
Curso de Sistemas de Informação
Universidade Estadual de Mato Grosso do Sul

**INTERNET DAS COISAS (IOT)
AUTOMAÇÃO RESIDENCIAL USANDO TOMADA
ELÉTRICA**

Magno de Oliveira Severino

Prof. MSc. André Chastel Lima (Orientador)

Dourados - MS
2021

Internet das Coisas (IOT)

Automação Residencial Usando Tomada Elétrica

Magno de Oliveira Severino

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Magno de Oliveira Severino e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Dourados, 23 de novembro de 2021.

Prof. MSc. André Chastel Lima (Orientador)

S525i Severino, Magno de Oliveira
Internet das coisas (IOT): automação residencial usando
tomada elétrica / Magno de Oliveira Severino. – Dourados,
MS: UEMS, 2021.
57p.

Monografia (Graduação) – Sistemas de Informação –
Universidade Estadual de Mato Grosso do Sul, 2021.
Orientador: Prof. MSc. André Chastel Lima

1. Internet das coisas 2. Automação residencial 3.
Comodidade I. Lima, André Chastel Lima II.Título

CDD 23. ed. - 004.678

Curso de Sistemas de Informação
Universidade Estadual de Mato Grosso do Sul

INTERNET DAS COISAS (IOT)
AUTOMAÇÃO RESIDENCIAL USANDO TOMADA
ELÉTRICA

Magno de Oliveira Severino

Novembro de 2021

Banca Examinadora:

Prof. MSc. André Chastel Lima (Orientador)
Área de Computação – UEMS

Profa. Dra. Mercedes Rocío Gonzales Márques
Área de Computação – UEMS

Profa. Dra. Raquel Marcia Muller
Área de Computação – UEMS

Este trabalho é dedicado a minha esposa, aos meus pais e professores que sempre estiveram ao meu lado incentivando e ajudando a continuar meus estudos.

AGRADECIMENTOS

Agradeço ao professor Chastel por sempre nos inspirar com seus ensinamentos, aos meus pais que embora possuam poucos estudos, sempre me ajudaram e incentivaram a alcançar um futuro através dos estudos dos quais não tiveram a mesma oportunidade e também a minha esposa, uma grande incentivadora na retomada dos meus estudos sempre apoiando desde quando voltei a fazer cursinho e principalmente no ingresso na universidade.

“Nós só podemos ver um pouco do futuro, mas o suficiente para perceber que há muito a fazer”. Alan Turing

RESUMO

Este trabalho propõe o estudo de Internet das Coisas voltado para automação residencial, mais precisamente na automação de tomada elétrica, com o intuito de realizar o acionamento via navegador da web local. Um tema em bastante expansão nos dias atuais, por se tratar de trazer autonomia e comodidade na realização de tarefas simples do dia a dia. A automação residencial esta cada vez mais presente, principalmente com o advento da tecnologia e pelo fato de que a cada dia essa tecnologia esteja mais acessível. Temos avançado muito na utilização de dispositivos inteligentes seja pelo smartfone, Smart TVs (TVs inteligentes), GPS (Global Positioning System ou Sistema Global de Posicionamento), eletroeletrônicos Smart (Inteligente), dispositivos por rádio frequência ou mesmo por outros dispositivos, nos tornando cada vez mais interconectados com os recursos eletrônicos. Este trabalho visa essa interação, nos aproximar do uso de tecnologias para realizar o controle de aparelhos conectados na tomada a ser controlada, proporcionando bem estar e comodidade, facilidade e agilidade nas tarefas do dia a dia para o nosso lar.

Palavras-chave: Internet das Coisas. Automação Residencial. Comodidade.

LISTA DE FIGURAS

Figura 1: Escopo da IoT.....	19
Figura 2: Primeira Torradeira Conectada a Internet.....	21
Figura 3: RFID usada em objetos de rede do dia a dia.....	22
Figura 4: Raspberry Pi 3 Modelo B.....	25
Figura 5: Módulo Relé - 1 Canal 5v/10a.....	26
Figura 6: Modelos de Módulos Relés.....	27
Figura 7: Painel de Controle.....	28
Figura 8: Tela de Configurações.....	29
Figura 9: Cadastrando um novo usuário.....	29
Figura 10: Ensaio finais com dispositivos elétricos de maior corrente.....	30
Figura 11: Arduino UNO, uma das mais populares e acessíveis placas Arduino do mercado.	31
Figura 12: Raspberry Pi 3B+.....	32
Figura 13: Raspberry Pi Pinos GPIO.....	35
Figura 14: Esquema utilizado para realizar a ligação do Raspberry com o Módulo Relé.....	36
Figura 15: Esquemático de ligação do Módulo Relé no Arduino.....	37
Figura 16: Esquema de Montagem do Hardware usado.....	38
Figura 17: Servidor Node.js (Hello World).....	40
Figura 18: Interface do Protótipo Web de Acionamento de Tomada.....	43
Figura 19: Ligando a Tomada via acesso Web local.....	44
Figura 20: Desligando a Tomada via acesso Web local.....	44
Figura 21: Caso de Uso do Sistema de Automação Residencial usando Tomada Elétrica.....	45

LISTA DE ABREVIATURAS E SIGLAS

ABINC Associação Brasileira em Internet das Coisas
AC Alternating Current ou Corrente Alternada
ANATEL Agência Nacional de Telecomunicações
BLE Bluetooth Low Energy ou Bluetooth de baixo consumo de energia
CIMATEC Centro Integrado de Manufatura e Tecnologia
DC Direct Current ou Corrente Direta
GPS Global Positioning System ou Sistema de Posicionamento Global
HDMI High-Definition Multimedia Interface ou Interface Multimídia de Alta Definição.
IBM International Business Machines
IoT Internet of Things – Internet das Coisas
IPT Instituto de Pesquisas Tecnológicas
ITU International Telecommunications Union
NEPIoT Núcleo de Estudos e Pesquisas em Internet das Coisas
RFID Radio Frequency Identification – Identificação por Radiofrequência
SD Card Secure Digital Card ou Cartão de Memória Digital Segura
SENAI Serviço Nacional de Aprendizagem Industrial
UHF Ultra-High Frequency
USB Universal Serial Bus ou Barramento Serial Universal

SUMÁRIO

Índice

1 INTRODUÇÃO	15
1.1 OBJETIVOS	15
1.1.1 OBJETIVO GERAL	16
1.1.2 OBJETIVOS ESPECÍFICOS	16
1.1.3 JUSTIFICATIVA	16
1.1.4 ORGANIZAÇÃO DO TRABALHO	17
2 REFERENCIAL TEÓRICO	18
2.1 INTERNET DAS COISAS	18
2.1.1 INTERNET DAS COISAS BREVE HISTÓRICO	21
2.2 RASPBERRY PI 3 MODELO B	24
2.3 MÓDULO RELÉ	25
2.4 TRABALHOS CORRELATOS	27
2.5 ARDUINO X RASPBERRY PI	30
3 DESENVOLVIMENTO	34
3.1 IMPLEMENTAÇÕES DE HARDWARES	34
3.1.1 PINOS GPIO DA RASPBERRY	34
3.1.2 MÓDULO RELÉ ESPECIFICAÇÕES	37
3.2 APLICAÇÃO	38
4 CONCLUSÃO	46
REFERÊNCIAS BIBLIOGRÁFICAS	47
APÊNDICE A – CÓDIGOS DO PROJETO	50

1 INTRODUÇÃO

A internet vem evoluindo com o passar dos anos e com essa evolução novas formas de uso tem surgido a fim de melhorar a qualidade de vida das pessoas, proporcionando maior conforto através de um clique. A internet das coisas vem sendo muito usada como uma forma de propiciar automações, um exemplo, já pensou se antes de chegar em casa pudesse ligar a cafeteira ou outro utensílio doméstico pelo celular, para que ao chegar, já esteja pronto para o consumo.

A Internet das Coisas ou *Internet of Things* (IoT) desponta como uma evolução da internet e um novo paradigma tecnológico, social, cultural e digital. A Internet das Coisas revolucionará os modelos de negócios e a interação com o meio ambiente, por meio de objetos físicos e virtuais, em que limites se tornam cada vez mais tênues (LACERDA, LIMA MARQUES, 2015).

O ser humano passa parte de sua vida em sua residência, um ambiente próprio para seu descanso, um local em que realiza afazeres cotidianos e de modo casual. Neste local ele busca adquirir bens de consumo que lhe proporcione praticidade, conforto e entretenimento. Dentro deste contexto, a automação residencial proporciona mais comodidade e agilidade ao usuário que pode se utilizar da tecnologia para realizar suas tarefas, como usar comandos de voz para controlar lâmpadas, centralizar o controle do portão e acionamento de eletrônicos pelo celular usando sistemas inteligentes.

Este trabalho consiste em desenvolver um sistema web simples e prático que forneça automação para o uso de uma tomada da casa usando um servidor Raspberry Pi e um microcontrolador para executar as ações.

1.1 OBJETIVOS

Este trabalho tem por objetivos o estudo de formas de automações residenciais, visando a comodidade, ao utilizar os recursos eletrônicos do ambiente doméstico através da tecnologia para essa residência com o controle de uma tomada.

1.1.1 OBJETIVO GERAL

Este trabalho tem por objetivo a construção de um sistema computacional e um dispositivo eletrônico capaz de controlar uma simples toma elétrica de forma remota.

O objetivo deste trabalho é desenvolver um sistema que atue na automação de uma tomada elétrica da residência com intuito de ligar e desligar aparelhos que estejam atrelados a esta, de forma remota sem interferência direta no eletrônico conectado.

1.1.2 OBJETIVOS ESPECÍFICOS

1. Estudar conceitos de IoT e realizar o levantamento de requisitos para automação da tomada.
2. Construir uma sistema computacional que possibilite a intercomunicação entre os hardwares envolvidos e o usuário.
3. Criar uma aplicação web que forneça uma interface amigável e que seja possível realizar os testes de conectividade entre os hardwares distintos.

1.1.3 JUSTIFICATIVA

A automação traz uma maior facilidade para as pessoas através da realização de algumas atividades corriqueiras mais rapidamente exigindo uma menor interação com o dispositivo utilizado no processo.

Desta forma podemos então pensar, que aplicando a automação dentro das residências, poderíamos diminuir muitas ações feitas pelo ser humano através de sistemas automáticos. Desta forma é possível controlar componentes como iluminação, refrigeração, portões, janelas, entre muitos outros, analisar fatores como temperatura e umidade, além de aumentar a segurança das residências, com o controle de câmeras e alarme, tudo isso com o menor esforço humano possível (BRAGA, 2015).

Este trabalho emprega um sistema que utiliza funções de controle de tomada elétrica, com intuito de levar conforto e praticidade para o domicílio, de uma forma barata e simples,

sem ser necessário realizar a compra de dispositivos de preços elevados, tornando, dessa maneira de fácil acesso.

1.1.4 ORGANIZAÇÃO DO TRABALHO

No capítulo 2 segue o referencial teórico, obtido através da leitura de artigos científicos, pesquisas em web sites, livros, revistas. Na seção 2.1 o conceito de IoT é apresentado juntamente com histórico de uso de aqui no Brasil. Na seção 2.2 é apresentada uma plataforma processada chamada Raspberry, especificamente o modelo Raspberry Pi 3 Modelo B, que foi usado para este projeto, a seguir na seção 2.3 é apresentado o microcontrolador Módulo Relé suas características, o modelo usado e exemplos de outros modelos. Na seção 2.4 é apresentado um trabalho semelhante ao proposto e por seguinte na seção 2.5 é apresentada as diferenças entre as placas Arduino e Raspberry Pi.

O capítulo 3 aborda o desenvolvimento e as tecnologias usadas no trabalho. Apresenta as implementações de hardware na seção 3.1, mostrando na subseção a seguir 3.1.1 os pinos GPIO Raspberry (Pinos de entrada e saída de uso geral), formas de uso e a lógica usada na ligação do hardware, em seguida descreve o módulo relé na subseção 3.1.2, suas especificações, características e dificuldades encontradas na ligação com o Raspberry. A seção 3.2 mostra a aplicação desenvolvida, linguagens de programação usadas, dificuldades encontradas no estabelecimento da comunicação via software e soluções encontradas.

O tópico 4 apresenta a conclusão do trabalho, as dificuldades na realização e a expectativa de trabalhos futuros nessa área de conhecimento.

2 REFERENCIAL TEÓRICO

Com base nos artigos pesquisados o capítulo 2 apresenta os conceitos básicos e as características no que se refere a Internet das Coisas e do trabalho proposto. A seção 2.1 apresenta os conceitos Internet das Coisas. Na subseção 2.1.1 é apresentado o contexto histórico, primeiros usos e tecnologias usadas, seguindo a seção 2.2 apresenta A Raspberry PI 3, seus conceitos e características. A seção 2.3 demonstra o Módulo Relé, suas características e modelos. Na seção 2.4, é apresentado o trabalho do Lucas Cardoso Viero de 2016, que fez um trabalho parecido usando Arduino. Na seção 2.5 é demonstrado as diferenças entre as placas Arduino e Raspberry Pi.

2.1 INTERNET DAS COISAS (IOT)

O termo Internet das Coisas possui uma grande diversidade de conceitos e visões diferentes para diversos autores.

Dentre estes conceitos podemos destacar alguns:

A ideia básica desse conceito é a presença generalizada à nossa volta de uma variedade de coisas ou objetos – como tags de identificação por radiofrequência (RFID), sensores, atuadores, telefones celulares, etc. - que, por meio de esquemas de endereçamentos exclusivos, são capazes para interagir uns com os outros e cooperar com outros objetos para alcançar objetivos comuns (Atzori et al, 2011, p.2787).

Do inglês Internet of Things (IoT), a Internet das Coisas refere – se 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 esses dados, gerem informações e serviços em escala inimaginável (Revista Computação Brasil, 2015, p.7).

Uma infraestrutura de rede local, interligando objetos físicos e virtuais por meio da exploração de captura e comunicação de dados e capacidade de comunicação. Essa infraestrutura inclui a internet existente e em evolução, bem como os desenvolvimentos de rede. Ela oferecerá identificação de objetos específica e capacidade de sensoriamento e de conexão como base para o desenvolvimento de aplicações e serviços independentes cooperativos. Estes

serão caracterizados por elevado grau de captura autônoma de dados, transferência de eventos, conectividade e interoperabilidade de rede (CASAGRAS, Amazonas, 2010, tradução).

Ubiquidade, computação pervasiva, ambiente inteligente e internet das coisas são conceitos praticamente idênticos. Ubiquidade é a contínua otimização e promoção de processos sociais e econômicos por inúmeros microprocessadores e sensores integrados ao ambiente (Friedwald, Michel; Raabe, Oliver; 2011).

Dentro deste contexto de intercomunicação entre todos objetos podemos determinar que a IoT compreende uma rede complexa que se conecta a vários objetos por meio de protocolos de comunicação podendo ter representação física ou virtual disponível no mundo. A figura 1 nos mostra a ideia inicial do escopo da IoT.

Figura 1: Escopo da IoT



Fonte: Faccioni Filho, Mauro. Internet das Coisas, UnisulVirtual, Palhoça (2016)

As aplicações de Internet das Coisas são inúmeras e diversas, e permeiam praticamente a vida diária das pessoas, das empresas e sociedade como um todo, transformando o mundo *smart world* (FREITAS DIAS, 2016; PATEL, 2016).

Dentro das atuações da internet das coisas podemos citar:

- **Bens de consumo.** Bens adquiridos pelos consumidores, tais como smartphones, smart house, smart car e smart TV;

- **eHealth.** Fitness, bioeletrônica e cuidados com a saúde. Por exemplo, monitoramento e controle da frequência cardíaca durante os exercícios; monitoramento das condições dos pacientes em hospitais e casas de idosos;

- **Transporte Inteligente.** Notificação de tráfego, controle inteligente de rotas, monitoramento remoto de veículo, coordenação das rodovias e integração inteligente de plataformas de transporte;

- **Distribuição de energia (smart grid).** Acompanhamento de instalações de energia, subestações inteligentes, distribuição de energia automática e medições remotas de relógios residenciais;

- **Casas Inteligentes.** Medições remotas de consumo, economia de energia, controle inteligente de equipamentos residenciais e segurança residencial;

- **Distribuição e Logística.** Smart e-commerce, rastreabilidade, gerenciamento na distribuição e inventário;

- **Segurança Pública.** Monitoramento no transporte de cargas perigosas e químicas, monitoramento da segurança pública, monitoramento das estruturas de construções de utilidade pública;

- **Indústria e Manufatura.** Economia de energia, controle de poluição, segurança na manufatura, monitoramento do ciclo de vida dos produtos, rastreamento de produtos manufaturados na cadeia de abastecimento, monitoramento de condições ambientais e controles de processos de produção;

- **Gestão da agricultura e dos recursos naturais.** Segurança e rastreabilidade de produtos agrícolas, gerenciamento de qualidade, monitoramento ambiental para a produção e cultivo, gerenciamento no processo de produção, utilização de recursos para a agricultura;

- **Smart Cities.** Monitoramento estrutural: monitoramento de vibrações e condições dos materiais em edifícios, pontes e monumentos históricos. Energia elétrica: iluminação inteligente e adaptável conforme a rua. Segurança: monitoramento por meio de vídeo digital, gerenciamento de controle de incêndio e sistemas de anúncio público. Transporte: estradas inteligentes com avisos, mensagens e desvios de acordo com as condições climáticas e eventos inesperados como acidentes e engarrafamentos. Estacionamentos: monitoramento em tempo real da disponibilidade de espaços de estacionamento, sendo possível identificar e reservar vagas disponíveis. Gestão de resíduos: detecção de níveis de lixo em recipientes para otimizar a rota de coleta de lixo (MANCINI, Monica; Internet das Coisas: História, Conceitos, Aplicações e Desafios, 2018).

2.1.1 INTERNET DAS COISAS BREVE HISTÓRICO

Em 1990, John Ronkey criou o primeiro dispositivo em internet das coisas. Esse autor criou uma torradeira que poderia ser ligada e desligada pela Internet e a apresentou na INTERROP 89 Conference. Dan Lynch, presidente da Interop na época, prometeu a John Ronkey que, se a torradeira fosse ligada pela internet, o aparelho seria colocado em exposição durante a conferência. Diante desse desafio, John Ronkey conectou a torradeira a um computador com rede TCP/IP, e foi um tremendo sucesso. Porém, durante esse teste, o pão foi incluído manualmente na torradeira. Após um ano, esse requisito foi corrigido e apresentado na mesma conferência, por meio de um pequeno guindaste robótico no sistema (Figura 2). Esse robô era controlado pela Internet, pegou a fatia de pão e colocou na torradeira, automatizando, dessa forma, o sistema de ponta a ponta (DEORAS, 2016).

Figura 2: Primeira Torradeira Conectada a Internet



Fonte: Gumption (2016)

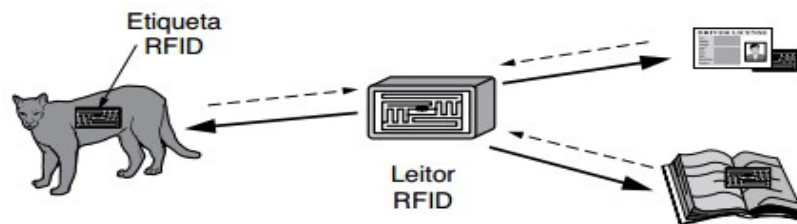
Em 1991, Weiser (1991) escreveu o artigo *The Computer for the 21st Century*, que, aborda o futuro da Internet das Coisas. Esse autor chama de “computação ubíqua”. No artigo, o autor afirma que os dispositivos serão conectados em todos os lugares de forma tão transparente para o ser humano, que se tornará “invisível”, possibilitando, de forma natural, a realização das atividades, sem haver a preocupação em instalar, configurar e manter os recursos computacionais (WEISER, 1991; GALEGALE et al., 2016). Este **artigo é um marco na pesquisa sobre Internet das Coisas** e é citado em praticamente em toda a literatura sobre esse assunto (SINGER, 2012).

Em setembro de 1999, Kevin Ashton, cofundador e diretor executivo do *Auto-ID Center*, proferiu uma palestra para a *Procter & Gamble*, e apresentou uma nova ideia do sistema RFID para a rastreabilidade do produto na cadeia de suprimentos. Para chamar a atenção dos executivos, ele colocou no título da apresentação a expressão *Internet of Things*. Para Ashton (2009), os objetos do mundo físico poderiam se conectar à internet, criando um mundo mais inteligente (FINEP, 2015). **Esse termo foi utilizado primeiramente por Kevin Ashton, considerado o criador desse termo** (POSTCAPES, 2017a).

A partir de 2005, a discussão sobre Internet das Coisas se generalizou e ganhou atenção dos governos em relação à privacidade e segurança de dados. Nesse ano, a *International Telecommunications Union* (ITU) publicou um relatório com o conceito de Internet das Coisas com uma visão abrangente e holística. No seu ponto de vista, internet das coisas poderia conectar qualquer objeto, por meio de tecnologias, como RFID, sensores, redes de sensores sem fio, sistemas embarcados e nanotecnologia, além de transpor alguns desafios importantes como padronização, privacidade, espectro de frequência e questões sociais éticas (FREITAS DIAS, 2016).

Uma etiqueta RFID se parece com um selo postal adesivo, que pode ser afixado (ou embutido) em um objeto para que possa ser rastreado. O ‘objeto’ pode ser uma vaca, um passaporte, um livro ou um palete de armazenagem. A etiqueta consiste em um pequeno microchip com um identificador exclusivo e uma antena que recebe transmissões de rádio. Leitores de RFID instalados nos pontos de rastreamento encontram as etiquetas quando elas entram no alcance e solicitam informações, como mostra a Figura 3. As aplicações incluem verificação de identidades, gerenciamento da cadeia de suprimentos, marcação de tempo em corridas e substituição de códigos de barras (Tanenbaum, Wettrall, pg, 45, 2011).

Figura 3: RFID usada em objetos de rede do dia a dia



Fonte: Tanenbaum, Wettrall, 2011

A grande vantagem dessa tecnologia é o fato de a grande maioria dispensar o uso de baterias ou tomadas elétricas, toda a energia é proveniente das ondas de rádio pelos leitores de RFID.

Uma forma comum de RFID é a UHF RFID (Ultra-High Frequency RFID). Ela é usada em paletes de armazenagem e em algumas carteiras de habilitação. Os leitores enviam sinais na banda 902-928 MHz nos Estados Unidos. As etiquetas se comunicam a distância de vários metros, mudando o modo como elas refletem os sinais do leitor; este é capaz de apanhar essas reflexões (Tanenbaum, Wettrall, 2011, pg 45).

Em 2010, Salvador sediou o primeiro evento em Internet das Coisas, conhecido como o “1º Congresso de Tecnologia, Sistema e Serviços com RFID”, organizado pelo CIMATEC SENAI e Saint Paul Etiquetas Inteligentes. A 2ª edição ocorreu em Búzios, em 2011, e mudou o nome para “Congresso Brasileiro de Internet das Coisas e RFID”. Ainda em 2010, a cidade do Rio de Janeiro operava com tecnologia de cidades inteligentes da IBM, por meio de um telão com o mapa da cidade e imagens de câmeras, que permitiam visualizar o trânsito e diversas ocorrências (Mancini, Monica, 2018)

Em junho de 2011 foi criado o Fórum Brasileiro de Internet das Coisas. Seu objetivo é mostrar a importância da Internet das Coisas para a sociedade, as novas tecnologias, e como o Brasil pode ser participante global nesse seguimento (FÓRUM BRASILEIRO DE IOT, 2017).

Em dezembro de 2015 foi fundada a Associação Brasileira em Internet das Coisas (ABINC), cujo propósito é representar o mercado perante a Anatel, o Ministério das Comunicações, autoridades constituídas e outros órgãos reguladores setoriais ou de fomento de pesquisa por meio de seus associados (ABINC, 2017).

Em 2016, foi criado o **núcleo de estudos e pesquisas em internet das coisas (nepiot)**, com o propósito de ser um hub de experimentação para estudos e projetos de pesquisa, desenvolvimento e inovação baseado em internet das coisas. É formado por empresas conveniadas ao Instituto de Pesquisas Tecnológicas (IPT), que “operam em rede, para conduzir estudos e projetos em Internet das Coisas, e desenvolver um ecossistema experimental para a promoção e validação das tecnologias associadas com aplicações em ambiente urbano e/ou rural” (NEPIoT, 2017).

A partir de 2017 o governo brasileiro abriu uma Consulta Pública para realizar diagnóstico e propor políticas públicas no tema de Internet das Coisas, estimulando a cooperação e articulação entre empresas, poder público, universidades e centros de pesquisas (PARTICIPA.BR, 2017)

2.2 RASPBERRY PI 3 MODELO B

O Raspberry Pi é um computador barato, portátil e versátil, usado principalmente em projetos de programação, robótica e em iniciativas em geral com software e hardware livre. Algumas utilidades do dia a dia incluem criação de servidor de arquivos pessoal e videogame retrô (ALVES, Paulo. 2017, site techtudo.com).

O mini PC roda Ubuntu, Raspbian e outras distribuições do sistema operacional Linux. Além disso, é compatível com o Windows 10 IoT, versão do software da Microsoft feita para automação doméstica e outras aplicações envolvendo Internet das Coisas (ALVES, Paulo. 2017, site techtudo.com).

O Raspberry Pi 3 Model B vendido no país conta com Wi-Fi e Bluetooth integrados, gerando ainda mais economia em projetos que necessitam de mais conectividade (ALVES, Paulo. 2017, site techtudo.com).

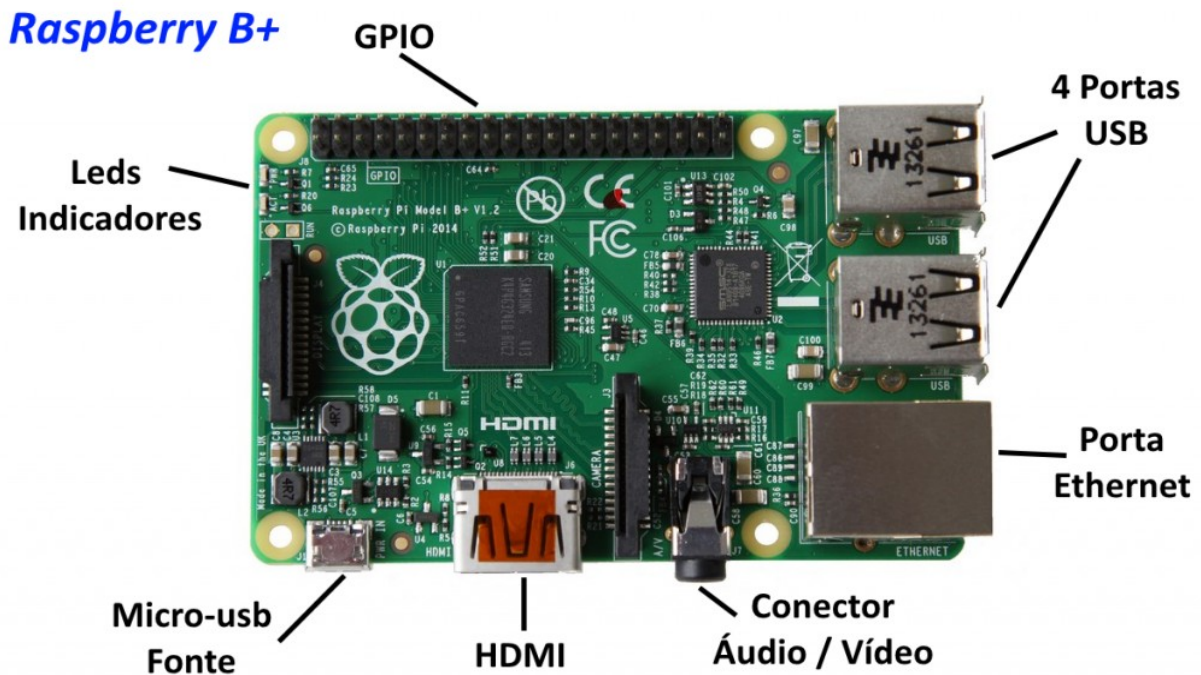
Especificações técnicas:

- Computador de placa única com processador quadcore Broadcom BCM 64 bits rodando a 1,2 GHz e com 1GB de Ram
- BCM 43143 WiFi integrado
- Bluetooth com baixa energia (BLE Bluetooth Low Energy, ou Bluetooth de baixa consumo de energia) integrado
- 40 pinos GPIO (General Purpose Input/Output ou Entrada e saída de uso geral)
- 4 USB 2.0
- Saída estéreo de 4 pólos e porta de vídeo composto
- Porta de câmera HDMI (High-Definition Media Interface ou Interface Multimídia de Alta Definição) CSI
- Porta de exibição para conectar tela de toque do Raspberry Pi
- Porta Micro SD para carregar o sistema operacional

- Fonte de Alimentação Micro USB (Universal Serial Bus ou Barramento Serial Universal) (suporta até 2,4 Amps) (KOFUJI, Sergio Takeo, site edisciplinas.usp.br).

A figura 4 a seguir apresenta os componentes do Raspberry Pi 3 Modelo B que foi usado para esse projeto.

Figura 4: Raspberry Pi 3 Modelo B



Fonte: <https://www.arduinoecia.com.br/comparativo-raspberry-pi-b-plus/>

Este projeto em questão foca na utilização desses pinos (GPIO General Purpose Input/Output ou Entrada e Saída de uso geral) para ativar ou desativar um microcontrolador, sendo possível o controle via software de uma tomada vinculada.

2.3 MÓDULO RELÉ

Os microcontroladores, tais como Atmega, PIC e MSP, são dispositivos lógicos. Eles são usados com o intuito de ser a inteligência do circuito, o cérebro do nosso sistema inteligente (MOTA, Alan, 2017, portal.vidadesilicio.com.br).

Relés são componentes eletromecânicos. Isso é, eles tem uma parte elétrica e outra mecânica. Eles são muito utilizados para acionar cargas maiores que a tensão do seu microcontrolador. Um grande exemplo que podemos citar é o ligar e desligar de lâmpadas utilizando Arduino (autocore, 2015 site autocorerobotica.blog.br).

O Módulo Relé 5V 1 Canal permite que a partir de uma plataforma microcontrolada seja possível controlar cargas AC (alternada) de forma simples e prática. Por ter apenas 1 canal, é possível controlar apenas uma carga AC de até 10A. Comumente é utilizado em projetos de automação residencial para controle de lâmpadas, ventiladores e outras saídas que possam ser acionadas através de relé (site blogmasterwalkershop.com.br)

A figura 5 apresenta o modelo de relé utilizado neste projeto.

Figura 5: Módulo Relé - 1 Canal 5v/10a



Fonte: Disponível em <https://produto.mercadolivre.com.br/>

Especificações:

- Tensão de operação: 5 V DC (VCC e GND)
- Tensão de Sinal: TLL – 5V DC (IN)
- Corrente típica de operação: 15~20mA
- O Relé possui contato NA, NF, COM
- Capacidade do Relé: 30 V DC e 10A ou 250V AC e 10A
- Tempo de Resposta: 5~10ms
- Indicador LED de funcionamento
- Dimensões: 43 mm (L) x 17 mm (C) x 19 mm (H) (site vidadesilicio.com.br).

Pinagem:

GND – 0V

IN – entrada controle relé

VCC – 5V (site vidadesilicio.com.br).

O relé de 1 canal usa 5V podendo ser conectado direto na Raspberry ou Arduino, entretanto ao usar relés com mais de 1 canal é indicado o uso de uma fonte de energia a parte.

No mercado existem diversas opções de relés com várias quantidades de canais podendo ir até 16 numa mesma placa.

A figura 6 mostra alguns exemplos de relés, podendo controlar cargas de acordo com sua quantidade de canais.

Figura 6: Modelos de Módulos Relés



Fonte: Disponível em <https://produto.mercadolivre.com.br/>

2.4 TRABALHOS CORRELATOS

Alguns trabalhos que tratam do mesmo assunto foram pesquisados com o intuito de verificar outras formas de abordar o mesmo tipo de automação, dentre os pesquisados, resolvi citar o do Lucas Cardoso Viero de 2016, que muito se aproxima do projeto realizado.

O trabalho Sistema de Automação Residencial Utilizando Rede Wifi com Arduino e Website de Lucas Cardoso Viero, 2016, se assemelha ao meu projeto, pois, o objetivo geral do seu trabalho foi a implementação de um sistema de automação residencial de baixo custo, utilizando a rede sem fio do próprio ambiente juntamente com as funcionalidades do módulo Arduino e um website desenvolvido para controlar diferentes módulos de segurança de uma residência através de dispositivo móvel.

Neste trabalho ele usou o Servidor Apache, linguagem de programação HTML, linguagem PHP, Banco de Dados MySQL, os hardwares usados foram o Arduino Uno R3 e Módulo relé de 2 estados com 4 canais de operação.

Para o site dele, foi criado as telas de autenticação com telas de login e cadastro de usuários, a tela de painel de controle com função de configuração podendo ser verificado os usuários cadastrados, permitindo a exclusão , para as senhas foi usado criptografia. O Arduino foi usado para a integração do sistema por possuir diversas portas de entrada e saída de dados I/O. O módulo relé utilizado foi o de 4 canais. As figuras abaixo, mostram as telas da aplicação desenvolvida.

Figura 7: Painel de Controle



Fonte: <https://www.ufsm.br/app/uploads/sites/495/2019/05/2016-Lucas-Viero.pdf>

Tela de Configurações.

Figura 8: Tela de Configurações



Fonte: <https://www.ufsm.br/app/uploads/sites/495/2019/05/2016-Lucas-Viero.pdf>

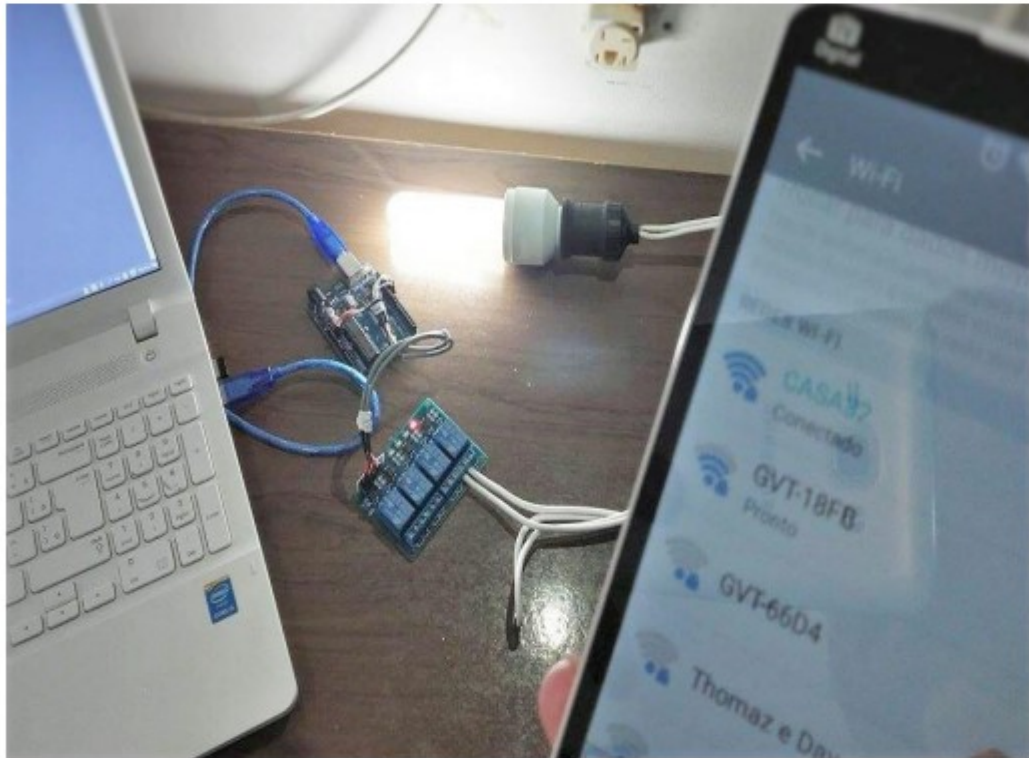
Tela de cadastro de usuários.

Figura 9: Cadastrando um novo usuário

Fonte: <https://www.ufsm.br/app/uploads/sites/495/2019/05/2016-Lucas-Viero.pdf>

Nestes ensaios foram utilizadas lâmpadas ligadas na energia elétrica e os próprios relés de acionamento, conforme a figura abaixo.

Figura 10: Ensaio finais com dispositivos elétricos de maior corrente



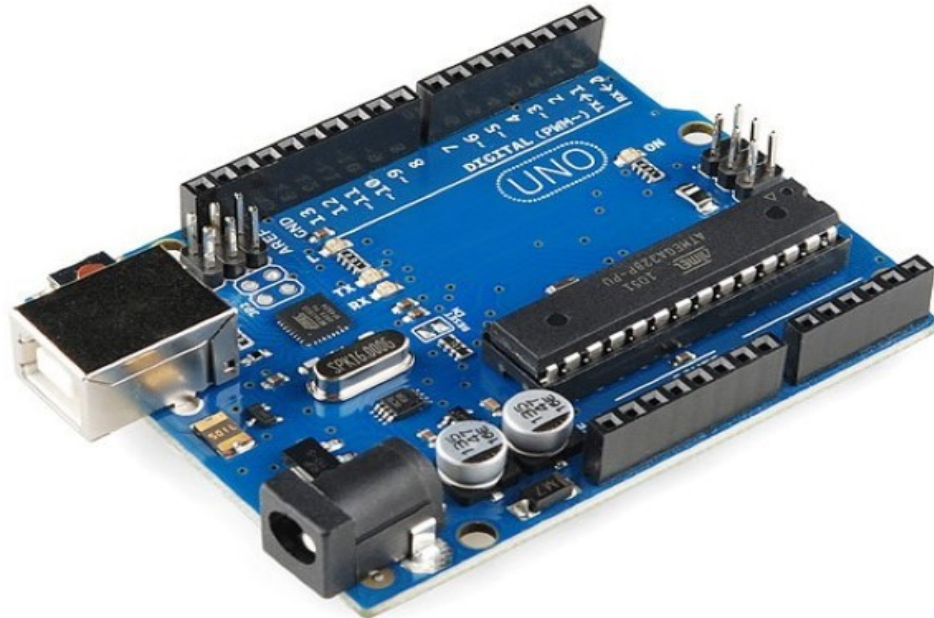
Fonte: <https://www.ufsm.br/app/uploads/sites/495/2019/05/2016-Lucas-Viero.pdf>

2.5 ARDUINO X RASPBERRY PI

Nos últimos tempos, as placas de prototipação rápida e kits de desenvolvimento passaram a ser disponibilizadas de maneira muito mais acessível – tanto em preço quanto em disponibilidade no mercado – aos profissionais, hobistas e makers. As mais famosas são Arduino e Raspberry Pi (site Filipeflop.com).

Há um grande número de placas da plataforma Arduino disponíveis no mercado, inclusive você pode produzir a sua própria, já que o Arduino se trata de um projeto Open-Source em hardware e software, como pode ser visto aqui um dos mais populares e acessíveis Arduinos é o Arduino UNO, conforme a figura abaixo (site filipeflop.com).

Figura 11: Arduino UNO, uma das mais populares e acessíveis placas Arduino do mercado



Fonte: <https://www.filipeflop.com/blog/arduino-x-raspberry-pi-quais-as-diferencas-entre-as-placas>

Entretanto, todas elas possuem alguns elementos em comum:

- Um microcontrolador, usualmente de baixo custo (originalmente os ATMEGA, sendo hoje já possível de utilizar microcontroladores de outros fabricantes senão a ATMEL), rodando a frequências tipicamente iguais ou inferiores a 48MHz.
- Poucos Kb de memória RAM e Flash
- Conexão USB, para alimentação e programação
- Pin-headers ou pin sockets para GPIOs, entradas analógicas, saídas PWN, etc
- Os pin-headers ou pin-sockets já são no formato certo para o encaixe dos Shields Arduinos (placas de expansão, que permitem adicionais mais funcionalidades e recursos ao seu sistema embarcado)
- LEDs, sendo pelo menos 1 livre para o usuário / desenvolvedor (site

filipeflop.com).

Em termos simples, podemos dizer que a Raspberry Pi é “muito turbinada” em relação ao Arduino. A Raspberry Pi possui recursos computacionais e de hardware que permitem que seu uso como um computador pessoal comum, logo possui, mas muito mais “poder de fogo” que um Arduino (site filipeflop.com).

Ao invés de um microcontrolador de baixo custo e limitado, a Raspberry faz o uso de um SoC (System on Chip), podendo conter mais de uma CPU e suporte a protocolos e conexões avançadas (Ethernet, USB, HDMI, wifi, etc.) (site filipeflop.com).

Assim como o Arduino, as placas da linha Raspberry Pi são Open-Source tanto em hardwares quanto em software. Isso significa que você pode, se desejar, construir sua própria placa baseada na Raspberry Pi, sendo uma das mais populares a Raspberry Pi 3B+ (site filipeflop.com).

Figura 12: Raspberry Pi 3B+



Fonte: <https://www.filipeflop.com/blog/arduino-x-raspberry-pi-quais-as-diferencas-entre-as-placas>

Apesar dos vários modelos disponíveis, todas as Raspberry Pi dispõe de:

- Uma SoC, com uma ou mais CPUs, rodando menos 700 MHz
- 256MB ou mais de memória RAM
- Interfaces USB
- Interface HDMI

- Suporte a câmera via interface CSI (Camera Serial Interface).
- 26 ou 40 pinos no header de expansão, permitindo o uso de GPIOs, protocolo UART, protocolo I2C, etc (site filipeflop.com).

Poder Computacional – em termos de poder computacional, a Raspberry Pi tem o mesmo porte de um computador de uso pessoal comum, ou seja, roda a frequências altas (700MHz ou mais), utiliza um sistema operacional de propósito geral (uma distro Linux, por exemplo), utiliza as mesmas aplicações de um computador pessoal comum, possui megabytes (ou gigabytes) de memória RAM e possui armazenamento em massa na casa de dezenas de gigabytes. Já o Arduino não é capaz de rodar um sistema operacional de propósito geral, possui kilobytes de memória RAM e armazenamento e roda, tipicamente, no máximo a 48MHz (site filipeflop.com).

Operação em tempo real – sistemas operacionais de propósito geral não possuem capacidade de rodar em hard real time. Isso significa que podem ocorrer latências (mínimas, mas existem) na execução das aplicações. Isso faz com que este tipo de sistema operacional não seja apto a funcionar em projetos com altíssimas criticidade em tempo, como controle de aviões militares, por exemplo. Já Arduinos (e outras plataformas semelhantes) com microcontroladores que utilizam sistemas operacionais de tempo real (RTOS) podem garantir altíssima fidelidade nos tempos de execução de tarefas, atendendo às exigências de projetos com altíssima criticidade em tempo (site filipeflop.com).

Versatilidade – como dito no tópico anterior as placas da linha Raspberry Pi operam com sistemas operacionais de propósito geral. Isso significa dizer que tarefas muito importantes e corriqueiras, como por exemplo aplicação de camadas de segurança de rede, acesso remoto, atualização remota de software e backup, já são normalmente cobertas pelo sistema operacional, bastando o desenvolvedor utilizar tais recursos. Já com Arduino, se algo desse tipo for possível, pode ter certeza que seria uma dor de cabeça sem fim implementar (site filipeflop.com).

Custo final – às vezes, o projeto que estamos trabalhando pode requerer um hardware baratíssimo, com poucos recursos computacionais. Ou seja, um hardware que seja capaz de lidar com tarefas simples, custando pouco. Nesse caso, usar uma Raspberry Pi pode ser o equivalente a matar uma formiga com uma bala de canhão. Já um Arduino (ou outra plataforma equivalente) pode ser uma ótima solução (site filipeflop.com).

3 DESENVOLVIMENTO

Nos seções seguintes, são detalhados os meios utilizados para montar o projeto desde a implementação do hardware seção 3.1 e a aplicação na seção 3.2, bem como a lógica usada e os desafios encontrados ao estabelecer a comunicação entre os dispositivos.

3.1 IMPLEMENTAÇÕES DE HARDWARES

Para este trabalho como já foi dito anteriormente, foi usado um Raspberry Pi 3 modelo B e um módulo relé de 1 canal, nesta seção é mostrado como funciona os dispositivos e como foi realizada a conexão entre estes.

A maior dificuldade encontrada é nessa parte, até então nunca havia entrado em contato com essa tecnologia hardware/hardware, sendo necessário um estudo de como funcionam os pinos da Raspberry e como poderia usar para realizar as conexões sem colocar o hardware em questão em risco. A subseção 3.1.1 trata de como funciona os pinos da Raspberry e como cheguei a lógica de adicionar o relé ao Raspberry.

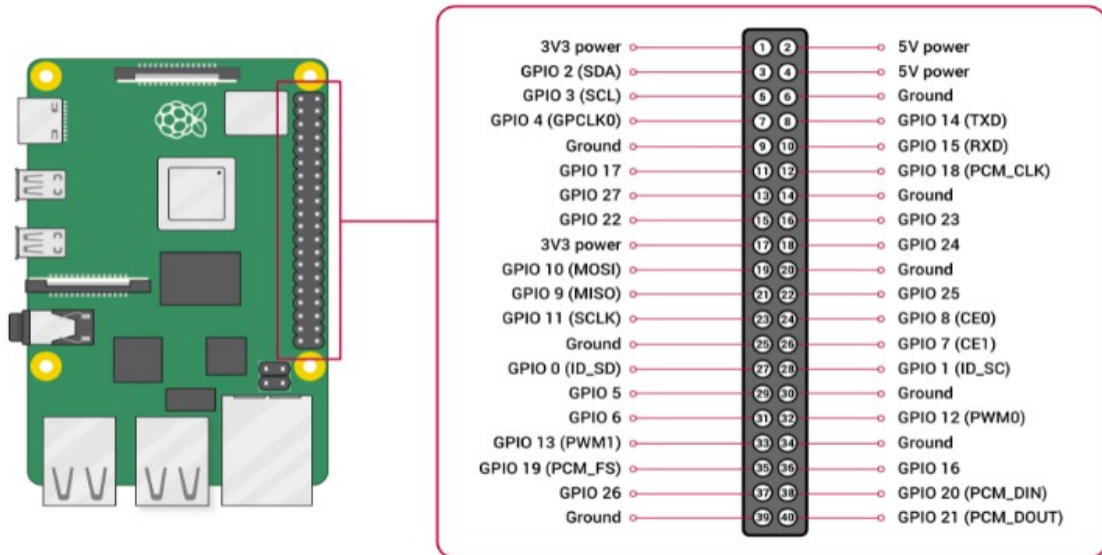
A subseção 3.1.2 traz o módulo relé e seu funcionamento e também como foi a conexão no Raspberry.

3.1.1 PINOS GPIO DA RASPBERRY

Um recurso poderoso do Raspberry Pi é a linha de pinos GPIO (entrada/saída de uso geral). Um cabeçalho de 40 pinos é encontrado em todas as placas Raspberry atuais (não preenchidas em Pi Zero e Pi Zero W). Antes do Pi 1 Modelo B+ (2014), as placas eram compostas por um cabeçalho mais curto de 26 pinos (Documentação do Raspberry, site raspberrypi.org).

A figura 7, a seguir apresenta o mapa de pinos da Raspberry Pi 3.

Figura 13: Raspberry Pi Pinos GPIO



Fonte: <https://www.raspberrypi.com/documentation/computers/os.html#gpio-and-the-40-pin-header>

Dois pinos de 5V e dois pinos de 3V3 estão presentes na placa, bem como vários pinos de aterramento (0V), que não são configuráveis. Os pinos restantes são todos pinos 3V3 de uso geral, o que significa que as saídas são definidas como 3V3 e as entradas são tolerantes a 3V3 (Documentação do Raspberry, site raspberrypi.org).

Assim como dispositivos simples de entrada e saída, os pinos GPIO podem ser usados com uma variedade de funções alternativas, algumas estão disponíveis em todos os pinos, outras em pinos específicos.

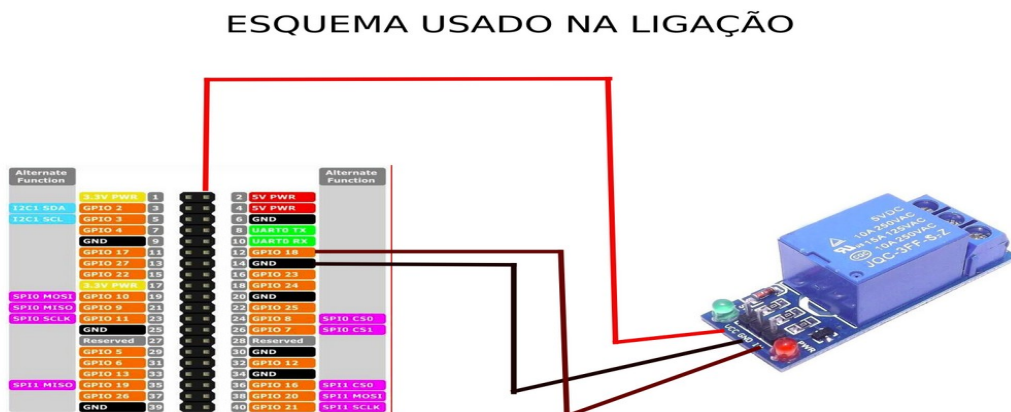
- PWN (modulação por largura de pulso)
 - Software PWN disponível em todos os pinos.
 - Hardware PWN disponível em GPIO12, GPIO13, GPIO18, GPIO19
- SPI
 - SPIO: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CEO (GPIO8), CE1 (GPIO7).

- SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)
- I2C
 - Dados: (GPIO2), Relógio (GPIO3)
 - Dados EEPROM: (GPIO0); Relógio EEPROM (GPIO1)
- Serial
 - TX (GPIO14); RX (GPIO15) (Documentação do Raspberry, site raspberrypi.org).

Sendo as nomenclaturas a seguir: GPIO – entrada/saída de uso geral, SPI (Interface Periférica Serial), I2C (Circuito Inter-integrado), UART (receptor/transmissor assíncrono universal); PCM (modulação por código de pulso); GND (terra), 5v (potência) e 3,3V (potência).

Partindo dessa especificação a ligação utilizada no trabalho foram os pinos 2 de 5V o pino 14 GND e o pino 12 GPIO18. O módulo relé tem 3 pinos para conectar ao Raspberry ou Arduino, são eles o VCC (5V) ligado ao pino 2 da Raspberry, GND (0V) ligado ao pino 14 da Raspberry e IN (input) ligado ao pino 12 (GPIO18), a figura 8 abaixo mostra a ligação entre os hardwares.

Figura 14: Esquema utilizado para realizar a ligação do Raspberry com o Módulo Relé



Fonte: Pinos (labprototipando.com.br), Módulo Relé (mercadolivre.com.br)

A seção a seguir, mostra o funcionamento do Relé e como se realizou a comunicação entre os hardwares por meio do Software.

3.1.2 MÓDULO RELÉ ESPECIFICAÇÕES

O módulo relé é ideal para acionar uma lâmpada ou outra carga que exija até no máximo 10A contínuos utilizando o Arduino ou qualquer microcontrolador (Módulo Relé Arduino, site robocore.net).

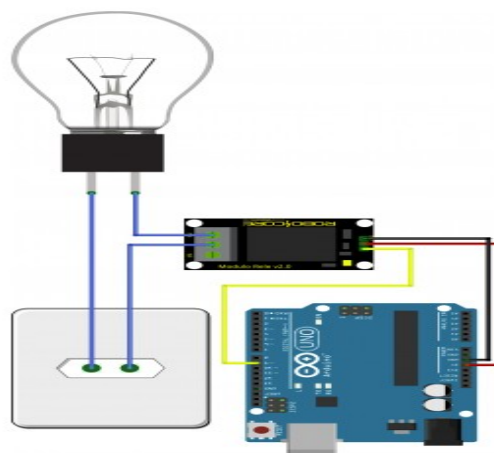
Ele funciona exatamente como uma chave (interruptor). No borne cinza há 3 conexões: NA (Normalmente Aberto), C (Comum), e NF (Normalmente Fechado) (Módulo Relé Arduino, site robocore.net).

Geralmente, os dispositivos estão escritos com as siglas em inglês, o NA (Normalmente aberto) é equivalente ao NO (Normally Open), NF (Normalmente fechado) como NC (Normally Close) e o C (Comum) como COM (Common).

Ou seja, quando o módulo relé estiver “desligado”, C estará conectado à NF. Quando estiver ligado, C estará conectado a NA. (Módulo Relé Arduino, site robocore.net).

No esquemático abaixo figura 9, demonstra a ligação do Módulo Relé no Arduino e em uma lâmpada.

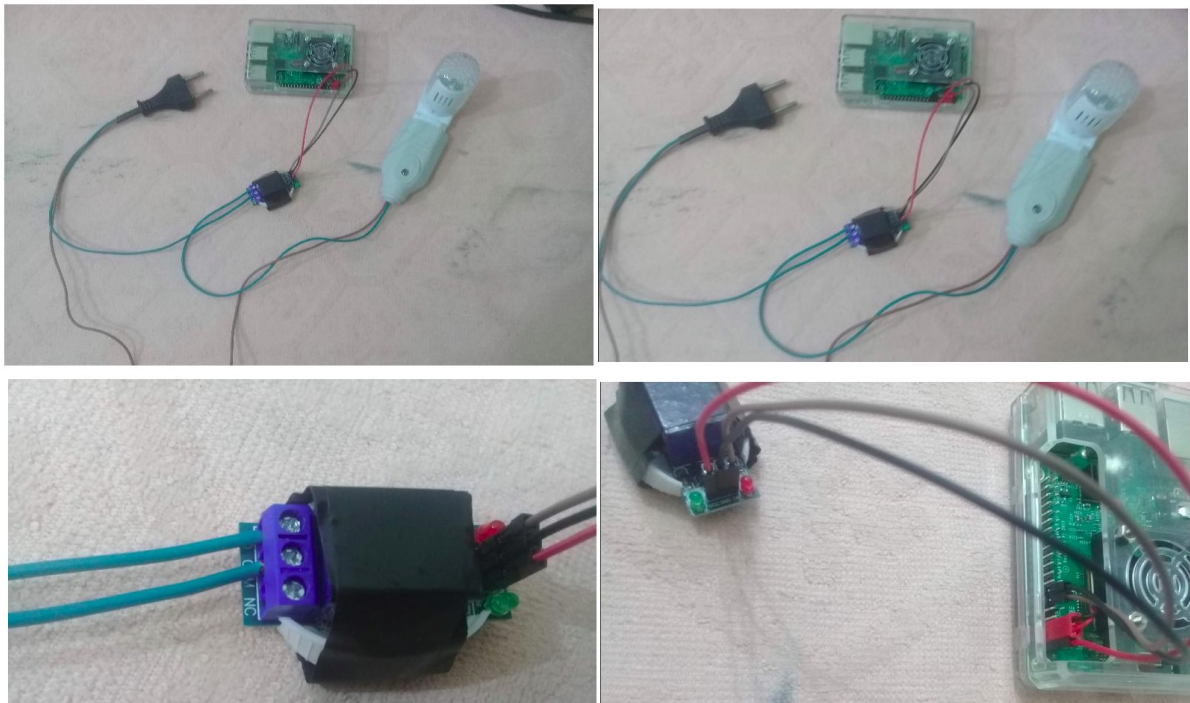
Figura 15: Esquemático de ligação do Módulo Relé no Arduino



Fonte: <https://www.robocore.net/tutoriais/modulo-rele-arduino>

Usando a ideia acima, adaptei para o Raspberry Pi 3, no meu projeto uso uma tomada conectada na COM e no NO do módulo relé como demonstrado na figura 10 abaixo. Para o Raspberry foi usado um case de acrílico e no módulo relé foi feita uma adaptação com uma tampa de plástico de embalagem pet, apenas para que os contatos elétricos das placas não entrassem em contato com a superfície, mesmo não sendo necessário foi apenas um cuidado a mais.

Figura 16: Esquema de Montagem do Hardware usado



Esquema Utilizado na Montagem do Hardware

Fonte: Autor

3.2 APLICAÇÃO

Para este projeto foram utilizado as linguagens de programação, Javascript, HTML e CSS.

O JS ou Javascript é uma linguagem de programação de alto-nível criada no meio da década de 90, mais precisamente em 1996 pelo lendário programador Brendan Eich que, além de criar Javascript, foi também um dos fundadores da Mozilla Corporation (Carlos E. ago 19, 2021. Site hostinger.com.br).

Pouco tempo depois, cerca de 1 ano depois do seu lançamento, a Microsoft começou a aceitar esta linguagem em seu navegador da época, o que foi um dos grandes impulsionadores para consolidar o que é Javascript como uma das principais linguagens de programação dos dias de hoje (Carlos E. ago 19, 2021. Site hostinger.com.br).

As definições a seguir de Javascript, HTML e CSS é baseada no trabalho de Carlos E. ago 19, 2021 para o site hostinger.com.br.

Javascript é a terceira camada do bolo de desenvolvimento web e front-end, junto com HTML, CSS e PHP.

HTML é uma linguagem de marcação que serve para dar significado e estrutura ao conteúdo web, como, por exemplo, definir cabeçalhos, parágrafos, fazer referências ou inserir imagens ou vídeos.

CSS é uma linguagem de folha de estilos, que tem o papel de tornar uma página apresentável na web, relacionada diretamente com o design e aparência (Carlos E. ago 19, 2021).

Neste projeto foi usado uma aplicação cliente servidor utilizado o Node.js como servidor.

O Node.js pode ser definido como um ambiente de execução Javascript server-side (lado servidor). Isso significa que com o Node.js é possível criar aplicações Javascript para rodar como uma aplicação standalone (rodar sozinho) em uma máquina, não dependendo de um browser (navegador web) para a execução, como estamos acostumados. (Node.js – O que é, como funciona e quais as vantagens Ienon, 05/09/2018, site opus-software.com.br).

Apesar de recente, o Node.js já é utilizado por grandes empresas no mercado de tecnologia, como Netflix, Uber e LinkedIn (Node.js – O que é, como funciona e quais as vantagens Ienon, 05/09/2018, site opus-software.com.br).

O principal de sua adoção é a sua alta capacidade de escala. Além disso, sua arquitetura, flexibilidade e baixo custo, o tornam uma boa escolha para implementação de Microserviços e componentes da arquitetura Serverless (sem servidor). Inclusive, os principais fornecedores de

produtos e serviços Cloud (nuvem) já têm suporte para desenvolvimento de soluções escaláveis utilizando Node.js (Node.js – O que é, como funciona e quais as vantagens Ienon, 05/09/2018, site opus-software.com.br).

A principal característica que diferencia o Node.js de outras tecnologias, como o PHP, Java, C#, é o fato de sua execução ser single-thread (único fio) Serverless. Ou seja, uma thread é responsável por executar o código Javascript da aplicação, enquanto que nas outras linguagens a execução é multi_thread (vários fios) (Node.js – O que é, como funciona e quais as vantagens Ienon, 05/09/2018, site opus-software.com.br).

Como um ambiente de execução assíncrono orientado a objetos, o Node.js é projetado para desenvolvimento de aplicações escaláveis de rede. No exemplo a seguir, diversas conexões podem ser controladas ao mesmo tempo. Em cada conexão a função de callback (chamada de volta) é chamada. Mas, se não houver trabalho a ser realizado, o Node.js ficará inativo (Disponível em <https://nodejs.org/pt-br/about/>, visualizado, set/2021).

Figura 17: Servidor Node.js (Hello World)

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Fonte: <https://nodejs.org/pt-br/about/>

Essa é uma alternativa que contrasta com o modelo de concorrência mais comum, onde são utilizadas threads do SO. Aplicações de rede baseadas em threads são relativamente ineficientes e difíceis de usar. Além disso, os usuários do Node.js não precisam se preocupar

com deadlocks (impasses) de processos, pois não existem locks (fechaduras). Quase nenhuma função no Node.js realiza diretamente operação de E/S, por essa razão o processo nunca bloqueia. Por não existirem operações bloqueantes, sistemas escaláveis são razoavelmente fáceis de serem desenvolvidos em Node.js (Disponível em <https://nodejs.org/pt-br/about/>, visualizado, set/2021).

Para o protótipo foi utilizado a linguagem HTML 5 com CSS 3 para a criação da página web. A aplicação é cliente servidor ambos implementados no Raspberry Pi 3, podendo o lado cliente ser acessado por outros dispositivos da rede local apenas digitando o IP da Raspberry na porta 8090 em um Browser (navegador) como por exemplo, no caso deste projeto digitando 192.168.1.5:9080 em um browser (navegador) de qualquer dispositivo conectado na mesma rede local em que se encontra o Raspberry é possível acessar o lado cliente da aplicação.

A aplicação é uma página HTML e possui um check-box ou caixa de seleção que ao marcar aciona a tomada e ao desmarcar desativa tomada. Ou seja, quando marcado, é enviado o estado (ativo) para o servidor, este que está esperando as conexões, verifica se é **true** (ativo) ou **false** (desativo), em caso de **true** (verdadeiro), aciona o pino 12 (GPIO18) da Raspberry, enviando um comando ao módulo relé para ligar ou, desligar em caso de **false** (falso).

Essa comunicação é feita utilizando a biblioteca socket.io versão 4.x.

Socket.io é uma biblioteca que permite a comunicação em tempo real, bidirecional e baseada em eventos entre o navegador e o servidor. Isso consiste de:

- . um servidor Node.js:
- . uma biblioteca cliente Javascript para o navegador (que também pode ser executada em Node.js) (O que é socket.io, site socket.io/docs/v4/).

As outras bibliotecas usadas foram rpio, express e onoff que estão dentro do NPM.

NPM é o maior registro de software do mundo. Os desenvolvedores de código aberto de todos os continentes usam o NPM para compartilhar e pegar pacotes emprestados, e muitas organizações também usam o NPM para gerenciar o desenvolvimento privado (npm Docs, site [npm.js](https://npm.js.org/)).

Rpio 2.4.2 este é um complemento Node.js de alto desempenho que fornece acesso às interfaces GPIO Raspberry Pi e SunXi (Allwinner V40), suportando GPIO regular, bem como I²C, PWN e SPI (rpio, site [npmjs.com](https://npm.js.org/)).

Express é uma estrutura simples para Node, seu principal recurso é roteamento robusto (Express 4.17.1, site npmjs.com).

Supondo que eu tenha a seguinte estrutura de pastas Server → public e na pasta Server eu tenho o server.js e nas pasta public eu tenho o HTML e CSS da página web lado cliente, no código do servidor eu tenho que criar uma rota para que ao rodar o servidor seja possível carregar os arquivos externos, por exemplo;

const app = express(); Onde app é uma variável recebendo a função **express();**

app.use(express.static('public')); Aqui é passado a variável app para usar o caminho da pasta public.

app.get("/", (req,res)=>{ res.render("index");}); Nesse trecho é chamado o index.html que esta dentro de public, caso eu não aponte o caminho usando o app.use, não será carregado o index.html.

Onoff 5.0 Acesso ao GPIO e detecção de interrupção com Node.js em placas como Raspberry Pi ou BeagleBone (Onoff 5, site npmjs.com).

Instalando os componentes:

Primeiro passo foi instalar o Node.js e as dependências citadas, no terminal linux, como o comando **sudo apt install nodejs**, para o Nodejs, depois foi instalado o pacote NPM que contém as bibliotecas que o Nodejs usa. O pacote npm também instala o Node.js junto, mas por questão de versão é melhor instalar o Node.js primeiro e depois o NPM. O comando para instalar NPM é **sudo apt install npm**.

Para o projeto a versão do Node.js usada é v10.24.0 e o npm 7.24.0. Com esses dois instalados, criamos a pasta server e abrimos esta pasta no terminal linux para incluir os pacotes socket.io, express, rpio, onoff, e algum outro caso seja necessário, para isso no terminal inserimos os comandos **sudo npm install socket.io rpio --save express onoff**, estes itens serão instalados e uma nova pasta será criada dentro de server a pasta node_modules onde estes módulos ficaram a disposição para usar no código.

Voltando para a aplicação, a caixa de seleção ou checkbox possui apenas dois estados checado ou não checado, fazendo o mesmo papel que um botão de liga e desliga faria, quando esta checado ou acionado envia a mensagem **checkedTrue** ao servidor que por meio do socket que emite on para o servidor afim ligar e mudar o estado do pino usando o comando **rpio.open(LED, rpio.OUTPUT, rpio.LOW)**, ou seja abre o pino 12 (variável LED), do tipo

saída e aciona, quando o estado passar para **checkedFalse** o socket emite off para o servidor desligar e mudar o estado do pino usando **rpio.open(LED, rpio.INPUT, rpio.LOW)**, sendo que, abre o pino 12 (LED) tipo entrada e desativa. Aqui foi a grande dificuldade do projeto eu estava trabalhando apenas com **rpio.OUTPUT** que fazia o módulo relé ficar sempre ativo e tentava mudar o estado usando **rpio.open(LED, rpio.OUTPUT, rpio.LOW)** trocando apenas o último parâmetro de **rpio.LOW** para **rpio.HIGH**, mas depois de ver algumas referências percebi que tinha que mudar a forma fazendo uma chamada de saída para ligar e uma de entrada para desligar, alterando apenas o segundo parâmetro da função **rpio.OUTPUT** para ligar e **rpio.INPUT** para desligar.

A figura 12 abaixo mostra a interface do projeto. Na figura está sendo usado o IP local do Raspberry na porta 8090, podendo ser acessado por todos os dispositivos conectados na rede local apenas inserindo no navegador o endereço `http://192.168.1.5:8090`, na figura a tomada está desligada.

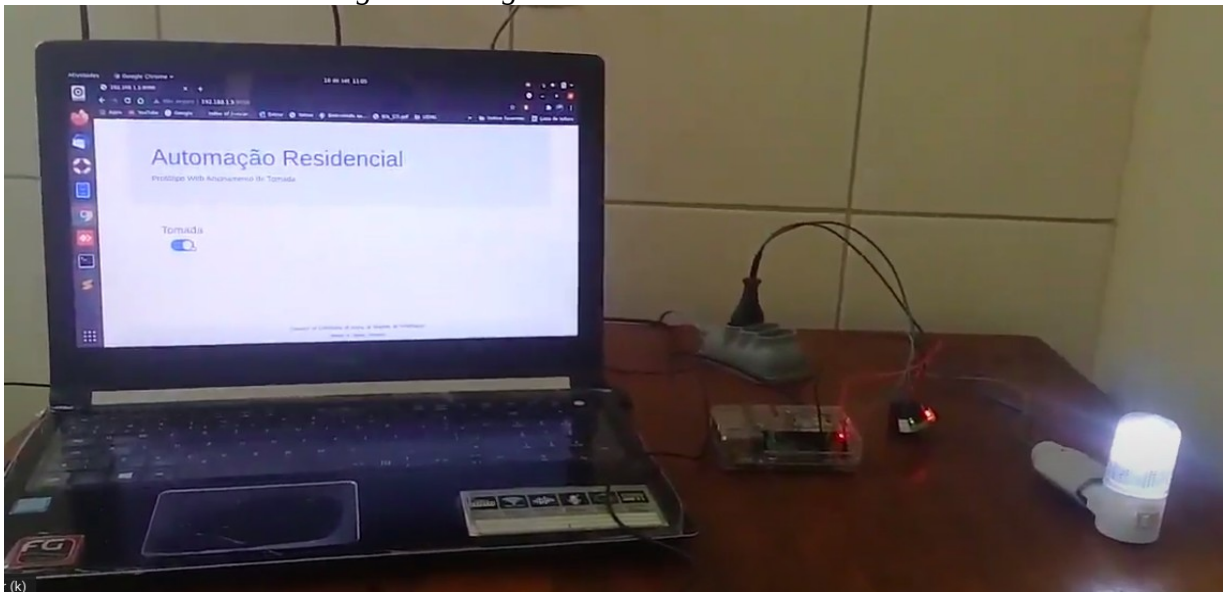
Figura 18: Interface do Protótipo Web de Acionamento de Tomada



Fonte: Autor

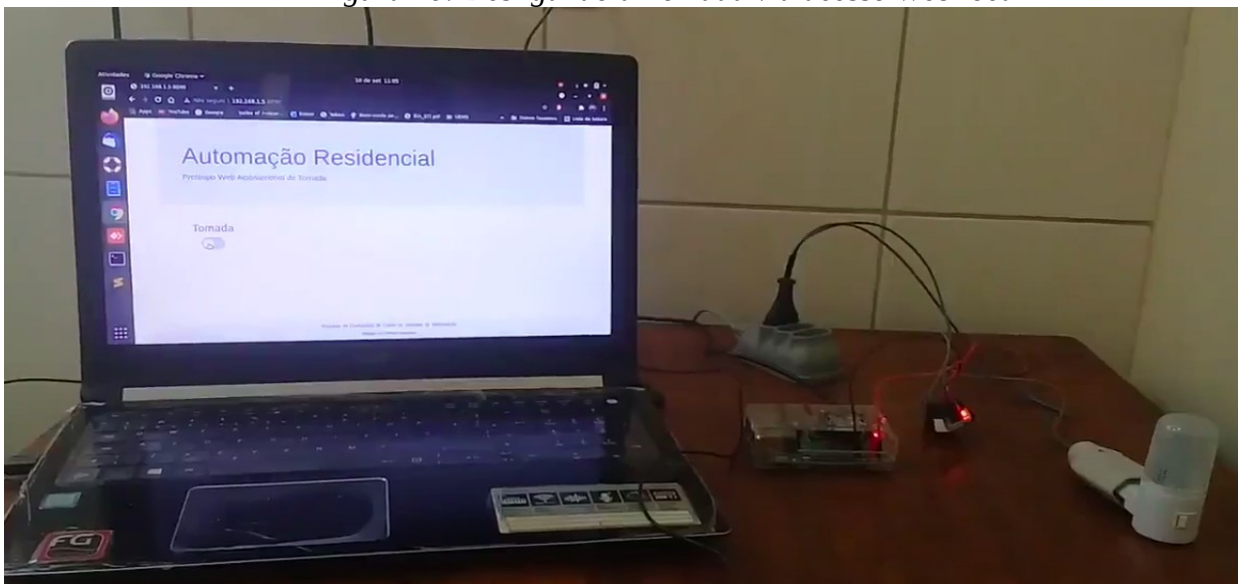
As figuras 13 e 14 demonstram o comportamento ao acionar o checkbox pelo navegador, ligando e desligando a tomada. A figura 13 mostra o estado ligado e a figura 14 mostra o estado desligado. Utilizando um notebook conectado à mesma Wi-Fi, emitindo os comandos através da página web para o servidor Raspberry que recebe os comandos e passam ao microcontrolador.

Figura 19: Ligando a Tomada via acesso Web local



Fonte: Autor

Figura 20: Desligando a Tomada via acesso Web local

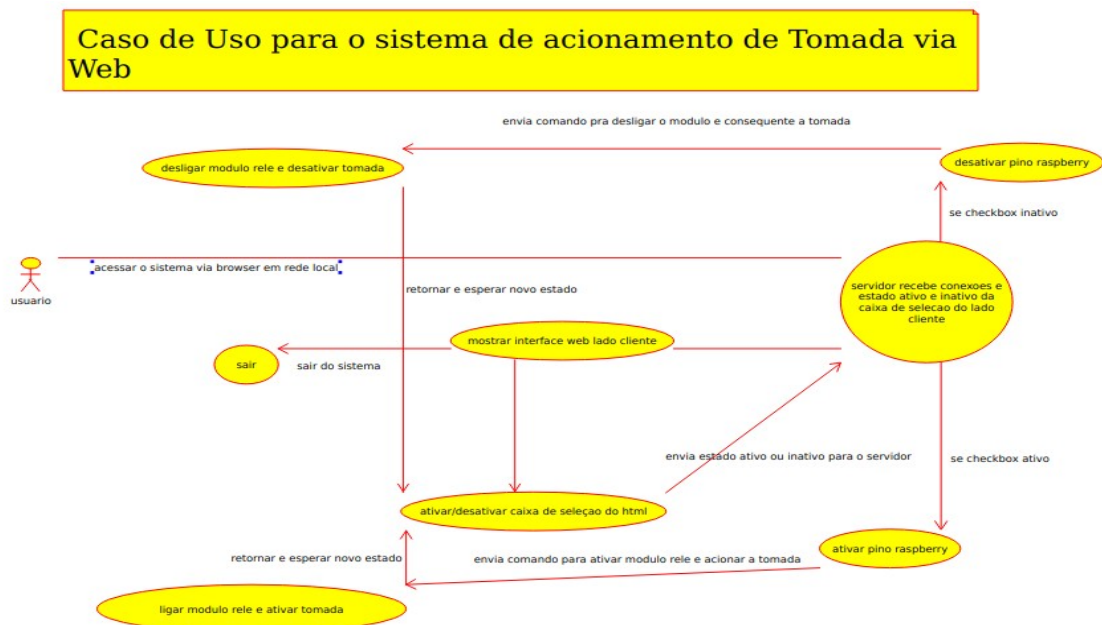


Fonte: Autor

Para o projeto foi usado uma pasta central (server) com o arquivo *server.js*, com as dependências que ele utiliza, para a parte Web uma pasta *public* contendo os códigos *index.html*, corpo da página, uma folha de estilos *styles.css* que estiliza a página e um arquivo *index.js* que atua no lado cliente estabelecendo as comunicações com o servidor.

A figura 15 a seguir, mostra o caso de uso do sistema de Automação Residencial mostrando o comportamento do sistema a partir da interação com o usuário.

Figura 21: Caso de Uso do Sistema de Automação Residencial usando Tomada Elétrica



Fonte: Autor

4 CONCLUSÃO

A execução deste trabalho proporcionou uma maior percepção sobre a automação residencial e as grandes possibilidades que o Raspberry pode oferecer, bem como a simplicidade de implementar programas web, por proporcionar uma gama diversificada de ferramentas e grande quantidade de bibliotecas para serem utilizadas.

A maior dificuldade deste trabalho foi a interação do servidor com o hardware do Raspberry, sendo preciso aprofundar estudos em formas de utilização dos pinos Gpios, bibliotecas para esse fim e também a forma que estes pinos se comportam como entrada e saída de dados. Para esse fim, tomamos alguns cuidados a mais com os componentes afim de se evitar danos ou falhas nos controladores, bem como no uso da Raspberry com case protetora e uma adaptação para o módulo relé afim de evitar que os contatos elétricos das placas entrassem em contato com a superfície.

O objetivos foram alcançados, no levantamento de requisitos utilizando como base a ideia simples de acionamento de tomadas, depois a comunicação entre máquinas servidor para controlador e depois com a aplicação web para o usuário final interagir com a tomada pelo web se utilizando de navegador de internet.

Uma característica deste trabalho é a sua expansão, pois ainda pode ser implementado diversos outros componentes utilizando módulos relés já com placa de rede embutida o tornando independente do Raspberry depois de configurado, desta forma, este trabalho pode ser o início para outros trabalhos mais elaborados usando melhorias como o uso de hardwares mais elaborados.

Por fim, a automação traz comodidade e impacta na vida do usuário oferecendo a simplicidade em controlar itens da casa pelo simples clique em um botão no seu dispositivo, aumentando a comodidade do lar.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALVES**, Paulo para o Tech Tudo 16/11/2017. Mini Pc Chega Oficialmente ao Brasil e ganha cor Exclusiva, Disponível em: <https://www.techtudo.com.br/noticias/2017/11/minipc-raspberry-pi-3-chega-oficialmente-ao-brasil-e-ganha-cor-exclusiva.ghtml>, acessado por último em 26-09-2021.
- BERTOLETI**, Pedro. Arduino x Raspberry Pi: Quais as diferenças entre as placas? Site filipeflop.com, 18 de dezembro 2020. Disponível em: <https://www.filipeflop.com/blog/arduino-x-raspberry-pi-quais-as-diferencas-entre-as-placas/>, acessado por último dia 03/12/2021.
- BIJANA**, L. Risteka Stojkoska*, KIRE, V. Trivodaliev, A review of Internet of Things for smart home: Challenges and solutions. Journal of Cleaner Production. Artigo Elsevier.
- Computação Brasil Edição 29**, 04/2015, Revista da Sociedade Brasileira de Computação, Disponível em: <https://www.sbc.org.br/component/flippingbook/book/23>, acessado por último dia 10/09/2021.
- FACCIONI**, Mauro Filho. Internet das Coisas (Internet of Things), UnisulVirtual, Palhoça, 2016. Livro.
- Introduction ao Raspberry Pi 3**, site, Disponível em: <https://www.theengineeringprojects.com/2018/04/introduction-to-raspberry-pi-3.html>, acessado por último em 10/09/2021.
- KOFUJI**, Sergio Takeo. Raspberry Pi 3 Modelo B. Disponível em: https://edisciplinas.usp.br/pluginfile.php/3444590/mod_resource/content/2/181-Raspberry%20Pi%203%20modelo%20B.pdf, acessado por último em 26-09-2021.
- KROTOV**, Vlad. The Internet of Things and new business opportunities. Business Horizons, 2017 ScienteDirect Disponível em: <https://www.sciencedirect.com/journal/business-horizons>, acessado por último 10/09/2021.
- MAGRANI**, Eduardo, Internet das Coisas. 1a edição, 2018 FGV Editora.
- MANCINI**, Mônica. Internet das Coisas: História, Conceitos, Aplicações e Desafios, artigo junho 2018. Disponível em

<https://www.researchgate.net/publication/326065859> Internet das Coisas Historia Conceitos Aplicacoes e Desafios, acessado por último 10/09/2021.

MISHRA, Harshvardhan. Control LED with Raspberry Pi using Nodejs. 1 Junho 2019. Disponível em: <https://iotbyhvm.ooo/control-led-with-raspberry-pi-using-nodejs/>, acessado por último dia 10/09/2021.

Modulo Relé – Acionando Cargas com Arduino, site Vida de Silício, Disponível em: <https://portal.vidadesilicio.com.br/modulo-rele-com-arduino/>, acessado por último em 10/09/2021.

MONK, Simon. O Raspberry Primeiros Passos com Python, Novatec, 2013.

SCHULTZ, Chistian. Acessando GPIOs da Raspberry Pi 3 por meio de um servidor web embarcado (node.js, express e wiringpi-node. Fevereiro 20, 2018. Disponível em: <https://www.embarcados.com.br/servidor-web-embarcado-com-node-js/>, acessado por último dia 10/09/2021.

Site Arduinoecia. Qual a diferença entre a Raspberry Pi modelo B e B+?, 30 de outubro de 2014. Disponível em: <https://www.arduinoecia.com.br/comparativo-raspberry-pi-b-plus/>, acessado por último dia 03/12/2021.

Site Hostinger, Carlos E. agosto, 19, 2021. O que é Javascript. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-e-javascript?ppc_campaign=google_performance_max&gclid=Cj0KCQjwwNWKBhDAARIsAJ8HkheovelgUheS1j2C1TcCvhQOiZ312hrTzoEKtuJpFVfRsqPwBXDi6XgaAm5OEALw_wcB, acessado por último em 30-09-2021.

Site NodeJs, Sobre Node.js Disponível em: <https://nodejs.org/pt-br/about/>, acessado por último 10/09/2021.

Site Opus-Software. Node.js – O que é, como funciona e quais vantagens. Lenon 05/09/2018. Disponível em: <https://www.opus-software.com.br/node-js/>, acessado por último em 30-09-2021.

Site Prototipando. Raspberry Pi GPIO o Básico Explicado. Disponível em: <https://labprototipando.com.br/2020/05/27/raspberry-pi-gpio-o-basico-explicado/>, acessado por último em 28-09-2021.

Site Raspberry. Documentação do Raspberry PI. Disponível em: <https://www.raspberrypi.org/documentation/computers/os.html>, acessado por último em 28-09-2021.

Site Robocore. Módulo Relé Arduino. Disponível em: <https://www.robocore.net/tutoriais/modulo-rele-arduino>, acessado por último em 28-09-2021.

TANEMBAUM, Andrew S., **WETHERALL**, David J., Redes de Computadores, 5a edição, Pearson, ebook.

TED, Saarikko*, **ULRIKA**, H. Westergren, **TOMAS**, Blonquist, The Internet of Things: are you ready for wat's coming? Artigo Business Horizons 2017, ScienceDirect. Disponível em: <https://www.sciencedirect.com/journal/business-horizons>, acessado por último dia 10/09/2021.

VIERO, Lucas Cardoso. Sistema de Automação Residencial Utilizando Rede Wifi Com Arduino e Wesite, 2016. Disponível em: <https://www.ufsm.br/app/uploads/sites/495/2019/05/2016-Lucas-Viero.pdf>, acessado por último dia 03/12/2021.

WATSON, Reginald. Control Raspberry PI GPIO with Node.js, May 30, 2019. Site Disponível em: <https://maker.pro/raspberry-pi/tutorial/how-to-control-a-raspberry-pi-gpio-pin-with-a-nodejs-web-server>, acessado por último dia 01/09/2021.

APÊNDICE A – CÓDIGOS DO PROJETO

Aqui é apresentado os códigos usados no trabalho, foi usado node.js e algumas dependências como socket.io e express.

Lado servidor arquivo server.js.

```
/*
```

```
Trabalho de Conclusão de Curso de Sistemas de Informação
```

```
Aluno: Magno de Oliveira Severino RGM 27687
```

```
Orientador: Andre Chastel de Lima
```

```
server.js lado servidor
```

```
*/
```

```
//Importando modulo express
```

```
const express = require('express');
```

```
const app = express();
```

```
//criando servidor
```

```
const server = require('http').Server(app);
```

```
//importando o socket.io pacote
```

```
const io = require('socket.io')(server);
```

```
const Gpio = require('onoff').Gpio;
```

```
const rpio = require('rpio'); //uso do rpio
```

```
const LED = 12;//pino usado no raspberry
```

```
//mapeando o caminho da pasta com os arquivos html, css e js
```

```
app.use(express.static('public'))
```

```
// chamando o index da pagina
```

```
app.get("/", (req,res)=>{
```

```
    res.render("index");
```

```
});
```

```
// função socket
io.on("connection" , (socket)=>{
  socket.on("checkedTrue" , ()=>{//se o checkbox estiver checado
    socket.emit("on")//emite on para ligar
    rpio.open(LED, rpio.OUTPUT, rpio.LOW); //aciona o modulo rele e liga a tomada

  })
  socket.on("checkedFalse", ()=>{
    socket.emit('off') //emite off para desligar
    rpio.open(LED, rpio.INPUT, rpio.LOW); //desliga o modulo rele e desliga a tomada

  })
})

// Executando o servidor na porta 8090
server.listen(8090 , ()=>{
  console.log("servidor em execucao porta 8090");
})

  Lado cliente:
  Index.html
<!--
  Trabalho de Conclusão de Curso de Sistemas de Informação
  Aluno: Magno de Oliveira Severino RGM 27687
  Orientador: Andre Chastel de Lima
  index.html lado cliente

-->
```



```

<!DOCTYPE html>
<html>
  <head>
    <!-- declarando dependencias e linkando arquivos externos-->
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel = "stylesheet" href =
"http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="style.css">
    <script src="socket.io/socket.io.js" defer></script>
    <script src="index.js" defer></script>
  </head>

  <body>
    <!--bloco para a primeira tomada-->
    <div class = "container">
<div class = "jumbotron">
  <h1>Automação Residencial</h1>
  <p>
    Protótipo Web Acionamento de Tomada
  </p>
</div>
<div class = "row">
  <div class = "col-md-3">
    <div id="tom1">
      <h2>Tomada</h2>
      <label class="switch">

```

<!--checkbox com a chamada da função para emitir o evento para o servidor-->

```
<input id="check" type="checkbox" onchange="changed()">
```

```
<span class="slider round"></span><br>
```

```
</label>
```

```
</div>
```

```
</div>
```

```
<!--rodapé da página-->
```

```
<div class="footer">
```

```
<h5>Trabalho de Conclusão de Curso de Sistema de Informação</h5>
```

```
<h6>Magno de Oliveira Severino</h6>
```

```
</div>
```

```
</body>
```

```
</html>
```

```
Style.css.
```

```
body {
```

```
color:blue;
```

```
/*background-color: steelblue;*/
```

```
}
```

```
/*checkbox*/
```

```
.switch {
```

```
    position: relative;
    display: inline-block;
    width: 60px;
    height: 34px;
  }
.switch input {
  opacity: 0;
  width: 0;
  height: 0;
}
.slider {
  position: absolute;
  cursor: pointer;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-color: #ccc;
  -webkit-transition: .4s;
  transition: .4s;
}
.slider:before {
  position: absolute;
  content: "";
  height: 26px;
  width: 26px;
  left: 4px;
  bottom: 4px;
  background-color: white;
  -webkit-transition: .4s;
  transition: .4s;
```

```
    }
input:checked + .slider {
    background-color: #2196F3;
}
input:focus + .slider {
    box-shadow: 0 0 1px #2196F3;
}
input:checked + .slider:before {
    -webkit-transform: translateX(26px);
    -ms-transform: translateX(26px);
    transform: translateX(26px);
}
.slider.round {
    border-radius: 34px;
}
.slider.round:before {
    border-radius: 50%;
}
/*divs*/
#tom1{
    color: black;
    text-align: center;
}
h2 {
    text-align: center;
    color: blue;
}
.row{
    background-color: white;
}
/*rodape*/
```

```
.footer {  
    position: fixed;  
    width: 93%;  
    bottom: 0px;  
    margin-right: auto;  
    margin-left: auto;  
    background-color:#eee;  
    text-align: center;
```

```
    }  
    index.js:
```

```
// inicializando o socket
```

```
const socket = io('/');
```

```
function changed()//função pra verificar o estado do checkbox e mandar pro servidor
```

```
{  
    // pegando estado atual do checkbox  
    var a = document.getElementById('check').checked;  
    console.log("Estatus", a); //mostra o estado  
    if(a)//verifica o estado do checkbox  
    {  
        // se esta marcado emite true para o servidor  
        socket.emit("checkedTrue");  
  
    }else{  
        // se não esta marcado emite false para o servidor  
        socket.emit("checkedFalse");  
  
    }  
}
```

```
}  
  
//apresentando o evento para teste  
socket.on("on" , ()=>{  
    //alert("Ligando"); // teste para checar o socket  
  
});  
socket.on("off" , ()=>{  
    //alert("Desligando"); // teste para checar o socket  
});
```