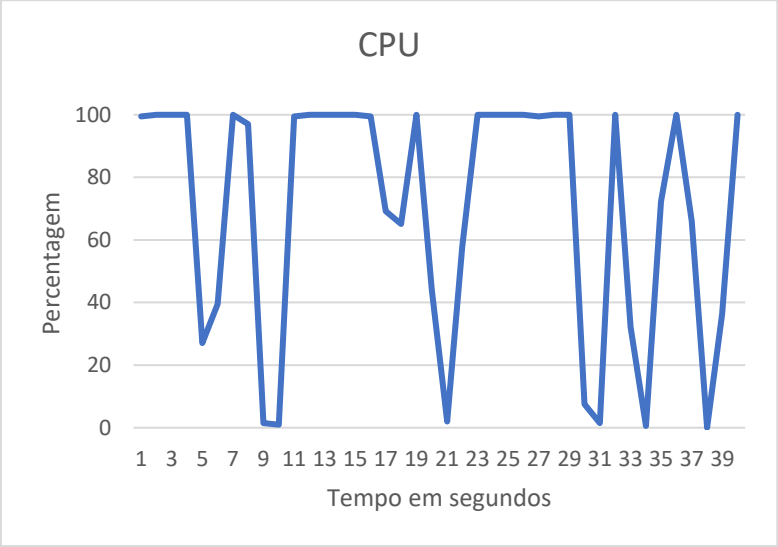

Figura A-235 - Valores de CPU do node 3 para a repetição 1 do teste 11

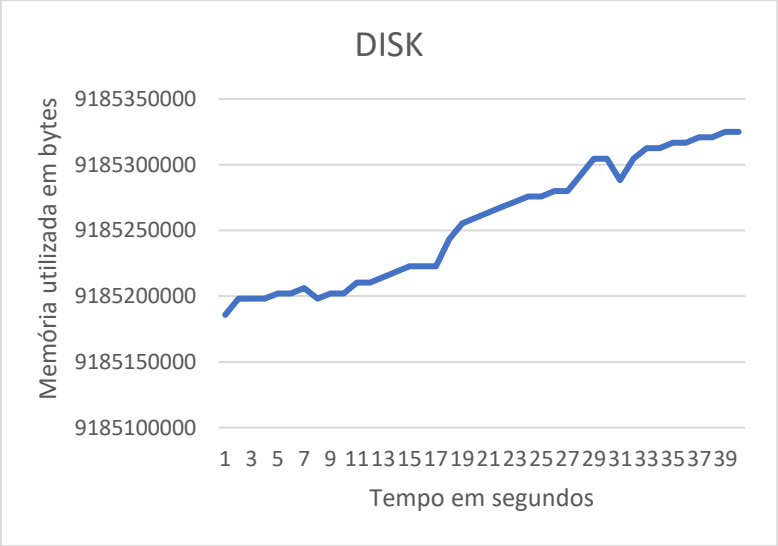

Figura A-236 - Memória utilizada de disco em bytes do node 3 para a repetição 1 do teste 11

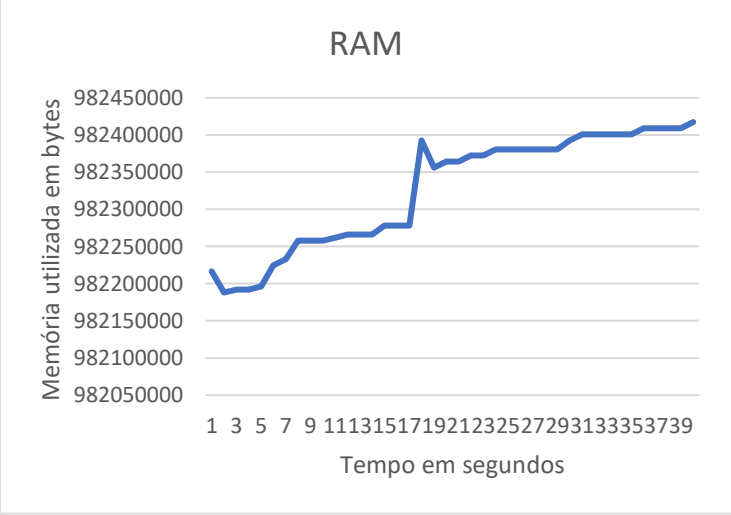

Figura A-237 - Memória em bytes que está a ser utilizada no node 3 para a repetição 1 do teste 11

Node 4

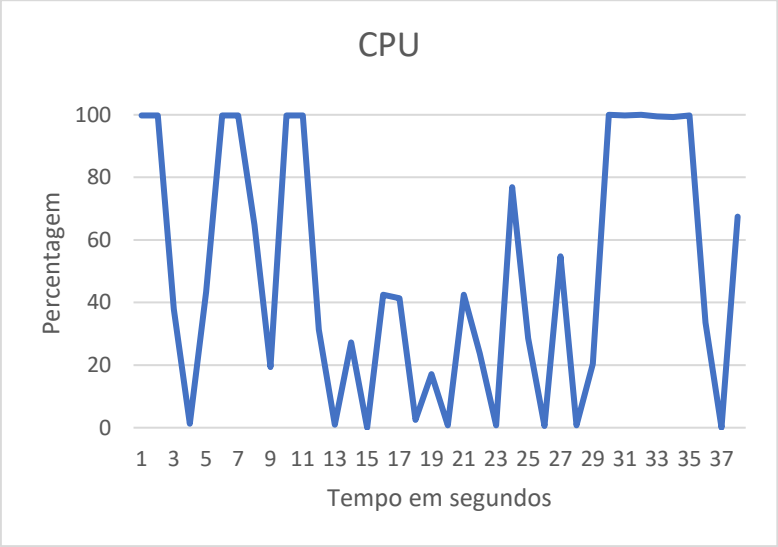

Figura A-238 - Valores de CPU do node 4 para a repetição 1 do teste 11

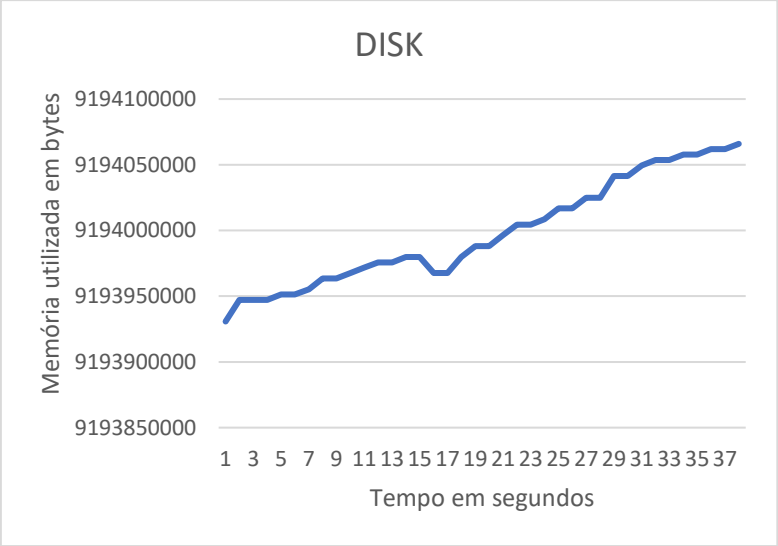

Figura A-239 - Memória utilizada de disco em bytes do node 4 para a repetição 1 do teste 11

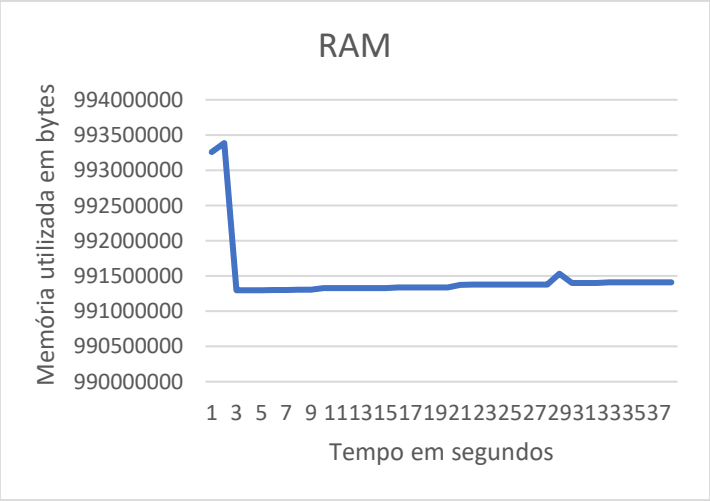

Figura A-240 - Memória em bytes que está a ser utilizada no node 4 para a repetição 1 do teste 11

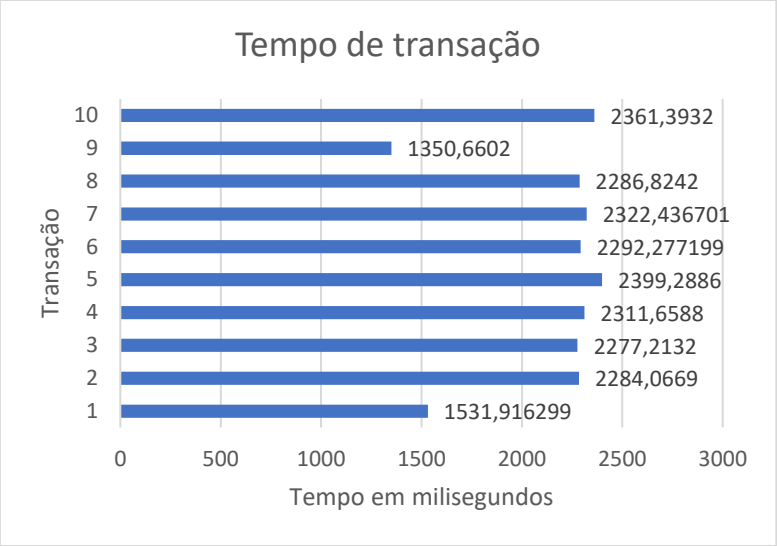

Figura A-241 - Tempo de processamento das transações realizadas na repetição 1 do teste 11

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

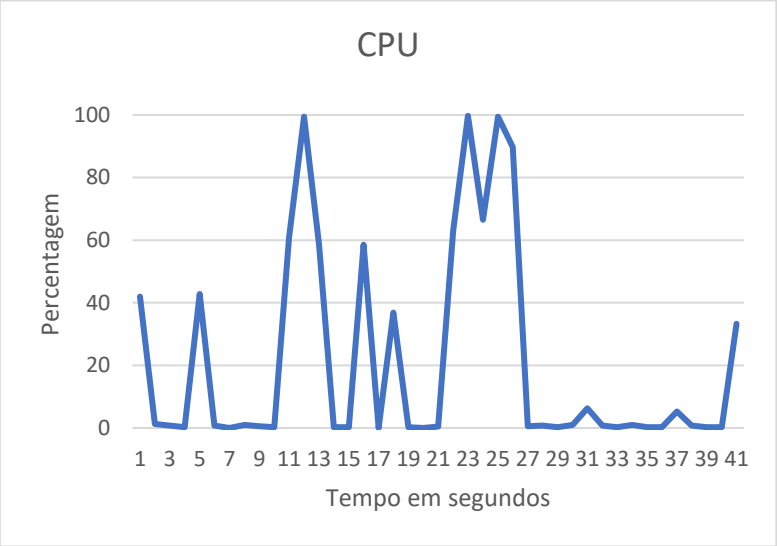

Figura A-242 - Valores de CPU do node 1 para a repetição 2 do teste 11

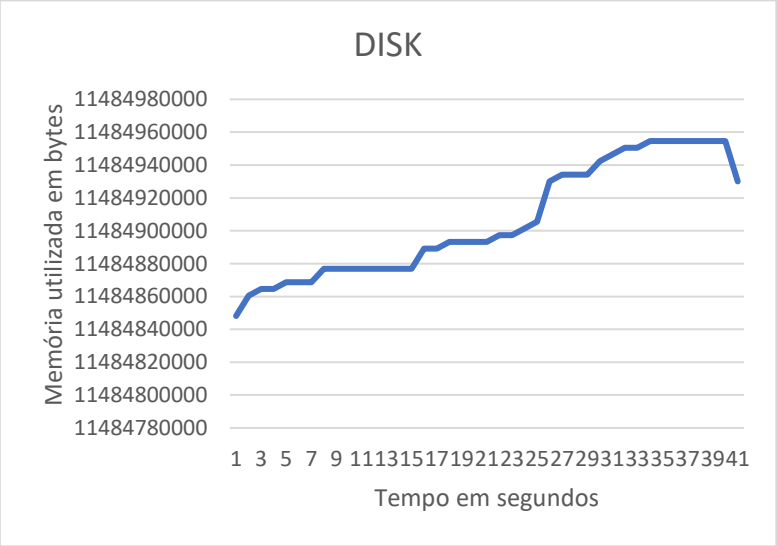

Figura A-243 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 11

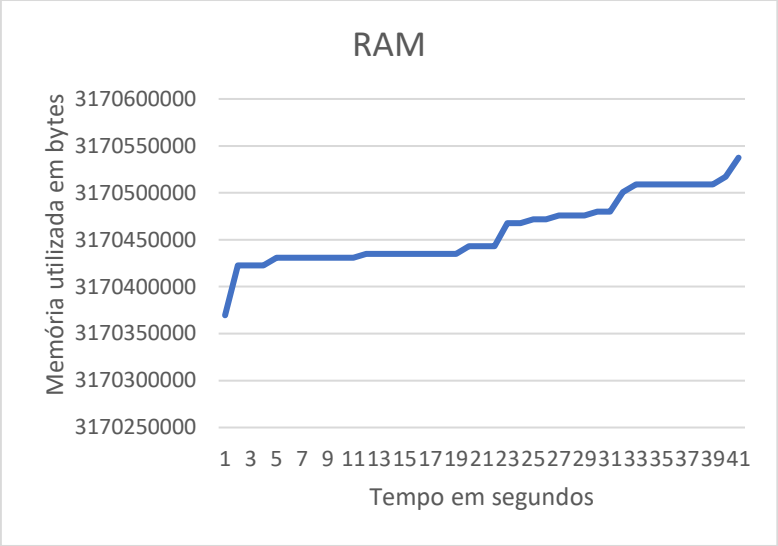

Figura A-244 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 11

Node 2 – Miner2

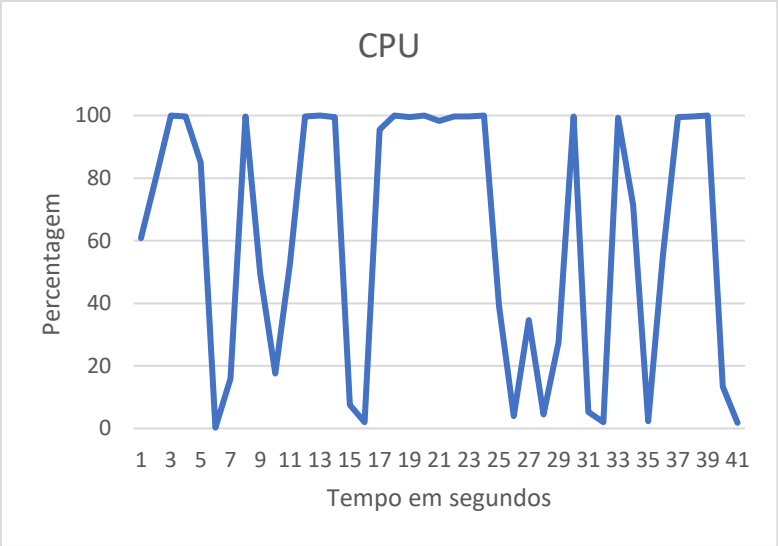

Figura A-245 - Valores de CPU do node 2 para a repetição 2 do teste 11

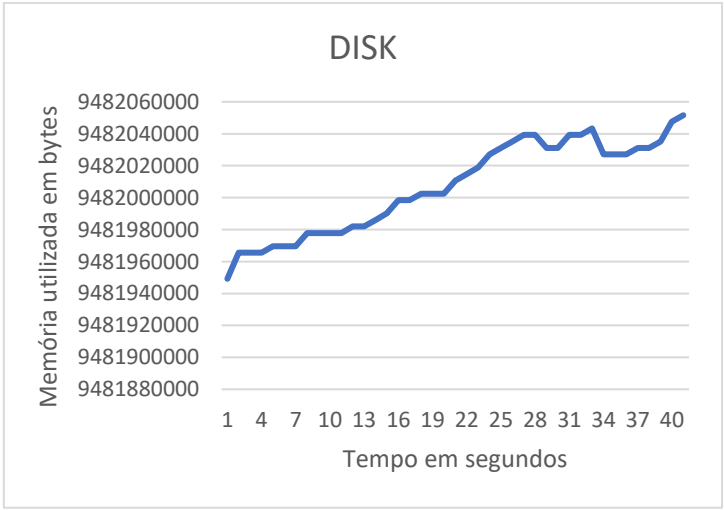

Figura A-246 - Memória utilizada de disco em bytes do node 2 para a repetição 2 do teste 11

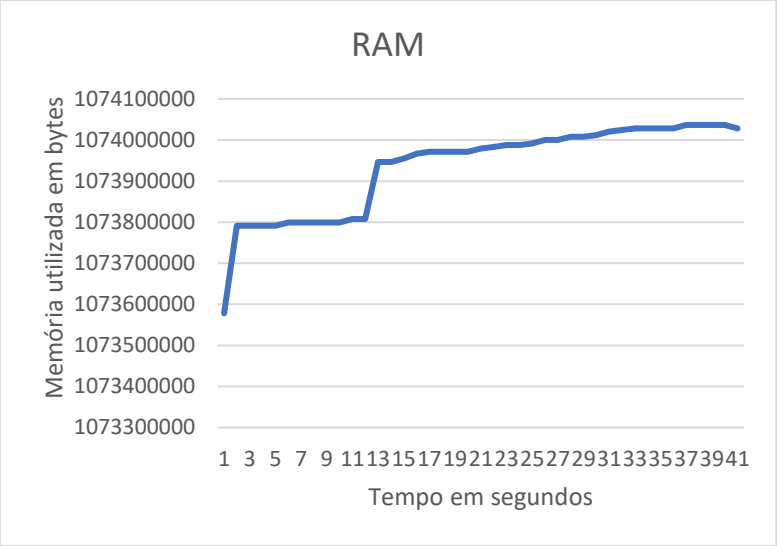

Figura A-247 - Memória em bytes que está a ser utilizada no node 2 para a repetição 2 do teste 11

Node 3

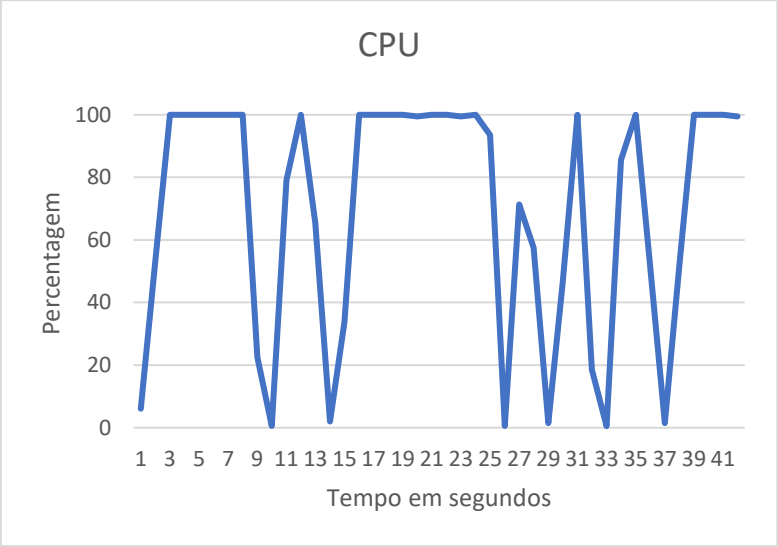

Figura A-248 - Valores de CPU do node 3 para a repetição 2 do teste 11

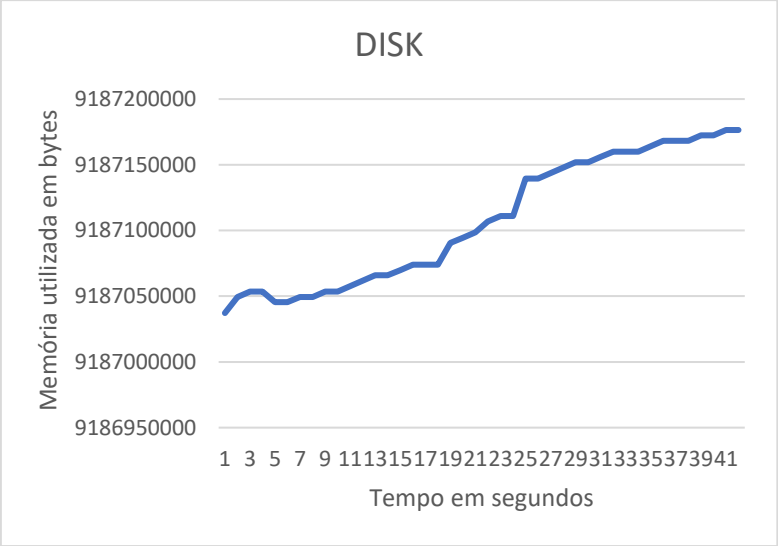

Figura A-249 - Memória utilizada de disco em bytes do node 3 para a repetição 2 do teste 11

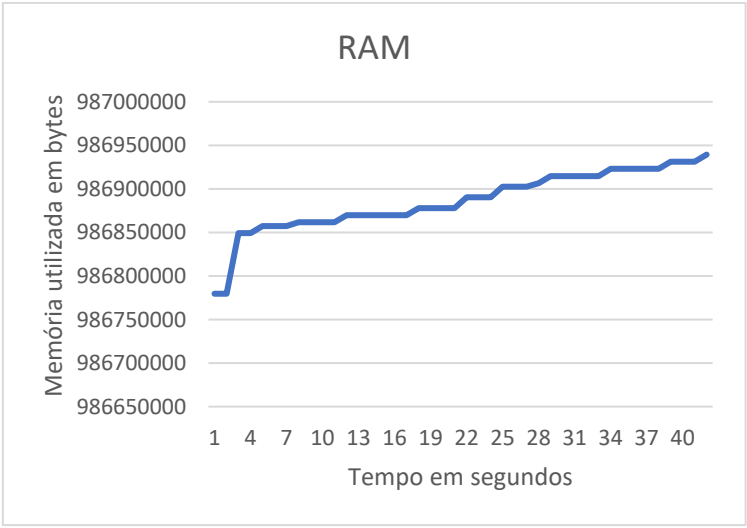

Figura A-250 - Memória em bytes que está a ser utilizada no node 3 para a repetição 2 do teste 11

Node 4

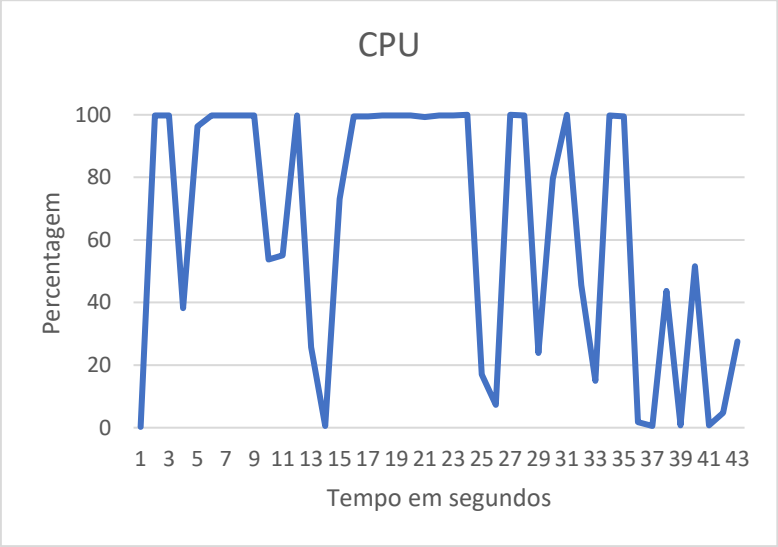

Figura A-251 - Valores de CPU do node 4 para a repetição 2 do teste 11

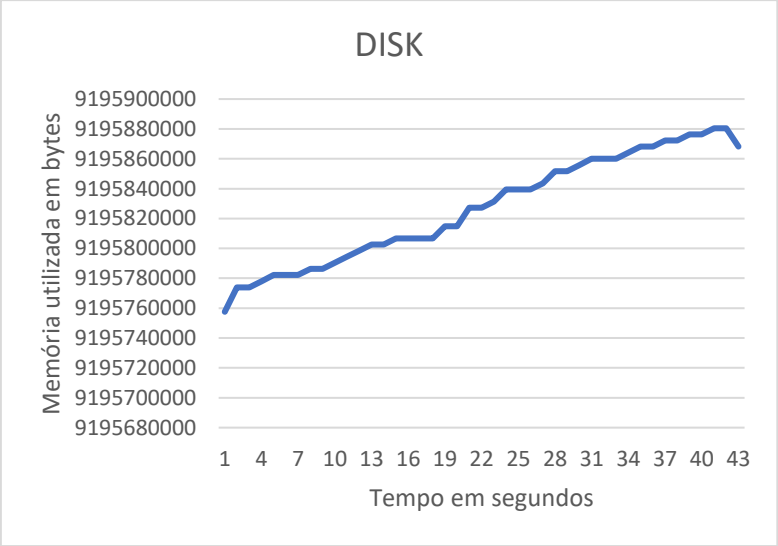

Figura A-252 - Memória utilizada de disco em bytes do node 4 para a repetição 2 do teste 11

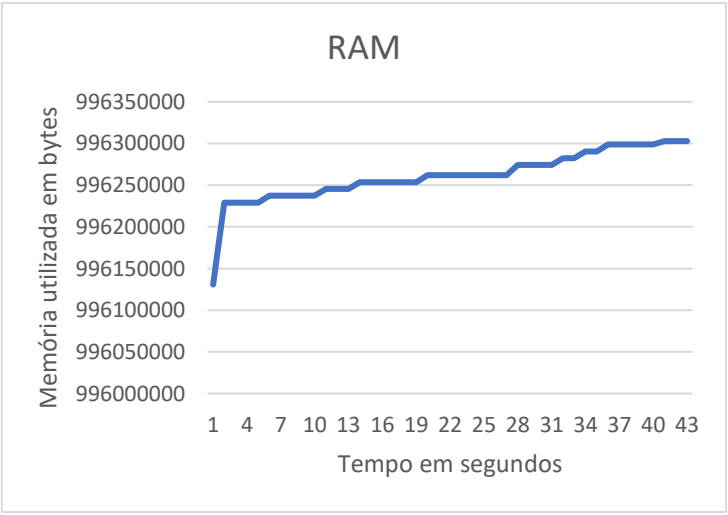

Figura A-253 - Memória em bytes que está a ser utilizada no node 4 para a repetição 2 do teste 11

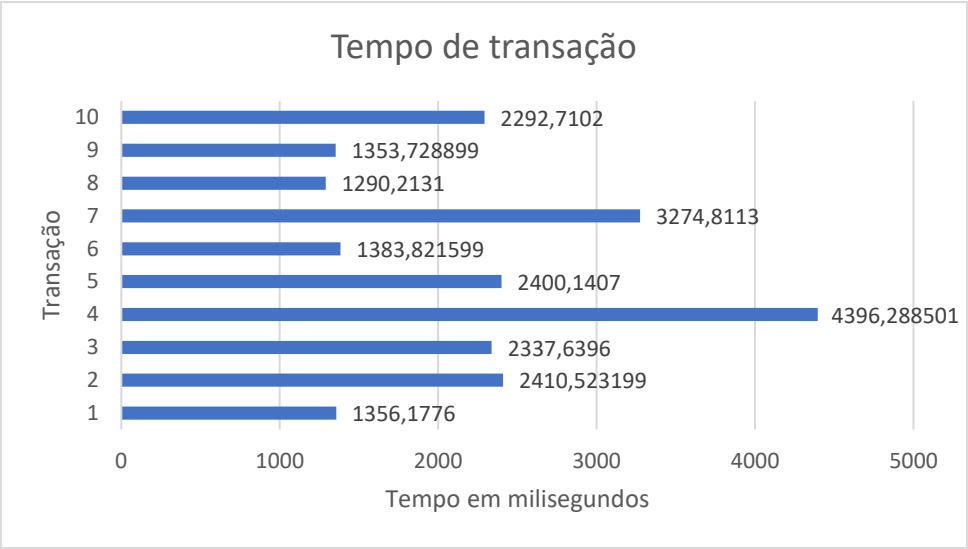

Figura A-254 - Tempo de processamento das transações realizadas na repetição 2 do teste 11

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

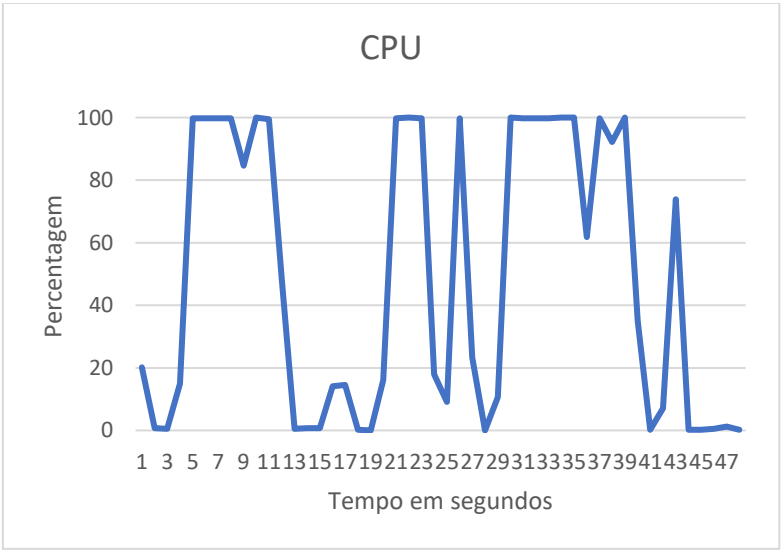

Figura A-255 - Valores de CPU do node 1 para a repetição 3 do teste 11

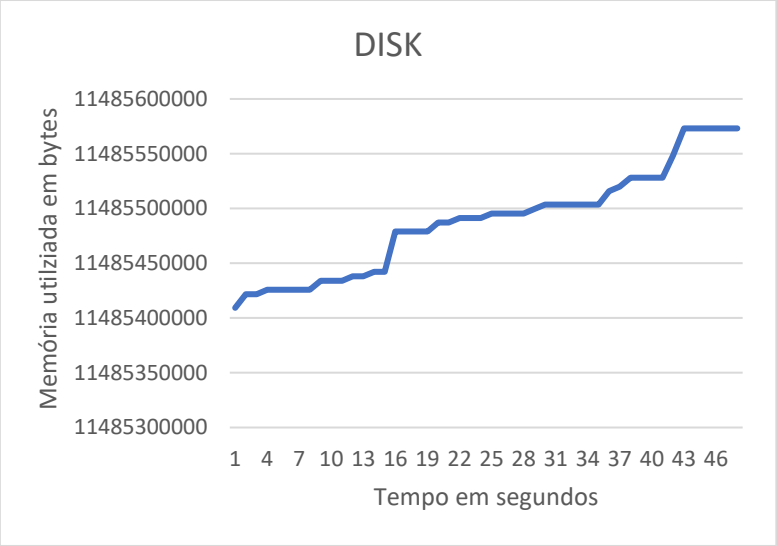

Figura A-256 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 11

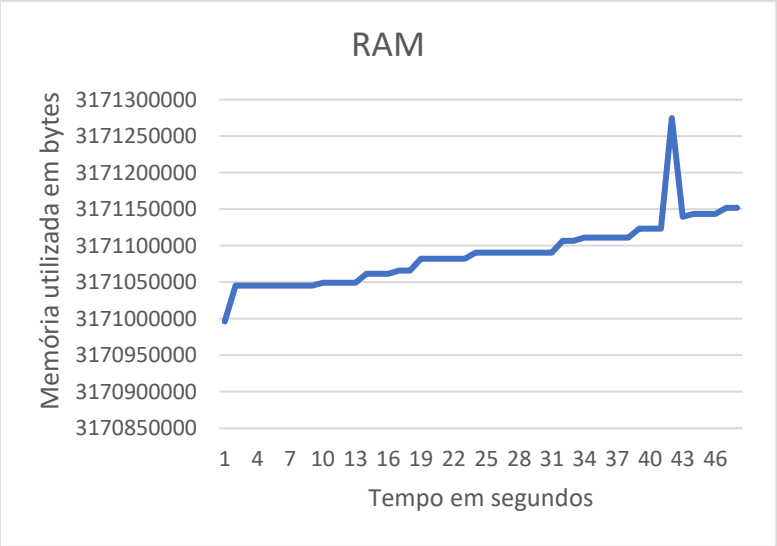

Figura A-257 - Memória em bytes que está a ser utilizada no node 1 para a repetição 3 do teste 11

Node 2 – Miner2

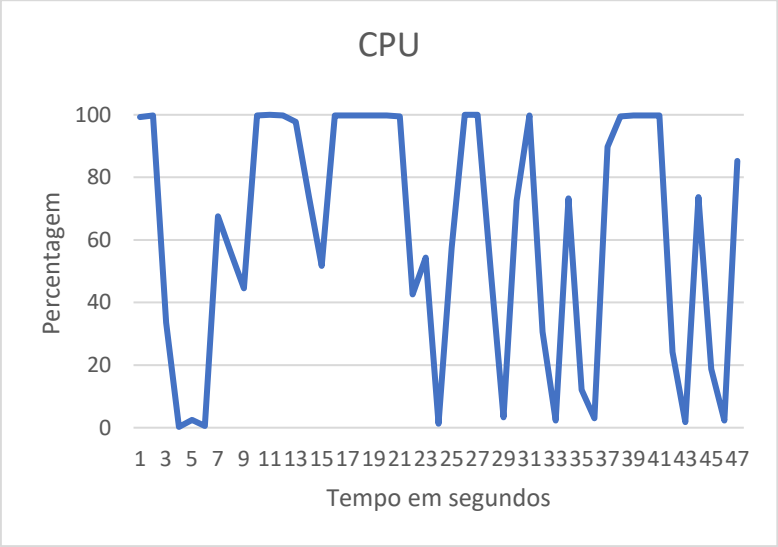

Figura A-258 - Valores de CPU do node 2 para a repetição 3 do teste 11

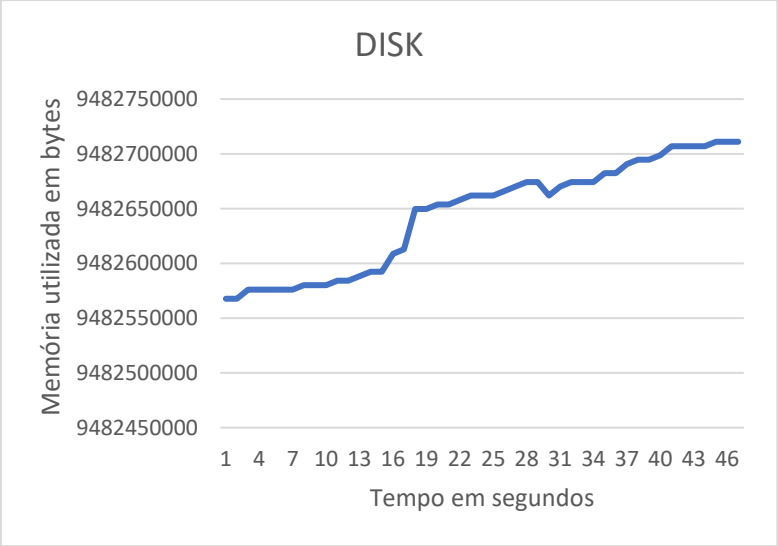

Figura A-259 - Memória utilizada de disco em bytes do node 2 para a repetição 3 do teste 11

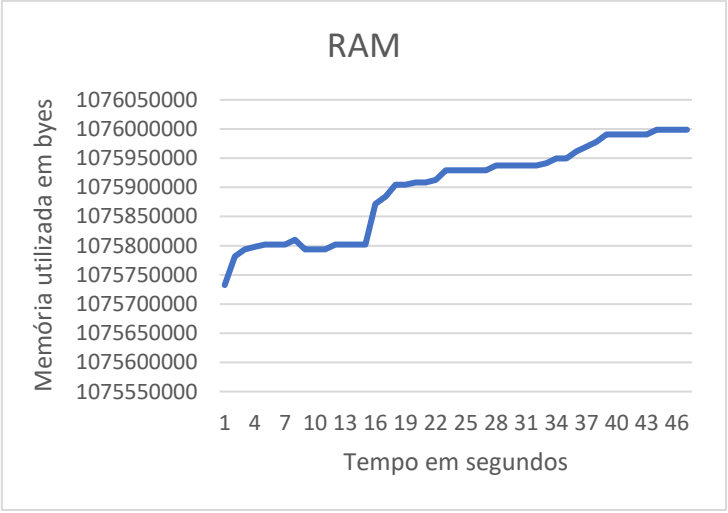

Figura A-260 - Memória em bytes que está a ser utilizada no node 2 para a repetição 3 do teste 11

Node 3

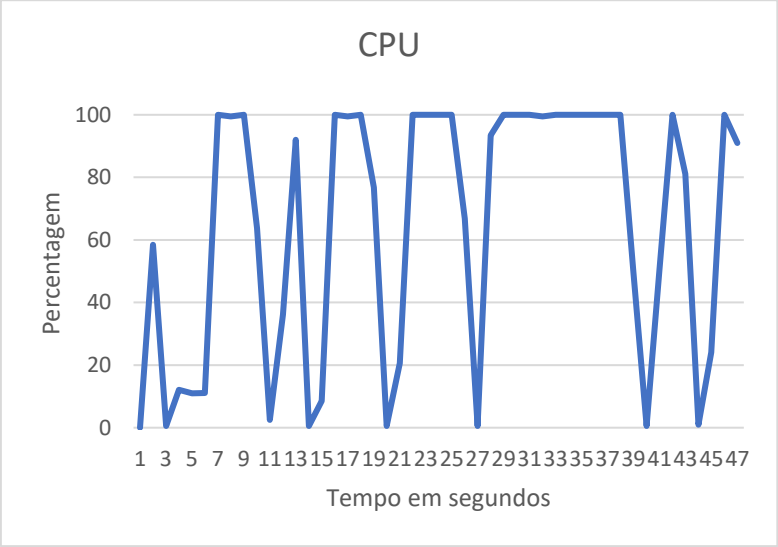

Figura A-261 - Valores de CPU do node 3 para a repetição 3 do teste 11

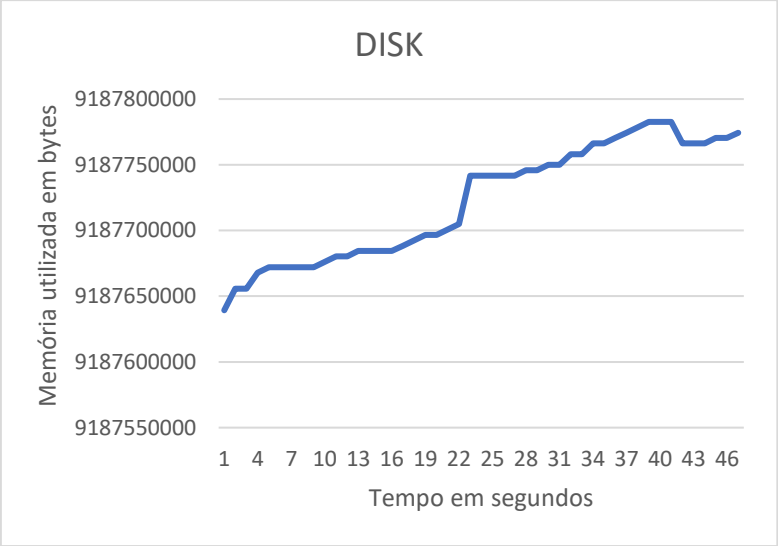

Figura A-262 - Memória utilizada de disco em bytes do node 3 para a repetição 3 do teste 11

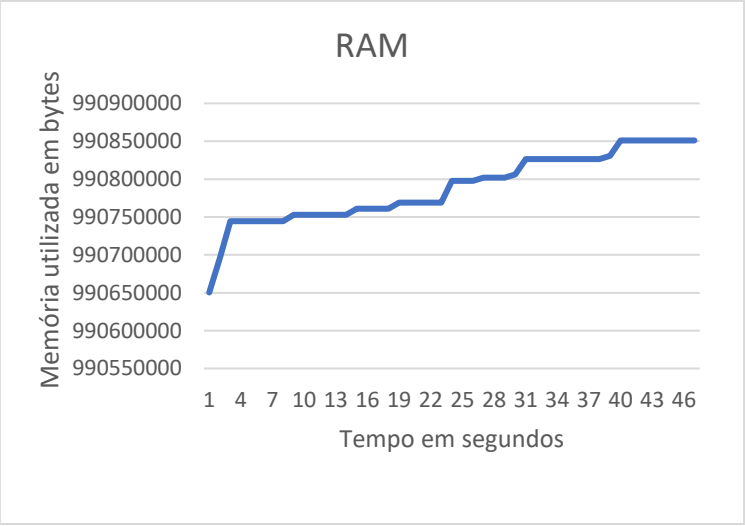

Figura A-263 - Memória em bytes que está a ser utilizada no node 3 para a repetição 3 do teste 11

Node 4

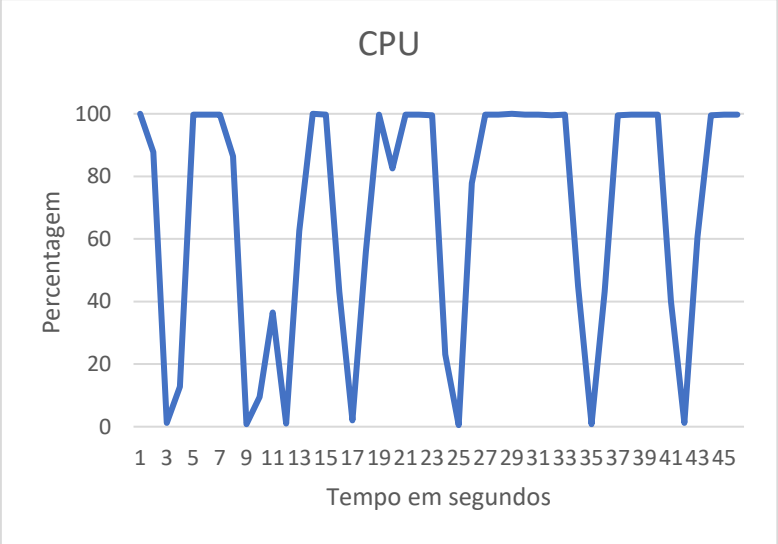

Figura A-264 - Valores de CPU do node 4 para a repetição 3 do teste 11

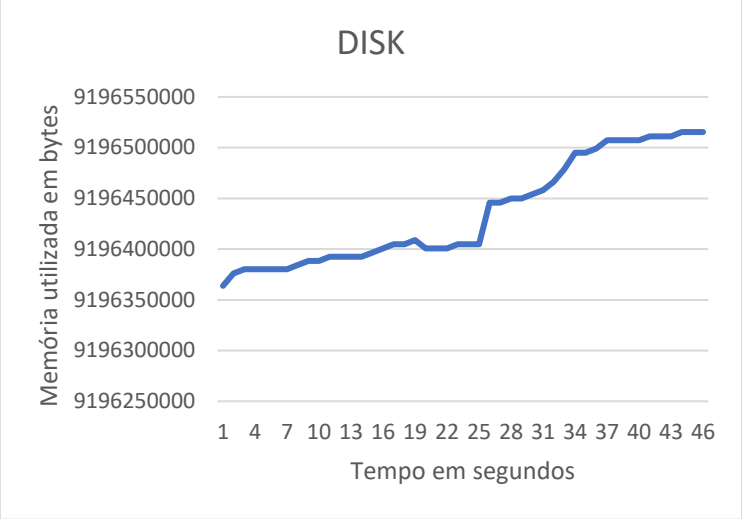

Figura A-265 - Memória utilizada de disco em bytes do node 4 para a repetição 3 do teste 11

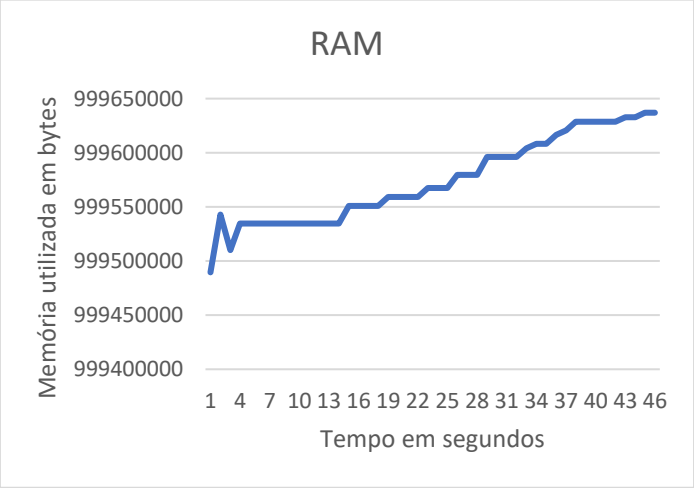

Figura A-266 - Memória em bytes que está a ser utilizada no node 4 para a repetição 3 do teste 11

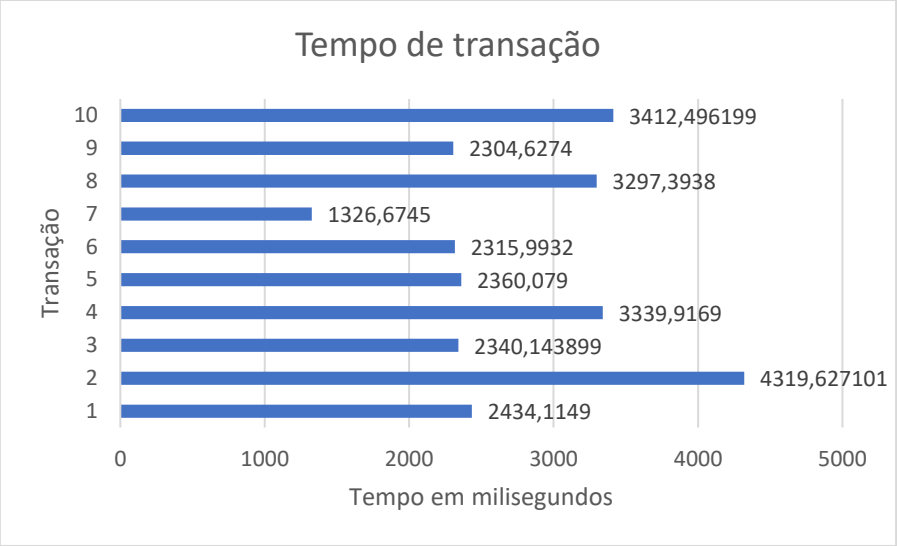

Figura A-267 - Tempo de processamento das transações realizadas na repetição 3 do teste 11

Teste 12

Repetição 1

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na primeira repetição do teste.

Node 1 – Miner

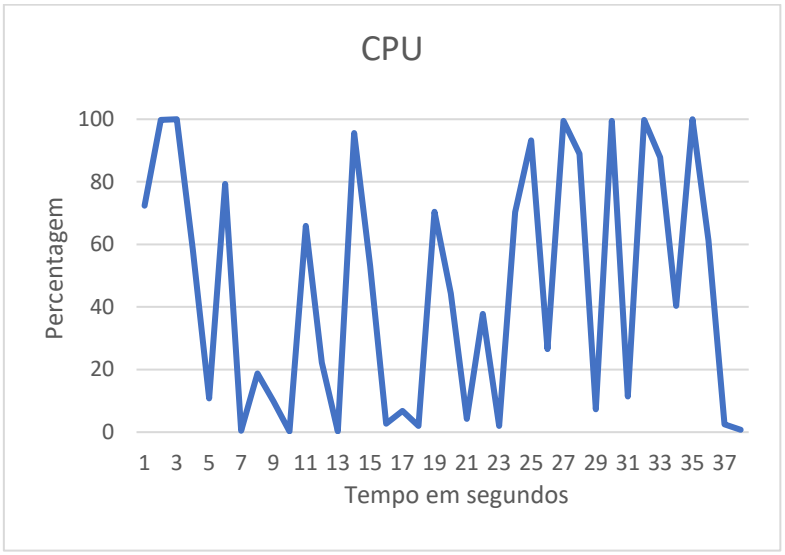

Figura A-268 - Valores de CPU do node 1 para a repetição 1 do teste 12

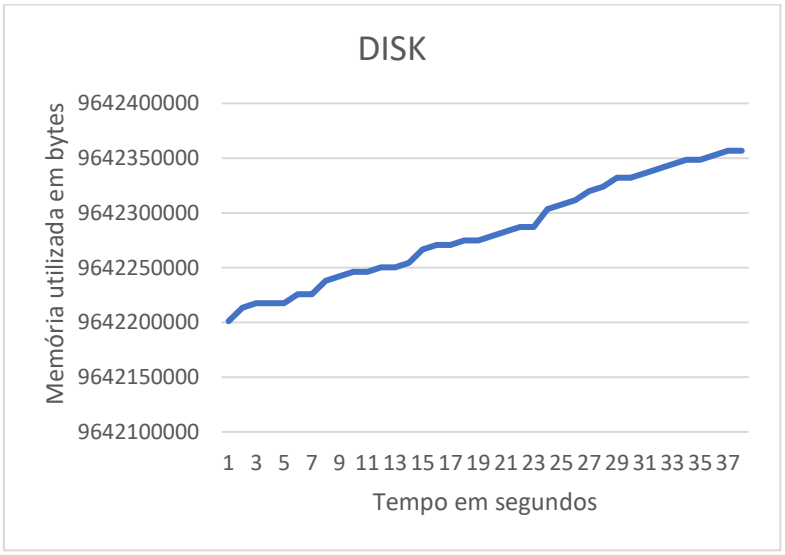

Figura A-269 - Memória utilizada de disco em bytes do node 1 para a repetição 1 do teste 12

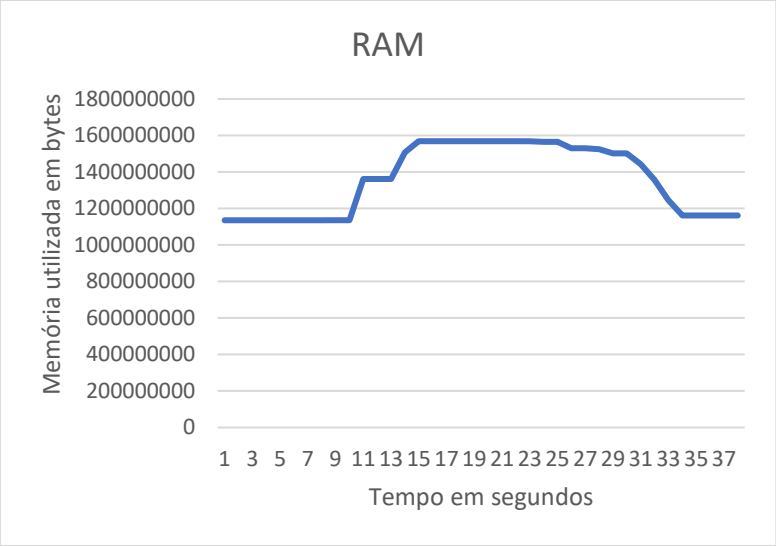

Figura A-270 - Memória em bytes que está a ser utilizada no node 1 para a repetição 1 do teste 12

Node 2 – Miner2

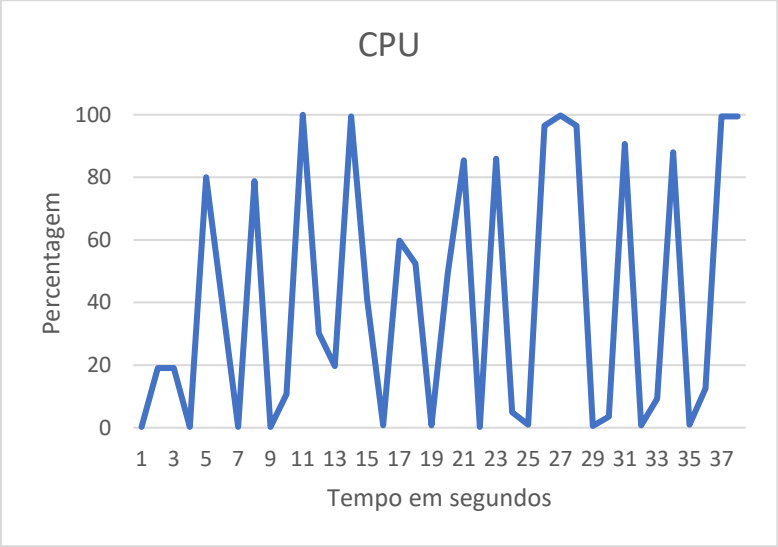

Figura A-271 - Valores de CPU do node 2 para a repetição 1 do teste 12

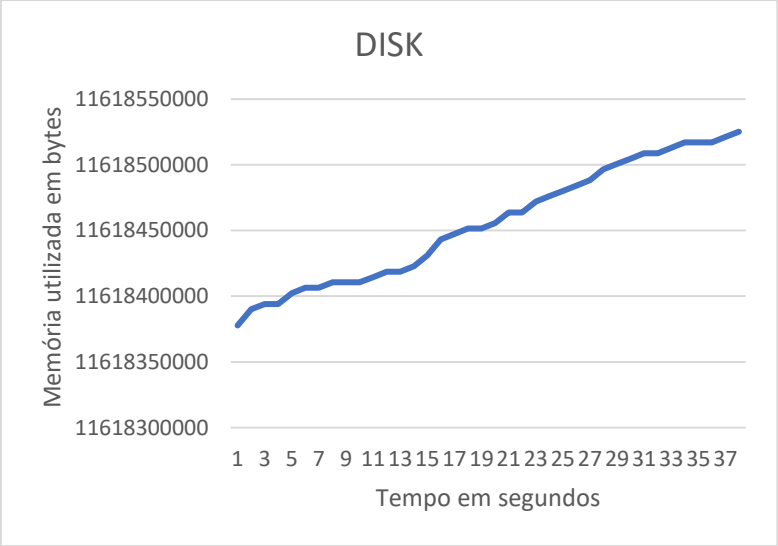

Figura A-272 - Memória utilizada de disco em bytes do node 2 para a repetição 1 do teste 12

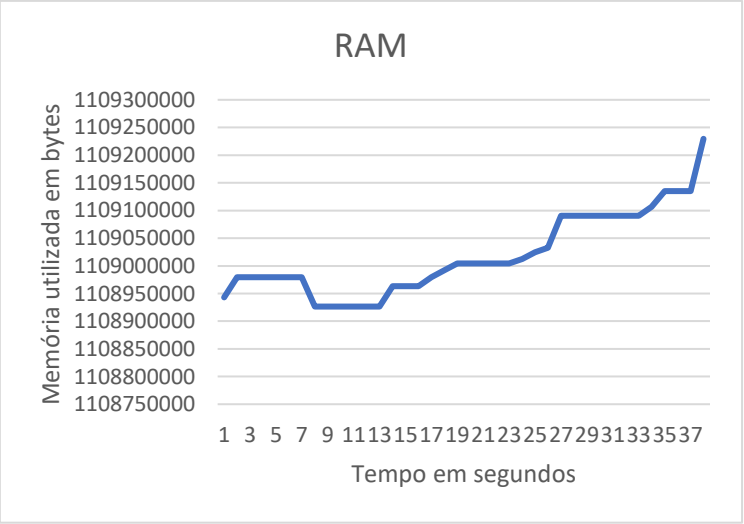

Figura A-273 - Memória em bytes que está a ser utilizada no node 2 para a repetição 1 do teste 12

Node 3 – Miner

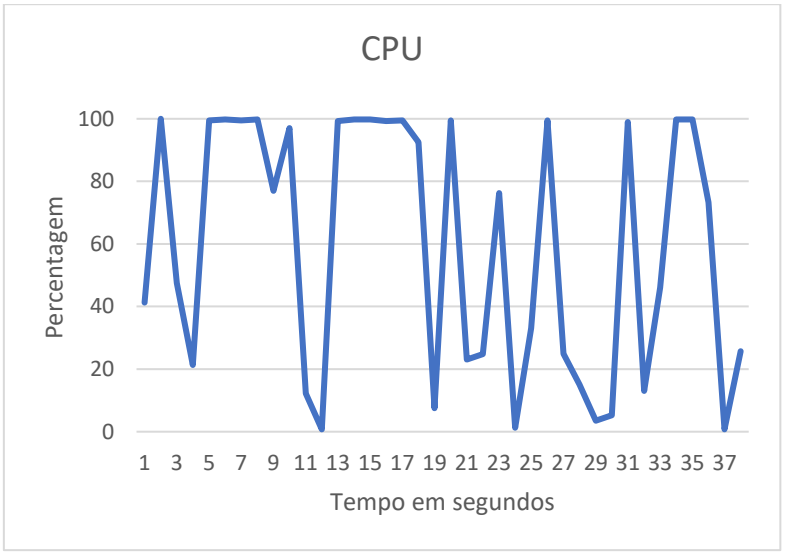

Figura A-274 - Valores de CPU do node 3 para a repetição 1 do teste 12

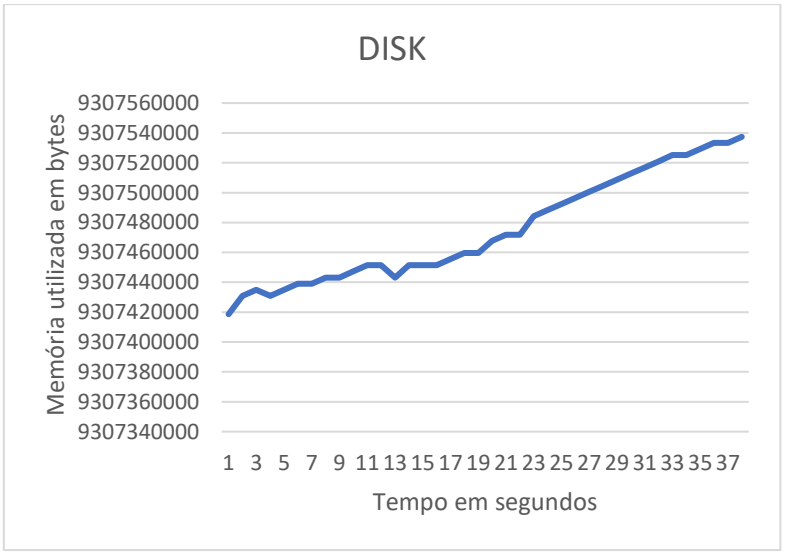

Figura A-275 - Memória utilizada de disco em bytes do node 3 para a repetição 1 do teste 12

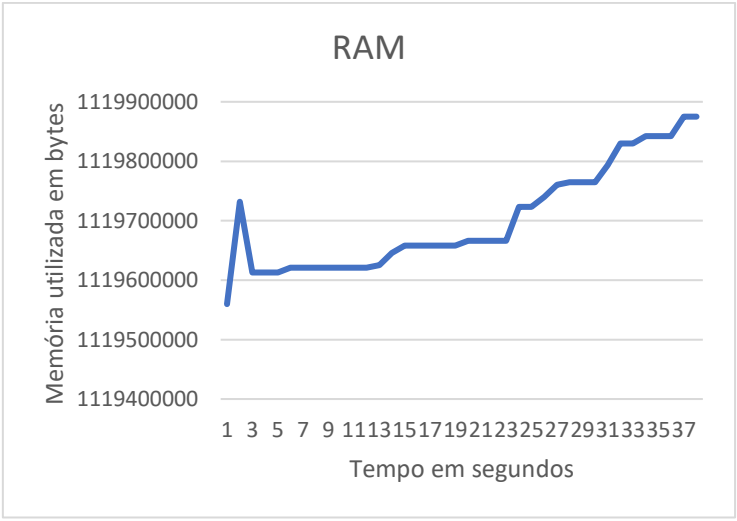

Figura A-276 - Memória em bytes que está a ser utilizada no node 3 para a repetição 1 do teste 12

Node 4 – Miner

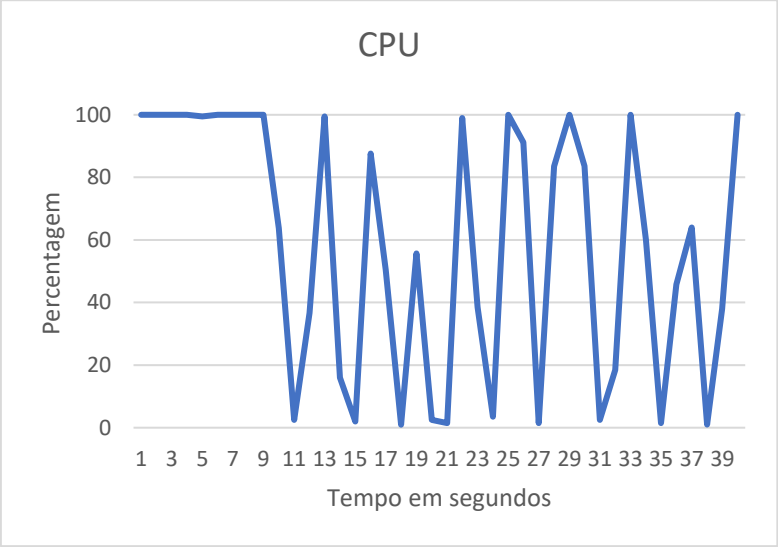

Figura A-277 - Valores de CPU do node 4 para a repetição 1 do teste 12

Figura A-278 - Memória utilizada de disco em bytes do node 4 para a repetição 1 do teste 12

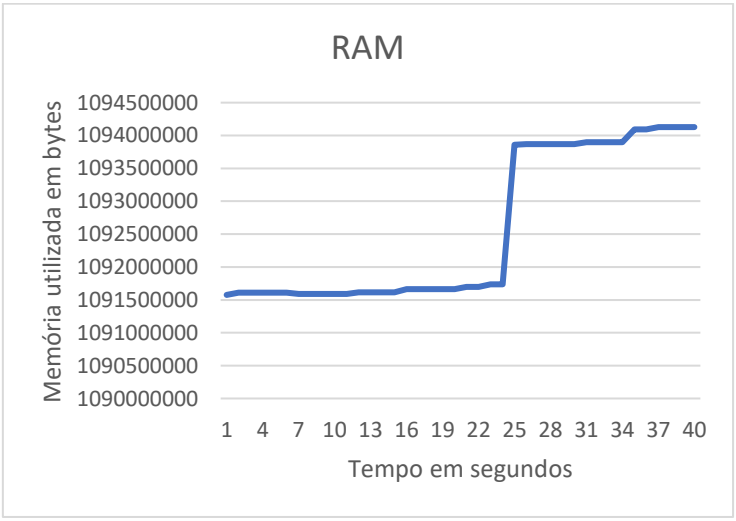

Figura A-279 - Memória em bytes que está a ser utilizada no node 4 para a repetição 1 do teste 12

Node 5 – Miner

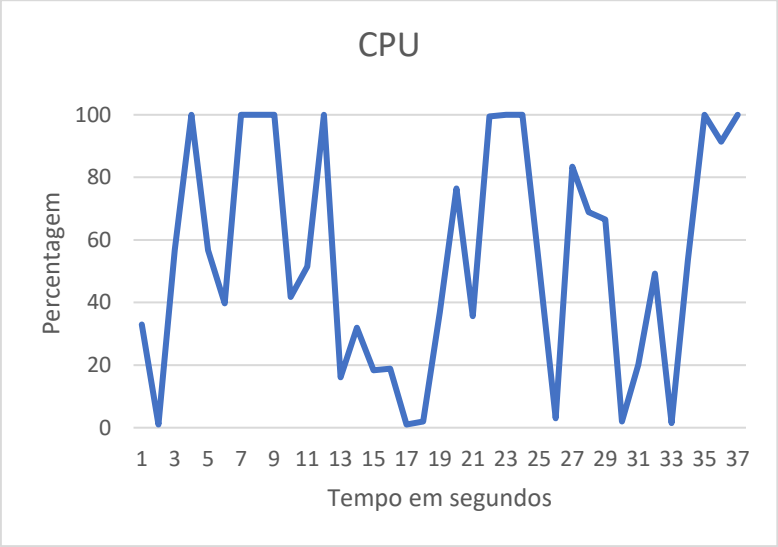

Figura A-280 - Valores de CPU do node 5 para a repetição 1 do teste 12

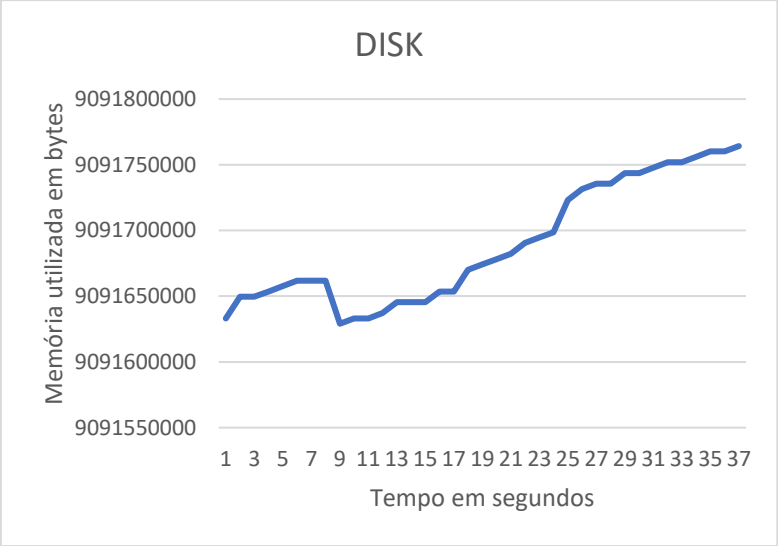

Figura A-281 - Memória utilizada de disco em bytes do node 5 para a repetição 1 do teste 12

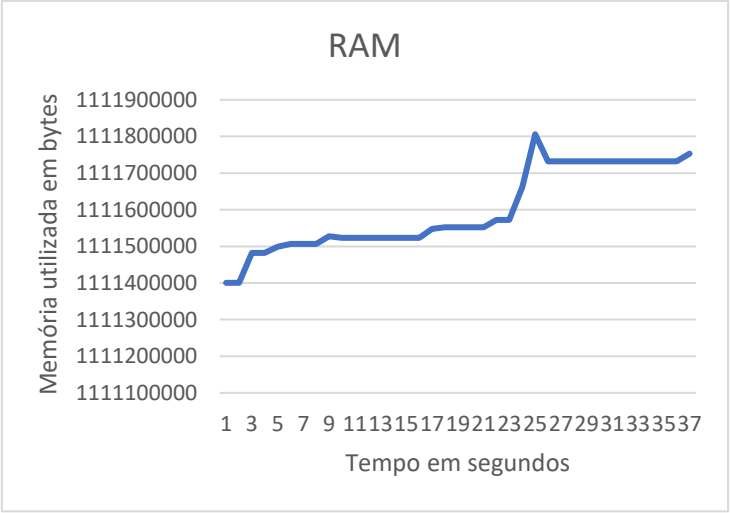

Figura A-282 - Memória em bytes que está a ser utilizada no node 5 para a repetição 1 do teste 12

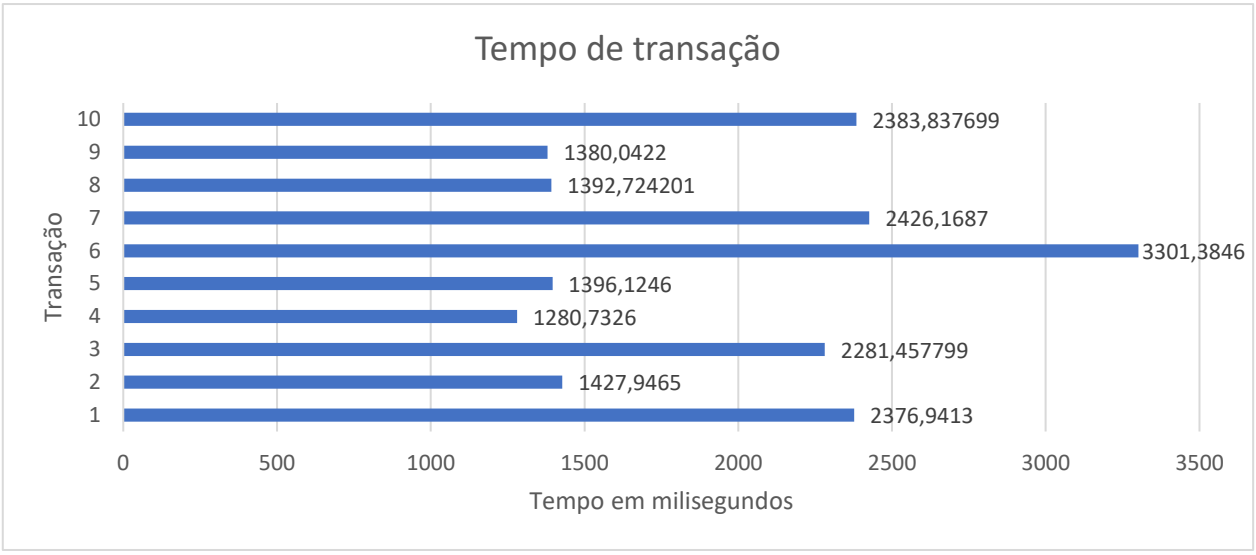

Figura A-283 - Tempo de processamento das transações realizadas na repetição 1 do teste 12

Repetição 2

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na segunda repetição do teste.

Node 1 – Miner

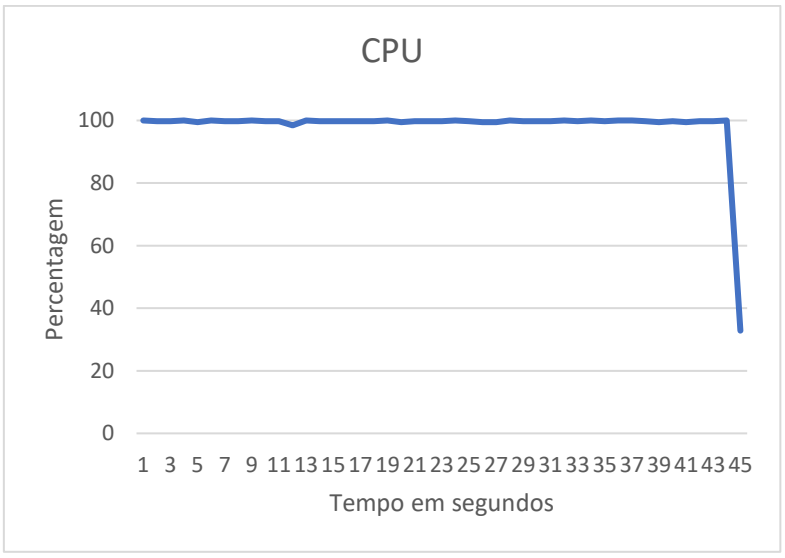

Figura A-284 - Valores de CPU do node 1 para a repetição 2 do teste 12

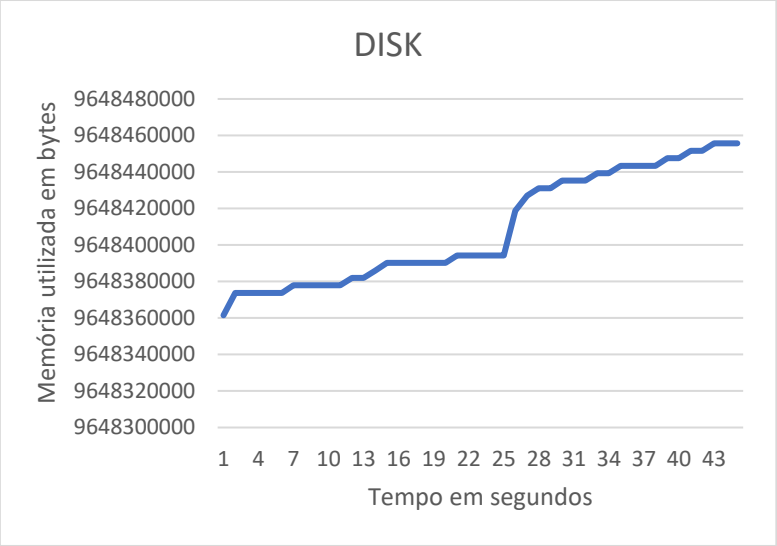

Figura A-285 - Memória utilizada de disco em bytes do node 1 para a repetição 2 do teste 12

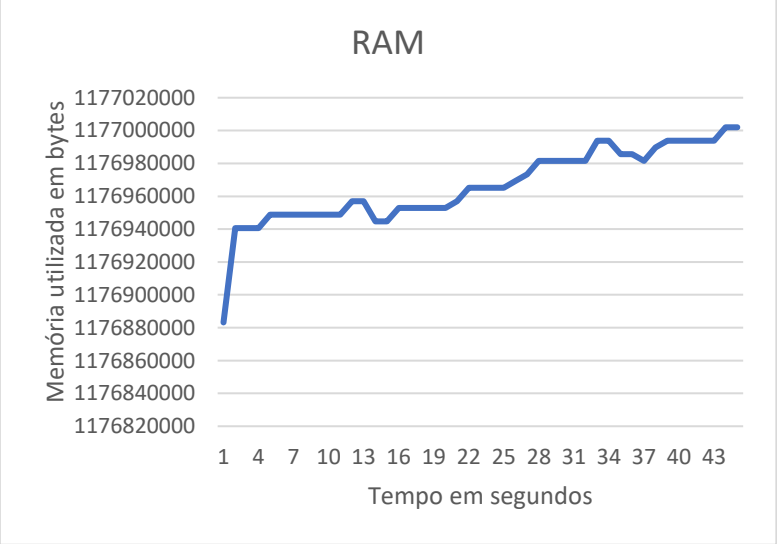

Figura A-286 - Memória em bytes que está a ser utilizada no node 1 para a repetição 2 do teste 12

Node 2 – Miner

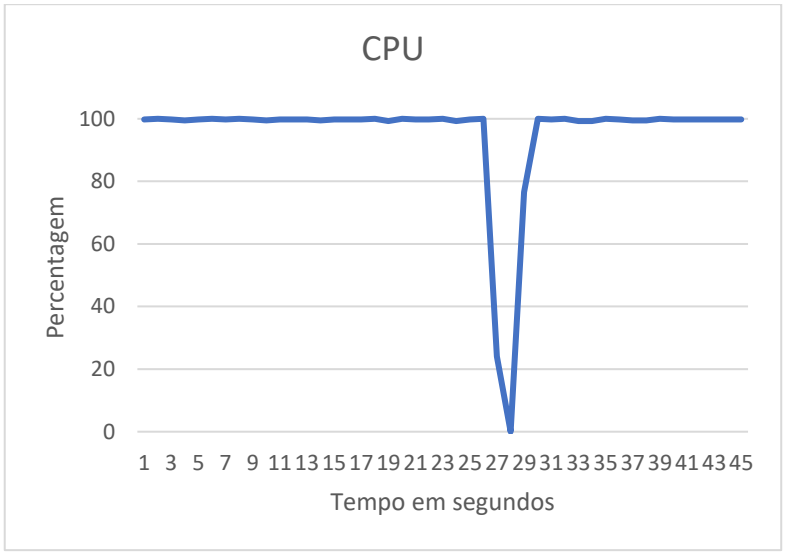

Figura A-287 - Valores de CPU do node 2 para a repetição 2 do teste 12

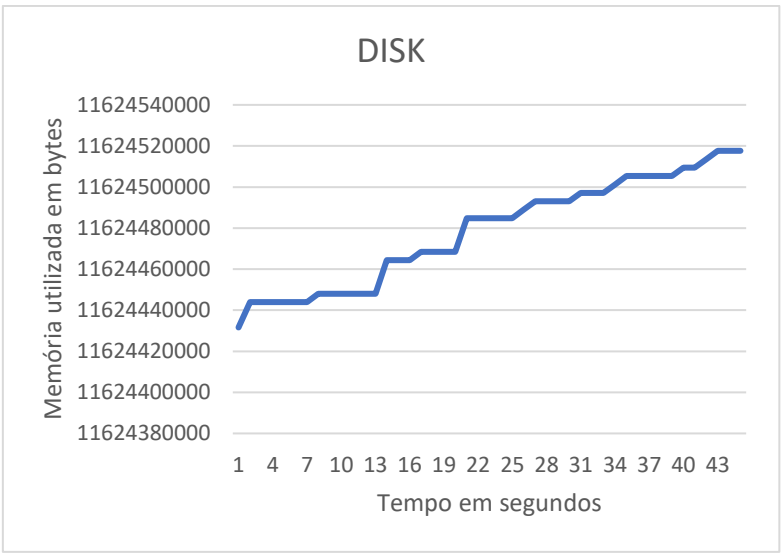

Figura A-288 - Memória utilizada de disco em bytes do node 2 para a repetição 2 do teste 12

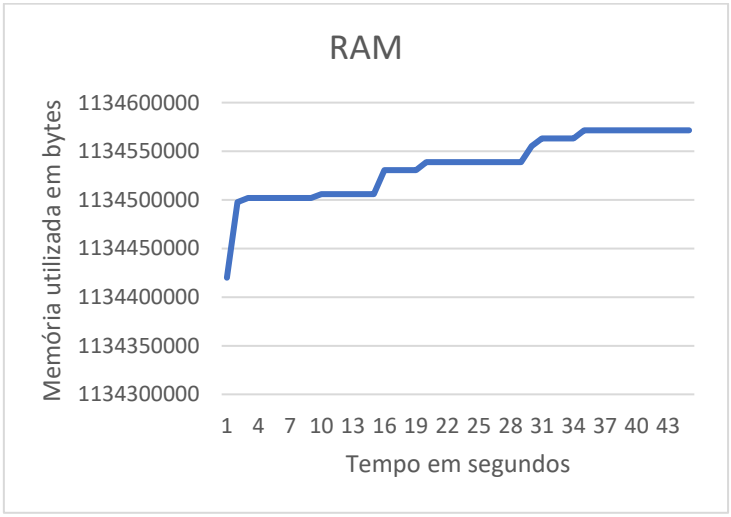

Figura A-289 - Memória em bytes que está a ser utilizada no node 2 para a repetição 2 do teste 12

Node 3 – Miner

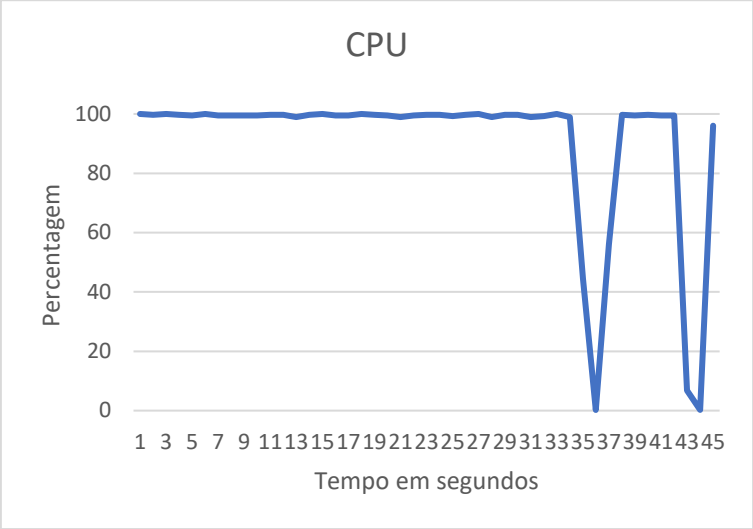

Figura A-290 - Valores de CPU do node 3 para a repetição 2 do teste 12

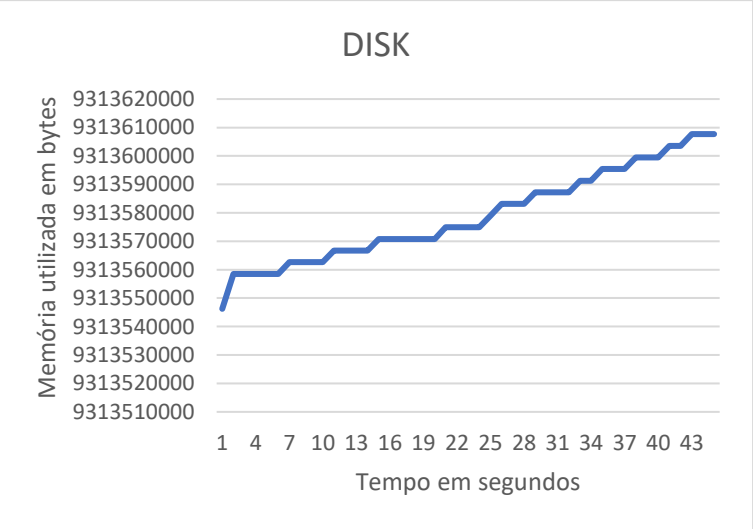

Figura A-291 - Memória utilizada de disco em bytes do node 3 para a repetição 2 do teste 12

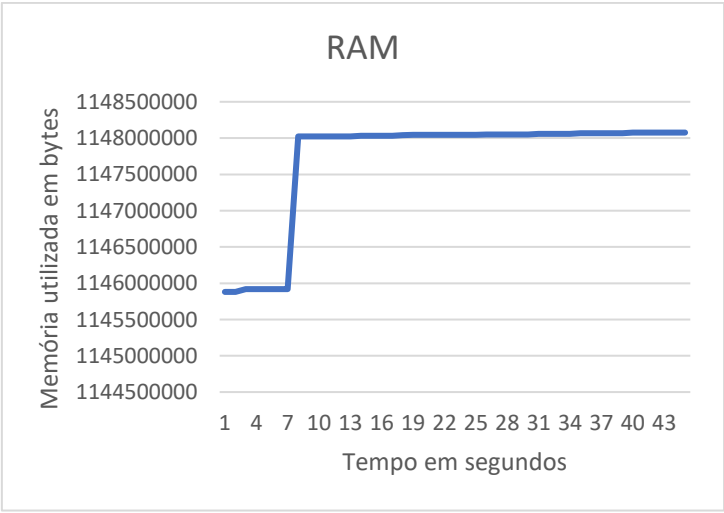

Figura A-292 - Memória em bytes que está a ser utilizada no node 3 para a repetição 2 do teste 12

Node 4 – Miner

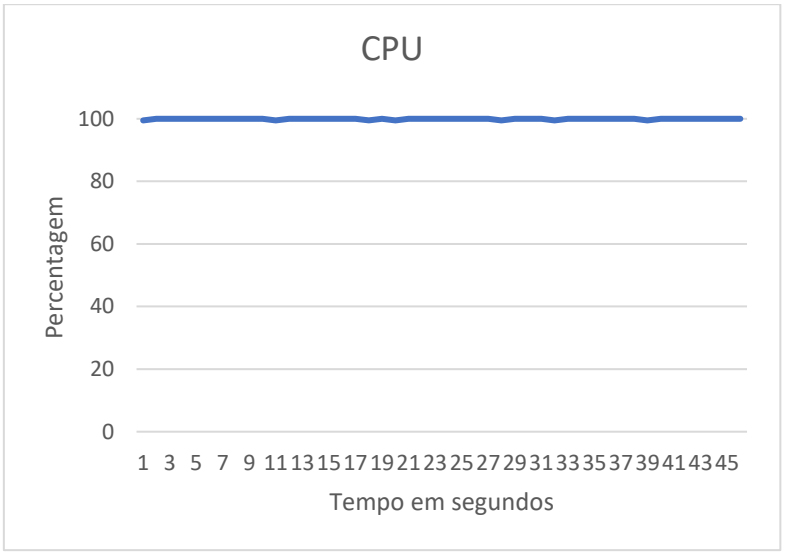

Figura A-293 - Valores de CPU do node 4 para a repetição 2 do teste 12

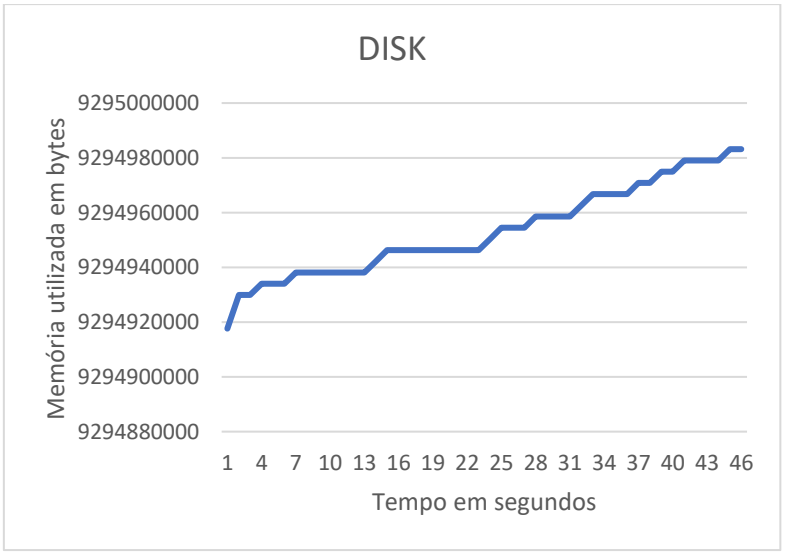

Figura A-294 - Memória utilizada de disco em bytes do node 4 para a repetição 2 do teste 12

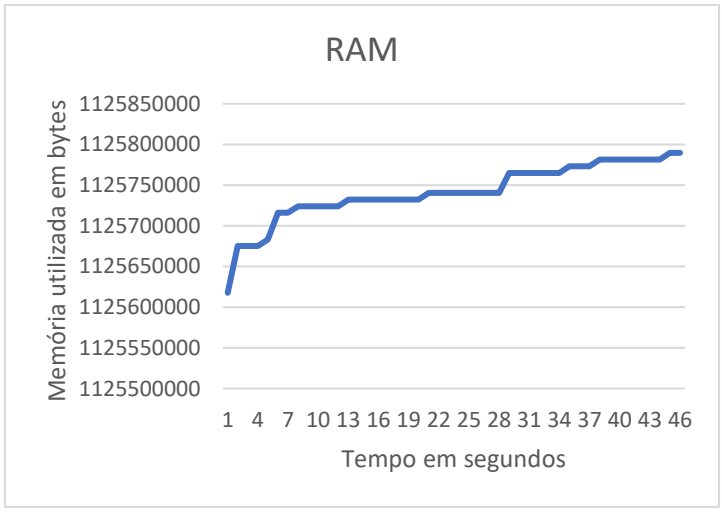

Figura A-295 - Memória em bytes que está a ser utilizada no node 4 para a repetição 2 do teste 12

Node 5 – Miner

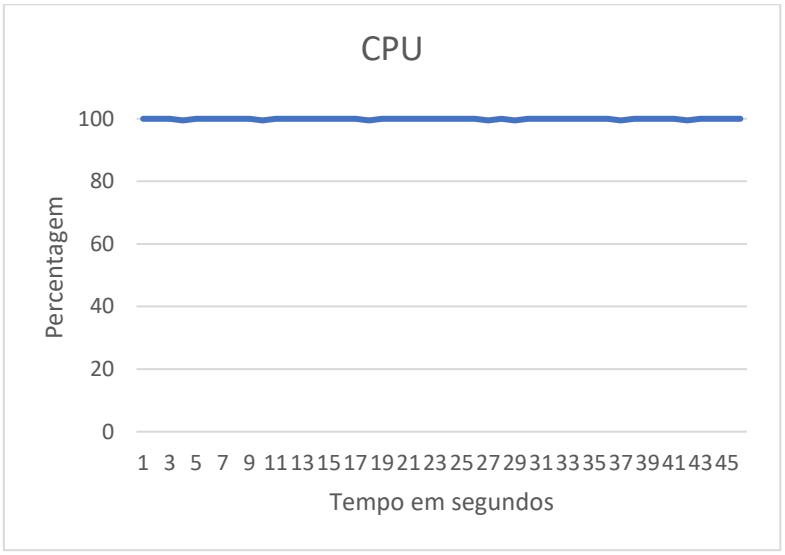

Figura A-296 - Valores de CPU do node 5 para a repetição 2 do teste 12

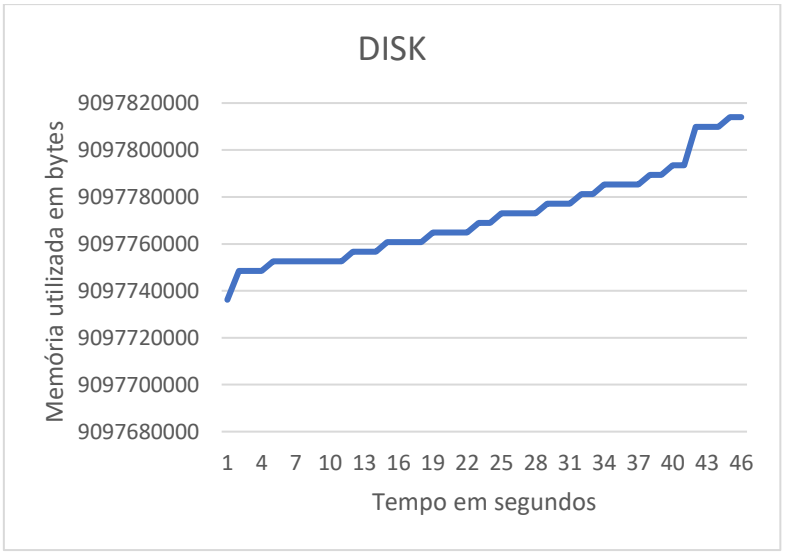

Figura A-297 - Memória utilizada de disco em bytes do node 5 para a repetição 2 do teste 12

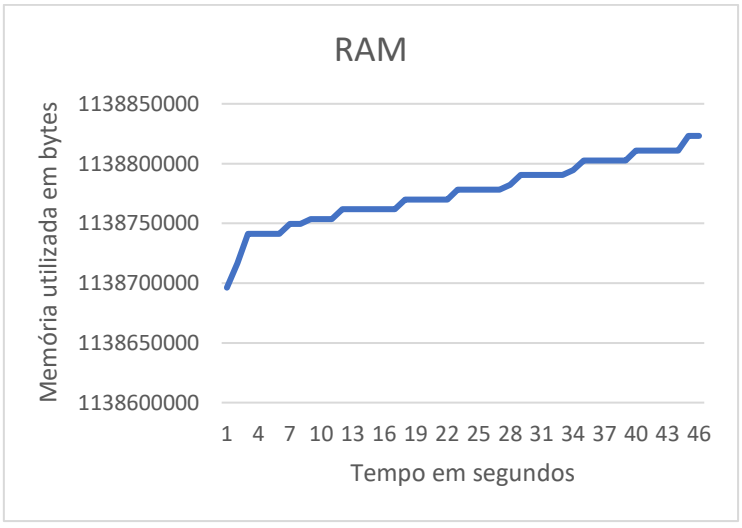

Figura A-298 - Memória em bytes que está a ser utilizada no node 5 para a repetição 2 do teste 12

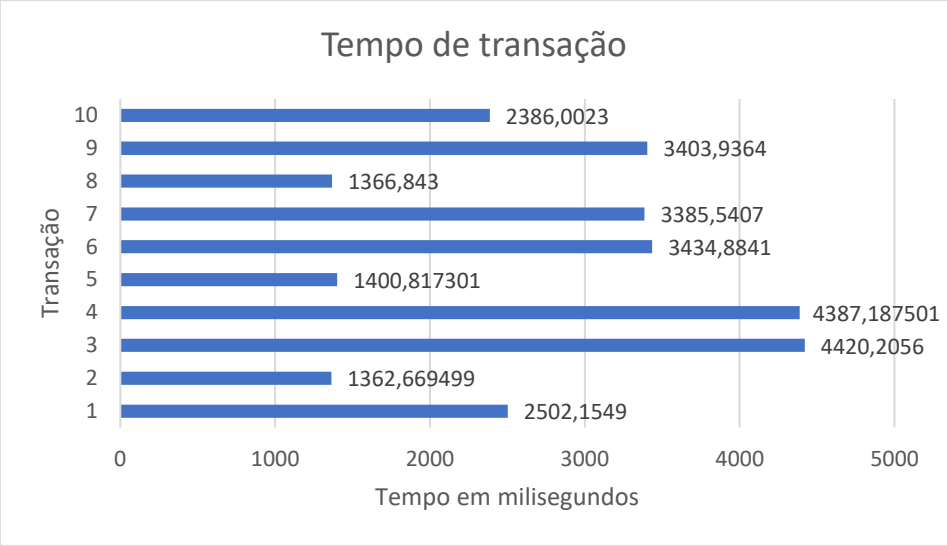

Figura A-299 - Tempo de processamento das transações realizadas na repetição 2 do teste 12

Repetição 3

Nesta secção são apresentados os resultados de cada um dos nós da rede Blockchain na terceira repetição do teste.

Node 1 – Miner

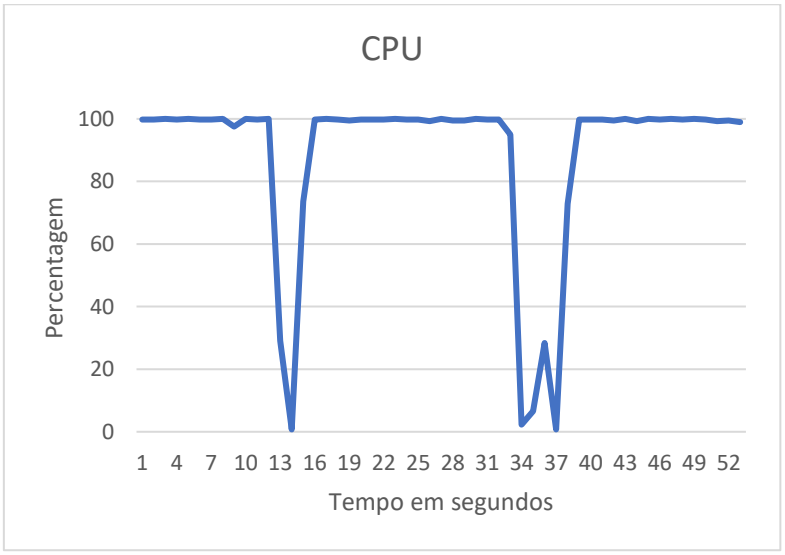

Figura A-300 - Valores de CPU do node 1 para a repetição 3 do teste 12

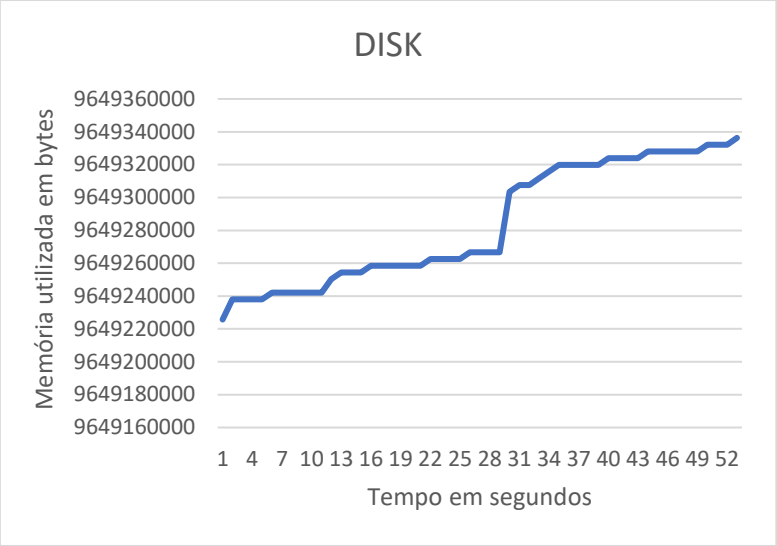

Figura A-301 - Memória utilizada de disco em bytes do node 1 para a repetição 3 do teste 12

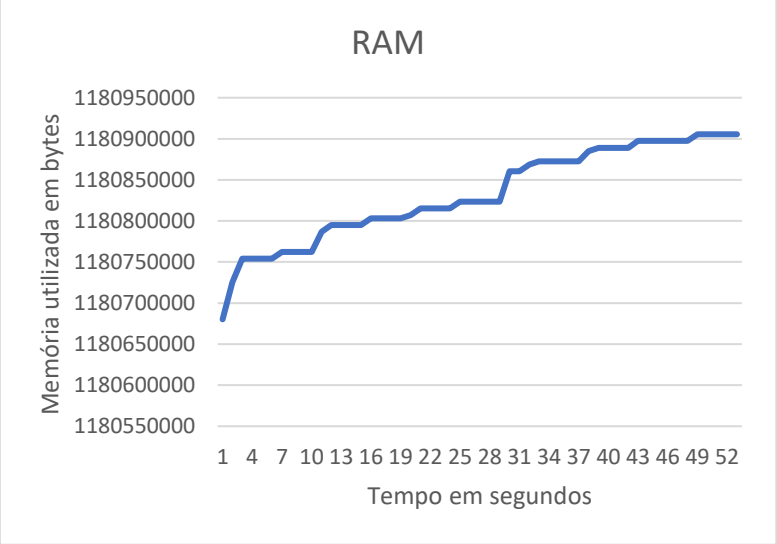

Figura A-302 - Memória em bytes que está a ser utilizada no node 1 para a repetição 3 do teste 12

Node 2 – Miner

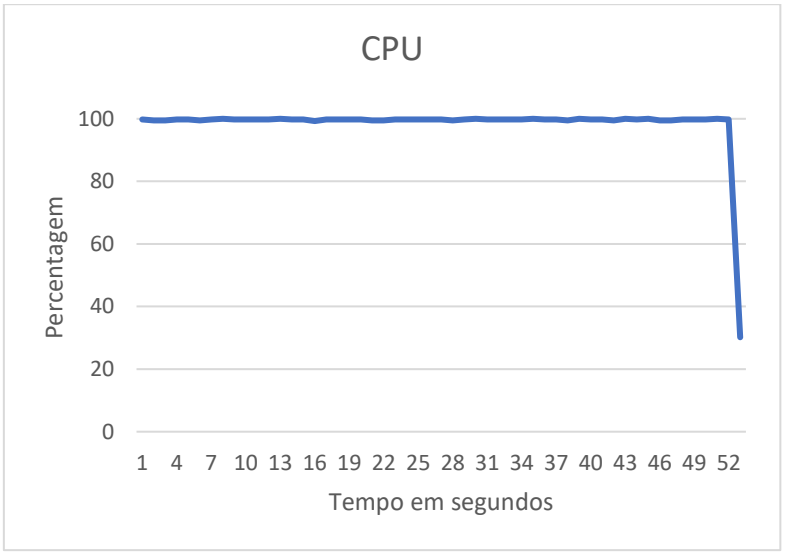

Figura A-303 - Valores de CPU do node 2 para a repetição 3 do teste 12

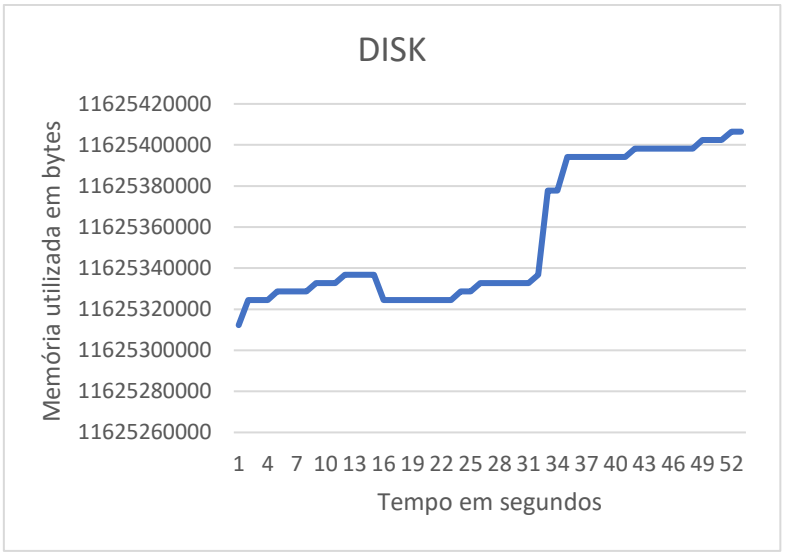

Figura A-304 - Memória utilizada de disco em bytes do node 2 para a repetição 3 do teste 12

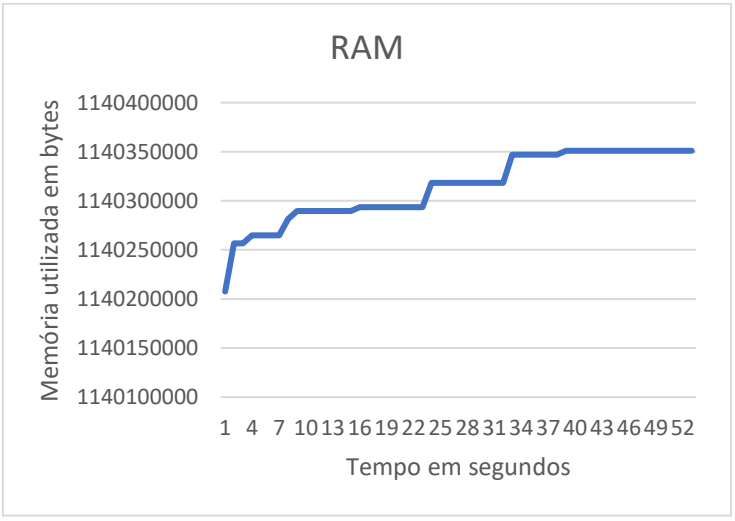

Figura A-305 - Memória em bytes que está a ser utilizada no node 2 para a repetição 3 do teste 12

Node 3 – Miner

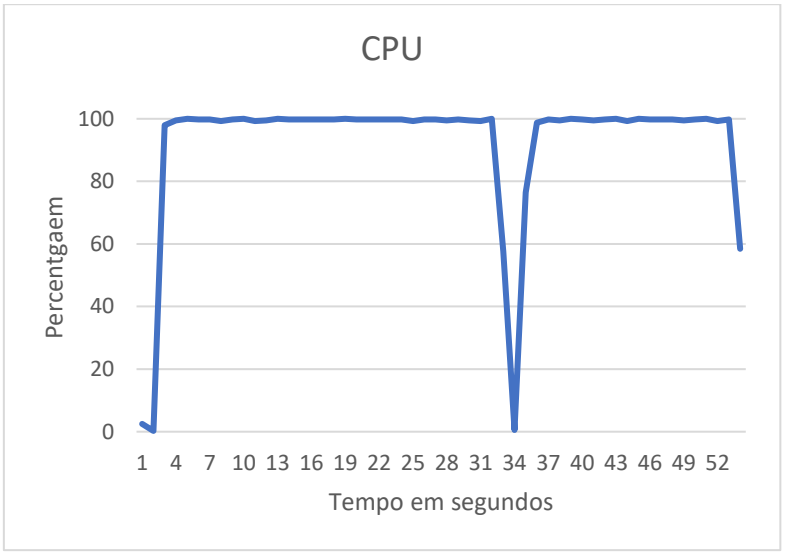

Figura A-306 - Valores de CPU do node 3 para a repetição 3 do teste 12

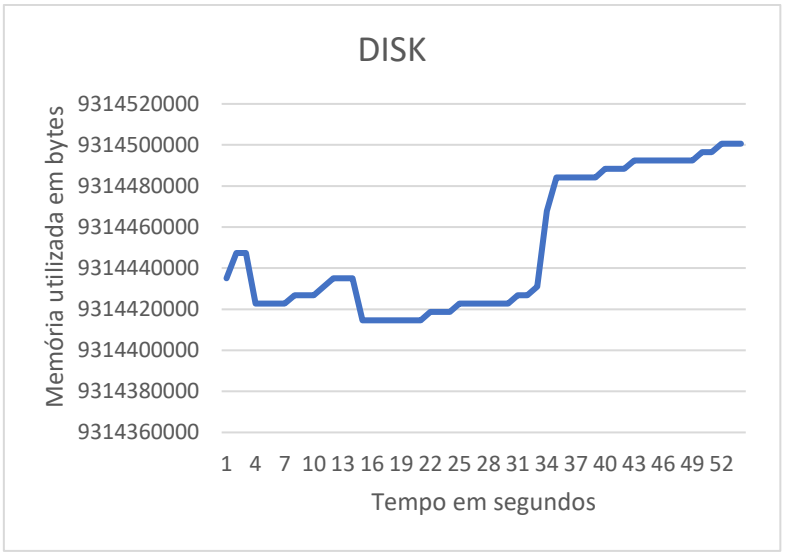

Figura A-307 - Memória utilizada de disco em bytes do node 3 para a repetição 3 do teste 12

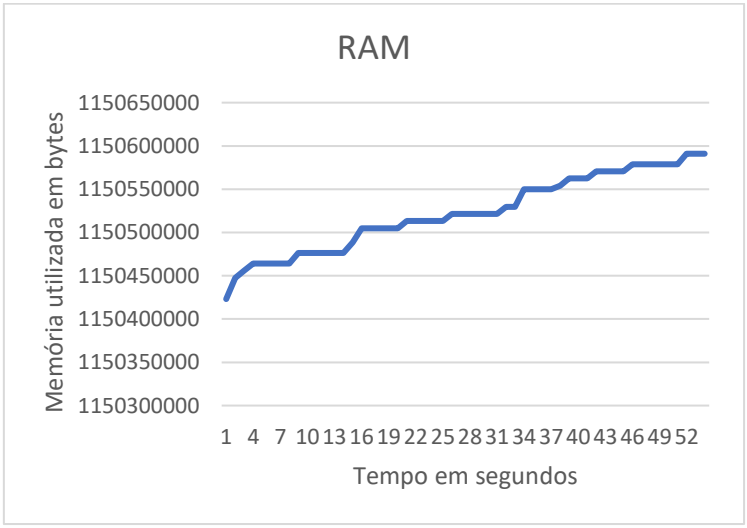

Figura A-308 - Memória em bytes que está a ser utilizada no node 3 para a repetição 3 do teste 12

Node 4 – Miner

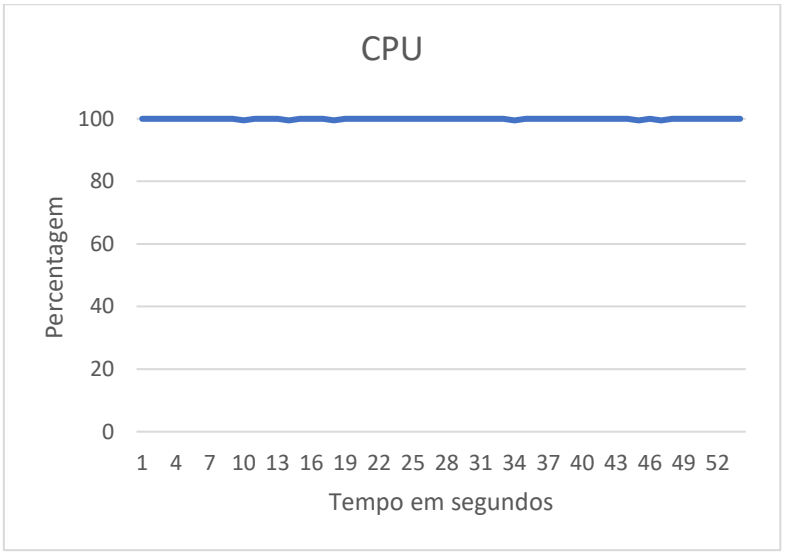

Figura A-309 - Valores de CPU do node 4 para a repetição 3 do teste 12

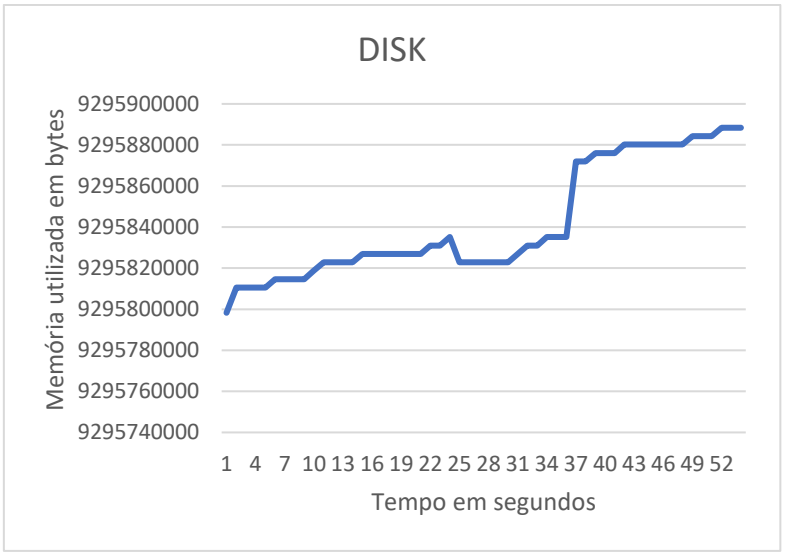

Figura A-310 - Memória utilizada de disco em bytes do node 4 para a repetição 3 do teste 12

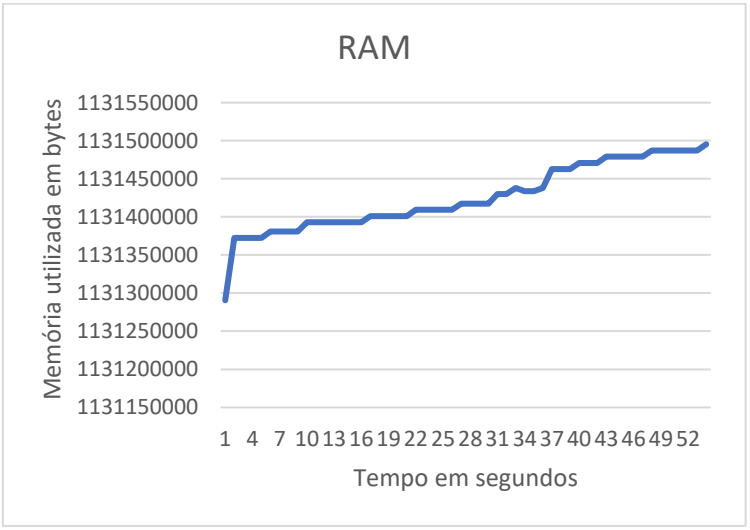

Figura A-311 - Memória em bytes que está a ser utilizada no node 4 para a repetição 3 do teste 12

Node 5 – Miner

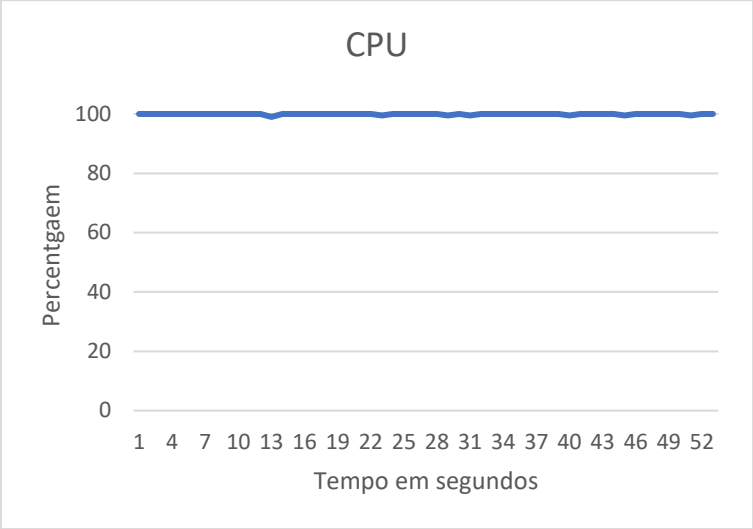

Figura A-312 - Valores de CPU do node 5 para a repetição 3 do teste 12

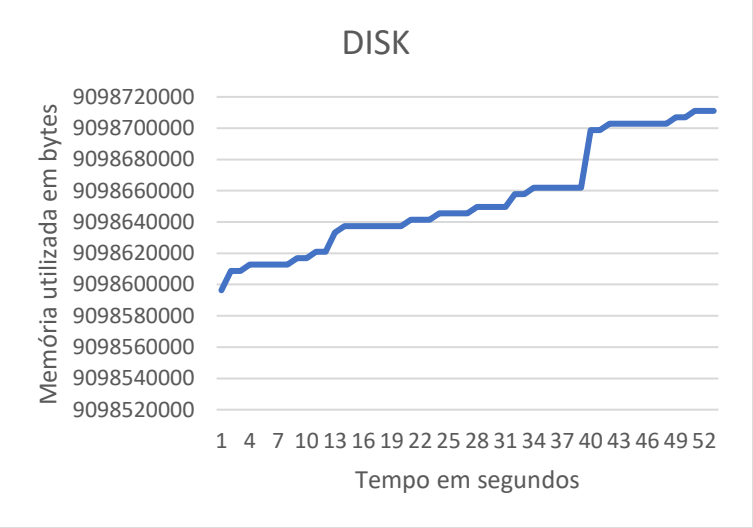

Figura A-313 - Memória utilizada de disco em bytes do node 5 para a repetição 3 do teste 12

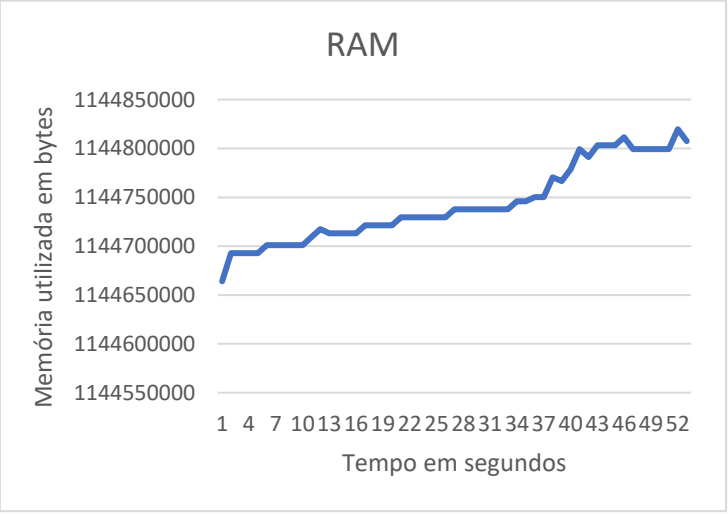

Figura A-314 - Memória em bytes que está a ser utilizada no node 5 para a repetição 3 do teste 12

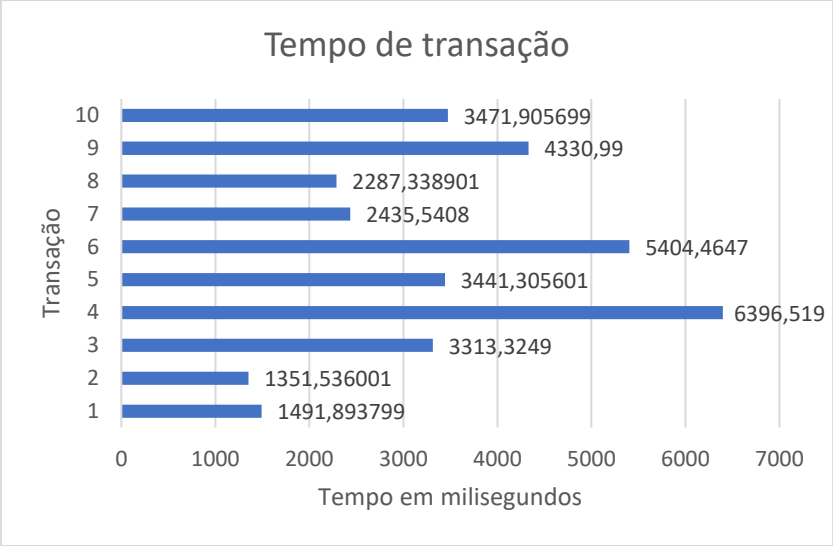

Figura A-315 - Tempo de processamento das transações realizadas na repetição 3 do teste 12