

Controle e monitoramento de dispositivos elétricos pela internet em 3 níveis de aplicação

Magna M. V. da Silva.* Bruno de Lima.**
Fábio T. Brito***

Instituto Federal de Educação, Ciência e Tecnologia do Ceará, Maracanaú, CE, Brasil

*(e-mail: magnavitaliano@gmail.com).

***(e-mail: brunodelima.ldb@gmail.com)

***(e-mail: fabio@ifce.edu.br)

Abstract: In recent years the Internet of Things (IoT) technology has developed numerous solutions and applications on the market, however, taking in consideration the development, the first impression of this technology could generate uncertainties on which the best resources to use for the construction of the project. This work presents the development of three IoT models systems for controlling and monitoring electrical devices. It aims to present different load monitoring and control platforms, analysing their functionalities, communication protocols, reducing the programming time and costs. For this purpose, it has been developed three approaches: the first one has been developed for controlling and monitoring a single load, using a lamp as a model; The second one is a system for control the air-conditioning system in classrooms of an educational building at the IFCE Campus Maracanaú; The third one is a system for controlling and monitoring the air-conditioning system, lighting, access control and energy consumption monitoring in an educational laboratory at the IFCE Campus Maracanaú. According to this study, it can be noticed that using the MQTT protocol, it's possible to design a decentralized structure. However, the choice of protocol is associated with the user interface chosen and, in this case, the Blynk[®] has more advantages.

Resumo: Atualmente, a tecnologia *Internet* das Coisas (IoT) possui inúmeras soluções e aplicações existentes no mercado, entretanto, quando se fala em desenvolvimento, o primeiro contato com esta tecnologia pode gerar incertezas sobre quais os melhores recursos utilizar para o desenvolvimento do projeto. Neste sentido, este trabalho apresenta o desenvolvimento de três modelos de IoT para o controle e monitoramento de dispositivos elétricos. Com isso, objetiva-se apresentar diferentes plataformas de monitoramento e controle de cargas, analisando suas funcionalidades, protocolos de comunicação, diminuindo o tempo de programação e custos. Para isso, foram realizadas três abordagens: o primeiro é um sistema para controle e monitoramento de uma única carga, sendo utilizado de modelo o acionamento de uma lâmpada; O segundo é um sistema para controle de equipamentos de climatização de salas de aula de um bloco didático do IFCE *Campus* Maracanaú; O terceiro é um sistema para controle e monitoramento de climatização, iluminação, controle de acesso e leitura de consumo de energia em um laboratório didático do IFCE *Campus* Maracanaú. A partir desse estudo, observou-se que com o uso do protocolo MQTT torna-se possível projetar uma estrutura descentralizada. Porém, a escolha do protocolo está associada à interface do usuário escolhida e, neste caso, o Blynk[®] apresenta mais vantagens.

Keywords: Internet of Things; Building Automation; Supervisory; Protocols.

Palavras-chaves: Internet das coisas; Automação Predial; Supervisório; Protocolos.

1. INTRODUÇÃO

O uso da *internet* se torna cada vez mais essencial nas atividades pessoais e profissionais nas sociedades modernas e com isso é evidente o aumento significativo de uma variedade de dispositivos conectados à ela. O ecossistema *Internet* das Coisas - em inglês, *Internet of Things* (IoT) - pode ser entendido como uma grande rede de objetos físicos incorporados a sensores, *softwares* e outras tecnologias com o objetivo de conectar e trocar dados com outros dispositivos e sistemas pela *internet*. Diante disso, a tecnologia IoT exerce um importante papel. Além das inúmeras possibilidades de desenvolvimento de aplicações, a IoT também oferece ferramentas para coletar, armazenar, monitorar e analisar os

dados gerados pelos dispositivos utilizados pelo sistema (Oliveira; Silva,2020).

O grande crescimento das redes está permitindo a entrada de um maior número de dispositivos e, simultaneamente, os custos com a implementação de soluções no ecossistema IoT diminuíram. Em 2020, no Brasil, foi sancionada a Lei nº 14.108, por meio da qual são retiradas taxas de fiscalização, instalação e funcionamento de sistemas de comunicação máquina a máquina - em inglês, *machine to machine* (M2M). Com isso, torna-se ainda mais fácil sua difusão ao longo do país (Brasil, 2020). A projeção de receita com IoT no Brasil até o fim de 2021 é de US \$3,2 bilhões e o número de dispositivos conectados pode ultrapassar 416 milhões até

2023 (Colón, Navajas e Terry, 2019).

Entretanto, as inúmeras possibilidades existentes para a criação de um sistema com a arquitetura IoT causam dúvidas sobre quais *softwares*, *hardwares* e protocolos de comunicação são mais adequados para dar início a um estudo sobre esta tecnologia. Então, a atividade de avaliar cada funcionalidade, as diversas limitações e custos presentes em cada solução para atender os requisitos mínimos de um protótipo demanda um tempo considerável (Oliveira; Silva, 2020).

Nesse contexto, como forma de auxiliar no desenvolvimento de aplicações IoT e na tomada de decisões que esse processo exige, apresentam-se neste artigo três modelos de sistema de automação predial com uso dessa tecnologia em diferentes níveis de aplicação e com *hardware* mais acessível. A abordagem inclui a análise das plataformas de monitoramento Blynk[®], MQTT *Dashboard*, Node-RED[®] e dos protocolos de comunicação *Message Queuing Telemetry Transport* (MQTT) e *Hypertext Transfer Protocol* (HTTP). Para o controle dos periféricos, foram utilizadas duas placas de desenvolvimento com microcontroladores da Espressif, ESP8266 e ESP32.

2. INTERNET DAS COISAS

Esta seção aborda de maneira detalhada esse fenômeno conhecido como *internet* das coisas, a saber: definição, características e arquitetura.

2.1 Definição

O termo “*internet* das coisas” foi citado pela primeira vez em 1999 por Kevin Ashton quando trabalhava em um projeto que envolvia identificação por radiofrequência - em inglês, *Radio-Frequency Identification* (RFID) - para um laboratório do *Massachusetts Institute of Technology* (MIT). Para Ashton, era necessário que os computadores fossem capazes de entender o mundo sem depender da inserção de dados por humanos, fazendo isso através de sensores conectados a eles (Lopez Research LLC, 2013).

Definir de forma abrangente a IoT não é uma tarefa fácil, visto que cada entidade proponente apresentará uma visão particular baseada nos ativos da IoT que são considerados mais relevantes. Isto é, muitas definições se tornam tendenciosas por se restringirem aos ativos que aquele proponente busca enfatizar (Minerva; Biru; Rotondi, 2015).

Para a IBM (2021), o IoT se refere à crescente gama de dispositivos conectados que enviam dados pela internet, onde a “coisa” é qualquer objeto que possua algum componente eletrônico que lhe permita transferir dados através de uma rede, sem que haja interferência humana. Já a Microsoft[®] (2021) considera que IoT não se limita apenas a dispositivos conectados, mas também leva em consideração as informações poderosas e imediatas que eles fornecem. Para a empresa, possuir este tipo de dado confiável e em tempo real pode transformar negócios, reduzir custos, reduzir perdas ou materiais, melhorar processos operacionais e mecânicos ou até mesmo ajudar a construir novas linhas de negócios possíveis.

A União Internacional de Telecomunicações (2012) - em

inglês, *International Telecommunication Union* (ITU) - apresenta uma padronização técnica onde IoT é vista como uma infraestrutura global para a sociedade da informação, possibilitando serviços avançados interconectando coisas (físicas e virtuais) com base nas tecnologias de informação e comunicação interoperáveis existentes e em evolução. Nessa subdivisão do significado de “coisas”, têm-se:

- Coisas físicas: tudo aquilo que está no mundo físico que é capaz de sentir, pode ser atuado e conectado;
- Coisas virtuais: tudo aquilo que está no mundo da informação que pode ser armazenado, processado e acessado.

A “coisa física” pode estar interligada a uma ou mais “coisas virtuais” para realizar o seu mapeamento, porém a “coisa virtual” não necessariamente precisa estar relacionada a uma “coisa física” para existir. A ITU também apresenta um modelo que representa a dimensão da IoT como uma rede ubíqua, como pode ser observado na Figura 1.

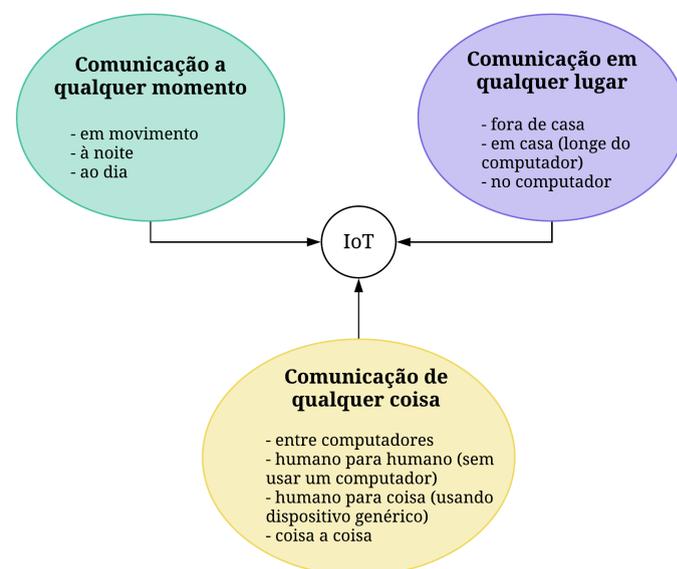


Fig. 1 Adaptação do modelo apresentado pela ITU para definição de IoT.

Fonte: Autoria própria (2021)

Por fim, é importante listar as principais características que um sistema IoT possui, sendo: (Minerva; Biru; Rotondi, 2015):

- Interconexão das coisas: o sistema deve tratar de interconectar qualquer objeto físico que seja relevante para o usuário ou aplicação;
- Conexão das coisas na internet: o sistema não restringe os objetos a redes particulares, como Intranet ou Extranet;
- Coisas identificáveis de forma única: o sistema é composto por objetos que possuem identificação individual;
- Ubiquidade: o sistema deve estar disponível sempre que for necessário a hora que for necessário;
- Capacidade de detecção ou atuação: o sistema possui sensores ou atuadores em seus objetos, tornando-os capazes de realizar ações independentes;

- Inteligência incorporada: os objetos inteligentes do sistema utilizam-se de seu conhecimento como ferramenta para analisar suas próprias operações;
- Interoperabilidade: o sistema utiliza-se de protocolos de comunicação padrão e interoperáveis;
- Autoconfiguração: os objetos do sistema devem realizar o gerenciamento de si mesmo, como suas configurações de software ou hardware e a utilização de recursos (energia, por exemplo);
- Programabilidade: os objetos podem alterar seu comportamento através de comandos dos usuários sem necessitar de mudanças nos seus componentes físicos.

2.2 Arquitetura

De acordo com Sethi (2017), não há um modelo padrão que caracterize um sistema IoT, porque também não há um consenso sobre isso. O mais comum é o modelo de três camadas, como apresentado na Figura 2, que define a ideia central desse sistema. Porém existem outros modelos com maior número de camadas quando há a necessidade de refinar os aspectos da pesquisa.

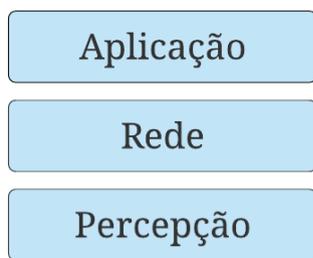


Fig. 2 Arquitetura IoT de 3 camadas.
 Fonte: Autoria própria (2021)

A camada de percepção, também chamada de camada de dispositivo, é a camada que representa os objetos físicos do sistema. Ela é responsável por interagir com o ambiente, coletando informações do mundo real através de sensores de forma automática e transmitindo essas informações pela rede, ou atuando com componentes mecânicos após receber comandos da nuvem (Villa-Henriksen et al, 2020).

A camada de rede é responsável por interligar as camadas de percepção e aplicação, transportando dados entre elas, utilizando-se de protocolos de comunicação. Ela também realiza o processamento e armazenamento dessas informações, agregando ou fundindo os dados coletados (Sassi; Jedidi; Fourati, 2019).

A camada de aplicação é a camada que está mais próxima do usuário final, oferecendo-o serviços inteligentes através da análise dos dados recebidos pela camada de rede. É nela que os dados realmente se tornam informações significativas (Villa-Henriksen et al, 2020).

3. TECNOLOGIAS APLICADAS

Nesta seção será possível compreender quais estruturas de *hardwares*, supervisórios e protocolos de comunicação são utilizados.

3.1 Hardware

3.1.1 Módulo Wi-Fi Esp-01

Na aplicação de primeiro nível, visando garantir um hardware mais reduzido e de menor custo, foi utilizado o módulo Wi-Fi ESP-01, Figura 3, onde seu *chip* principal é o ESP8266 com suporte ao padrão IEEE 802.11. Seu processador é de 32-bit com CPU de baixa potência e integrado ao mesmo há uma antena e conversor analógico digital - em inglês, *Analog-to-Digital Converter* (ADC) - com resolução de 10 bits. Ele possui 36 KB de RAM, porém não possui memória ROM, então o programa do usuário é armazenado na memória flash SPI externa de 1 MB que está integrada no módulo. Ele conta com duas portas de entrada e saída de uso geral e comunicação serial pelos pinos RX e TX, além de operar a uma tensão de 3,3V (AI-THINKER CO. LTD, 2015).

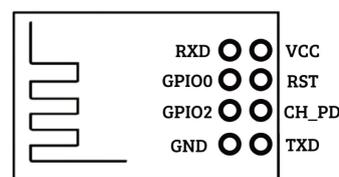


Fig. 3 Módulo Wi-Fi Esp-01.
 Fonte: Autoria própria (2021)

3.1.2 DOIT Esp32 DevKit v1

Nas aplicações de segundo e terceiro níveis, foi utilizado a interface de desenvolvimento DOIT Esp32 DevKit v1, apresentado na Figura 4. Seu *chip* principal é o ESP32 com suporte ao padrão de comunicação IEEE 802.11 e comunicação *Bluetooth* v4.2 BR/EDR e *Bluetooth Low Energy* (BLE). Este microcontrolador tem 448 KB de ROM, 520 KB de SRAM, 4 MB de Flash, 8 KB de RTC slow SRAM, 8 KB de RTC fast SRAM, 1 KB de eFuse e suporte a memória externa de até 16 MB de Flash e 16 MB de SRAM. Ele conta com 25 portas de entrada e saída de uso geral, entre elas há 18 canais ADC com resolução de 12 bits, 16 canais de saída *Pulse Width Modulation* (PWM), e 10 sensores capacitivos nas portas de uso geral. Este dispositivo opera com tensão de 5V (ESPRESSIF SYSTEMS CO. LTD, 2020).

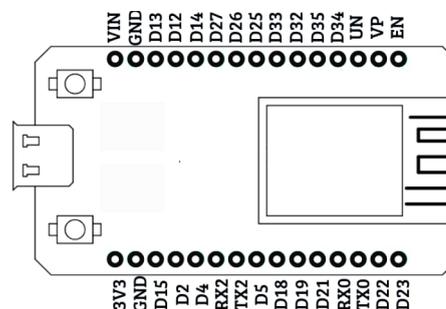


Fig. 4 Interface de desenvolvimento DOIT Esp32 DevKit v1.
 Fonte: Autoria própria (2021)

3.2 Protocolos

3.2.1 MQTT

Na aplicação de primeiro e terceiro nível foi utilizado o protocolo MQTT. Este protocolo é baseado na pilha TCP/IP e usa modelo publicação e assinatura para comunicação entre dispositivos. Por exemplo, um sensor (publicador) envia seus dados para um *broker*, responsável por armazenar e distribuir informações, e são identificados por um tópico. Um aplicativo (assinante) poderá consultar aquele tópico para receber os dados publicados. Ambas, escrita e leitura, são independentes, isto é, a comunicação é assíncrona. É um protocolo projetado para dispositivos e redes limitadas (Menezes et al, 2017).

3.2.2 HTTP

Na aplicação de segundo nível foi utilizado o protocolo HTTP. Este protocolo foi desenvolvido para realizar a transferência de arquivos hiper-mídia, como vídeos, imagens, textos e sons, que são lidos de forma não linear. Seu modo de comunicação é cliente-servidor, onde as informações são enviadas quando há uma requisição do cliente. Suas requisições são independentes, não compartilham informações entre si, e também é projetado para ser facilmente compreendido por humanos. A comunicação é realizada por métodos e entre eles os mais utilizados são: GET para capturar informações. POST para submeter informações que comumente realizam mudanças de estado, PUT para submeter informações que comumente são para atualizar e DELETE para deletar recursos específicos (Soares, 2019).

3.3 Supervisórios

3.3.1 Node-RED[®]

O Node-RED[®] é uma ferramenta visual de ambiente de código aberto originalmente desenvolvida pela IBM, mas que hoje faz parte da JS Foundation. Seu propósito é conectar dispositivos de *hardware*, *Application Programming Interface* (API) e serviços *on-line*, como é visto na Figura 5. Assim sendo, por meio dos *nodes* (ou nós) é possível ler arquivos CSV, escutar eventos HTTP, *websocket*, MQTT, entre outros. Sua instalação pode ser realizada em um servidor local ou em nuvem e seu editor de fluxos, que é a forma como ele é programado, é acessado através de um navegador (Node-RED, 2021). Seus nós permitem o envio, recebimento e processamento de dados, além de possibilitar o armazenamento dessas informações nos bancos de dados mais populares, sendo sua apresentação em painéis de forma agrupada ou não, sem limitação de quantidade de itens.

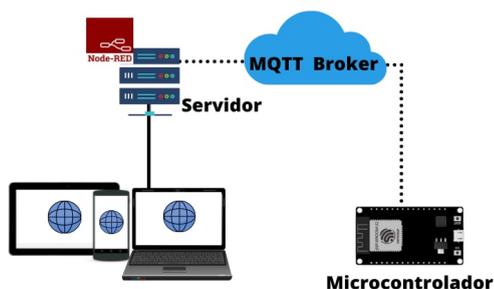


Fig. 5 Estrutura de um sistema usando Node-RED[®].
Fonte: Autoria própria (2021)

3.3.2 MQTT Dashboard

O MQTT Dashboard é uma ferramenta desenvolvida para realizar comunicação com dispositivos ou aplicações que estão conectadas a um *broker* MQTT, como visto na Figura 6. Ele pode tanto inscrever-se como publicar em um tópico criado pelo usuário e não há limites para inclusão de objetos no painel. Em seu painel, é possível adicionar botões, caixas de texto, *displays* e gráficos, além de tornar possível agrupar os itens em um contêiner. Porém, os gráficos criados são limitados a 20 pontos e não é possível recuperar esses dados posteriormente. Também, apesar de todas as demais funções serem gratuitas, não é possível adicionar mais de um servidor para criação de novos painéis. Ainda assim, o valor cobrado é uma taxa única e o custo dessa atualização é significativamente muito baixo.

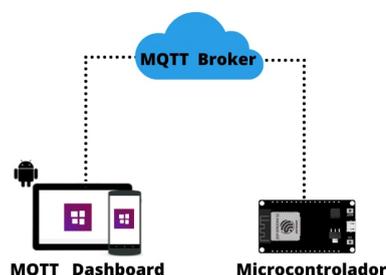


Fig. 6 Estrutura de um sistema usando MQTT Dashboard.
Fonte: Autoria própria (2021)

3.3.3 Blynk[®]

Blynk[®] é um serviço projetado para IoT que inclui um aplicativo para *smartphone* disponível para sistema Android e iOS, um servidor que realiza a comunicação entre *hardware* e aplicativo e, por último, as bibliotecas que permitem que os *hardwares* mais populares, como Raspberry Pi, ESP8266 e similares, se comuniquem com o servidor. Sua estrutura é apresentada na Figura 7. O aplicativo permite a personalização do painel, sendo possível adicionar botões, gráficos, *displays* e criar rotinas com temporizadores (Blynk[®], 2021).

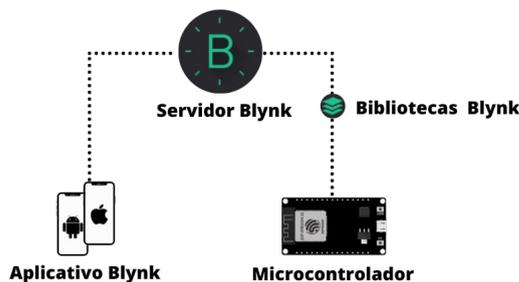


Fig. 7 Adaptação da estrutura do sistema apresentado pelo Blynk[®].

Fonte: Autoria própria (2021)

As informações recebidas no servidor ficam salvas na nuvem e podem ser baixadas em formato *comma-separated values* (CSV). Entretanto, apesar das bibliotecas e o acesso ao servidor serem gratuitos, o aplicativo permite somente um

número limitado de itens no painel, sendo necessário aquisição de pacotes adicionais para inclusão de mais itens. Todavia, o número de itens iniciais já é suficiente para desenvolvimento de aplicações mais usuais, além de que o custo de aquisição dos pacotes adicionais é consideravelmente baixo.

4. DESCRIÇÃO DO PROJETO

Nesta seção serão apresentados três projetos com diferentes níveis de complexidade e como eles foram desenvolvidos.

4.1 Primeiro nível

O sistema de primeiro nível tem como finalidade o controle de forma remota de apenas uma carga real, neste caso uma lâmpada de 220V, com dois estados de funcionamento, ligado e desligado. Para isso, utilizou-se o módulo Wi-Fi Esp-01 e um módulo relé, ambos energizados por uma fonte externa, como apresentado nas Figuras 8 e 9. A expansão de *hardware* é bem limitada, uma vez que o microcontrolador só possui duas portas para interação com sensores ou atuadores.

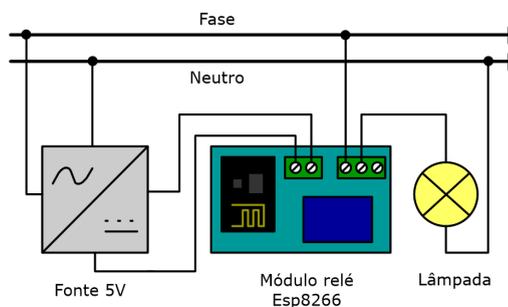


Fig. 8 Esquema elétrico do primeiro sistema.
Fonte: Autoria própria (2021)

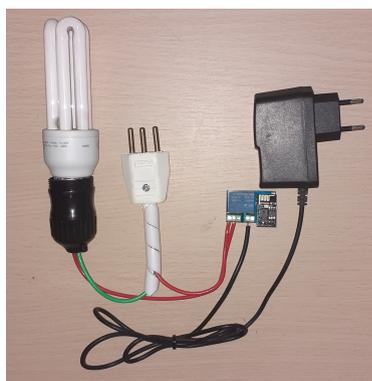


Fig. 9 Montagem do circuito do primeiro sistema.
Fonte: Autoria própria (2021)

Um *broker* MQTT público da HiveMQ foi escolhido para o teste por não se tratar de informações sensíveis. Aqui, o microcontrolador cumpre o papel de assinante, se inscrevendo em tópico e aguarda o recebimento de uma mensagem para alterar o sinal enviado ao relé.

O Node-RED® foi utilizado para criar uma interface de controle para o usuário e publicar os dados no tópico definido. Nesta aplicação, ele foi instalado em um computador, então o acesso ao painel de controle era limitado à rede que este computador estava conectado. Isso pode ser

contornado caso a instalação ocorra em um servidor em nuvem, como, por exemplo, o IBM Cloud. Todavia, a comunicação com o *hardware* é realizada de forma remota, uma vez que foi escolhido um *broker on-line*. Na Figura 10 é possível visualizar a interface do usuário.

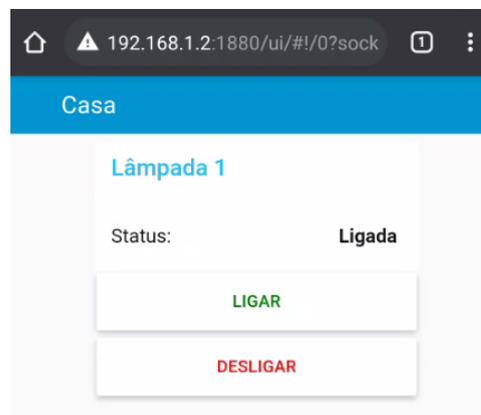


Fig. 10 Supervisório do primeiro sistema.
Fonte: Autoria própria (2021)

4.2 Segundo nível

O sistema de segundo nível visa o controle e monitoramento de múltiplas cargas com a inclusão de rotinas automatizadas. A aplicação desenvolvida gerencia 10 aparelhos condicionadores de ar da marca Fujitsu® de diferentes salas de aula do mesmo bloco didático do IFCE Campus Maracanaú. Inicialmente, foi realizada a cópia de dois sinais infravermelhos do controle remoto com auxílio da placa de desenvolvimento Arduino UNO e um receptor infravermelho. O primeiro sinal é o comando para desligar os aparelhos e o segundo é o comando para ligar os aparelhos com configurações adequadas para um consumo eficiente de energia. Posteriormente, foi construído o circuito elétrico, apresentado na Figura 11, o qual é responsável por conectar ao servidor do Blynk® para receber comandos e enviar o sinal para o aparelho correto. Para isso, foram necessários 10 emissores infravermelhos e um placa de desenvolvimento DOIT Esp32 DevKit v1. A energização do circuito é feita através do módulo Hi-Link HK-PMD01 que é ligado diretamente à rede elétrica e tem sua saída retificada de 5VDC. O painel de controle desenvolvido, Figura 12, foi instalado na sala da coordenação deste mesmo bloco didático e de lá foram distribuídos os emissores para cada sala de aula.

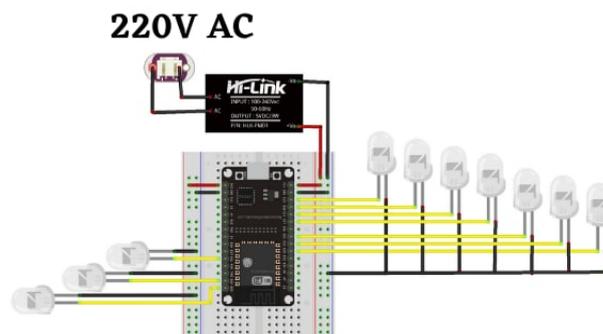


Fig. 11 Esquema elétrico do segundo sistema.
Fonte: Autoria própria (2021)

A leitura e o envio de valores para os pinos virtuais (aplicativo) ou reais (*hardware*) foram implementados somente através do HTTP REST API. Dentro do Blynk[®] aplicativo, Figura 13, foi elaborada a interface do usuário para controlar os aparelhos de todas as salas de aula de forma remota e nele também foram incluídas rotinas diárias para desligamento dos aparelhos ao fim do horário de cada aula, na intenção de evitar o desperdício de energia.



Fig. 12 Painel de controle do segundo sistema.
Fonte: Autoria própria (2021)



Fig. 13 Supervisório do segundo sistema.
Fonte: Autoria própria (2021)

4.3 Terceiro nível

O sistema de terceiro nível mantém o foco no controle e monitoramento de múltiplos dispositivos, mas dessa vez lida-se com diferentes tipos de sensores e atuadores. A aplicação foi instalada em um laboratório didático do IFCE Campus Maracanaú para controle de dois condicionadores de ar, três fileiras de lâmpadas, destravamento da porta, movimentação da persiana através de um servomotor e medição de consumo de energia através do sensor SCT013. Foram implementadas rotinas para desligamento automático dos condicionadores de ar em horários

pré-determinados pelo usuário e a movimentação da persiana pode ser feita através de comandos do usuário ou de modo automático dependendo da luminosidade detectada no ambiente pelo sensor LDR. Neste modelo, os dados da rede Wi-Fi podem ser reconfigurados pelo próprio usuário, pois ao acionar um botão, o microcontrolador apaga as informações de rede que estavam salvas na sua memória permanente e reinicia em modo de configuração. Nas Figuras 14 e 15 são apresentados o circuito elétrico e o painel de controle, respectivamente.

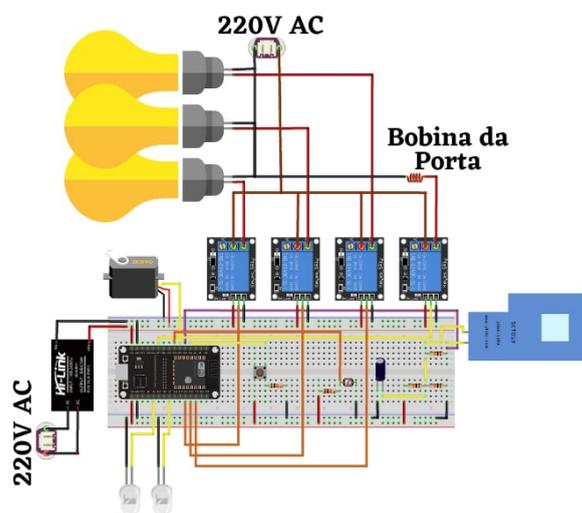


Fig. 14 Esquema elétrico do terceiro sistema.
Fonte: Autoria própria (2021)



Fig. 15 Painel de controle do terceiro sistema.
Fonte: Autoria própria (2021)

Assim como nas demais aplicações, a programação do microcontrolador é feita na IDE do Arduino e na sua comunicação são utilizados diferentes tipos de protocolos. Por exemplo, a sincronização do relógio é feita através do *Network Time Protocol* (NTP), mas o microcontrolador também se comunica com o aplicativo através do MQTT e o envio de informações para banco de dados se dá por HTTP REST API.

Dentro do MQTT *Dashboard*, limitou-se a adicionar botões virtuais para controle do usuário e a configuração dos horários de desligamento dos condicionadores de ar, como apresentado na Figura 16. Então, o processamento de dados

passou a ser responsabilidade do microcontrolador. Foi necessário também elaborar uma visualização de dados de consumo de energia que eram salvos no banco de dados, então foi desenvolvido uma página *web* para apresentação desses dados de forma gráfica, Figura 17, que foi hospedada no serviço de *hosting* do próprio *Firebase*.

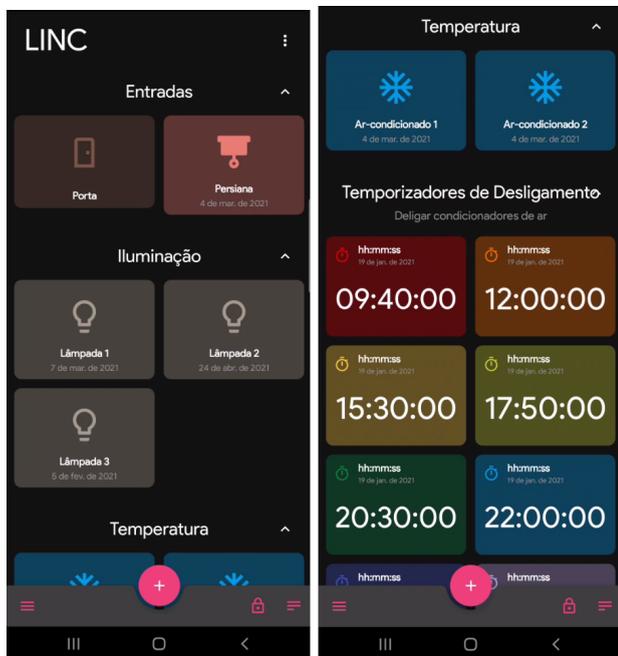


Fig. 16 Supervisório do terceiro sistema.

Fonte: Autoria própria (2021)



Fig. 17 Leitura de dados de consumo de energia.

Fonte: Autoria própria (2021)

4.4 Resumo dos sistemas implantados

Este tópico apresenta uma visão geral de cada uma dos projetos apresentados nos tópicos anteriores da Seção 4 e o custo para implantá-los.

Tabela 1. Informações gerais dos sistemas

Especificações	1° Nível	2° Nível	3° Nível
Interface de desenvolvimento	Esp-01	Esp32 DevKit	Esp32 DevKit
N° de sensores	0	0	3

Especificações	1° Nível	2° Nível	3° Nível
N° de atuadores	1	10	7
Protocolo de comunicação	MQTT	HTTP	MQTT e HTTP
Supervisório	Node-RED [®]	Blynk [®]	MQTT Dashboard e Página Web
Custo de Hardware	R\$ 45,90	R\$ 135,38	R\$ 206,32
Custo do Supervisório	R\$ 0,00	R\$ 11,99	R\$ 0,00

Fonte: Autoria própria (2021)

5. CONCLUSÕES

Considerando o desenvolvimento de cada projeto, observou-se que a escolha dos protocolos de comunicação se tornou dependente da escolha do supervisório. Isto porque cada supervisório opera com o *hardware* sobre um único protocolo, então essa falta de compatibilidade com os demais tipos acaba limitando a personalização da estrutura do projeto.

Dentre as três abordagens, observa-se que a utilização do MQTT permite o desenvolvimento de estrutura mais descentralizada, sendo possível remodelar o *hardware* sem necessariamente alterar configurações no supervisório e vice-versa. Desse modo, nos sistemas apresentados nos tópicos 4.1 e 4.3 pode-se alternar entre Node-RED[®] e MQTT Dashboard sem que seja necessário modificar a programação do microcontrolador. Por outro lado, no sistema apresentado no tópico 4.2, na programação do microcontrolador, discutido no tópico 3.3.3, utiliza-se de uma biblioteca própria do serviço Blynk[®] e ela só permite a comunicação com os servidores e aplicações pertencentes ao mesmo.

Por fim, observando somente os supervisórios, o MQTT Dashboard obtém pior desempenho, isto porque ele não apresenta muitas funções e nem rotinas de automação, além de não possuir integração direta com outras plataformas ou bancos de dados. Então, para se trabalhar com ele, é necessário escolher um microcontrolador que suporte todo o processamento de dados necessários para o funcionamento adequado do projeto. O Node-RED[®], por sua vez, possui inúmeros nós que permitem integração com diferentes serviços e pode ser acessado por qualquer aparelho que possua um navegador. Porém a ferramenta não possui servidor próprio para sua hospedagem e armazenamento de dados, podendo haver um custo adicional com aquisição de planos de serviços de computação em nuvem que atendam as especificações do projeto. Por esses motivos, que o serviço Blynk[®] acaba se destacando, pois é uma plataforma que integra bibliotecas para programação facilitada do microcontrolador, um serviço de nuvem com armazenamento ilimitado e uma interface de usuário intuitiva que não exige programação para sua personalização e configuração de

rotinas automatizadas. Contudo, este serviço necessita de uma interface de usuário para computador.

AGRADECIMENTOS

Agradecemos ao IFCE Campus Maracanaú pelo apoio material e financeiro e por ceder seus ambientes para desenvolvimento deste trabalho.

REFERÊNCIAS

- AI-THINKER CO. LTD (CN). (2015). *Esp-01 WiFi Module: Version 1.0*. Shenzhen, China: Ai-Thinker Co LTD. 19 p. <https://www.microchip.ua/wireless/esp01.pdf>
- BRASIL. (2020). *Lei nº 14.108, de 16 de dezembro de 2020. Altera as Leis nos 12.715, de 17 de setembro de 2012, e 9.472, de 16 de julho de 1997, para dispor sobre os valores da Taxa de Fiscalização de Instalação, da Taxa de Valorização de Funcionamento, da Contribuição para o Fomento da Radiodifusão Pública e da Contribuição para o Desenvolvimento da Indústria Cinematográfica Nacional (Condecine) das estações de telecomunicações que integrem sistemas de comunicação máquina a máquina, e sobre a dispensa de licenciamento de funcionamento prévio dessas estações*. Diário Oficial da União Seção:1. <https://www.in.gov.br/en/web/dou/-/lei-n-14.108-de-16-de-dezembro-de-2020-294616158>
- COLINA, A. L. et al. (2016) *Internet of Things in 5 days*. 1.1 S. L: S. N. 227 p. <https://github.com/marcozenaro/IPv6-WSN-book/tree/master/Releases>
- COLÓN, R. P., NAVAJAS, S., TERRY, E. (2019) *IoT IN LAC 2019: Taking the Pulse of the Internet of Things in Latin America and the Caribbean*. Inter-American Development Bank. S. L: Inter-American Development Bank. 40 p. DOI: <http://dx.doi.org/10.18235/0001968>
- ESPRESSIF SYSTEMS CO. LTD (CN). (2020) *ESP32-S2-WROOM ESP32-S2-WROOM-I: Datasheet*. Shanghai, China: Espressif Systems (Shanghai) Co. Ltd. v. 1.2. 26 p. https://www.espressif.com/sites/default/files/documentati/on/esp32-s2-wroom_esp32-s2-wroom-i_datasheet_en.pdf
- IBM (EUA). *What is the IoT?* <https://www.ibm.com/internet-of-things/learn/what-is-iot/>
- INTERNATIONAL TELECOMMUNICATION UNION. (2012). *ITU-T Y.4000/Y.2060: Overview of the Internet of things*. 1 ed. Genebra: ITU Telecommunication Standardization Sector. 22 p. <http://handle.itu.int/11.1002/1000/11559>
- LOPEZ RESEARCH LLC (EUA). (2013) *Uma Introdução à Internet das Coisas (IoT): Parte 1. da "Série de IoT"*. White Paper. São Francisco, CA: Lopez Research LLC. 8 p. https://www.cisco.com/c/dam/global/pt_br/assets/brand/iot/iot/pdfs/lopez_research_an_introduction_to_iiot_102413_final_portuguese.pdf
- MENEZES, A. R. F. et al. (2017) *Internet das Coisas e os Principais Protocolos*. Revista Expressão Científica, [S.L.]. v. 2. 14 p. <https://aplicacoes.ifs.edu.br/periodicos/index.php/REC/article/download/270/192/>
- MICROSOFT (EUA). *What is IoT?* <https://azure.microsoft.com/en-us/overview/internet-of-things-iot/what-is-the-internet-of-things/>
- MINERVA, R. BIRU, A. ROTONDI, D. (2015) *Towards a definition of the Internet of Things (IoT)*. [Turim]: IEEE Internet Initiative. 86 p. https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf
- NEBBIONE, G. e CALZAROSSA, M. C. (2020) *Security of IoT Application Layer Protocols: challenges and findings*. Future Internet, [S.L.], v. 12, n. 3, p. 55. MDPI AG. <http://dx.doi.org/10.3390/ii12030055>
- Node-RED (EUA). *About: Node-RED*. <https://nodered.org/about/>
- OLIVEIRA, L. C. L. e SILVA, F. O. (2020). *Análise comparativa de plataformas baseadas em Cloud para o desenvolvimento de aplicações IoT*. In: Workshop De Trabalhos De Iniciação Científica E De Graduação - Simpósio Brasileiro De Redes De Computadores E Sistemas Distribuídos (SBRC), 38. , 2020, Rio de Janeiro. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 257-264. ISSN 2177-9384. DOI: https://doi.org/10.5753/sbrc_estendido.2020.12427
- SASSI, M. S. H., JEDIDI, F. G., CHAARI, L. (2019). *A New Architecture for Cognitive Internet of Things and Big Data*. Elsevier: Procedia Computer Science. Sfax, p. 534-543. vol. 159. DOI: <https://doi.org/10.1016/j.procs.2019.09.208>
- SETHI, P. e SARANGI, S. R. (2017) *Internet of Things: Architectures, Protocols, and Applications*. Journal Of Electrical And Computer Engineering. Nova Deli. ID: 9324035. v. 2017. 25 p. DOI: <https://doi.org/10.1155/2017/9324035>
- SOARES, R. F. R. D. (2019). *Comparação entre protocolos da camada de aplicação para IoT*. 2019. 46 f. Monografia (Graduação) - Curso de Bacharelado em Sistemas de Informação, Universidade Federal Rural de Pernambuco, Recife. https://repository.ufpe.br/bitstream/123456789/2149/1/tc_c_raphaelfeliperamosduartesoaes.pdf
- VILLA-HENRIKSEN, A. et al. (2020). *Internet of Things in arable farming: Implementation, applications, challenges and potential*. Elsevier: Biosystems Engineering. Aarhus, p. 60-84. v. 191. DOI: <https://doi.org/10.1016/j.biosystemseng.2019.12.013>