

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
CURSO DE ESPECIALIZAÇÃO EM INTERNET DAS COISAS

CLEVERSON FERREIRA FRANCO

**TECNOLOGIA EM NUVEM APLICADA A INTERNET DAS COISAS**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA

2018

CLEVERSON FERREIRA FRANCO

## **TECNOLOGIA EM NUVEM APLICADA A INTERNET DAS COISAS**

Monografia de Especialização, apresentada ao Curso de Especialização em Internet das Coisas, do Departamento Acadêmico de Eletrônica – DAELN, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Dr. Glauber Gomes de Oliveira Brante

CURITIBA  
2018



Ministério da Educação  
Universidade Tecnológica Federal do Paraná  
Câmpus Curitiba

Diretoria de Pesquisa e Pós-Graduação  
Departamento Acadêmico de Eletrônica  
Curso de Especialização em Internet das Coisas



---

## TERMO DE APROVAÇÃO

TECNOLOGIA EM NUVEM APLICADA A INTERNET DAS COISAS

por

CLEVERSON FERREIRA FRANCO

Esta monografia foi apresentada em 28 de Novembro de 2018 como requisito parcial para a obtenção do título de Especialista em Internet das Coisas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dr. Glauber Gomes de Oliveira Brante  
Orientador

---

Prof. M. Sc. Danillo Leal Belmonte  
Membro titular

---

Prof. M. Sc. Omero Francisco Bertol  
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

Dedico este trabalho à Leni e Clóvis.

## **AGRADECIMENTOS**

A Deus pela capacidade, e oportunidade.

Aos professores Glauber Gomes de Oliveira Brante, Omero Francisco Bertol, e Danillo Leal Belmonte, pela paciência, e apoio para a concretização desta etapa em minha vida.

## RESUMO

FRANCO, Cleverson Ferreira. **Tecnologia em nuvem aplicada a Internet das Coisas**. 2018. 31 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

A nuvem já vem sendo muito usada para hospedar aplicações web nos últimos anos, por sua segurança, velocidade, e confiabilidade. Com a crescente onda de internet das Coisas chegando é necessário evoluir com a tecnologia na camada de infraestrutura, e tecnologia web, para que uma aplicação de IoT tenha um ecossistema completo. O estudo teve como objetivo construir a parte web de uma aplicação, de forma segura, de alta performance, monitorada e com alta disponibilidade, utilizando de novas tecnologias tanto na parte estrutural usando sistema de filas, hospedagem em nuvem, monitoramento em tempo real, quanto na parte de código, onde foi utilizado o PHP, uma das linguagens de programação mais usadas no mundo. O resultado obtido atingiu as metas esperadas, pois resultou em uma plataforma simples, organizada, e de fácil gerenciamento, além de adaptável a novas melhorias e implementações futuras.

**Palavras-chave:** Internet das Coisas. Software. Infraestrutura. Nuvem. Disponibilidade.

## ABSTRACT

FRANCO, Cleverson Ferreira. **Cloud technology applied to the Internet of Things**. 2018. 31 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

The cloud has long been used to host web applications in recent years for its security, speed, and reliability. With the growing Internet wave of Things coming up, it is necessary to evolve with technology in the infrastructure layer, and web technology, so that an IoT application has a complete ecosystem. The study aimed to build the web part of a secure, high performance, monitored and high availability application using new technologies in both the structural part using queuing system, cloud hosting, real-time monitoring, and in the part of code, where PHP was used, one of the most used programming languages in the world. The result achieved reached the expected goals, as it resulted in a simple, organized, easy-to-manage platform, and adaptable to new improvements and future implementations.

**Keywords:** Internet of Things. Software. Infrastructure. Cloud. Availability.

## LISTA DE ILUSTRAÇÕES

Figura 1. Entendendo computação em nuvem.....	15
Figura 2. Tipos de distribuição de mensagem suportados pelo protocolo MQTT.....	17
Figura 3. Tela de registro de usuário na plataforma web .....	20
Figura 4. Tela de login de usuário na plataforma web.....	20
Figura 5. Dashboard do usuário na plataforma web.....	21
Figura 6. Tela de novo local na aplicação web.....	21
Figura 7. Wifi da placa sendo listada nas redes disponíveis .....	22
Figura 8. Tela inicial de configuração da placa .....	22
Figura 9. Tela com configurações de acesso preenchidas .....	22
Figura 10. Botão de controle de dispositivo disponibilizado após cadastro de placa	24
Figura 11. Esquema de configuração entre firmware, fila, e aplicação web.....	24
Figura 12. Esquema de funcionamento do broker MQTT .....	25
Figura 13. Droplets no painel Digital Ocean.....	26
Figura 14. Informações de Droplet no painel Digital Ocean .....	26
Figura 15. Histórico de pagamento de droplets na Digital Ocean .....	27
Gráfico 1. Fluxograma de comunicação entre placa e aplicação em nuvem .....	23

## LISTA DE ABREVIATURAS

Kwh	Quilowatts/hora
LED	<i>Light Emitting Diode</i>
USD	<i>United States Dollar</i>

## LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
BSDs	<i>Berkeley Software Distribution</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
M2M	<i>Machine-to-Machine</i>
MVC	<i>Model-View-Controller</i>
MySQL AB	<i>My Structured Query Language AB</i>
MySQL	<i>My Structured Query Language</i>
SGBD	<i>Sistema de Gestão de Bases de Dados</i>
SQL	<i>Structured Query Language</i>
SSL	<i>Secure Socket Layer</i>
TCP	<i>Transmission Control Protocol</i>
XML	<i>Extensible Markup Language</i>

## LISTA DE ACRÔNIMOS

IoT	<i>Internet of Things</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
PHP	<i>Hypertext Preprocessor</i>

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>12</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>15</b>
2.1 INFRAESTRUTURA EM NUVEM.....	15
2.2 MYSQL .....	16
2.3 MQTT.....	16
2.4 LINGUAGEM DE PROGRAMAÇÃO.....	17
2.4.1 PHP.....	17
2.5 ARDUINO .....	18
<b>3 DESENVOLVIMENTO DA APLICAÇÃO .....</b>	<b>20</b>
3.1 ESTRUTURA WEB.....	20
3.1.1 Aplicação Web .....	20
3.1.2 Arduino.....	23
3.1.3 API .....	24
3.2 SISTEMA DE FILAS .....	25
3.2.1 Mosquitto.....	25
3.3 INFRAESTRUTURA .....	25
<b>4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS .....</b>	<b>27</b>
<b>5 CONCLUSÃO .....</b>	<b>29</b>
<b>REFERÊNCIAS.....</b>	<b>30</b>

## 1 INTRODUÇÃO

O aumento do volume de informações e o surgimento de novos tipos de dados provocou a necessidade do desenvolvimento de novas tecnologias com maior capacidade de processamento, maior capacidade de armazenamento, menor custo e menores dimensões físicas. Com a expansão e aperfeiçoamentos dessas tecnologias durante o tempo, esses sistemas computacionais passaram a fazer parte da vida das pessoas e das empresas mudando significativamente a forma de se trabalhar e se comunicar (PEREIRA; APPEL, 2013).

O termo computação em nuvem, segundo Taurion (2009), surgiu em 2006 em uma palestra de Eric Schmidt, da Google, sobre como sua empresa gerenciava seus data centers. Hoje, computação em nuvem, se apresenta como o cerne de um movimento de profundas transformações do mundo da tecnologia.

A nuvem é uma representação para a internet ou infra-estrutura de comunicação entre componentes arquiteturais, baseada em uma abstração que oculta à complexidade da infra-estrutura. Cada parte desta infraestrutura é provida como um serviço, e estes serviços são normalmente alocados em *data centers*, utilizando hardware compartilhado para computação e armazenamento (SOUSA; MOREIRA; MACHADO, 2009).

Após estudo onde foram consideradas várias definições distintas para o conceito de computação na nuvem, os autores de Vaquero et al. (2009) chegaram a definição de que nuvens são grandes repositórios de recursos virtualizados, tais como hardware, plataformas de desenvolvimento e software, que são facilmente acessíveis. Além disto, estes recursos podem ser configurados dinamicamente de modo a ajustar-se a diferentes cargas de trabalho com a intenção de otimizar sua utilização. O modelo de cobrança utilizado para a exploração destes repositórios está baseado em pagamento pelo uso.

A computação em nuvem, ou *Cloud Computing*, é um novo modelo de computação que permite ao usuário final acessar uma grande quantidade de aplicações e serviços em qualquer lugar e independente da plataforma, bastando para isso ter um terminal conectado (SILVA, 2011).

A palavra nuvem sugere uma idéia de ambiente desconhecido, o qual pode-se ver somente seu início e fim. Por este motivo esta foi muito bem empregada na

nomenclatura deste novo modelo, onde toda a infra-estrutura e recursos computacionais ficam “escondidos”, tendo o usuário o acesso apenas a uma interface padrão através da qual é disponibilizado todo o conjunto de variadas aplicações e serviços (SILVA, 2011).

Segundo Sousa, Moreira e Machado (2009), o modelo de computação em nuvem foi desenvolvido com o objetivo de fornecer serviços de fácil acesso e de baixo custo e garantir características tais como disponibilidade e escalabilidade. Este modelo visa fornecer, basicamente, três benefícios. O primeiro benefício é reduzir o custo na aquisição e composição de toda infra-estrutura requerida para atender as necessidades das empresas, podendo essa infraestrutura ser composta sob demanda e com recursos heterogêneos e de menor custo. O segundo é a flexibilidade que esse modelo oferece no que diz respeito à adição e troca de recursos computacionais, podendo assim, escalar tanto em nível de recursos de hardware quanto software para atender as necessidades das empresas e usuários. O terceiro benefício é prover uma abstração e facilidade de acesso aos usuários destes serviços. Com isso, os usuários dos serviços não precisam conhecer aspectos de localização física e de entrega dos resultados destes serviços.

No estudo realizado o foi montada uma estrutura em nuvem utilizando algumas das tecnologias web mais usadas na atualidade, que são:

- PHP: foi usado como linguagem de programação, por facilitar muito o desenvolvimento, ter um código simples, além de ser umas das linguagens mais usadas no mundo.
- Laravel: juntamente com o PHP usa-se o Framework Laravel, que ajudar a acelerar o desenvolvimento, além de manter o código padronizado.
- MySQL: como base de dados, o MySQL foi escolhido por ser o mais usado com PHP e por ter uma sintaxe de fácil entendimento.

Para a parte de IoT foram usadas as seguintes tecnologias:

- Mosquitto: é um sistema de filas que usa o protocolo MQTT, ele foi escolhido por ser open-source e ser um dos mais usados para gerenciamento de mensagens entre dispositivos de IoT.

- Arduino: para programar a placa que irá fazer a comunicação com a estrutura web, utiliza-se o Arduino como linguagem, por ter uma IDE muito prática e intuitiva, além de facilitar a integração com bibliotecas e usar o C++ como base.

Sendo assim, o objetivo do presente estudo, foi avaliar se é possível integrar essas tecnologias, como foram aplicadas, o custo para manter essa estrutura hospedada utilizando a tecnologia cloud, e quais os principais desafios e dificuldades para fazer com que todas essas estruturas se comuniquem de uma maneira eficiente, segura e sem perda de pacotes

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 INFRAESTRUTURA EM NUVEM

Computação em nuvem está associada ao novo paradigma da infraestrutura da computação, que passa a ser utilizada intensamente pela internet. Existem diversas definições para o que é computação em nuvem, existindo pontos de consenso e de dúvidas nas acepções existentes. De um modo simplificado, é um acesso transparente a recursos em um sistema de uso por demanda, baseado numa infraestrutura de elasticidade infinita e instantânea, provida por uma terceira parte ou fornecida internamente (BUYAA et al., 2009).

Para que seja compreendido o que é computação em nuvem, é necessário antes entender a evolução das tecnologias computacionais até o estágio atual. De uma maneira geral, este trabalho irá utilizar um modelo adaptado de Khorshed, Ali e Wasimi (2012), buscando explicar de uma maneira mais completa todas as características atreladas à tecnologia da computação em nuvem (Figura 1). Dentro da computação em nuvem existem sete conceitos que serão usados para explicar seus atributos. São eles: surgimento, modelo de entrega de serviço, modelos estruturais, papéis, camadas, localização, características, e pontos indefinidos.

Figura 1. Entendendo computação em nuvem



Fonte: adaptado de Khorshed, Ali e Wasimi (2012).

## 2.2 MYSQL

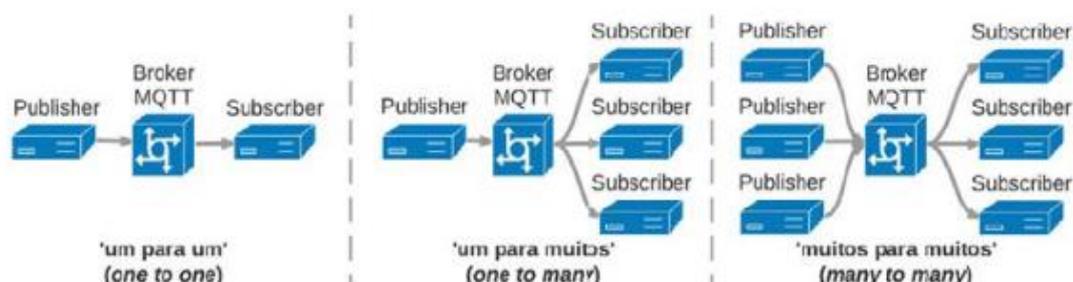
O MySQL surgiu a partir da necessidade da equipe que criou o SGBD (Sistema de Gestão de Bases de Dados), de utilizar algum mecanismo que permitisse a conexão de tabelas criadas na linguagem SQL (*Structured Query Language*) para um determinado fim. A princípio, o grupo iria utilizar o `mySQL`, mas logo perceberam que esta ferramenta não era rápida o suficiente para atender às necessidades do projeto. O jeito foi criar uma solução própria. Nascia o MySQL. O MySQL é um banco de dados relacional gratuito, eficiente e otimizado para aplicações Web, é desenvolvido e mantido pela empresa MySQL AB, que também oferece uma versão comercial (paga). Esse SGBD também é multi-plataforma, sendo compatível com o Windows, Linux, BSDs, entre outros sistemas operacionais. As tabelas criadas podem ter tamanho de até 4 GB. Fora isso, o MySQL é compatível com várias linguagens de programação, tais como PHP, C, Java, Visual Basic, entre outros (SANTOS; SILVA, 2018).

## 2.3 MQTT

Criado em 1999 pela IBM, MQTT (*Message Queuing Telemetry Transport*), é um protocolo aberto de mensagens projetado para comunicação M2M, na qual deve lidar com alta latência, instabilidade na comunicação e baixa largura de banda. O protocolo MQTT foi padronizado pelo OASIS em 2013 e atualmente está na versão 3.1.1 (MQTT, 2014), sendo livre de royalties desde 2010.

O protocolo MQTT adota o protocolo TCP (*Transmission Control Protocol*) e o padrão de mensagens *publisher/ subscriber* (publicador/ assinante), onde todos os dados são enviados para um intermediário, chamado broker, que se encarrega de enviar as mensagens aos destinatários corretos. Esta estrutura permite desacoplar o produtor do cliente, assim, apenas o endereço do broker precisa ser conhecido, possibilitando a comunicação de um para um (*one-to-one*, Figura 2), um para muitos (*one-to-many*, Figura 1) ou muitos para muitos (*many-to-many*, Figura 1). Já no quesito de segurança, até a versão 3.1.1, MQTT não implementa qualquer tipo de criptografia SSL (*Secure Socket Layer*) pode ser utilizado independentemente, existindo apenas um método de autenticação com nome de usuário e senha.

**Figura 2. Tipos de distribuição de mensagem suportados pelo protocolo MQTT**



Fonte: MQTT (2014).

## 2.4 LINGUAGEM DE PROGRAMAÇÃO

Para a estrutura web foi usado a linguagem de programação PHP aplicada a um Framework chamado Laravel. Juntos consegue-se manter um alto padrão de código.

### 2.4.1 PHP

PHP (*PHP Hypertext Preprocessor*), é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a web e pode ser mesclada dentro do código HTML. A sintaxe da linguagem lembra C, Java e Perl, e é fácil de aprender. O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente rapidamente, mas você pode fazer muito mais do que isso com PHP (PHP, 2018).

#### 2.4.1.1 Framework Laravel

O Laravel é um framework PHP utilizado para desenvolvimento web. Possui uma arquitetura MVC (*Model-View-Controller*) e tem, como aspecto principal, o de auxiliar no desenvolvimento de aplicações seguras e de alto desempenho de forma ágil e simplificada, com código limpo. Outra característica é o incentivo do uso de boas práticas de programação e a utilização de padrões específicos a ele determinados. Assim como outros frameworks, o Laravel possui testes já definidos e a estrutura possui métodos auxiliares convenientes que permitem testar expressamente suas aplicações. No diretório de testes podem ser encontrados testes de banco de dados, testes de navegador e testes que geram entradas de dados fictícios. Quando

executados os testes, devem ser definidas as variáveis e realizadas as chamadas dos dados que serão testados na aplicação (LARAVEL, 2018).

Segundo Verma (2014), a arquitetura MVC utilizada pelo Laravel é um padrão que visa aumentar a modularidade de sistemas de software, sendo apresentada em três camadas:

1. *Model*: gerencia os modelos de dados da aplicação, fazendo a interação com o banco de dados, analisando a lógica, os dados e as regras. É uma camada entre os dados e a aplicação, que pode armazenar diversos tipos de dados, de sistemas gerenciadores de bancos de dados, como o MySQL, ou arquivos *eXtensible Markup Language* (XML);
2. *View*: representa a camada de interação com o usuário por meio de interfaces. Diz respeito à representação da aplicação web e é responsável por mostrar os dados que a camada controller recebe da camada model. Pode ser implementado facilmente com o uso do pacote “.blade.php” ou com código PHP. O Laravel inicia sua execução com a extensão de arquivo (.blade.php ou .php) determinando quem deve prosseguir com a execução do modelo;
3. *Controller*: recebe as solicitações dos usuários através da view para a exibição ou atualização de dados e faz requisições na camada model de acordo com a solicitação correspondente. É considerado um link entre a model e a view e dispõe de duas opções de desenvolvimento da lógica: Router e Controller. Os Routers são mais viáveis para páginas web estáticas. Controllers são definições de escrita para cada página web.

## 2.5 ARDUINO

O Arduino é uma plataforma eletrônica de código aberto baseada em hardware e software fáceis de usar. Placas Arduino são capazes de ler entradas - luz em um sensor, um dedo em um botão ou uma mensagem no Twitter - e transformá-lo em uma saída - ativando um motor, ligando um LED, publicando algo on-line. Você pode dizer à sua placa o que fazer enviando um conjunto de instruções para o microcontrolador na placa. Para isso, você usa a linguagem de programação Arduino (baseada em Fiação ) e o Software Arduino (IDE) , baseado em Processamento.

Ao longo dos anos, o Arduino tem sido o cérebro de milhares de projetos.

Arduino nasceu no “Ivrea Interaction Design Institute” como uma ferramenta fácil para prototipagem rápida, destinada a estudantes sem formação em eletrônica e programação. Assim que atingiu uma comunidade mais ampla, a placa Arduino começou a mudar para se adaptar às novas necessidades e desafios, diferenciando sua oferta de placas de 8 bits simples para produtos para aplicativos IoT , wearable, impressão 3D e ambientes incorporados. Todas as placas do Arduino são completamente de código aberto, capacitando os usuários a construí-las independentemente e, eventualmente, adaptá-las às suas necessidades específicas. O software também é de código aberto e está crescendo através das contribuições dos usuários em todo o mundo (ARDUINO, 2019).

## 3 DESENVOLVIMENTO DA APLICAÇÃO

### 3.1 ESTRUTURA WEB

#### 3.1.1 Aplicação Web

A aplicação web foi toda estruturada em cima do Framework Laravel, focando em segurança, praticidade e rapidez no desenvolvimento e hospedagem. Na Figura 3, tem-se a ilustração de um sistema com uma interface de registro e *login* para o usuário.

**Figura 3. Tela de registro de usuário na plataforma web**

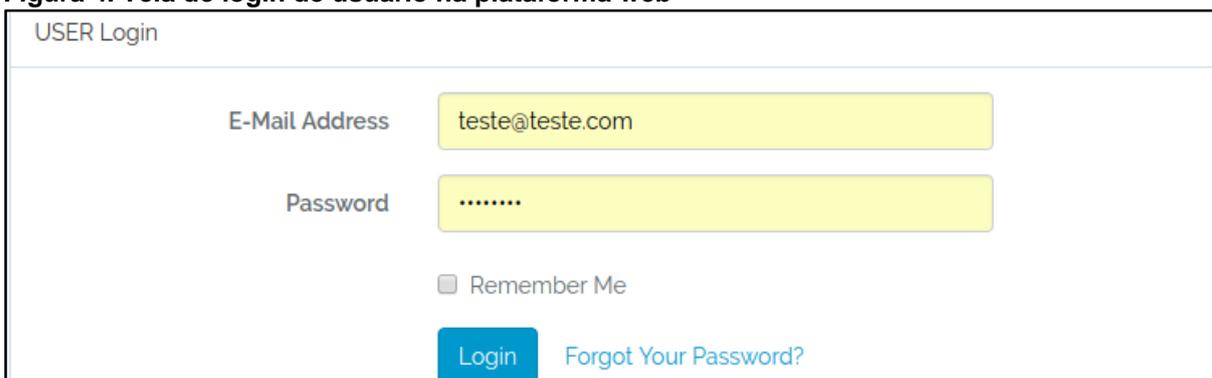


The screenshot shows a registration form titled "Register". It contains four input fields: "Name" with the value "Teste", "E-Mail Address" with the value "teste@teste.com", "Password" with masked characters ".....", and "Confirm Password" with masked characters ".....". A blue "Register" button is located at the bottom of the form.

Fonte: Autoria própria.

Ao se cadastrar, as informações são registradas em uma base MySQL, e a partir desse momento o usuário pode fazer login na plataforma (Figura 4), onde é levado até o dashboard da aplicação.

**Figura 4. Tela de login de usuário na plataforma web**

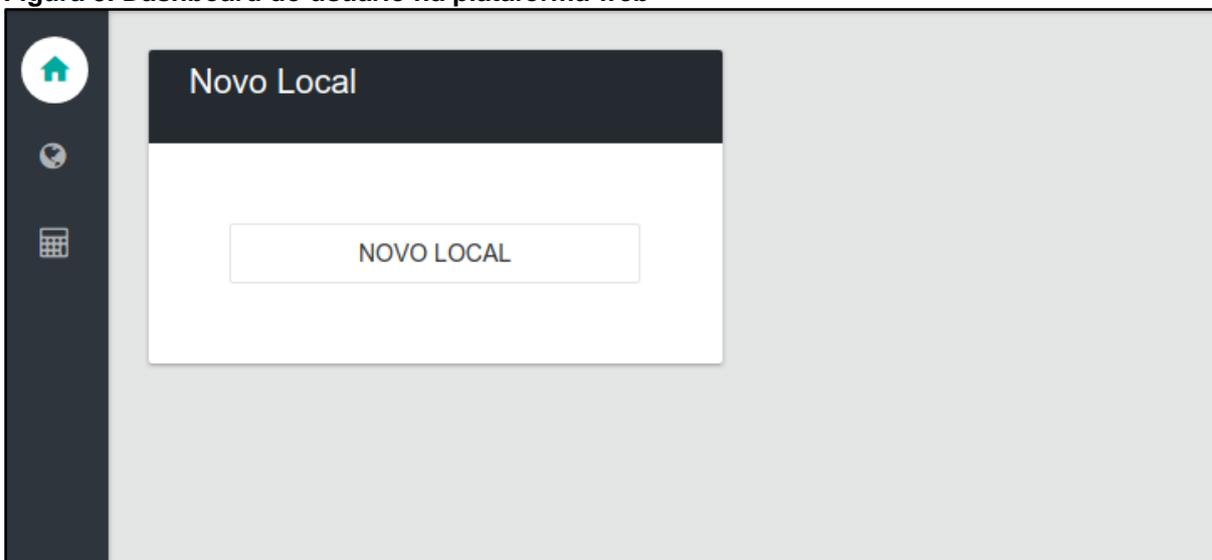


The screenshot shows a login form titled "USER Login". It contains two input fields: "E-Mail Address" with the value "teste@teste.com" and "Password" with masked characters ".....". Below the password field is a checkbox labeled "Remember Me" which is unchecked. At the bottom, there is a blue "Login" button and a link labeled "Forgot Your Password?".

Fonte: Autoria própria.

Na Figura 5 observa-se que no dashboard o usuário tem a opção de criar um local, onde serão cadastradas as placas via integração por API.

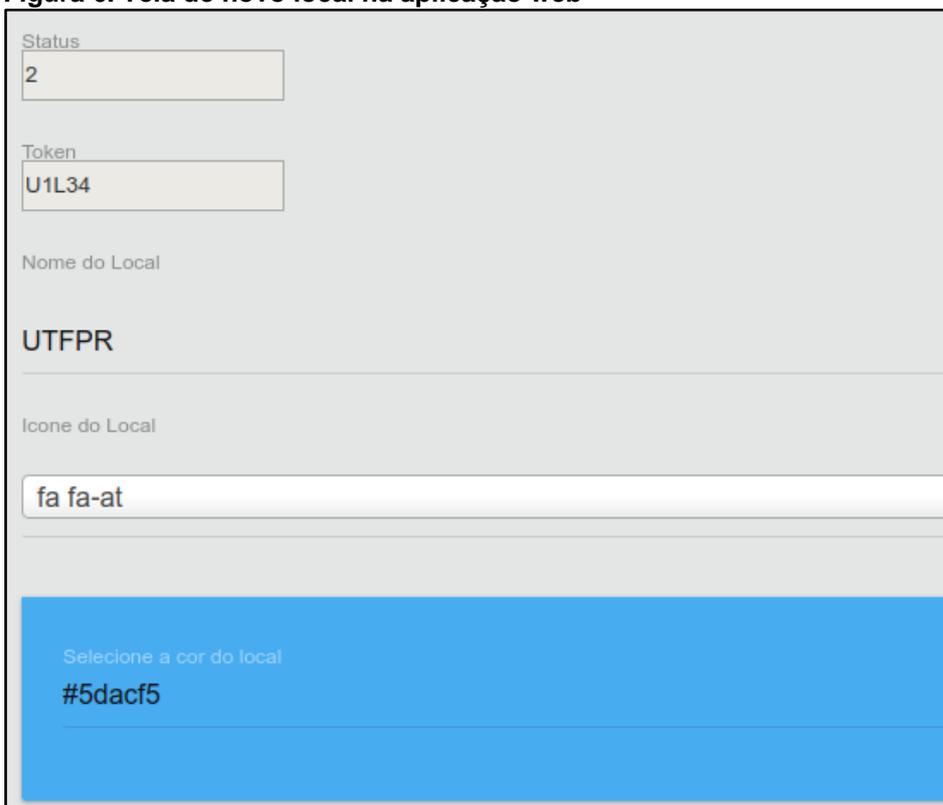
**Figura 5. Dashboard do usuário na plataforma web**



Fonte: Autoria própria.

Quando a placa é cadastrada, um token é gerado (Figura 6), e o mesmo será usado para autenticação na placa arduino.

**Figura 6. Tela de novo local na aplicação web**

A screenshot of a web form titled 'Novo Local'. The form has several input fields: 'Status' with the value '2', 'Token' with the value 'U1L34', 'Nome do Local' with the value 'UTFPR', and 'Icône do Local' with the value 'fa fa-at'. At the bottom, there is a section for selecting a color, with the text 'Selecione a cor do local' and the hex code '#5dacf5' displayed above a blue color swatch.

Fonte: Autoria própria.

Quando a placa com o código arduino é ligada a energia um web-server é criado, nesse caso o nome é remot-9847, conforme ilustram as Figuras 7, 8 e 9.

**Figura 7. Wifi da placa sendo listada nas redes disponíveis**



Fonte: Autoria própria.

**Figura 8. Tela inicial de configuração da placa**



Fonte: Autoria própria.

**Figura 9. Tela com configurações de acesso preenchidas**

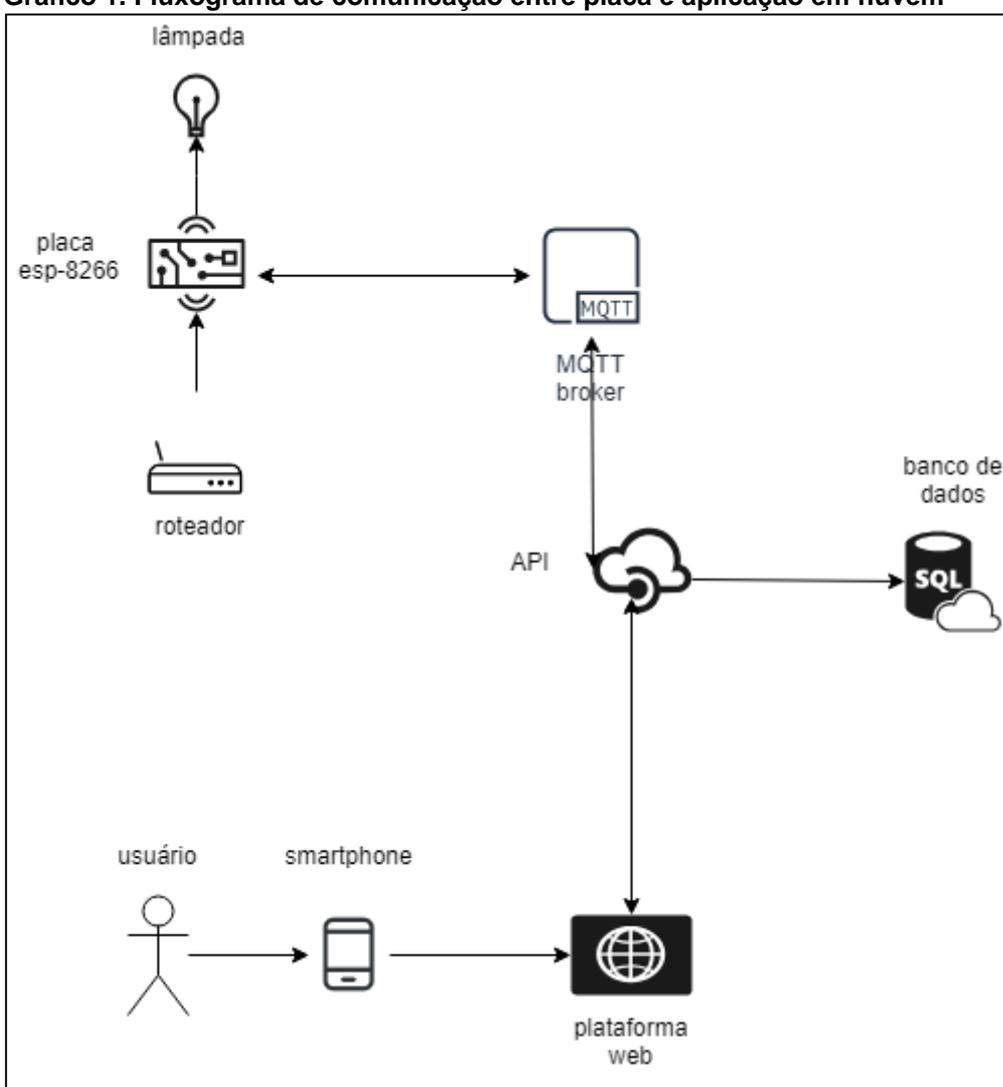


Fonte: Autoria própria.

### 3.1.2 Arduino

Ao acessar a rede wifi criada, o usuário escolhe a rede wifi de sua casa, e logo em seguida insere sua senha e também seu código gerado na plataforma, nesse momento a placa se conecta a internet usando as credenciais do wifi informado, e faz uma requisição à API da aplicação na nuvem passando o token gerado anteriormente na plataforma, então a API retorna para a placa as credenciais de conexão do broker MQTT, sistema de filas responsável pela troca de mensagens entre placa e plataforma, esse fluxo é exemplificado no Gráfico 1.

**Gráfico 1. Fluxograma de comunicação entre placa e aplicação em nuvem**



Fonte: Autoria própria.

Após ter a conexão estabelecida um novo dispositivo aparecerá no local criado pelo usuário e ele poderá ligar e desligar a lâmpada pela plataforma (Figura 10).

Figura 10. Botão de controle de dispositivo disponibilizado após cadastro de placa

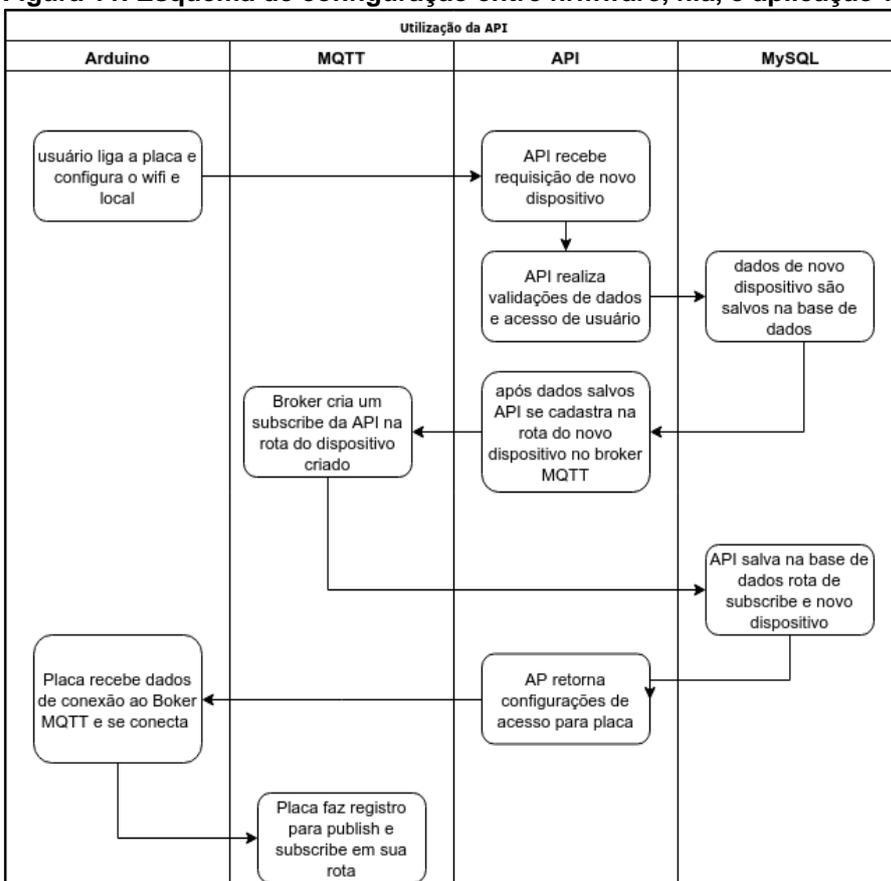


Fonte: Autoria própria.

### 3.1.3 API

Dentro da aplicação web, utilizando a estrutura Laravel foi estruturado um sistema de API (*Application Programming Interface*) para comunicação com a placa programada em Arduino, a API consiste em um endpoint que usa HTTP para realizar a configuração de um novo dispositivo na plataforma, e em caso de sucesso na transação a placa recebe os dados de conexão ao Broker MQTT, em seguida a placa realiza a conexão no broker e se inscreve como publisher e subscriber em sua rota, para poder enviar e receber comandos, como é exibido na Figura 11.

Figura 11. Esquema de configuração entre firmware, fila, e aplicação web



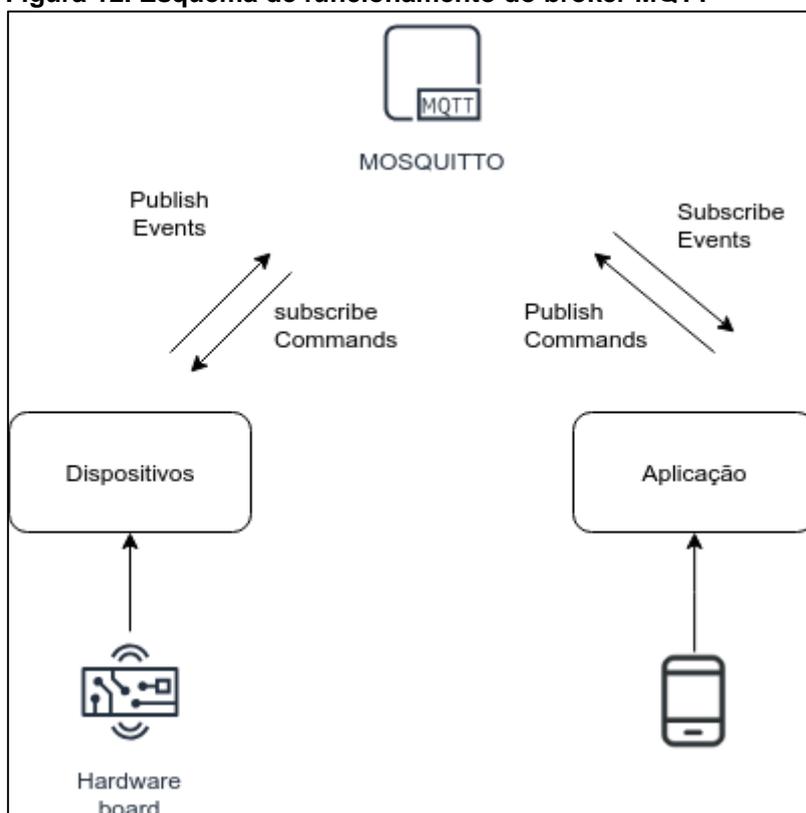
Fonte: Autoria própria.

## 3.2 SISTEMA DE FILAS

### 3.2.1 Mosquitto

Na aplicação foi usado o MOSQUITTO um broker de código aberto que faz o gerenciamento de mensagens usando o protocolo MQTT, ele tem um papel fundamental na aplicação, controlando todo o fluxo de envio e recebimento de mensagens de seus *publishers* à seus *subscribers*, conforme observa-se na Figura 12.

Figura 12. Esquema de funcionamento do broker MQTT



Fonte: Autoria própria.

## 3.3 INFRAESTRUTURA

Para hospedar toda a aplicação usamos a plataforma DigitalOcean, Inc. que é um provedor de infraestrutura em nuvem, que separa as máquinas criadas em droplets. Para essa estrutura foram criados dois droplets, uma para a aplicação web, API, e MySQL, e outro para o broker MQTT (Figura 13).

Figura 13. Droplets no painel Digital Ocean

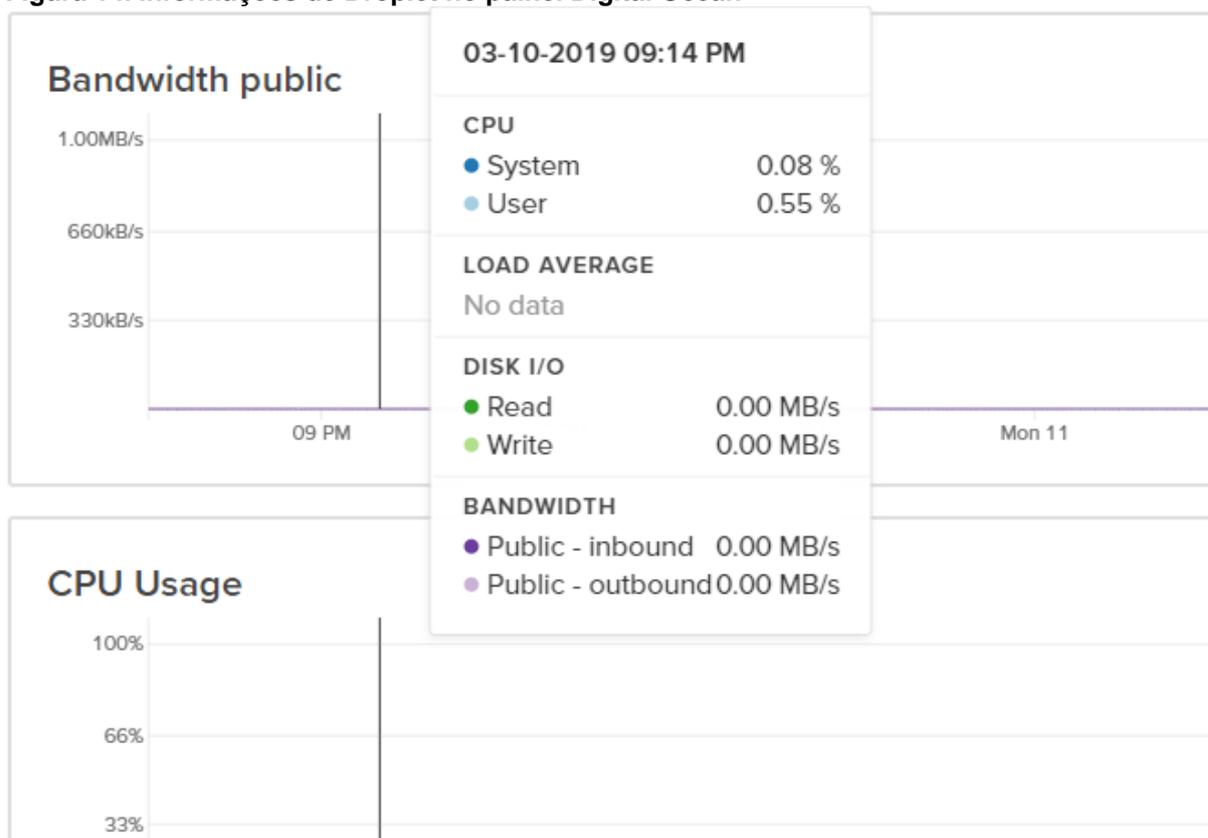
## Droplets

Name	IP Address	Created ▲
 <b>mqtt-remot</b> 512 MB / 20 GB Disk / NYC3 - Ubuntu 16.04.3 x64	104.131.94.23	1 year ago
 <b>remot</b> 512 MB / 20 GB Disk / NYC3 - Ubuntu 16.04.2 x64	104.131.104.235	2 years ago

Fonte: Autoria própria.

Ambos são máquinas com distribuição Linux, onde são instaladas as versões de linguagem e bibliotecas necessárias e cada droplet tem um ip próprio. Para cada droplet é possível saber informações específicas sobre banda, CPU e disco da máquina (Figura 14).

Figura 14. Informações de Droplet no painel Digital Ocean



Fonte: Autoria própria.

## 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Para testar essa estrutura, o sistema arquitetado foi deixado em teste por meses, utilizando placas instaladas em interruptores, causando um tráfego relativamente baixo o que dava uma média de custos na Digital Ocean de 10 USD, para forçar a estrutura a trabalhar mais, um script foi desenvolvido para que simulasse lâmpadas acendendo e apagando em forma de mensagens no *broker* MQTT, a princípio a média de interruptores ligados e desligados foram multiplicados por 20, simulando o que seriam 20 casas na mesma média de uso, com isso o valor final da conta no mês subiu para 16.70 USD, e o mesmo teste simulando a multiplicação por 50 resultou em um aumento para 25 USD como é mostrado na Figura 15.

**Figura 15. Histórico de pagamento de droplets na Digital Ocean**  
Billing history

Date	Description	Amount	
10 days ago	Invoice for February 2019	\$25.00	<a href="#">View Details</a>
1 month ago	Invoice for January 2019	\$25.00	<a href="#">View Details</a>
2 months ago	Invoice for December 2018	\$16.70	<a href="#">View Details</a>
3 months ago	Invoice for November 2018	\$10.00	<a href="#">View Details</a>
4 months ago	Invoice for October 2018	\$10.00	<a href="#">View Details</a>
5 months ago	Invoice for September 2018	\$10.00	<a href="#">View Details</a>
6 months ago	Invoice for August 2018	\$10.01	<a href="#">View Details</a>
7 months ago	Invoice for July 2018	\$10.00	<a href="#">View Details</a>
8 months ago	Invoice for June 2018	\$10.00	<a href="#">View Details</a>
9 months ago	Invoice for May 2018	\$10.03	<a href="#">View Details</a>
10 months ago	Invoice for April 2018	\$10.00	<a href="#">View Details</a>
1 year ago	Invoice for March 2018	\$10.00	<a href="#">View Details</a>
1 year ago	Invoice for February 2018	\$10.00	<a href="#">View Details</a>

Fonte: Autoria própria.

Com esses valores obtidos a parte econômica da aplicação foi muito satisfatória, pois levando em consideração que toda a estrutura está hospedada em um servidor dedicado, com o total controle da aplicação e dos *droplets*, e uma taxa de disponibilidade de 99,99%, é um custo que vale muito a pena.

Em relação à estrutura usada e ao comportamento das aplicações, o resultado foi mais positivo do que o esperado, a placa conseguiu se comunicar de forma

impecável com a API *web*, e vice versa, não ocorreram problemas de falha de comunicação ou erros durante o período de testes.

Com esses resultados obtidos, o estudo concluiu que estruturas *web* em *cloud* e IoT são uma ótima combinação, pois além da segurança, e disponibilidade da aplicação hospedada em *cloud*, os custos são baixos para se iniciar um projeto, e conforme crescem podem ser escalados facilmente e com controle de gastos.

## 5 CONCLUSÃO

O objetivo deste projeto, era validar as possibilidades de uma integração de uma aplicação IoT usando tecnologias web, usando conceitos de fila, API, usando tecnologia WiFi para conexão de placas e estrutura web para manipulação e armazenamento de dados. E toda essa integração mostrou um grande potencial, provando que a internet das coisas pode ser adaptada a tecnologias já usadas no mercado, e mesmo assim serem muito performáticas, graças às facilidades que a computação em nuvem possibilita.

O principal desafio foi integrar o hardware com a aplicação web fazendo a API trabalhar de forma autônoma na configuração de novos dispositivos, mas após isso pronto, houve uma grande facilidade em manipular mensagens entre placa e aplicação, o que abre uma infinidade de possibilidades, como integrar tudo isso à nuvem, que possibilita segurança e alta disponibilidade, tudo isso passando por uma camada de autenticação da plataforma web.

Por fim, esse projeto conclui que a tecnologia em nuvem vem para facilitar ainda mais a integração da internet das coisas em novas aplicações, pois surpreende desde o custo, até a qualidade do serviço entregue.

## REFERÊNCIAS

ARDUINO. **What is Arduino?** Copyright© Arduino, 2018. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 24 nov. 2018.

BUYAYA, Rajkumar; et al. **Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility.** Computer Physics Communications, Geneva, dez. 2009. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167739X08001957>>. Acesso em: 15 nov. 2018.

KHORSHED, Tanzim; ALI, Shawkat; WASIMI, Saleh A. **A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing.** Future Generation Computer Systems. v. 28, n. 6, jun. 2012. p 833-851. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167739X12000180?via%3Dihub>>. Acesso em: 24 nov. 2018.

LARAVEL. **Laravel - the php framework for web artisans.** Copyright© Taylor Otwell, 2018. Disponível em: <<https://laravel.com/>>. Acesso em: 24 nov. 2018.

MQTT. **MQTT Version 3.1.** Editado por Andrew Banks and Rahul Gupta. OASIS Standard. Copyright© OASIS, publicado em: 29 out. 2014. Disponível em: <<http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>>. Acesso em: 24 nov. 2018.

PEREIRA, Adan Lucio; APPEL, Ana .Paula. **Modeling and storing complex network with graph-tree.** In: NEW TRENDS IN DATABASES AND INFORMATION SYSTEMS, Workshop Proceedings of the 16th East European Conference. ADBIS 2012, Pozna, Poland, set. 17-21, 2013, p. 305-315.

PHP. **Manual oficial do PHP: Prefácio.** Copyright © 2001-2018 The PHP Group, 2018. Disponível em: <[http://php.net/manual/pt\\_BR/preface.php](http://php.net/manual/pt_BR/preface.php)>. Acesso em: 24 nov. de 2018.

SANTOS, Jonis Nogueira; SILVA, José Adelar Souza da. **SGBD MySQL.** Faculdades de Informática de Taquara, Faculdades de Taquara (FACCAT), 2018. Disponível em: <[https://fit.faccat.br/~jonis/Artigo\\_mySQL.pdf](https://fit.faccat.br/~jonis/Artigo_mySQL.pdf)>. Acesso em: 24 nov. 2018.

SILVA, Fabrício Rodrigues Henriques da. **Um estudo sobre os benefícios e os riscos de segurança na utilização de Cloud Computing**. Artigo de conclusão de Curso, Centro Universitário Augusto Motta (UNISUAM), Rio de Janeiro, mar. 2011. 15 p. Disponível em: <[https://fabriciorhs.files.wordpress.com/2011/03/cloud\\_computing.pdf](https://fabriciorhs.files.wordpress.com/2011/03/cloud_computing.pdf)>. Acesso em: 28 out. 2018.

SOUSA, Flávio R. C.; MOREIRA, Leonardo O.; MACHADO, Javam C. **Computação em nuvem: Conceitos, tecnologias, aplicações e desafios**. Anais da II Escola Regional de Computação Ceará, Maranhão e Piauí (ERCEMAPI). 2009. cap. 7, p. 150-175.

TAURION, Cezar. **Cloud Computing - Computação em Nuvem: Transformando o mundo da tecnologia da informação**. Rio de Janeiro: Brasport, 2009.

VAQUERO, Luiz M.; et al. **A break in the clouds: Towards a cloud definition**. ACM SIGCOMM Computer Communication Review, v. 39, n. 1, jan. 2009. Disponível em: <<http://ccr.sigcomm.org/online/files/p50-v39n1l-vaqueroA.pdf>>. Acesso em: 24 out. 2018.

VERMA, Archit. **MVC architecture: A comparative study between ruby on rails and laravel**. Computer Science and Engineering, Manipal Institute of Technology, Manipal, India, v. 5, n. 5, out-nov 2014. Disponível em: <<http://www.ijcse.com/docs/INDJCSE14-05-05-053.pdf>>. Acesso em: 10 nov. 2018.