



DEVELOPMENT OF AN AUTOMATION SYSTEM, INTEGRATING A VIRTUAL ASSISTANT AND IoT DEVICES

Priscila Reis Soares da Paixão¹, Mauro Henrique da Silva Wencelewski², Manoel Henrique Reis Nascimento³

^{1,2,3} Fundação Centro de Análise Pesquisa e Inovação Tecnológica (FUCAPI) - Av. Gov. Danilo de Matos Areosa, 381 - Distrito Industrial, Manaus - AM.

Email: priscila.wencelewski@gmail.com, mauro.wencelewski@gmail.com, hreys@bol.com

Received: June 03th, 2019

Accepted: June 06th, 2019

Published: September 30th, 2019

Copyright ©2016 by authors and Institute of Technology Galileo of Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<https://creativecommons.org/licenses/by/4.0/>



Open Access

ABSTRACT

The Internet of Things (IoT) is rapidly evolving along with network peripherals that contain embedded communication technology. In this way, it is increasingly common to find in homes devices connected to each other or to external systems. The next step in this evolution is the devices that allow interaction by voice commands, virtual assistants are already a reality in almost all the smartphones of the big manufacturers in the market. But in the residential market is still growing, even if there are several models of large manufacturers, it is still possible to develop customized solutions, using low cost devices, libraries and APIs free to build a custom version customized to their needs. The purpose of this work is to demonstrate the development of a low-cost solution using a Raspberry Pi to develop the residential assistant and the ESP8266 controller to control peripherals, being activated by voice command.

Keywords: Virtual Assistant, Internet of Things, Raspberry.

DESENVOLVIMENTO DE UM SISTEMA DE AUTOMAÇÃO, INTEGRANDO UMA ASSISTENTE VIRTUAL E DISPOSITIVOS IoT.

RESUMO

A Internet das Coisas (IoT) está evoluindo rapidamente, como também, junto com a rede periféricos que contém tecnologia de comunicação embarcada. Dessa forma, esta cada vez mais comum encontrarmos em residências dispositivos conectados entre si ou com sistemas externos. O próximo passo dessa evolução são os dispositivos que permitem interação por comandos de voz, as assistentes virtuais já são uma realidade em quase todos os smartphones dos grandes fabricantes do mercado. Mas no mercado residencial ainda está em crescimento, mesmo existindo diversos modelos de grandes fabricantes, ainda é possível o desenvolvimento de soluções customizadas, utilizando dispositivos de baixo custo, bibliotecas e APIs gratuitas para construir uma versão personalizada e customizada as suas necessidades. A proposta desse trabalho, é demonstrar o desenvolvimento de uma solução de baixo custo, utilizando-se de uma Raspberry Pi para desenvolver a assistente residencial e um controlador ESP8266 para controle dos periféricos, sendo acionado por comando de voz.

Keywords: Assistente Virtual, Internet das Coisas, Raspberry.

I. INTRODUÇÃO

O uso de dispositivos inteligentes no dia-a-dia aumenta qualidade de vida e melhora a produtividade. Isso levou a um aumento da busca por benefícios como conforto, controle centralizado de equipamentos, redução de custo, consumo de energia e segurança que são basicamente os propulsores do crescimento das tecnologias de automação em residências e indústrias [1].

Até alguns anos atrás, os dispositivos eram limitados a interação com usuário apenas via toque, texto ou controle remoto. Mas desde o lançamento de softwares como *Google Home* e *Amazon Echo*, a tecnologia de reconhecimento de voz rapidamente está mudando a forma como interagimos com dispositivos [2].

A Internet das Coisas (IoT) está evoluindo rapidamente junto com rede periféricos que contém tecnologia de comunicação embarcada. Analistas preveem que o mercado mundial de IoT irá atingir US\$ 1.7 trilhões até 2020 com crescimento de mercado de 16.9%. [3]

A interconectividade de praticamente todos os equipamentos é possível através da internet, mídias sociais e comunicação máquina a máquina [1].

Ainda de acordo com [3], é previsto que dos 25% das residências utilizando assistentes virtuais terão um ou mais dispositivos conectados.

Isso significa uma evolução natural dos dispositivos eletrônicos domésticos que eram limitados a serem controlados por toque, texto ou controle remoto serão controlados por voz.

Dessa forma, existe um grande mercado de automação residencial envolvendo dispositivos IoT e assistentes virtuais a ser explorado. De acordo com [3] no mundo IoT existe um grande encorajamento a cultura DIY (*do it yourself*) o que significa que os usuários podem customizar dispositivos e aplicações de acordo com as suas necessidades.

A proposta desse trabalho, é demonstrar o desenvolvimento de uma solução de baixo custo, utilizando-se de uma Raspberry Pi para desenvolver a assistente residencial e um controlador ESP8266 para controle dos periféricos, sendo acionado por comando de voz.

II. REFERENCIAL TEÓRICO

II.1 CHATBOTS

De acordo com [4], chatbots ou robôs de conversação podem ser definidos como softwares (ou até, algumas vezes, hardwares) que são capazes de responder a mensagens de texto ou voz. Esses programas tentam simular de um ser humano, portanto, agindo como uma interface de linguagem natural para provedores de dados e serviços. Eles podem ser programados para responderem de formas diferentes, dependendo da existência de certas palavras chaves, ou eles podem ser mais sofisticados e aplicarem técnicas de *machine learning* para basear suas respostas, baseando-se em palavras incluídas nas solicitações.

Chatbots com inteligência artificial podem ser utilizados em diferentes cenários, como sistemas bancários, atendimento a clientes e Educação [5].

Os chatbots surgiram como uma nova forma de interagir com serviços web e *smart devices*, fornecendo interfaces de diálogo que são ostensivamente mais intuitivas e naturais que as tradicionais [6].

II.2 ASSISTENTES VIRTUAIS

Segundo [7], a invenção das assistentes virtuais foi um importante evento na história da computação. Elas são úteis para ajudar usuários de sistemas computacionais a automatizar tarefas e cumpri-las com o mínimo de esforço e interferência humana.

Assistentes virtuais podem ser úteis para uma série de tarefas, como tomar notas, previsões do tempo, tocar música, realizar pesquisas na internet e mais recentemente, integrações com outros periféricos nas residências.

De acordo com [8], as assistentes de virtuais controladas por voz estão substituindo a interação tradicional entre humanos e computadores, redefinindo a forma como acessamos a internet, dados e aplicativos. Facilitando a vida dos usuários, vendo que os usuários não necessitam digitar nenhum comando ou ter nenhum tipo de interação especial com os dispositivos, percebendo o conceito de computação ubíqua, e assim criando uma experiência mais próxima da interação com um ser humano.

O funcionamento de uma assistente virtual pode ser descrito com auxílio do diagrama abaixo (Figura 1).

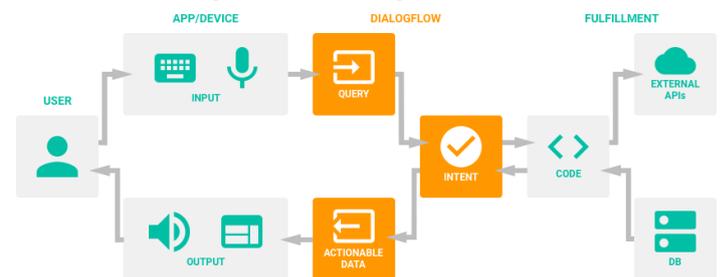


Figura 1 - Arquitetura Assistente Virtual.

Fonte: [10].

O usuário faz uma requisição (comando) para um dispositivo ou aplicativo, essa requisição é interpretada por uma *Application Programming Interface* (API) de *Natural Language Processing* (NLP) podendo ou não chamar serviços externos, em seguida, retornar os dados para serem expostos ao usuário de forma de áudio ou informações numa interface.

II.3 DIALOGFLOW

De acordo com [9], Dialogflow é um conjunto de ferramentas de desenvolvimento para criar interfaces conversacionais para sites, aplicativos móveis, plataformas populares de mensagens e dispositivos IoT, com capacidade de desenvolvimento de ponta a ponta e implantação simplificada.

Essa API utiliza algoritmos de *Machine Learning* para processar os possíveis caminhos de conversação e roda totalmente na infraestrutura cloud do Google.

A estrutura de diálogo pode ser dividida da seguinte forma:

- **Agentes**

São os módulos de Entendimento de Linguagem Natural, que podem ser incluídos em aplicações, websites, produtos ou serviços e traduzir texto ou comandos de voz e ações reais.

- **Intenções**

São uma funcionalidade que permite que você crie intenções no seu agente para mapear entradas (comandos) a respostas. São definidos exemplos de comandos do usuário que podem ativar cada uma das intenções, que informações extrair e como responder ao usuário.

Os quatro principais componentes que permite mapearmos do que usuário disse ao que o agente responde:

- **Nome da Intenção**

É o nome utilizado para o fulfillment e identifica a respectiva intenção disponibilizada pelo usuário.

- **Frases de Treinamento**

São os exemplos de frases que o usuário pode falar para acionar uma intenção específica. DialogFlow automaticamente expande a quantidade expressões possíveis.

- **Ações e Parâmetros**

Define o quão importante são as informações (parâmetros) extraídos de iterações do usuário. Podem ser datas, horários, nomes, lugares e etc. Essas informações podem ser usadas como entrada em outras lógicas.

- **Resposta**

A expressão que é falada ou exibida de volta ao usuário.

- **Entidades**

São um recurso utilizado para extrair dados importantes através das entradas de linguagem natural. Existem entidades padrões e entidades que podem ser configuradas durante o desenvolvimento.

- **Fulfillment**

É o código que é implantado como um webhook que permite que seu agente Dialogflow chame a lógica de negócios com base em intenção por intenção. Durante uma conversa, o preenchimento permite usar as informações extraídas pelo processamento de linguagem natural do Dialogflow para gerar respostas dinâmicas ou acionar ações no seu back-end.

II.4 SMART HOME

O conceito de *Smart Home* (Casa Inteligente) pode ser definido, de acordo com [10], como uma combinação de vários subsistemas de tecnologias avançadas relacionados com vida residencial, ou seja, utilizando tecnologias computacionais, de exibição de imagem e comunicações para conectar vários ambientes para obter um sistema com controle e gerenciamento conveniente.

O maior objetivo é fornecer aos usuários um ambiente eficiente, confortável, seguro, conveniente e amigável. Integrando dessa forma, sistemas, serviços e gerenciamento.

Ainda de acordo com [10], a implantação tradicional de uma *Smart Home* geralmente envolve o controle e comunicação dos ambientes através do cabeamento físico, o que ocasiona uma dificuldade em relação ao cabeamento, ao custo alto de instalação e a escalabilidade do sistema é baixa. Mas os sistemas de *Smart Home* baseados em tecnologias de sensores sem fio, pode além de, eliminar o problema do cabeamento, reduzir custo de instalação, aumentar a escalabilidade do sistema.

II.5 RASPBERRY PI

Raspberry Pi (Figura 2) é um computador do tamanho de um cartão de crédito desenvolvida e fabricada no Reino Unido com a intenção inicial de proporcionar um dispositivo computacional barato para educação [11].



Figura 2 - Raspberry Pi 3 A+
Fonte: [11].

A história desse minicomputador inicia-se nos laboratórios de computação da universidade da Cambridge em 2006. O nascimento da Raspberry Pi Foundation se deu com a união de quatro cientistas da computação que estavam preocupados com os estudantes de ciência da computação deficientes em aspectos técnicos da computação.

Durante os próximos seis anos, esses cientistas trabalharam para desenvolver um hardware barato e acessível que ia ajudar escolas a ensinarem conceitos de computação e programação.

Desde 2012, quando a primeira Raspberry Pi foi lançada, as placas sofreram uma grande quantidade de revisões e geralmente foram disponibilizadas em duas vertentes de modelos, A e B.

II.6 ESP8266

É um microcontrolador fabricado pela empresa chinesa Espressif, de baixo custo e que inclui como uma das suas principais características a capacidade de conexão WiFi.

O Espressif ESP8266 entrega uma solução de um System-On-Chip (SoC) com WiFi para atender a demanda de usuários por conectividade, eficiência energética, compacto e alta confiabilidade para o mercado de Internet das Coisas [12].

As especificações técnicas são as seguintes:

- Suporte a rede 802.11 b/g/n
- Suporte a rede 802.11n (2.4Ghz) até 72Mbps
- Desfragmentação
- 2 x Interface Wi-Fi virtual

Devido ao fato de Espressif fabricar somente o microcontrolador, para melhor realizar uma interface com o ambiente, foi escolhida uma plataforma de desenvolvimento baseada no mesmo, chamada NodeMCU, como pode ser vista na Figura 3.



Figura 3 – NodeMCU.
Fonte: [14].

De acordo com NodeMCU, a plataforma é composta por um firmware e kit de desenvolvimento que permite ao usuário desenvolver protótipos rápidos de produtos IoT com algumas linhas de scripts LUA. No nosso caso não iremos utilizar o Firmware e bibliotecas do NodeMCU, somente a plataforma de desenvolvimento (Hardware).

II.7 MQTT

MQTT significa Message Queuing Telemetry Transport. É um protocolo de mensagens extremamente simples e leve de publicação / assinatura, projetado para dispositivos restritos e redes de baixa largura de banda, alta latência ou não confiáveis. Os

princípios de design são minimizar a largura de banda da rede e os requisitos de recursos do dispositivo, ao mesmo tempo em que tentam garantir a confiabilidade e um certo grau de garantia de entrega. Esses princípios também tornam o protocolo ideal para o emergente mundo de dispositivos conectados “máquina a máquina” (M2M) ou “Internet das Coisas” de dispositivos conectados, e para aplicações móveis onde a largura de banda e a bateria são preciosas [13].

A internet das coisas possibilita a habilidade de conectar um grande número de dispositivos através da internet. Esses dispositivos possuem identificadores únicos, o IoT (Internet das Coisas) cria um ambiente inteligente por conectar esses dispositivos e equipa-los com a habilidade de recolher e trocar dados [14].

Como parte da arquitetura do protocolo, toda implementação é dividida em *Broker* e *Client*, onde o broker é o servidor responsável por armazenar e orquestrar o envio de mensagens aos *clients*.

Mosquito Broker

De acordo com [15], Mosquitto é uma implementação de servidor e cliente compatível com padrões do protocolo de mensagem MQTT. Que é um modelo de publicação/assinatura, tem baixa sobrecarga de rede e pode ser implementado em dispositivos de baixa potência, tais microcontroladores que podem ser usados em sensores remotos da Internet das Coisas.

III. SOLUÇÃO PROPOSTA

A arquitetura do sistema proposto pode ser dividida em duas partes: A Assistente Virtual e Dispositivos IoT, conforme pode ser observado na Figura 4.



Figura 4 - Arquitetura do Sistema.

Fonte: Autores, (2019).

Todo o processamento do sistema é realizado na Raspberry Pi, com exceção das API de conversão Voz para Texto (*speech to text*) do Google e a API do *Dialogflow* utilizada para criar a rotina de conversação. A outra etapa, é a etapa de automação residencial feita por microcontrolares ESP8266 conectados a uma rede MQTT.

III. 1 ASSISTENTE VIRTUAL

O módulo da Assistente Virtual, é composto por uma integração de diferentes tecnologias, utilizando-se de APIs de NLP, assim como bibliotecas de conversão de áudio para texto, como pode ser observado na Figura 5.

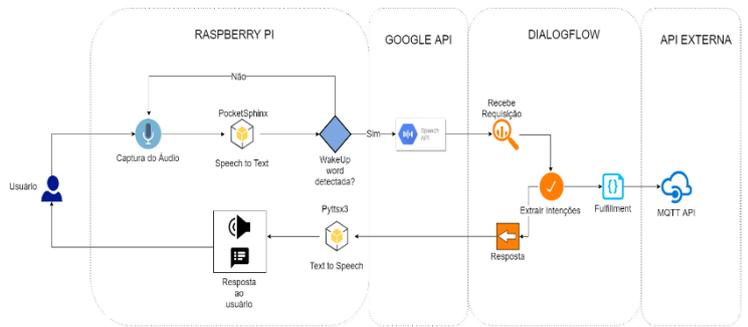


Figura 5 – Fluxograma.

Fonte: Autores, (2019).

Durante um intervalo de tempo o áudio é capturado através do microfone, utilizando a biblioteca *PocketSphinx* é possível converter esse áudio para texto localmente, para checar se a *wakeup word* foi dita. Essa biblioteca foi criada para rodar em dispositivos com hardware e processamento limitado, utilizando de técnicas de aprendizado de máquina para reconhecer os padrões acústicos das palavras.

Se a palavra não é identificada, o algoritmo permanece em um loop infinito até a palavra ser encontrada, quando a palavra é identificada, no entanto, é dada uma resposta ao usuário utilizando a biblioteca *Pyttsx3* que permite converter texto em uma voz sintetizada para o usuário.

Após o usuário ter sido sinalizado que a assistente está acordada, o microfone captura a requisição do usuário, mas ao invés de ser processada pela *PocketSphinx* o áudio passa pelo processo de conversão necessário, e é enviado a API do *Google Cloud de Speech to Text* (STT), retornando assim o áudio em formato de texto.

O resultado é então enviado a API de NLP *DialogFlow* onde as intenções da requisição serão capturadas e interpretadas. Dependendo do resultado da interpretação, se positivo, utiliza-se a funcionalidade de *Webhook* da API do *DialogFlow* para fazer uma requisição a API MQTT que foi desenvolvida para fazer uma integração com *Client MQTT Paho*, dessa forma, publicando uma mensagem no canal utilizado pelos dispositivos IoT da residência.

API MQTT

Serviço desenvolvido utilizando a linguagem de programação Python para fazer uma integração com *Broker MQTT Paho*, dessa forma viabilizando que a API *DialogFlow* utilize a funcionalidade de *Webhook* para assim fazer uma requisição diretamente ao *Broker MQTT*.

Permitindo, dessa forma, que ao entender a intenção de um comando no *DialogFlow*, o mesmo envie o comando (ligar ou desligar) para a rede MQTT, passando o identificador do dispositivo a ser acionado.

III. 2 DISPOSTIVOS IoT

Como a segunda metade da solução proposta, foram desenvolvidos periféricos com funcionalidade IoT utilizando a placa de desenvolvimento *NodeMCU* baseada no *ESP8266*.

Cada uma das placas foi associada a um circuito de acionamento de cargas AC, a ideia inicial é permitir a automação de *Ligar/Desligar* periféricos da residência, para o caso específico, os testes foram feitos automatizando o controle de lâmpadas na residência.

O funcionamento é dado da seguinte forma, todas as placas se “inscrevem” em um canal específico do *Broker MQTT*, dessa forma, quando uma mensagem é publicada com um comando

específico, o mesmo é interpretado pelo microcontrolador, assim, ligando ou desligando o periférico conectado a placa.

III. 3 MONTAGEM

Assistente Virtual

A assistente virtual foi montada utilizando uma Raspberry Pi 3 A+ e um microfone Adafruit I2S MEMS SPH0645LM4H seguindo o diagrama a seguir (Figura 6):

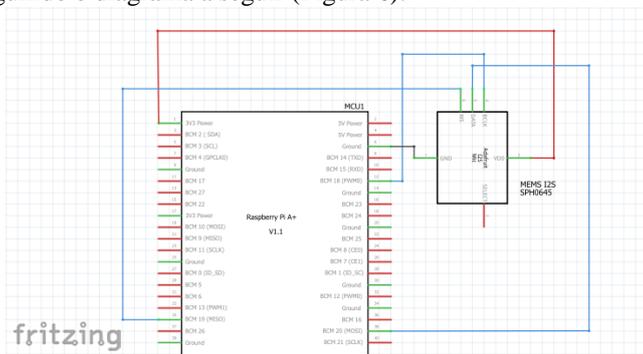


Figura 6 - Circuito Assistente Virtual.

Fonte: Autores, (2019).

A etapa da assistente virtual é composta apenas pela ligação do Microfone MEMS ao pino da Raspberry Pi, nesse caso, não foi feita nenhuma montagem de placa PCI ou em Protoboard.

Dispositivos IoT

Os dispositivos de acionamento foram criados utilizando um controlador ESP8266 acionando um relé 9V para ligar ou desligar os equipamentos, como demonstrado na Figura 7.

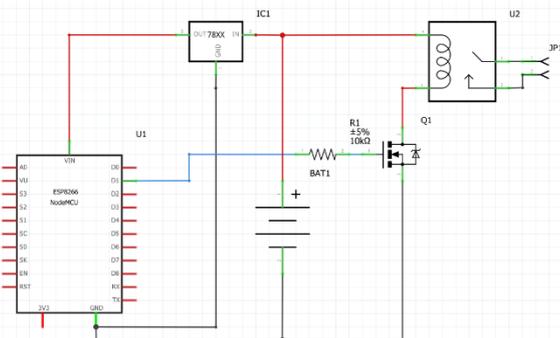


Figura 7 - Circuito de Acionamento.

Fonte: Autores, (2019).

O dispositivo trabalha com um regulador de tensão LM7805 para regular a tensão de 9V para 5V e assim alimentar o microcontrolador. Na etapa de acionamento de carga, foi utilizado o Mosfet IRF540N para acionar um relé 9V, que iria de fato acionar a carga AC. Foram utilizados, componentes discretos e de fácil aquisição no mercado, por se tratar de um protótipo.

IV. CONCLUSÃO

Dispositivos conectados já são uma realidade nos dias de hoje, estudos preveem um aumento considerável nos próximos anos. Dessa forma uma evolução desse crescimento é a necessidade de os usuários controlarem seus dispositivos e interagirem com eles através de comandos de voz. Diversos fabricantes estão atuando nesse mercado, mas também existe a possibilidade de usuárias com conhecimento e interesse necessário, construir soluções customizadas as suas necessidades. Ainda que o projeto proposto apenas receba interprete comandos simples, é possível treinar o

agente a entender frases e engajar em conversas mais complexas com o usuário. Assim, podemos concluir que existe muito espaço para ser explorado nesse ramo de tecnologias e que diversas abordagens a dispositivos conectados e controlados por voz serão exploradas no futuro, sempre havendo espaço para as pessoas com interesse desenvolverem seus próprios sistemas com base em tecnologias de acesso gratuito.

V. REFERENCIAS

[1] Abadulrahman, T. A., Isiwekpeni, O. H., Surajudeen-Bakinde, N. T., Otuoze, A. O. Design, Specification and Implementation of a Distributed Home Automation System. The 2nd International Workshop on Internet of Things: Networking, Applications and Technologies. Montreal. CA. 2016.

[2] Homesales. Home Automation and Virtual Assistants – What are People Using Now? 2017. Disponível em: <<https://homesales.com.au/news/latest/whats-new-home-automation-virtual-assistance>>. Acessado em: 22/03/2019.

[3] H. Chung, et al. Digital Forensic Approaches for Amazon Alexa Ecosystem. Digital Forensics Research Workshop. Austin. TX. 2017.

[4] Veglis, Andreas. Maniou, Theodora A. Chabots on the Rise: A New Narrative in Journalism. Redfame Publishing [online].7. 2019. [Data da consulta, 25 de maio de 2019]. Disponível em: <<http://redfame.com/journal/index.php/smc/article/view/3986>>

[5] Dutta, Debasatwa. Developing an Intelligent Chat-bot Tool to assist high school students for learning general knowledge subjects. School of Computer Science Technical Reports. Atlanta, GA.

[6] McTear, Michael. Conversational Modelling For Chatbots: Current Approaches and Future Directions. School of Computing and Matematics. Ulster. Northern Ireland.

[7] T. Pant. Building Virtual Assistants for Raspberry Pi. India: Apress, 2016.

[8] Alepis, Efthimos. Patsakis, Constantinos. Monkey Says, Monkey Does: Security and Privacy on Voice Assistants. IEEE Access [online]. 5. 2017. [Data da consulta, 31 de maio de 2019]. Disponível em: <<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=8023746>>

[9] Google. DialogFlow Enterprise Edition. 2018. Disponível em: <<https://cloud.google.com/dialogflow-enterprise>>. Acessado em: 27/03/2019.

[10] Li, Min. Gu, Wenbin. Wei, Chen. He, Yeshen, Wu, Yannian. Zhang, Yinying. Smart Home: Architecture, Technologies and Systems. 8th International Congress and Communication Technology. Xiamen, CH. 2018.

[11] Raspberry Pi Foundation. What is a Raspberry Pi? 2018. Disponível em: <<https://www.raspberrypi.org/help/what-is-a-raspberry-pi>>. Acessado em: 26/03/2019

[12] Espressif. ESP8266EX Datasheet. 2018. Disponível em: <https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>. Acessado em: 06/04/2019.

[13] MQTT. What is MQTT? 1999. Disponível em: <<http://mqtt.org/faq>>. Acessado em: 06/04/2019.

[14] Kashyap, Monika. Sharma, Vidushi. Gupta, Neeti. Taking MQTT and NodeMCU to IOT: Communication in Internet of Things. 8th International Congress and Communication Technology. Xiamen, CH. 2018.

[15] R. A. Light. Mosquitto: server and cliente implementation of the MQTT protocol. The Journal of Open Source. Vol 2. No 13. 2017. Disponível em: <<https://www.theoj.org/joss-papers/joss.00265/10.21105.joss.00265.pdf>>. Acessado em: 06/04/2019.