



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE GOIÁS  
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE  
COMPUTAÇÃO



HENNER SOARES CAMPOS FILHO  
MATEUS MENDES DE ARAÚJO

COMPARATIVO ENTRE MÉTODOS DE AUTOMAÇÃO RESIDENCIAL  
DE BAIXO CUSTO

GOIÂNIA – GO

2019

---

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR  
VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE  
GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG**

---

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC nº 1204/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

**1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG):**

Nome completo do autor: Henner Soares Campos Filho; Mateus Mendes de Araújo

Título do trabalho: Comparativo entre métodos de automação residencial de baixo custo.


**2. Informações de acesso ao documento:**

Concorda com a liberação total do documento ☒ SIM ☐ NÃO<sup>1</sup>

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF do TCCG.

Mateus Mendes de Araújo / Henner Soares Campos Filho  
(Nome completo do autor)<sup>2</sup>

Ciente e de acordo:

  
(Nome completo do orientador)<sup>2</sup>

Data: 16 / 12 / 19

---

<sup>1</sup> Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

<sup>2</sup> As assinaturas devem ser originais sendo assinadas no próprio documento, imagens coladas não serão aceitas.



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE GOIÁS  
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE  
COMPUTAÇÃO



HENNER SOARES CAMPOS FILHO  
MATEUS MENDES DE ARAÚJO

## COMPARATIVO ENTRE MÉTODOS DE AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO

Trabalho de Conclusão de Curso apresentado à  
Universidade Federal de Goiás como um dos pré-  
requisitos para a obtenção do grau de Bacharel em  
Engenharia Elétrica, sob a orientação do Prof. Dr.  
Geyverson Teixeira de Paula.

GOIÂNIA – GO

2019

Ficha de identificação da obra elaborada pelo autor, através do  
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Campos Filho, Henner Soares

Comparativo entre métodos de automação residencial de baixo  
custo [manuscrito] / Henner Soares Campos Filho, Mateus Mendes  
De Araújo. - 2019.  
xxi, 21 f.

Orientador: Prof. Dr. Geyverson Teixeira de Paula.

Trabalho de Conclusão de Curso (Graduação) - Universidade  
Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de  
Computação (EMC), Engenharia Elétrica, Goiânia, 2019.

Bibliografia.

Inclui fotografias, gráfico, tabelas.

1. Automação residencial. 2. Baixo custo. 3. Blynk. 4. Comparativo.  
5. Custo benefício. I. De Araújo, Mateus Mendes. II. de Paula,  
Geyverson Teixeira, orient. III. Título.

CDU 621.3

## ATA DE AVALIAÇÃO DE PROJETO FINAL

<b>CURSO</b>
--------------

(X) Eng. Elétrica      ( ) Eng. Mecânica      ( ) Eng. de Computação  
( ) Projeto Final 1      ( ) Projeto Final II

<b>AVALIAÇÃO DE PROJETO FINAL</b>
-----------------------------------

Título do projeto: COMPARATIVO ENTRE MÉTODOS DE AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO

<b>BANCA AVALIADORA</b>
-------------------------

Membro 1: Prof. Dr. Geyverson Teixeira de Paula

Membro 2: Prof. Dr. Igor Kopcak

Membro 3: Prof. Dr. Gelson Antônio Andréa Brigatto

ESTUDANTES	
Matrícula	Nome
201407750	Henner Soares Campos Filho
201406787	Mateus Mendes de Araújo

NOTAS													
Matrícula	Membro 1				Membro 2				Membro 3				Média
	NPT	NTE	NAA	NF	NPT	NTE	NAA	NF	NPT	NTE	NAA	NF	
201407750	10,0	9,5	9,5	9,6	10,0	8,5	10,0	9,3	10,0	8,0	9,5	8,9	9,3
201406787	10,0	9,5	9,5	9,6	10,0	8,5	10,0	9,3	10,0	8,0	9,5	8,9	9,3

NPT – Nota plano de trabalho; NTE – Nota do trabalho escrito; NAA – Nota de apresentação e arguição

Para Eng. Elétrica, Mecânica e PFC2 da Eng. Da Computação:  $NF = 0,1 \times NPT + 0,45 \times NTE + 0,45 \times NAA$

Para PFC1 da Eng. Da Computação:  $NF = 0,3 \times NPT + 0,7 \times NAA$

Goiânia, 06 de dezembro de 2019.

[Assinatura]  
Membro 1

[Assinatura]  
Membro 2

[Assinatura]  
Membro 3

# COMPARATIVO ENTRE MÉTODOS DE AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO

Henner Soares Campos Filho, Mateus Mendes de Araújo

**Resumo** — O projeto consiste na implementação de métodos disponíveis para automação residencial de baixo custo sem fio usando relés, rotinas, sensores, microcontroladores e smartphones. O produto da implementação é uma comparação de pontos fortes e fracos dos métodos de acordo com parâmetros pré-estabelecidos. O objetivo é testar 4 tecnologias com 4 aplicativos, explicando o uso e comparando os pontos positivos e negativos de cada um.

**Palavras chave** — automação residencial, baixo custo, blynk, comparativo, custo benefício, mqtt, sem fio, sonoff.

**Abstract** — The project consists of implementing methods available for low cost wireless home automation using relays, routines, sensors, microcontrollers and smartphones. The product of the implementation is a comparison of strengths and weaknesses of the methods according to predefined parameters. The goal is to test 4 technologies with 4 applications, explaining their use and comparing the strengths and weaknesses of each.

**Index Terms** — blynk, comparative, cost benefit, home automation, low cost, mqtt, sonoff, wireless.

## I. INTRODUÇÃO

Desde o início da humanidade, o ser humano busca alterar suas condições de modo que facilite sua vida. A automação é uma das maneiras de se trabalhar esta facilidade, muitas vezes não somente por comodidade, mas para eficiência.

A internet das coisas (IoT - *Internet of Things*) consiste em dispositivos inteligentes que se comunicam entre si. Isto possibilita infinitas aplicações.

Mesmo com toda a tecnologia já possível de ser utilizada, ainda no Brasil é difícil se ver utilização nas residências e pequenos centros comerciais. Isto se deve ao alto preço de equipamentos e serviços. Porém, já existem tecnologias disponíveis para pequena escala, de maneira não invasiva que facilmente cumpriria a solicitação desses consumidores.

Um dos métodos é o Blynk, que consiste numa plataforma com aplicativos para IOS e Android para controlar Arduino, Raspberry PI, Esp8266 entre outros módulos. [1]

O Blynk é um painel de controle digital onde se pode construir uma interface gráfica para o projeto em blocos, selecionados de acordo com a função.

O Blynk não está vinculado a nenhuma placa ou *shield* específico. Ou seja, suporta *hardwares* diferentes. Quando seu Arduino, Raspberry ou chip Esp8266 está conectado à internet através do *wifi*, o Blynk ficará online e pronto para a “internet das coisas” (IoT, *internet of things*).

Outra plataforma que será implementada são os produtos da linha SonOff, que é uma marca que fabrica dispositivos

que podem ser utilizados na automação residencial. São fabricados pela empresa ITEAD, especializada no desenvolvimento e fabricação de dispositivos domésticos inteligentes. É possível controlar tais dispositivos através do aplicativo eWelink, que tem comunicação direta aos produtos da SonOff. [2]

Será implementada ainda a tecnologia de comunicação MQTT (acrônimo para *Message Queuing Telemetry Transport*, ou Transporte de Filas de Mensagem de Telemetria), que é um protocolo criado por Andy Stanford (IBM) -Clark e Arlen Nipper (Eurotech) em 1999, sendo um protocolo de comunicação M2M (*Machine to Machine* ou Máquina a Máquina). Desenvolvido para conectar sistemas de telemetria de oleodutos via satélite. O seu uso foi liberado gratuitamente em 2010, e atualmente sua utilização é muito mais ampla. Tornou-se um padrão OASIS (Organization for the Advancement of Structured Information Standards) em 29/10/2014. [3]

OASIS é um consórcio sem fins lucrativos que impulsiona o desenvolvimento, a convergência e a adoção de padrões abertos para a sociedade da informação global. O OASIS promove o consenso da indústria e produz padrões mundiais de privacidade, internet das coisas (IoT), computação em nuvem, energia, tecnologias de conteúdo, gerenciamento de emergências e outras áreas. Os padrões abertos do OASIS oferecem o potencial de reduzir custos, estimular a inovação, aumentar os mercados globais e proteger o direito à livre escolha de tecnologia. [3]

O protocolo MQTT tem se tornado bastante utilizado em soluções de IoT devido a sua eficiência, facilidade de implementação, por exigir uma baixa largura de banda, manter-se confiável mesmo com altas latências, características que o tornam ideal para tal aplicação. Pode-se realizar o monitoramento dos equipamentos, e envio dos comandos através do aplicativo IoT MQTT *Dashboard*.

Faz parte do escopo deste projeto apresentar e comparar algumas destas tecnologias disponíveis no mercado e compará-las visando auxiliar futuros usuários a implementarem em suas localidades.

Quando possível, será implementado ainda comandos via *Google Assistant*, para que se compare a usabilidade utilizando não somente os comandos nativos de cada método, mas também comandos por voz.

O trabalho será dividido em 6 partes, a primeira contempla a introdução (já mencionada) que contextualiza o trabalho feito, explica introdutoriamente os produtos utilizados para o comparativo do trabalho. Na 2ª parte, é explicado detalhadamente sobre as tecnologias utilizadas e a maneira que se deve configurá-las para o correto funcionamento do método. A 3ª parte aborda a metodologia implementada para poder mensurar a eficiência dos métodos, e assim, montar a tabela de comparativos. A 4ª parte apresenta os resultados parciais de cada método individualmente, explica-se as notas atribuídas a eles bem como o motivo do método ter recebido

essa determinada nota. Na 5ª parte apresenta-se as conclusões obtidas através da metodologia utilizada, e é mostrado um comparativo geral, contemplando todas as metodologias utilizadas bem como suas respectivas notas. Possuirá ainda uma parte reservada para os agradecimentos aos nomes que fizeram parte da elaboração deste documento de maneira direta ou indireta. Na 7ª e última parte estão as referências bibliográficas utilizadas para a elaboração deste artigo.

## II. TECNOLOGIAS E SUAS CONFIGURAÇÕES

Foram estruturados 4 métodos para se avaliar aspectos relevantes para a escolha da tecnologia a ser aplicada, baseado no trabalho [4]. Cada um é indicado para determinada aplicação. Neste trabalho foi definido que seria explicado como foi desenvolvida a implementação de maneira suficiente para que qualquer pessoa com relativo nível técnico pudesse reproduzir. Com o intuito de se avaliar os métodos, foram escolhidas algumas aplicações que poderiam ser úteis tanto no dia a dia de uma residência, quanto para ambientes comerciais. São as aplicações:

- Relé;
- Rotina;
- Sensor de Temperatura e Umidade;
- Sensor de Presença.

Essas aplicações são suficientes para que se avalie o método em relação aos outros. Todos os testes são realizados com o intuito de avaliar os métodos com os parâmetros indicados na tabela 1.

Os relés podem ser utilizados para acionamentos de lâmpadas, acionar uma bomba de uma caixa d'água, travas de portas e janelas, etc.

As rotinas são ações pré-programadas onde uma determinada ação gera uma resposta do sistema, como por exemplo: ao chegar em casa e abrir uma porta as luzes se acendem, ou depois de um determinado intervalo de tempo, caso não haja ninguém no ambiente as luzes são apagadas.

Portanto, com esse dispositivo é possível conectar a casa inteira, e controla-la utilizando apenas o *smartphone*. [5]

Tabela 1 - Parâmetros de Comparação entre métodos

Parâmetro
Tempo de Resposta
Facilidade de implementação
Preço
Facilidade de manutenção
Utilização do Usuário
Liberdade de rotina
Acessibilidade
Custo benefício

fonte: próprio autor

Serão definidos ainda pesos de acordo com o desvio padrão das notas que os métodos conseguirem. Notas para cada parâmetro são definidas entre 0 e 10. A nota 10 é dada ao método que é nível de referência em determinado quesito, logo, a cada término de método, as notas são revisadas para que seja o mais justo possível. Todavia, são explicados e definidos os motivos e problemas e comentários durante os resultados dos métodos, justificando-se, assim, cada nota.

Todo este trabalho resultará num produto final que será uma tabela de comparação entre métodos, com notas e pesos, e uma última classificação, ordenando os métodos por custo benefício.

### A. Blynk

Para o aplicativo do Blynk foi utilizada a placa NodeMCU ESP32, que são microcontroladores que já possuem tudo que é necessário para se conectar à internet. São similares a um Arduino, mas com a integração Wi-fi e bluetooth já integrados na placa. [1]

Algumas das muitas aplicações do ESP32 são:

- Tomadas inteligentes;
- Automação residencial;
- Monitoramento remoto;
- Privacidade doméstica, comercial e industrial;
- Rede de sensores;
- Controle industrial sem fio;
- Monitores de bebês e crianças;
- Eletrônica vestível;
- Dispositivos para localização via Wi-Fi;
- Câmeras IP;
- Robótica;
- E muito mais; [6]

As características da placa são:

- Memória ROM: 448 Kbytes;
- Clock máximo: 240MHz;
- Memória RAM: 520Kbytes;
- Memória Flash: 4MB;
- Wireless padrão 802.11 b/g/n;
- Conexão Wi-Fi de 2.4GHz (máximo de 150 Mbps)
- Antena embutida na placa;
- Conector micro USB para comunicação e alimentação;
- Bluetooth BLE 4.2;
- Tensão de operação: 4,5 ~ 9V;
- Conversor analógico digital (ADC);

Os módulos ESP8266 são fornecidos numa ampla variedade de modelos, com diferenças perceptíveis principalmente no que tange à quantidade de I/Os disponíveis para acesso externo. A figura 4 ilustra as variantes da ESP8266.





Figura 1- Família do ESP  
fonte: [7]

Foram realizados alguns testes com o objetivo de qualificar o método frente a algumas aplicações (acionamento de LEDs e relés através de sensores e por voz). O objetivo desses testes foram automatizar algumas aplicações cotidianas para então mensurar e quantificar a eficiência do método.

Para iniciar a implementação do método é necessário primeiro ter instalado em seu *smartphone* o aplicativo do Blynk, e criar a sua conta como visto na Figura 2.

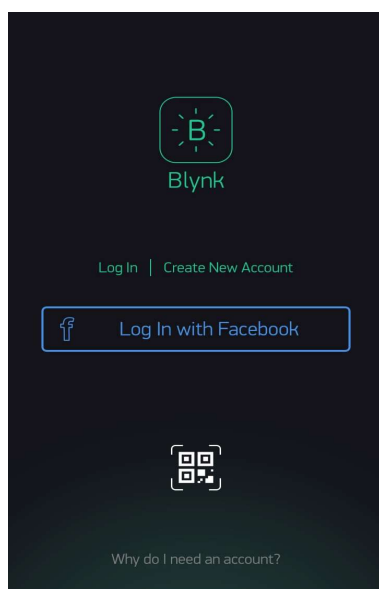


Figura 2- Interface inicial do Blynk  
fonte: próprio autor

Com o cadastro feito, pode-se então criar o primeiro projeto, configurando o dispositivo (NodeMCU, no nosso caso) e o tipo de comunicação utilizada, como pode ser visto na Figura 3.

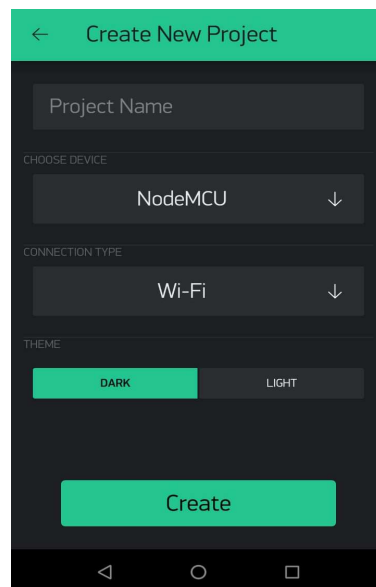


Figura 3- Criando um novo projeto  
fonte: próprio autor

Após a criação do projeto, recebe-se um e-mail informando o *authtoken*, que se trata da chave de segurança para poder acessar seu projeto (e será a chave que deverá ser definida nos programas que se irá passar ao ESP32.) abre-se a página inicial de controle do projeto (vazia, até o momento). Na qual se pode inserir diversos botões/ gráficos de monitoramento e acionamento dos nossos dispositivos, cabe então ao implementador da automação escolher a solução ideal para a situação.

Na figura 4, pode-se observar um exemplo de Layout, utilizado nos testes do aplicativo



Figura 4- Exemplo de Layout  
fonte: próprio autor

Embora o aplicativo seja excelente há uma limitação de soluções gratuitas que podem ser selecionadas para o seu Layout, essa limitação é chamada de “energias” pelo aplicativo, e cada uma dessas soluções absorve uma certa quantia de energia.

Como já dito anteriormente, cada projeto possui um *authtoken* que é automaticamente enviado ao e-mail cadastrado no projeto no instante de sua criação. Esse



*authtoken* deve ser utilizado no código para programação do esp, visando utilizar o método Blynk.

Para programação foi utilizado o ambiente de desenvolvimento do Arduino (IDE Arduino), há então a necessidade de adicionar algumas bibliotecas, que são disponibilizadas gratuitamente.

Após feitas as configurações necessárias, deve-se então abrir o código exemplo do Blynk ESP8266 na IDE do Arduino e fazer as alterações necessárias para que o método funcione adequadamente.

Após realizadas as alterações no código exemplo do NodeMCU e transferi-lo para a placa de desenvolvimento, pode voltar ao aplicativo do Blynk, e seleciona-se o botão “play” para executar o projeto. Caso o dispositivo esteja energizado e com a comunicação funcionando adequadamente, será visualizada uma notificação na cor amarela no canto inferior esquerdo de sua tela com a seguinte frase “(nome do projeto) *connected*” que significa que o seu dispositivo foi reconhecido e está conectado ao seu projeto. A partir desse momento você poderá controlar e monitorar as portas do seu dispositivo através do layout montado em seu projeto.

## B. Blynk utilizando o Google Assistant

De maneira similar ao método descrito anteriormente, neste caso há adição do *Google Assistant* para auxiliar na automação.

É necessário possuir os aplicativos instalados, semelhante ao método anterior (Blynk, *Google Assistant* e *google home*), criação do projeto na interface do Blynk, configurar o código, etc.

O próximo passo então é vincular o *Google Assistant* ao NodeMCU utilizando o método de integração IFTTT.

IFTTT (*If This Than That*) é uma plataforma que auxilia na tomada de decisões nos dispositivos de automação, podendo escolher diversos gatilhos diferentes, no nosso caso, o gatilho serão frases captadas pelo comando de voz do *Google Assistant*. [8]

Para que o *Google Assistant* se comunique de maneira adequada e possa realizar os comandos da automação é necessário criar uma conta no IFTTT (pode ser vinculada a sua conta do Google). Após isso, deve-se então criar o seu *applet*. Para isso você deve escolher entre diversas opções de gatilhos (*Spotify*, *Youtube*, *Facebook*, etc), no caso, utiliza-se o *Google Assistant*. Depois de selecionar o atilho, selecionar então a ação que será tomada ao receber tal gatilho, nesse caso, utiliza-se o *Webhooks*, pois tem a capacidade de se comunicar com o Blynk, utilizando o link [9].

O *Google Assistant* ao receber o comando de voz específico irá enviar uma “ordem” ao IFTTT que será traduzido de forma que o *Webhooks* crie um gatilho que será enviado ao Blynk. Após o Blynk receber esse comando, será então enviado o sinal para que o NodeMCU realize a programação utilizando o link [10].

Deve-se alterar os campos “*authtoken*” pelo *authtoken* gerado ao criar o seu projeto, “PINO” pelo pino que se deseja comutar (uma observação importante é que nesse campo o *Pinout* deverá ser baseado no Arduino, e não no NodeMCU).

O *Webhooks* irá então enviar a ação ao *Blynk* que fará com que o NodeMCU execute a ação.

Primeiramente foram testadas aplicações para o acionamento de LEDs, com o botão através do método do

Blynk, pôde-se também alterar a intensidade luminosa utilizando um PWM. Verificou-se também o funcionamento através de um sensor de presença utilizando uma rotina. Estes foram os testes básicos.

Testes similares foram feitos para o método com o *Google Assistant*, exceto o de controle de luminosidade com o PWM.

Prosseguiu-se então para testes com aplicações mais reais, como o acionamento de lâmpadas, travas elétricas, etc.

Também se verificou a possibilidade de utilizar o *spotify* através de comandos dados ao *Google Assistant*.

## C. SonOff

Testou-se nesse método a linha de dispositivos para automação residencial feitos pela empresa ITEAD, *SonOff*. Marca largamente comentada em diversos fóruns e bastante utilizada no mundo.

Para poder testar o método e comparar com os demais métodos implementados escolheu-se alguns dispositivos específicos que cumprissem com os requisitos de tópicos de teste criados neste projeto. Foram adquiridos os seguintes módulos:

- *SONOFF BASIC* Interruptor;
- *SONOFF TH10* Interruptor;
- *SONOFF CT60 PIR2* (sensor de movimento);
- *SONOFF Gateway* de ponto RF Wi-Fi 433 MHz;

Para iniciar o método é necessário possuir o aplicativo *eWelink* instalado, criar uma conta e configurá-lo em seu smartphone.

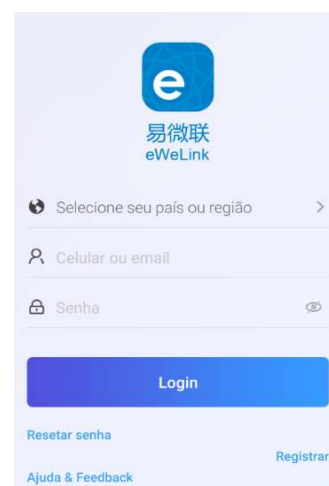


Figura 5- interface inicial do eWelink  
fonte: próprio autor

Após a criação da conta, deve-se vincular os dispositivos para continuar a configuração. É por este aplicativo que se vai monitorar e controlar todas as aplicações do *SonOff*.

Para associar um dispositivo deve-se clicar na tela + que se encontra na parte inferior do dispositivo, onde irá abrir uma nova tela que dará as opções de emparelhamento.

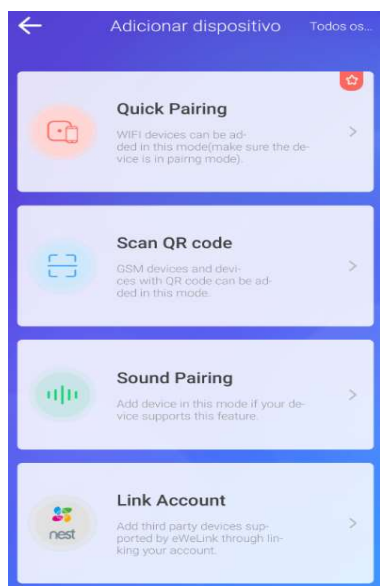


Figura 6- associação de dispositivos  
fonte: próprio autor

Para os dispositivos escolhidos para os testes, utilizou-se a primeira opção, de emparelhamento rápido, que como próximo passo, deve-se colocar o dispositivo no modo de pareamento. Cada módulo possui um botão que deve ser pressionado e segurado por aproximadamente 5 segundos, para que entre no modo pareamento. Logo após isso, uma luz piscará e então aguarde até que o dispositivo seja reconhecido pelo *smartphone*.



Figura 7- SonOff Basic (botão de pareamento em destaque)  
fonte: [11]

É de extrema importância que o *smartphone* esteja conectado ao *Wi-Fi* local. Pois somente assim ele “enxergará” o módulo. Após o reconhecimento já é possível comutar o estado do dispositivo utilizando o *smartphone*. O processo é simples e rápido.

#### ➤ SonOff Basic

Este módulo é o que foi utilizado para demonstrar como associar um dispositivo modular SONOFF ao *smartphone*. É o produto mais básico da linha, possui uma variação que pode ser acessada por RF.

Especificações:

- Tensão de trabalho: 90 – 250 Vac (50/60 Hz);
- Corrente máxima: 10A;
- Potência máxima: 2200 W;
- Dimensões: 88x38x23 mm;
- Temperatura de operação: 0° – 40° C;
- Humidade de operação: 5 – 90 % RH;
- Frequência operacional: 2,4 GHz;

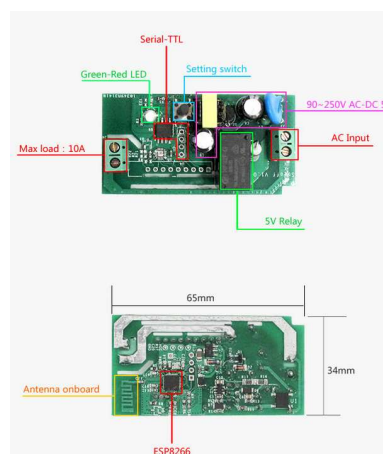


Figura 8- hardware interno SonOff basic  
fonte: [12]

Todos os módulos possuem servidor em nuvem e se conectam pelo mesmo SSID. Foi adicionado recentemente a possibilidade de se conectarem localmente, não necessitando desse servidor em nuvem.



Figura 9- ilustração do trabalho em nuvem  
fonte: [12]

Como se pode ver na figura 9, o módulo possui um relé que comuta utilizando os comandos vindos do *smartphone*. Pode-se utilizar ainda o widget do aplicativo *eWelink*, que demonstra o estado e dá mais acessibilidade ao usuário.

#### ➤ SonOff TH10

O *SonOff TH10*, tem o funcionamento igual ao do modelo Basic, sua diferença reside na capacidade de medir temperatura e umidade do ar, através do sensor DHT22. Sua parametrização e tomada de decisões podem ser controladas através da interface do próprio aplicativo, que informam os seus dados em tempo real.



Figura 10- SonOff TH10  
fonte: [13]



Figura 11- interface de monitoramento TH10  
fonte: próprio autor

Esse módulo pode ainda ser usado, juntamente com outros sensores, como por exemplo:

- Si7021 – sensor de umidade;
- AM2301 – umidade e temperatura;
- DS18B20 – sensor de temperatura (a prova de água);
- DHT11- sensor de temperatura e umidade;

Foi utilizado o sensor DHT11 na implementação.



Figura 12- TH10 em funcionamento  
fonte: próprio autor

A luz azul, funciona como indicador para informar que o módulo está conectado ao servidor. O vermelho indica o estado do relé.

Especificações:

- Tensão de funcionamento – 90 – 250 Vca;
- Corrente máxima – 10 A;
- Potência – 2200 W;
- Temperatura de operação: 0° – 40° C;
- Umidade de operação: 5 – 90 %;
- Frequência operacional: 2,4 GHz;



Figura 13- sensor DHT11  
fonte: próprio autor

#### ➤ SonOff CT60 PIR2

Foi utilizado o sensor de presença da linha SonOff para poder testar a utilização de módulos sensores de comunicação de RF.



Figura 14- SonOff CT60 PIR2  
fonte: [14]

Seu funcionamento é bem simples, visto que sua função é somente de sensoriamento. O sensor é carregado por duas pilhas do tipo AAA e demanda uma corrente de aproximadamente 30 mA. O sensor emite uma comunicação em tecnologia de rádio frequência (433 MHz). Faz se necessário nessa aplicação o uso de um gateway para poder realizar a comunicação com o servidor, que traduz o sinal para o servidor em nuvem.

Seu pareamento é bem parecido com o de um módulo comum, porém, você deve associá-lo utilizando a ajuda do gateway, pois ele é quem receberá o sinal.

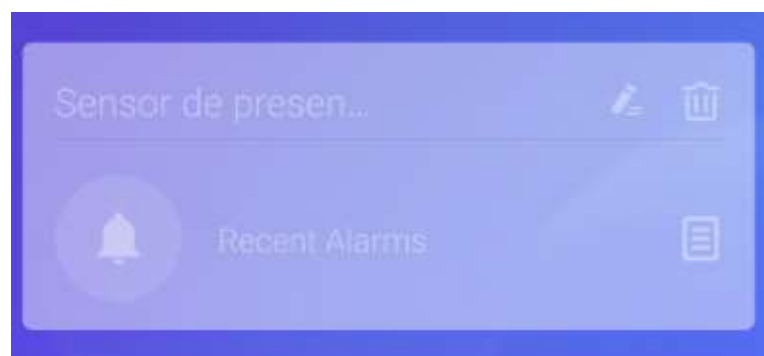


Figura 15- interface com PIR cadastrado  
fonte: próprio autor

➤ SonOff Gateway de ponte RF

Utilizou-se ainda o *gateway* para realizar a comunicação entre os sensores em RF e o servidor *Wireless*. Ele precisa de uma fonte e trabalha com tensão de 5V.

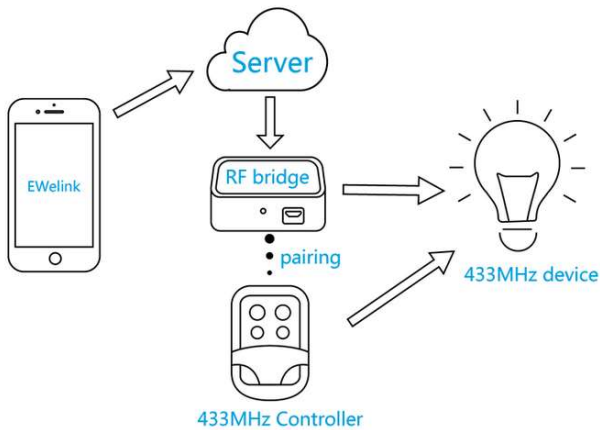


Figura 16- Fluxo de trabalho gateway-servidor  
fonte: [15]



Figura 17- gateway SonOff RF 433 MHz  
fonte: [16]



Figura 18- gateway SonOff em funcionamento  
fonte: próprio autor

Para conectar o aplicativo-servidor *eWelink* com o servidor da *Google*, foi utilizado o aplicativo *Google Home*, que vincula as contas *Google* e *eWelink*. É possível renomear os dispositivos e separá-los por locais (casa1, casa2, comércio...) facilitando o mapeamento no caso de vários dispositivos conectados juntos. Os relés estão cadastrados e o aplicativo indica o estado (On/Off) do relé.

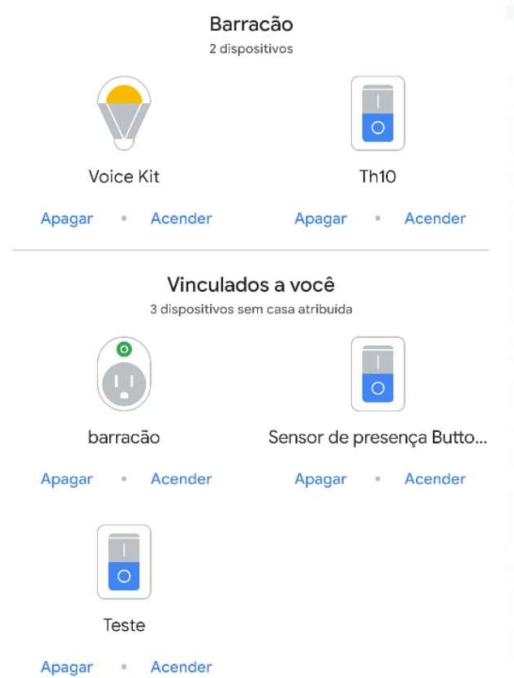


Figura 19- interface de controle no app Home da Google  
fonte: próprio autor

#### D. MQTT

O protocolo MQTT funciona com uma metodologia de publicação e assinatura (*publisher* e *subscriber*). Há um maior nível de segurança pois o foco é no transporte de dados, e não no conteúdo em si, ou seja, há uma garantia de entrega dos dados devido aos vários níveis de serviço.

A funcionalidade primária era a comunicação de sistemas de telemetria de oleodutos via satélite, ou seja, um protocolo feito para suportar comunicações instáveis. É executado através de protocolos de rede, por exemplo TCP/IP, ou qualquer outro protocolo de rede que forneça conexões sem perdas.

Definições:

**Publisher** – Cliente que envia os pacotes de dados, ou seja, o publicador.

**Subscriber** – Recebe os pacotes de dados enviados pelo *Publisher*, é o assinante.

**Broker** – Realiza o recebimento dos dados enviados pelo *Publisher*, e os armazena fazendo a distribuição para os respectivos *subscribers*.

**Message** - Pacote de dados trafegados entre clientes e *broker*

**Topic** – Repositório de dados único que recebe a mensagem, acessível aos clientes, instantâneo e sem histórico.

**Publish/Subscribe** – Ações de enviar/receber uma mensagem do *broker* para/de um determinado tópico.

**Payload** – Conteúdo dos dados enviados na mensagem.

**QoS – quality of service:**

- 0 – a mensagem poderá ser recebida no máximo uma vez;
- 1 – garantia que a mensagem será recebida no mínimo uma vez;
- 2 – a mensagem será recebida exatamente uma vez;

Para realizar a comunicação *Wireless* entre o ESP e o smartphone, utilizou-se o protocolo MQTT. O ESP possui uma biblioteca de comunicação chamada "*PubSubClient*", que consegue utilizar o hardware do ESP para comunicar com



o *broker*, independente se é local ou em nuvem. Esta biblioteca foi acrescentada na IDE do Arduino para que se possa programar o microcontrolador sem a necessidade de uma programação extensa e pesada para realizar essas comunicações, facilitando assim o seu acesso há um maior número de usuários.

Testou-se como ponto de partida, um projeto realizado pelo grupo FelipeFlop, que utilizou o ESP para controlar sua saída, utilizando o mesmo protocolo de interesse (o código utilizado, poderá ser encontrado no apêndice). O código se inicia com a comunicação *Wi-fi* importada pela biblioteca “ESP8266WiFi.h”. Após estabelecida essa comunicação são realizadas sucessivas tentativas de conexão ao *broker* indicado. As ações deste dependem do que se deseja realizar em função dos parâmetros envolvidos no sistema montado e neste caso, foi escrito para que pudesse acender o LED embutido na placa, pois a conexão de um relé para controlar saídas direto da rede, é similar ao método do *Blynk*, que foi utilizado e comprovado o seu funcionamento.

Logo após definir o programa que será transmitido ao ESP, foi decidido utilizar um *broker* em nuvem. Utilizou-se ainda outro trabalho para auxiliar na definição do *broker* que seria escolhido para o teste. Utilizou-se o trabalho do Olair Ricardo Junior da Universidade Estadual de Londrina, que enfrentou os mesmos problemas citados. O *broker* escolhido foi o HIVEMQ, que utiliza um *broker* em nuvem compartilhado e gratuito, com condições (utilização limitada para usuários gratuitos).

Sua utilização de visualizar o funcionamento do código escolhido. Após o ESP se comunicar, deverá se escolher outro *client* para estabelecer os comandos, no caso, o smartphone que é o objeto de estudo. Instalou-se o aplicativo baseando-se ainda no projeto do Olair Ricardo Junior, já citado, foram feitos os testes.

Após a constatação do funcionamento do mesmo, o LED embutido do microcontrolador piscando de acordo com os comandos enviados, é momento de se estabelecer o servidor como local.

Utilizou-se então uma Raspberry pi zero W, para ser o *broker* localmente, em substituição do HIVEMQ, utilizado anteriormente. Foi criado o servidor local, que se comunica via *Wi-fi* com os *Clients* cadastrados na rede. Uma pequena alteração deve ser feita na programação, substituir o endereço do servidor pelo endereço de IP da Raspberry conectada à rede, seguida da indicação da porta de comunicação a ser utilizada, 1883 no nosso caso.

### E. Bancada de Testes

Os dispositivos foram implementados, utilizando os mesmos materiais em todos os métodos, para padronizar a comparação. As aplicações da lâmpada e trava eletrônica foram reunidas num mesmo local para facilitar a transição entre os módulos e as aplicações.

Colocou-se então, uma placa de madeira, 3 conectores para lâmpada, uma trava eletrônica (a fonte para acionamento da trava foi alocada na parte traseira da placa de madeira).

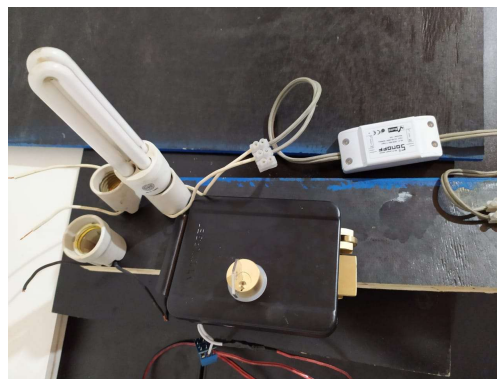


Figura 20- materiais usados no método SonOff  
fonte: próprio autor

## III. METODOLOGIA

Com a experiência adquirida nos métodos e utilização, construiu-se uma tabela com todos os valores de nota adquiridos. Utilizou-se o procedimento descrito no começo deste documento para que se quantifique as notas dos métodos.

Para o cálculo da média, comparando todos os métodos, optou-se em colocar duas opções para que o leitor se decidisse qual seria mais cabível para sua utilização.

O primeiro cálculo foi baseado na experiência como usuário, e com isso dá para se ter uma direção do que se deve observar em um método ou produto.

Os parâmetros, como citados anteriormente são:

- Tempo de Resposta – O tempo que leva para o comando ser executado ou a informação ser recebida;
- Facilidade de Implementação – Nível de Trabalho e conhecimento técnico que o usuário precisa ter para implementar o método;
- Preço – Custo total para a implementação dos requisitos para os testes;
- Manutenção – Relacionada a facilidade que o usuário terá de rastrear problemas e consertá-los;
- Acessibilidade aos comandos – Tempo médio que o usuário terá para acessar aos comandos e facilidade de acesso aos mesmos;
- Liberdade de Rotina – Liberdade que o usuário tem de criar rotinas, de modo que se encadeie comandos de acordo com gatilhos pré-estabelecidos;
- Acessibilidade – Facilidade do usuário adquirir os equipamentos para poder utilizar.

É importante ressaltar que a pontuação do tempo e do preço é calculada inversamente proporcional ao seu valor, de maneira que o menor tempo e o menor custo receberão as maiores notas, e os demais métodos receberão notas proporcionais a estas.

O cálculo para conversão inversamente proporcional é calculada com a equação (1).

$$Nota = Notam\acute{a}x * \left( \frac{Menor\ Medida}{Medida} \right) \quad (1)$$

Utilizando disso, foi parametrizado as maneiras de se obter essa nota. Com as aplicações escolhidas, pode-se notar a usabilidade para se executar comandos, a velocidade de resposta e os outros itens citados acima. São elas:

- Relé, onde pode-se conectar diversos equipamentos nos seus terminais, como lâmpadas, portões, motores, travas eletrônicas e etc.

- Sensor de Temperatura e Umidade, para aplicações de monitoramento de ambiente.

- Sensor de Presença, por ser bastante utilizado em residências para iluminação e outras aplicações como alarmes.

- Rotinas, para que se veja qual é o grau de liberdade que o usuário tem de deixar sua residência incorporada ao seu dia, tomando as próprias decisões baseadas em configurações do usuário e nas informações adquiridas pelo próprio equipamento.

Com estas aplicações já se pode desenvolver diversos sistemas inteligentes e com isso infinitas utilizações tanto residenciais quanto comerciais.

De acordo de como foi a montagem e configuração, dificuldade para acessar a tecnologia ou qualquer outra dificuldade, será relatada e sua nota cai em relação aos outros métodos.

#### IV. RESULTADOS

Os resultados a seguir relatam como ocorreram os testes e impressões sobre a utilização e configuração. Todas as tecnologias foram configuradas e testadas para todas as aplicações definidas anteriormente.

Todas as tecnologias utilizaram todas as aplicações definidas, para que se utilizasse uma comparação justa.

##### A. Blynk

Decidiu-se utilizar primeiramente o método blynk. Todo o projeto precisaria de um computador (para instalar a IDE do Arduino), uma placa Esp32 e um smartphone. Encontrou-se o procedimento no próprio site do fabricante e tudo foi realizado de acordo com o esperado no que se trata de encontrar os códigos e suas bibliotecas de desenvolvimento. No momento em que se foi programar o ESP32, encontrou-se diversas dificuldades para configurar a IDE. Foi procurado em diversos projetos previamente realizados que tenham temas parecidos como foi o caso de [1] e [6], além do auxílio de [8], [9] e [7]. Foi somente com algum tempo de testes que foi configurado corretamente e logo após isso não se relata outro problema com este tópico. As configurações utilizadas são:

- *Board: Generic ESP8266Module;*
- *Flash Mode: DOUT (compatible);*
- *Flash Size: 4M (3M SPIFFS);*
- *Debug Port: Serial;*
- *Debug Level: Wifi;*
- *IwIP Variant: v2 Lower Memory;*
- *Reset Method: nodemcu;*

- *Crystal Frequency: 26 MHz;*
- *SSL Support: All SSL ciphers (most compatible);*
- *VTables: Flash;*
- *Espressif FW: nonos-sdk2.2.1 (legacy);*
- *Flash Frquency: 40MHz;*
- *CPU Frequency: 80MHz;*
- *Exceptions: Enabled;*
- *Velocidade de comunicação: 115200[bit/s];*
- *Erase Flash: Sketch + Wifi Settings.*

No momento em que se constatou a ativação do pino, foi testado o mesmo utilizando um relé para acionamento de uma lâmpada. Observou-se que o relé estava sempre considerado acionado pelo relé.

Foi verificado que o relé não se acionava naturalmente utilizando o esp32, como se utilizando Arduino. Este problema se deve pelo fato de que ao se adquirir o módulo pronto, ele possui um circuito já construído, feito para 5v, e o sinal do esp possui 3,3V. Foi utilizado então, um transistor chamado BC547 para que o mesmo funcionasse como uma chave [17], alimentando com 5V ao lado de +VCC para que ao invés do sinal de 3,3V, fosse utilizado o de 5V e não fosse necessário reconstruir a placa de circuito que já existe no relé.

Foram feitos então estes ajustes e não houveram mais problemas com o nível de sinal enviado ao NodeMCU e consequentemente com os acionamentos. O esquema utilizado se encontra na figura 21.

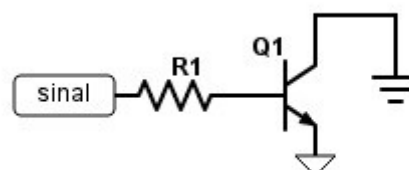


Figura 21- Circuito utilizado para conectar o relé  
fonte: próprio autor

Concluiu-se com sucesso então, os acionamentos das lâmpadas e outros itens como portão eletrônico e outros.

Verificou-se ainda que o sensor adquirido possuía uma sensibilidade que necessitava de uma determinada distância para se dispor do acionamento. Foi adquirido então o sensor da figura 22 para que fosse possível realizar os testes em ambientes pequenos.



Figura 22- Sensor de Presença utilizado com reguladores  
fonte: Usina Info[18]

Pelos motivos acima apresentados, definiu-se as seguintes notas ao método:

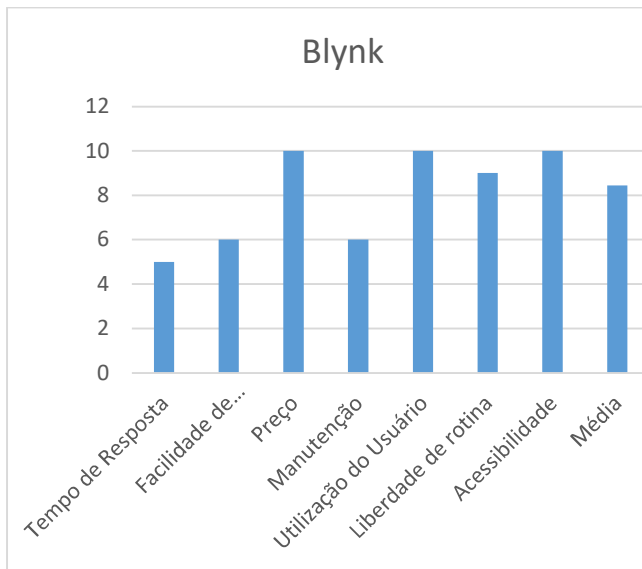


Figura 23- Distribuição de Notas Método Blynk  
fonte: próprio autor

Neste gráfico da figura 23, observa-se que o pior indicador fica com o tempo de resposta do método. Como o fluxo de trabalho dele é longo, (servidor em nuvem) acaba por ter um maior tempo para executar os comandos.

### B. Blynk + Google Assistant

Primeiramente não foi possível conectar diretamente o aplicativo Blynk com o app do *google home*, que controla os dispositivos voltados para automação residencial. Foi necessário que se utilizasse um aplicativo intermediário que funciona criando links entre as plataformas disponíveis.

Conectou-se então a conta do *google* no *ifttt* [5] e acessou-se o servidor que determina os comandos do *blynk* através de um endereço do servidor (esta conexão se chama *Webhooks*). Configurou-se um *Applet* (uma conexão entre plataformas) onde era colocado alto ao introduzir determinada frase no *google assistant* (“*Let there be light!*”) e baixo ao dizer a frase escolhida (“*Make it darker*”), como se pode ver na figura 24. Os *applets* criados podem ser acessados ao pesquisar. Colocou-se ainda respostas customizadas para os comandos do *google*.

What do you want to say?

Make it darker

What's another way to say it? (optional)

Turn off test one

And another way? (optional)

test one off

What do you want the Assistant to say in response?

fear the dark

Language

English

Figura 24- Configuração Google Assistant IFTTT  
fonte: próprio autor

### Make a web request

This action will make a web request to a publicly accessible URL.  
NOTE: Requests may be rate limited.

#### URL

`http://cloud.blynk.cc:8080  
/e9a2720d33d546e9b927448f8826316e/pin/D5`

Surround any text with "<<>>" to escape the content

Add ingredient

#### Method

PUT

The method of the request e.g. GET, POST, DELETE

#### Content Type (optional)

application/json

Optional

#### Body (optional)

[<<<0>>>]

Surround any text with "<<>>" to escape the content

Add ingredient

Figura 25- Configuração Webhooks  
fonte: próprio autor

Caiu-se no problema ainda de se encontrar o link referido na figura 25, porém, encontrou-se procurando novamente em projetos anteriores.

O URL define o endereçamento do comando. O método define a natureza do comando. Considerou-se aqui que o comando será do tipo que vai enviar uma definição a ser cumprida. As outras opções de manipular dados são diversas, como: buscar, enviar, deletar, alocar e etc.

Quanto ao resto da utilização, o método provou-se bastante eficiente e não deixou a desejar quanto a facilidade de implementação. Toda a instalação, programação, acessibilidade e manutenção provou-se bastante simples e não exige extremo conhecimento técnico para que seja realizado.

Constatou-se, porém, certa dificuldade na reconexão em certos momentos em que a internet oscilava e reiniciava o dispositivo, o que o torna mais complicado de ser implementado em locais com necessidade de maior precisão.

### C. Sonoff

Neste método, observou-se maior constância no quesito de dificuldades técnicas. Por ser um método mais profissional e consolidado no mercado, possui uma plataforma mais robusta e seu uso já é comumente usado ao longo do mundo. Este método foi escolhido para que o projeto tivesse determinados parâmetros já comprovados no mundo, porém que ainda não fossem largamente utilizados no Brasil e muito menos de alto valor, pelo contrário, os módulos têm baixo valor e por isso ainda se encaixam no tema do projeto.

Toda a implementação foi extremamente rápida e não foram encontrados empecilhos para a realização das atividades programadas.

Deve-se, porém, comentar, que em comparação com os outros métodos testados, este possui uma menor liberdade para criar rotinas pois enquanto os outros demandam certo nível de programação, este não necessita. Por outro lado, demandando menor nível técnico para a implementação,



aumenta o número de usuários que poderiam utilizar. Conforme segue na figura 26.

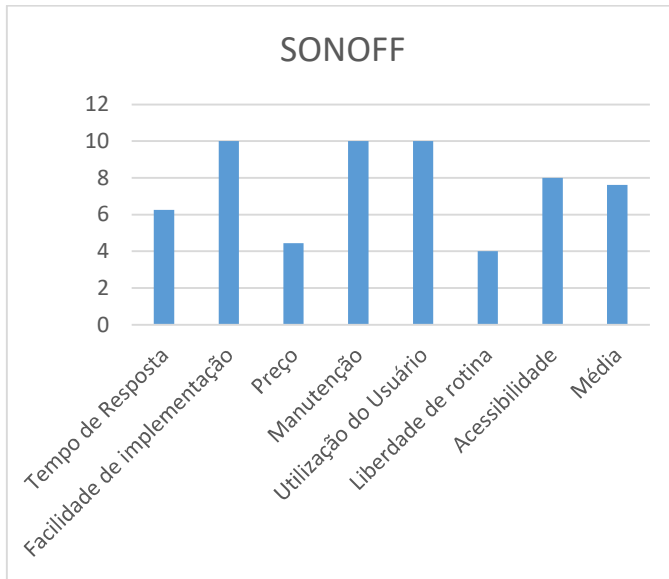


Figura 26- Distribuição de Notas Sonoff  
fonte: próprio autor

#### D. MQTT

Antes de ser definido, foi testado o *openhbab* como comunicação para que tivesse uma maior acessibilidade com os usuários, visto que é uma plataforma de uso livre e conta com uma maior quantidade de possibilidades do ponto de vista de conexão. Este pode se conectar com equipamentos por MQTT, *sonoff*, *blynk*, *google home* e etc. Este método estava definido como o que se iria testar. Porém, observou-se que por ainda ser relativamente novo, encontra-se ainda em fase de construção em determinados aspectos. Por diversas vezes foram encontrados problemas onde a comunidade, ainda pequena, esperava uma solução. Foi testado utilizar na placa de desenvolvimento raspberry pi zero w, instalar o sistema operacional próprio desta plataforma. Constatou-se diversas dificuldades em termos de configurar a raspberry pi zero, visto que possuía uma interface mais simples e modificada. A documentação no site, explicava qual deveria ser o procedimento para realização, mas ao encontrar alguns erros de inicialização e configuração, expressavam uma mensagem do tipo:

*“Attention! The openHABian setup process seems to have failed on your system. Sorry, this shouldn't happen! Please restart the installation process. Chances are high setup will succeed on the second try...”*

Onde a tradução é:

“Atenção! O processo de instalação do openHABian parece ter falhado no seu sistema. Desculpe, isso não deveria acontecer! Por favor, reinicie o processo de instalação. As chances são altas da configuração ser bem-sucedida na segunda tentativa.”

Algumas partes da documentação eles expõem que ainda está em desenvolvimento, portanto, não possui documentação. Está como “A ser feito. Esta parte ainda não foi escrita. Desculpe...”

Foram encontradas dificuldades ainda na construção da IHM que poderia ser customizada para cada usuário. Porém deve-se configurá-la através de um servidor criado localmente com alguma máquina rodando o sistema. Observou-se ainda que este servidor era instável e

constantemente deveria atualizar a página para reconectar e prosseguir com a configuração.

Foi decidido então que seria utilizado apenas o MQTT. Seria primeiramente testado com o servidor em nuvem, onde ainda não iria necessitar da configuração de um servidor local. Este funcionou de maneira correta e não necessitou contorno de problemas, após acertado o programa que seria utilizado e as configurações da IDE do Arduino.

Testou-se então o servidor local. Para que se tivesse um método que não dependesse de internet, mas somente do sinal *wifi*, caso fosse necessário.

O servidor não foi problema nenhum neste método, foram utilizadas as referências [10], [19], [20] e [21] como modelos. Como os sensores já haviam sido configurados e os erros tratados, não foram observados problemas nestes.

Pode-se dizer que este é o método que mais demanda conhecimento técnico. Foi necessário que se estudasse o funcionamento da estrutura de recebimento e envio de dados, para que se utilizasse o servidor da maneira correta. A interface não é muito convidativa, porém, existem métodos para que se melhore e interaja com outras plataformas. Neste projeto foi somente realizado o trabalho suficiente para que se observasse o método funcionando e avaliasse frente aos outros.

As informações foram mais rápidas, posto que são trabalhadas localmente e não demandam servidor externo.

A necessidade de reprogramar os clientes, toda vez que o servidor mudar de endereço IP também é um grande limitador deste método, que torna mais complicado de realizar alguma alteração na rede ou caso desligue os aparelhos e no momento em que o broker reconectar, entre com outro endereço IP.

É possível então separar esta tecnologia em dois métodos:

- MQTT com servidor em nuvem;
- MQTT com servidor local.

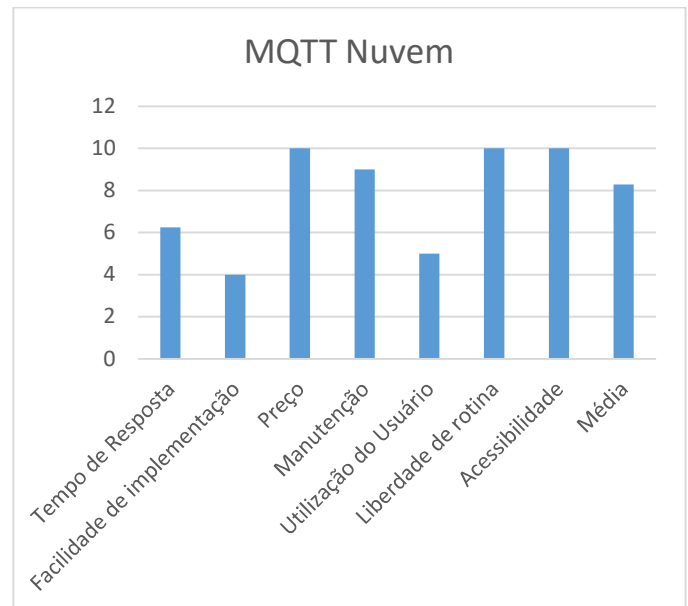


Figura 27- Distribuição de Notas MQTT em Nuvem  
fonte: próprio autor

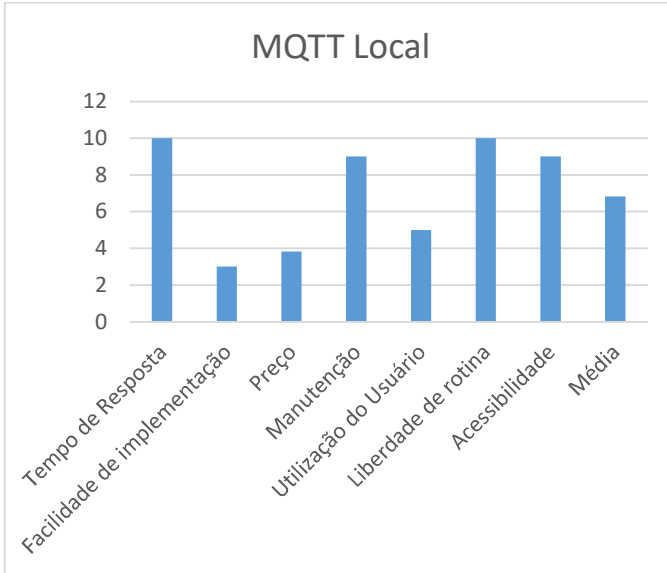


Figura 28- Distribuição de Notas MQTT Local  
fonte: próprio autor

As notas referentes a estes métodos variam no investimento necessário para utilização, pois um necessita do servidor local, que representa um custo a mais para implantação, no tempo de resposta, pois o servidor em nuvem possui um tempo de resposta bem maior em relação ao servidor local, pelo fato das informações percorrerem um caminho bem maior e a facilidade de implementação, pois necessita da configuração do servidor local. As figuras onde se pode encontrar a nota correspondente de cada um são as figuras 27 e 28.

Diante do exposto, criou-se uma tabela com todas as notas referentes aos experimentos desenvolvidos. As notas foram distribuídas de maneira que o melhor em cada aspecto fosse nota 10, para que fosse considerado o parâmetro máximo de classificação.

Foi separado o método MQTT em duas partes, com servidor em nuvem e com servidor local. Visto que fez certa divergência na implementação, no investimento necessário para realização e principalmente no tempo de resposta.

No quesito tempo de resposta, o método MQTT mostrou-se mais rápido que os outros, portanto receberia a maior nota. Pelo servidor estar dentro da própria localidade, a informação não necessita viajar grandes distâncias. Vale ressaltar que o *Sonoff* obteve uma atualização que utiliza comunicação dentro da mesma rede, caso possível, o que torna sua resposta mais rápida que os testes anteriores, porém, ainda não alcança a capacidade do MQTT. O *Blynk* não possui esta funcionalidade, portanto fica com a menor nota. Observou-se que a diferença entre os tempos de resposta fica na casa dos milissegundos, justificando assim o desvio padrão pequeno em relação aos outros, como se pode ver na figura 29. As notas foram calculadas utilizando um tempo aproximado de resposta e utilizando deste, foi considerado o menor tempo como 10. As demais foram calculadas utilizando uma proporção inversa. Na figura 29 pode-se ver ainda como o servidor local influencia no tempo de resposta. Segue ainda na figura 30 as medições por amostragem do tempo de resposta.

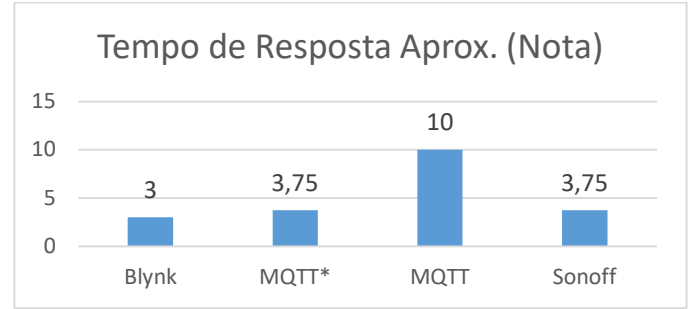


Figura 29- pontuação relativa ao tempo de resposta  
fonte: próprio autor

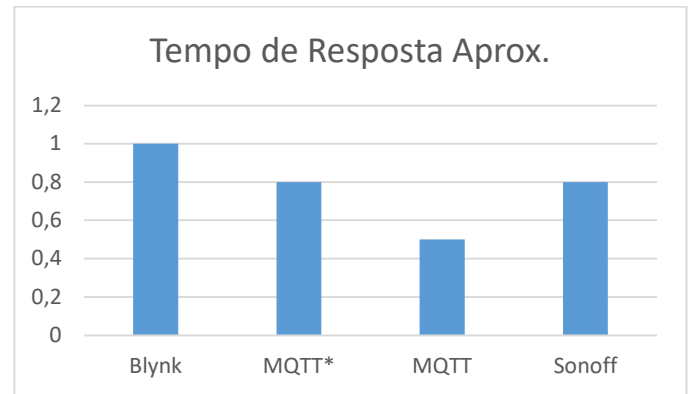


Figura 30 - Tempo de Resposta pego por Amostragem

Se tratando da facilidade de implementação obtém-se uma notável diferença. Os métodos são bem diferentes uns dos outros, com destaque para os dispositivos Sonoff. Os equipamentos já possuem implementação rápida, sem a necessidade de programação. Colocando-o assim com nota superior neste quesito. Os outros métodos demandam maior nível técnico para poder utilizá-los, o que reduz o número de pessoas com acessibilidade a esta tecnologia. O Blynk é ainda menos trabalhoso comparado ao MQTT, visto que já possui o código disponível para *upload* no microcontrolador, configurando sua utilização apenas no aplicativo. A nota foi calculada igualmente o tempo de resposta.

Ao se tratar do preço, foi anotado os equipamentos que foram necessários para implementar cada um deles e podem ser vistos na figura 31. Pode-se observar que o gráfico de notas do tempo de resposta possui semelhança com o de investimento necessário para implementação.

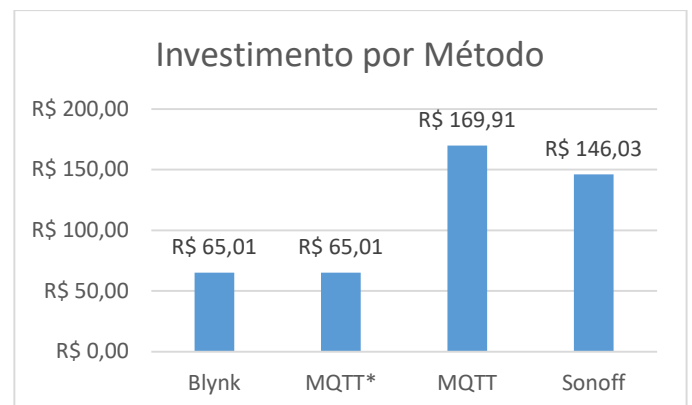


Figura 31- custo para a implementação do método  
fonte: próprio autor

É importante ressaltar que a pontuação referente ao preço foi obtida da seguinte forma: o método que exige menos despesas obtém a nota mais alta, e as demais notas, dos outros métodos

tem notas (menores) calculadas com base no método mais barato.

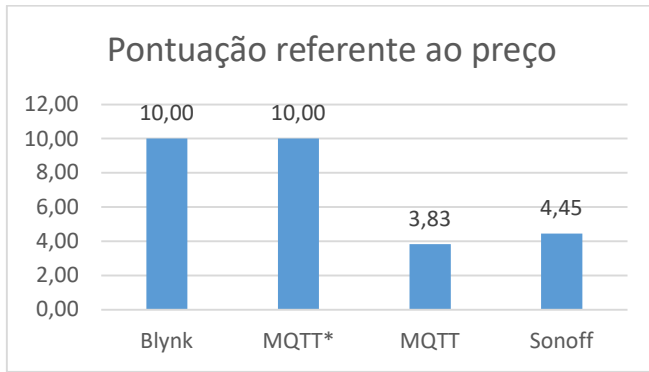


Figura 32- custos já convertidos para as notas  
fonte: próprio autor

O gráfico permite enxergar que existe uma correlação, onde a velocidade de resposta do equipamento está diretamente ligada ao investimento para aplicação. Isso se deve, ao fato de que quanto melhor o servidor, mais oneroso se torna o equipamento ou a aplicação. Mesmo o método de MQTT em nuvem, possuindo seu software gratuito e *OpenSource*, o custo de um servidor dedicado à localidade se torna um empecilho para quem possui recursos mais limitados.

Podemos observar no gráfico da figura 33 o comparativo entre os gráficos de custo e nota de tempo de resposta.

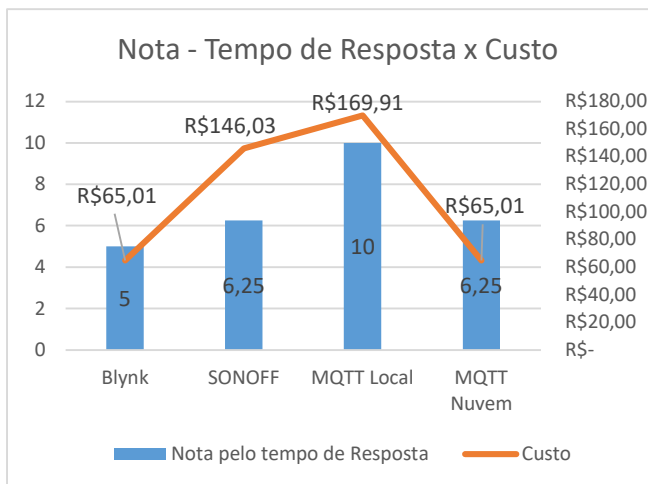


Figura 33 - Gráfico de Nota de Resposta x Custo de Implementação

Para comentar sobre a questão da manutenção, considerou-se o acabamento de cada um dos produtos finais e a sensibilidade a oscilações do equipamento durante os testes. O Blynk foi o que mais se viu sensível às alterações, tanto no fornecimento de internet via *WiFi*, quanto ao desligamento do equipamento. Ele demora a retornar à conexão com o servidor. O Sonoff possui melhor acabamento, visto que é o único que é vendido especificamente para este objetivo, portanto já foi projetado para tais ocorrências. O MQTT não deixou a desejar neste quesito. Tanto utilizando servidor local quanto em nuvem, a velocidade de reconexão foi alta, e não teve queda frente às oscilações da internet *WiFi*. O acabamento pode melhorar, porém, como os materiais não são específicos para este fim, pode-se observar uma maior fragilidade devido a condições de uso.

A utilização do usuário é de extrema relevância pois, por mais que o tempo de resposta seja diferente, o tempo para acessar os comandos é maior. Considerou-se ainda a

capacidade de conexão com o *Google assistant* para que se obtenha uma maior acessibilidade. No método *Sonoff*, observa-se uma grande gama de possibilidades, no que diz respeito a integrar as rotinas ao uso normal do *smartphone*. A criação de *Widgets* faz diferença no uso do dia a dia e isso é mais observado nos métodos *Sonoff* e *Blynk*. No método *MQTT*, constata-se maior dificuldade em criar *widgets* e atalhos. O *Google assistant* pode ser integrado com todos as 3 tecnologias.

A liberdade de rotina diz respeito à integração dos comandos e informações com outras plataformas. O *MQTT* ganha destaque neste, pois como se utiliza programação dos dispositivos, se obtém um maior controle e consequentemente, uma maior liberdade para se adaptar para os comandos e informações e encadeá-las. O *Sonoff* já possui uma maior limitação pois o aplicativo não possui tanta liberdade igual ao do *Blynk* e nem a possibilidade de programação do *MQTT*.

A acessibilidade diz respeito à dificuldade de se obter os materiais necessários para a implementação. O *Sonoff* hoje pode ser encontrado em diversas lojas pela internet até mesmo no Brasil. As placas, microcontroladores como são de uso mais geral são mais facilmente encontrados em lojas de eletrônica, tornando mais popular nas lojas virtuais também. Portanto a diferença das notas não foi grande, porém, o método *Blynk* sai na frente com seu número reduzido de equipamentos necessários, juntamente com o *MQTT* com servidor em nuvem.

Como dito anteriormente foi feita uma média entre as notas adquiridas, com pesos sendo o desvio padrão entre os equipamentos. O resultado é o gráfico que pode ser visto na figura 3.

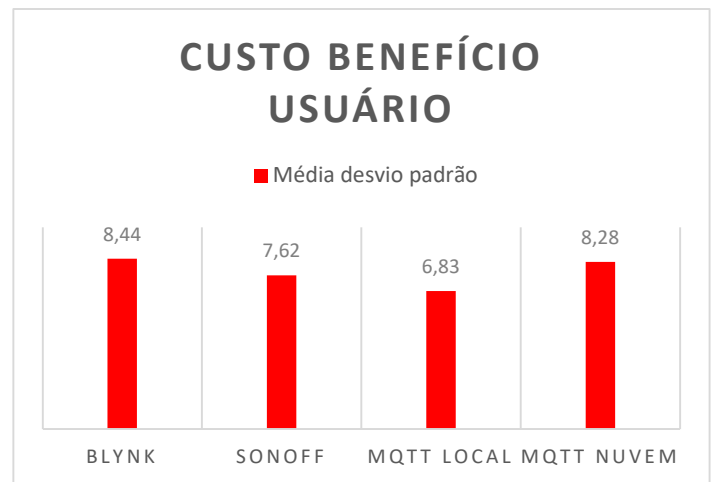
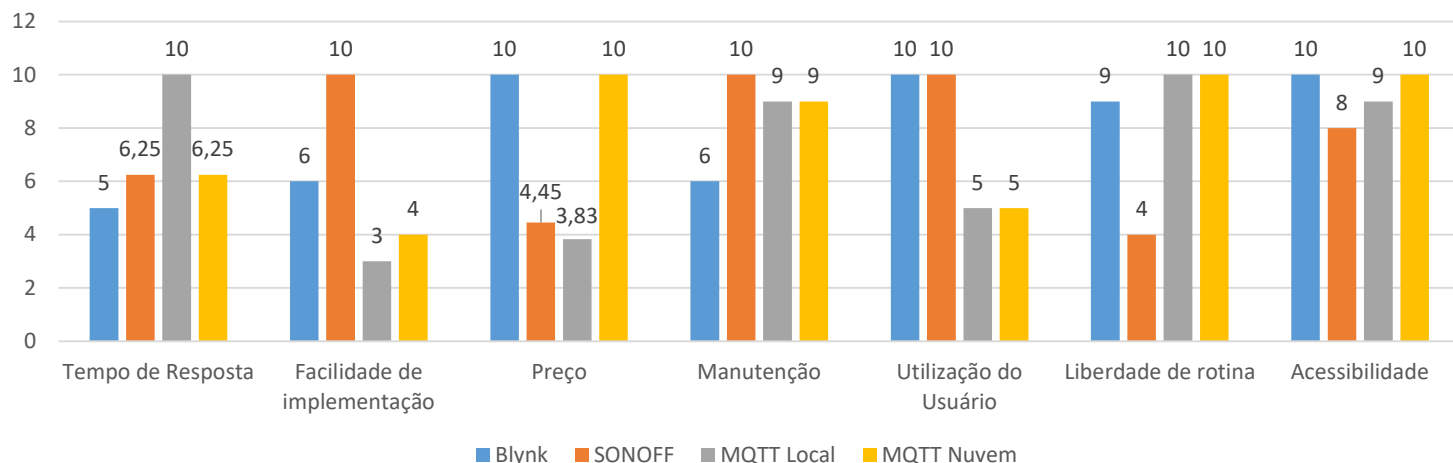


Figura 34- custo benefício dos métodos  
fonte: próprio auto

## Distribuição de Notas



*Figura 35- comparativo geral dos métodos em todas as métricas  
fonte: próprio autor*

## V. CONCLUSÃO

As notas retratam o melhor custo benefício entre os métodos, independente do usuário, considerando apenas o quão diferente são as divergências nos aspectos relacionados. Mas pode-se inferir que cada método poderá ter um público ao qual serviria melhor.

O método Blynk serve mais a entusiastas que gostariam de aprender e controlar um número limitado de aplicações e com comandos simples.

Já o método Sonoff se encaixa melhor à residências e centros comerciais onde se necessita de uma maior eficiência e confiança nas tarefas. Onde se pode utilizar o sistema e não acarretaria mão de obra especializada para a configuração.

O método MQTT local possui um diferencial em relação aos outros pois possui a capacidade de trabalhar offline.

Esta funcionalidade pode chamar atenção para aplicações até mesmo industriais. As informações possuem um maior nível de privacidade e a característica de confirmação de recebimento, configurado pelo QoS, promete uma maior aplicação prática a nível de tomada de decisões através de condições produzidas pelas próprias máquinas ou equipamentos.

Para o futuro, espera-se que o tempo de resposta, tanto quanto o custo, caiam, aumentando suas notas, como é a tendência da tecnologia.

Portanto, seria prudente, um novo trabalho, com explicações e abordando mais tecnologias. Este tipo de conhecimento enriquece e viabiliza o mercado a acessar tais bens. Isto é necessário para que se otimize os processos no país.

## VI. AGRADECIMENTOS

**Mateus:** Gostaria de agradecer primeiramente a Deus, por sempre me dar forças e estar comigo, por me permitir alcançar lugares que sempre sonhei.

Agradeço aos meus pais Josemar Mendes da Silva e Maria de Fátima Araújo Ferreira Mendes por me incentivarem nos estudos e por terem me acompanhado durante toda essa jornada e na graduação.

Agradeço também a minha noiva Dhébora Jessica Andrade Santos por ser minha companheira, me apoiar e auxiliar

durante esta nova etapa da minha vida, e por me motivar a ser sempre uma pessoa melhor.

Também sou grato ao meu professor orientador Geyverson Teixeira de Paula, por toda a paciência e empenho em nos ajudar a elaborar este artigo da melhor maneira possível.

**Henner:** Agradeço imensamente a Deus, pelas incríveis oportunidades e me apoiar até mesmo nas horas mais difíceis. Ao lado, dedico este trabalho aos meus pais, Rosemary Tavares Lustosa Campos e Henner Soares Campos, por todo apoio e confiança desde sempre, todo apoio e carinho, além da grande amizade e companheirismo.

Agradeço minha namorada Jéssica Fagundes Ribeiro Fernandes, por sempre estar do meu lado nos momentos em que mais precisei.

Agradeço ainda nosso professor orientador Geyverson Teixeira de Paula, por sempre se apresentar disposto a ajudar e sempre com total paciência.

Aqui fica um reconhecimento diante de todos que me auxiliaram nesta longa caminhada direta ou indiretamente. Todos possuem parte desta vitória e ninguém, nunca, consegue alguma coisa sozinho. Obrigado.

## VII. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MORAIS, José. O que é ESP8266 – A Família ESP e o NodeMCU. 9 de junho de 2017. Portal Vida de Silício. Disponível em: <<https://portal.vidadesilicio.com.br/o-que-esp8266-nodemcu/>>. Acesso em 26/02/2019 15:39
- [2] ITEAD Main page Introduction. ITEAD. Disponível em <[https://www.itead.cc/wiki/Main\\_Page](https://www.itead.cc/wiki/Main_Page)>. Acesso em: 01/11/2019.
- [3] ABOUT US. Oasis: Open standarts. OpenSource., Disponível em: <<https://www.oasis-open.org/org>>.
- [4] SGABARDI, Júlio André. DOMÓTICA INTELIGENTE: AUTOMAÇÃO RESIDENCIAL BASEADA EM COMPORTAMENTO. Dissertação de Mestrado – Centro Universitário da FEI. São Bernardo do Campo, 2007
- [5] IFTTT: Conheça o aplicativo de inteligência artificial. USEMOBILE, 01 de novembro de 2018. Disponível em: <<https://usemobile.com.br/ifttt-conheca-o-aplicativo/>>.

Acesso em: 02/05/2019

[6] CURVELLO, André. ESP32 – Um grande aliado para o Maker IoT. 27 de fevereiro de 2018. FilipeFlop. Disponível em: <<https://www.filipeflop.com/blog/esp32-um-grande-aliado-para-o-maker-iot/>>. Acesso em 15/11/2019 as 19:03

[7] CURVELLO, André. Apresentando o módulo ESP8266. 29 de abril de 2015. Disponível em <<https://www.embarcados.com.br/modulo-esp8266/>>. Acesso em: 26/09/2019

[8] Getting Started with Blynk. Blynk.io. Disponível em: <<https://blynk.io/en/getting-started/>>. Acesso em: 15/01/2019

[9] de Lacerda Segundo, Ramon Temporim. Silva Moraes, Rodrigo da. Arduino e módulo ESP8266-Sistemas de Comunicação, novembro de 2017. Trabalho de conclusão de Curso – UniCEUB, Brasília, 2017.

[10] BERTOLLETI, Pedro. Controle e Monitoramento IoT com NodeMCU e MQTT. 30 de maio de 2016. FilipeFlop. Disponível em: <<https://www.filipeflop.com/blog/controle-monitoramento-iot-nodemcu-e-mqtt/>>. Acesso em 15/11/2019 as 17:32

[11] Sonoff Basic – WiFi Wireless Smart Switch por ITEAD. ITEAD. Disponível em <<https://www.itead.cc/sonoff-wifi-wireless-switch-1.html>>. Acesso em: 26/09/2019

[12] *Sonoff Overview: Home automation is "The Internet of Things"*. ITEAD. Disponível em <<https://www.itead.cc/wiki/Sonoff>>. Acesso em: 26/09/2019.

[13] *Sonoff temperature and humidity*. ITEAD. Disponível em <[https://www.itead.cc/wiki/Sonoff\\_TH\\_10/16](https://www.itead.cc/wiki/Sonoff_TH_10/16)>. Acesso em: 20/10/2019.

[14] *Sonoff PIR 2 Auto PIR Sensor Infrared IR Wireless Motion Detector Sensor*. ITEAD. Disponível em <<https://www.sonoff.in/%20Sonoff-PIR2>>. Acesso em 20/10/2019.

[15] *Sonoff RF Bridge 433 Overview*. ITEAD. Disponível em <[https://www.itead.cc/wiki/Sonoff\\_RF\\_Bridge\\_433](https://www.itead.cc/wiki/Sonoff_RF_Bridge_433)>. Acesso em: 01/11/2019.

[16] *Sonoff RF Bridge*. ITEAD. Disponível em <<https://www.itead.cc/sonoff-rf-bridge-433.html>>. Acesso em: 26/10/2019.

[17] SEDRA, S.; SMITH, K.. Microeletrônica. 4ª. Edição, Pearson Makron Books, São Paulo, Brasil, 2005.

[18] Minissensor PIR/ Sensor de Movimento para Arduino – HC-SR505. Portal UsinaInfo. Disponível em <<https://www.usinainfo.com.br/sensor-de-movimento-arduino/mini-sensor-pir-sensor-de-movimento-para-arduino-hc-sr505-4760.html>>. Acesso em: 26/10/2019.

[19] Enviando mensagens MQTT com módulo ESP32.Arduino e Cia, 17 de fevereiro de 2019 Disponível

em: <<https://www.arduinoecia.com.br/enviando-mensagens-mqtt-modulo-esp32-wifi/>>. Acesso em: 10/2019

[20] OLIVEIRA, Ricardo Rodrigues. USO DO MICROCONTROLADOR ESP8266 PARA AUTOMAÇÃO RESIDENCIAL, fevereiro de 2017. Trabalho de conclusão de Curso – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2017.

[21] JÚNIOR, Olair Ricardo. Sistema de monitoramento residencial baseado em Internet das Coisas. Trabalho de conclusão de Curso – Universidade Estadual de Londrina, Londrina, 2017.