

DESENVOLVIMENTO DE UM PROTÓTIPO IOT PARA COMANDAR E MONITORAR TOMADAS.

Fabio Daniel dos Santos Silva¹, Idalmir de Souza Queiroz Junior²

Resumo: O crescimento da utilização de equipamentos que possuem a função *stand-by*, fez com que o consumo de energia elétrica nas residências aumentasse. Atualmente existem diferentes maneiras de eliminar ou reduzir totalmente esse gasto. O presente trabalho tem como objetivo criar um protótipo capaz de comandar e obter estimativas de grandezas de uma tomada remotamente através da internet, como também, desligar todas as tomadas ao detectar uma corrente baixa na tomada principal. Evitando assim o consumo dos aparelhos que façam uso da função *stand-by*. Para isso foi utilizado um *nodeMCU*, um microcontrolador bastante eficiente para projetos que envolvem internet das coisas, além da plataforma *Adafruit IO*, uma solução para a realização da proposta do trabalho. A partir dela foi possível fazer toda a comunicação entre máquinas, além de comandar tomadas e visualizar graficamente os valores das grandezas medidas.

Palavras-chave: Internet das coisas; *Standby*; tomadas inteligentes; Eficiência energética.

1. INTRODUÇÃO

O consumo de energia elétrica residencial no Brasil vem aumentando, e um dos causadores desse crescimento é a utilização de equipamentos com a função de *Stand-by*. O modo *stand-by* significa que o aparelho está temporariamente em repouso, ou seja, ele não está desligado, continua consumindo energia e está aguardando algum comando para que o equipamento exerça sua função principal [1]. Apesar dessa função consumir pouca energia, utilizar vários equipamentos em *Stand-by* pode aumentar consideravelmente a fatura de energia elétrica da residência. Segundo o Instituto Akatu, esse tipo de consumo em uma cidade metropolitana pode representar 15% a 45% do consumo médio total de energia dos equipamentos de uma residência [2]. Diante desse problema, existe alguma maneira de reduzir o consumo de energia elétrica proveniente do uso da função *standby*?

Um grande exemplo de aparelho que usa da função de *stand-by* é a televisão, e só em 2017 foi vendida cerca de 10,4 milhões de unidades, um incremento de 21% em relação a 2016 [3]. De 69,3 milhões de domicílios particulares permanentes no Brasil, apenas 2,8%, ou 1,9 milhão, não tinham televisão, com destaque para o Norte do país, onde o percentual é o mais elevado, 6,3% [4].

Uma das propostas para solucionar esse problema, é a utilização de equipamentos que eliminam ou reduzem significativamente o consumo de energia, como por exemplo *Bye-bye stand-by* que é um adaptador a ser acoplado às tomadas que pode ser desligado por controle remoto [1], uma outra proposta é um sistema no qual você possa ligar/desligar tomadas e monitorar grandezas através da internet como por exemplo: corrente e potência, tal ideia pode ser encaixada no conceito de filtro de linha inteligente, um exemplo desse equipamento é o *i9plug* [5].

Esse trabalho tem como objetivo desenvolver um dispositivo com três tomadas, alternativo e similar a um filtro de linha inteligente e através dele monitorar grandezas de corrente e potência, e também ligar e desligar as tomadas, todo esse controle será feito por um aplicativo web podendo ser acessado por qualquer dispositivo conectado à internet, nesse aplicativo serão dispostos botões para o acionamentos das tomadas e um botão para desligamento automático das tomadas caso verifique que algum equipamento está em modo *stand-by*, no aplicativo também será exibido os dados medidos de corrente e potência através de gráficos para uma fácil visualização e interpretação.

2. REFERENCIAL TEÓRICO

2.1. Internet das coisas

O termo internet das coisas vem do inglês *Internet of Things* (IoT), ela se refere à integração de objetos físicos e virtuais em redes conectadas à internet, permitindo que “coisas” colem, troquem e armazenem uma enorme quantidade de dados numa nuvem, em que uma vez processados e analisados esses dados, gerem informações e serviços em escala inimaginável [6]. Devido a essa integração entre os objetos físicos e virtuais foi possível se criar os objetos inteligentes como por exemplo: *smartphones*, *notebook*, TV, entre outros. Esses objetos tem um papel fundamental para a evolução da internet das coisas, ao utilizar os recursos desses itens surgem uma gama de novas possibilidades para aplicação dessa tecnologia, como por exemplo cidades inteligentes, casas inteligente e saúde [7].

A IoT vem gerando impactos em diversos setores, tais como a indústria, saúde e também como a sociedade vem consumindo informação. Ela foi apontada como uma revolução tecnológica com o mercado mundial estimado em 1,7 trilhão de dólares [6].

2.2. Filtro de linha inteligente

Apesar de possuir versões comerciais, ainda não há exatamente um conceito para o que é um filtro de linha inteligente, ele pode ser definido como aquele de além de prover eletricidade e proteger os equipamentos, possuem algumas dessas funções [5]:

- Agendar horários para ligar/desligar aparelho
- Controlar todos equipamentos através de um aplicativo
- Controlar as tomadas com ou sem internet
- Proteger equipamentos
- Monitorar temperatura

A utilização de um filtro de linha inteligente é uma das soluções mais eficiente para reduzir o consumo de energia elétrica em uma residência devido o usuário conseguir eliminar totalmente o consumo de energia elétrica por conta da função *stand-by* de um aparelho apenas desligando com auxílio do aplicativo.

Atualmente existem várias versões de filtro de linha inteligente no mercado nacional e internacional, um dos mais famosos é o *Broadlink MP1*, com ele é possível controlar até quatro tomadas separadamente com o smartphone, porém ele não possui o padrão brasileiro de tomada, e ele é encontrado na internet em conversão direta por aproximadamente 110,00 sem considerar frete e os impostos [11]. Um produto brasileiro bastante semelhante é o *i9Plug* com ele é possível também controlar até quatro tomadas de forma independente e diferente da versão anterior ele possui o padrão brasileiro de três pinos.

2.3. Protocolo MQTT

Message Queuing Telemetry (MQTT) é um protocolo de troca de mensagens entre M2M (*Machine-to-Machine*) criada pela IBM e padronizada depois pela empresa OASIS, esse protocolo é bastante utilizado em projetos de IoT por necessitar pouca largura de banda e *hardware* extremamente simples [8].

O protocolo MQTT é baseado no TCP/IP e no paradigma de mensagens *publisher* e *subscriber*, onde os dados são enviados para um intermediário chamado MQTT *broker*. A função do *publish* é simplesmente pegar o valor lido por algo, como por exemplo um sensor e publicar esse valor no *broker* ou para os clientes (notebook/celular), já o *subscriber* é a maneira do cliente solicitar uma requisição para o *broker* e receber os valores publicados pelo os sensores. Esse esquema pode ser visto na Figura 1.

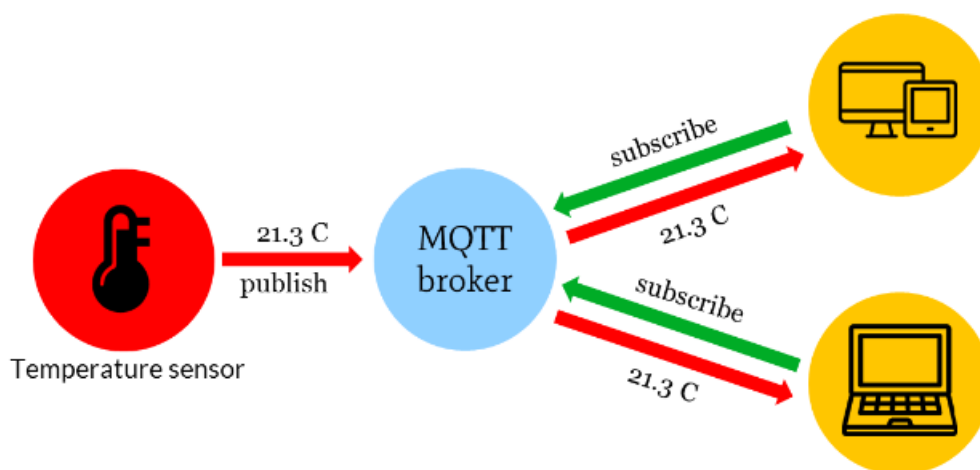


Figura 1. Esquematização M2M. [9]

2.4. Adafruit IO

O *Adafruit IO* é uma solução criada pela *adafruit* para conectar os *hardwares* de maneira fácil com a nuvem. Podendo funcionar como um MQTT *broker*, que é um intermediário entre os clientes. O *Adafruit IO* é uma solução bastante completa, nela possui: *dashboard*, *websockets*, *api*, e bibliotecas para facilitar a utilização dos *hardwares* [10].

A plataforma é excelente para se trabalhar em projetos de IoT, já que ela possui uma biblioteca bastante simples e é compatível com diversos *hardware*, como por exemplo: *Arduino*, *raspberry pi*, *ESP8266* e *NodeMCU*. Apesar de ser uma plataforma paga, nela possui uma versão gratuita, porém com algumas limitações [10].

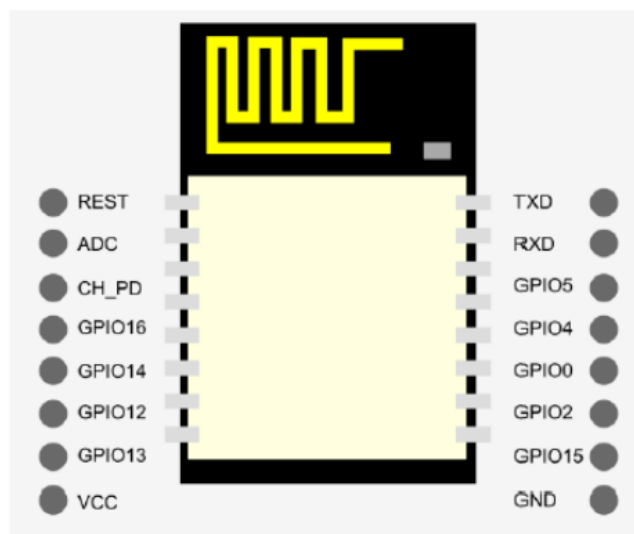
2.5. ESP8266 e NodeMCU

2.5.1. ESP8266

O ESP8266 é um microcontrolador desenvolvido pela *espressif*, sua principal característica é que ele possui wifi embutido com o padrão 802.11 b/g/n e por ser bastante compacta, o chip ainda possui uma CPU 32 bits xtensa LX101 da *Tensilica* com um *clock* 80 MHz e com a possibilidade de operar em até 160 MHz, o seu chip wi-fi possui um alcance aproximado de 90 metros, a tensão de operação do microcontrolador é entre 2,6 V a 3,3 V e a corrente de operação de 80 mA [12]. Por conta de sua característica de ser bastante compacto e de requerer uma quantidade baixa energia elétrica, ele é bastante viável para projetos de pequeno porte.

No mercado atualmente existem uma grande variedade de ESP8266 como por exemplos os módulos ESP-01, ESP-05 e ESP-12. Apesar de eles possuírem os mesmos processador cada um possui características próprias sejam elas em tamanho ou a quantidade de entradas GPIOs [13]. A Figura 2 mostra o diagrama de pinagem do ESP-12

Figura 2. Diagrama de pinagem ESP-12 [13]



2.5.1. NodeMCU

O módulo NodeMCU faz parte da família EPS8266. Nele possui o chip ESP-12E isso faz com que o módulo possua comunicação via Wi-Fi, além disso é possível encontrar nele 11 pinos de entrada e saída e ainda circuito regulador de tensão fazendo com que possa ter uma tensão de operação entre 4,5 V à 9 V, uma outra característica importante é que ele possui conectividade USB para transferência do código fonte do programa através uma porta micro usb [14]. O modulo NodeMCU pode ser visto na Figura 3.

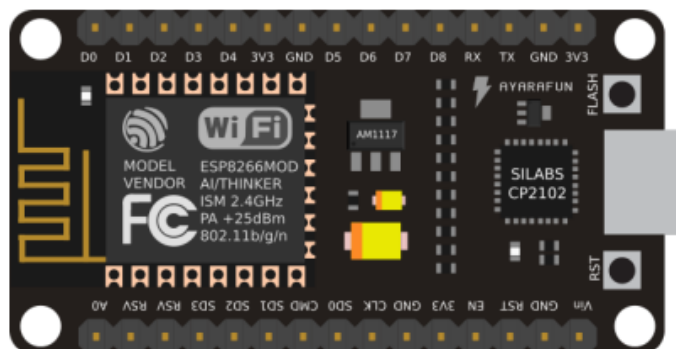


Figura 3. NodeMCU [14]

O NodeMCU é uma das placas mais interessantes para IoT atualmente [14], por ser uma placa bastante compacta, completa e simples de se usar e ainda possui módulo WiFi já integrado na placa que é um fator de extrema importância em projetos que envolvam IoT, um outro fator importante é que nela já possui uma porta micro usb isso faz com que não haja a necessidade de utilizar outros módulos em conjunto com o nodeMCU como por exemplo: Arduino, *Raspberry* ou módulo FTDI.

A placa pode ser facilmente encontrada no mercado, sendo capaz de ser encontrada no exterior por apenas R\$13,00 sem considerar frete e taxas de importação [15], além disso pode ser encontrada no mercado brasileiro, no site do Mercado Livre foi possível encontrar o módulo por até R\$50,00, igualmente sem considerar as taxas de frete [16].

2.6. Módulo Relé

O Relé é um módulo capaz de chavear vários dispositivos elétricos, como por exemplo: lâmpadas, tomadas, motores e etc. Ele pode ser controlado por diversos tipos de microcontroladores, seja ele, Arduino, NodeMCU ou até mesmo o *Raspberry pi*, o controle desse módulo é feito a partir de um envio de sinal digital do microcontrolador, fazendo com que acione ou não o dispositivo dentro do relé [17].

Os módulos relé podem ser encontrados em vários modelos, com diferentes tipos de tensão e corrente de operação ou com a quantidade de relé disponível no módulo diferente, sendo capaz de ser encontrado módulos com 2 a 16 relés em uma placa. Os relés contidos no módulo são do tipo eletromecânicos. Na Figura 4 pode ser visto o módulo relé com 2 canais.



Figura 4. Módulo relé de 2 canais [17].

O princípio de funcionamento do relé eletromecânico se dá a pôr efeitos eletromagnéticos. Dentro dele possui um eletroímã e próximo a ele há uma armadura de metal, o eletroímã tem o intuito de controlar um jogo de contatos, quando há uma corrente percorrendo esse eletroímã, um campo magnético é criado, e isso faz com que haja uma atração magnética, ocorrendo a junção dos contatos e por consequência a corrente pode fluir através da armadura de metal, conforme mostra a Figura 5.

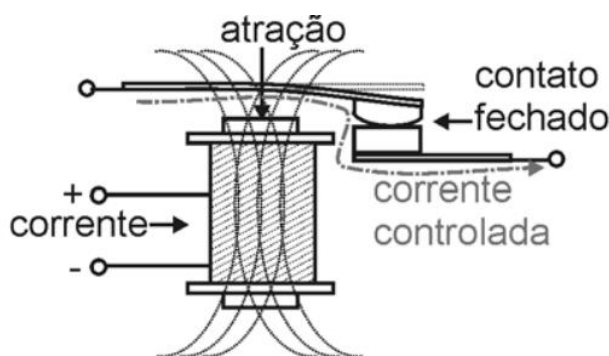


Figura 5. Funcionamento do relé eletromecânico [18].

Quando a corrente que percorre o eletroímã é interrompida não há mais a atração magnética, por consequência abrindo o circuito, dessa maneira não haverá mais corrente fluindo através da armadura.

3. MATERIAIS E METODOS

O Trabalho foi dividido em três etapas, como pode ser visto na Figura 6. A primeira etapa foi definida como será feito o esquema do circuito destinado a comandar e monitorar tomadas, devido a limitações do microcontrolador utilizado (nodeMCU) possuir apenas uma entrada analógica (ADC), somente será possível monitorar grandezas de uma tomada. A segunda etapa será feita a montagem do protótipo em si, onde foi definido o material que vai compor o protótipo e como foi disposto cada elemento do circuito. E por fim a última etapa será a da programação do microcontrolador, nessa etapa foi definido como será feita a comunicação entre dispositivos para comandar e monitorar as tomadas.

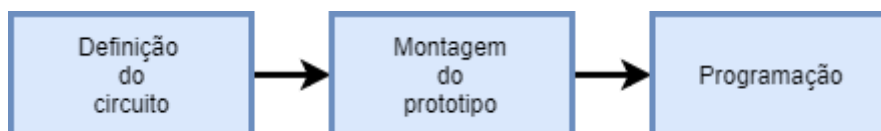


Figura 6. Fluxograma do trabalho. (Autoria própria)

3.1. Definição do circuito

As partes que vão compor o circuito são: 1 NodeMCU, 2 módulos relés de 2 canais, 1 ACS712 30 A e 3 tomadas.

A monitoração de corrente e potência da tomada será feita através do módulo medidor de corrente ACS712 30A, este é um sensor desenvolvido pela *Allegro*. Este sensor tem a finalidade de medir corrente entre -30 a 30 A, utilizando-se do efeito hall para medição e a posteriori gerar na saída do ACS712 uma tensão proporcional da corrente encontrada, e essa proporção para o módulo de 30 A é de 66 mV/A, ou seja para cada variação de 1 A, na saída varia uma tensão de 66 mV [19]. O ACS712 é considerado um sensor invasivo, visto que se há a necessidade de abrir o circuito. Devido a limitação do NodeMCU ter apenas uma entrada analógica, a potência aparente consumida pelo aparelho conectado à tomada será obtida pelo produto da corrente elétrica com a tensão de 220V da rede.

As tomadas serão conectadas à rede de energia elétrica através 2 módulos relés de 2 canais, sendo capaz de controlar até 4 tomadas de maneira individual. O modelo escolhido foi um módulo relé possível de trabalhar com 125 V a 250 V (AC) e 28 V a 30 V (DC), podendo suportar uma corrente de até 10 A.

O microcontrolador escolhido foi o NodeMCU, pois apesar de suas limitações de conter apenas uma entrada analógica, é um módulo excelente para trabalhar com pequenos projetos, e ter um custo pequeno, além de já possuir um módulo Wi-Fi (ESP-12E) já embutido, algo essencial para o projeto, visto que se trata de internet das coisas.

Com todos componentes que vão compor o circuito já escolhidos, foi definido a melhor forma de montar o circuito. O esquema do circuito pode ser visto na Figura 7.

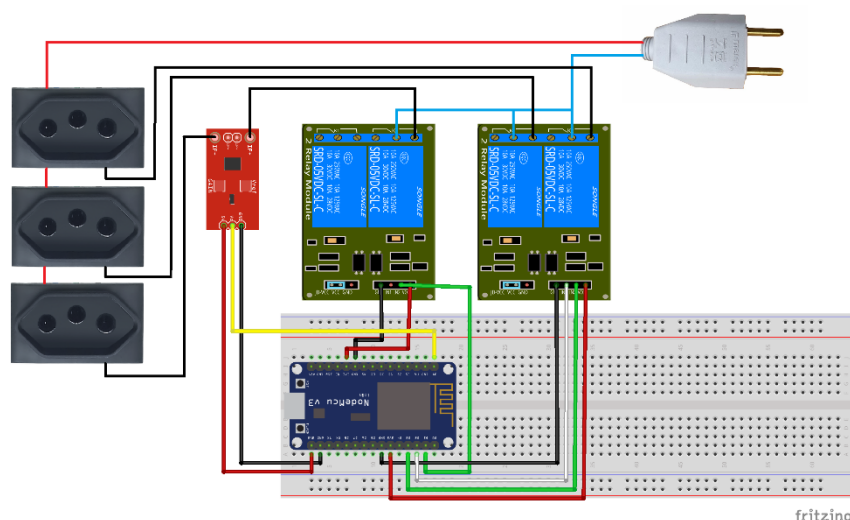


Figura 7. Esquema do circuito proposto. (Autoria própria)

O circuito da Figura 7 contém 3 LEDs um para cada tomada, omitidos por simplicidade, os LEDs têm a finalidade de acender enquanto a tomada estiver energizada, desta forma se pode observar quais tomadas estão ligadas.

3.2. Montagem do protótipo

Foi utilizado madeira para a construção de uma caixa retangular onde serão dispostos os elementos do protótipo apresentado na Figura 7, escolheu-se madeira por ser simples de trabalhar e fácil de encontrar. A caixa possui as seguintes dimensões, 23,5 cm de comprimento, 12,5 cm de largura e com 5,5 cm de altura. No topo da caixa há um corte retangular de 7x4 cm e 3 furos de cerca de 5 mm, o corte retangular tem a finalidade de se inserir as tomadas do circuito, já os furos tem o intuito de se inserir os 3 LEDs.

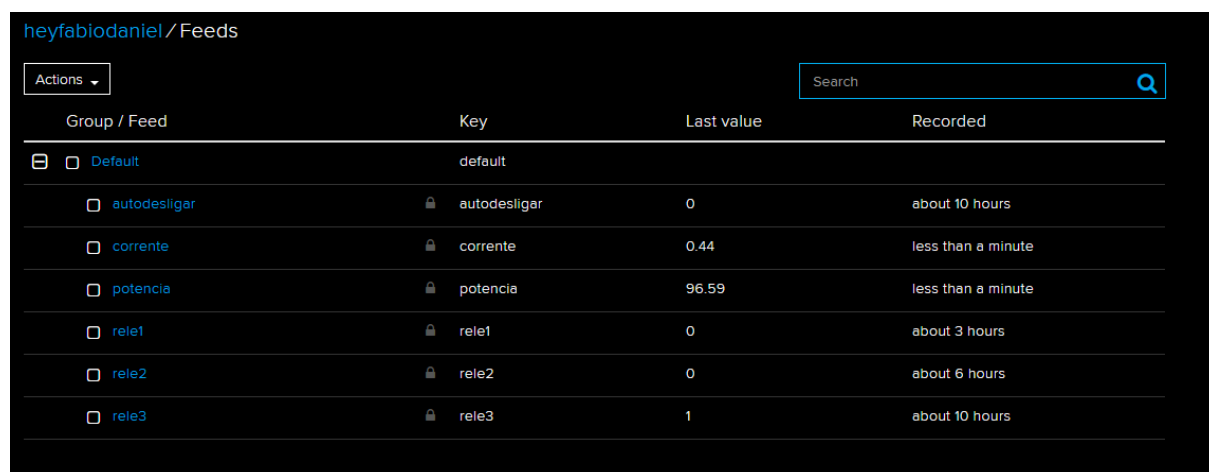
Para o protótipo foi definido que haveria apenas 3 tomadas, apesar do circuito visto na Figura 7 ser capaz de controlar até quatro tomadas individualmente

Outro fator importante definido nessa etapa foi como será feita a alimentação do microcontrolador, para isso foi adicionada uma tomada extra dentro da caixa para ligar um carregador para alimentação do NodeMCU através de sua porta micro usb, portanto, assim que o dispositivo for conectado a tomada o carregador poderá energizar o microcontrolador.

3.3. Programação

O meio de transmissão de dados escolhido foi a internet, via Wi-Fi, utilizando o protocolo MQTT, por meio do modulo ESP-12E contido no NodeMCU. O controle das tomadas e armazenamento dos dados será realizado através do site *Adafruit IO*, por sua simplicidade e atuação como um *broker*.

O armazenamento de dados será feito na plataforma *Adafruit IO*, para tanto, foi definido na plataforma 6 *feeds*. Os *feeds* são uma espécie de variável que vão receber os valores enviados pelo sensor através do *publish* ou armazenar a requisição enviada pelo cliente através do *subscribe*. Os 6 *feeds* criados foram: autodesligar, corrente, potência, rele1, rele2 e rele3, nos *feeds* corrente e potência serão armazenados o último valor lido pelo sensor ACS712, nos *feeds* rele1, rele2 e rele3 serão armazenados o último valor do estado do relé 0 (baixo) ou 1 (alto), o feed autodesligar irá guardar valores 0 ou 1 onde será definido se o desligamento automático vai estar ligado ou não. A página com os *feeds* pode ser observado na Figura 8.



Group / Feed	Key	Last value	Recorded
Default	default		
☐ autodesligar	autodesligar	0	about 10 hours
☐ corrente	corrente	0.44	less than a minute
☐ potencia	potencia	96.59	less than a minute
☐ rele1	rele1	0	about 3 hours
☐ rele2	rele2	0	about 6 hours
☐ rele3	rele3	1	about 10 hours

Figura 8. Pagina do Adafruit IO com os feeds. (Autoria própria)

O *Adafruit IO* ainda possui uma maneira de controlar os relés por meio de botões, e uma maneira de visualizar de maneira mais fácil, através de gráficos em função do tempo, os valores armazenados nos *feeds* corrente e potência, isso tudo é feito por meio da *dashboard* disponível na plataforma. A Figura 9 mostra a *dashboard* do projeto.



Figura 9. Dashboard do projeto. (Autoria própria)

A versão utilizada do *Adafruit IO* é gratuita, apesar de ter algumas limitações, a versão consegue suprir bastante a necessidade do projeto. A *dashboard* pode ser acessada através de smartphone, assim o usuário poderá ter diferentes meios para acessar a *dashboard*.

Com a figura 10 é possível observar o fluxograma do código fonte do projeto.

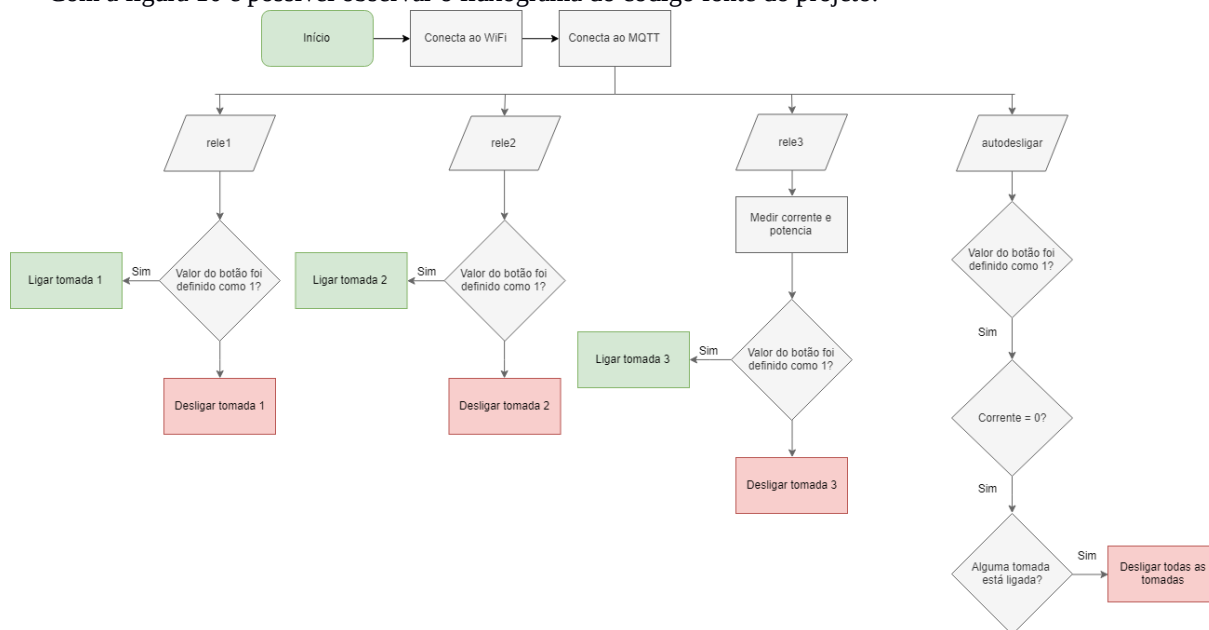


Figura 10. Fluxograma do código fonte. (Autoria própria)

Primeiramente o microcontrolador é conectado ao wi-fi e posteriormente ao protocolo MQTT. No *broker* do *Adafruit IO* são definidos os botões para controle das tomadas e o botão para desligamento automático. Se o botão das tomadas estiver ligado, o valor 1 (alto) é enviado para o NodeMCU através do *publish* fazendo com que energize a tomada, caso contrário, desligando a tomada. O botão para desligamento automático faz com que ative ou desative essa função, o desligamento automático somente ocorre se a corrente medida for igual a zero e se alguma das tomadas estiverem energizada. Quando um equipamento está em *stand-by* o valor medido pelo sensor é bastante baixo dessa forma a corrente foi considerada igual a zero. As grandezas de corrente e potência são medidas pelo módulo ACS712 e enviadas ao *broker* para serem armazenados, e no mesmo instante é possível ser visto esses dados através dos gráficos e do *gauge* disponível na *dashboard*.

4. RESULTADOS E DISCUSSÕES.

4.1. Custo total do circuito

A Tabela 1 mostra os valores dos principais componentes utilizados no projeto junto com preço total gasto. Uma observação importante é que grande parte dos produtos foram comprados pela internet através do site do Mercado Livre e os valores de fretes não foram considerados.

Tabela 1 – Custo total do circuito. (Autoria Própria)

Componentes	Valor unitário	Valor total
2x modulo relé 2 canais	R\$ 18,00	R\$ 36,00
1x ACS712	R\$ 22,00	R\$ 22,00
20x jumpers	R\$ 10,00	R\$ 10,00
1x NodeMCU	R\$ 40,00	R\$ 40,00
1x tomada tripla	R\$ 18,00	R\$ 18,00
3x LEDs	R\$ 1,00	R\$ 3,00
3x Resistor	R\$ 0,25	R\$ 0,75
TOTAL		R\$ 129,75

O valor total do custo gasto no circuito poderia ter sido menor, se a compra dos componentes tivesse sido feita em um site internacional. O preço total ficou aproximadamente 30% do valor de um *Broadlink MP1* que custa R\$ 110,00 e um pouco a baixo do produto nacional o *i9plug* que custa R\$ 190,00. O *i9plug* atualmente está sem estoque e o preço final dele pode ser um pouco mais caro. Apesar disso o valor total pode ser considerado aceitável levando em consideração que os componentes foram comprados no Brasil e de que o protótipo proposto pode ser adicionado diferentes tipos de funções a partir dos valores obtidos de corrente e potência.

4.2. Resultados

Para os resultados, foram feitos dois diferentes testes. Os testes foram feitos com diferentes aparelhos com potencias diferentes. A Figura 11 mostra o protótipo final montado.



Figura 11. Protótipo final. (Autoria própria)

O primeiro teste tem intuito de verificar o devido funcionamento do sensor de corrente ACS712, para isso foram conectados diferentes aparelhos para medição de corrente e potência. O primeiro aparelho ligado foi um ventilador que possui uma potência nominal de 80 W, a Figura 12 mostra o *gauge* com os valores obtidos de corrente medido pelo sensor de corrente como também a corrente medida através de um multímetro.



Figura 12. Valor de correntes obtido por multímetro e ACS712. (Autoria própria)

O mesmo teste foi feito também com outros equipamentos, a tabela 2 mostra os valores obtidos da potência para cada equipamento junto com seu erro percentual. O erro foi calculado a partir da equação 1.

$$Erro = \frac{|Corrente\ multímetro - Corrente\ ACS712|}{Corrente\ multímetro} * 100 \quad (1)$$

Tabela 2 – Potência obtida e erro percentual. (Autoria Própria)

Equipamento	Corrente multímetro	Corrente ACS712	Erro percentual
Ventilador	0,26A	0,27A	3,84%
Televisão	0,60A	0,65A	8,33%
Ferro de solda	0,06A	0,02A	66,67%
Torradeira	3,17A	3,57A	12,62%
Liquidificador	0,69A	1,01A	46,37%

Como visto, os erros foram pequenos para certos casos, porém deve-se levar em consideração que sempre irá existir erros devido a medições incorretas, provenientes de falha humana ou até mesmo do multímetro. Um outro fator importante que fez com que gerasse erros é de que o ACS712 possui um ruído de aproximadamente 80mA, para medições com corrente mais alta esse ruído pode ser desconsiderado, porém ao se medir correntes baixas faz com que ocorra erros maiores.

Para o segundo teste, foi observado o comportamento dos gráficos de potência e correntes visto na *dashboard* e também o funcionamento do desligamento automático das tomadas. Para isso o protótipo foi ligado por um dia inteiro a uma televisão de 145W de potência conectada em uma das tomadas. A Figura 13 mostra os gráficos obtido do uso diário do aparelho.



Figura 13. Valor de corrente e potência obtido. (Autoria própria)

Como visto na figura 11, todos os dados de correntes e potências foram armazenados corretamente, os gráficos de corrente e potência ficaram semelhantes, diferenciando-se apenas os valores. Isto se deve ao fato de o NodeMCU só possuir uma entrada analógica, por este motivo a tensão foi definida com o valor de 220V. O plano gratuito do *Adafruit IO* possui uma limitação de 30 envios de dados por minuto, isso fez com que houvesse a necessidade de adicionar um *delay* no envio de dados para que não excedesse esse valor. O *delay* não prejudica o projeto visto que não há necessidade de armazenar o valor da corrente a cada segundo. O plano gratuito ainda dá o direito do usuário visualizar e baixar todos os dados armazenados nos *feeds* dos últimos 30 dias, o *download* pode ser feito em dois formatos: JSON e CSV.

O botão do desligamento automático permaneceu desativado o dia inteiro, somente a noite foi ativado. Ao deixar a televisão em *standby* todas tomadas foram desligadas automaticamente depois de alguns segundos. Logo, tanto o armazenamento de dados e quanto o botão de desligamento automático funcionaram conforme desejado.

5. CONCLUSÃO.

O projeto teve o objetivo de construir um aparelho capaz de monitorar e controlar tomadas como também a de reduzir o consumo de energia elétrica proveniente do uso do *standby*. Obtendo êxito em praticamente todos os objetivos. Um dos maiores problemas do projeto foi o seu preço final que não obteve uma resposta muito satisfatória, já que é bastante semelhante à de um *broadlink MP1*, porém existem maneiras de deixar o protótipo ainda mais barato, como por exemplo comprar os componentes dos circuitos em sites internacionais ou utilizar componentes que não seja acabado ou próprio para prototipagem, já que o custo real de um componente que faça a mesma função é mais barato. Utilizar o desligamento automático é uma maneira bastante inteligente para reduzir o custo de energia elétrica, visto que uma das maneiras mais eficiente para reduzir o consumo é eliminar totalmente o gasto proveniente da função *standby*. O *Adafruit IO* foi uma excelente plataforma e o plano gratuito foi suficiente para as necessidades do projeto mesmo com a grande quantidade de envio de dados, um outro fator excelente dessa plataforma é de que eles possuem um site responsivo fazendo com que o site seja acessível em diferentes tipos de telas, dessa forma o usuário tem inúmeras maneiras de controlar seu projeto.

Como sugestão para trabalhos futuros, recomenda-se utilizar um outro microcontrolador que seja capaz de medir a corrente de todas tomadas individualmente e não apenas uma, já que o *nodeMCU* há apenas uma entrada analógica. Outra sugestão é adicionar novas funções ao protótipo, como por exemplo: agendamentos para ligar ou desligar as tomadas. E por fim uma última sugestão é acrescentar alguma maneira de proteção para os equipamentos ligado ao protótipo.

6. REFERÊNCIAS.

- [1] RODRIGUES, Jean Ronir Ferraz. **AValiação da Utilização do Modo Standby em Eletrodomésticos e de Propostas de Soluções Mitigadoras para Redução do Seu Consumo Energético**. 2009. 124 f. Dissertação (Mestrado) - Curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Paraná, Curitiba, 2009.
- [2] AKATU (Brasil) (Org.). **Economia de energia: um assunto para tirar do standby**. 2012. Disponível em: <<https://www.akatu.org.br/noticia/economia-de-energia-um-assunto-para-tirar-do-standby/>>. Acesso em: 14 jul. 2019.
- [3] FOLHA DE S. PAULO (Brasil) (Ed.). **Venda de TVs deve crescer 22% com Copa, afirma consultoria**. 2018. Disponível em: <<https://www1.folha.uol.com.br/mercado/2018/03/venda-de-tvs-deve-crescer-22-com-copa-afirma-consultoria.shtml>>. Acesso em: 14 jul. 2019.
- [4] EBC (Brasil). **Pesquisa diz que, de 69 milhões de casas, só 2,8% não têm TV no Brasil**. 2018. Disponível em: <<http://agenciabrasil.ebc.com.br/economia/noticia/2018-02/uso-de-celular-e-acesso-internet-sao-tendencias-crescentes-no-brasil>>. Acesso em: 14 jul. 2019.
- [5] I9PLUG (Org.). **A tomada inteligente brasileira**. Disponível em: <<https://i9plug.com.br>>. Acesso em: 14 jul. 2019.
- [6] INTERNET DAS COISAS: Nós, as cidades, os robôs, os carros: tudo conectado!. [s.i.]: Sociedade Brasileira de Computação, abr. 2015.
- [7] SANTOS, Bruno P. **Internet das Coisas: da Teoria à Prática**. 2012. 50 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2012.
- [8] OASIS. **MQTT Version**. 2014. Disponível em: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Introduction>. Acesso em: 14 jul. 2019.
- [9] ISTSOS. **MQTT Broadcasting**. Disponível em: <http://istsos.org/en/trunk/doc/ws_mqtt.html>. Acesso em: 14 jul. 2019.
- [10] ADAFRUIT. **Adafruit IO**. Disponível em: <<https://io.adafruit.com>>. Acesso em: 15 jul. 2019.
- [11] GEARBEST (Org.). **Broadlink MP1 Smart WiFi Power Strip with 4 Outlet**. Disponível em: <https://br.gearbest.com/power-strips/pp_360376.html>. Acesso em: 15 jul. 2019.
- [12] SYSTEMS, Espressif. ESP8266EX: Datasheet. 6. ed. Espressif Systems. 2018.
- [13] OLIVEIRA, Ricardo Rodrigues. **USO DO MICROCONTROLADOR ESP8266 PARA AUTOMAÇÃO RESIDENCIAL**. 2017. 55 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2017.
- [14] VELOSO, Artur F. S. et al. Prototipação com nodeMCU para Internet das Coisas em SmartCities: Uma Pesquisa. **Iii Escola Regional de Informática do Piauí. Livro Anais - Artigos e Minicursos**, [s.i.], v. 1, n. 1, p.105-110, jun. 2017.
- [15] ALIEXPRESS. **V3 NodeMcu módulo Sem Fio**. Disponível em: <<https://pt.aliexpress.com/item/32801590957.html>>. Acesso em: 16 jul. 2019.
- [16] MERCADO LIVRE. **NodeMCU**. Disponível em: <<https://lista.mercadolivre.com.br/nodemcu>>. Acesso em: 16 jul. 2019.
- [17] GBK, Robotics. **Modulo Rele de 2 Canais 10A: Especificações**. GBK Robotics.
- [18] BRAGA, Newton C. **Relés - Conceitos e Aplicações**. São Paulo: Ncb, 2012.
- [19] MICROSYSTEMS, Allegro. ACS712: Datasheet. Allegro Microsystems. 2017.



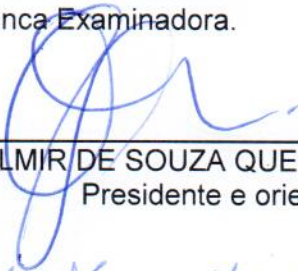
MINISTÉRIO DA EDUCAÇÃO
Universidade Federal Rural do Semi-Árido – UFERSA
Centro de Ciências Exatas e Naturais – CCEN

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Às 09:00 horas do dia nove de agosto de dois mil e dezenove, no laboratório de Energia, Prédio de Laboratórios de Engenharias I, reuniu-se a Banca Examinadora de defesa de trabalho de conclusão de curso de autoria do aluno **FABIO DANIEL DOS SANTOS SILVA**, aluno do curso de Bacharelado em Ciência e Tecnologia desta universidade, N° de matrícula **2016010248**, com o título **“DESENVOLVIMENTO DE UM PROTÓTIPO IOT PARA COMANDAR E MONITORAR TOMADAS”**. A Banca Examinadora ficou assim constituída por três membros: Prof. Dr. IDALMIR DE SOUZA QUEIROZ JUNIOR, presidente da banca e orientador do Trabalho de Conclusão de Curso; Prof. Me. SAMANTA MESQUITA DE HOLANDA e Prof. Me. DANIEL PABLO DANTAS DIOGENES, como membros. Concluída a defesa, procedeu-se o julgamento pelos membros da banca examinadora, tendo o aluno obtido as seguintes notas: 10,0; 10,0; 10,0. Apuradas as notas verificou-se que o aluno foi aprovado com média geral 10,0. E para constar, eu, Idalmir de Souza Queiroz Júnior, lavrei a presente ata que, após lida e aprovada pelos membros da banca examinadora, será assinada por todos.

Mossoró, 09 de agosto de 2019.

Assinatura dos membros da Banca Examinadora.



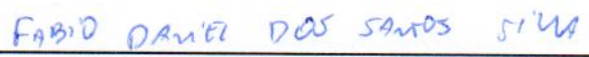
Prof. Dr. IDALMIR DE SOUZA QUEIROZ JUNIOR – UFERSA
Presidente e orientador



Prof. Me. SAMANTA MESQUITA DE HOLANDA – UFERSA
Primeiro Membro



Prof. Me. DANIEL PABLO DANTAS DIOGENES - UFERSA
Segundo Membro



FABIO DANIEL DOS SANTOS SILVA
Discente