Research, Society and Development, v. 10, n. 12, e239101220273, 2021
(CC BY 4.0) | ISSN 2525-3409 | DOI: http://dx.doi.org/10.33448/rsd-v10i12.20273

Transparência de conectividade de Serviços Blockchain em Sistemas IoT: uma proposta de arquitetura

Connectivity Transparency of Blockchain Services in IoT Systems: a proposed architecture

Transparencia de la conectividad de los servicios Blockchain en los sistemas IoT: una propuesta de arquitectura

Recebido: 06/09/2021 | Revisado: 13/09/2021 | Aceito: 17/09/2021 | Publicado: 19/09/2021

Edgar Natanael de Vasconcelos Gregório
ORCID: https://orcid.org/0000-0002-1678-8221
Universidade Federal Rural de Pernambuco, Brasil
E-mail: edgarnatanael28@hotmail.com
Fernando Antonio Aires Lins
ORCID: https://orcid.org/0000-0002-4007-3891
Universidade Federal Rural de Pernambuco, Brasil
E-mail: fernandoaires@ufrpe.br
Obionor de Oliveira Nóbrega
ORCID: https://orcid.org/0000-0003-1721-9669
Universidade Federal Rural de Pernambuco, Brasil
E-mail: obionor.nobrega@ufrpe.br

Resumo
A Internet das Coisas criou um mundo conectado através dos mais diversos tipos de sensores, possibilitando que objetos de nossas vidas cotidianas estejam interligados à rede. No entanto, à medida que a quantidade de dados gerados por esses dispositivos aumentam, o risco de roubo e adulteração deles se torna ainda mais relevante. Para melhorar o nível de segurança dentro da Internet das Coisas, o *Blockchain* se apresenta como uma potencial solução para a integridade dos dados. Contudo, a realização da conexão entre dispositivos IoT e *Blockchain* é uma tarefa de razoável complexidade. Neste contexto, este trabalho propõe uma arquitetura para transparência de conectividade de serviços *Blockchain* em Sistemas IoT. Esta arquitetura busca proporcionar ao usuário final a percepção da utilização do sistema proposto como um sistema único e totalmente centralizado, ocultando as complexidades desta integração. Para este fim, foi realizado um estudo sobre as características das redes *Blockchain*, apresentando benefícios na implementação em sistemas de Internet das Coisas e um estudo dos tipos de plataformas e dos protocolos de comunicação mais utilizados. Para avaliar a arquitetura proposta, foi realizado estudo de caso que buscou analisar a percepção do usuário final na utilização da proposta e o nível de transparência aplicado.
Palavras-chave: Internet das coisas; Blockchain; Sistemas distribuídos; Transparência.

Abstract
The Internet of Things has created a world connected through many different types of sensors, allowing objects from our everyday lives to be networked. However, as the amount of data generated by these devices increases, the risk of theft and tampering becomes even more relevant. To improve the level of security within the Internet of Things, Blockchain presents itself as a potential solution for data integrity. However, realizing the connection between IoT devices and Blockchain is a task of reasonable complexity. In this context, this work proposes an architecture for transparency of connectivity of Blockchain services in IoT Systems. This architecture seeks to provide the end user with the perception of using the proposed system as a single and fully centralized system, hiding the complexities of this integration. To this end, a study was conducted on the characteristics of Blockchain networks, presenting benefits in the implementation in Internet of Things systems and a study of the types of platforms and the most commonly used communication protocols. To evaluate the proposed architecture, a case study was conducted to analyze the perception of the end user in the use of the proposal and the level of transparency applied.
Keywords: Internet of things; Blockchain; Distributed systems; Transparency.

Resumen
La Internet de los objetos ha creado un mundo conectado a través de los más diversos tipos de sensores, permitiendo que los objetos de nuestra vida cotidiana estén interconectados a la red. Sin embargo, a medida que aumenta la cantidad de datos generados por estos dispositivos, el riesgo de que sean robados y manipulados se vuelve aún más relevante. Para mejorar el nivel de seguridad dentro de la Internet de las cosas, Blockchain se presenta como una

solución potencial para la integridad de los datos. Sin embargo, la realización de la conexión entre los dispositivos IoT y Blockchain es una tarea de razonable complejidad. En este contexto, este trabajo propone una arquitectura para la transparencia de la conectividad de los servicios Blockchain en los sistemas IoT. Esta arquitectura pretende proporcionar al usuario final la percepción de utilizar el sistema propuesto como un sistema único y totalmente centralizado, ocultando las complejidades de esta integración. Para ello, se ha realizado un estudio sobre las características de las redes Blockchain, presentando beneficios en la implementación en sistemas de Internet de las Cosas y un estudio de los tipos de plataformas y los protocolos de comunicación más utilizados. Para evaluar la arquitectura propuesta, se realizó un estudio de caso que buscaba analizar la percepción del usuario final en el uso de la propuesta y el nivel de transparencia aplicado.
Palabras clave: Internet de las cosas; Blockchain; Sistemas distribuidos; Transparencia.

1. Introdução

Internet das Coisas (*Internet of Things* - IoT) é definida como a interconexão de dispositivos do nosso cotidiano com a Internet por meio de identificadores únicos (Id) e de suas respectivas propriedades, o que possibilita realizar a coleta de informações sobre o dispositivo e seu ambiente, assim como operações de alterações no seu estado atual a partir de qualquer lugar (Kim *et al*., 2017).

A IoT oferece possibilidades de integração entre os dispositivos, contudo, a segurança das informações coletadas por estes é um grande desafio. Com bilhões de dispositivos conectados, gerando informações com níveis variados de importância, a ameaça de roubo dessas informações é um fator relevante. Desta forma, é imprescindível construir soluções que proporcionem níveis de segurança apropriados para os dispositivos IoT que se comunicam através de redes de comunicação.

Neste contexto surge o *Blockchain*, definido como sistema de cadeia de blocos criptografados, que oferece recursos relevantes para o fornecimento de privacidade e segurança distribuídas na Internet. A estrutura do *Blockchain* permite a criação de um banco de dados inviolável configurado em uma rede *peer-to-peer* (Carrion & Quaresma, 2019). Diante deste cenário, o *Blockchain* se torna um grande aliado para os dispositivos IoT, oferecendo serviços de segurança de dados como imutabilidade e não-repudiação.

Contudo, um fato relevante neste cenário é que os protocolos iniciais de redes *Blockchain* foram desenvolvidos para serem aplicados nas transações de criptomoedas e, portanto, a utilização no cenário da Internet das Coisas não foi seu objetivo primário (Panarello *et al*., 2018). Os desafios para realizar a implementação neste contexto tem motivado a proposição de novos trabalhos, os quais visam em sua grande maioria mitigar os problemas de interoperabilidade entre os dispositivos IoT e o *Blockchain*. Pode-se afirmar também que existe a necessidade de integração dessas soluções com o usuário final, uma vez que dificuldades de configuração ou usabilidade tendem a aumentar a não aceitação da solução pelo mercado consumidor (Dedeoglu *et al*., 2020).

Diante das dificuldades de integração de dispositivos IoT e *Blockchain*, o desenvolvimento de um modelo arquitetural voltado para este fim pode proporcionar um serviço de integração transparente dessas tecnologias para o usuário final. Desta forma, este trabalho visa propor uma arquitetura para transparência de conectividade de serviços *Blockchain* em sistema IoT. Esta arquitetura objetiva diminuir a complexidade de integração para o usuário final, possibilitando uma melhor experiência no uso de dispositivos IoT em conjunto com *Blockchain*.

2. Trabalhos Relacionados

Atualmente, existem diversas propostas de uso de *Blockchain* para sistemas IoT com o intuito de resolver os problemas de segurança dos dados. Diante desta necessidade, alguns autores trazem em suas pesquisas estas propostas como uma solução para integridade, confiabilidade e segurança das informações geradas dentro do sistema IoT.

O autor Dorri *et al*. (2017), onde o mesmo se baseia em um modelo arquitetural de *Blockchain* privado aplicado em residências, gerenciando as informações externas e internas da casa. O trabalho apresenta como diferencial o uso do minerador interno, responsável por validar as informações de entrada no sistema *Blockchain*, tendo como os principais fundamentos a confiabilidade, a imutabilidade e a ininterruptibilidade do sistema.

As informações geradas dentro do sistema são acessadas apenas por usuários devidamente credenciados, garantindo assim que essas informações sejam privativas unicamente ao grupo de usuários da rede *Blockchain* privada. Para avaliar o desempenho, o sistema foi simulado em um cenário de casa inteligente no COOJA (simulador de aplicações do sistema operacional Contiki), utilizando três sensores mote Z1 (que imitam dispositivos IoT domésticos), enviando dados a cada 10 segundos.

As métricas aplicadas para avaliação deste trabalho foram sobrecarga de pacote (comprimento da transmissão do pacote), a sobrecarga de tempo (tempo de processamento para cada transação no minerador) e o consumo de energia (energia consumida pelo minerador para processar as transações). Os resultados mostraram um bom desempenho nos resultados obtidos a partir de experimentos indicando a viabilidade de uso da arquitetura proposta.

Já o autor Yu *et al*. (2018) propuseram como principal contribuição o projeto de uma plataforma *Blockchain* para dispositivos inteligentes, utilizando arquitetura de rede distribuída e o mapeamento inteligente entre os nós de dispositivos inteligentes na rede. A plataforma projeta um algoritmo de consenso *Blockchain* para os dispositivos IoT, fornecendo maior eficiência no consenso e garantindo a descentralização, como também, proporcionando maior prestabilidade.

Para o desenvolvimento deste trabalho, foi adotada uma arquitetura em três camadas (camada de dispositivos IoT, camada de *Blockchain* e camada DAPP). Os dados coletados do ambiente são inseridos e enviados da camada de dispositivos IoT para a camada de *Blockchain*, onde são compartilhados na rede que a compõem. Por fim, a camada DAPP (Aplicação Descentralizada) é responsável pelo fornecimento de serviços de interface de dados para os usuários da rede.

Em outra iniciativa de pesquisa relevante, o autor Pinno *et al*. (2017) propuseram uma arquitetura para controle de acesso baseada em *Blockchain*. A arquitetura é totalmente descentralizada, ou seja, sem necessidade de terceiros, escalável, transparente, tolerante a falhas e compatível com uma ampla gama de modelos de controle de acesso empregados na IoT.

A arquitetura denominada "ControlChain" foi desenvolvida a partir da junção de quatro arquiteturas de *Blockchains* diferentes e conectados entre si: o primeiro é o *Blockchain* de relacionamentos, que é responsável pelo armazenamento das credenciais públicas e relacionamentos de todas as entidades no sistema. Por sua vez, o *Blockchain* de contexto é responsável por registrar as informações obtidas a partir de sensores, dados processados e entradas manuais. Já o *Blockchain* tem por objetivo registrar as informações sobre permissões ou negar o acesso ao objeto. Por fim, o *Blockchain* de regras é aplicado para analisar, prestar contas, realizar auditoria de acessos e para verificar a estabilidade do sistema.

Para avaliar a arquitetura, os autores realizaram a comparação do ControlChain com outras arquiteturas (FairAccess, XACML, OAuth, UMA). As métricas utilizadas foram: *scalability* (avalia a capacidade de manipular uma quantidade crescente de dados de forma uniforme), *faulttolerant* (avalia o impacto causado por falhas em dispositivos ou links de comunicação), *no third-parties* (avalia dependência de terceiros na prevenção e a detecção de fraudes e interferências), *new authorization* (avalia a latência para fazer ou alterar uma autorização), *getauthorization* (avalia a latência para obter uma autorização), *integratedrelationship* (estuda a permissão de relacionamentos diretamente nas regras), *compatibility* (analisa a compatibilidade das arquiteturas com a abundância de modelos empregados atualmente na IoT), e, por fim, *lowobject overhead* (avalia o quanto o objeto está sobrecarregado pelo processo de autorização).

3. Solução Proposta

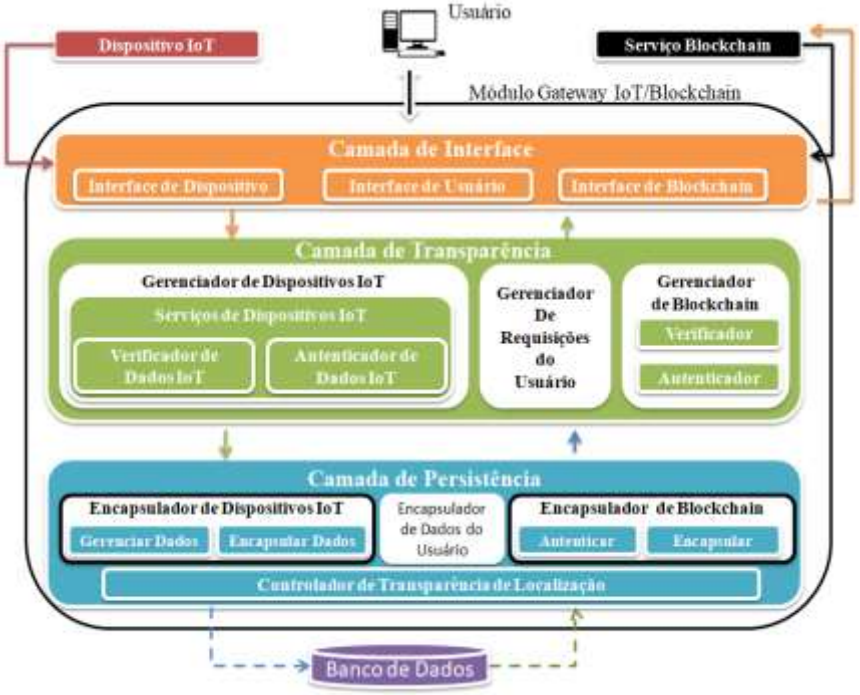
A arquitetura para transparência de conectividade de serviços *Blockchain* em Sistemas IoT proposta consiste em um Módulo instalado dentro de um Gateway IoT, onde serão disponibilizados serviços para serem consumidos por usuários/dispositivos. O módulo será capaz de receber dados de dispositivos inteligentes que se comunicam com o Gateway IoT e enviá-los para um fornecedor de serviços *Blockchain.*

A arquitetura proposta está organizada em camadas, onde cada camada desempenha um papel específico. Arquitetura em camadas é um sistema Cliente-Servidor onde as camadas que a compõem são separadas, podendo ser compreendida como um processo de decomposição do sistema (Estevão *et al*., 2018).

Neste contexto, este trabalho propõe a arquitetura ilustrada na Figura 1. Esta arquitetura se fundamenta em três camadas, dividindo suas funcionalidades em: Camada de Interface, responsável pela apresentação dos dados para o usuário final; Camada de Transparência, que especifica as regras de negócio e determina como os dados e as requisições serão utilizadas e, por fim, a Camada de Persistência, que armazena e/ou recupera dados quando solicitado.

Nesta arquitetura, também estão presentes componentes responsáveis pelo funcionamento, segurança e abstração da comunicação entre as camadas.

Figura 1 - Arquitetura Proposta.



Fonte: Autores.

3.1 Camada de interface

A camada de Interface tem por objetivo a interação com o usuário, exibindo as informações requisitadas. Nesta camada estão os clientes (usuários, dispositivos e *Blockchain*) ou aplicações que utilizarão os dados coletados através dos dispositivos IoT. Esses usuários podem ser páginas Web ou qualquer outro serviço habilitado com conexão à internet.

A interface do usuário "*user interface*" é o *frontend* onde o usuário final interage e realiza todas as ações do sistema, como cadastro e pesquisa dos dispositivos IoT, geração de relatórios e o cadastro dos dispositivos IoT em redes *Blockchain*, como também toda operação envolvendo a segurança e o acesso do usuário no sistema. As principais responsabilidades dos componentes desta camada são: apresentar uma interface inicial, para o acesso do usuário; receber requisições dos usuários;

receber informações dos dispositivos IoT cadastrados no sistema, tais como suas características (Id) e dados coletados (*Value*) e apresentar informações e relatórios aos usuários de forma personalizada.

3.2 Camada de transparência

A camada de Transparência é responsável por gerenciar as funcionalidades necessárias para conexões dos dispositivos IoT com os serviços de *Blockchain*, de forma que o usuário não tenha a percepção da complexidade e da estrutura dos métodos aplicados dentro desta camada, resultando em uma transparência para o usuário final.

Esta é a camada intermediária, onde são interpretados e processados os comandos realizados na camada de Interface, gerenciando as regras de negócios, como também realizando validações e autenticações dos dados de entrada, antes da camada de Transparência se comunicar com os métodos da camada de Persistência.

O conceito de transparências aplicado na arquitetura proposta é responsável por gerenciar e manipular os arquivos e métodos das classes onde estão inseridas, com o objetivo de ocultar a localização dos recursos da arquitetura, executando os processos de leitura e escrita de arquivos de forma remota ou local, sem que haja diferença no processo, como também, visando ocultar a localização física dos arquivos.

3.2.1 Transparência de acesso

A transparência de acesso objetiva ocultar para o usuário o modo no qual os recursos existentes dentro do sistema podem ser acessados (Bombardelli, 2010). Esta transparência é um atributo que está relacionado diretamente com o formato de desenvolvimento da arquitetura proposta, e este atributo está configurado de modo que todos os recursos dos dispositivos IoT possam ser acessados, ocultando as diferenças entre as funcionalidades e possibilitando transparência para o usuário.

Diante deste contexto, a comunicação do usuário com o sistema está relacionada exclusivamente com a camada de interface, onde o usuário fará as requisições de serviços para dentro do sistema. Contudo, a abstração aplicada ocultará o conhecimento das demais camadas distribuídas e, desta forma, os objetos e dados gerados serão acessados de forma local ou remota usando operações idênticas.

3.2.2 Transparência de localização.

A transparência de localização é aplicada para ocultar do usuário o lugar onde os recursos ou dados do sistema estão localizados (Coulouris *et al*., 2013). A terceira camada da arquitetura proposta (Camada de Persistência) é constituída por um conjunto de classes e métodos responsáveis em prover a criação, remoção, alteração e recuperação dos dados nos sistemas de gerenciamento de bancos de dados (SGBD). As classes e métodos são componentes do conceito de orientação a objetos que encapsulam a lógica de negócio para acessar uma fonte de dados (Ramos *et al*., 2004), tendo por objetivo centralizar as funcionalidades e desacoplar a infraestrutura da arquitetura do sistema proposto da tecnologia do sistema de gerenciamento de bancos de dados, conforme Figura 2.

Figura 2 - Arquitetura em três camadas desacopladas do banco de dados.

Fonte: Autores.

A camada de persistência fornece à camada de transparência a capacidade de acessar as funcionalidades e encaminhar os dados para a camada de interface. A vantagem deste desacoplamento entre a camada de persistência do sistema de gerenciamento de banco de dados (SGBD) é dificultar uma comunicação direta. Sendo assim, é possível serem realizadas modificações no SGBD "A" para outro SGBD "B" sem afetar a arquitetura, a camada de persistência e principalmente a percepção do usuário em relação à localização física do SGBD.

3.3 Camada de persistência

A camada de Persistência é responsável por gerenciar a manipulação dos dados dentro do sistema através da linguagem de manipulação de dados (ou DML, Data Manipulation Language). O DML é um conjunto de comandos dentro da linguagem de consulta estruturada (ou SQL, Structured Query Language), utilizado para recuperar, incluir, remover e modificar informações em bancos de dados (Ali & Shibghatullah, 2016).

Para realizar a manipulação dos dados a camada de persistência se conecta a um sistema de armazenamento de dados, onde através de um gerenciador de sistema de banco de dados é possível aplicar o DML. Estes gerenciadores de banco de dados podem ser MySql; Oracle; PostgreSql; MongoDB, dentre outros (Soares & Matos, 2017).

4. Metodologia

4.1 Implementação da arquitetura

Para o desenvolvimento da arquitetura para transparência de conectividade de serviços *Blockchain* em sistemas IoT foi aplicado o padrão *cliente Web thin*, que envolve padrões usados para sistemas com base na *internet* e sua principal característica é a configuração mínima do usuário, ou seja, o usuário necessita apenas de um navegador Web para acessar o sistema e toda lógica de negócio será executado dentro do servidor Web (De Lemos *et al.*, 2013).

Para implementar o modelo arquitetural, foi utilizado como linguagem de desenvolvimento o PHP versão 7 (sete), visto a flexibilidade de sua utilização, tanto no lado do cliente "*frontend*", quanto do lado do servidor "*backend*". Através da robustez desta linguagem de desenvolvimento, foi possível criar as camadas que compõem a arquitetura proposta, como também os módulos e componentes de execução. Sendo uma linguagem *opensource* aplicada a uso geral na Web (Dall'oglio, 2018), a linguagem de desenvolvimento PHP (*Hypertext Preprocessor*) é capaz de suportar grandes quantidades de dados, obtendo um bom desempenho na execução de muitas funções e consumindo muitos recursos ao mesmo tempo, sem comprometer o desempenho e a velocidade do servidor (Rotermel & Sommariva, 2016).

A compatibilidade com os principais banco de dados comerciais (MySQL, SQLite, Firebird, Interbase e Oracle) é uma grande vantagem na utilização da linguagem PHP em projetos Web (Teixeira & Catanduva, 2018), motivo pelo qual adotamos esta linguagem de programação como padrão. A estrutura de hierarquia dos diretórios e arquivos que caracteriza as três camadas da arquitetura proposta está descrita conforme a Figura 3. Esta estrutura está baseada no conceito de MVC; um acrônimo de "*Model – View – Controller*". O MVC é uma forma de estruturar os diretórios permitindo divisão das funcionalidades do sistema em camadas.

Figura 3 - Estrutura de Pacote da Arquitetura.

Fonte: Autores.

Dentro do diretório da pasta raiz (Gateway_Iot_Blockchain) estão contidos os principais arquivos de configurações e as respectivas camadas para o funcionamento do sistema proposto. Cada camada tem sua estrutura administrativa de comunicação e de gerenciamento. O arquivo index.php é responsável por iniciar a aplicação do sistema no navegador Web.

Por sua vez, a pasta config contém as configurações de conexão com o banco de dados que é utilizado para o armazenamento das informações sobre os dispositivos IoT, dados coletados pelos dispositivos e informações sobre a conectividade com o serviço *Blockchain*. Dentro da pasta auxiliar estão os arquivos de imagens utilizadas no sistema, como também, os arquivos de Java script (aplicação *frontend*) e arquivos para de estilo css (aplicação *frontend*).

A pasta Views é um pacote que contém arquivos dinâmicos criados para interagir com o usuário, realizando a visualização da informação de acordo com o tipo de requisições solicitadas pelo usuário final. Esta pasta representa a camada de interface. A camada de Transparência é representada pela pasta Controller. Esta pasta foi desenvolvida com objetivo de armazenar todos os métodos (procedimentos) da aplicação, com objetivo de receber e processar as requisições da camada views (Camada de Interface). Cada tipo de método contido dentro da pasta Controller possui uma função específica que realizará uma determinada ação.

Por fim, esta camada é responsável por fazer o intermédio entre a Camada de Persistência (pasta models) e a Camada de Interface (pasta views), contendo todos os métodos (procedimentos) necessários para receber, processar, autenticar e validar as requisições do usuário e/ou dispositivos. A pasta Models representa a Camada de Persistência e, esta pasta contém todos os métodos (procedimentos) para o recebimento das requisições da pasta Controller (Camada de Transparência). Através dos métodos da pasta Models são utilizados o conjunto de comandos SQL para gerenciar a persistência dos dados no banco de dados.

4.2 Implementação das transparências

A usabilidade da "Transparência", na arquitetura proposta, caracteriza-se:

4.2.1 Transparência de acesso

Na utilização do servidor Web Apache (Chen et al., 2004), foi instalado dentro do diretório *File Transfer Protocol* - FTP um arquivo denominado (.htaccess), permitindo o gerenciamento descentralizado das configurações. Este arquivo direciona e controla as páginas requisitadas pelo usuário e também configura a regravação da URL, desenvolvendo uma URL amigável. As configurações do arquivo (.htaccess), utilizadas na configuração da URL, são mostradas conforme a Figura 4.

Figura 4 - Script htaccess.

```
1    <?php
2    * .htaccess
3    RewriteEngine On
4    RewriteCond %{REQUEST_FILENAME} !-f
5    RewriteCond %{REQUEST_FILENAME} !-d
6    RewriteRule ^(.*)$ index.php/$1 [L]
7
```

Fonte: Autores.

Na configuração do arquivo .htaccess apresentado na Figura 4, foi inicialmente ativado o *engine* para a reescrita de URL, como consta na linha 3. Caso a requisição solicitada pelo usuário não seja um arquivo ou um diretório, a linha 4 e 5 realizará a negação de redirecionamento da página solicitada. Por fim, na linha 6, o script *RewriteRule* verifica o que foi solicitado pela URL e redireciona para a página index.php; no entanto, esse redirecionamento apenas será executado quando as regras da linha 4 e 5 estiverem satisfeitas.

Nesta configuração de acesso ao sistema, à função "*autoload*" irá carregar a classe do "Controller - Camada de Transparência" de forma automática, mostrando para o usuário o método contido na classe, ao invés do endereço da página requisitada, como mostra a Figura 5, linha 5. Desta forma, todos os endereços das páginas requisitadas pelo usuário serão ocultados e o que será mostrado em substituição serão os endereços de localização dos métodos das classes do Controller (Camada de Transparência).

Figura 5 - Script autoload.

```
1  <?php
2
3  // autoload.php @generated by Composer
4
5  require_once __DIR__ . '/composer/autoload_real.php';
6
7  return ComposerAutoloaderInit6fffeb2901d47a6fe38b2179fe3bd90c::getLoader();
```

Fonte: Autores.

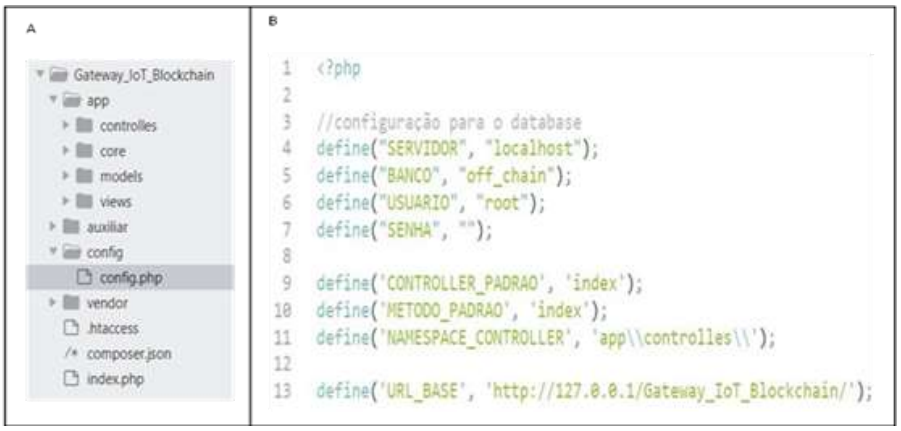
4.2.2 Transparência de localização

A utilização da transparência de localização consiste na forma de comunicação entre a própria arquitetura proposta com o sistema de gerenciamento de banco de dados - SGBD. A última camada da arquitetura proposta, a camada de persistência, possui métodos com diretrizes para realizar a comunicação com diferentes tipos de SGBD. Logo, a camada de persistência isola toda a comunicação com o Sistema de Gerenciamento de Banco de Dados – SGBD do restante das camadas.

Este isolamento garante que se houver qualquer modificação no tipo do SGBD, incluindo a forma de comunicação do SGBD, a camada de Interface não será afetada e o usuário não terá conhecimento da localização do armazenamento dos dados.

A Figura 6, Quadro A, apresenta a pasta "config", responsável por configurar o SGBD que irá se conectar com a arquitetura proposta. Esta pasta foi criada dentro da pasta raiz do sistema app, separado das camadas que compõem a arquitetura, para isolar a forma de conexão.

Figura 6 - Arquivo config /Conexão com SGBD.



Fonte: Autores.

Em relação ao Quadro B da Figura 6, o script ilustrado na linha 3 informa o nome do servidor que se conectará com sistema, por sua vez, a linha 4 informa o nome do banco de dados criado para armazenamento dos dados. As linhas 5 e 6 informam o login e a senha de acesso. Por fim, as linhas 8 até 11 configuram a URL que será acessada quando o arquivo index.php for chamado.

4.3 Implementação do blockchain

Com o objetivo de ilustrar a arquitetura proposta, um *Blockchain* foi desenvolvido exclusivamente para adaptar-se à estrutura desenvolvida para esta arquitetura. Para o desenvolvimento do *Blockchain* foi utilizado à linguagem de desenvolvimento PHP, contendo todas as características e padrões que validam uma rede *Blockchain*. Primeiramente foram definidos a estrutura dos blocos e o tipo do arquivo que irá armazenar o *Blockchain*, o JSON (modelo para armazenamento e transmissão de informações no formato texto).

Os atributos criados para cada bloco contém como características o *índice*, *hash*, *timestamp*, a prova de trabalho e o conteúdo. O Índice é um valor numérico inteiro, único e incremental, responsável por identificar cada bloco como único dentro da cadeia de blocos. O primeiro valor do *hash* do bloco é criado com um valor arbitrário. No entanto, os valores que irão compor o *hash* do bloco posterior serão calculados utilizando uma função específica. O *Timestamp* tem por objetivo especificar a data e o horário de criação de cada bloco. A prova de trabalho pode ser compreendida como o resultado da solução de um determinado problema criptografado e complexo utilizado para autenticar a transação pelos demais nós existentes na rede. Por fim, o conteúdo é um atributo que especifica os dados recebidos dos dispositivos IoT.

5. Avaliação

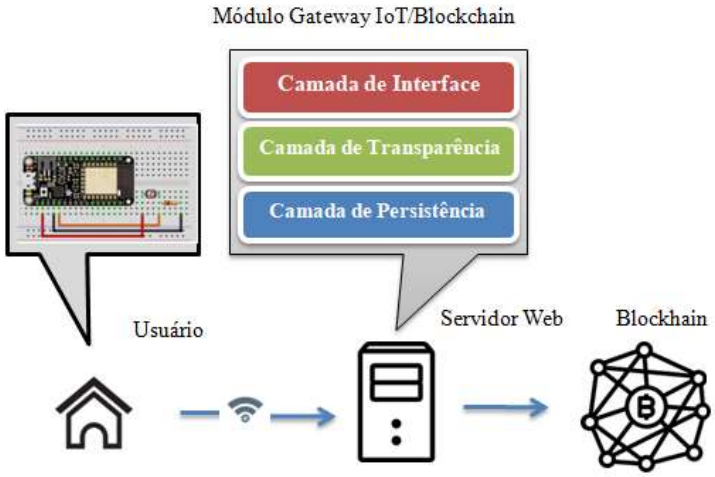
Neste capítulo, são apresentados dois estudos de casos que objetivam ilustrar e avaliar a utilização do modelo arquitetural proposto nesta pesquisa.

5.1 Estudo de caso 1: casa inteligência

Esta seção descreve como o modelo arquitetural proposto nesta pesquisa pode ser inserido em um cenário real com o intuito de ilustrar e avaliar o uso da referida arquitetura. Este estudo de caso tem por objetivo avaliar o comportamento estrutural do sistema como um todo, onde os resultados serão obtidos através da execução completa do sistema.

Esta execução completa do sistema consiste desde a captação dos dados coletados do ambiente através dos dispositivos IoT até ao sistema proposto, onde esses dados serão processados e enviados ao serviço *Blockchain*, finalizando assim, na criação da cadeia de blocos (*Blockchain*), como ilustrado na Figura 7.

Figura 7 - Estudo de Caso 1.



Fonte: Autores.

Os dados coletados através dos dispositivos IoT que se encontram nos cômodos A e B da casa inteligente são enviados para o servidor Web através do gateway. No servidor Web se encontram o sistema proposto como também a pasta com os arquivos contendo scripts de desenvolvimento do serviço *Blockchain*.

Dentro do sistema proposto, os dados enviados pelos dispositivos IoT são validados, como também, se verifica o cadastro dos dispositivos IoT no serviço *Blockchain* disponível. Após estes protocolos de validação e verificação, os dados são persistidos no banco de dados e enviados também para o *Blockchain* de acordo com a permissão de cada dispositivo.

5.1.1 Definição das coisas

Considerando recursos da casa inteligente como cenário de estudo de caso para ilustrar a implantação do modelo arquitetural, em dois cômodos da casa inteligente foram instalados dois sensores de luminosidade ("*Light Dependent Resistor* ou Resistor Dependente de Luz – LDR) com o objetivo de ligar de forma automática as lâmpadas dos respectivos cômodos.

A aplicação dos sensores LDR no cenário do estudo de caso é motivada pelo objetivo de economia de energia e também pelo desejo de monitoramento e controle da luminosidade nos cômodos. Ao ler a incidência da luz dentro dos cômodos, os sensores LDR enviam os dados coletados para dois módulos NodeMCU ESP-12, que estão conectados em placas protoboards distintas. O intervalo de cada leitura nos sensores, diante do ambiente inserido, foi estabelecido em escala de tempo de 1000 milissegundos com trinta iterações para cada experimento, com o intuito de avaliar o comportamento e estrutura deste cenário.

5.1.2 Execução e avaliação

Conforme pode ser visto na Figura 8, após a execução de todos os processos para a realização dos experimentos na estrutura da arquitetura proposta obtive como resultado a cadeia de blocos (*Blockchain*) gerada contendo o primeiro bloco denominado gênesis.

Figura 8 - Cadeia de Blocos (Blockchain).


```
{
    "chain": [
        {
            "nonce": 0,
            "index": 0,
            "timestamp": 1483225200,
            "data": "Genesis Block",
            "previousHash": null,
            "hash": "34054c41850bdf3f6f62dd60ae824dd35b98c872d420ed0dde942d0e54cd94d6"
        },
        {
            "nonce": 35868,
            "index": 1,
            "timestamp": 1605365467,
            "data": "1",
            "previousHash": "34054c41850bdf3f6f62dd60ae824dd35b98c872d420ed0dde942d0e54cd94d6",
            "hash": "000063e411e0742b191a3c0468db9faadc5bb20224e49a86d654d60256fec2dc"
        },
```

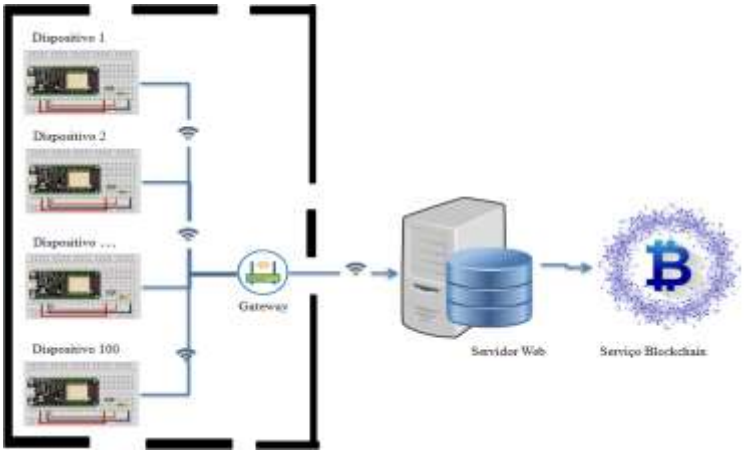

Fonte: Autores.

5.2 Estudo de caso 2: avaliação de desempenho

A avaliação de desempenho é realizada aplicando cargas de trabalho em um ambiente virtualizado, com 4 tipos diferentes de clientes virtuais criados sobre a plataforma JMeter.

Para esta avaliação, serão adotados dois cenários. No primeiro cenário, o estudo de caso é executado sem a solução proposta (caso base), ou seja, sem a utilização do Módulo Gateway IoT/Blockchain. Já no segundo cenário, o estudo de caso é executado utilizando a solução proposta nesta pesquisa (Módulo Gateway IoT/Blockchain), se efetuando inclusive a conexão com o *Blockchain*. O objetivo é comparar os dois cenários para mensurar a sobrecarga de desempenho imposta pela solução. Com isto, será possível analisar se a sobrecarga imposta se encontra em um limite aceitável tendo em vista a percepção do usuário. Assim, o primeiro cenário consiste em avaliar o tempo de execução na geração e envio dos dados dos dispositivos IoT até o Servidor Web, onde os dados serão processados e enviados para o serviço *Blockchain*, como ilustra a Figura 9.

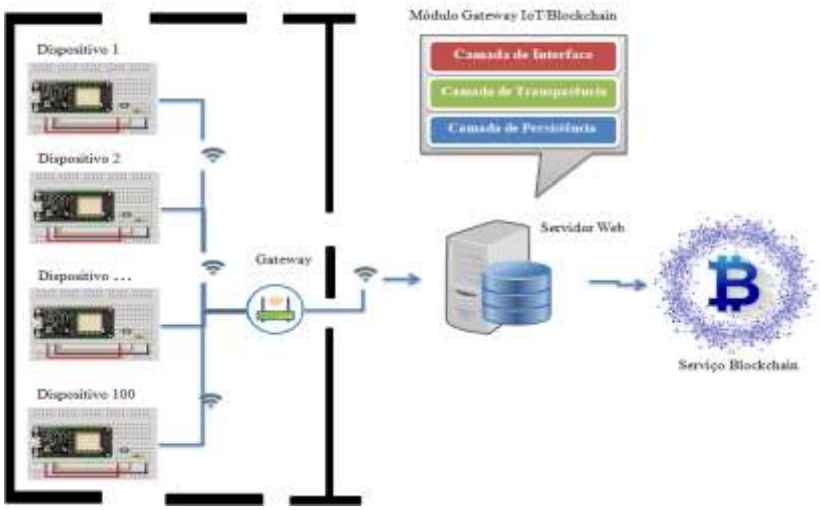
Figura 9 - Primeiro Cenário.



Fonte: Autores.

Por sua vez, o segundo cenário consiste em avaliar o tempo de execução usando a solução proposta nesta pesquisa, onde os dados serão processados e enviados para o serviço *Blockchain* por meio do Módulo Gateway IoT/Blockchain, como mostra a Figura 10.

Figura 10 - Segundo Cenário.



Fonte: Autores.

A ideia dos dois cenários é avaliar a sobrecarga de desempenho imposta pela solução proposta nesta pesquisa. Desta forma, o primeiro cenário não irá utilizar a solução, e o seu desempenho será medido. Depois, o segundo cenário é executado, e a solução é utilizada. Por fim, se avalia a diferença de desempenho observada nos dois cenários para poder avaliar se a solução proposta impõe uma sobrecarga de desempenho dentro de um limite aceitável.

Segundo os autores (Sarzosa Choque, 2019), o principal requisito que influencia negativamente os usuários em navegações de páginas da Web é o tempo de resposta (execução) das requisições realizadas. O tempo de resposta impacta diretamente na aceitação do usuário final se o valor no tempo de resposta for superior a 5 (cinco) segundos.

Devido ao conjunto de critérios estabelecidos, a métrica utilizada para avaliar o desempenho foi o tempo de execução. O tempo de execução é o tempo necessário para o atendimento de uma solicitação do usuário, onde sua unidade de tempo representativo é segundos.

5.2.1 Execução e avaliação

Por cumprir os requisitos necessários para a execução da carga (requisições) nos dois experimentos, foi utilizada a ferramenta JMeter na versão 5.2.1. O teste de carga do experimento tem por objetivo avaliar se os resultados estão de acordo com o esperado, garantindo assim níveis aceitáveis de qualidade de serviço ao usuário final (Sarzosa Choque, 2019).

Os valores aplicados para ambos os experimentos com clientes virtuais são de quantidades diferenciadas, no valor de (1, 25, 50, 100) clientes, com o intuito de encontrar um limite de desempenho do sistema proposto. A execução dos experimentos foi realizada primeiramente com um cliente virtual disparando uma requisição por vez, sinalizando uma iteração. Este processo foi realizado até completar o valor de 30 iterações. Seguindo esta forma de coleta de dados, o processo para aquisição dos dados coletados dos demais clientes virtuais foi realizado de forma semelhante. O tempo utilizado para a inicialização das requisições foi no valor de 2 milissegundos.

O número de requisições realizadas para os experimentos pelos clientes virtuais é respectivamente igual à quantidade de clientes virtuais utilizados, como ilustra a Tabela 1.

Tabela 1 - Valores de Carga.

Clientes Virtuais	Requisições	Iterações	Tempo de Inicialização
01	01	30	2 milissegundos
25	25		
50	50		
100	100		

Fonte: Autores.

Foi pré-definida a quantidade de 30 repetições/iterações para cada variação de cliente virtual (Dispositivos IoT) aplicado na simulação, com o objetivo de observar o comportamento da arquitetura proposta sobre altas demandas, de forma que seja representada sua utilização em situação normal de uso. Segundo os autores (De Morais et al., 2020) a quantidade de 30 repetições traz um intervalo de confiança maior nos resultados, ficando assim mais próximo de uma situação real.

5.2.2 Resultados

Na execução da avaliação de desempenho, foi definido o uso de diferentes grupos de clientes virtuais para prover os resultados desejados. Com isso, foi separado um servidor local isolado para ambos os experimentos utilizados na coleta dos dados. Este isolamento do servidor local foi para evitar interferências e alterações nos resultados coletados.

Os resultados referentes à análise do tempo de execução utilizando o protocolo HTTP foram medidos e avaliados de acordo com a quantidade de clientes virtuais, requisições e iterações já definidas na Tabela 1. A Tabela 2 apresenta o resultado geral dos experimentos realizados em escala crescente usando como referência os clientes virtuais para o primeiro cenário.

Tabela 2 - Tempo de execução medido do primeiro cenário (caso base).

Clientes Virtuais	Iterações	Tempo de Resposta Média (ms)
01	30	23
25	30	150,1
50	30	309,2
100	30	813

Fonte: Autores.

De acordo com os dados apresentados na Tabela 2, é possível obter o valor da média quando aplicado 30 interações com um cliente virtual (dispositivos IoT) obtendo assim o valor resultante de 23 milissegundos. Porém, quando realizado os experimentos com o valor de vinte e cinco clientes virtuais o resultado da média do tempo de execução foi de 150,1 milissegundos. Sendo o valor de clientes virtuais aumentado para cinquenta, a média obtida através das 30 iterações foi de 309,2 milissegundos.

Por fim, foi estabelecido o experimento com cem clientes virtuais, e a média resultante foi no valor de 813 milissegundos. É possível identificar que, quando analisada a quantidade com cem clientes virtuais, o valor resultante da média está dentro do aceitável em comparação com o valor de tempo de execução aplicado a sistemas Web utilizando o protocolo http, que é de 5000 milissegundos (Chen et al., 2004).

A Tabela 3 apresenta o resultado geral dos experimentos realizados aplicado ao segundo cenário, que consiste no ciclo completo da arquitetura proposta (módulo gateway iot/blockchain) em escala crescente usando como referência os clientes virtuais.

Tabela 3 - Tempo de execução medido do segundo cenário.

Clientes Virtuais	Iterações	Tempo de Resposta Média (ms)
01	30	45,1
25	30	265,2
50	30	475,2
100	30	1501,5

Fonte: Autores.

Na primeira execução realizada com um cliente virtual (dispositivo IoT) utilizando 30 iterações, é possível identificar a média geral no valor de 45,1 milissegundos. Quando o valor aplicado aumentou para vinte e cinco clientes virtuais realizando 30 iterações, o valor gerado foi uma média de 265,2 milissegundos.

Pode-se afirmar que o valor obtido é tolerável para uma aplicação Web. Adicionalmente, foram realizadas outras duas simulações da mesma operação, aumentando a quantidade de clientes virtuais para o valor de cinquenta e cem clientes. Os valores resultantes para os experimentos com cinquenta e cem clientes virtuais a 30 iterações foram respectivamente 475,2 e 1501,5 milissegundos.

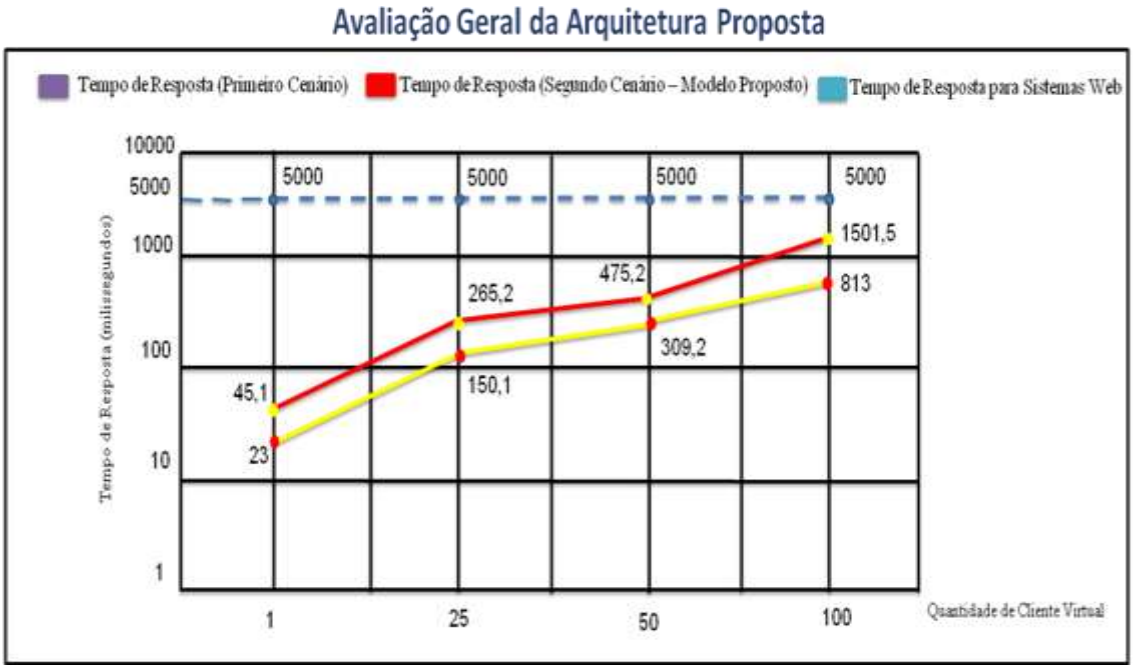
Ao se analisar o resultado geral nos dois cenários realizados, os valores obtidos dos experimentos com apenas um cliente virtual (primeiro grupo) tiveram um aumento na variação entre eles de aproximadamente 1 (um) segundo. Contudo, esse valor é compreensivo, devido a todo o processo de validação e autenticação em que os dados são submetidos dentro do sistema proposto. Já na análise geral com vinte e cinco clientes virtuais (segundo grupo) também houve um aumento na variação no valor de aproximadamente 1 (um) segundo.

Quando analisado os valores gerais obtidos com cinquenta clientes virtuais (terceiro grupo) foi percebido também um aumento aproximado na variação de 1/2 (meio) segundo, contudo, foi verificado que o valor de variação foi menor em relação ao valor obtido com vinte e cinco clientes virtuais, porém, o aumento entre as variações analisadas nos três primeiros grupos de clientes virtuais, pode indicar um padrão de comportamento com tendência de crescimento.

Por fim, o valor aproximado do aumento na variação com cem clientes virtuais (quarto grupo) foi de 1 (um) segundo, demonstrando o início de uma tendência de disparidade mais acentuada a partir desta quantidade de clientes (dispositivo IoT) e projetando um aumento no valor da variação com o terceiro grupo de cliente virtual.

É importante ressaltar que o maior valor obtido no segundo cenário (Módulo Gateway IoT/Blockchain) foi de 1501,5 milissegundos, indicando que o resultado está abaixo do valor usado como referência para aplicação em métricas de QoS para navegação Web em sistemas convencionais utilizando o protocolo HTTP, que é de 5000 milissegundos, o que pode ser considerado um aspecto positivo, como ilustra a Figura 11.

Figura 11 - Avaliação Geral.



Fonte: Autores.

6. Considerações Finais

O crescimento tecnológico vem se mostrando acelerado e a tecnologia das redes *Blockchain* tem se mostrado uma alternativa interessante para prover para os sistemas de Internet das Coisas é uma alternativa para melhorar a segurança e a imutabilidade dos dados gerados. No entanto, a integração entre essas duas tecnologias apresenta desafios não triviais.

Tendo em vista a complexidade desta integração, esta pesquisa propôs uma arquitetura para transparência de conectividade de serviços *Blockchain* em sistemas IoT, de forma que o usuário final possa utilizar serviços *Blockchain* em seus sistemas IoT de forma facilitada.

A arquitetura proposta foi estruturada baseada na hierarquia em camadas, onde cada camada da arquitetura implementa uma parte do sistema. A solução proposta permite que dispositivos IoT enviem dados coletados do ambiente em que estão para um serviço *Blockchain*. A arquitetura em três camadas promoveu a separação das funcionalidades, permitindo a aplicação da transparência de acesso, possibilitando a ocultação na diferença da representação de dados e no modo de acesso ao usuário final, como também, contribuiu na inserção da transparência de localização, ocultando onde os recursos do sistema e os dados estão sendo armazenados.

A partir da execução e análise dos estudos de casos, foi possível observar que o desenvolvimento de uma arquitetura para transparência de conectividade de serviços *Blockchain* em sistemas IoT pode ser considerada uma contribuição interessante, pois diante dos resultados obtidos foi possível observar que, em ambiente controlado, o sistema proposto obteve valores de desempenho compatíveis com o aceitável para sistemas Web convencionais.

Adicionalmente, através da análise do estado da arte descrito no Capítulo 2, pode ser observado que nenhum dos trabalhos referenciados nesta pesquisa tem foco na proposição de uma arquitetura para transparência de conectividade de serviços *Blockchain* em sistemas IoT. Este fato evidencia a importância da contribuição científica deste trabalho científico.

Diante das melhorias futuras que podem ser realizadas na pesquisa desenvolvida, se vislumbram ampliar os estudos para análise e introdução das demais transparências aplicadas em sistemas distribuídos, como também, realizar experimentos para conectividade de serviços *Blockchain* utilizadas no mercado, como a Ethereum.

Ainda como trabalho futuro é possível realizar estudos com o objetivo de implantar o sistema proposto diretamente dentro do Gateway IoT, para abstrair do usuário a sensação de descentralização do sistema, dado que, o Módulo Gateway IoT/Blockchain poderá ser instalado dentro do Gateway IoT e o banco de dados em um servidor qualquer e, por fim, promover avaliação da arquitetura proposta com vários cenários de cargas, com o objetivo de tentar extrapolar os limites julgados aceitáveis de tolerância ao atraso da entrega de dados.

Referências

Ali, N. S., & Shibghatullah, A. S. (2016). Protection web applications using real-time technique to detect structured query language injection attacks. *International Journal of Computer Applications*, *149*(6), 26-32.

Bombardelli, F. G. URF (Framework Unificado de Robótica): proposta de interface para sistemas distribuídos.

Carrion, P., & Quaresma, M. (2019). Internet da Coisas (IoT): Definições e aplicabilidade aos usuários finais. *Human Factors in Design*, *8*(15), 049-066.

Chen, Y., Farley, T., & Ye, N. (2004). QoS requirements of network applications on the Internet. *Information Knowledge Systems Management*, *4*(1), 55-76.

Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2013). *Sistemas Distribuídos-: Conceitos e Projeto*. Bookman Editora.

Dall'Oglio, P. (2018). *PHP Programando com orientação a Objetos*. Novatec Editora.

de Lemos, M. F., Oliveira, P. C., Ruela, L. C., da Silva Santos, M., Slveira, T. C., & de Sousa Reis, J. C. (2013). Aplicabilidade da arquitetura MVC em uma aplicação web (WebApps). *RE3C-Revista Eletrônica Científica de Ciência da Computação*, *8*(1).

de Morais, A. M., de Almeida Callou, G. R., & Lins, F. A. A. (2020). Simulação e Avaliação de Desempenho de uma Rede Blockchain Utilizando Containers Docker. *Cadernos do IME-Série Informática*, *44*, 73-87.

Dedeoglu, V., Jurdak, R., Dorri, A., Lunardi, R. C., Michelin, R. A., Zorzo, A. F., & Kanhere, S. S. (2020). Blockchain technologies for iot. In *Advanced Applications of Blockchain Technology* (pp. 55-89). Springer, Singapore.

Dorri, A., Kanhere, S. S., Jurdak, R., & Gauravaram, P. (2017). Blockchain for IoT security and privacy: The case study of a smart home. In *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)* (pp. 618-623). IEEE.

Estevão, B. D. S., Sandmann, A., & Santos, I. B. D. (2018). Aplicações ricas para Internet: proposta de Arquitetura de software na nuvem para atender ao Agronegócio.

Kim, T. H., Ramos, C., & Mohammed, S. (2017). Smart city and IoT.

Panarello, A., Tapas, N., Merlino, G., Longo, F., & Puliafito, A. (2018). Blockchain and iot integration: A systematic survey. *Sensors*, *18*(8), 2575.

Pinno, O. J. A., Gregio, A. R. A., & De Bona, L. C. (2017). Controlchain: Blockchain as a central enabler for access control authorizations in the iot. In *GLOBECOM 2017-2017 IEEE Global Communications Conference* (pp. 1-6). IEEE.

Ramos, R. A., Camargo, V., Penteado, R., & Masiero, P. C. (2004). Reuso da implementaçao orientada a aspectos do padrao de projeto camada de persistência. In *The Fourth Latin American Conference on Pattern Languages of Programming-SugarLoafPLoP, Fortaleza-CE* (Vol. 27).

Rotermel, F., & Sommariva, L. W. (2016). Inovações advindas na nova versão da linguagem de programação web PHP 7.0. *Revista Interdisciplinar Científica Aplicada*, *10*(4), 1-20.

Sarzosa Choque, A. I. (2019). *Performance Testing A Web Services Con JMETER* (Doctoral dissertation).

Soares, A. S. S., & Matos, P. F. (2017). Uma Análise Comparativa entre Sistemas Gerenciadores de Bancos de Dados NoSQL no contexto de Internet das Coisas. In *SBBD (Short Papers)* (pp. 306-311).

Teixeira, M. A., & Catanduva, S. P. (2018). Servidor WEB Apache.

Yu, S., Lv, K., Shao, Z., Guo, Y., Zou, J., & Zhang, B. (2018, August). A high performance blockchain platform for intelligent devices. In *2018 1st IEEE international conference on hot information-centric networking (HotICN)* (pp. 260-261). IEEE.