

Sistema de gerenciamento remoto de dispositivos residenciais

João P. C. Leite, Rodolfo F. Oliveira

Curso superior em Tecnologia em Análise e Desenvolvimento de Sistemas – Instituto Federal de São Paulo (IFSP)

j.leite@aluno.ifsp.edu.br, rodolfo.oliveira@ifsp.edu.br

Abstract. *Home Automation is a concept that defines the utilization of the Internet of Things (IoT) in social environments for controlling smart-devices. IoT, on the other hand, uses the principle of connecting electro-electronic devices to each other through a communication network. One of the main problems when treating Home Automation is the lack of open-source solutions and low cost devices that deliver what is needed. Starting from this point, this article proposes the prototyping of a system for controlling devices in a remote way in residences using the Message Queueing Telemetry Transport communication protocol, with a Web application for controlling the devices and an electronic system to manipulate the devices remotely. The objective of this paper is to develop a low-cost system, open-source and with ease of maintenance for home automation systems.*

Resumo. *A Domótica é um conceito que define a utilização de Internet das Coisas (IoT) em ambientes sociais para controle de dispositivos inteligentes. IoT, por sua vez, parte do princípio de conectar os dispositivos eletro-eletrônicos entre si através de uma rede de comunicação. Um dos principais problemas quando se trata da automação residencial é a falta de soluções abertas e dispositivos de baixo custo que entreguem o necessário. Partindo deste ponto, este artigo propõe a prototipação de um sistema para gerenciamento de dispositivos de forma remota em residências através do protocolo de comunicação Message Queueing Telemetry Transport, com uma aplicação Web para controle dos dispositivos e o sistema eletrônico para manipular os dispositivos de forma remota. O objetivo deste trabalho é desenvolver um sistema de baixo custo, open-source e de fácil manutenção para sistemas de automação residencial.*

1. Introdução

A presença de cada vez mais dispositivos inteligentes no cotidiano é um fato que não pode ser negado. Levando em consideração apenas o Brasil, em 2021 existiam 440 milhões de dispositivos digitais, cerca de dois dispositivos por habitante [FGV 2021]. Esses valores podem ser extrapolados ao se considerar a média mundial. O 'Insider Intelligence' estima que, em 2025, existirão cerca de 64 bilhões de dispositivos conectados no mundo [Yassine A. et. al 2019].

O desenvolvimento de dispositivos inteligentes está em um horizonte próximo. Com a chegada de tecnologias que aceleram as conexões sem fio, como o 5G, a tendência é aumentar cada vez mais a utilização de dispositivos inteligentes para tarefas antes feitas manualmente [Nguyen 2021].

A Figura 1 apresenta um gráfico de tendência do desenvolvimento de tecnologias pelos próximos 8 anos. Dos pontos ressaltados no artigo, vários se relacionam com o tema abordado neste trabalho, indicando uma tendência de aumento no desenvolvimento da domótica e Internet das Coisas, ou *Internet of Things (IoT)*, nos próximos anos.



Figura 1. Tendência de tecnologias nos próximos 8 anos
(Fonte: Gartner, 2021)

Na Figura, as tecnologias são apresentadas como pontos que representam, a partir do tamanho, o grau de adesão da tecnologia. Quanto mais próximas do centro, mais perto é a sua chegada ao mercado (de 0 a 1 ano) e quanto mais longe, mais distante (de 6 a 8 anos) [Nguyen 2021].

Esses dados confirmam que o desenvolvimento da Internet das Coisas se mantém em constante crescimento. Com o aumento de dispositivos inteligentes pode-se desenvolver formas de automatizar tarefas cotidianas e repetitivas em residências, como controlar lâmpadas, janelas, portas, cortinas, temperatura, entre outras. Visto que a IoT conecta os diversos dispositivos em uma rede, pode-se interligá-los a um dispositivo central gerenciador e construir residências inteligentes, que respondem às ações humanas ou podem ser programadas para realizar tarefas.

O trabalho propõe um sistema para controle remoto de dispositivos em residências, através do uso de um aplicativo e um sistema eletrônico para o controle dos dispositivos. O controle é feito com relés conectados a um módulo capaz de enviar os dados a um servidor que realiza o gerenciamento desses dados. O módulo envia os dados através do protocolo *Message Queuing Telemetry Transport* (MQTT) para o servidor. Por fim, o Servidor determina se o dispositivo está ligado ou desligado, além de armazenar as informações como estado do dispositivo e seu identificador, feito a partir de um banco de dados.

O presente trabalho está organizado da seguinte maneira: inicialmente, serão apresentados os conceitos principais do sistema proposto (Seção 2), sendo eles: Internet das Coisas (Seção 2.1), Domótica (Seção 2.2) e MQTT (Seção 2.3). Em sequência, são apresentados trabalhos correlatos (Seção 3), nos quais são explicadas suas soluções para os problemas apresentados. Na Seção 4 é detalhada a metodologia utilizada; na Seção 5 é apresentada a arquitetura do sistema e, na Seção 6 é especificada a prototipação do mesmo.

Na Seção 7 são apresentados os resultados obtidos e, por fim, na Seção 8 o artigo é concluído, e são sugeridos trabalhos futuros para melhoria deste protótipo.

2. Revisão Bibliográfica

2.1. Internet das Coisas

O conceito de Internet das Coisas está relacionado à conexão de dispositivos eletro-eletrônicos, tais como veículos, eletrodomésticos, sensores e máquinas diversas à Internet [Lima 2018]. Essa conexão permite monitorar e acionar os dispositivos de forma mais dinâmica, rápida e eficiente, gerando um controle mais preciso da produção, automação de tarefas e melhoria na coleta de dados. O uso dessa tecnologia deverá gerar um impacto econômico entre 3,9 e 11,1 trilhões de dólares por ano a partir de 2025, podendo reduzir consumos energéticos e aumentar a eficiência de trabalho. [McKinsey Global Institute 2015]

A tecnologia pode ser trazida para todos os ambientes, sejam eles industriais ou sociais. Um exemplo de aplicação dessa tecnologia é a domótica, em que são utilizados sensores e dispositivos inteligentes para automatizar tarefas repetitivas em uma residência [Muratori & Dal Bó 2011].

Sensores inteligentes são usados com frequência em aplicações IoT, podendo ser sensores de medição, RFID (*Radio-Frequency Identification*), entre outros. Esses sensores têm características que fazem com que a automação seja facilitada, como por exemplo pagamentos de pedágios de forma automatizada, entradas de condomínios com *tags*, medição em tempo real em usinas, etc [TOTVS 2021].

Outro exemplo de aplicação relacionada a IoT é o desenvolvimento de carros inteligentes, em que carros autônomos conseguem se comunicar com os dispositivos móveis, criar rotas otimizadas a partir da utilização de mapas para definir seu trajeto, buscar o trajeto mais rápido ou econômico, além de serem mais seguros [TOTVS 2021].

Por fim, um último exemplo de aplicação, a comunicação 5G. Ainda em fase de implementação no Brasil, essa tecnologia promete velocidades de transmissão de dados muito acima da atual 4G, A velocidade do 5G pode chegar a duas vezes a velocidade do de seu antecessor. Esse aumento na velocidade beneficia ainda mais o desenvolvimento da IoT, por entregar menos latência na transmissão e fornecer dados mais confiáveis e rápidos. Além disso, o 5G tem uma capacidade de conexões simultâneas maior, podendo associar muitos dispositivos em um mesmo espaço físico, permitindo assim maior conectividade em cidades, casas e até mesmo em indústrias [TOTVS 2021].

A Figura 2 representa de forma gráfica várias das possíveis aplicações de sistemas IoT, com exemplos de dispositivos que podem ser integrados a essa tecnologia na parte exterior do círculo, como geladeiras, iluminação, câmeras, aparelhos de som, entre outros. Na parte interna do círculo estão os dispositivos gerenciadores desse sistema, podendo ser celulares, relógios ou computadores.



Figura 2. Representação visual de dispositivos IoT

Fonte: (Gigatron, 2022)

2.2. Domótica

A domótica tem sua origem a partir da união de duas palavras, *Domus* (palavra em Latim para “Casa”) e robótica. Sua finalidade é desenvolver sistemas mecânicos em união a sistemas eletrônicos para automatizar tarefas residenciais repetitivas, como abrir janelas, ligar e desligar dispositivos, trancar e destrancar portas, entre outras [Muratori & Dal Bó 2011]. A automação desses sistemas é realizada a partir da comunicação dos dispositivos entre si e seguindo um programa desenvolvido para definir as tarefas que cada nó deve realizar.

“Domótica é a automatização e o controle aplicados à residência. Esta automatização e controle se realizam mediante o uso de equipamentos que dispõem de capacidade para se comunicar interativamente entre eles e com capacidade de seguir as instruções de um programa previamente estabelecido pelo usuário da residência e com possibilidades de alterações conforme seus interesses. Em consequência, a domótica permite maior qualidade de vida, reduz o trabalho doméstico, aumenta o bem-estar e a segurança, racionaliza o consumo de energia e, além disso, sua evolução permite oferecer continuamente novas aplicações” [Muratori & Dal Bó 2011].

A Figura 3 mostra uma representação visual de como a domótica pode ser utilizada em uma residência, com exemplos de dispositivos tecnológicos que podem ser automatizados e assim auxiliar em tarefas do dia a dia, como acender luzes, regular a intensidade de iluminação, controlar a temperatura, trancar portas à distância, entre outras.

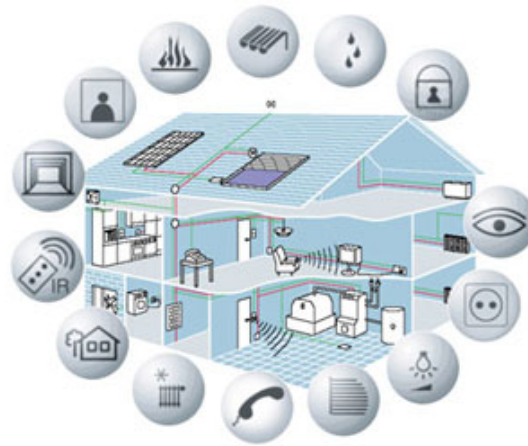


Figura 3. Representação Visual da domótica

Fonte: (<https://www.sislite.pt/domus.htm>, 2022)

A partir dessa definição, percebe-se que, para se ter uma residência automatizada, é necessário que existam sensores que coletam as informações necessárias para a tomada de decisão, aliados aos dispositivos de processamento, responsáveis por acionar os atuadores, que realizam a tarefa desejada. [Holtz 2022]

Alguns exemplos de aplicações na domótica são: acionamento de lâmpadas (de forma remota ou automática), sensores de presença para acionamento de dispositivos, ligar cafeteiras em horários programados, trancar portas e janelas à distância ou de forma automática, ligar para os bombeiros quando há incêndio ou para a polícia quando há uma invasão, entre outros [Cordella 2012 apud Silva 2021].

Os exemplos mencionados acima são demonstrações de aplicações da domótica que removem a necessidade do ser humano na realização de tarefas, trazendo benefícios como economia de tempo, energia elétrica e até mais segurança. A domótica pode ser separada em três grupos. O primeiro grupo associa as necessidades de segurança. Alguns exemplos relacionados à segurança podem ser um sistema de ligações automáticas em casos de invasão ou incêndio ou trancas automáticas. O segundo grupo trata do conforto ambiental, como controle de temperatura, umidade, controle acústico, visual e olfativo. Exemplos desse grupo podem ser robôs aspiradores, controles de iluminação, reguladores de temperatura e reguladores de som, que reduzem sons para um limite tolerável ao ser humano. Por último, existe o grupo para automatizar atividades do ser humano, como criar listas de mercado por voz, *Smart TVs*, sensores de presença para ligar ou desligar lâmpadas, entre outros [Cordella 2012 apud Silva 2021].

2.3. MQTT

MQTT é um protocolo de comunicação de arquitetura Cliente e Servidor com a estrutura de *Publishers* (Publicadores) e *Subscribers* (Assinantes). Foi desenvolvido pela IBM no final dos anos 90 com o objetivo de ser leve e de fácil implementação, voltado para dispositivos com recursos computacionais limitados [IBM 2021]. É muito utilizado em projetos de comunicação de Máquinas para Máquinas (*Machine to Machine* - M2M) e IoT, onde uma baixa quantidade de dados por pacote é necessária e a comunicação é instável [HiveMQ 2020, p. 5].

A arquitetura utilizada pelo sistema possui duas estruturas principais, uma responsável por filtrar os dados que chegam ao sistema, decidindo qual cliente vai publicar e qual vai

receber a mensagem e manter o roteamento dos dispositivos conectados à ela (*Broker*) e os Clientes, que podem ser *Publishers* ou *Subscribers*, os atores do sistema, que geram e consomem os dados produzidos [HiveMQ 2020, p. 7].

A Figura 4 representa a estrutura de *Publish/Subscribe* utilizada no protocolo MQTT. A estrutura central é o *Broker* e as periféricas são os Clientes que enviam e recebem os dados.

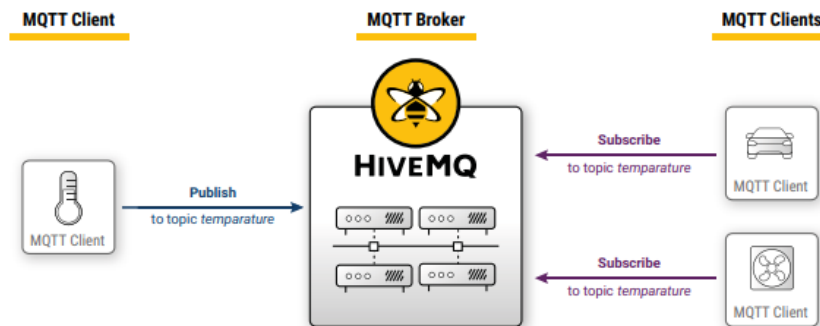


Figura 4. Estrutura da Arquitetura *Publish/Subscribe*

Fonte: (HiveMQ, 2022)

No protocolo MQTT e, seguindo a arquitetura apresentada na Figura 4, os clientes só precisam saber o endereço IP e a porta do *Broker*. Uma característica importante é a alta velocidade de entrega devido ao tamanho do pacote de dados [HiveMQ 2020 p. 9].

O protocolo funciona todo a partir de mensagens curtas e significativas, que determinam qual ação o *Broker* deve realizar para que ocorra a transmissão de dados. Como exemplo de mensagem do protocolo pode-se destacar o processo de conexão entre Cliente e *Broker*. Para iniciar a conexão, o cliente envia uma mensagem *CONNECT* e o *Broker*, ao receber a mensagem, retorna um *CONNACK*. A partir desse momento, os dispositivos estão conectados e, sempre que um Cliente enviar a mensagem *PUBLISH*, ele está enviando uma mensagem ao Broker, que por sua vez retorna uma mensagem *PUBACK* para reconhecer que a mensagem de publicação foi recebida. A conexão é finalizada ao enviar a mensagem de desconexão (*DISCONNECT*) para o *Broker*. A conexão é feita sempre entre um Cliente e o *Broker* e nunca pode ser feita diretamente entre dois Clientes.

3. Trabalhos Correlatos

Esta Seção apresenta alguns trabalhos que se assemelham ao tema abordado no artigo e foram utilizados como inspiração no desenvolvimento do mesmo. Tratam-se de Trabalhos de Conclusão de Curso de ex-alunos do IFSP Campus Hortolândia e da UNESP de Ilha Solteira.

Pereira (2018), apresenta uma solução de dispositivo IoT para o monitoramento de umidade do solo. Para isso foi desenvolvido um protótipo utilizando nós sensores com um NodeMCU para coleta de dados de sensores de umidade. Essas informações são enviadas para um banco de dados que as envia para um servidor *web* Apache, responsável por apresentar os dados. O projeto busca auxiliar na agricultura, com informações mais detalhadas do solo.

Cândido (2020) apresenta um projeto bem similar ao concebido neste artigo. Demonstra um protótipo de NodeMCU com sensores de temperatura para automatizar o controle de uma residência. O projeto contém uma estrutura muito similar ao primeiro trabalho mencionado, com *hardware* para aquisição de dados, um banco de dados para armazenar as informações enviadas pelos sensores e uma interface *web* para apresentação,

configuração e inclusão de sensores no sistema. Enquanto o projeto citado utiliza um sensor de temperatura para adquirir dados de temperatura, este artigo tem como objetivo utilizar relés para a atuação de lâmpadas, automatizando iluminações.

Holtz (2022), apresenta o desenvolvimento de um aplicativo para controle de dispositivos remotos em residências. Utiliza a plataforma *App Inventor* do MIT para criação do aplicativo de celular. O projeto desenvolveu um sistema de automação residencial de baixo custo e fácil implementação, utilizando um microcontrolador Arduino com módulos *bluetooth* para controle de lâmpadas e sensores.

Por fim, o trabalho desenvolvido por Costa (2019) apresenta um projeto de infraestrutura capaz de receber e processar dados com a estrutura de *Fog Computing*. Esse termo é utilizado para definir uma rede de distribuição de recursos e serviços de computação e dispositivos inteligentes ligados à rede, recebe esse nome devido à proximidade de serviços *Cloud* aos dispositivos inteligentes ligados à rede, fazendo uma analogia direta com a névoa (*Fog*) ser uma nuvem mais próxima ao solo. O artigo define a estrutura projetada e realiza a implementação de um sistema com PMS (*Proxy Manager Server*) responsável por realizar as tarefas relacionadas ao dispositivo, armazenar em um banco de dados e aplicar as configurações relacionadas ao PMW (*Proxy Manager Web*). O PMS foi desenvolvido em um Raspberry Pi 3, um mini computador utilizado frequentemente em sistemas embarcados. O PMW é responsável por gerenciar os vários PMS associados ao sistema e possui uma interface gráfica para configuração e gerenciamento dos dispositivos.

4. Metodologia

Analisando projetos desenvolvidos na área da IoT, uma das metodologias mais utilizadas é a Metodologia baseada em Protótipos. Nessa metodologia cria-se um produto inicial e então são feitos testes e validações que satisfaçam os requisitos definidos na fase de projeto. A partir de uma versão funcional, são feitas melhorias e/ou novos protótipos, podendo ser implementadas novas funcionalidades até que atinja um ponto satisfatório [DevMedia 2013].

Essa forma de desenvolvimento de projeto tem como principal vantagem a possibilidade de realizar testes a cada nova iteração. Essas implementações graduais permitem um rastreamento dos eventuais problemas ocorridos a cada nova funcionalidade, além de ser um bom termômetro para definir o grau de satisfação das pessoas em relação ao produto. A Figura 5 apresenta o ciclo de desenvolvimento de um projeto através da prototipação, com fases bem definidas que visam visitar o produto e fazer melhorias contínuas.

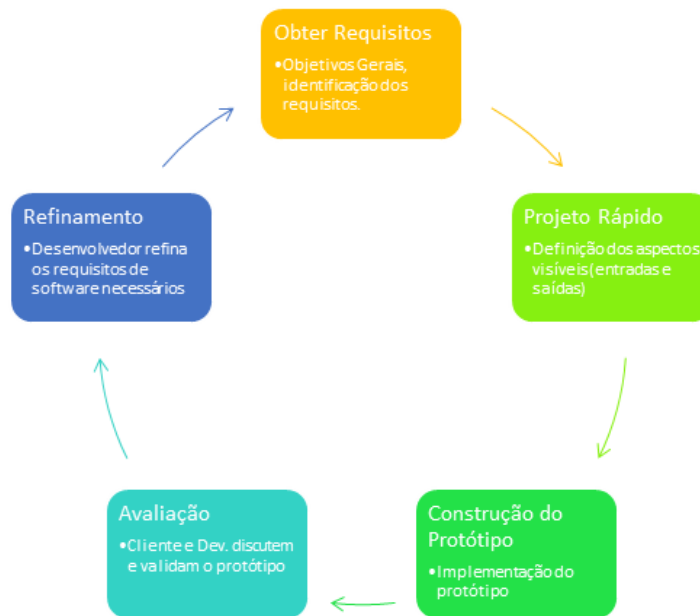


Figura 5. Etapas do Desenvolvimento de Protótipos

Fonte: (DevMedia, 2013)

5. Arquitetura do Sistema

Durante a etapa de concepção do projeto foi definida a arquitetura geral do sistema (Figura 6), dentro dela estão definidas todas as estruturas necessárias para seu funcionamento. O Sistema é separado em três partes principais, os Usuários, responsáveis pelo *input* de informações para o banco, além do controle dos objetos a partir da aplicação, o Sistema de Gerenciamento, onde estão todas as estruturas responsáveis pela inteligência do projeto e o controle no envio de informações do usuário para os Nós Sensores. Por fim, existem os Nós Sensores que representam os dispositivos conectados ao sistema e as estruturas que são controladas pelo Sistema de Gerenciamento a partir do comando do Usuário.

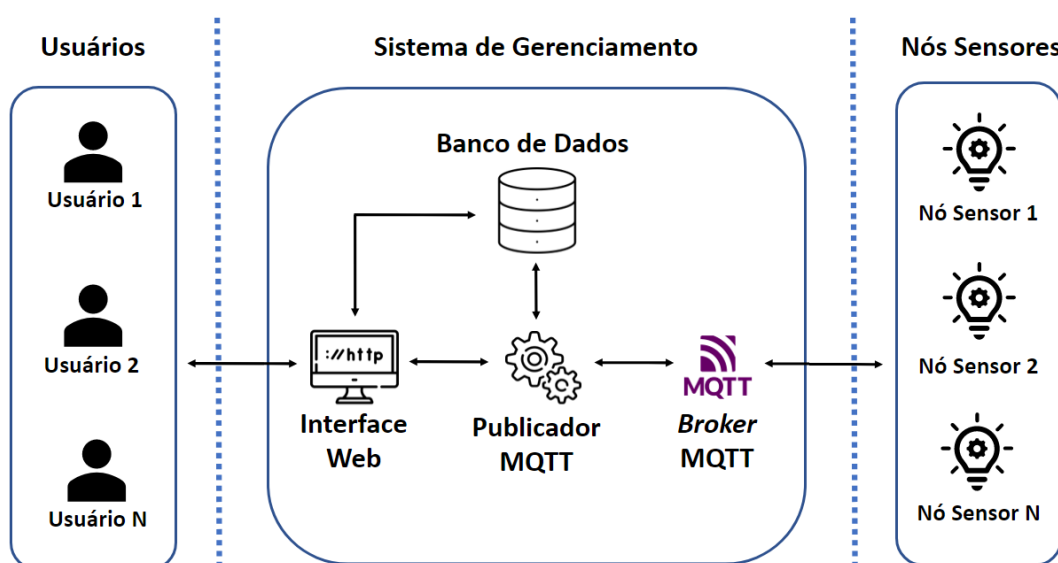


Figura 6. Arquitetura Geral do Sistema

Fonte: (Elaborado pelo autor, 2022)

Com a arquitetura pronta, foi desenvolvido um Diagrama de Caso de Uso, que define as responsabilidades que cada ator possui no sistema e quais funcionalidades cada um pode realizar. Os diagramas foram separados por ator, cada um com suas funções. Na Figura 7 podemos verificar as funções dos Usuários e na Figura 8, as funções dos Administradores.

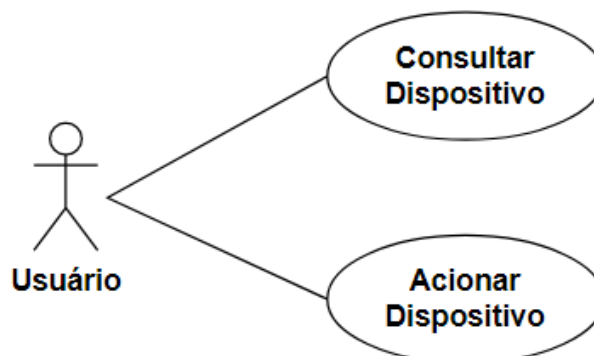


Figura 7. Diagrama de Caso de Uso (Usuário)

Fonte: (Elaborado pelo autor, 2022)

O Usuário tem como principais funções o controle dos dispositivos na residência. Para isso ele deve realizar consultas para verificar o estado dos dispositivos e acioná-los conforme a necessidade.

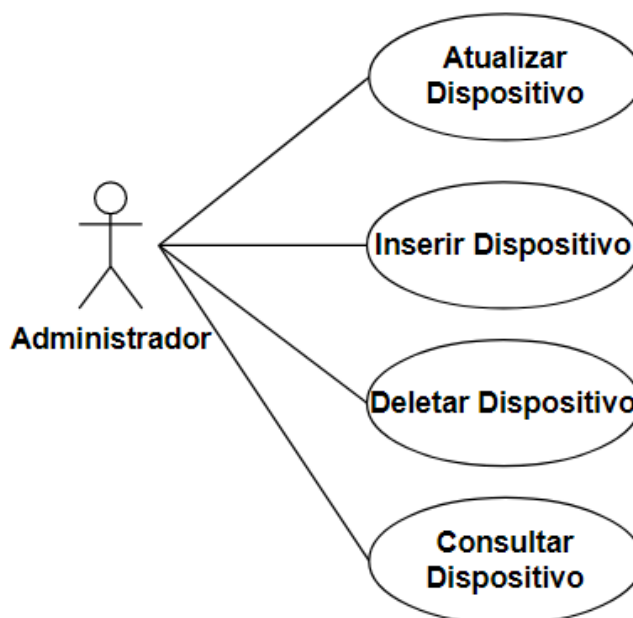


Figura 8. Diagrama de Caso de Uso (Administrador)

Fonte: (Elaborado pelo autor, 2022)

O Administrador é responsável por fazer a manutenção do sistema. Para isso, é necessário que ele tenha uma tela para fazer as operações básicas do sistema, também conhecido como CRUD (acrônimo para *Create, Read, Update, Delete*).

A Figura 9 apresenta o DER proposto, com as entidades participantes do sistema: Usuário, Dispositivo, Registro e Dispositivo x Usuário.

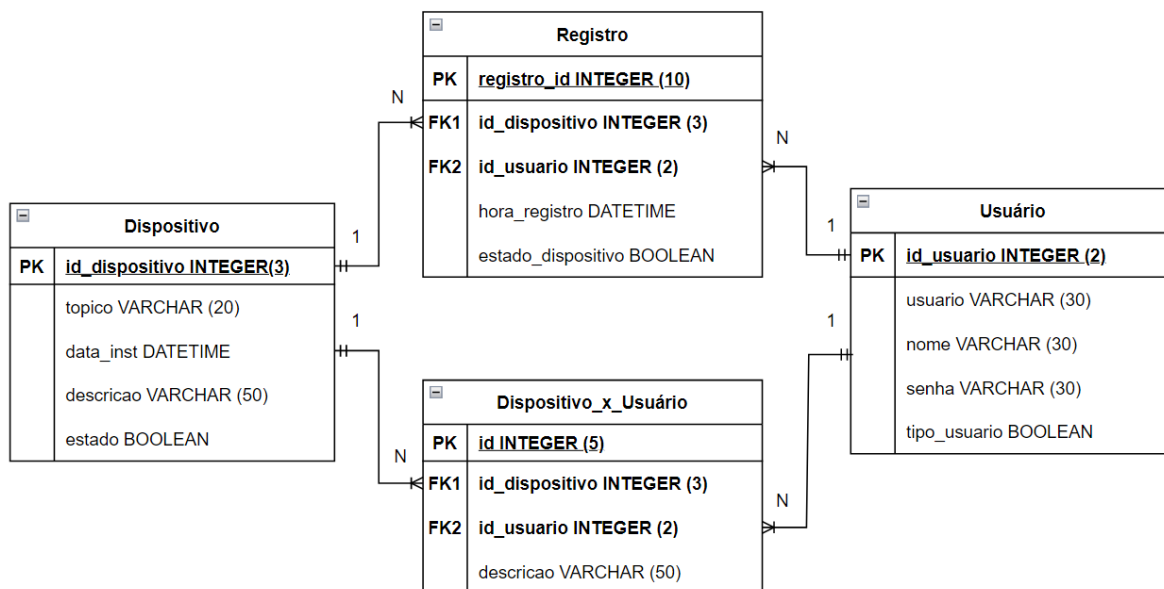


Figura 9. Diagrama Entidade-Relacionamento

Fonte: (Elaborado pelo autor, 2022)

A entidade “Dispositivo” é responsável por armazenar as informações relacionadas ao Nó Sensor, como o tópico (que representa a função do dispositivo para o *Broker*), quando foi instalado, descrição do dispositivo e o estado atual do dispositivo (ligado/desligado).

A entidade “Usuário” armazena informações de login (usuário e senha), o nome do usuário e o tipo de usuário, podendo este ser do tipo “1”, quando o usuário possui privilégios de administrador ou do tipo “0”, quando não tem. Os privilégios de administrador representam a possibilidade de cadastrar novos dispositivos, verificar o histórico ou dar permissão para outros usuários.

“Dispositivo_x_Usuário” é a entidade responsável por associar os dispositivos aos usuários, sendo assim, ela define qual usuário pode acionar qual dispositivo, funcionando como um controle de permissões para, por exemplo, controle parental. Ela possui duas chaves estrangeiras, uma vindo da tabela Dispositivo e outra da tabela Usuário, além de uma descrição à qual associação pertence.

Por fim, há a entidade “Registro”, que armazena um histórico dos acionamentos no sistema. Essa tabela registra quem foi o usuário que acionou, qual dispositivo foi utilizado, quando esse acionamento ocorreu e qual o estado que o dispositivo ficou.

6. Materiais e Métodos

Esta seção apresenta as principais tecnologias utilizadas no desenvolvimento do sistema, para isso, foi dividida em tópicos, que abordam os detalhes de cada subdivisão do projeto.

6.1. Banco de Dados

O banco de dados foi desenvolvido a partir do DER na Figura 9. Para esse projeto foram estudadas duas possibilidades de utilização de ferramenta para banco de dados. Inicialmente se pensou em utilizar o SQLite [SQLite 2022], por ser uma ferramenta leve e bem utilizada

em dispositivos embarcados, devido à eficiência em pequena escala e a facilidade de uso. A outra possibilidade era a utilização do MySQL [MySQL 2022], visando um sistema mais robusto.

A escolha foi feita visando o escalonamento do sistema, permitindo a escrita dos dados de forma simultânea, e visando maior segurança a partir do processo de autenticação já implementado desde o começo via MySQL.

6.2. Interface Web

A interface *Web* foi desenvolvida com linguagens comumente utilizadas para o desenvolvimento de páginas para a Internet. Para tanto, foi utilizado HTML, CSS e JavaScript (JS) para o *Front-End* (Cliente). O HTML é responsável pela estrutura da página [Mozilla 2022], o CSS pela apresentação e estilização dos elementos da página [Mozilla 2022], e o JS pelas funcionalidades e comportamento desses elementos, como por exemplo a ação de um botão [JavaScript 2022]. Além disso, foi utilizada a biblioteca *React* para o desenvolvimento de forma acelerada. A biblioteca traz diversas funções para páginas dinâmicas que diminuem o esforço necessário para criar aplicações responsivas [React 2022].

Para o desenvolvimento do *Back-End* (Servidor) foi utilizado o *Node.js*. É um ambiente de Javascript executado no *host* local (máquina instalada), dessa forma, o código implementado via Node pode ser acessado pelo sistema operacional e pode acessar estruturas como o banco de dados e o *broker* MQTT, que não devem ser acessadas pelo usuário no navegador [Node.js 2022]. No servidor foi implementada uma arquitetura REST, na qual são realizadas requisições HTTP do Cliente para o Servidor e são enviadas respostas como métodos *GET*, *POST*, *PUT* ou *DELETE*. Essas requisições são as responsáveis por fazer a comunicação entre a página *web*, banco de dados e *broker* MQTT [Lima 2020].

6.3. Nó Sensor

O sistema físico foi desenvolvido utilizando um NodeMCU conectado ao dispositivo que se desejava automatizar. Além disso, foi utilizado um relé para o acionamento do dispositivo.

O NodeMCU é uma plataforma de baixo custo para desenvolvimento de aplicações IoT, com ele é possível realizar comunicação entre partes do sistema com conexões sem fio. Utiliza de programação em Arduino, sendo ideal para sistemas embarcados, por ser um código leve e otimizado [NodeMCU 2018], como circuito apresentado na Figura 10.

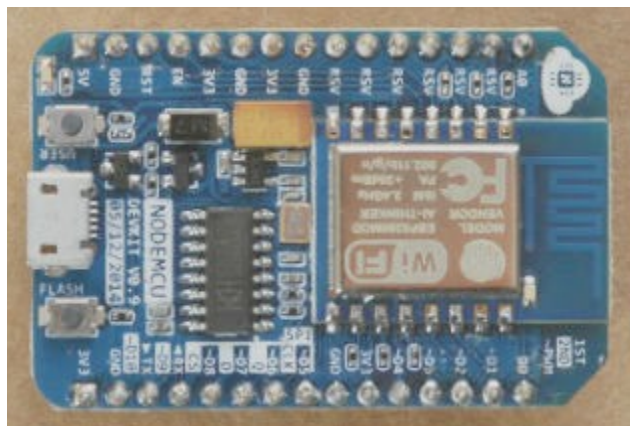


Figura 10. NodeMCU

Fonte: (NodeMCU, 2018)

6.4 *Broker* MQTT

O *broker* MQTT foi desenvolvido a partir da utilização do Eclipse Mosquitto, que implementa um servidor MQTT na máquina em que foi instalado. A API adiciona funcionalidades para conexão e envio de dados pelos dispositivos, a partir dela é possível conectar dispositivos neste servidor, através de linhas de comando que acessam um endereço provido, uma porta e o tópico que deseja publicar, como descrito no parágrafo seguinte [Mosquitto 2022].

Para execução do sistema é utilizado um programa para iniciar o servidor em uma instância local. Ao se inscrever para um tópico utilizando o comando “mosquitto_sub -h [endereço] -p [porta] -t [tópico]” e o *broker* retorna mensagens dizendo que a conexão foi aceita. Para enviar mensagens ao dispositivo inscrito ao tópico enviamos o comando “mosquitto_pub -h [endereço] -p [porta] -t [tópico] -m [mensagem]” [Mosquitto 2022].

7. Desenvolvimento

O desenvolvimento deu início com a criação de um banco de dados com as tabelas e colunas apresentadas na Figura 10.

Para o desenvolvimento da aplicação foi utilizada a metodologia de prototipação, portanto foram criados modelos iniciais de telas para visualização dos dados, e, a cada fim de ciclo de desenvolvimento eram feitas análises de melhorias para o sistema e implementadas novas alterações em um novo protótipo.

Inicialmente, foi criado um protótipo de interface *web* em que apenas as funções relacionadas a banco de dados estavam presentes, portanto, foram desenvolvidas telas responsáveis por inserir novos dispositivos ou usuários no banco de dados e para consultar os dispositivos presentes no banco.

Além dessa primeira interface, foi desenvolvido um protótipo físico do sistema para acionamento de dispositivos. Foram realizados testes com os componentes separados, como acionamentos “manuais” de relés (alternando os estados de forma estática) e testes de comunicação com o NodeMCU em rede local. Durante este período também foram realizados testes com o protocolo MQTT visando a futura integração da interface *Web* com o *hardware*.

Durante a fase de desenvolvimento do protótipo foi gerado um código em C++ a partir da utilização da IDE Arduino para realizar a comunicação entre o NodeMCU e o *broker* MQTT. A Figura 10 apresenta o fluxograma do código desenvolvido, apresentando o funcionamento de forma simplificada.

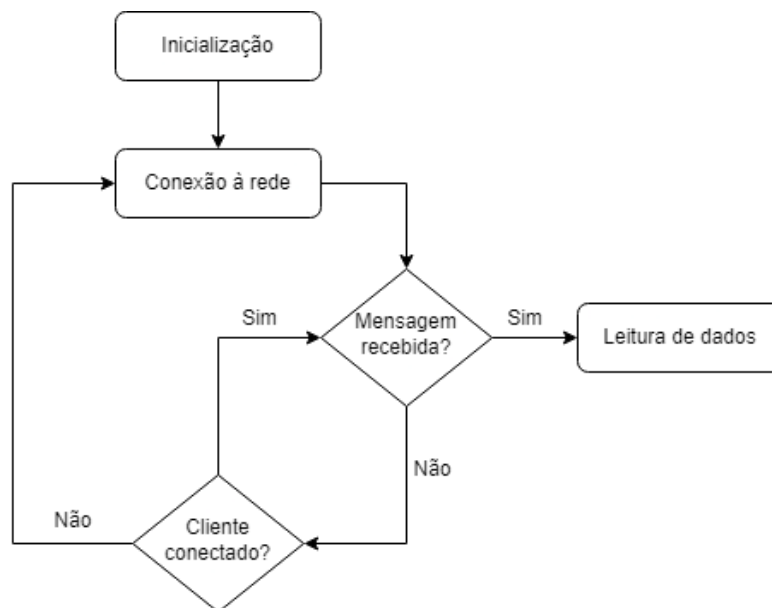


Figura 11. Fluxograma do Código desenvolvido

Fonte: (Elaborado pelo autor, 2022)

Comprovada a funcionalidade dos protótipos iniciais, deu-se início ao segundo protótipo da interface, onde foram criados métodos de comunicação entre o dispositivo físico (nó sensor) e a interface. Para isso, foi utilizada a biblioteca MQTT.js que implementa a comunicação MQTT em JavaScript, permitindo que sejam realizadas chamadas ao banco de dados, para determinar o estado atual do dispositivo e alterar o mesmo, publicando uma mensagem para o broker que dispara a mensagem para o dispositivo que se deseja acionar.

Para testar a comunicação entre dois dispositivos, foi criado um arquivo simples que simula um dispositivo e, ao rodá-lo através da linha de comando, foi possível receber as informações quando era pressionado o botão de acionamento na tela, sem nenhuma tratativa no funcionamento, apenas visualização dos dados, como apresentado na Figura 11.

```

[
  RowDataPacket {
    id_dispositivo: 7,
    topico: 'Luz',
    data_inst: '9-10-2022',
    descricao: 'Luz da casa',
    estado: 0
  }
]
C:\Users\Joao\Desktop\TCC>cd projtcc
C:\Users\Joao\Desktop\TCC\projtcc>cd src
C:\Users\Joao\Desktop\TCC\projtcc\src>cd server
C:\Users\Joao\Desktop\TCC\projtcc\src\server>node sub.js
Client has subscribed successfully
testing if it works tópic: Luz estado: 0

```

Figura 12. Resultado de acionamento (esquerda - Servidor | direita - Dispositivo Simulado)

Fonte: (Elaborado pelo autor, 2022)

O último protótipo da interface teve alterações relacionadas à comunicação com o dispositivo físico e adição de comportamento aos métodos implementados, fazendo o acionamento alterar o estado, por exemplo. Para isso, foram feitas alterações no código para remover código de simulação e as informações do dispositivo começam a vir do banco de dados, alterando o estado para 1 quando o dispositivo deve ser ligado e para 0 quando desligado.

8. Resultados Obtidos

Inicialmente serão apresentadas as principais telas criadas para controle da aplicação (Figuras de 13 a 16) e, em seguida, modelos representativos dos dois protótipos de nós sensores desenvolvidos (Figuras 17 e 18). Por fim, o protótipo que foi desenvolvido para os testes com a interface (Figura 19).

Cadastrar Usuário

Usuário	Nome Completo	Senha
<input type="text"/>	<input type="text"/>	<input type="text"/>
Tipo de Usuário		
<input type="text" value="Tipo de Usuário"/>		▼
		<input type="button" value="Cadastrar"/>

Figura 13. Tela de Cadastro de Usuários

Fonte: (Elaborado pelo autor, 2022)

Cadastrar Dispositivo

Tópico	Descrição
<input type="text"/>	<input type="text"/>
<input type="button" value="Cadastrar"/>	

Figura 14. Tela de Cadastro de Dispositivos

Fonte: (Elaborado pelo autor, 2022)

Consulta de Dispositivos

ID ↑↓	Dispositivo ↑↓	Estado ↑↓	Descrição ↑↓	Editar
4	Luz da Sala	Desligado	Luz para acender a lâmpada da sala	
7	Luz	Desligado	Luz da casa	
8	Luz do Quarto	Desligado	Luz para acender a lâmpada do quarto	

Figura 15. Tela de Consulta de Dispositivos (com acesso à edição)

Fonte: (Elaborado pelo autor, 2022)

Acionamento de Dispositivos

Dispositivo

Dispositivo ▼

Acionar

Luz da Sala
 Luz
 Luz do Quarto

Figura 16. Tela para Acionamento de Dispositivos

Fonte: (Elaborado pelo autor, 2022)

As Figuras 13 e 14 apresentam as telas responsáveis por cadastrar novos dispositivos e usuários ao banco de dados. A partir da inserção de novos dispositivos é possível consultá-los na tela apresentada na Figura 15 e acioná-los na tela da Figura 16. O campo de texto presente na Figura 16 é preenchido a partir de uma consulta SQL que traz os dispositivos cadastrados no banco e, ao pressionar o botão “Acionar” envia a mensagem para o *broker*, que disponibiliza para o dispositivo a mensagem que liga ou desliga a iluminação.

O 1º protótipo visava implementar o sistema de acionamento de forma simplificada, com o NodeMCU conectado a um Led para acionamento direto pela porta digital do NodeMCU, como apresentado na Figura 17.

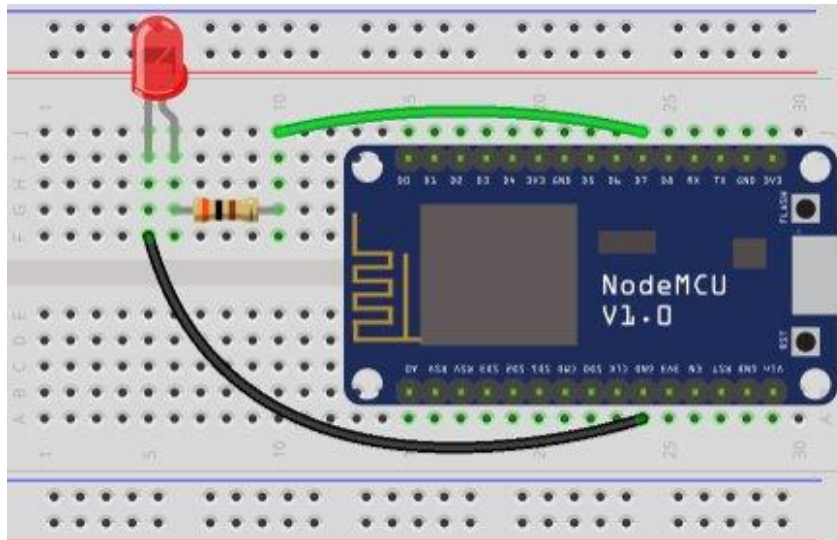


Figura 17. Representação do 1º Protótipo de Nó Sensor

Fonte: (Robocore, 2022)

O 2º protótipo implementa o sistema final, com a adição de um relé ao sistema, conectado à uma lâmpada (Figura 18). Dessa forma, o acionamento da porta digital fecha o contato do relé e energiza a lâmpada.

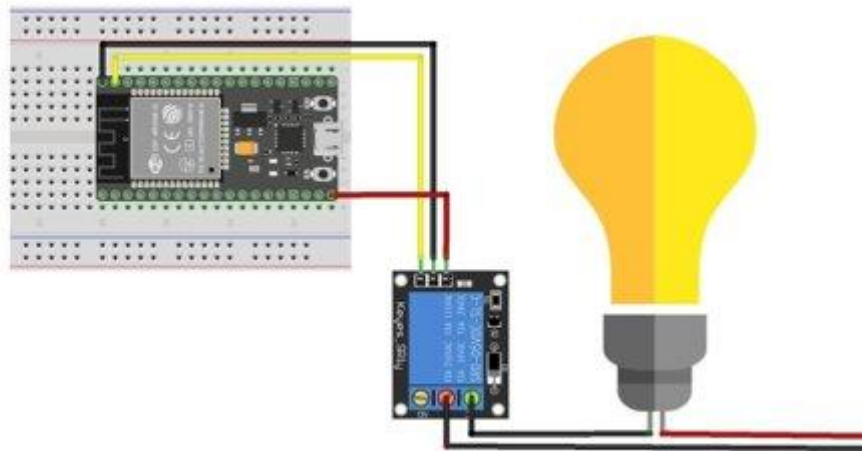


Figura 18. Representação do 2º Protótipo de Nó Sensor

Fonte: (UsinalInfo, 2019)

A Figura 19 apresenta o estado final do *hardware* desenvolvido para o projeto, conforme representação apresentada para o 2º protótipo.

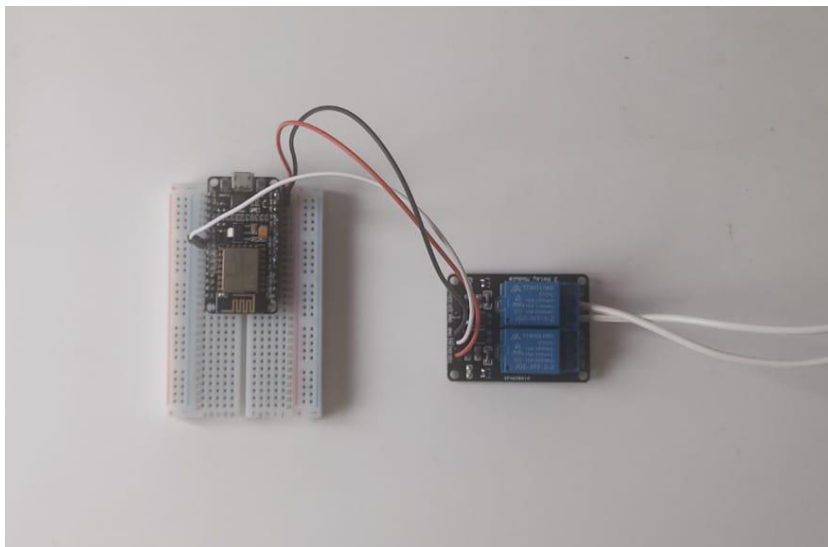


Figura 19. Protótipo Final

Fonte: (Elaborado pelo autor, 2022)

9. Conclusões

Como apresentado no início deste artigo, a proposta era o desenvolvimento de um sistema que realizasse a comunicação entre um dispositivo embarcado e uma aplicação que fosse capaz de resolver as requisições do usuário, possibilitando o acionamento de dispositivos à distância, como apresentado na seção Resultados Obtidos.

O projeto apresentou diversos desafios que ajudaram a consolidar o aprendizado durante o curso, além disso, foi uma oportunidade para desenvolver conhecimento em novas tecnologias muito utilizadas no mercado de trabalho atualmente, como o React como ferramenta de aceleração de desenvolvimento e a criação de aplicações para dispositivos embarcados, que são assuntos pouco abordados no curso como um todo.

Durante o desenvolvimento do sistema foram abordadas diversas áreas do conhecimento que compõem a grade curricular do curso. Os mais importantes foram Arquitetura de Sistemas, para a confecção da arquitetura do sistema e definição dos requisitos para o projeto, Desenvolvimento Web I e II, para o desenvolvimento da aplicação *web* que funciona como interface para o usuário interagir com o sistema, Banco de Dados I e II, para o desenvolvimento do banco que armazena as informações de cada dispositivo e usuário do sistema. Além disso, a disciplina eletiva de Sistemas Distribuídos contribuiu para o entendimento do protocolo de comunicação utilizado.

Como proposta de trabalhos futuros, o *design* é um ponto que não foi priorizado e, portanto, necessita da aplicação de conceitos de Interação Humano Computador para melhor interação com o usuário. A aplicação *Web* também não conta com um sistema de login, por isso não foi implementado um dos requisitos do projeto, de diferenciar as ações permitidas na interface por tipo de usuário. Outra proposta é o acionamento de grupos de dispositivos simultaneamente, por exemplo, diversas lâmpadas ao mesmo tempo. Finalmente, como proposta futura, existe a possibilidade do desenvolvimento de uma plataforma para gerenciamento dos dispositivos através de uma plataforma *mobile*, transpondo a interface *Web* para o *smartphone*.

10. Referências Bibliográficas

Cândido, M. S. & Oliveira, R. F. (2020). Automação Residencial utilizando uma Plataforma Open-Source.

Costa, E. P. & Oliveira, R. F. (2019). Gerenciador de Coleta de Dados para Redes de Sensores sem Fio.

DevMedia. (2013) “Introdução a prototipação e apresentação do Axure RP 6.5”. Disponível em:

<<https://www.devmedia.com.br/introducao-a-prototipacao-e-apresentacao-do-axure-rp-6-5/27978>> Acesso em Julho 2022.

FGV. (2021) “Brasil tem dois dispositivos digitais por habitante, revela pesquisa da FGV”. Disponível em:

<https://portal.fgv.br/noticias/brasil-tem-dois-dispositivos-digitais-habitante-revela-pesquisa-fgv?utm_source=portal-fgv&utm_medium=fgvnoticias&utm_campaign=fgvnoticias-2021-05-25> Acesso em Maio 2022.

Gigatron. (2021) “O que é IoT?” Disponível em:

<<https://www.gigatron.com.br/post/o-que-%C3%A9-iot>> Acesso em Junho 2022.

HiveMQ. (2020) “MQTT & MQTT 5 Essentials, A comprehensive overview of MQTT facts and features for beginners and experts alike”. Disponível em:

<<https://www.hivemq.com/downloads/hivemq-ebook-mqtt-essentials.pdf>> Acesso em Junho de 2022.

Holtz, I. O. A. Desenvolvimento de aplicativo baseado em arduino para domótica. Universidade Estadual Paulista (Unesp), 2022. Disponível em:

<<http://hdl.handle.net/11449/217221>>.

Hostinger. (2022). “SQLite vs MySQL – What’s the Difference”. Disponível em:

<<https://www.hostinger.com/tutorials/sqlite-vs-mysql-whats-the-difference/>> Acesso em Setembro 2022.

IBM. (2021). “Getting to know MQTT”. Disponível em:

<<https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/>> Acesso em Julho 2022.

Iqbal, S., Sharif, Z., Shahid M. A., & Muhammad, Z. A. (2021). *Internet-of-Things based Smart Home Automation System using Android Phone*.

Indústria 4.0: conceitos e fundamentos / organizado por José Benedito Sacomano... [et al.]; Alessandro Wendel Borges de Lima... [et al.]. São Paulo: Blucher, 2018. 182 p.: il.

JavaScript. (2022). “Learn JavaScript”. Disponível em: <<https://www.javascript.com/learn>> Acesso em: Agosto 2022.

Lima, G. (2020). “REST: Conceito e fundamentos”. Disponível em:

<[https://www.alura.com.br/artigos/rest-conceito-e-fundamentos#:~:text=REST\(Representational%20State%20Transfer%2C%20que,existentes%20da%20Web%20sem%20a](https://www.alura.com.br/artigos/rest-conceito-e-fundamentos#:~:text=REST(Representational%20State%20Transfer%2C%20que,existentes%20da%20Web%20sem%20a)> Acesso em Novembro 2022.

Mosquitto. (2022). “Eclipse Mosquitto™. An open source MQTT broker”. Disponível em:

<<https://mosquitto.org/>> Acesso em: Setembro 2022.

Mozilla. (2022). “CSS”. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>> Acesso em Agosto 2022.

- Mozilla. (2022). “HTML: Linguagem de Marcação de Hipertexto”. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>> Acesso em Agosto 2022.
- Muratori, J. R. & Dal Bó, P.H. (2011). Automação residencial: histórico, definições e conceitos. O Setor Elétrico, v. 62.
- MySQL. (2022). “MySQL :: MySQL Documentation”. Disponível em: <<https://dev.mysql.com/doc/>> Acesso em Novembro 2022.
- Nguyen, T. H. Gartner. (2021) “5 Impactful Technologies From the Gartner Emerging Technologies and Trends Impact Radar for 2022”. Disponível em: <<https://www.gartner.com/en/articles/5-impactful-technologies-from-the-gartner-emerging-technologies-and-trends-impact-radar-for-2022>> Acesso em Julho 2022.
- Node. (2022). “About Node.js®”. Disponível em: <<https://nodejs.org/en/about/>> Acesso em Novembro 2022.
- NodeMCU. (2018). “NodeMcu. Connect Things EASY”. Disponível em: <http://www.nodemcu.com/index_en> Acesso em Agosto 2022.
- Pereira, H. A. F. & Oliveira, R. F. (2018). Sistema de monitoramento de umidade de solo via web.
- React. (2022). “React. Uma biblioteca JavaScript para criar interfaces de usuário”. Disponível em: <<https://pt-br.reactjs.org/>> Acesso em Agosto 2022.
- Robocore. (2022). “Como programar o NodeMCU com Arduino IDE”. Disponível em: <<https://www.robocore.net/tutoriais/como-programar-nodemcu-arduino-ide>> Acesso em Outubro 2022.
- Silva, D. A. P. & Mata, G. A. (2021). O Uso da Tecnologia em Casas Inteligentes. Revista de Tecnologia Invest v. 5.
- SQLite. (2022). “SQLite Documentation”. Disponível em: <<https://www.sqlite.org/docs.html>> Acesso em Novembro 2022.
- TOTVS. (2021) “Internet das Coisas: o que é, surgimento, aplicações e impactos”. Disponível em: <<https://www.totvs.com/blog/inovacoes/aplicacoes-da-internet-das-coisas/>> Acesso em Junho 2022.
- UsinaInfo. (2019) “ESP32 Projeto Controlando um Relé”. Disponível em: <<https://www.usinainfo.com.br/blog/esp32-projeto-controlando-um-rele/>> Acesso em: Outubro 2022.
- Yassine, A., Singh, S., Hossain, M. S., & Muhammad, G. (2019). *IoT big data analytics for smart homes with fog and cloud computing. Future Generation Computer Systems*, 91, 563-573.

Documento Digitalizado Restrito

Artigo Final

Assunto: Artigo Final
Assinado por: Rodolfo Oliveira
Tipo do Documento: Relatório
Situação: Finalizado
Nível de Acesso: Restrito
Hipótese Legal: Direito Autoral - conservar a obra inédita (Art. 24, III, da Lei nº 9.610/1998)
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- **Rodolfo Francisco de Oliveira, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 19/12/2022 21:39:00.

Este documento foi armazenado no SUAP em 19/12/2022. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1191598

Código de Autenticação: 41c3ee937b

