

Assistente Pessoal Integrada com Multiagentes para Economia de Energia

¹Luis Guilherme Gonçalves da Silva

¹Murilo Ibrahim Bonomo

²Fábio Andrijauskas

mu.bonomo@gmail.com

¹Aluno do Curso de Engenharia da Computação, Universidade São Francisco; Campus Itatiba

²Professor Orientador Fábio Andrijauskas, Curso de Engenharia da Computação, Universidade São Francisco; Campus Itatiba.

Resumo.

Com o avanço do conceito do uso de dispositivos conhecido como Internet das Coisas (IoT) e pensando no desperdício de energia que ocorre nos aparelhos eletrônicos de uma residência, mesmo em modo standby, esse artigo propõe uma solução de economia, gerenciamento desses valores através de uma tomada inteligente com base nos conceitos de IoT e uma assistente pessoal auxiliando nos comandos possíveis que o usuário pode realizar. A Tomada Inteligente é um dispositivo que funciona como um interruptor para os equipamentos conectados a ela, que pode ser comandado através de um smartphone ou de uma assistente pessoal integrada a ela. Para o gerenciamento e obtenção de dados, usaremos um microcontrolador, ESP32, para a obtenção dos dados de tensão e corrente dos aparelhos eletrônicos, dois sensores, um de corrente e o outro de tensão. O protótipo desenvolvido para este projeto visa ser o mais barato possível, porém mantendo-se uma alta qualidade e desempenho. O usuário será capaz de acionar diferentes tipos de equipamentos através da assistente pessoal, visualizar os dados de tensão e corrente dos equipamentos através de uma planilha do Google.

Palavras-chave: IoT, energia, economia, tomadas, inteligentes, assistentes.

Abstract. *With the advancement of the concept and use of devices known as the Internet of Things (IoT) and thinking about the waste of energy that occurs in the electronic devices of a home, even in standby mode, this article proposes a solution for saving and managing these values through a smart socket based on IoT concepts and a personal assistant assisting in the possible commands that the user can perform. The Smart Socket is a device that works as a switch for the equipment connected to it, which can be controlled via a smartphone or a personal assistant integrated with it. For data management and retrieval, we will use a microcontroller, ESP32, and for obtaining voltage and current data from electronic devices, two sensors, one for current and the other for voltage. The prototype developed for this project aims to be as cheap as possible while maintaining high quality and performance. The user will be able to activate different types of equipment through the personal assistant and view the voltage and current data of the equipment through a Google spreadsheet.*

Keywords: *IoT, energy, economy, outlets, smart, assistants.*

1. Introdução

Apesar do grande crescimento de energias renováveis, o crescimento econômico do país exige cada vez mais a demanda por energia elétrica e recentemente o Brasil tem sofrido com um desperdício enorme desse recurso, que vai desde a produção até o consumo. De acordo com um estudo da ABESCO (Associação Brasileira das Empresas de Serviços de Conservação de Energia), esse desperdício custou mais de 60,0 bilhões de reais para o país, que poderiam ter sido aplicados em outros setores. Parte desse desperdício ocorre no setor residencial e comercial, que muitas vezes negligencia atitudes básicas de conservação de energia. Em vista disso, pensar e elaborar formas do uso eficiente de energia elétrica para esses setores é essencial para diminuir esse desperdício, além de economizar dinheiro, recursos e colaborar com o meio ambiente.

Com o crescimento cada vez maior das tecnologias, principalmente das que conectam coisas do seu cotidiano com a internet, mais conhecida como IoT (Internet das Coisas), é possível automatizar objetos a uma comunicação em rede e desenvolver melhorias para o seu funcionamento, podendo ser controlados remotamente ou então, sem nenhuma interação humana. Com isso, as possíveis aplicações para essa tecnologia são

quase infinitas, muitas já podem ser vistas no nosso cotidiano e em alguns setores de produção.

Atualmente o sistema de medição de energia das empresas distribuidoras de energia consiste em um sistema de monitoramento mensal, desta forma, os consumidores possuem o conhecimento de quanto gastaram e de quanto irão pagar de energia somente a cada 30 dias. Para uma melhor forma de controle e monitoramento desses gastos, seria ideal que os consumidores tivessem a disponibilidade de monitorar esses dados em tempo real diariamente, para que pudessem responder de forma mais rápida e precisa. Pensando nesse problema, como reduzir gastos com a energia elétrica utilizando a tecnologia?

Com o avanço das tecnologias que facilitam o cotidiano das pessoas, Tomadas Inteligentes (*Smart Power Plugs*) integram qualquer equipamento eletrônico de uma residência a uma rede de comunicação, permitindo controlar esses eletrônicos remotamente através de aplicativos, web ou até mesmo por assistentes pessoais, como Alexa e Google Home. Uma vez feita essa integração todos os dados podem ser monitorados em tempo real e analisados para determinar onde está havendo maior consumo de energia e com isso determinar soluções para a redução desse gasto. Além disso, integrando-se com assistentes pessoais é possível fazer como que elas aprendam com os padrões e realize ações imediatas de economia.

Em um mundo que as informações podem ser analisadas em tempo real e que podem ser entregues quase que instantaneamente, é difícil considerar que ainda não existam, pelo menos consideravelmente, aplicações que fornecem esse tipo de trabalho para setores que seriam extremamente importantes terem esse tipo de informação como por exemplo, prestadoras de energia, e dessa forma deixar os consumidores dependentes dessas informações dos prestadores. A Internet das Coisas possui um grande potencial para resolver esse problema de forma que seja benéfica para os dois lados. Nesse contexto, a proposta deste trabalho é prototipar uma tomada inteligente mais barata que as presentes atualmente no mercado, mas que também atenda a todas as necessidades dos consumidores.

Para o desenvolvimento deste trabalho foram utilizadas pesquisas bibliográficas, estudo de casos e prototipação. Para a realização da pesquisa bibliográfica foram utilizadas publicações científicas da área de tecnologia da informação. O Estudo de caso foi desenvolvido através da aplicação do protótipo em uma residência. A prototipação foi

desenvolvida através de trabalhos semelhantes, porém mantendo-se a ideia de criar um protótipo mais barato e viável.

Esse trabalho se baseia em três etapas principais, sendo a primeira, definições de Assistentes pessoais e IoT explicando cada funcionalidade e características de tais tecnologias. A segunda etapa caracteriza a prototipagem do produto e suas principais funcionalidades e funcionamento do mesmo. A terceira etapa apresentará os resultados do produto no cenário de estudo com os resultados esperados e obtidos.

2. Referencial Teórico

Para o desenvolvimento do projeto será utilizado conceitos e tecnologias, como IoT(Internet das Coisas), microcontroladores, assistentes pessoais e tomadas inteligentes. A tomada inteligente será o dispositivo IoT que contará com um microcontrolador responsável por gerenciar os dados e estabelecer uma conexão do dispositivo com a rede, possibilitando a integração da assistente pessoal, assim, será possível realizar ações por meio de comandos de voz e facilitar a interação do usuário com os seus equipamentos.

2.1 Internet das Coisas (IoT)

A Internet das Coisas (do inglês Internet of Things (IoT)) é uma tecnologia recente que vem crescendo cada dia mais como uma maneira de se criar objetos do dia a dia com conexão à rede de internet para monitorar dados e facilitar seu uso. (ASHTON, 2009). O termo foi utilizado pela primeira vez em 1999 com Kevin Ashton apresentando uma nova ideia de RFID para a empresa Protector & Gamble (P&G).

A tecnologia IoT não possui somente uma ideia de definição, ela pode ser trabalhada de diferentes formas dependendo da interpretação em que ela está sendo tratada, em vista disso, algumas definições surgiram com o tempo (WAKA, 2020), são elas:

1. Visão orientada às coisas: Nessa visão a interpretação de IoT baseia-se nos objetos, ou seja, a integração das “coisas” numa mesma estrutura/rede.
2. Visão orientada à Internet: Nessa visão temos que a interpretação baseia-se na comunicação, na estrutura da rede e seus protocolos necessários para que se haja a integração e comunicação com o ambiente.

3. Visão orientada à semântica: Diferente das duas anteriores visões, em que tinham seu foco relacionado com termos do nome IoT, essa visão baseia-se na interpretação dos dados que são gerados devido a essa tecnologia, ou seja, trata sobre a busca, o armazenamento, a organização e a representação dos dados. (WANGHAM; DOMENECH; MELLO, 2013).

“Com a Internet das Coisas, é possível realizar uma série de tarefas até então impensáveis, como controlar objetos remotamente ou utilizá-los como provedores de serviços. Por isso, essa tecnologia trouxe grandes avanços não só para o comércio ou para a indústria, mas também para o âmbito doméstico.

Além disso, a Internet das Coisas traz, ainda, outra vantagem: ela aumenta a eficiência dos objetos e até mesmo possibilita que eles desempenhem mais funções.” (CRAZE, 2020).

A arquitetura da IoT, de forma geral, apresenta cinco camadas, são elas: camada de borda, camada de gateway de acesso, camada de Internet, camada de middleware e camada de aplicação. A camada de borda trata da obtenção de dados, do processamento das informações e do apoio a comunicação, através de sistemas embarcados, etiquetas RFID e sensores em geral. A camada de gateway de acesso cuida do encaminhamento dos dados. A camada de Internet trata sobre a comunicação, sobre a rede e sobre o processamento da grande quantidade de dados obtidos. A camada de middleware trabalha entre a camada de Internet e a camada de Aplicação e ela é responsável por filtrar as informações obtidas. E a camada de aplicação efetua o processamento dos dados obtidos sobre a camada de borda e efetua serviços ao usuário. (WAKA, 2020.)

Juntamente com o avanço dos IoTs os microcontroladores, que surgiram na década de 70 ganharam um papel extremamente importante como ferramenta para criação de novas tecnologias. Basicamente, é um pequeno computador em um circuito integrado que contém um processador, memória e periféricos de entrada e saída programáveis. Para esse projeto foi utilizado o microcontrolador ESP 32, além de ser um microcontrolador de baixa energia, alto processamento já possui, integrado ao seu hardware funcionalidades como bluetooth e wi-fi tornando-se mais viável e agradável trabalhar com esse microcontrolador.

Através de uma conexão à uma rede wi-fi é possível fazer o microcontrolador, juntamente conectado em alguns sensores enviar os dados coletados para algum reservatório para serem monitorados, analisados e manipulados. Essas análises e tomadas de decisão podem ser feitas com a ajuda de uma Inteligência Artificial.

2.2 Assistentes Pessoais

“A Inteligência Artificial (IA) ou artificial intelligence (AI) está enraizada na história da humanidade, seja ela por meios reais ou na ficção das telas do cinema. Além dos robôs e máquinas, que são mais fáceis de associar, pois são palpáveis, a IA está presente em objetos que vão desde uma smart TV até assistentes pessoais que auxiliam nas tarefas do dia a dia, como os presentes smartphones.” (JONCO; SILVEIRA, 2015).

Além da análise dos dados feita pela IA, o acesso a internet permite controle do hardware ao microcontrolador, como por exemplo, ligar e desligar. Esse controle pode ser feito por um smartphone, computador ou outro dispositivo com acesso à mesma rede wi-fi. Porém existe um meio mais prático para realizar esse controle e com menos intervenção humana que é por meio das assistentes pessoais.

Essas assistentes pessoais são equipamentos eletrônicos dotados de Inteligência Artificial capazes de auxiliar as pessoas com relação as suas tarefas do dia a dia, sejam elas, lembrar sobre uma reunião ou marcar compromissos até algo mais sofisticado como pedir para acender uma luz da casa, ligar ou desligar algum equipamento, entre muitas outras funcionalidades, tudo através de comando de voz ou aplicativo.

No mercado atual já existem algumas assistentes pessoais, sendo as mais famosas a Alexa, da Amazon, o Google Assistente, da Google e a Siri, da Apple. Devido sua maior compatibilidade com aplicativos e serviços foi estabelecido que, a Alexa, teria uma vantagem e facilidade maior de integração com a tomada inteligente, proposta nesse projeto.

Com o grande desenvolvimento da tecnologia, principalmente da Internet das Coisas, diversos dispositivos eletrônicos comuns do nosso dia a dia ganharam melhorias e ou ganhado maior notoriedade na sociedade atual. Apesar das Tomadas Inteligentes (ou Smart Plugs em inglês), ainda não possuem grande destaque nas notícias de tecnologia, sua funcionalidade pode ser extremamente útil e vantajosa. Sendo dispositivos capazes de integrar objetos do seu cotidiano à uma rede de internet, proporcionando dessa maneira com que eletrodomésticos, luzes ou qualquer equipamento ligado na tomada, possa ser controlado a distância ou por voz, através de aplicativos e assistentes pessoais.

Atualmente, uma grande variedade de tomadas inteligentes estão presentes no mercado, com diferentes funcionalidade e de diferentes formas. Podendo ser do tipo de proteção contra vazamentos, do tipo remota ou do tipo de monitoramento. As tomadas

inteligentes são desenvolvidas através de um microcontrolador integrado internamente capazes de controlar algumas situações rotineiras das pessoas. (WANG; PENG; ZHANG, 2015).

3. Metodologia

O desenvolvimento do projeto será baseado no estudo de como reduzir o consumo de energia elétrica e conseqüentemente economizar gastos futuros utilizando as tecnologias de IoT integradas com assistentes pessoais. Através desse estudo foi desenvolvido um protótipo de uma régua de tomada inteligente onde é possível acompanhar os gastos em um dispositivo móvel e controlar essa tomada por meio de uma assistente pessoal.

É possível dividir essa frente em três partes, sendo elas: O hardware, que irá contar com o desenvolvimento físico do protótipo. O software, que será responsável por gerenciar e apresentar os dados obtidos do dispositivo para o usuário, além de permitir que a assistente pessoal desenvolva melhores formas de interação. E os dados, que serão os valores obtidos pelo dispositivo IoT para que seja feita a análise dos valores referentes à tensão e corrente, e conseqüentemente os valores de potência, consumo dos aparelhos e gastos da residência.

3.1 Hardware

Foi preciso estabelecer os componentes que seriam preciso para a execução do projeto, levando em consideração que o projeto tem como finalidade ser de baixo custo, os componentes necessários estabelecidos são:

- Microcontrolador Esp32 DevkitV1;
- Sensor de tensão ZMPT101B MWS bivolt 2mA;
- Sensor de corrente ACS712 Montimport 30A;
- Módulo Relé Bestep JQC3F-03VDC-C.

O ESP32 é um microcontrolador de baixo custo e baixo consumo de energia, criado com a capacidade de estabelecer comunicações sem fio através de Wi-fi e Bluetooth, e com uma alta capacidade de processamento. Tendo um tamanho bastante compacto e mantendo uma grande eficiência foi que esse componente se tornou bastante atrativo para

compor no desenvolvimento do projeto. Como foi apresentado no capítulo 2, esse componente possui a função de coordenar os dados, estabelecendo uma rede para que os equipamentos conectados na tomada tenham essa integração com a internet e junto com os sensores de tensão e corrente, obter os dados da tensão e da corrente da rede elétrica da residência.

Para medir os valores de tensão foi escolhido o sensor ZMPT101B, com ele é possível identificar a presença de tensão alternada. Uma de suas características que fizeram com que ele fosse empregado neste projeto, além de possuir diversas aplicações e um bom preço, foi a de possuir uma grande capacidade de medição. Capaz de medir o valor da tensão, ele consegue fornecer ótimas leituras desses valores. De acordo com o datasheet fornecido pelo fabricante ele consegue medir tensões entre 0V e 250V, num ambiente com temperatura de -40°C à aproximadamente 70°C . O sensor de tensão conectado ao ESP32 será responsável por coletar os dados da tensão e transmiti-los ao microcontrolador, que irá obter esses dados e encaminhará para o banco de dados.

Foi escolhido para medir o valor de corrente o sensor ACS712, por ser de baixo custo e por ser o mais encontrado no mercado. Ele utiliza o efeito hall, que é basicamente um efeito que surge quando há uma diferença de potencial num condutor elétrico gerando um campo magnético perpendicular à corrente. De acordo com o datasheet do fabricante é possível medir correntes entre -30 e +30A e obter uma tensão de saída no pino OUT de 66 até 185 mV/A, com um erro de 1,5% funcionando num ambiente com temperatura de 25°C . Assim como no sensor de tensão, o sensor de corrente estará conectado junto ao microcontrolador e passará os dados referente a corrente, que posteriormente serão enviados ao banco de dados.

Para controlar a comutação liga/desliga de um circuito foi utilizado um módulo Relé. Devido uma corrente elétrica passando pelas espiras da bobina do relé, cria-se um campo magnético que irá atrair uma alavanca responsável pela alteração dos estados do circuito. Com ele será possível ligar ou desligar certos equipamentos eletrônicos da residência através de um comando de voz com a Alexa.

A Alexa nesse projeto será um equipamento para que se tenha uma maior abrangência de interação com os equipamentos através do IoT. Desta forma, a Alexa, capaz de interagir por meio de comandos por voz e realizar tarefas do seu cotidiano, como citado no referencial teórico, ela será uma intermediária entre os usuários e o IoT. Para a realização deste projeto utilizamos a Alexa Echo Dot, por ser a mais barata em comparação

com outros modelos, além de podermos realizar comandos por voz e ela estar sempre aprendendo e adicionando novas *skills*, além de possuir compatibilidade com dispositivos Android e iOS e uma integração mais fácil de se realizar.

3.2 Software

Para o desenvolvimento deste projeto foi preciso determinar os componentes (Hardware) que seriam utilizados para o desenvolvimento do mesmo, assim como foi preciso estabelecer os softwares que seriam preciso para a execução do projeto. Nesta seção iremos discutir os softwares empregados, que são:

- MQTT
- Alexa Skill Kit(ASK);
- Sinric Pro;

Com grande quantidade de dados que se pode obter com as leituras de tensão e corrente, e principalmente do local onde se está fazendo essas leituras, um banco de dados relacional exigiria uma enorme capacidade de infraestrutura, uma vez que, quanto maior o número de dados, maior é a necessidade de servidores para atender essa aplicação.

O MQTT, sigla para Message Queuing Telemetry Transporte, é um protocolo de mensagens entre máquinas, destinado principalmente para realizar a conexão entre dispositivos através de uma quantidade baixa de largura de banda, por esse motivo, para que seja realizada a conexão entre o dispositivo IoT e o banco de dados não relacional de uma forma mais rápida e eficiente, o MQTT é uma das tecnologias que mais tem ganhado destaque na área de automação.

A Alexa Skill Kit é um conjunto de API's e de ferramentas para desenvolvedores. Essa ferramenta permite criar interações de voz com a Alexa sem ter conhecimento em linguagem natural de voz. É possível também criar diversas *skills* (interação por voz com a Alexa) em diversos idiomas compreendidos pela Alexa.

Para o projeto foi utilizada a skill Sinric Pro. Com essa skill é possível fazer com que a Alexa descubra seu IoT e envie comando como se fosse um dispositivo inteligente, sendo possível configurar os comando que serão executado. O comando configurado nesse projeto foi o comando de Switch (troca), nesse caso o comando será de ON/OFF (Liga/Desliga).

3.3 Dados

O IoT, através dos sensores, irá acompanhar os dados de tensão (em Volts), e corrente (em Amperes) dos equipamentos ligados nele, sendo possível calcular a potência (em Watt) que o equipamento está dissipando. Ao obter esse valor pode se calcular o consumo deste aparelho (em Quilowatt hora) por mês. Com o consumo mensal do equipamento consegue realizar o cálculo do gasto (em reais) com os equipamentos conectados. O IoT calcula o valor médio da tarifa observado em junho/2020 segundo a ANEEL (Agência Nacional de Energia Elétrica) no valor de R\$ 0,560 kWh

Equação 1 - Cálculo da Potência

$$Potência = Tensão \times Corrente$$

Equação 2 - Cálculo do Consumo de um Aparelho Mensal

$$Consumo\ do\ Aparelho = Potência \times Quantidade\ de\ Dias \times Quantidade\ de\ Horas$$

Equação 3 - Cálculo do Gasto Mensal

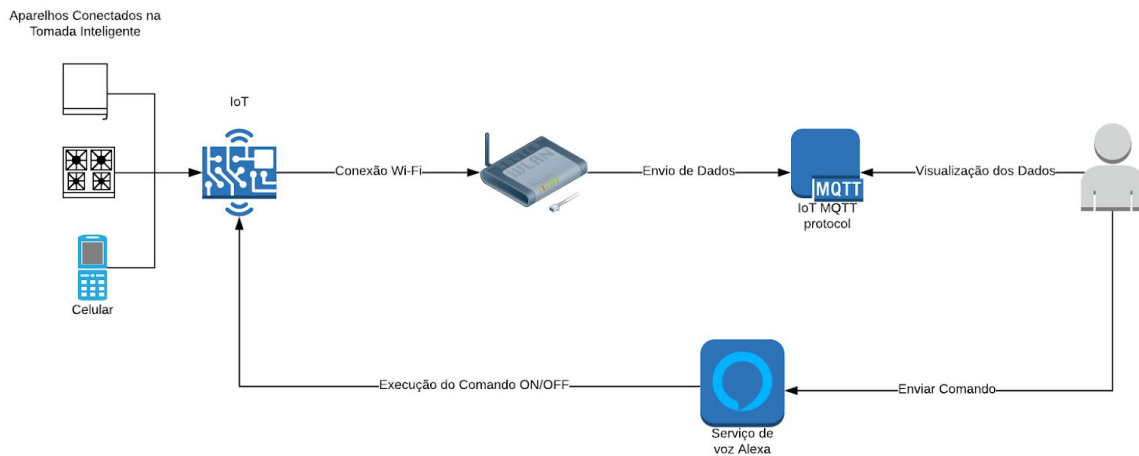
$$Gasto = Consumo\ do\ Aparelho \times Tarifa$$

4. Resultados

A arquitetura do sistema funcionará segundo a Figura 1, o usuário, através de um comando por voz envia os comando de liga/desliga para a assistente pessoal e monitora o consumo de energia elétrica através de uma página HTML. Essa interface comunica-se com o aplicativo MQTT Dash, que fica responsável por receber os dados de consumo. A comunicação entre os dispositivos conectados na tomada e a rede é feita através de um microcontrolador (ESP32).

Figura 1 - Fluxograma dos Dados: O IoT (protótipo desenvolvido) receberá os dados de aparelhos como por exemplo, carregador de celular, eletrodomésticos e qualquer outro tipo de equipamento. Esses dados recebidos serão enviados ao a tela de um aplicativo (MQTT DASH) via protocolo MQTT. O aplicativo será a

parte visual apresentada ao usuário sendo possível visualizar informações da tensão, corrente e potência do equipamento conectado ao IoT.

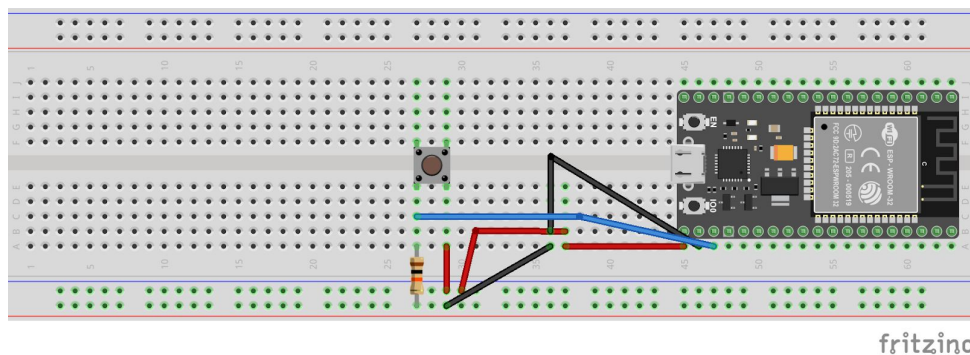


(Fonte: Autores Próprios)

4.1 Hardware

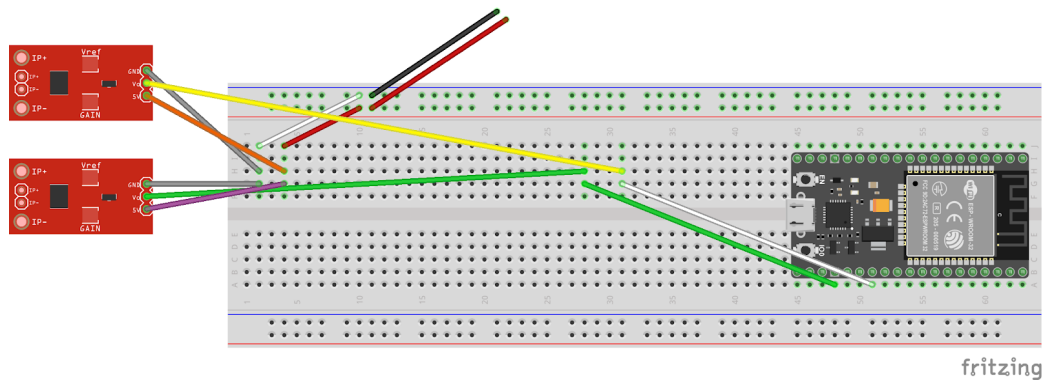
O ESP32 terá conectado um sensor de tensão que medirá a entrada da fonte (em Volts) e um sensor de corrente que medirá a corrente entre o equipamento e a fonte (em Amperes), ambos sensores ligado em 5V do ESP32 (Figura 3). O circuito base contará também com um relé, responsável pelo sistema de liga/desliga que será acionado via comando de voz pela Alexa (Figura 4 e 5) e um botão Push Button que servirá como reset do equipamento para o modo de Access Point (Figura 2).

Figura 2 - Circuito utilizado para o Push Button: O Push Button servirá como reset do microcontrolador ESP32 do modo Access Point para End Point para ligar o botão foi necessário um resistor de 10kΩ.



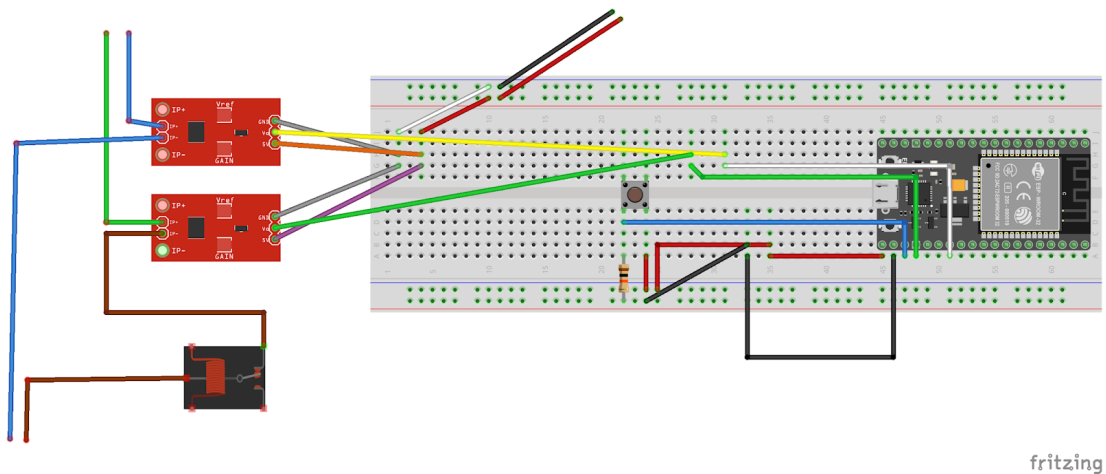
(Fonte: Autores Próprios)

Figura 3 - Circuito utilizado para os sensores: O ESP32 fornece apenas 3,3V e os sensores necessitam de uma alimentação de 5V, com isso, foi necessário acrescentar um fonte de 5V para alimentar os sensores.



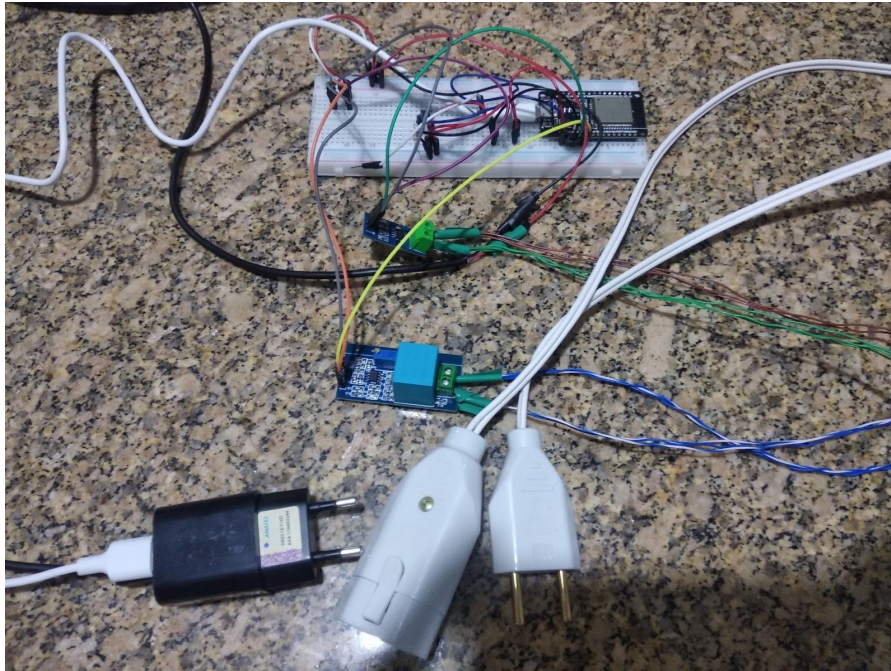
(Fonte: Autores Próprios)

Figura 4 - Circuito completo: É a junção da Figura 2 com a Figura 3 representando o circuito inteiro do protótipo.



(Fonte: Autores Próprios)

Figura 5 - Circuito completo: Protótipo real finalizado.



(Fonte: Autores Próprios)

Com o Hardware montado existe outra frente para o desenvolvimento do projeto, a parte que faz o hardware executar os comandos e coletar os dados necessário para o funcionamento completo do protótipo, o Software

4.2 Software

Para realizar a conexão Wi-Fi é necessário utilizar a biblioteca wifimanager para gerenciamento de redes com acesso na internet para primeira conexão, essa biblioteca faz com que o ESP32 se torne um *Access Point* (AP) (Figura 6), permitindo assim que outros dispositivos se conectem no microcontrolador. Ao se conectar na rede criada pela biblioteca (nesse caso a ESP_AP) é possível selecionar uma rede para conectar o IoT, deixando de ser um AP.

Essa biblioteca trabalhar em conjunto com outras duas bibliotecas padrões do ESP32, como por exemplo a DNSServer para o ESP32 realizar as requisições de conexão para a rede na qual o microcontrolador irá se conectar, *WebServer* para assim que se conectarem no AP criado iniciar um *WebServer* no qual será possível ser feita a interação de conexão com o usuário e a Wi-Fi que é necessário para realizar qualquer tipo de conexão Wi-Fi (Figura 7).

Figura 6 - Include da Biblioteca de Gerenciador de Wi-Fi. Na linha 2 a biblioteca de Wi-fi para realizar a conexão, na linha 4 a biblioteca responsável para acessar o domínio DNS, na linha 6 a biblioteca de WebServer que permite enviar os dados coletados à um WebServer local e a biblioteca de WiFiManager na linha 6 responsável pelo gerenciamento das redes conectáveis.

```
//Biblioteca padrão para WiFi
#include <WiFi.h>
//Biblioteca de DNS para requisições
#include <DNSServer.h>
//Biblioteca WebServer para configurar o tela HTML
#include <WebServer.h>
//Biblioteca WiFi Manager para gerenciamento das Conexões do ESP32
#include <WiFiManager.h>
```

(Fonte: Autores Próprios)

Figura 7 - Configuração da Biblioteca: Principais funções para configurar o Wi-Fi do ESP32, a função setAPCallback() define em qual modo de uso o ESP32 estará (*Access Point* ou *End Point*).

```
//declaração do objeto wifiManager
WiFiManager wifiManager;
//callback para quando entra em modo de configuração AP
wifiManager.setAPCallback(configModeCallback);
//callback para quando se conecta em uma rede, ou seja, quando passa a trabalhar em modo estação
wifiManager.setSaveConfigCallback(saveConfigCallback);
//cria uma rede com esse nome e essa senha para conexão
wifiManager.autoConnect("ESP_AP", "12345678");
```

(Fonte: Autores Próprios)

Com os sensores conectados no ESP32 é possível usar a biblioteca emonlib que é responsável pelo monitoramento dos componentes eletrônicos (Figura 8). Após a configuração das bibliotecas pode-se enviar os dados para uma planilha do google (Figura 9) ou para um aplicativo via protocolo MQTT (Figura 10).

Figura 8 - Código para os sensores de tensão e corrente: A biblioteca para ambos os sensores é a EmonLib.h (Linha 1). As definições são para calibração dos sensores, esses valores são definidos com auxílio de um multímetro. Na linha 5 foi necessário criar uma instância da biblioteca (emonl). Os cálculos são definidos na linha 13 com o emonl.calcVI (17 semiciclos, tempo limite para fazer a medição). A partir da linha 18 o ESP32 mostrará no monitor os resultados dos sensores.

```
1 #include "EmonLib.h"
2 #define VOLT_CAL 105.8
3 #define CURRENT_CAL 10
4
5 EnergyMonitor emon1;
6
7 void setup(){
8   Serial.begin(9600);
9   emon1.voltage(4, VOLT_CAL, 1.7);
10  emon1.current(15, CURRENT_CAL);
11 }
12 void loop(){
13   emon1.calcVI(17,2000);
14   float corrente = emon1.Irms;
15   float tensao = emon1.Vrms;
16   float potencia = corrente * tensao;
17
18   Serial.print("Corrente: ");
19   Serial.print(corrente);
20   Serial.println("A");
21
22   Serial.print("Tensão: ");
23   Serial.print(tensao);
24   Serial.println("V");
25
26   Serial.print("Potencia: ");
27   Serial.print(potencia);
28   Serial.println("W");
29
30   delay(1000);
```

(Fonte: Autores Próprios)

Figura 9 - Código para o ESP32 enviar os dados coletados pelos sensores e preencher o formulário do Google: As primeiras linhas servem para incluir as bibliotecas de wi-fi e na linha 3 Cria uma conexão segura de client. É necessário definir uma *string* com o *link* do formulário, linha 5 (String textFix = "GET /forms/d/e/1FAIpQLSdR_4pPVAzD2vJOvNDTEjvUttGYBsRrInFI4MVQeCj-YZ7cLQ/formResponse?ifq"); e após isso definir os dados para envio ao formulário, linha 16 String toSend = textFix+"&entry.157353091="+corrente+"&entry.600866828="+tensao+"&entry.179559370="+potencia; no qual o primeiro entry representa o campo do formulário que recebe a corrente, o segundo entry representa o campo do formulário que recebe a tensão e o terceiro representa o campo calculado da potência.

```
1 #include <WiFi.h>
2 #include <WiFiClientSecure.h>
3
4 WiFiClientSecure client;
5 String textFix = "GET /forms/d/e/1FAIpQLSdR_4pFVAzD2vJOvNDTEjVuttGYBsRrInFl4MVQeCj-YZ7cLQ/formResponse?ifq";
6
7 void setup(){
8   Serial.begin(9600);
9   WiFi.mode(WIFI_MODE_APSTA);
10  WiFi.begin("NOME_DO_WIF", "SENHA");
11 }
12 void loop(){
13
14   if (client.connect("docs.google.com", 443) == 1)
15   {
16     String toSend = textFix+"&entry.157353091="+corrente+"&entry.600866828="+tensao+"&entry.179559370="+potencia;
17
18     toSend += "%submit=Submit HTTP/1.1";
19
20     client.println(toSend);
21     client.println("Host: docs.google.com");
22     client.println();
23     client.stop();
24     Serial.println("Dados enviados.");
25   }
26   else
27   {
28     Serial.println("Erro ao se conectar");
29   }
30 }
```

(Fonte: Autores Próprios)

Figura 10 - Código para conexão de dispositivos através do protocolo MQTT-Broker, na linha 1 é incluída a biblioteca que cria o client MQTT, na linha 4 e 7 é configurado os campos inseridos no aplicativo. As linhas 20 e 21 são utilizadas para definir o servidor e chamar a função caso não encontre o mesmo definido na linha 29. Na linha 44 é colocado a função da biblioteca responsável por enviar os dados do ESP32 ao aplicativo MQTT Dash


```
#include <PubSubClient.h>

/* Defines do MQTT */
#define topico_teste      "text01"

#define ID_MQTT  "esp32_mqtt"    //id mqtt (para identificação de sessão)
                                //IMPORTANTE: este deve ser único no broker (ou seja,
                                //           se um client MQTT tentar entrar com o mesmo
                                //           id de outro já conectado ao broker, o broker
                                //           irá fechar a conexão de um deles).

/* Função: inicializa parâmetros de conexão MQTT (endereço do
 *         broker, porta e seta função de callback)
 * Parâmetros: nenhum
 * Retorno: nenhum
 */
void initMQTT(void)
{
    MQTT.setServer(BROKER_MQTT, BROKER_PORT);
    MQTT.setCallback(mqtt_callback);
}

/* Loop principal */
void loop()
{
    char dados[10] = {0};

    /* garante funcionamento das conexões WiFi e ao broker MQTT */
    VerificaConexoesWiFiMQTT();

    /* Envia as strings ao dashboard MQTT */
    int i = 0;
    while (Serial.available() > 0)
    {
        dados[i] = Serial.read();
        i++;
    }

    if (i != 0)
    {
        Serial.print(dados[0]);
    }

    MQTT.publish(topico_teste, dados);

    /* keep-alive da comunicação com broker MQTT */
    MQTT.loop();

    /* Refaz o ciclo após 2 segundos */
    delay(2000);
}

```

(Fonte: Autores Próprios)

Figura 11 - Código para interação com a Alexa: Para enviar comandos de voz para a Alexa utilizamos a biblioteca SinricPro.h" e SinricProSwitch.h. A Sinric Pro é uma plataforma Skill Kit da Alexa que permite cadastrar 5 dispositivos IOTs gratuitamente. Após cadastrar o dispositivo é necessário incluir a chave de acesso e as credenciais do Sinric Pro no código (Linha 7, 8 e 10). Com essas informações é configurado uma rede Wi-fi (Linha 15 e 16) e coletado as informações do *device* e o estado dele (Linha 17).

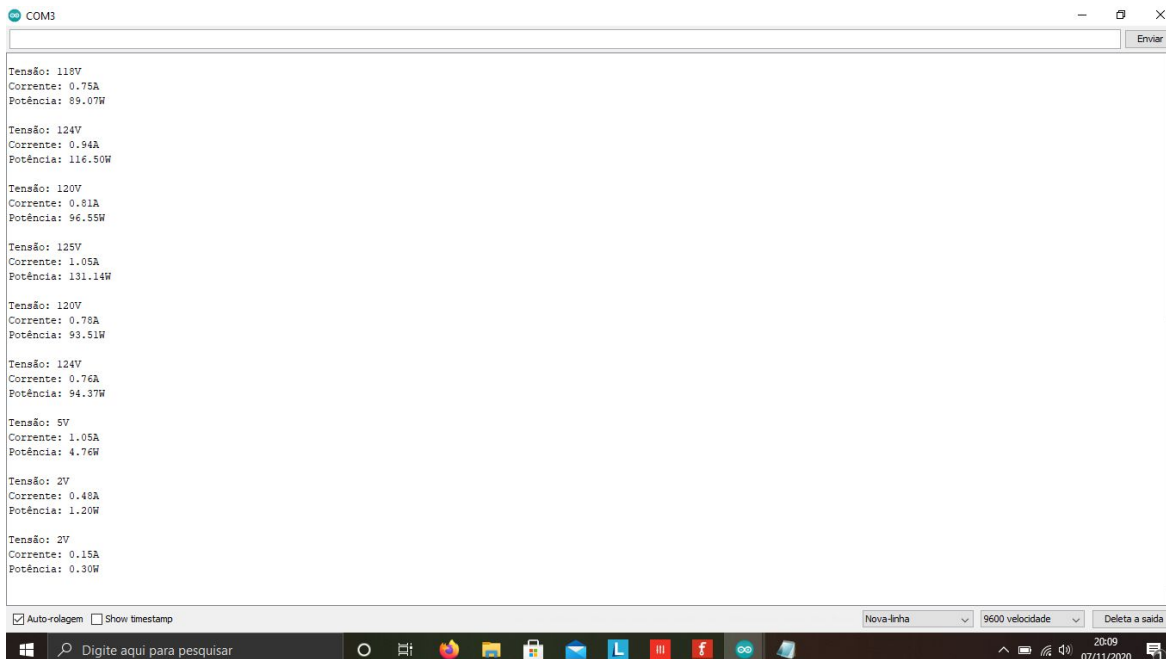
```
7 #define APP_KEY "0b0e4f05-c150-439e-bd79-422a3df618e8"
8 #define APP_SECRET "80f55e8e-f899-4862-bd7f-e35bbb751b8d-9d72b0aa-d6e3-48fa-91ae-2a8ee8778a0a"
9
10 #define LED_ID "5fb001d0cef657243fb0b80a"
11 #define LED_Pin 4
12
13 #define BAUD_RATE 9600
14
15 void setupWiFi();
16 void setupSinricPro();
17 bool LEDState(const String &deviceId, bool &state);
18
19 void setup() {
20     Serial.begin(BAUD_RATE); Serial.printf("\r\n\r\n");
21     setupWiFi();
22     setupSinricPro();
23     pinMode(LED_Pin, OUTPUT);
24 }
25
26 void loop() {
27     SinricPro.handle();
28 }
29
30 bool LEDState(const String &deviceId, bool &state) {
31     Serial.printf("O LED foi %s\r\n", state?"ligado":"desligado");
32     digitalWrite(LED_Pin, state);
33     return true;
34 }
```

(Fonte: Autores Próprios)

4.3 Dados

Com o ESP32 coletando os dados (Figura 12) e estando conectado em uma rede de internet wi-fi é possível enviar os dados coletados pelos sensores de tensão e corrente para uma planilha do google através de um formulário (Figura 13).

Figura 12 - Monitor serial exibindo os resultados obtidos pelos sensores com a tomada ligada e desligada respectivamente. O equipamento utilizado para essa medição foi um carregador de Notebook ligado em 110V e nas especificações 1A.



```
COM3
Enviar
Tensão: 118V
Corrente: 0.75A
Potência: 89.07W

Tensão: 124V
Corrente: 0.94A
Potência: 116.50W

Tensão: 120V
Corrente: 0.81A
Potência: 96.55W

Tensão: 125V
Corrente: 1.05A
Potência: 131.14W

Tensão: 120V
Corrente: 0.78A
Potência: 93.51W

Tensão: 124V
Corrente: 0.76A
Potência: 94.37W

Tensão: 5V
Corrente: 1.05A
Potência: 4.76W

Tensão: 2V
Corrente: 0.48A
Potência: 1.20W

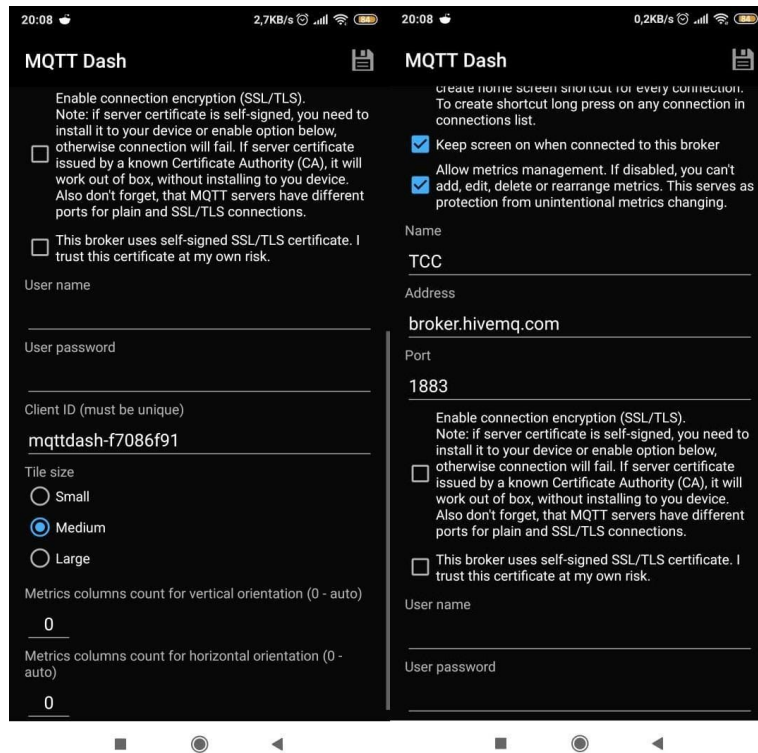
Tensão: 2V
Corrente: 0.15A
Potência: 0.30W

 Auto-rolagem  Show timestamp
Nova-linha 9600 velocidade Deleta a saída
20:09 07/11/2020
```

(Fonte: Autores Próprios)

Após os dados serem enviados à planilha do Google é necessário realizar a configuração do aplicativo MQTT Dash (Figura 13), em que na primeira tela as configurações são padrões e por esse motivo não precisam ser alteradas, somente user name/password são opcionais. Na segunda tela, ele solicita um nome para o broker, o endereço do broker (existem diversos MQTT Brokers públicos que podem ser utilizados, para a realização desse projeto foi utilizado o HiveMQ, que utiliza o endereço visto na imagem acima.) e por fim a porta(todos irão funcionar na mesma porta, no caso 1883).

Figura 13 - Configuração do MQTT Dash.

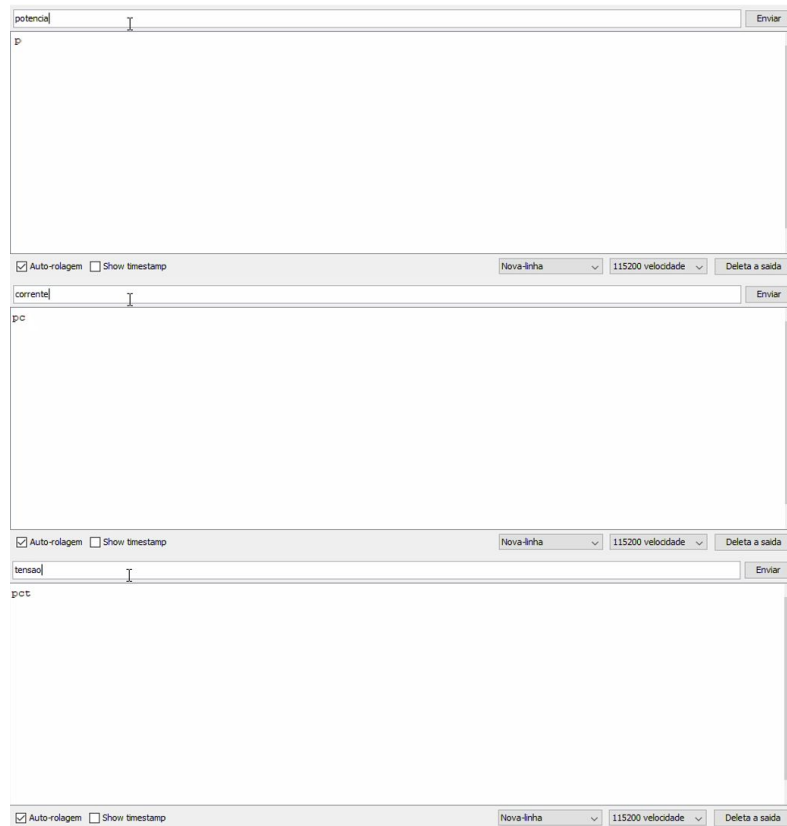


(Fonte: Autores Próprios)

5. Testes

Para fazer os testes foi utilizado o próprio serial do Arduino IDE para realizar o *input* dos dados, simulando basicamente o que o ESP32 faria enviando os dados relacionados à tensão, corrente e potência dos aparelhos (Figura 14).

Figura 14 - Monitor serial recebendo dados para encaminhar ao aplicativo MQTT Dash.



(Fonte: Autores Próprios)

O acionamento dos equipamentos via Alexa foi testado em cargas indutivas, na qual o campo magnético é gerado através de bobinas (Motor DC 6V, carregador de celular com entrada bivolt e saída de 2,1A, 5Vcc e potência 10,5W e ventilador bivolt 60W) e capacitiva, onde o campo magnético é gerado através de condensadores (lâmpada 12W). A resposta de acionamento a partir do comando até a sua execução nesses diversos testes foi a seguinte:

Tabela 1 - Comparação de diversos equipamentos no comando de voz *ON/OFF* (Liga/Desliga).

Equipamento	On	Off
LED (sem relé)	2,30s	1,95s
Motor DC	1,75s	1,57s
Carregador de Celular	1,78s	1,56s
Ventilador	2,28s	1,85s

Lâmpada	1,33s	1,86s
---------	-------	-------

(Fonte: Autores Próprios)

Os resultados mostram que o acionamento, média de 1,89s nos testes, é mais demorado que o desligamento, média de 1,76s, pela coleta desses dados é possível calcular o desvio padrão (dp) e observar que o acionamento (dp=0,41) é menos homogêneo que o desligamento (dp=0,18). Essas medições podem ser influenciadas por latência na rede, tempo de resposta da Alexa e influência do hardware (tanto a própria Alexa como o dispositivo móvel que o serviço da Alexa está instalado).

6. Discussão

O intuito do funcionamento do protótipo era com que ele funcionasse em conjunto com a assistente pessoal, Alexa e conseguisse realizar o acionamento dos dispositivos, bem como o monitoramento dos dados relacionado à tensão e corrente. Devido a uma incompatibilidade entre as bibliotecas que realizam as operações matemáticas para a leitura dos dados de tensão e corrente e a biblioteca wi-fi que conecta o IoT na rede (Emonlib.h e Wifi.h), essa integração não pode ser realizada pelo fato de que quando feita, os valores lidos pelos sensores tornavam-se incorretos. Por esse motivo, foram realizados testes separados e posteriormente será buscado uma solução para esse problema.

Conseqüentemente, como o dispositivo não apresentava um bom funcionamento na leitura dos dados quando conectado a rede, não foi possível realizar o tratamento dos dados através do banco de dados não relacional, em vista disso foi proposto realizar o monitoramento dos dados através do Google Forms, onde os dados são encaminhados para uma planilha excel.

Por fim, quando os testes são realizados separadamente, um deles acionando os dispositivos com a Alexa e o outro coletando a leitura dos dados, os resultados obtidos são positivos

7. Conclusão

O objetivo no desenvolvimento deste trabalho foi desenvolver uma tomada inteligente utilizando conceitos de IoT e apresentar um diferencial não muito visto ainda no

mercado, a integração com uma assistente pessoal. O objetivo no desenvolvimento do protótipo era realizar a visualização e o possível tratamento dos dados relacionados à tensão e corrente da residência, com o intuito de diminuir o desperdício de energia. Podemos afirmar que o resultado foi satisfatório e atingimos o objetivo. Além disso, como objetivo principal, foi desenvolvido uma integração do IoT com uma assistente pessoal, permitindo ao usuário o controle e monitoramento dos aparelhos eletrônicos de uma residência, concedendo ao usuário mais praticidade e comodidade na operação dos mesmos. Com relação a esse objetivo podemos dizer que ele foi atingido e o usuário é capaz de acionar os equipamentos através da assistente pessoal.

8. Referências Bibliográficas

ASHTON, Kevin. **That 'Internet of Things' Thing** -RFID Journal. 2009.

CRAZE, Cedric. **A verdadeira história da Internet das Coisas e como ela vai mudar o mundo**. Disponível em:

<<https://www.pollux.com.br/blog/a-verdadeira-historia-da-internet-das-coisas/>> Acesso em: 23 de Maio de 2020.

JONCO, C. M.; SILVEIRA, S. C. **Hey Siri: Inteligência Artificial e a humanização dos assistentes pessoais**. Universidade do Vale do Rio dos Sinos, 2015. Disponível em:

<http://www.repositorio.jesuita.org.br/bitstream/handle/UNISINOS/6407/Camila%20Medeiros%20Jonco_.pdf?sequence=1>. Acesso em: 25 de Maio de 2020.

Ranking das Tarifas. Disponível em: <<http://www.aneel.gov.br/ranking-das-tarifas>>. Acesso em: 14/06/2020.

WAKA, Grace Mungunda. **Controle remoto de tomadas elétricas baseado nos conceitos da Internet das Coisas**. Universidade Federal do Rio Grande do Sul, 2015.

Disponível em:

<<https://www.lume.ufrgs.br/bitstream/handle/10183/126073/000972344.pdf?seque>>. Acesso em: 25 de Maio de 2020.

WANG, L; PENG, D.; ZHANG, T. Desing of Smart Home System Based on WiFi Smart Plug. **International Journal of Smart Home**, s.l., v.9, n. 6, p. 173-182, jun/set. 2015.

WANGHAM, M.; DOMENECH, M.; MELLO, E. **Infraestrutura de autenticação e de autorização para internet das coisas**. [s.l.]: Minicursos do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, 2013. Disponível em: <<http://docente.ifsc.edu.br/mello/artigos/wangham-mc-sbseg13.pdf>>. Acesso em: 25 de Maio de 2020.

SILVA, Igor Alexandre Dutra. **Sensor tensão-corrente inteligente com monitoramento e controle on-line por smartphone**. Universidade Federal do Rio Grande do Sul, 2016. Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/151012/001009767.pdf?sequence=1>>. Acesso em: 13 de Setembro de 2020

MELLO, Raphael Vasconcelos Nunes de; LIMA, Rachel Batalha de; SILVA, Edgard Luciano Oliveira da. **Monitoramento do Consumo de Energia Elétrica Utilizando Uma Tomada Inteligente**. Universidade do Estado do Amazonas, Manaus, 2016. Disponível em: <<http://lcv.fee.unicamp.br/images/BTSym18/Papers/138.pdf>>. Acesso em: 16 ago. 2020.