UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS QUIXADÁ

CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

NATAN NOBRE CHAVES

UMA ARQUITETURA BASEADA EM INTERNET DAS COISAS APLICADA À AGRICULTURA INTELIGENTE

QUIXADÁ

2022

NATAN NOBRE CHAVES

UMA ARQUITETURA BASEADA EM INTERNET DAS COISAS APLICADA À AGRICULTURA INTELIGENTE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Cristiano Bacelar de Oliveira

QUIXADÁ

2022

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

C439a Chaves, Natan Nobre.
Uma arquitetura baseada em internet das coisas aplicada à agricultura inteligente / Natan Nobre Chaves. – 2022.
42 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Computação, Quixadá, 2022.
Orientação: Prof. Dr. Cristiano Bacelar de Oliveira.

1. Informática agrícola. 2. Internet das coisas. 3. Blockchains (Base de dados). I. Título.
CDD 621.39

NATAN NOBRE CHAVES

UMA ARQUITETURA BASEADA EM INTERNET DAS COISAS APLICADA À AGRICULTURA INTELIGENTE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: ____/_____/_____

BANCA EXAMINADORA

Prof. Dr. Cristiano Bacelar de Oliveira (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. André Ribeiro Braga
Universidade Federal do Ceará (UFC)

Prof. Dr. Thiago Werlley Bandeira da Silva
Universidade Federal do Ceará (UFC)

Soli Deo Gloria.

AGRADECIMENTOS

À Deus, por ter me sustentado e provido todos os meios para que este trabalho pudesse ser escrito, me capacitando e direcionando sempre.

À minha esposa, Kathrine, por ser uma mulher bíblica, me apoiando e cuidando de nossa família com amor e carinho. Sempre esteve ao meu lado e me fortaleceu quando precisei, me ajudando a ser o homem que sou hoje.

Aos meus pais, Wilton e Maria do Carmo por todo o esforço, carinho, amor e apoio em cada etapa, me mantendo motivado e preparado para ser um homem responsável e correto na vida cotidiana.

Aos amigos da graduação que me ajudaram nesta jornada, compartilhando aprendizados e experiências. Em especial aos membros da Resistência E.C.

Ao professor Cristiano, pelas orientações e auxílios para a produção deste trabalho.

À toda comunidade acadêmica da Universidade Federal do Ceará por todo apoio oferecido para o meu desenvolvimento pessoal, acadêmico e profissional.

"O certo é certo, mesmo que ninguém o faça. O errado é errado, mesmo que todos se enganem sobre ele."

(G. K. Chesterton)

RESUMO

A agricultura é uma base econômica de diversos países, incluindo o Brasil. Avanços tecnológicos nesse setor acontecem diariamente, colaborando para uma agricultura mais moderna, eficiente e sustentável. Esses avanços também têm impactado aspectos importantes, tais como garantir a transparência, confiança e a segurança das informações. Com o uso de tecnologias como *IoT* e *blockchain*, o setor agrícola tem passado a usufruir da interconexão entre dispositivos de monitoramento e de atuação no campo, bem como do aspectos de confiança no armazenamento de dados. Com o crescente número de estudos e pesquisa na área, diversos benefícios podem surgir para a agricultura inteligente e para a academia. Para isso, foi desenvolvida uma arquitetura, integrando as tecnologias *IoT* e *Blockchain*. Essa arquitetura realiza o monitoramento de dados em fazendas, campos ou estufas, baseando-se nas técnicas de agricultura inteligente. As informações podem ser acessadas em qualquer local que possui acesso à internet, e o armazenamento é feito de maneira descentralizada para garantir a integridade e confiança das informações. Uma aplicação *WEB* também foi desenvolvida para permitir o acesso e o gerenciamento de usuários ao sistema. Experimentos foram realizados com a arquitetura para validar seu comportamento em ambientes com uma maior carga de processamento. Com a análise dos resultados obtidos, foi possível validar a solução como escalável e capaz de suportar uma aplicação em campo.

Palavras-chave: agricultura inteligente; internet das coisas; blockchain.

ABSTRACT

Agriculture is an economic base of several countries, including Brazil. Technological advances in this sector happen daily, thus collaborating to a more modern, efficient and sustainable agriculture. These advances have also impacted important aspects of this sector, particularly in those that ensure transparency, reliability and information security. Upon the usage of technologies such as IoT and Blockchain, the agricultural sector began to take advantage of the interconnection between monitoring and actuating devices, as well as aspects of reliability in data storage. Along with the growing number of studies and research in this area, several benefits can arise for smart farming and academia. For this purpose, an architecture was developed, integrating IoT and Blockchain technologies. This platform performs data monitoring on farms, fields or greenhouses based on smart farming techniques. Furthermore, the information can be accessed from any location with internet connection. The storage is designed in a decentralized manner to ensure the integrity and reliability of the data. Additionally, a web application was also developed to allow users to access and manage the system. Experiments performed on the platform validated its behavior in environments with higher processing loads. The analysis of the obtained results certified the solution as scalable and apt to support a real-world application.

Keywords: smart farming; internet of things; blockchain.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

IoT	*Internet Of Things*
REST	*Representational State Transfer*
MVC	*Model-View-Controller*
SenSE	*Sensor Simulation Environment*

SUMÁRIO

1 INTRODUÇÃO

O setor agrícola é um dos setores mais importantes em uma nação, principalmente em um país como o Brasil, onde a agricultura é uma das principais bases econômicas. Com o avanço tecnológico nesse setor, tiveram-se também problemas graves, tais como a quantidade de agrotóxicos utilizados para alterar as características originais do que é plantado. Outro problema também é manter o acompanhamento de algumas métricas importantes para um plantio, principalmente em fazendas extensas, como a humidade do solo.

Nas últimas décadas, têm-se acompanhado um crescimento exponencial do uso de soluções tecnológicas em diversas áreas e na agricultura. Pesquisas e melhorias estão sendo realizadas diariamente para garantir a transparência, confiança e a segurança. Então, passa-se a ver mais comumente o uso de tecnologias, tais como *blockchain*, que auxiliam para garantir que os aspectos citados anteriormente estejam presentes e sejam percebidos pelo consumidor (BERMEO-ALMEIDA *et al.*, 2018).

O conceito de *Internet Of Things* (*IoT*), ou Internet das coisas, refere-se à capacidade de interconexão de aparelhos digitais em uma rede. O uso de tecnologias *IoT* tem crescido consideravelmente nos últimos anos. Pesquisas são feitas para ajudar à solucionar o problema de monitoramento dos plantios, que podem ir desde fazendas até jardins e estufas, criando um ambiente inteligente, onde os sensores são capazes de ler dados de forma independente, no ambiente em que são instalados (XIE *et al.*, 2017).

Nesse contexto, este trabalho apresenta uma arquitetura capaz de fazer o monitoramento de sensores *IoT* que coletam dados em campos agrícolas, como fazendas e plantações. Essa arquitetura também garante uma maior confiabilidade com as informações coletadas dos sensores por se utilizar de uma rede *blockchain* descentralizada. Essa tecnologia propõe que todos os dados que são armazenados nela tenham cópias entre outros usuários da rede, e para que alguma nova informação seja inserida é necessário obter um consenso de mais da metade da rede de que os dados estão íntegros (NOFER *et al.*, 2017). Além de receber os dados dos sensores e guardá-los de forma confiável, a arquitetura também inclui uma aplicação WEB. O servidor da arquitetura, o banco de dados e a aplicação funcionam em um servidor na nuvem, para que possam receber dados de sensores em diferentes distâncias.

O objetivo geral deste trabalho foi desenvolver um sistema para realizar o monitoramento de dados em fazendas, baseado em técnicas de agricultura inteligente. A solução desenvolvida inclui uma rede *blockchain*, visando a descentralização e a integridade dos dados.

A solução implementa uma aplicação para monitorar dados de campos agrícolas e guardar um histórico com essas informações, bem como, projeta uma infraestrutura escalável com armazenamento confiável e descentralizado. O sistema também provê acesso aos dados monitorados em qualquer local com acesso à internet.

Este trabalho está organizado em capítulos. No Capítulo 2 é introduzido com um comparativo entre pesquisas similares a este trabalho e em que se diferenciam. No Capítulo 3 é apresentado os principais conceitos para o trabalho. No Capitulo 4 é descrito os materiais e os métodos que foram realizados para desenvolver esta solução. No Capítulo 5 é abordado uma análise dos resultados obtidos a partir dos experimentos realizados. Por fim, no Capítulo 6 são abordadas as considerações finais e abordado quais a possibilidades para futuros trabalhos.

2 TRABALHOS RELACIONADOS

No artigo de Xie *et al.* (2017), os autores descrevem a importância de manter seguro os dados de todo o processo de produção de alimentos agrícolas, o qual é um problema de integridade de dados em qualquer etapa do processo, isso pode gerar sérios riscos. Esse trabalho também ressalta o amplo emprego de tecnologias *IoT* na agricultura ao longo dos últimos anos.

A arquitetura proposta por eles é baseada é uma rede *blockchain*. Sabendo que em geral os dispositivos *IoT* geram muitos dados para serem armazenados, os autores perceberam um problema que aconteceria ao armazenar ou ler os dados na *blockchain* ao longo do tempo. Assim, sua proposta foca em armazenar o histórico dos processos e desenvolvimento dos produtos agrícolas durante o plantio. Portanto, foi desenvolvido um módulo *IoT* com sensores que leem informações em tempo real e as enviam a um servidor. Os dados que chegam ao servidor são guardados em uma estrutura de duplo-armazenamento, de forma concorrente, de modo que o sistema possa fazer requisições e prover os dados para a camada de *software*. Para resolver o problema de segurança dos dados eles propõem o uso de uma rede descentralizada utilizando *blockchain*.

Contudo, essa solução não é *end-to-end*, pois os autores resolvem a camada física e a camada do servidor, mas seu foco está em resolver o problema de armazenamento e segurança. Assim, não implementam a camada de software para visualização e manipulação dos dados lidos.

No artigo Tse *et al.* (2017), foi abordado sobre as vantagens e melhorias que uma rede *blockchain* poderia trazer para a segurança dos alimentos na China. Esse trabalho analisa todos os setores envolvidos na cadeia de produção de alimentos, e como é imperativo que dado o contexto de problema com os alimentos, possa ser realizado um monitoramento de cada etapa dessa cadeia. Os autores enfatizam que a tecnologia *blockchain*, tem um potencial muito grande para garantir a segurança e integridade das informações sobre os alimentos. Neste trabalho não é dado uma solução de arquitetura para o problema, o foco está em analisar a viabilidade e importância do uso dessa tecnologia para fornecer mais confiança para os consumidores.

Em Papa (2017), o autor fala sobre como a tecnologia *blockchain* pode ser impactante e a compara com a inovação da internet para a geração anterior. Ele também comenta como essa tecnologia tem o potencial de transformar as transações a nível mundial. Nesse trabalho, o autor mostra que a *blockchain* não se limita ao campo do mercado financeiro, mas que aplicado ao agronegócio pode trazer diversos benefícios para o processo, dentre eles é citado a maior confiança que será gerada nos consumidores finais, sabendo de todo o percurso que aquele

alimento passou até chegar em suas mãos, a *blockchain* pode se destacar facilmente porque sua proposta já garante uma maior integridade e segurança nos dados que são armazenados. O autor também apresenta problemas que a tecnologia irá trazer, como irá acontecer uma transição de documentos físicos para digitais, um desconforto pode ser gerado entre as partes que realizam uma transação no agronegócio, e pode haver até uma resistência para aderir a tecnologia.

Na solução proposta por Gomes (2022), o autor trata sobre uma arquitetura *end-to-end* para ler dados de sensores *IoT* de saúde e armazena-os em uma rede *blockchain*. Para mitigar o problema de muitos acessos de leitura e escrita na rede, a solução também utiliza um banco de dados relacional *offchain*, para manipular mais rapidamente os dados armazenados. O autor também elabora uma aplicação WEB para apresentar as informações lidas dos sensores através de gráficos para os usuários.

2.1 Comparação entre os trabalhos

O Quadro 1 aborda uma comparação entre os trabalhos relacionados com base nos critérios relevantes. A primeira coluna refere-se ao domínio ou subdomínio da pesquisa. Os trabalhos tratados estão divididos no domínio da agricultura, que são os trabalhos ligados à fazenda e ao plantio, alimentos, que se refere à cadeia alimentar e segurança dos alimentos, agronegócio, que se relaciona com todas as etapas da operação dos alimentos até chegar nas mãos do consumidor final e a forma como pode ser entregue uma solução mais confiável, e por último temos o domínio da saúde que refere-se aos cuidados com pacientes e as informações coletadas deles. A segunda coluna é com relação ao método de armazenamento utilizado no trabalho, podendo ser utilizado uma rede *blockchain*, ou utilizando também um banco *offchain*, que são os bancos fora da rede *blockchain* como o PostgreSQL. A terceira coluna é sobre a utilização de sensores *IoT*, e por fim, a última coluna descreve a respeito da solução de software implementada.

Quadro 1 – Comparacação entre os trabalhos

Trabalho	**Domínio**	**Armazenamento**	**Sensores IoT**	**Aplicação Web**
(XIE *et al.*, 2017)	Agricultura	*Blockchain*; Database *Offchain*	Sim	Não se aplica
(TSE *et al.*, 2017)	Alimentos	*Blockchain*	Não se aplica	Não se aplica
(PAPA, 2017)	Agronegócio	*Blockchain*	Não se aplica	Não se aplica
(GOMES, 2022)	Saúde	*Blockchain*; Database *Offchain*	Sim	Sim
Este Trabalho	Agricultura	*Blockchain*; Database *Offchain*	Sim	Sim

Fonte: elaborado pelo autor.

Os trabalhos contextualizados nesta seção estão no campo de pesquisa da agricultura e da saúde. Os trabalhos de (TSE *et al.*, 2017) e (PAPA, 2017) são limitados por não usarem a tecnologia *IoT*, pois a arquitetura proposta por eles não é *end-to-end*. Além disso, também não é proposta uma aplicação para interação com o usuário. Esses trabalhos se relacionam com a solução desenvolvida por implementarem uma rede *blockchain* para resolver um problema de confiança do campo da agricultura.

O trabalho de (XIE *et al.*, 2017) é mais próximo da solução desenvolvida, pois além de implementar uma rede *blockchain* é utilizado sensores *IoT* para coletar dados de campos agrícolas. É interessante também o fato dos autores terem solucionado o problema de excessivo gasto de tempo ao fazer consultas na *blockchain*, utilizando um banco de dados auxiliar. No entanto, não é proposta uma solução à nível de software, e não é informado qual plataforma de comunicação *IoT* foi utilizada.

O trabalho de (GOMES, 2022) propõe uma arquitetura muito similar ao que está sendo tratado neste trabalho, as principais diferenças são encontradas primeiramente no domínio da aplicação, que no caso desse autor se trata da saúde portátil, enquanto que este trabalho refere-se a área da agricultura. Outra mudança significativa é no servidor que foi utilizado pelo autor, que foi implementado em *node.js*[1], no caso desta solução, é utilizado o framework *Ruby on Rails*[2], em busca de otimizar a performance da aplicação.

Por fim, a solução desenvolvida neste trabalho usa a tecnologia *blockchain* como forma de armazenamento distribuído dos dados. As informações adquiridas dos sensores *IoT* são transmitidas em uma alta velocidade, por se tratar de muitas leituras em um pequeno intervalo de tempo. Seria inviável fazer consultados à rede *blockchain* sempre que fosse necessário carregar ou escrever no banco, por isso foi utilizado um outro banco de dados separado da rede *blockchain* (*offchain*). Os sensores *IoT* se comunicam através da plataforma FIWARE, que foi o *middleware* desta solução entre os sensores e o servidor.

[1] Link da documentação: https://nodejs.org/en/docs/
[2] Link da documentação: https://rubyonrails.org/

3 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo tratam-se os conceitos e ferramentas abordados neste trabalho, e como eles se relacionam. Na Seção 3.1 é introduzido o conceito de agricultura inteligente e quais ferramentas estão sendo utilizadas nesta área. Na Seção 3.2 são abordados os conceitos principais da tecnologia *IoT* e como é utilizado na agricultura. E, por fim, na Seção 3.3 é abordado sobre a tecnologia *blockchain*, suas características e o problema que busca resolver.

3.1 Agricultura Inteligente

A agricultura inteligente, ou *smart farming*, utilizar-se do avanço da tecnologia, principalmente na área de telecomunicações para expandir e melhorar os aspectos ligados a este campo. Desde o uso de robôs autônomos e sensores portáteis, até o aprendizado de máquina, podem influenciar para melhorar o desempenho das colheitas. Aeronaves não tripuladas para um monitoramento aéreo registrando imagens dos campos podem ser utilizadas para medir a biomassa por exemplo. Outro exemplo é a utilização de árvores de decisão para diferenciar doenças nas plantas, baseado em imagens. Com o uso dessas tecnologias para coleta e para análise de dados, podemos contribuir para uma agricultura mais sustentável.

Com a agricultura inteligente, pode-se melhorar ecologicamente em diversos aspectos, dentre eles, diminuir a utilização de pesticidas e fertilizantes dado que se tem uma maior precisão do plantio. Isto teria como uma possível consequência a diminuição da emissão de gases que aumentam o efeito estufa.

Com o avanço dos sensores, tem-se um monitoramento contínuo da plantação, o que irá ter uma influência direta no estado e uso das plantas, da água e de fertilizantes. Consequentemente, a agricultura torna-se um negócio ainda mais lucrativo, pois se tem um melhor gerenciamento dos recursos que serão utilizados, e alta qualidade na produção. Tudo isso irá causar um efeito positivo para os consumidores, que sentirão uma maior confiança na origem e qualidade do alimento que estarão consumindo (WALTER *et al.*, 2017).

O uso de dispositivos *IoT* na agricultura têm oferecido uma maior precisão da produção dos alimentos. Juntamente com uma maior rotatividade de colheitas no mesmo solo, os sensores podem garantir um valor ótimo na produção do alimento, tornando todo o processo mais automatizado para os fazendeiros. Existem sensores que podem medir a qualidade da água, do ar e do solo, como também a quantidade de nutrientes necessários, e isso é algo muito importante,

principalmente unido às previsões climáticas, que irão auxiliar numa produtividade ótima, dado que existem períodos de seca sazonal no Brasil (KHAN *et al.*, 2020).

A utilização de *blockchain* na agricultura ainda é uma área incipiente, que tem contribuições muito relevantes para os problemas que são enfrentados atualmente. O primeiro problema que torna a solução *blockchain* importante está relacionado com a segurança dos alimentos e da saúde pública. Nos últimos anos, foram relatadas diversas doenças espalhadas devido a uma má administração no processo de plantio dos alimentos, com isso os governos tentam aumentar a segurança com leis e organizações responsáveis por este monitoramento, mas na agricultura inteligente, possibilita-se unir a tecnologia *IoT* com uma rede *blockchain* e garantir esse monitoramento e histórico dos alimentos, garantindo confiança na integridade dos dados por se tratar de uma rede descentralizada de armazenamento. Essa utilização irá, inclusive, colaborar para uma maior confiança dos consumidores com relação aos alimentos que estarão ingerindo (DEMESTICHAS *et al.*, 2020).

3.2 Internet das Coisas

Internet das Coisas é uma tecnologia que permite "coisas"conectadas através da rede. Na maioria dos casos se refere a sensores que se comunicam entre si ou com um servidor de forma *wireless*. Esse têm sido um ramo da tecnologia que tem ganhado muito espaço nos anos mais recentes, dado seu amplo campo de possibilidades de aplicações, das quais podem ser citados a saúde, a agricultura e a automação industrial (LI *et al.*, 2015).

O conceito de *IoT* foi concebido em 1999, onde era mais comum o uso de comunicação via radiofrequência. Com o avanço da tecnologia, passa-se a encontrar com mais facilidade sensores ou objetos se comunicando via *bluetooth* ou Wi-Fi, o que trouxe uma padronização para estabelecer a comunicação de dispositivos, além de outras vantagens como a velocidade na transferência de dados (LI *et al.*, 2015).

Campos de aplicação de tecnologias *IoT* são numerosos e variados. Hoje tem sido investido cada vez mais dinheiro na área de casas inteligentes, por exemplo, pode-se comprar luzes e ventiladores que são controlados por uma aplicação. Também pode ser visto essa tecnologia aplicada à sistemas de segurança. A inovação que essa tecnologia traz é uma combinação de componentes físicos e digitais, influenciando diretamente os avanços no ramos de microprocessadores e microcontroladores (WORTMANN; FLÜCHTER, 2015).

No campo da agricultura tem-se tornado cada vez mais comum o uso de *IoT* e

de tecnologia em nuvem. A tecnologia *IoT* ajuda no avanço e melhoria concernentes ao gerenciamento de recursos, custo-benefício do plantio, gerenciamento da colheita, melhoria na qualidade, quantidade e monitoramento da colheita. O uso dessa tecnologia na agricultura moderna tem tornado todo o processo de plantio mais cômodo, econômico, diminuindo o custo da mão de obra e aumentando a produção em boa qualidade (KHAN *et al.*, 2020).

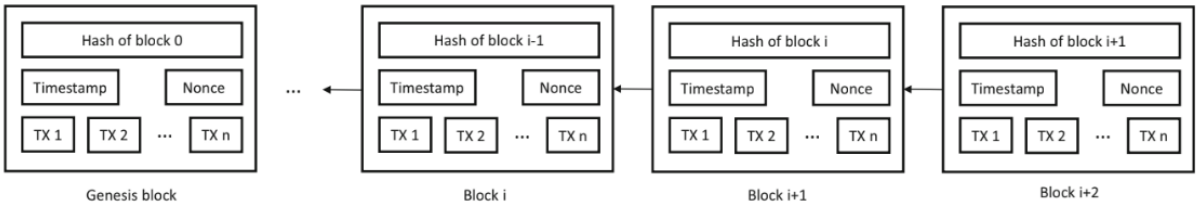
3.3 Blockchain

Blockchain é uma tecnologia que tem crescido cada vez mais no últimos anos. Sua primeira aplicação foi no mercado financeiro ao ser introduzido o *Bitcoin*. Essa tecnologia vai além de aplicações financeiras, e tem atraído muitos segmentos industriais diferentes. O principal problema resolvido com esta tecnologia foi o de como estabelecer confiança em uma rede distribuída. Mais especificamente, uma rede de armazenamento distribuído onde ninguém conseguirá manipular ou alterar os dados sem que seja percebido, pois existem várias cópias dos dados que foram armazenados. (PIERRO, 2017).

A rede *blockchain* consiste em um conjunto de dados, que são compostos por pacotes com parte dos dados (blocos), onde o bloco possui múltiplas transações. A rede de blocos se extende a cada novo bloco inserido que representa um histórico completo das transações realizadas. Os blocos podem ser validados pela rede utilizando criptografia. Em cada bloco é encontrado também um *timestamp*, armazenando a data e hora, um valor de resumo representando o bloco anterior (ou parente), e um número aleatório (*"nonce"*) que valida o *hash*. Os dados guardados em cada bloco não podem ser alterados, e cada bloco possui a responsabilidade de validar o bloco anterior (ou bloco pai). A Figura 1 mostra a cadeia da *blockchain*.

Estes conceitos garantem que há integridade em toda a rede *blockchain* até o primeiro bloco, conhecido como *genesis block*, já que os valores dos *hash* são únicos e qualquer modificação no bloco altera sua respectiva *hash*, qualquer tentativa de fraude seria identificada ao ser comparado com as outras cópias da rede *blockchain* distribuídas (NOFER *et al.*, 2017).

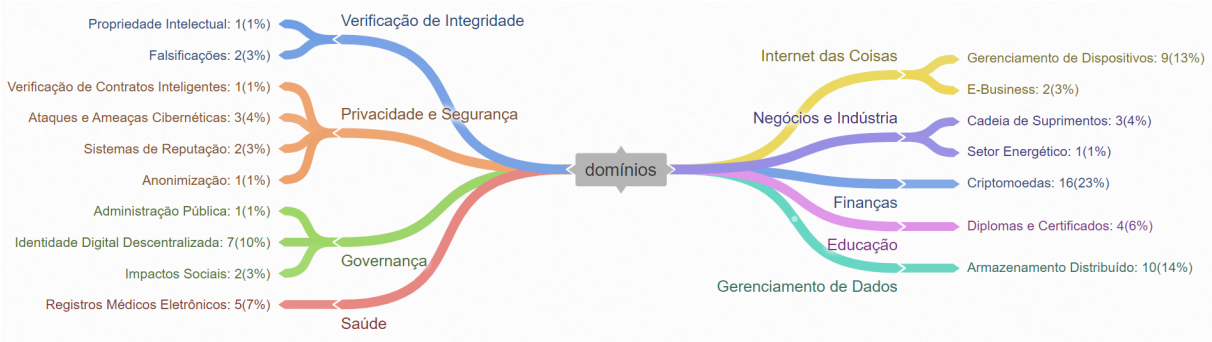
Figura 1 – Cadeia de blocos.



Fonte: Adaptado de Nofer *et al.* (2017)

No Brasil, os domínios de aplicações dessa tecnologia ainda são limitados. A área que mais se encontram os trabalhos é de finanças, seguida por *IoT* e gerenciamento de dados. É compreensível finanças ser o campo majoritário dado que foi o domínio que introduziu a tecnologia *blockchain* (GONÇALVES *et al.*, 2021). Tendo como objetivo investigar os benefícios dessa tecnologia a comunidade brasileira tem realizado pesquisas em alguns domínio como mostrado na Figura 2. Dessa forma, a rede *blockchain* irá ser utilizada neste trabalho para permitir uma maior confiança nos dados que serão armazenados, pois a tecnologia de consenso ajuda a prevenir que usuários tentem manipular e alterar as informações lidas nos sensores.

Figura 2 – Domínio e subdomínios observados para Aplicações da tecnologia *Blockchain*.



Fonte: Adaptado de Gonçalves *et al.* (2021)

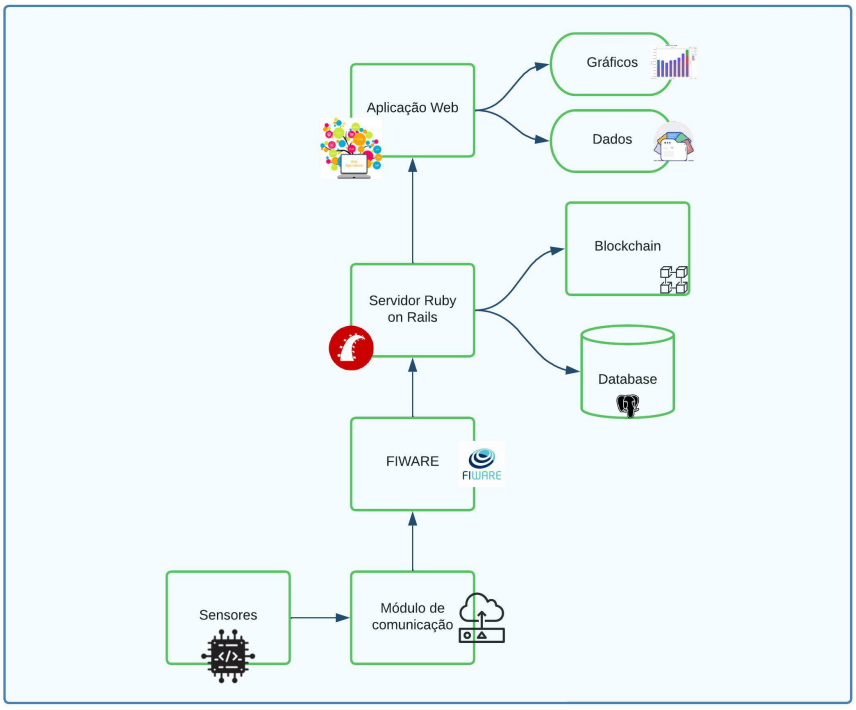
4 PROCEDIMENTOS METODOLÓGICOS

Neste capítulo serão abordados os passos tomados para executar este trabalho. Na Seção 4.1 é introduzido uma visão geral da arquitetura que está sendo proposta. Na Seção 4.2, será explicado o processo para escolher cada tecnologia que foi utilizada na aplicação. Em seguida, é descrito com mais detalhes da implementação de cada camada da arquitetura, a camada física, de comunicação, de armazenamento e da aplicação. Por fim, é tratado o processo de união das camadas.

4.1 Visão Geral da Arquitetura

Na Figura 3 é ilustrado a arquitetura desenvolvida neste trabalho, onde todos os componentes utilizados são *open-source*. É possível interpretá-la como possuindo 4 camadas: física, comunicação, armazenamento e aplicação.

Figura 3 – Visão geral da arquitetura.



Fonte: Elaborado pelo autor.

A camada física é composta pelos sensores e pelo módulo de comunicação, é responsável pela coleta de dados do campo onde ela está instalada, e o módulo de comunicação irá enviar via Wi-Fi esses dados para o *middleware*. A camada superior à física é a de comunicação.

Nela o principal componente é o FIWARE, o qual recebe os dados do módulo, encapsulando-os em mensagens que serão enviadas para o servidor WEB. Em seguida, tem-se a camada de armazenamento, composta pelo o servidor WEB, a rede *blockchain* e o banco de dados *offchain*. O servidor WEB é responsável por receber as informações via protocolo REST do FIWARE, ele irá repassar os dados para o banco *offchain* e para a rede *blockchain*, ele também irá manipular os dados que irão para a camada superior. O uso da rede *blockchain* irá colaborar para atingir o objetivo de manter o armazenamento confiável e descentralizado. A camada mais superior é a da aplicação, nela tem-se um software WEB que irá tratar os dados que foram lidos nos sensores em gráficos e tabelas cumprindo assim o objetivo de implementar uma aplicação para monitorar os dados lidos dos sensores. Essa camada, juntamente com a de armazenamento, estarão em uma plataforma na nuvem, garantindo acesso aos usuários do sistema em qualquer local que tenha internet.

O FIWARE, o servidor *back-end* e a API da rede blockchain, utilizam o modelo de *design Representational State Transfer* (REST), para estabelecer a comunicação via protocolo HTTP. REST é um grupo de restrições, utilizadas para realizar transferências de pacotes na WEB com um contexto bem definido, também estabelecendo um padrão para grande parte dos serviços WEB da atualidade. Como grande parte da WEB se comunica obedecendo às restrições de serviço REST, isso faz com que a plataforma seja de fácil comunicação com API's externas (GUPTA, 2021).

O servidor *Ruby on Rails* tem por convenção a utilização do padrão de arquitetura de software *Model-View-Controller* (MVC) que separa a aplicação em 3 camadas distintas, a camada dos modelos (*Models*) são entidades centrais que guardam a lógica da aplicação. A camada visual (*Views*) descrevem as páginas da interface gráfica e podem ser escritas em HTML ou XML. A camada de controle (*Controllers*) é onde acontece o mapeamento de toda aplicação, onde também é tratada as ações dos usuários da aplicação, essa camada tem acesso direto às *Views* e *Models* (LUCIANO; ALVES, 2017).

4.1.1 Componentes da Arquitetura

O primeiro passo neste trabalho foi definir as tecnologias, ferramentas e componentes que serão utilizados tendo como base o problema que foi apresentado.

4.1.1.1 SenSE

Para a camada física foi utilizado o simulador de sensores *Sensor Simulation Environment* (*SenSE*). Foram realizadas diversas simulações com sensores enviando dados com frequências diversas para replicar um uso real. O software *SenSE* é um simulador de tráfego para sensores, utilizado para simular ambientes mais complexos de *IoT*. Esta ferramenta nos permite simular diversos tipos de sensores, enviando mensagens simultâneas. Sua flexibilidade permite que os sensores sejam configurados pelo usuário, ou utilizar opções de *templates* pré-configurados já fornecidos pelo simulador (ZYRIANOFF *et al.*, 2017).

Utilizar o *SenSE* nesta solução, auxiliou na simulação dos sensores, bem como nos testes de escalabilidade da infraestrutura, de modo que fosse possível documentar a latência e a perda de pacotes em diferentes cenários, variando o número de sensores e a periodicidade no envio de mensagens. Durante os testes o software gerou gráficos para visualização do número de mensagens enviadas e no final é disponibilizado uma planilha constando as mensagens e seus *timestamps* de envio.

4.1.1.2 FIWARE

O FIWARE é uma plataforma *open-source* de padrões em diversos domínios da aplicação, tais como agricultura, água, energia, cidades e indústrias inteligentes, para ajudar na implementação de soluções portáteis e que se comuniquem com dispositivos inteligentes. Fornecendo soluções com uma melhor performance, de fácil acesso e baixo custo (RODRIGUEZ *et al.*, 2018).

Nos últimos anos a comunidade de desenvolvimento da plataforma FIWARE tem investido na implementação de soluções para a agricultura visando auxiliar no monitoramento, processamento de dados e gerenciamento de recursos (RODRIGUEZ *et al.*, 2018). O produto desenvolvido pela FIWARE tem como alvo otimizar a produção em plantações utilizando os meios mais modernos e sustentáveis, entregando os melhores produtos em termos de qualidade, enquanto melhora o retorno. Essa solução se utiliza de sensores *IoT*, sensores vestíveis, serviço de GPS, robôs e drones, garantindo dados em tempo-real para monitoramento da linha de produção e ajudar nas decisões baseadas em dados.

A plataforma FIWARE dispõe de vários componentes para auxiliar nas soluções desenvolvidas, também fazem uso de protocolos de comunicação relevantes para o contexto *IoT*.

Neste trabalho os principais componentes da plataforma FIWARE que foram utilizados são: o *Orion Context Broker*, o *IoT Agent* e o banco de dados *MongoDB*[1]. O protocolo utilizado é o *UltraLight 2.0*.

O *IoT Agent* é componente que permite dispositivos mandarem e receberem dados de um *Context Broker*, utilizando o protocolo de comunicação nativo. Esse componente também lida com os aspectos de autenticação e autorização da plataforma FIWARE. Quando o *IoT Agent* é configurado para utilizar o protocolo *UltraLight 2.0*, que é um protocolo baseado em texto, leve, utilizado em contextos onde a largura de banda e a memória do dispositivo são limitados, este componente funciona como uma ponte entre o padrão de mensagens MQTT e HTTP.

O *Orion Context Broker* é um componente da plataforma que permite o gerenciamento de todo o ciclo de vida das informações no contexto da aplicação, desde atualizações nos dispositivos até notificações sobre alterações. Todas as requisições a ele sendo realizadas via HTTP. A Figura 4 ilustra um exemplo de fluxo utilizando essas configurações do FIWARE.

No contexto da solução desenvolvida neste trabalho uma camada intermediária entre o servidor e a camada física se faz necessária. Esta camada é a de comunicação, que utilizará a plataforma FIWARE. Os dados são transmitidos dos sensores para o *Mosquitto*[2] *Broker MQTT* que por sua vez são repassados para o *IoT Agent* que está inscrito no tópico MQTT dos dispositivos. O *IoT Agent* converte a mensagem recebida no tópico para o protocolo *UltraLight 2.0* e repassa para o *Orion Context Broker*, API REST responsável por gerenciar informações e operações dos dispositivos. O banco de dados MongoDB, por fim, auxilia a guardar as informaçõe necessárias para as entidades do FIWARE.

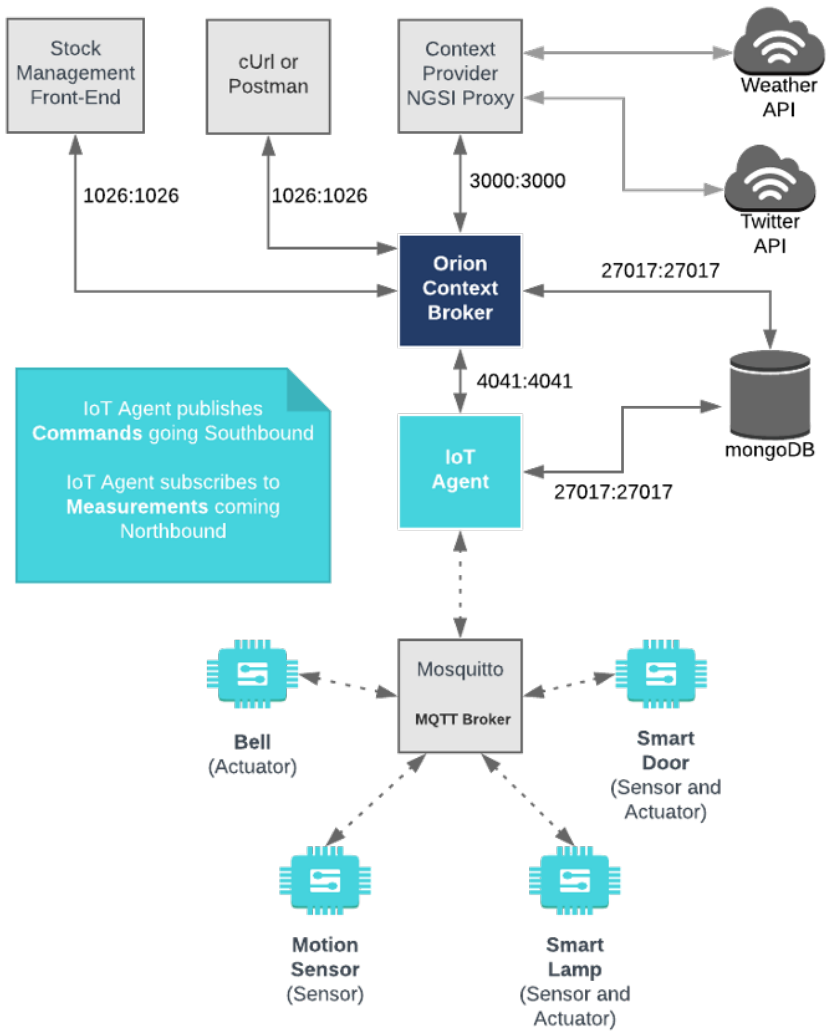
4.1.1.3 Servidor Ruby on Rails e Entidades de Armazenamento

O *framework* que foi utilizado para desenvolver o servidor *back-end* da solução, foi o *Ruby on Rails* que é desenvolvida na linguagem para a WEB, *Ruby*. O servidor é principal componente da solução, ele é quem define todos os fluxos da aplicação WEB, suas rotas, suas características e acessos às entidades de armazenamento.

Uma estratégia utilizando duas formas de armazenamento diferentes foi utilizada nesta solução, um banco de dados sequencial e *offchain* foi utilizado para guardar o grande volume de mensagens lidas dos sensores, este banco é o PostgreSQL, e para trazer os aspectos

[1] Link da documentação: https://www.mongodb.com/docs/
[2] Link da documentação: https://mosquitto.org/documentation/

Figura 4 – Arquitetura de funcionamento do FIWARE utilizando MQTT



Fonte: Adaptado de Fox (2018)

de confiança e integridade dos dados uma rede *blockchain open-source* foi utilizada[3].

4.1.1.4 Interface Gráfica da Aplicação WEB

Por fim, a interface gráfica da aplicação WEB foi implementada utilizando o framework *bootstrap 5*[4], capaz de mostrar de forma clara os dados lidos pelos sensores em um *dashboard*. As ferramentas descritas nesta subseção foram escolhidas para que o objetivo de monitorar os dados e guardar um histórico fosse alcançado.

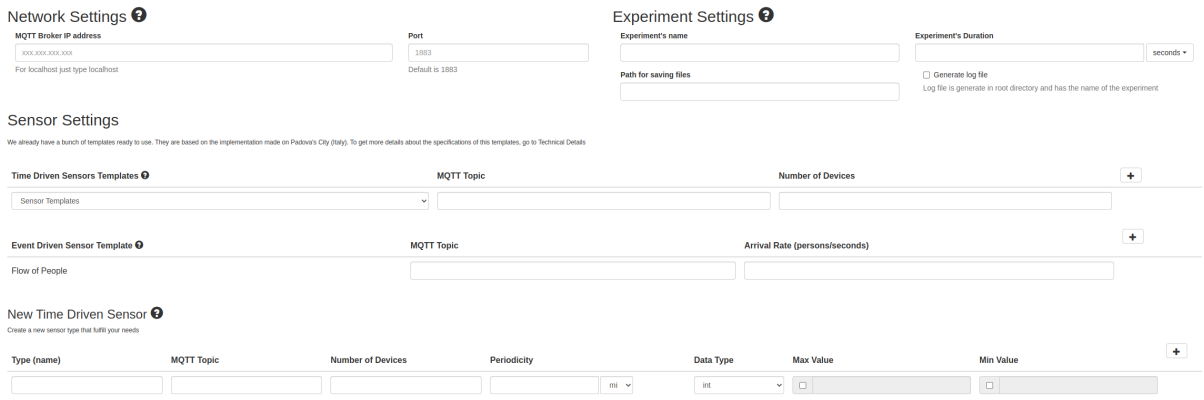
[3] Link para download: https://github.com/adhavpavan/FabricNetwork-2.x
[4] Link da documentação: https://getbootstrap.com/docs/5.0/getting-started/introduction/

4.2 Etapas do Desenvolvimento

4.2.1 Implementação das Camadas da Aplicação de Forma Independente

Nesta fase da execução foi implementado cada um dos componentes que serão necessários para compor a infraestrutura. Começando pela camada física, foi necessário executar o software *SenSE*. Para isso, foi preciso baixar a aplicação disponível em Zyrianoff (2017) e alterar o código fonte para adaptar o padrão das mensagens para *UltraLight 2.0*, após isto, a aplicação funcionou utilizando o servidor WEB java*Apache Tomcat*. A Figura 5 ilustra a página de configuração dos sensores no *SenSE*.

Figura 5 – Página de configuração dos sensores no *SenSE*.



Fonte: Elaborado pelo autor.

Para configurar todas as entidades do FIWARE, utilizam-se as imagens *docker*[5] disponíveis. Todas essas entidades foram configuradas no arquivo *docker-compose.yml* da aplicação. O *Orion* foi mapeado para a url *orion:1026* e configurado para ser dependente do serviço *MongoDB*. O *IoT Agent* foi mapeado para *iot-agent:4041* e configurado para o protocolo *UltraLight 2.0*, depende dos serviços do *MongoDB* e do broker *mosquitto*. O *MongoDB* foi mapeado para *mongo-db:27017*. Já o *Broker Mosquitto* foi mapeado para *mosquitto:1883*. Com isso configuramos os serviços do FIWARE para estarem habilitados a serem utilizados. Na Figura 6 é mostrado um exemplo de serviço configurado no arquivo *docker-compose.yml*, neste caso o serviço é do *IoT Agent*.

O servidor *Ruby on Rails* foi implementado utilizado sua versão estável mais atualizada (*Rails* 7), por ter novas bibliotecas já incluídas no framework, como o *Stimulus*[6], framework

[5] Link da documentação: https://docs.docker.com/
[6] Link da documentação: https://stimulus.hotwired.dev/handbook/origin

Figura 6 – Exemplo de configuração do *IoT Agent*.

```
iot-agent:
  image: fiware/iotagent-ul:${ULTRALIGHT_VERSION}
  hostname: iot-agent
  container_name: fiware-iot-agent
  depends_on:
    - mongo-db
    - mosquitto
  networks:
    - default
  expose:
    - "${IOTA_NORTH_PORT}"
  ports:
    - "${IOTA_NORTH_PORT}:${IOTA_NORTH_PORT}" # localhost:4041
  environment:
    - IOTA_CB_HOST=orion
    - IOTA_CB_PORT=${ORION_PORT}
    - IOTA_NORTH_PORT=${IOTA_NORTH_PORT}
    - IOTA_REGISTRY_TYPE=mongodb
    - IOTA_LOG_LEVEL=DEBUG
    - IOTA_TIMESTAMP=true
    - IOTA_CB_NGSI_VERSION=v2
    - IOTA_AUTOCAST=true
    - IOTA_MONGO_HOST=mongo-db
    - IOTA_MONGO_PORT=${MONGO_DB_PORT}
    - IOTA_MONGO_DB=iotagentul
    - IOTA_MQTT_HOST=mosquitto
    - IOTA_MQTT_PORT=1883
    - IOTA_DEFAULT_RESOURCE=
    - IOTA_PROVIDER_URL=http://iot-agent:${IOTA_NORTH_PORT}
    - IOTA_DEFAULT_TRANSPORT=MQTT
  healthcheck:
    interval: 5s
```

Fonte: Elaborado pelo autor.

javascript, utilizado para facilitar e organizar o *javascript* da aplicação. A arquitetura utilizada para o desenvolvimento foi a monolítica, utilizando o modelo MVC para organização da aplicação. Foi utilizada uma *gem* para realizar autenticação de usuários chamada *Devise*[7]. O servidor também é responsável por comunicar-se com o banco de dados PostgreSQL, este banco foi utilizado para melhorar a latência de acesso às informações dos sensores, dado que na rede *blockchain* o tempo de leitura e escrita de cada bloco é considerável. O servidor utiliza o padrão de projeto *Active record* para realizar as operações ao banco *offchain PostgreSQL*.

Para implementar a rede blockchain foi utilizado uma API em *Node.js* para mandar requisições REST. A rede possui um contrato inteligente que define as operações de leitura e escrita entre as máquinas conectadas.

A aplicação WEB foi implementada em um modelo de *dashboard* utilizando o framework *bootstrap 5* para visualizar os dados lidos dos sensores, a aplicação também conta com tabelas para visualizar as informações gerais sobre os sensores e da validade das transações *blockchain*. Também conta com um sistema de autenticação de usuário utilizando a biblio-

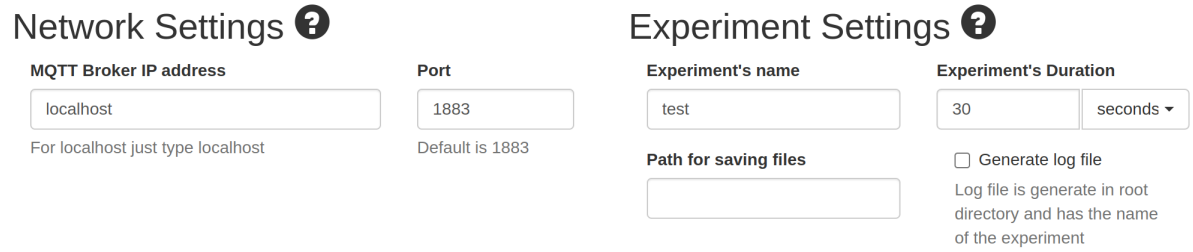
[7] Link da documentação: https://github.com/heartcombo/devise

teca *Devise*. Essas camadas cumprem o objetivo de projetar um armazenamento confiável e descentralizado utilizando *blockchain*, bem como fornecer uma aplicação para acesso aos dados.

4.2.2 Integração das camadas da Arquitetura

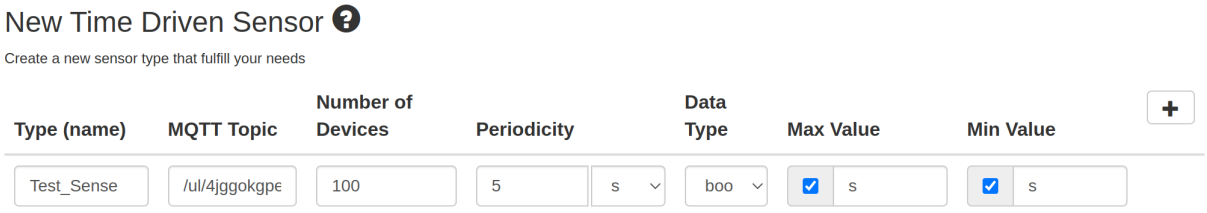
Após validado o correto funcionamento de cada componente da arquitetura, houve a integração deles. O SenSE após alterado para enviar mensagens no formato *UltraLight 2.0* foi configurado para mandar mensagens para um tópico MQTT específico, onde o *IoT Agent* estava recebendo as modificações. Nas Figuras 7 e 8 são ilustrados exemplos de como configurar o simulador.

Figura 7 – Configuração geral dos sensores *SenSE*.



Fonte: Elaborado pelo autor.

Figura 8 – Configuração específica dos sensores *SenSE*.



Fonte: Elaborado pelo autor.

O *IoT Agent* sempre atualiza no *MongoDB* o valor enviado, não mantendo um histórico de todos os valores já cadastrados para um mesmo sensor, mas mantendo no banco apenas o valor mais recente. O *Orion Context Broker* sempre é notificado quando um novo dado chega no *IoT Agent*.

O servidor *Rails* comunica-se diretamente com duas entidades da plataforma FIWARE, o *Orion* e o *IoT Agent*, através de requisições HTTPS no padrão REST. O *Orion* implementa uma API que fornece a possibilidade de criar *webhooks*, ferramenta muito importante para receber a atualização das leituras dos sensores, pois ao invés de ser necessário realizar *polling*, para atualizar as leituras dos sensores (algo que seria insustentável quando a plataforma escalar),

foi possível cadastrar *webhooks* que eram mapeados para uma rota na aplicação que tratam todas as notificações de atualizações dos sensores, e este processo acontece de forma assíncrona. Assim, utilizando essa estratégia foi possível escalar a quantidade de sensores suportados pela plataforma. A Figura 9 mostra um exemplo de assinatura de um *webhook* da aplicação.

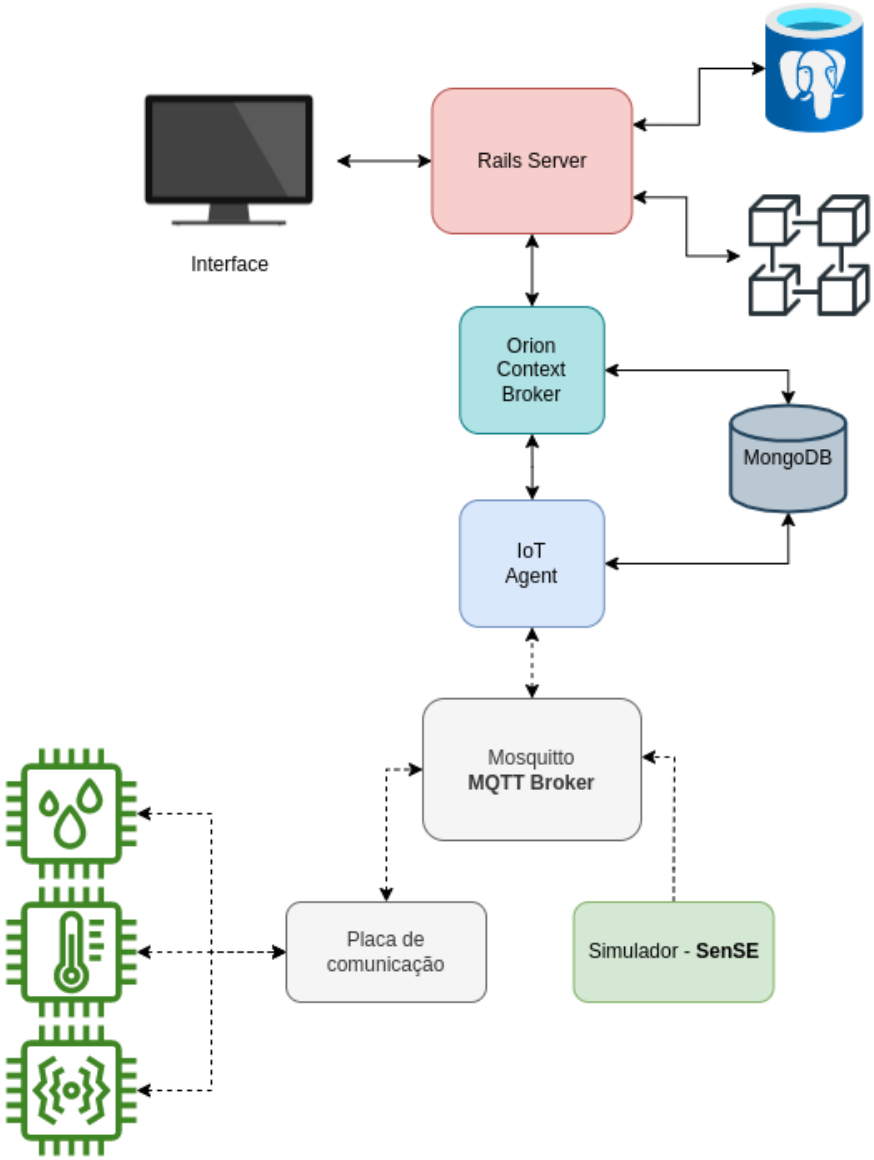
Figura 9 – Exemplo de *webhook* criado.

```
{
    "id": "6385543f5c1470325c1c780c",
    "description": "A subscription to get info about temperature",
    "status": "active",
    "subject": {
        "entities": [
            {
                "idPattern": "urn:ngsd-ld:temp001"
            }
        ],
        "condition": {
            "attrs": []
        }
    },
    "notification": {
        "attrs": [],
        "onlyChangedAttrs": false,
        "attrsFormat": "normalized",
        "http": {
            "url": "http://smart-farming:3000/devices/data_notifications"
        },
        "covered": false
    }
}
```

Fonte: Elaborado pelo autor.

Após cada evento recebido pelos *webhooks*, o servidor salva no banco de dados *offchain PostgreSQL* o valor e um *timestamp* para guardar quando o dado foi criado e/ou modificado. Através da aplicação o usuário tem a opção de salvar também na rede *blockchain*, onde uma requisição será enviada para a rede de forma assíncrona para guardar o dado. Todos esses dados guardados podem ser visualizados através da interface gráfica da aplicação, os dados mostrados pela interface são os dados do banco *offchain* para que as operações ao banco sejam mais rápidas. Pode-se ver pela Figura 10 a arquitetura completa.

Figura 10 – Arquitetura Desenvolvida.



Fonte: Elaborado pelo autor.

5 EXPERIMENTOS E RESULTADOS

A infraestrutura foi avaliada com o intuito de validar seu comportamento quando submetida à diferentes tipos de cenários. Os cenários irão variar de acordo com o número de sensores conectados ao sistema, o que foi útil para observar como essa variação impacta no tempo de que o sistema leva para tratar as mensagens, ou se perdeu pacotes.

Para realizar os experimentos foi utilizado o simulador de sensores *SenSE*, abordado na subseção 4.1.1.1. Foram feitos vários testes alterando o número de sensores que transmitem dados para o sistema. Neste momento o software gerou gráficos e tabelas para analisar a latência e a perda de pacotes que ocorreu durante o período de testes.

O objetivo principal destes experimentos foi observar a latência do momento em que a mensagem foi gerada pelos sensores até o momento em que os dados foram armazenados no banco *offchain* da aplicação. Foi observado também a relação da quantidade de sensores na rede com a quantidade de pacotes perdidos.

Em seguida, foram analisados os resultados dos experimentos. Os valores de tempo de resposta do sistema foram comparados, e com isso foi possível saber como a alteração no número de sensores afetou o sistema. Essa etapa visou validar que o objetivo de projetar uma infraestrutura escalável foi alcançado. Gráficos foram utilizados para realizar as comparações.

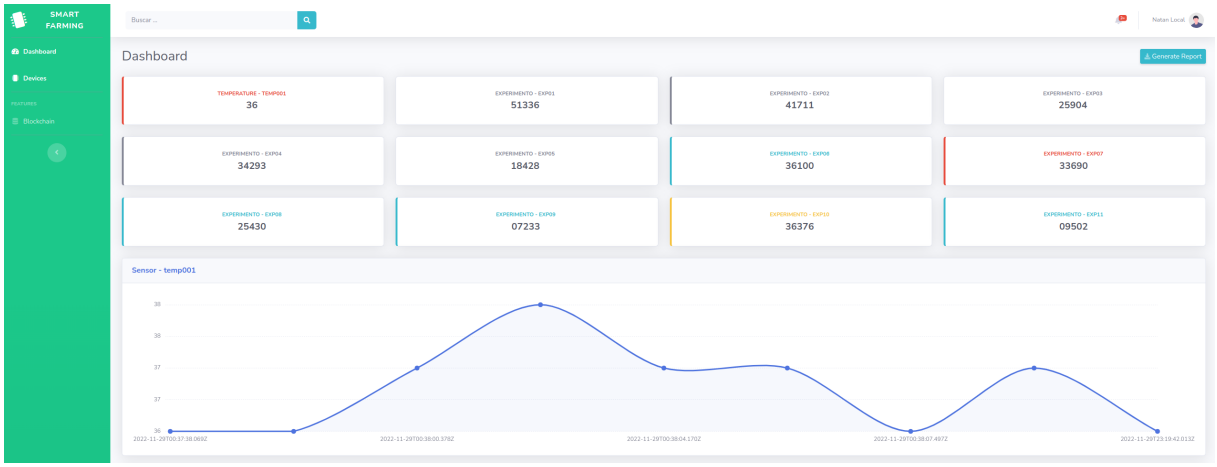
5.1 Interface Gráfica

Um dos resultados obtidos foi a interface gráfica desenvolvida utilizando as linguagens *Ruby*, *javascript* e o *framework bootstrap 5*, como pode ser visto na Figura 11. O *dashboard* é a tela da aplicação que serão mostrados os gráficos e as informações principais. Na lateral esquerda da figura é tratado o menu onde o usuário tem acesso à área dos dispositivos e também às transações com a rede *blockchain*. Na parte supeior da tela foi colocado o componente navbar, onde apresenta o usuário que está logado e uma barra de pesquisas, logo abaixo no canto superior direito existe um botão *Generate Report* que gera um arquivo *.csv* com todos os dados lidos por todos os sensores.

Na parte das informações apresentadas, visualiza-se alguns cartões que mostram o tipo de dispositivo, seu nome e o último valor que foi lido. Logo abaixo dos cartões são gerados gráficos para cada dispositivo, sendo o eixo X o *timestamp* da leitura do dado e o eixo Y o valor da leitura. Os cartões e os gráficos são gerados de forma dinâmica pela aplicação, ou seja, sempre

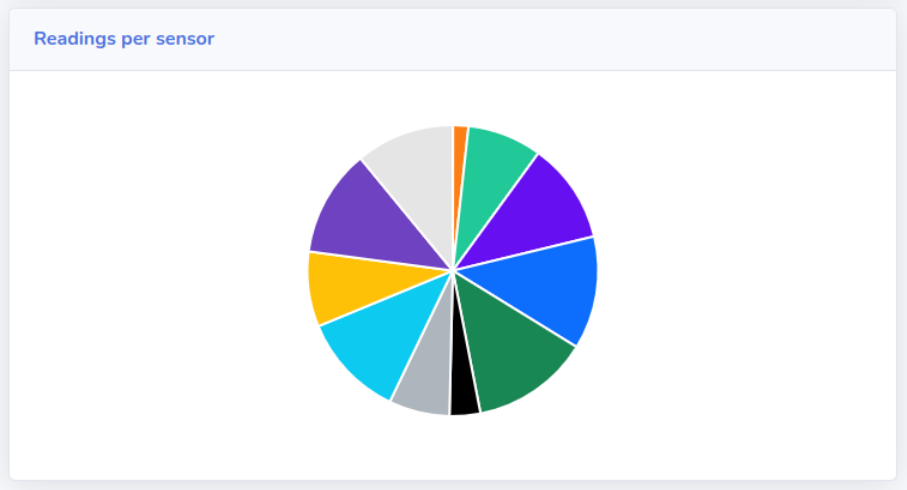
que um novo dispositivo é criado, automaticamente a aplicação prepara essas informações no dashboard. Na Figura 12, ainda no *dashboard*, é mostrado o número de leituras feitas por cada sensor ao passar o mouse por cima de cada fatia.

Figura 11 – Dashboard da plataforma.



Fonte: Elaborado pelo autor.

Figura 12 – Gráfico de leituras por sensor.

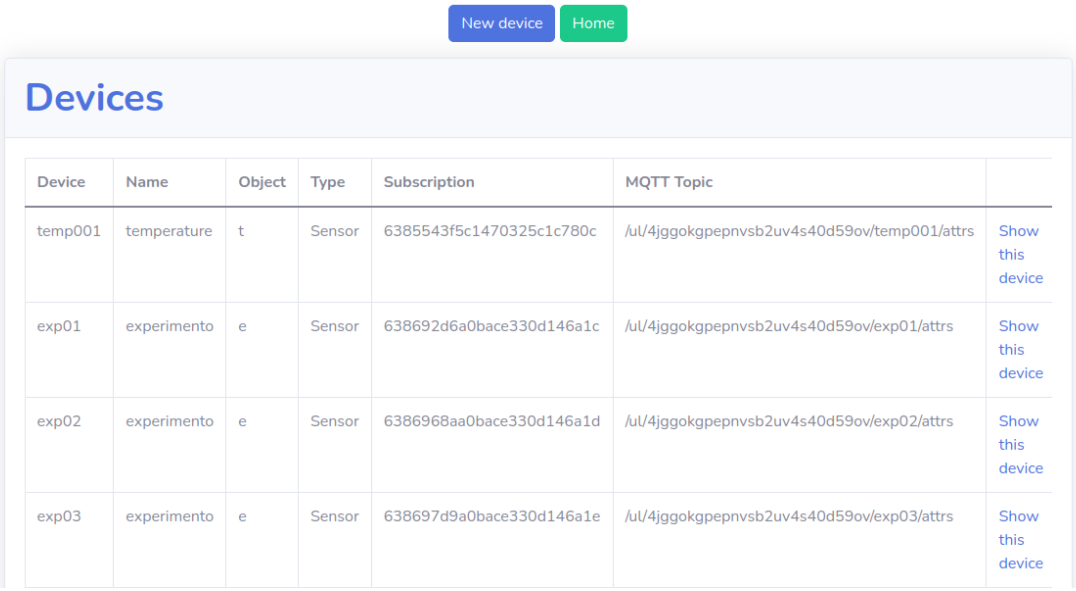


Fonte: Elaborado pelo autor.

Também foi desenvolvida uma tabela para mostrar os dispositivos cadastrados e as principais informações relacionadas a eles. A tabela pode ser vista na Figura 13. A primeira coluna da tabela mostrada na figura mostra o nome dado ou dispositivo, na coluna seguinte(*Name*) é apresentado o tipo daquele dispositivo. Na coluna *Object*, é mostrado a variável cadastrada para o dispositivo. Essa variável é importante para seguir o padrão esperado pelo FIWARE ao receber as mensagens dos sensores. Essa variável pode ser um único caractere ou uma *string* sem espaços. Na coluna *Subscription* é apresentado o id do *webhook* que o dispositivo é associado.

Se o dispositivo não tiver uma assinatura relacionada à ele a aplicação não conseguirá receber os dados vindos do sensor. Na coluna *MQTT Topic* mostra a *string* correspondente ao tópico *MQTT* que os sensores precisam fazer a publicações das mensagens. Esse tópico é formado por uma *chave* cadastrada nas variáveis de ambiente da aplicação concatenada com o nome do sensor. A última coluna é um botão que leva para uma página mostrando todas as informações de um dispositivo específico.

Figura 13 – Página de dispositivos cadastrados.



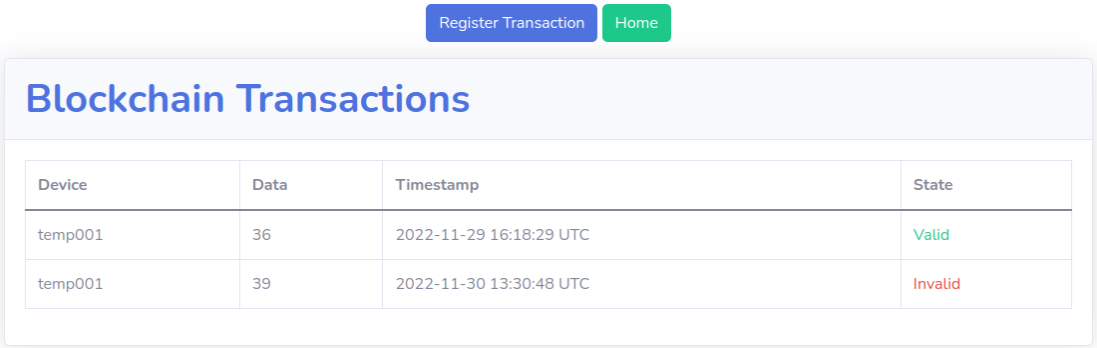
Fonte: Elaborado pelo autor.

Foi desenvolvido também uma página para mostrar as transações *blockchain* que foram cadastradas na aplicação. A Figura 14 mostra esta tabela. Na primeira coluna, *Device*, é o nome do dispositivo que fez a transação. Na coluna seguinte, *Data*, é o valor que foi salvo na rede *blockchain*. Na terceira coluna, *Timestamp*, é a data em que a transação foi realizada. Na última coluna, *State*, é um *status* que checa se o valor que foi armazenado ainda está confiável. Essa checagem é feita pela comparação das informações da transação armazenadas no *PostgreSQL* e na rede *blockchain*. Na imagem vemos o exemplo de um dado que está íntegro e outro que foi corrompido ou houve uma tentativa de ser manipulado no banco *offchain*.

5.2 Configuração dos Experimentos

Para validar que a plataforma é escalável, foram realizados 11 experimentos, variando 2 parâmetros: a quantidade de sensores conectados ao sistema e a periodicidade com que cada sensor envia dados. Como foi comentados no início do Capítulo 5 estes experimentos foram

Figura 14 – Página de transações *blockchain*.



Fonte: Elaborado pelo autor.

realizados utilizando o *SenSE*, um simulador de sensores comentado na subseção 4.1.1.1.

Na Tabela 1 é apresentado os experimentos que foram realizados e como a arquitetura se comportou com relação aos pacotes recebidos. Para todos os experimentos demonstrados na tabela, foi executado o teste durante 1 minuto, para validar que a plataforma iria lidar com o envio de mensagens em grandes quantidades.

Tabela 1 – Resultado dos Experimentos

Experimento	**Quantidade de sensores**	**Periodicidade (s)**	**Quantidade de mensagens enviadas**	**Quantidade de mensagens recebidas**	**Pacotes recebidos (%)**
1	10	1	592	592	100,0
2	50	1	2870	2870	100,0
3	100	1	5367	3877	72,2
4	200	1	10015	4356	43,5
5	500	1	20702	4556	22,0
6	100	5	1187	1187	100,0
7	200	5	2344	2344	100,0
8	500	5	5436	4029	74,1
9	500	10	2891	2891	100,0
10	1000	10	5427	4135	76,2
11	1000	15	3792	3792	100,0

Fonte: elaborado pelo autor.

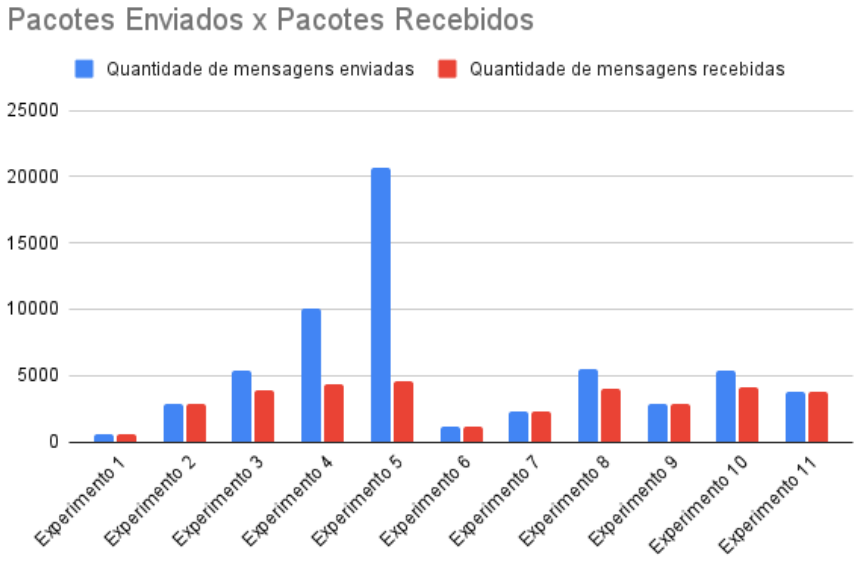
Na primeira coluna da Tabela 1 aborda o indicativo de qual experimento a linha se refere. Na coluna seguinte, tem-se a *quantidade de sensores* conectados ao sistema. A terceira indica a *periodicidade* em segundos que os dispositivos mandaram mensagens para a arquitetura. A quarta coluna mostra a *quantidade de mensagens enviadas* no total, pelo simulador. Na quinta, trata a quantidade de mensagens que foram armazenadas no banco *offchain*, PostgreSQL. E na última coluna, é abordada uma razão dos dados recebidos em relação ao que foi enviado.

Nas Figuras 15 e 16 estão sendo ilustradas à relação de quantidade de pacotes enviados pelo simulador e a quantidade de pacotes que efetivamente foram salvos no banco de

Figura 15 – Relação dos pacotes enviados e recebidos em linhas.

Fonte: Elaborado pelo autor.

Figura 16 – Relação dos pacotes enviados e recebidos em barras.



Fonte: Elaborado pelo autor.

dados PostgreSQL. É possível perceber que com uma grande quantidade de mensagens em um período de tempo curto a plataforma tende a ter uma perda considerável de pacotes, conforme experimento 5, tendo conectado 500 sensores à aplicação, mandando mensagem a cada segundo. Obtivemos 20.702 pacotes enviados em apenas 1 minuto do experimento, a aplicação conseguiu receber apenas 22% desses pacotes. Também é importante perceber que com uma periodicidade de 15 segundos por leitura, foi possível conectar 1.000 sensores na plataforma sem nenhuma perca de pacotes, como mostra o experimento 11.

5.3 Análise das Latências

Cada vez que o simulador de sensores executa, é gerado um arquivo de *log* com um id, *timestamp* e valor de cada mensagem enviada. Esse arquivo foi essencial para realizar o cálculo das latências. Outro aspecto importante é o fato de sempre ser guardado no banco de dados *offchain* um *timestamp* de quando a tupla foi inserida na tabela. Então tendo o momento em que a mensagem foi enviada e o momento em que a tupla foi inserida na tabela. Um *script* foi escrito para realizar o cálculo das latências médias em milissegundos de cada um dos experimentos. A Tabela 2 mostra o valor das latências para cada um dos experimentos que não houveram perca de pacotes.

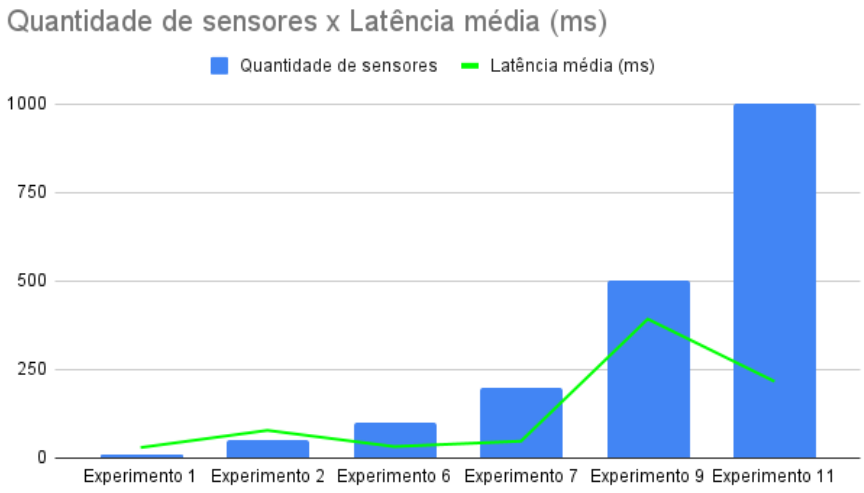
Tabela 2 – Latência média dos experimentos

Experimento	Quantidade de sensores	Periodicidade (s)	Latência média (ms)
1	10	1	29,82
2	50	1	78,26
6	100	5	32,52
7	200	5	47,76
9	500	10	392,66
11	1000	15	216,55

Fonte: elaborado pelo autor.

Na Figura 17 é ilustrado uma relação entre a latência e a quantidade de sensores. Essa relação não é diretamente proporcional, mas é possível perceber que a medida que a quantidade de sensores aumenta a latência tende a aumentar também.

Figura 17 – Relação da quantidade sensores e a latência.



Fonte: Elaborado pelo autor.

6 CONCLUSÕES E TRABALHOS FUTUROS

A agricultura inteligente têm crescido nos últimos anos, mais pesquisas e soluções tecnológicas estão sendo propostas a cada dia. A união da agricultura com dispositivos *IoT* têm tornado os plantios cada vez mais produtivos, e deixado o trabalho no campo mais fácil para os envolvidos. Com os dados disponíveis em tempo real para os administradores de um campo agrícola, o processo de tomada de decisões se tornou mais fácil. Dessa maneira, a união de componente de arquitetura de software e de hardware para solucionar problemas no domínio da agricultura, podem trazer resultados ainda maiores, tanto para pesquisas acadêmicas quanto para a indústria.

6.1 Considerações Finais

Este trabalho tratou o desenvolvimento de uma arquitetura que monitorasse sensores *IoT* que estariam instalados em campos agrícolas, como fazendas, plantações, ou até estufas. O sistema deveria garantir integridade e confiança nos dados armazenados, utilizando uma rede *blockchain* descentralizada. Uma interface gráfica na WEB deve mostrar os dados coletados em tempo real para o usuário em qualquer local com acesso à internet. Desta forma, o objetivo geral deste trabalho é fazer o monitoramento de dados utilizando técnicas da agricultura inteligente.

Foi construída uma arquitetura, utilizando os componentes descritos na subseção 4.1.1, o simulador de sensores *SenSE*, a plataforma FIWARE, o *framework Ruby on Rails*, o banco de dados *PostgreSQL*, uma rede *blockchain*, e uma interface gráfica WEB. Os experimentos foram realizados para validar o potencial de escalabilidade da plataforma, neles foram conectados até 1.000 dispositivos de forma simultânea (não sendo esse um número limitante), mandando mensagens a cada 15 segundos, sem haver perda de pacotes, ou seja, é provável que em campo esta solução se encaixe para resolver o problema de muitas fazendas.

Um grande benefício desta arquitetura é ter sido desenvolvida de forma genérica e dinâmica, capaz de ser adaptada e utilizada em outros domínios da aplicação como na saúde, em casas inteligentes, ou indústrias. Basta adaptar os recursos do sistema às necessidades do contexto.

Dessa forma, a arquitetura pode ser aplicada a contextos menores como estufas até fazendas de médio porte. Os dados são armazenados de forma confiável e íntegra, e o usuário pode visualizar o status do dado à partir da aplicação para saber se o dado foi corrompido de

alguma forma. O acesso à plataforma é possibilitado em qualquer local com acesso à internet, cumprindo assim os objetivos estabelecido neste trabalho. Alguns problemas de latência foram encontrados, mas que podem ser resolvidos com melhorias na implementação da camada de armazenamento.

6.2 Trabalhos Futuros

Em trabalhos futuros podem ser melhoradas as características da plataforma para suprir melhor as necessidades da agricultura. Novas funcionalidades podem ser adicionadas e melhorias no que se realizou pelo presente trabalho em todas as camadas. Começando pela camada da aplicação, levando em conta a interface gráfica e o servidor, podem ser adicionado uma nova entidade para controlar dispositivos atuadores, essa entidade já é suportada pelo *middleware* que foi utilizado. Também pode ser alterado o uso da rede *blockchain* na aplicação para ter uma melhor cobertura dos dados armazenados. Outro ponto de possível melhoria, é integrar com API's externas que podem fornecer dados para ajudar nas análises dos campos agrícolas.

Experimentos mais aprofundados podem ser realizados na plataforma, para garantir sua performance em outros aspectos, ainda nesse sentido, realizar também experimentos em campo seria uma boa forma de entender outros aspectos para serem melhorados.

REFERÊNCIAS

BERMEO-ALMEIDA, O.; CARDENAS-RODRIGUEZ, M.; SAMANIEGO-COBO, T.; FERRUZOLA-GÓMEZ, E.; CABEZAS-CABEZAS, R.; BAZÁN-VERA, W. Blockchain in agriculture: A systematic literature review. In: SPRINGER. **International Conference on Technologies and Innovation**. [*S. l.*], 2018. p. 44–56.

DEMESTICHAS, K.; PEPPES, N.; ALEXAKIS, T.; ADAMOPOULOU, E. Blockchain in agriculture traceability systems: A review. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 10, n. 12, p. 4113, 2020.

FOX, K. S. J. **IoT Over MQTT**. 2018. Disponível em: https://github.com/FIWARE/tutorials.IoT-over-MQTT. Acesso em: 01 dez. 2022.

GOMES, A. N. Uma solução para compartilhamento de dados de saúde baseada em blockchain permissionada e internet das coisas para hospitais inteligentes. In: **Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação)**. Universidade Federal do Ceará, Campus de Quixadá, Quixadá, 2022. Disponível em: http://www.repositorio.ufc.br/handle/riufc/65342.

GONÇALVES, G. D.; COUTINHO, E.; FREITAS, A. E. S. Um panorama da pesquisa em blockchain no brasil. **SBC Horizontes**, 2021. ISSN 2175-9235. Disponível em: http://horizontes.sbc.org.br/index.php/2022/05/um-panorama-da-pesquisa-em-blockchain-no-brasil/. Acesso em: 31 maio. 2022.

GUPTA, L. **REST API Tutorial**. [S.l], 2021. Disponível em: https://restfulapi.net/. Acesso em: 05 dez. 2022.

KHAN, N.; SIDDIQUI, B. N.; KHAN, N.; ISMAIL, S. The internet of thing in sustainable agriculture. **Artech J. Res. Stud. Agric. Sci**, v. 2, p. 12–15, 2020.

LI, S.; XU, L. D.; ZHAO, S. The internet of things: a survey. **Information systems frontiers**, Springer, v. 17, n. 2, p. 243–259, 2015.

LUCIANO, J.; ALVES, W. J. B. Padrão de arquitetura mvc: model-view-controller. **EPeQ Fafibe**, [S.l], v. 1, n. 3a, p. 102–107, 2017.

NOFER, M.; GOMBER, P.; HINZ, O.; SCHIERECK, D. Blockchain. **Business & Information Systems Engineering**, Springer, v. 59, n. 3, p. 183–187, Jun 2017. ISSN 1867-0202. Disponível em: https://doi.org/10.1007/s12599-017-0467-3. Acesso em: 10 jun. 2022.

PAPA, S. F. Use of blockchain technology in agribusiness: Transparency and monitoring in agricultural trade. In: ATLANTIS PRESS. **2017 International Conference on Management Science and Management Innovation (MSMI 2017)**. [*S. l.*], 2017. p. 38–40.

PIERRO, M. D. What is the blockchain? **Computing in Science & Engineering**, IEEE, v. 19, n. 5, p. 92–95, 2017.

RODRIGUEZ, M. A.; CUENCA, L.; ORTIZ, A. Fiware open source standard platform in smart farming-a review. In: SPRINGER. **Working conference on virtual enterprises**. [*S. l.*], 2018. p. 581–589.

TSE, D.; ZHANG, B.; YANG, Y.; CHENG, C.; MU, H. Blockchain application in food supply information security. In: IEEE. **2017 IEEE international conference on industrial engineering and engineering management (IEEM)**. [*S. l.*], 2017. p. 1357–1361.

WALTER, A.; FINGER, R.; HUBER, R.; BUCHMANN, N. Smart farming is key to developing sustainable agriculture. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 114, n. 24, p. 6148–6150, 2017.

WORTMANN, F.; FLÜCHTER, K. Internet of things. **Business & Information Systems Engineering**, Springer, v. 57, n. 3, p. 221–224, 2015.

XIE, C.; SUN, Y.; LUO, H. Secured data storage scheme based on block chain for agricultural products tracking. In: IEEE. **2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)**. [*S. l.*], 2017. p. 45–50.

ZYRIANOFF, I. **SenSE - Sensor Simulation Environment**. 2017. Disponível em: https://github.com/ivanzy/SenSE-Sensor-Simulation-Environment. Acesso em: 27 nov. 2022.

ZYRIANOFF, I.; BORELLI, F.; KAMIENSKI, C. Sense–sensor simulation environment: Uma ferramenta para geração de tráfego iot em larga escala. **Simpósio Brasileiro de Redes e Sistemas Distribuídos (SBRC)**, [S.l], 2017.