



Universidade Federal
de Campina Grande

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

Cicero da Costa Freire Filho

**Módulo Interruptor IoT - Uma abordagem prática
no processo de desenvolvimento de um produto
eletrônico**

Dezembro de 2018
Campina Grande - Paraíba

Cicero da Costa Freire Filho

Módulo Interruptor IoT - Uma abordagem prática no processo de desenvolvimento de um produto eletrônico

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Danilo Freire de Souza Santos

Dezembro de 2018
Campina Grande - Paraíba

Cicero da Costa Freire Filho

Módulo Interruptor IoT - Uma abordagem prática no processo de desenvolvimento de um produto eletrônico

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Trabalho aprovado em: ___/___/_____

Professor Danilo Freire de Souza Santos, D.Sc.
Orientador

Professor Gutemberg Gonçalves dos Santos Júnior, D.Sc.
Avaliador

Campina Grande - Paraíba
Dezembro de 2018

*Dedico esse trabalho a meus pais,
por todo amor e suporte que me dão.*

Agradecimentos

Sendo uma pessoa limitada e falha como qualquer outra, minhas conquistas não vieram sem luta, mas jamais teria chegado nelas sozinho, só me resta então ser grato a quem me ajudou e continuar lutando.

E por isso eu agradeço...

Aos meus pais, Janeide e Cícero, pois se estou me formando é por causa deles. Toda e qualquer vitória minha só aconteceu ou acontecerá por ter tido pais que me deram todo o suporte que eu precisava para enfrentar os desafios. Lutaram para me dar a melhor educação possível e sempre estiveram comigo em todos os caminhos que escolhi seguir.

À minha namorada Mel, que me escutou, me aconselhou e me deu forças todas as vezes que me senti fraco ou incapaz e que me deu um colo para repousar nos momentos turbulentos desde que entrou em minha vida.

Aos meus amigos de Elétrica, do Xablau, do Motiva, ou da vida, que estiveram comigo nos momentos de comemoração ou de luta, que cresceram e mudaram comigo, enfrentando cada um a sua luta, não me deixaram cair na minha.

Ao meu Orientador Danilo, que dedicou seu tempo, paciência e conhecimento a me ajudar a preparar esse trabalho. Aos demais professores da UFCG, principalmente aqueles que não só transmitiram um pouco do seu conhecimento, mas que me inspiraram a ser um engenheiro melhor. Também à coordenação, nas pessoas de Tchai e Adail, que fazem muito mais do que prestar um serviço de qualidade, fazem também a experiência universitária um pouco mais leve para nós estudantes.

E também ao povo brasileiro, que estando eu numa universidade pública, só estou me tornando engenheiro por que o dinheiro dos impostos estão sendo investidos em mim. Que eu possa honrar esse investimento utilizando meu conhecimento à favor do Brasil.

*“-Can a man still be brave if he’s afraid?
-That is the only time a man can be brave”
George R.R. Martin, A Game of Thrones*

Resumo

Este projeto de conclusão de curso apresenta uma abordagem de desenvolvimento de circuitos eletrônicos implementada na prática na criação de um módulo interruptor Iot. São definidos requisitos claros do projeto, estruturada cada etapa do desenvolvimento, descritos os passos realizados e são apresentados os testes realizados para validar o funcionamento do protótipo criado. Foram utilizadas tecnologias como o Azure da Microsoft, o desenvolvimento de aplicativos Android, programação no microcontrolador Esp-32, simulação SPICE e projeto de placa de circuito impresso.

Palavras-chaves: Microcontrolador; Internet das coisas; Interruptor Iot; Computação em nuvem.

Abstract

This undergraduate final project introduces an electronic circuit development approach implemented in creation of an Iot switch module. Clear design requirements are defined, structured each stage of development, described the steps taken and the tests performed to validate the proper functioning of the prototype created. Technologies such as Microsoft's Azure, Android application development, Esp-32 microcontroller programming, SPICE simulation and printed circuit board design were used.

Key-words: Microcontroller; Internet of Things; Iot switch; Cloud Computing.

Lista de ilustrações

Figura 1 – Comparação entre pesquisas	16
Figura 2 – Ciclo de "Hype" das tecnologias emergentes	16
Figura 3 – Diagrama de funcionamento para o protocolo MQTT	19
Figura 4 – Placa de Desenvolvimento para o esp32	21
Figura 5 – Diagrama do modelo para o circuito do Interruptor IoT	24
Figura 6 – Diagrama de comunicação do sistema	24
Figura 7 – Fluxograma com uma metodologia para desenvolvimento de projetos eletrônicos	25
Figura 8 – Captura de tela de simulação realizada no <i>Multisim</i>	27
Figura 9 – Fotografia de teste realizado na <i>protoboard</i>	28
Figura 10 – Captura de tela do esquemático do Circuito no software online <i>EasyEDA</i> .	29
Figura 11 – Captura de tela da placa de circuito impresso projetada no <i>EasyEDA</i> .	29
Figura 12 – Fotografia da placa de circuito impresso projetada	30
Figura 13 – Vistas da caixa PB-107 utilizada para acomodar o circuito	31
Figura 14 – Fotografia do projeto dentro de uma caixa plástica com um interruptor	31
Figura 15 – Captura de tela da página para inserir credenciais da rede WiFi a ser conectada.	33
Figura 16 – Captura de tela do portal do Azure com a página do dispositivo criado	33
Figura 17 – Captura de tela do aplicativo criado para acionamento do módulo interruptor IoT	35

Lista de tabelas

Tabela 1 – Componentes utilizados	32
Tabela 2 – Teste de alimentação	36
Tabela 3 – Teste de acionamento do relé	37
Tabela 4 – Teste de conexão	37
Tabela 5 – Teste de recebimento de credenciais	37
Tabela 6 – Teste de desligar a energia.	38
Tabela 7 – Teste de interrupção	38
Tabela 8 – Teste de desconexão	38

Lista de abreviaturas e siglas

IOT	Internet das Coisas
HTTP	Hypertext Transfer Protocol
MQTT	Message Queuing Telemetry Transport
PCB	Printed Circuit Board
AP	Access-Point
SMD	Surface Mounting Device

Sumário

1	INTRODUÇÃO	14
1.1	Objetivos	14
2	REVISÃO BIBLIOGRÁFICA E TECNOLÓGICA	15
2.1	Internet das Coisas	15
2.1.1	Automação residencial e predial	17
2.1.2	Protocolos de comunicação	17
2.1.2.1	HTTP - <i>Hypertext Transfer Protocol</i>	18
2.1.2.2	MQTT - <i>Message Queuing Telemetry Transport</i>	18
2.2	Microsoft Azure	19
2.3	Microcontrolador Esp-32	20
3	PROPOSTA DE SOLUÇÃO	22
3.1	Requisitos de projeto	22
3.1.1	Requisitos funcionais	22
3.1.2	Requisitos não-funcionais	23
3.2	Modelo proposto	23
3.2.1	Integração do projeto	24
3.3	Planejamento do Projeto	24
4	DESENVOLVIMENTO DO PROJETO	27
4.1	Desenvolvimento do Hardware	27
4.1.1	Simulação do circuito	27
4.1.2	Prototipação na <i>protoboard</i>	28
4.1.3	Design da placa PCB	28
4.1.4	Protótipo em PCB	30
4.1.5	Lista de Componentes	30
4.2	Desenvolvimento do Software	32
4.2.1	Conexão WiFi	32
4.2.2	Conexão com o Azure IoT Hub	32
4.2.3	Aplicativo para acionamento	34
5	VALIDAÇÃO E DISCUSSÃO DOS RESULTADOS	36
5.1	Testes de validação	36
5.2	Discussão de resultados	38
6	CONCLUSÕES	40

REFERÊNCIAS	41
--------------------	-----------

1 Introdução

O desenvolvimento de circuitos eletrônicos é um processo longo, dinâmico e complexo, onde a aplicação vai definir a rigorosidade do processo de desenvolvimento. Um projeto pessoal simples pode ser realizado apenas na *proto-board*, outros podem requerer soldar numa placa perfurada, mas em um projeto profissional, o desenvolvimento de uma placa de circuito impresso (PCB - Printed Circuit Board) é necessário, de modo a definir o melhor projeto para segurar mecanicamente e conectar eletricamente os componentes do circuito, garantido o funcionamento planejado do circuito e facilitando a acomodação do mesmo.

Dentre os produtos eletrônicos comerciais, circuitos para Internet das Coisas (IoT - *Internet of Things*) vêm sendo uma tecnologia em grande crescimento no mundo, sendo uma peça importante da Indústria 4.0. O conceito de Internet das Coisas consiste da conexão de objetos do mundo físico ao digital, emergiu do avanço de áreas como sistemas embarcados, microeletrônica, comunicação e sensoriamento. Na IoT há três componentes necessários para implementação de um sistema, são eles dispositivos, redes de comunicação e sistemas de controle. Pensando nisso, este trabalho descreve uma abordagem prática de desenvolvimento de um produto eletrônico para Internet das coisas, voltando-se para o desenvolvimento de um módulo interruptor elétrico controlado remotamente.

O trabalho será dividido em uma revisão bibliográfica e tecnológica dos conceitos e tecnologias utilizadas, para familiarizar o leitor com o que foi utilizado no desenvolvimento deste projeto. O terceiro capítulo apresenta o planejamento do projeto, detalhando os objetivos e requisitos esperados para o mesmo. No quarto capítulo será apresentado como o projeto foi realizado e todas as suas etapas de desenvolvimento. No quinto capítulo será validado o projeto, testando o módulo e suas funcionalidades. Por fim uma conclusão do projeto desenvolvido com uma breve discussão do resultado final do projeto como um todo.

1.1 Objetivos

Têm-se como objetivos utilizar esse trabalho para classificar e executar várias etapas do desenvolvimento de um circuito eletrônico para Internet das coisas, esperando ao final deste, ter um protótipo funcional de um módulo interruptor IoT.

2 Revisão bibliográfica e tecnológica

Nesse capítulo será realizada uma introdução aos conceitos e tecnologias utilizadas no trabalho, buscando familiarizar o leitor ao que é tratado.

2.1 Internet das Coisas

Internet das Coisas é uma rede de objetos físicos dedicados (coisas) que contêm tecnologia incorporada para comunicar e interagir consigo mesmo ou com o ambiente externo, expandindo assim a capacidade de uso desses objetos e abrindo caminho para as mais diversas novas aplicações.

É uma área bastante ampla que tem comercialmente várias frentes de desenvolvimento. Aplicações industriais com a nova indústria 4.0, aplicações na agricultura e pecuária, automatização residencial e predial, aplicações médicas de monitoramento de sinais corporais, aplicações de infraestrutura para cidades inteligentes. É uma tecnologia que tem alto nível integração com outras tendências tecnológicas atuais.

O primeiro "dispositivo inteligente" conectado a rede foi uma máquina de vendas de Coca-Cola modificada no departamento de computação Carnegie Mellon University (COKE,). Kevin Ashton comentou, em junho de 2009 (ASHTON et al., 2009), que o termo Internet of Things foi primeiro utilizado em seu trabalho intitulado "I made at Procter & Gamble" em 1999. Na época, a IoT era associada ao uso da tecnologia RFID. Isso foi mudando conforme o tempo passava e o crescimento do termo IoT foi grande na última década, conforme o gráfico da Figura 1 que compara pesquisas sobre IoT e sobre Redes de sensores sem fio.

IoT é uma das tecnologias atuais em maior crescimento, está no topo do curva de "hype" das tecnologias emergentes da Gartner de 2018, como mostra a Figura 2 e é esperado que ultrapasse *smartphones* nos maiores consumidores de tráfego de internet no presente ano de 2018. (HAPNES, 2018).

O desenvolvimento dessa tecnologia emergente vem sendo relativamente lento analisado nível de aplicações que realmente mudem o modo de se fazer as coisas, no entanto vem sendo muito rápido quando observado o efeito coletivo dos milhões de dispositivos que são conectados em rede a cada dia.

Neste cenário, a possibilidade de novas aplicações é crescente, mas também a de novos desafios de conectar à Internet objetos com restrições de processamento, memória, comunicação e energia (LOUREIRO et al., 2003). Nesta situação os objetos divergem em implementação, recursos e qualidade. Surgem questões como, por exemplo, prover endere-

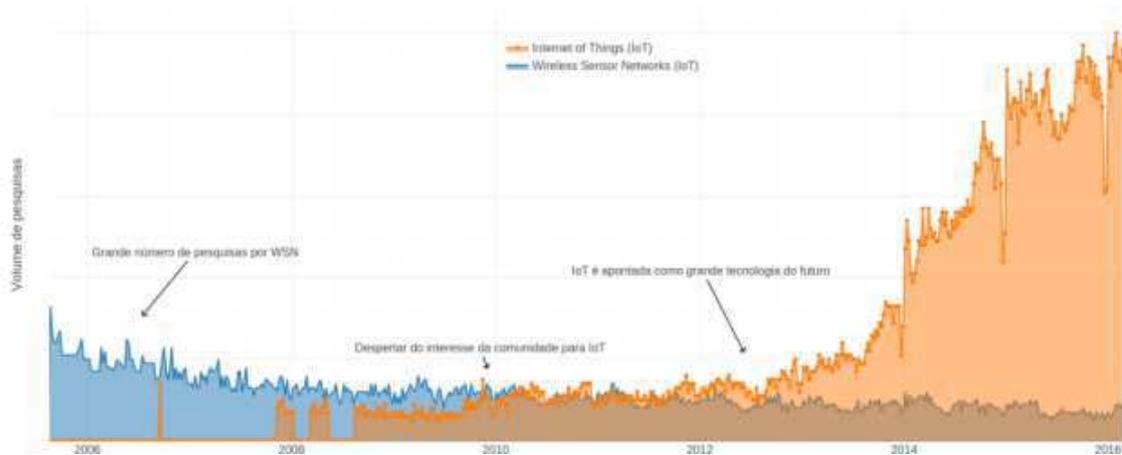


Figura 1 – Comparação entre pesquisas

Fonte: (SANTOS et al., 2016)

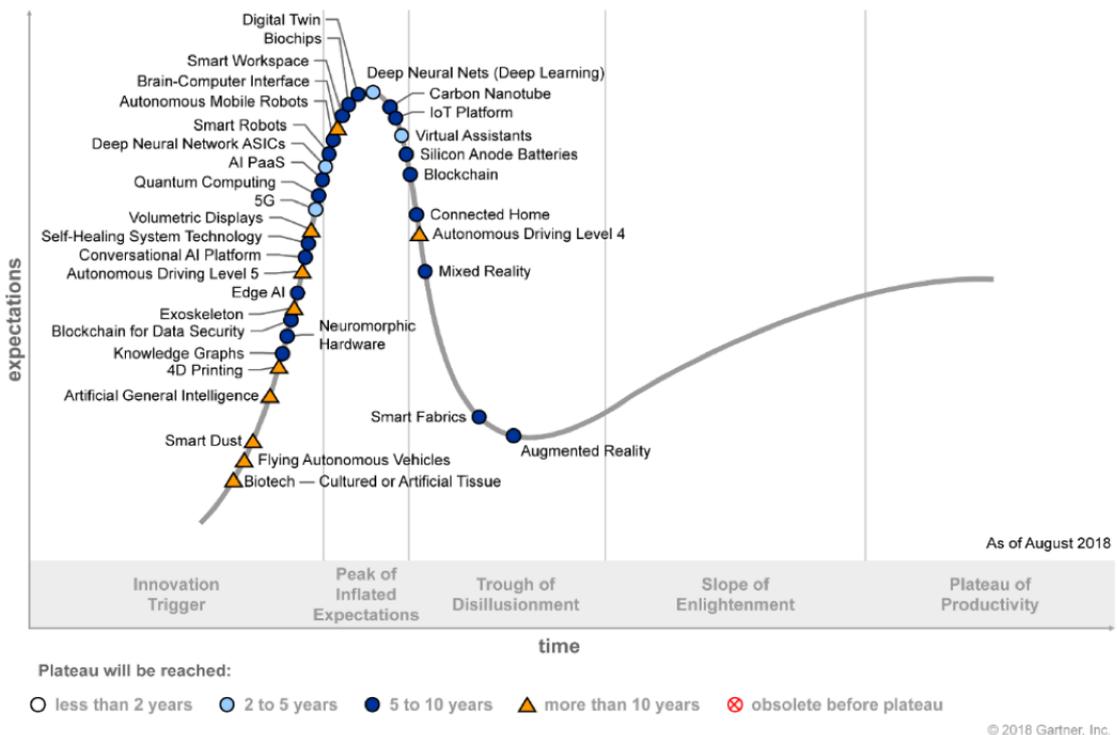


Figura 2 – Ciclo de "Hype" das tecnologias emergentes

Fonte: Garnet ¹

çamento aos dispositivos, encontrar rotas de boa vazão e que usem parcimoniosamente os recursos limitados dos objetos. Deste modo, fica evidente a necessidade da adaptação dos protocolos existentes. Além disso, sabe-se que os paradigmas de comunicação e roteamento nas redes de objetos inteligentes podem não seguir os mesmos padrões de uma rede como a Internet (CHAOUCHI, 2013).

2.1.1 Automação residencial e predial

Automação residencial ou domótica (Derivada do termo em francês *Domotique*, *Domus* que significa casa e *Imotique* significa automática (ADAMI, 2013)) é a utilização de dispositivos de IoT para monitorar e controlar os sistemas mecânicos, elétricos e eletrônicos.

Nesse contexto, três áreas principais estão sendo abordadas na literatura:

- A integração da Internet com os sistemas de gestão de energia dos edifícios, a fim de criar “edifícios inteligentes” eficientes em termos de energia e baseados em IoT.
- Os possíveis meios de monitoramento em tempo real para reduzir o consumo de energia e monitorar os comportamentos dos ocupantes.
- integração de dispositivos inteligentes no ambiente construído e como eles podem saber quem deve ser usado em aplicações futuras. (HAASE et al., 2016)

Automação residencial e IoT no geral sofre de fragmentação de plataforma e falta de padrões técnicos, uma situação em que a variedade de dispositivos de IoT, em termos de variações de hardware e diferenças no software executado neles, dificulta a tarefa de desenvolver aplicativos que funcionem consistentemente entre diferentes ecossistemas tecnológicos inconsistentes (BROWN, 2016).

2.1.2 Protocolos de comunicação

A comunicação entre dispositivos IoT pode ocorrer por meio de muitos métodos, por exemplo, *Bluetooth*², *Light-Fidelity (Li-Fi)*³, *Transport Layer Security*⁴, *WiFi*⁵, *Zig-Bee*⁶, *LTE-Advanced*⁷, *Low-power wide-area networking (LPWAN)*⁸ e *Very small aperture terminal (VSAT)*⁹. Neste trabalho escolheu-se utilizar de WiFi para realizar a comunicação mas mesmo definindo isso, ainda há a necessidade de escolher um método de transporte na camada de rede adequado a atividade que será realizada.

² <https://www.bluetooth.com/specifications>

³ <https://www.lifi.eng.ed.ac.uk>

⁴ <https://tools.ietf.org/html/rfc5246>

⁵ <https://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11-standards-tutorial.php>

⁶ <https://www.zigbee.org>

⁷ <http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced>

⁸ <https://tools.ietf.org/html/rfc8376>

⁹ http://www.comsys.co.uk/pdfs/p_v_r_i_r_o.pdf

2.1.2.1 HTTP - *Hypertext Transfer Protocol*

O HTTP é a base para a comunicação de dados da *World Wide Web*, onde os documentos de hipertexto incluem hiperlinks para outros recursos que o usuário pode acessar facilmente, por exemplo, por um clique do mouse ou tocando na tela. O HTTP foi desenvolvido para facilitar o hipertexto e a *World Wide Web*. No entanto, no contexto de IoT, HTTP não costuma ser muito utilizado por diversos fatores, entre eles (FIELDING et al., 1999):

- É um protocolo em que apenas um cliente e um servidor participam.
- Existe uma alta complexidade de cabeçalhos e regras, cujas mensagens trocadas entre cliente e servidor geralmente necessitam de uma rede com poucas restrições de velocidade.
- O protocolo necessita da solicitação do cliente ao servidor, formando assim um caminho unidirecional.
- O cliente ao solicitar um serviço, espera até o servidor responder existindo assim uma sincronia envolvida entre a troca de mensagens.

Observando essas características para comunicação utilizando HTTP é possível entender a necessidade de um protocolo em que as mensagens sejam transmitidas a vários dispositivos ao mesmo tempo de modo simples, cujas mensagens sejam mensagens curtas de modo que não precise de uma rede de alta velocidade para o funcionamento adequado da transmissão. É necessário também uma comunicação bidirecional e assíncrona, de modo que cliente e servidor possam se comunicar um com o outro para envio e requisições de dados a qualquer momento.

2.1.2.2 MQTT - *Message Queuing Telemetry Transport*

As características essenciais de protocolo de comunicação para um dispositivo IoT como o desenvolvido neste trabalho são saciadas na utilização do protocolo MQTT (ANDREW, 2018). Ele foi desenvolvido para dispositivos extremamente limitados e faz uso de *publish/subscribe* para transferir mensagens. O MQTT provê mecanismos para a garantia de entrega de mensagens e seu principal objetivo é minimizar o uso de largura de banda da rede e recursos dos dispositivos. Além disso, MQTT utiliza os protocolos das camadas de transporte e rede da arquitetura TCP/IP.

O MQTT consiste de três componentes básicos: o *subscriber*, o *publisher* e o *broker*. Inicialmente dispositivos se registram (*subscribe*) a um *broker* para obter informações sobre dados específicos, para que o *broker* os avise sempre que publicadores (*publishers*) publicarem os dados de interesse. Os dispositivos inteligentes (*publishers*) transmitem

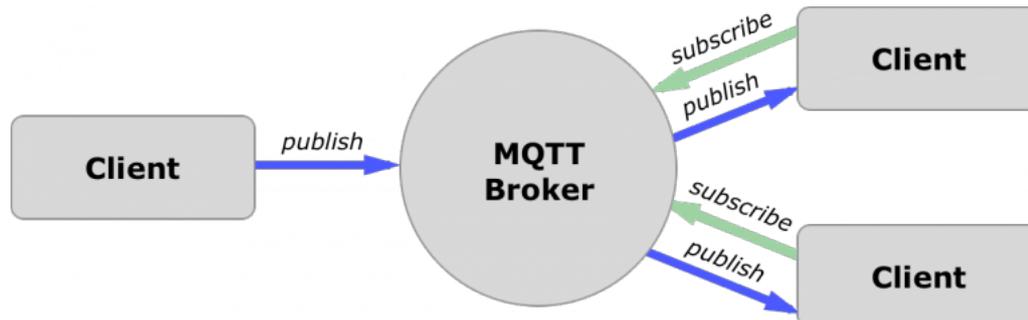


Figura 3 – Diagrama de funcionamento para o protocolo MQTT

Fonte: One sheeld¹⁰

informações para os *subscriber* através do *broker*. Desta forma toda informação passa pelo *broker*, que serve de intermediário e gerenciador das mensagens.

2.2 Microsoft Azure

Para implementação de *broker* da aplicação desenvolvida neste trabalho foi decidido utilizar de um *broker* em nuvem, pela não necessidade de um novo dispositivo para realizar o papel de *broker* e por não necessitar que o *publisher* e o *subscriber* estejam numa mesma rede para realizar a comunicação. Além de que, considerando o dispositivo como um produto comercial com milhares do mesmo dispositivo divididos entre múltiplos usuários faz-se necessário uma plataforma robusta e bem estruturada para realizar o gerenciamento da comunicação entre os dispositivos.

Foi escolhido então a plataforma Azure da Microsoft, ela é uma plataforma destinada a serviços na nuvem utilizada para serviços de host, gerenciamento e desenvolvimento de dados, possuindo uma vasta rede global. Azure dispõe de uma grande quantidade de serviços, onde é possível destacar quatro principais:

- *Compute services*: Responsável por executar aplicativos na nuvem, sendo composto por aplicativos como *Azure Virtual Machines*, *App Services*, etc.
- *Storage services*: Responsável por armazenar dados estruturados na nuvem, sendo composto por aplicativos como *Azure Storage*, *Azure SQL Database*, etc.
- *Application services*: Responsável por construção e operação de aplicativos, sendo composto por aplicativos como *Azure Active Directory*, *Azure Scheduler*, etc.
- *Network services*: Responsável por serviços de redes e tráficos entre *datacenters* do Azure, sendo composto por aplicativos como *Azure Virtual Network*, *Azure ExpressRoute*, etc.

Além disso, por oferecer uma versão trial com diversos serviços liberados gratuitamente por um ano, a Azure se apresenta como uma plataforma interessante para desenvolvimento de aplicações IoT.

Azure IoT Hub vai ser o recurso utilizado nesse trabalho, ele é um serviço gerenciado e hospedado na nuvem que atua como um hub central de mensagens para a comunicação bidirecional entre o aplicativo de IoT e os dispositivos que ele gerencia. É usado para criar soluções de IoT com comunicação segura e confiável entre milhões de dispositivos de IoT e um *back-end* de solução de nuvem hospedado com possibilidade de conectar praticamente qualquer dispositivo ao Hub IoT.

2.3 Microcontrolador Esp-32

Para desenvolvimento de um dispositivo IoT é necessário escolher uma plataforma adequada e nas aplicações de IoT recentes, o chip que vem se destacando bastante é o Esp-32. É uma melhoria no seu predecessor, o ESP8266, tem uma grande capacidade de processamento e é de custo baixo, sendo fácil de se encontrar um kit de desenvolvimento, como o da foto na Figura 4, por menos de 5 Dólares, o que o torna uma opção viável para praticamente toda aplicação IoT, desde as mais complexas como também as mais simples, facilitando a vida de um desenvolvedor cheio de ideias e com pouco tempo de aprender os pormenores de vários microcontroladores. O Esp-32 é projetado pela *Espressif Systems* que é uma empresa chinesa sediada em Xangai. O Esp-32 anuncia-se como uma ponte entre os microcontroladores existentes para o WiFi e por também ser capaz de executar aplicativos contidos dentro dele(KOLBAN, 2018).

As especificações dele são as seguintes:

- CPU: Xtensa® Dual-Core 32-bit LX6
- ROM: 448 KBytes
- RAM: 520 Kbytes
- Flash: 4 MB
- Clock máximo: 240MHz
- Wireless padrão 802.11 b/g/n
- Conexão Wifi 2.4Ghz (máximo de 150 Mbps)
- Antena embutida
- Conector micro-usb



Figura 4 – Placa de Desenvolvimento para o esp32

Fonte: AliExpress¹¹

- Wi-Fi Direct (P2P), P2P Discovery, P2P Group Owner mode e P2P Power Management
- Modos de operação: STA/AP/STA+AP
- Bluetooth BLE 4.2
- Portas GPIO: 11
- GPIO com funções de PWM, I2C, SPI, etc
- Tensão de operação: 4,5 – 9V
- Taxa de transferência: 110-460800bps
- Suporta Upgrade remoto de firmware
- Conversor analógico digital (ADC)
- Distância entre pinos: 2,54mm
- Dimensões: 49 x 25,5 x 7 mm

3 Proposta de solução

A proposta inicial foi desenvolver um circuito IoT que possa ser "produtizado", e a partir disto que os requisitos do projeto e as etapas de desenvolvimento foram planejadas. Considerando isto, é definido um plano para o modelo de desenvolvimento de dispositivos compatível com um projeto profissional, como os realizados na prática na indústria. Para isso utilizou-se de tutoriais práticos de desenvolvimento de circuitos comerciais como os de (TEEL, 2018) para formular este planejamento. O dispositivo a ser desenvolvido é um módulo interruptor IoT, ele será capaz de interromper um ponto de energia elétrica doméstica, seja ele uma tomada ou lampada através de comandos advindos do usuário através de um aplicativo.

3.1 Requisitos de projeto

Os requisitos de projeto definirão os caminhos a serem seguidos, eles devem ser únicos, inequívocos e verificáveis. Os testes para verificação e validação desses requerimentos serão apresentados no Capítulo 5.

Os requisitos serão diferenciados em requisitos funcionais e não-funcionais, sendo os funcionais os requisitos que definem atividades o dispositivo deve realizar e não funcionais os requisitos objetivos de características delimitantes do projeto.

3.1.1 Requisitos funcionais

Baseando-se nas ideias iniciais das funcionalidades que o dispositivo deveria cumprir, bem como no estudo da tecnologia IoT e nas capacidades do microcontrolador, foram definidos os seguintes requisitos funcionais:

- O sistema deve realizar a função de interruptor ao receber um comando.
- O sistema deve se conectar a Internet por WiFi.
- O sistema deve obter as credenciais de WiFi do usuário.
- O sistema deve ser de fácil montagem numa rede elétrica doméstica já existente.
- Um aplicativo deve enviar o comando ao dispositivo.

3.1.2 Requisitos não-funcionais

Os requisitos não funcionais são requisitos pensados nas limitações financeiras, limitações de tempo, segurança e qualidade de projeto para que fosse criar um projeto viável para este trabalho.

- O circuito deve estar uma placa de circuito impresso(PCB- *Printed Circuit Board*) projetada.
- O sistema deve ser colocado em um *case* plástico.
- As dimensões da placa devem ser menor que 10x7 cm.
- O sistema não deve conter bateria.
- O custo de produção do protótipo não deve ultrapassar 100 reais.

3.2 Modelo proposto

O modelo proposto, apresentado no diagrama da Figura 5, evidencia em módulos as partes do circuito que serão necessárias ser definidas. Cada decisão de projeto foi tomada de modo a atender cada um dos requisitos.

Para o sistema não conter bateria é necessário que haja um modo de retificar e reduzir a tensão da rede elétrica para alimentar o microcontrolador, portanto decidiu-se colocar uma micro-fonte acoplada no circuito para alimentar o microcontrolador e o circuito de acionamento. A fonte utilizada é Micro-fonte 5V TSP-05¹.

Como é necessário que haja conexão com a Internet, o microcontrolador escolhido deve ser capaz de se comunicar por WiFi e para receber os dados das credenciais do roteador, deve ser capaz de funcionar como ponto de acesso(AP - Access Point) WiFi, características que são facilmente executadas pelo Esp32.

Para causar a abertura ou fechamento da chave foi escolhido utilizar um relé com um circuito de acionamento simples. O circuito de acionamento projetado é apresentado na captura de tela da Figura 8, que mostrará mais adiante o circuito simulado em software.

Será utilizado o relé Songle SRD-5VDC-SL-C² que requer uma corrente na bobina de 71mA e utilizando o Transistor bipolar NPN BC337³ que tem um β de 250 faz necessário um resistor de 10k para o acionamento do relé.

Para a fácil montagem na rede elétrica doméstica é utilizado terminais borne para encaixe dos fios que entrarão no circuito.

¹ <https://img.filipeflop.com/files/download/Datasheet-HLK-PM01.pdf>

² <http://www.circuitbasics.com/wp-content/uploads/2015/11/SRD-05VDC-SL-C-Datasheet.pdf>

³ <https://www.onsemi.com/pub/Collateral/BC337-D.PDF>

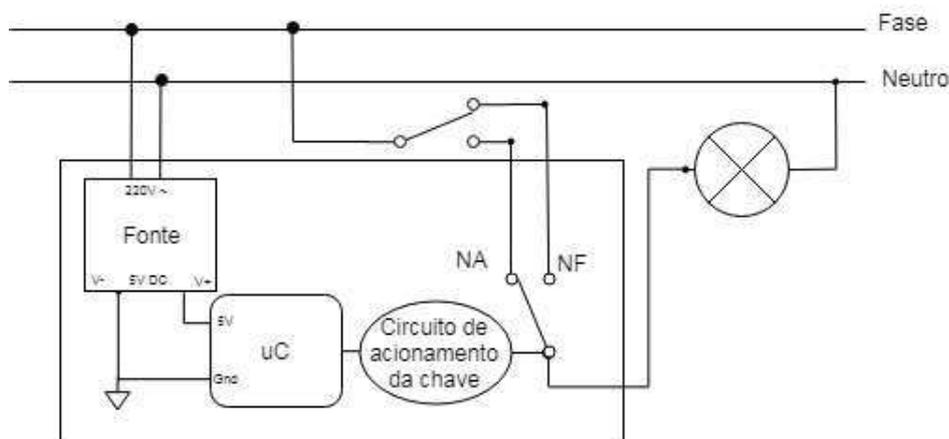


Figura 5 – Diagrama do modelo para o circuito do Interruptor IoT



Figura 6 – Diagrama de comunicação do sistema

3.2.1 Integração do projeto

Definido o circuito de acionamento e microcontrolador utilizado, pode-se definir exatamente o modelo de comunicação que deve ser executada. O modelo proposto, como apresentado no diagrama da Figura 6, apresenta o caminho do comando de acionamento do sistema, advindo do aplicativo Android, sendo tratado pela *Azure Iot Hub* e enviada ao módulo interruptor para que haja o chaveamento.

3.3 Planejamento do Projeto

O desenvolvimento de um projeto eletrônico se dá a partir de várias etapas de design, prototipação, testes, validação e aprimoramento até se ter um projeto que atinja uma gama completa de requisitos para torná-lo comercial, seja para manufatura em larga escala, seja para um projeto sob encomenda.

Este trabalho busca executar essa abordagem de desenvolvimento, no entanto por limitações de tempo, dinheiro e recursos, não se espera executar o protótipo final do circuito, apenas uma primeira versão para testes.

A Figura 7 apresenta um fluxograma de projeto adaptado para o propósito deste trabalho e a partir deste se elaborou o plano para o desenvolvimento do Interruptor IoT.

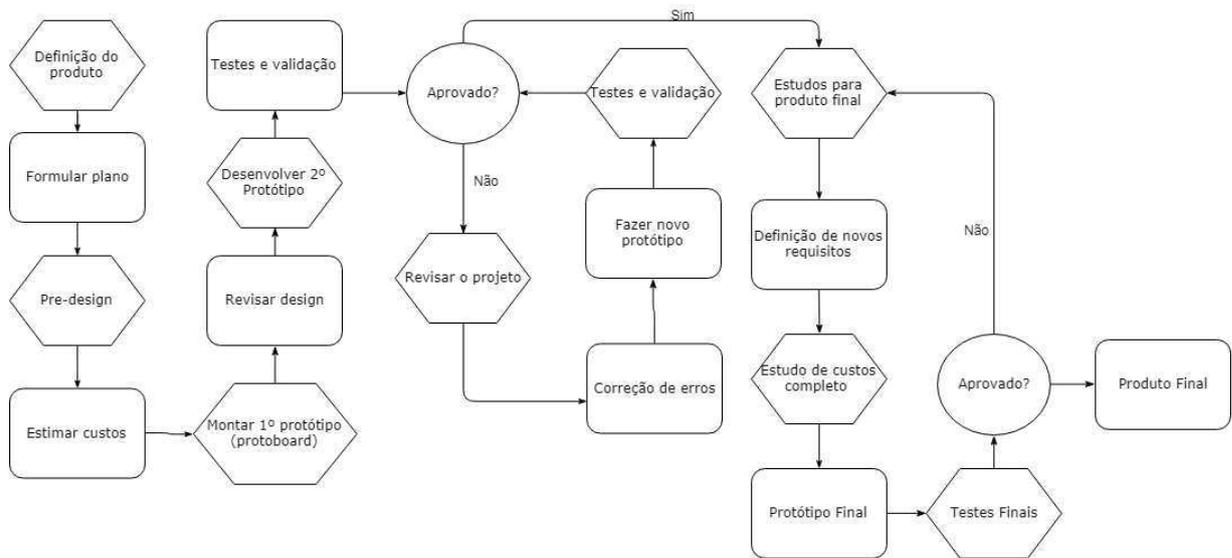


Figura 7 – Fluxograma com uma metodologia para desenvolvimento de projetos eletrônicos

1. Definição do projeto: Definir funcionalidades que quer que o projeto realize;
2. Definição dos requisitos: Definição de quais exigências o projeto deve atender;
3. Pré-design: projeto do circuito em módulos e interconexões sem desenvolver cada um deles;
4. Plano de desenvolvimento: Dividir e organizar o processo de desenvolvimento em etapas;
5. Familiarização com o Hardware e Software utilizados: Estudos de como programar o Esp-32 e realizar comunicação via AZURE;
6. Simulação *SPICE*: Desenhar o esquemático do circuito em software de simulação *SPICE*;
7. Protótipo "*proof of concept*": Montar circuito na *protoboard*;
8. Programar: Início do processo de programação do microcontrolador;
9. PCB: Projeto da primeira placa de circuito impresso(PCB);
10. *Design Check* da placa PCB: Revisar design da placa com outros profissionais da área;
11. Protótipo para testes: Montar circuito na placa PCB;
12. Comunicação broker-device: Realizar comunicação a partir do portal Azure;

13. Comunicação Client-broker device: Desenvolvimento do aplicativo para comunicação remota;
14. Validação: Análises e testes do projeto;

4 Desenvolvimento do projeto

Neste capítulo serão descritas as várias etapas do desenvolvimento do projeto, apresentando detalhes de funcionamento e as ferramentas utilizadas durante o processo. Foi escolhido dividir em duas partes: desenvolvimento do Hardware e desenvolvimento do Software, que são as duas frentes do trabalho que embora o produto final dependa de ambas, o desenvolvimento delas pôde ocorrer de forma bastante independente.

4.1 Desenvolvimento do Hardware

O desenvolvimento do Hardware ocorreu de modo com que cada etapa do projeto aumentasse a garantia de funcionamento do projeto final, iniciando com a simulação do circuito em software, passando para o teste em *protoboard*, seguido do circuito na placa de circuito impresso, pôde-se ter grande garantia do funcionamento no último protótipo.

4.1.1 Simulação do circuito

Utilizando o *software* para simulações Multisim da National Instruments¹, foi projetado o circuito de acionamento conforme proposto. Com uma alimentação de 5V para o relé e um acionamento das portas digitais do microcontrolador em 3.3V pode-se testar o circuito e comprovar seu funcionamento. A captura de tela do software na Figura 8 mostra a simulação para acionamento de uma lâmpada, onde o componente EDR201A05 na Figura representa o relé utilizado.

¹ <http://www.ni.com/pt-br/shop/electronic-test-instrumentation/application-software-for-electronic-test-and-instrumentation-category/what-is-multisim.html>

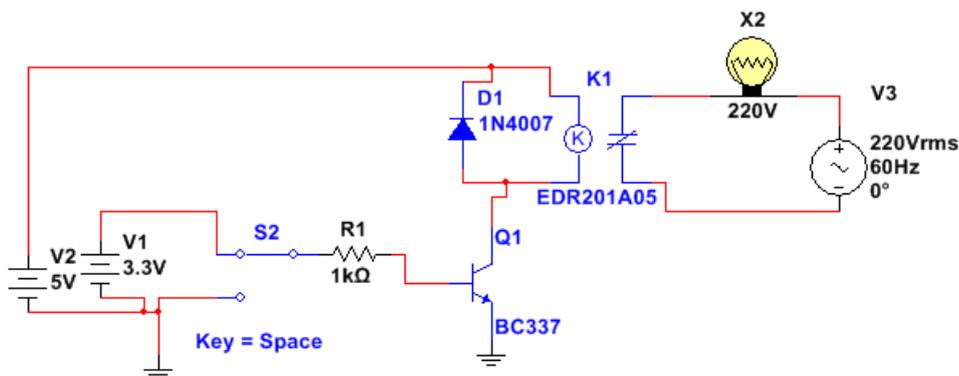


Figura 8 – Captura de tela de simulação realizada no *Multisim*

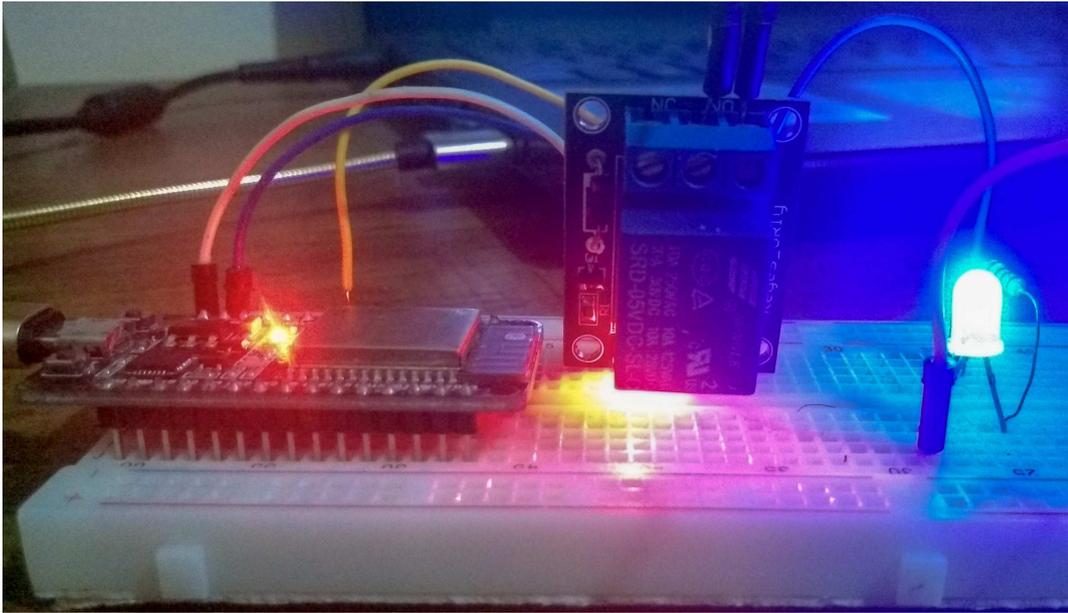


Figura 9 – Fotografia de teste realizado na *protoboard*

4.1.2 Prototipação na *protoboard*

O primeiro teste físico foi realizado na *protoboard* utilizando um módulo relé com o mesmo circuito de acionamento que o utilizado no projeto. Ele sendo acionado pelo pino digital do Esp-32 e alimentado com os 5V da porta USB que também alimenta o Esp-32. O circuito funcionou como esperado, sendo utilizado para ligar e desligar um LED nos terminais do relé como apresentado na foto da Figura 9.

4.1.3 Design da placa PCB

Com a segurança de que o circuito projetado funciona, pôde-se realizar o projeto da placa PCB. Para esta tarefa, o software online *EasyEDA*² foi escolhido para realizar a fazer o projeto. O esquemático do circuito foi montado novamente em sua interface, desta vez podendo escolher todos os componentes utilizados. Foram então selecionados os *footprints* adequados para cada componente e realizada a interligação entre eles conforme captura de tela na Figura 10.

Pode-se então ir para a interface de projeto de PCBs, como mostrado na captura de tela da Figura 11, posicionou-se todos os componentes em posição adequada fazendo a ligação entre eles, atentando-se a largura da trilha e dos *pads* para que não houvesse problemas de curto circuito, nem dificuldades na hora de soldar os componentes. Como o circuito era simples, utilizou-se uma placa de cobre de camada única.

² <https://easyeda.com>

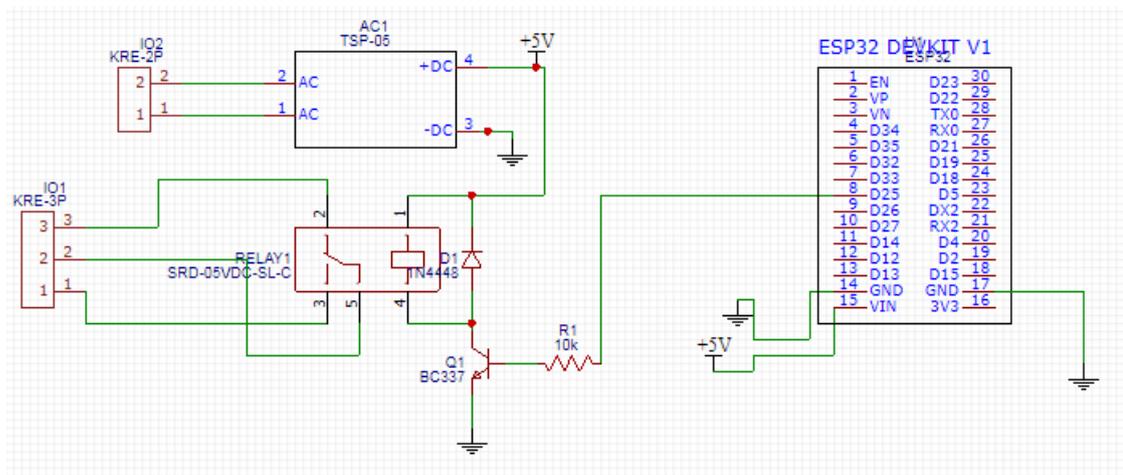


Figura 10 – Captura de tela do esquemático do Circuito no software online *EasyEDA*

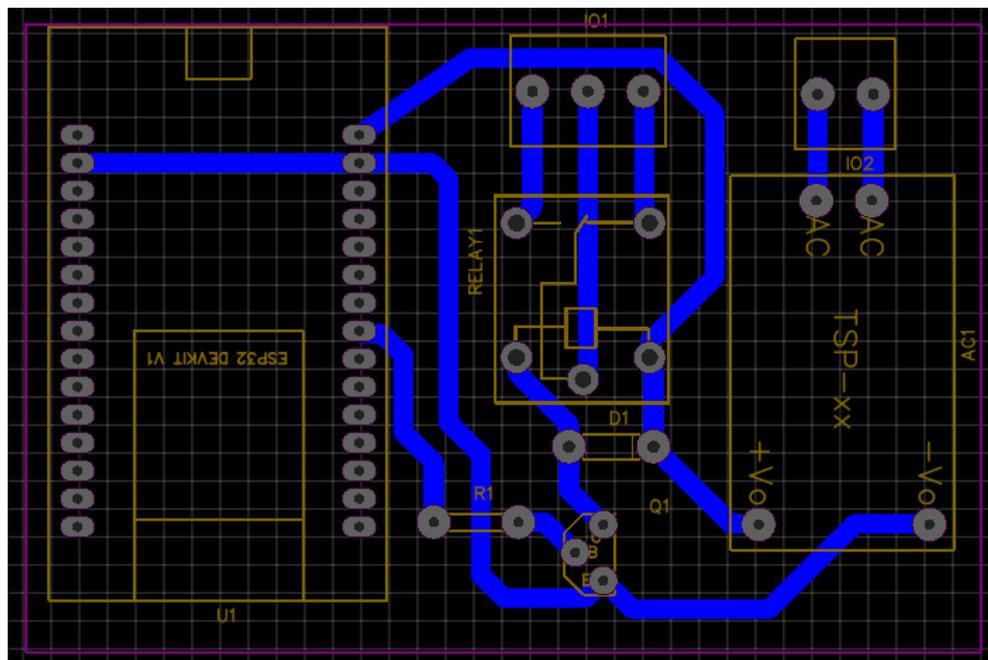


Figura 11 – Captura de tela da placa de circuito impresso projetada no *EasyEDA*

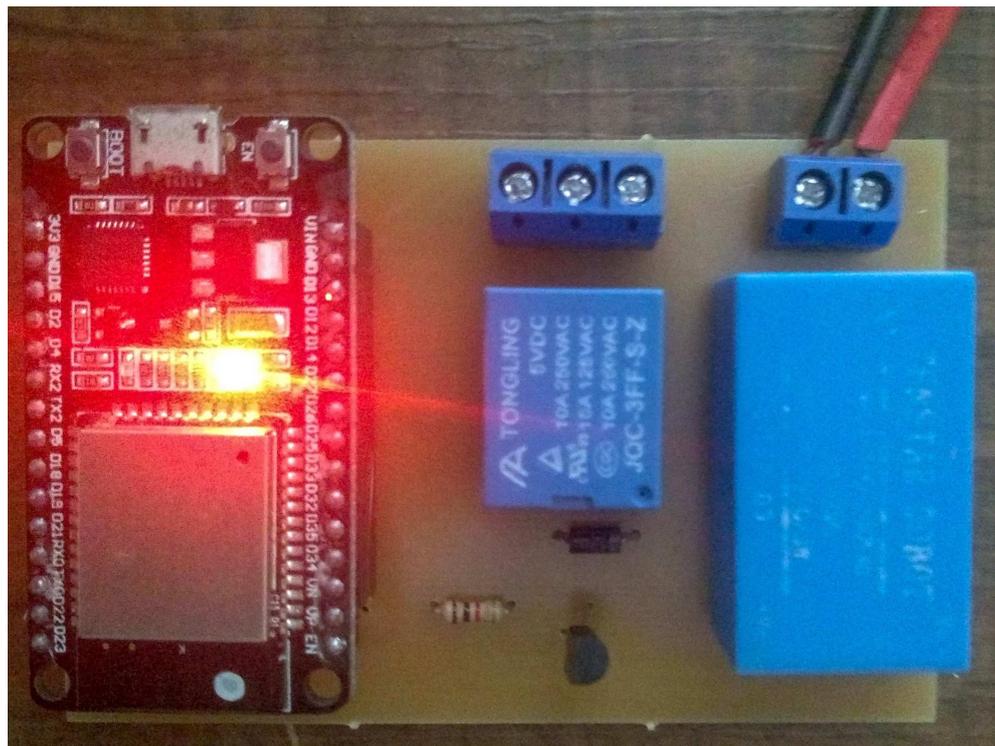


Figura 12 – Fotografia da placa de circuito impresso projetada

4.1.4 Protótipo em PCB

Após a impressão, os componentes foram soldados na placa, obtendo o circuito exposto na fotografia na Figura 12 como resultado. Foram realizados testes de continuidade e descontinuidade em cada trilha e terminal antes de ligar o circuito.

Por trabalhar com tensão alternada da rede, não é seguro manusear o circuito com ele em funcionamento, bem como uma placa PCB seria de difícil instalação em uma tomada ou interruptor na parede. Pensando nisso resolveu-se selecionar uma caixa plástica para acomodação do circuito e fácil instalação na parede. Após projeto da placa foi possível definir a caixa plástica adequada para envolver o circuito. Utilizando as dimensões da circuito que é de 86 mm x 57 mm x 19 mm, escolheu-se a caixa PB-107 que tem dimensões suficientes para o projeto e dispõe de abas para fixação como pode ser visto na Figura 13.

No entanto, para demonstração foi colocado o circuito junto ao módulo interruptor Iot numa caixa fechada conforme a Figura 14.

4.1.5 Lista de Componentes

A partir da completude do projeto a nível de Hardware, se tem uma lista de materiais completa para o circuito. A tabela 1 define os componentes utilizados nesse protótipo bem como o preço e quantidade utilizada de cada um deles.

Custando um total de R\$48,64 para compra dos componentes do circuito.

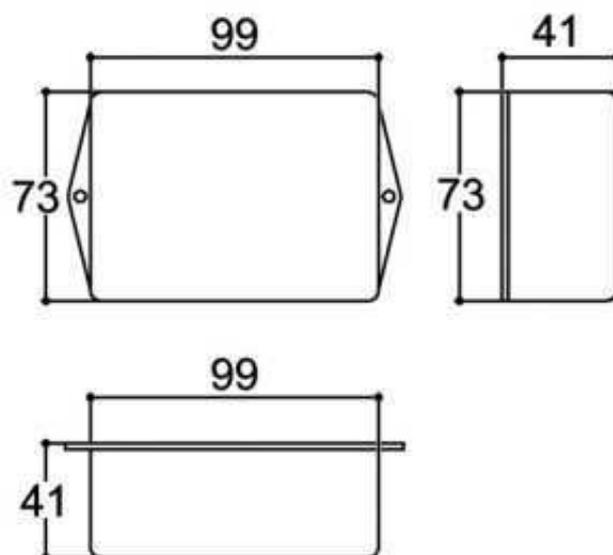


Figura 13 – Vistas da caixa PB-107 utilizada para acomodar o circuito

Fonte: Patola

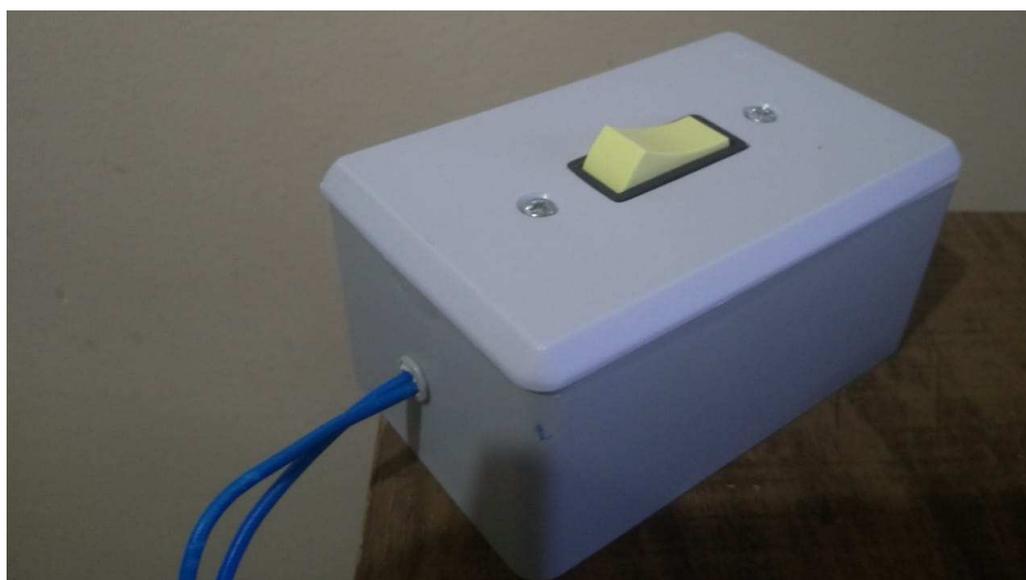


Figura 14 – Fotografia do projeto dentro de uma caixa plástica com um interruptor

Tabela 1 – Componentes utilizados

Componente	Quantidade	Preço unitário(R\$)
Esp32 Dev Kit	1	21,47
Relé SRD-03VDC-SL-C	1	3,00
Diodo 1N4448	1	0,30
Resistor 10k	1	0,02
Transistor NPN BC337	1	0,35
Borne de conexão KRE-3P	1	0,80
Borne de conexão KRE-2P	1	0,65
Micro-fonte 5V TSP-05	1	12,00
Placa revestida de cobre - Único lado	1	2,05
Caixa plástica PB-107	1	8,00

4.2 Desenvolvimento do Software

Entende-se aqui como desenvolvimento do software a programação do Esp-32, bem como a configuração do sistema de comunicação e do aplicativo de acionamento, sendo apresentadas a seguir cada uma destas etapas. Para programação do Esp-32 decidiu-se utilizar a IDE do Arduino, por facilitar o download de bibliotecas e integrar a compilação, gravação e monitoramento serial do microcontrolador em um só programa.

4.2.1 Conexão WiFi

A primeira tarefa a ser realizada é conectar o Esp-32 à Internet via WiFi, a funcionalidade de conexão sem fio foi realizada utilizando a biblioteca WiFi fornecida pelo microcontrolador. São definidas variáveis do tipo String para as credenciais e utilizadas como parâmetro para a função `Wifi.begin` realizar a conexão com o roteador.

É possível inserir diretamente no código as credenciais pessoais do WiFi, mas como o projeto foi pensado de modo a aproximar ao máximo o dispositivo a um produto, se faz necessário um modo de o usuário inserir suas credenciais e o dispositivo se "lembrar" dessas credenciais mesmo se reiniciado. Portanto, foi utilizado a memória não volátil do microcontrolador para salvar as credenciais de nome de rede e senha, estas serão adquiridas fazendo o microcontrolador operar no modo *Acess-Point(AP)* e gerar uma página WEB com um formulário onde o usuário, conectado com seu *smartphone* ou computador, pode colocar as credenciais de sua rede WiFi, bastando para isso acessar o endereço padrão do dispositivo é "http://192.168.4.1/". A Figura 15 mostra uma captura de tela da interface da página.

4.2.2 Conexão com o Azure IoT Hub

Como foi previamente dito, a plataforma Microsoft Azure é uma plataforma com diversas funcionalidades para soluções em nuvem, dentre elas está a Azure IoT Hub, que



A screenshot of a web interface for configuring a WiFi connection. At the top, there is a text input field containing the IP address "192.168.4.1/". To the right of this field are a circular refresh icon and a vertical ellipsis menu icon. Below the IP field, the text "Escolha por favor a rede wifi a ser conectada" is displayed. Underneath, there are two input fields: "Rede Wifi:" followed by a field containing "ssid", and "Senha:" followed by a field containing "password". At the bottom of the form is a "Submit" button.

Figura 15 – Captura de tela da página para inserir credenciais da rede WiFi a ser conectada.

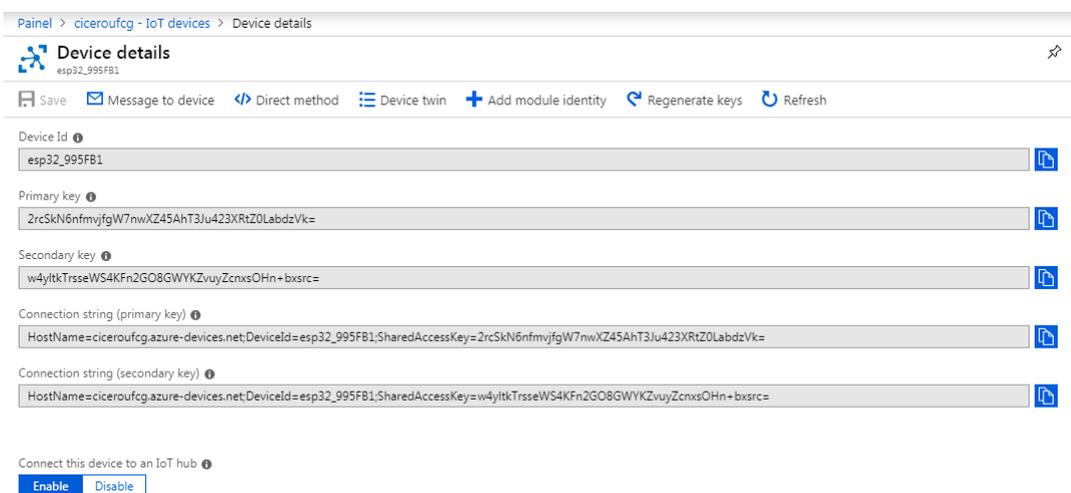


Figura 16 – Captura de tela do portal do Azure com a página do dispositivo criado

é uma plataforma que oferece suporte para gerenciamento de dispositivos IoT. Nela é possível enviar, receber e tratar dados de muitos dispositivos facilmente. Após criar uma conta Azure, foi necessário criar um grupo de recursos na plataforma IoT Hub, para o caso deste trabalho o grupo de recursos foi chamado "ciceroufcg" e neste grupo de recursos foi adicionado um dispositivo na plataforma, como é exibido na captura de tela na Figura 16.

Cada dispositivo adicionado deve ter um "*DeviceID*", que foi especificado como o mesmo ID do ESP-32. Com isso foi possível fazer a comunicação direta entre a plataforma e o ESP-32. Deste modo se pode desenvolver o código do Esp-32 para receber mensagens da Hub e lidar com elas da forma correta.

Para a conexão como *Client* na plataforma é necessário que seja coletado na página do dispositivo criado na *Hub* a *ConnectionString*, que é o texto que identifica o dispositivo dentro da rede.

Foi criado no Esp-32 um *"Handler"* para o método de comunicação direta implementado pela Hub, chamado *"Devicemethodcallback"*. Esse método direto, quando acionado, envia seu nome e uma *payload* com dados a serem recebidos pelo dispositivo. Na solução proposta, utilizando um método direto chamado "toggle" será tratado no Esp-32 como o comando de acionamento do relé, este inverterá o estado do pino digital sempre que o método for chamado, como pode ser visto no pedaço de código a seguir:

```
static int DeviceMethodCallback(const char *methodName, const unsigned
    char *payload, int size, unsigned char **response, int *response_size)
{
    LogInfo("Tentando invocar metodo %s", methodName);
    const char *responseMessage = "\Sucesso ao invocar o metodo\";
    int result = 200;

    if (strcmp(methodName, "toggle") == 0)
    {
        digitalWrite(relay, !digitalRead(relay));
    }

    else
    {
        LogInfo("Metodo %s nao encontrado", methodName);
        responseMessage = "\Metodo nao encontrado\";
        result = 404;
    }

    *response_size = strlen(responseMessage) + 1;
    *response = (unsigned char *)strdup(responseMessage);

    return result;
}
```

Estabelecendo assim a comunicação entre o dispositivo e a plataforma na nuvem, falta apenas estabelecer um modo do usuário enviar o comando de acionamento para o interruptor sem que seja necessário acessar o portal do Azure.

4.2.3 Aplicativo para acionamento

Pensando nisso decidiu-se desenvolver também um aplicativo que realizasse facilmente a função de acionar o interruptor IoT. Com a ferramenta de software Android Studio e os tutoriais e exemplos presentes na plataforma azure foi possível o criar um aplicativo que realiza a chamada do método direto para o dispositivo no apertar de um botão, como é apresentado na captura de tela do aplicativo na Figura 17. O aplicativo



Figura 17 – Captura de tela do aplicativo criado para acionamento do módulo interruptor IoT

envia apenas para o dispositivo cadastrado na plataforma.

5 Validação e discussão dos resultados

Neste capítulo está presente uma série de testes realizados ao projeto, testando o cumprimento dos requisitos, tanto de hardware como de software, buscando testar todos os aspectos de funcionamento e encontrar possíveis falhas de funcionamento. Logo após será realizada uma discussão sobre o projeto, analisando as dificuldades encontradas durante o processo de desenvolvimento.

5.1 Testes de validação

Testes de validação servem para garantir o funcionamento da solução projetada, para encontrar erros que impediriam o prosseguimento do projeto comprovar o cumprimento dos requisitos especificados no início do projeto. Serão utilizado uma série de tabelas que identificarão o cenário de teste, pre-condições, resultado esperado e resultado obtido no teste realizado.

Os testes realizados foram definidos de forma a comprovar o cumprimento de todos os requisitos bem como buscar falhas presentes. Alguns dos testes descritos a seguir foram realizados durante o desenvolvimento, no entanto a maioria foi realizada ao finalizar o protótipo.

Tabela 2 – Teste de alimentação

Cenário de teste	Teste de alimentação
Pré-Condição	Micro-fonte utilizada ligada na rede elétrica
Ação executada	Medir tensão de saída da fonte
Resultado esperado	Medir 5V DC estável na saída da fonte
Resultado obtido	5V DC estável medido na saída da fonte

Tabela 3 – Teste de acionamento do relé

Cenário de teste	Teste de acionamento do relé com 3.3V
Pré-Condição	Circuito de acionamento montado na protoboard
Ação executada	Colocar e retirar 3.3V na entrada do circuito e observar o acionamento do relé
Resultado esperado	Relé acionar quando tiver 3.3V na entrada e voltar ao estado não acionado quando for retirado os 3.3V
Resultado obtido	Relé aciona quando tem 3.3V na entrada e deixa de estar acionado quando é retirado os 3.3V da entrada

Tabela 4 – Teste de conexão

Cenário de teste	Teste de conexão WiFi
Pré-Condição	Esp-32 com credenciais da rede WiFi em seu código
Ação executada	Ligar o Esp-32 e observar se houve conexão a partir do Serial monitor.
Resultado esperado	Esp-32 se conectar e imprimir seu IP local no serial monitor.
Resultado obtido	Esp-32 se conectou e imprimiu seu IP local no serial monitor.

Tabela 5 – Teste de recebimento de credenciais

Cenário de teste	Teste de Recebimento de credenciais
Pré-Condição	Esp-32 sem nenhuma credencial de rede WiFi salva
Ação executada	Conectar-se ao AP do Esp-32, acessar a página gerada e colocar as credenciais
Resultado esperado	Credenciais submetida sendo impressas no Serial monitor.
Resultado obtido	Credenciais submetidas recebidas e impressas no Serial monitor pelo Esp-32.

Tabela 6 – Teste de desligar a energia.

Cenário de teste	Teste de desligar a energia.
Pré-Condição	Esp ser desligado tendo já recebido as credenciais.
Ação executada	Apertar botão Reset.
Resultado esperado	Após ser desligado o Esp-32 deve reter as informações de login e se conectar a rede
Resultado obtido	O Esp se conectou a rede.

Tabela 7 – Teste de interrupção

Cenário de teste	Teste da interrupção
Pré-Condição	Sistema completo montado com uma lâmpada para ser ligada e desligada
Ação executada	Apertar o botão no aplicativo
Resultado esperado	Lâmpada acender ou apagar cada vez que o botão for acionado
Resultado obtido	Lâmpada acender ou apagar a cada apertar do botão mas com um atraso de alguns segundos

Tabela 8 – Teste de desconexão

Cenário de teste	Teste da desconexão
Pré-Condição	Sistema funcionando
Ação executada	Reiniciar o roteador ou afastar o dispositivo e aproximar novamente.
Resultado esperado	Conexão perdida e depois reestabelecida.
Resultado obtido	Conexão perdida e depois reestabelecida.

5.2 Discussão de resultados

A partir dos testes e de todo o processo de desenvolvimento foi possível obter um circuito funcional integrado a um software que atente as necessidades de comunicação requeridas pelo projeto. Atentando ao teste de interrupção, vê-se que o atraso entre o apertar do botão no aplicativo e o acionamento do relé pode não ser algo muito conveniente, no entanto não chega a atrapalhar a utilização do dispositivo. O atraso vai ocorrer de todo

modo pois o comando passa por varias etapas até chegar ao microcontrolador, mesmo com uma internet mais rápida.

No teste de desconexão houve a descoberta de uma característica não esperada. Na primeira vez em que o teste foi realizado, esperava-se que não ocorresse a reconexão e fossem necessárias modificações no código para implementar essa funcionalidade, no entanto a própria função de conexão realiza a reconexão quando é possível.

Em relação a parte elétrica houve sucesso no projeto idealizado, obtendo um circuito de tamanho adequado para a aplicação mas ainda com o circuito bastante otimizável, onde Componentes Para Montagem em Superfície (SMD - *Surface Mounting Device*) poderiam ser usados e o microcontrolador puro ser integrado ao projeto em vez de seu kit de desenvolvimento.

6 Conclusões

Este projeto de conclusão de curso proporcionou a oportunidade de lidar com varias ferramentas e tecnologias que o concluinte não estava familiarizado, o que engrandeceu seu conhecimento e o deu mais experiência em desenvolvimentos de projetos. O desafio de seguir o planejamento, do projeto tendo que lidar com variados contratempos durante o desenvolvimento, bem como dificuldades normais encontradas ao trabalhar com algo novo, obrigou o concluinte a diminuir algumas das expectativas iniciais para o projeto.

Apesar disso, o interruptor IoT foi bem sucedido em seu objetivo, cumprindo todos os seus requisitos e realizando tudo de mais importante que se esperava realizar. De fato, não deu para continuar o aperfeiçoamento do projeto ainda neste trabalho, onde seria realizada uma nova definição de requisitos para que um novo protótipo mais competitivo com o que é feito na industria fosse produzido. Mas serve de proposta para trabalhos futuros a definição mais rígida de requisitos e projeto de uma placa PCB menor, com utilização de componentes SMD e montagem do circuito do microcontrolador sem a necessidade de acoplar a placa de desenvolvimento inteira no circuito.

Outra possibilidade é desenvolver novos modelos para o circuito, onde ocorra o acionamento de um numero maior de interruptores com um único microcontrolador, deste modo não ocorreria o congestionamento do roteador com um numero grande de dispositivos conectados, caso se quisesse colocar vários interruptores na casa. Essa possibilidade no entanto dificultaria a instalação elétrica local para fazer chaveamento de varias tomadas.

Uma melhoria interessante para trabalhos futuros, agora já para alguém mais voltado para computação, seria o desenvolvimento do aplicativo para que haja credenciamento utilizando e-mail e senha do administrador dos dispositivos, fazendo o usuário adicionar o dispositivo a sua conta e controlando apenas os dispositivos que tem acesso.

No mais, a estrutura de conexão e comunicação executada neste trabalho dá a oportunidade do desenvolvimento de diversos circuitos para IoT com o Esp-32, onde seja necessária o recebimento de comandos ou o sensoriamento remoto de alguma variável.

Referências

- ADAMI, A. *Domótica*. infoescola, 2013. Acesso em 09/12/2018. Disponível em: <<https://www.infoescola.com/tecnologia/domotica/>>.
- ANDREW, B. *MQTT Version 5.0*. OASIS, 2018. Acesso em 22/11/2018. Disponível em: <<http://docs.oasis-open.org/mqtt/mqtt/v5.0/cs02/mqtt-v5.0-cs02.html>>.
- ASHTON, K. et al. That ‘internet of things’ thing. *RFID journal*, Jun, v. 22, n. 7, p. 97–114, 2009.
- BROWN, E. Who needs the internet of things? *Linux. com. Retrieved*, v. 23, 2016.
- CHAOUCHI, H. *The Internet of things: connecting objects to the web*. [S.l.]: John Wiley & Sons, 2013.
- COKE, M. *The "Only" Coke Machine on the Internet*. Acesso em 21/11/2018. Disponível em: <https://www.cs.cmu.edu/~coke/history_long.txt>.
- FIELDING, R. et al. *Hypertext transfer protocol-HTTP/1.1*. [S.l.], 1999.
- HAASE, J. et al. The iot mediated built environment: A brief survey. In: IEEE. *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on*. [S.l.], 2016. p. 1065–1068.
- HAPNES, S. H. *Is The Internet Of Things Overhyped In 2018?* Forbes, 2018. Acesso em 20/10/2018. Disponível em: <<https://www.forbes.com/sites/quora/2018/08/30/is-the-internet-of-things-overhyped-in-2018/#107faf1722bd>>.
- KOLBAN, N. *Kolban's Book on ESP32*. [S.l.]: USA: Leanpub, 2018.
- LOUREIRO, A. A. et al. Redes de sensores sem fio. In: SN. *Simpósio Brasileiro de Redes de Computadores (SBRC)*. [S.l.], 2003. p. 179–226.
- SANTOS, B. P. et al. Internet das coisas: da teoriaa prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 2016.
- TEEL, J. *Want To Develop a New Electronic Hardware Product?* 2018. Acesso em 10/08/2018.