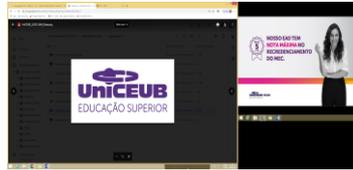


**FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS  
CURSO**

JONATHAS FORMIGA DE ARAUJO  
21341487

**AUTOMAÇÃO RESIDENCIAL COM OBJETOS INTELIGENTES:  
QUALIDADE DE VIDA E CONTROLE DE ENERGIA**

BRASÍLIA  
2020



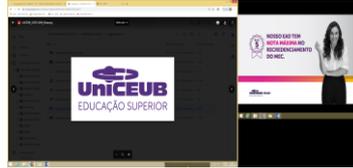
JONATHAS FORMIGA DE ARAUJO

## **AUTOMAÇÃO RESIDENCIAL COM OBJETOS INTELIGENTES: QUALIDADE DE VIDA E CONTROLE DE ENERGIA**

Trabalho de Conclusão de Curso (TCC) apresentado como um dos requisitos para a conclusão do curso de Engenharia de Computação do UniCEUB– Centro Universitário de Brasília

Orientador (a): **Prof. MsC Francisco Javier de Obaldia Diaz**

BRASÍLIA  
2020



JONATHAS FORMIGA DE ARAUJO

## **AUTOMAÇÃO RESIDENCIAL COM OBJETOS INTELIGENTES: QUALIDADE DE VIDA E CONTROLE DE ENERGIA**

Trabalho de Conclusão de Curso (TCC) apresentado como um dos requisitos para a conclusão do curso de Engenharia de Computação do UniCEUB – Centro Universitário de Brasília

Orientador (a): **Prof. MsC Francisco Javier de Obaldia Diaz**

Brasília, 2020.

### **BANCA EXAMINADORA**

---

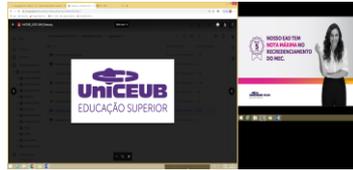
Me. Francisco Javier de Obaldia Diaz  
Orientador

---

Me. Ivandro da Silva Ribeiro  
Examinador

---

Dr. Sidney Cerqueira Bispo dos Santos  
Examinador



# AUTOMAÇÃO RESIDENCIAL COM OBJETOS INTELIGENTES: QUALIDADE DE VIDA E CONSUMO DE ENERGIA

## HOME AUTOMATION WITH INTELLIGENT OBJECTS: QUALITY OF LIFE AND ENERGY CONTROL

Jonathas Formiga de Araujo<sup>1</sup>, Francisco Javier de Obaldia Diaz<sup>2</sup>, Primeiro examinador<sup>3</sup>,  
Segundo examinador<sup>4</sup>

### RESUMO

Este artigo tem como objetivo analisar o desenvolvimento de um protótipo de uma residência automatizada construída a partir de componentes eletrônicos de baixo custo e desempenho notável. Para a automação residencial, foi utilizado como placa de desenvolvimento o NodeMCU, diodos emissores de luz (LED), sensores, para captar dados, tais como: de luminosidade, de temperatura e de umidade, de raios UV e ultrassônico. A programação foi realizada no IDE Arduino e a comunicação entre os periféricos se deu por meio do protocolo MQTT, que é um meio de comunicação bastante usado dentro do conceito de Internet das coisas. O *broker* utilizado foi o CloudMQTT, o qual pode ser acessado via web. Todas as informações recebidas dos sensores podem ser acessadas via aplicativo Dash MQTT, além de poder controlar dispositivos remotamente. A partir do desenvolvimento do projeto verificou-se que a automação residencial pode incrementar mais qualidade de vida às pessoas que a utilizam, por proporcionar um uso mais adequado e racional dos equipamentos eletrônicos residenciais.

**Palavras-chave:** Automação Residencial. Internet das Coisas. Sensores.

**Abstract:** This article aims to analyze the development of a prototype of an automated house built with low cost electronic components and good performance. For home automation, NodeMCU was used as a development board, light-emitting diodes (LED) and to capture data were used sensors such as: luminosity, temperature and humidity, UV and ultrasonic. The programming of the entire project was carried out on the Arduino IDE and the communication between the objects was performed using MQTT protocol, which is a means of communication widely used on Internet of things. The broker used was CloudMQTT, which can be accessed via the web. All information received from the sensors can be accessed via the Dash MQTT application, the app is able to control devices remotely as well. From the development of the project, it was found that home automation can increase the quality of life of people who use it, by providing a more appropriate and rational use of electronic home equipment.

**keywords:** Home Automation. Internet of Things. Sensors.

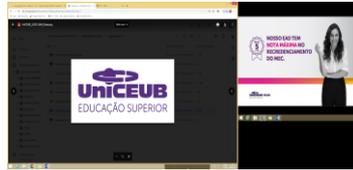
---

<sup>1</sup> UniCEUB, aluno.

<sup>2</sup> UniCEUB, orientador.

<sup>3</sup> UniCEUB, primeiro examinador.

<sup>4</sup> UniCEUB, segundo examinador.



## 1 INTRODUÇÃO

O avanço da tecnologia tem impactado positivamente as relações de trabalho, bem como todos os meios produtivos. Neste sentido, a automação residencial possibilita a execução de tarefas residenciais, tais como o uso controlado de eletrodomésticos. Desta forma, é possível a economia de energia e o uso correto de tais equipamentos.

Este artigo apresenta a aplicabilidade da automação residencial e analisa os benefícios decorrentes de sua implantação nas residências comuns.

Para tanto, utilizou-se como metodologia de pesquisa a análise de um protótipo de uma residência e verificação da automação utilizada no mesmo. Além disso, buscou-se na bibliografia especializada fundamentos e argumentos que possibilitaram a análise dos benefícios da aplicabilidade da automação residencial.

O projeto teve por objetivo demonstrar como a automação residencial por meio da Internet das Coisas pode colaborar no cotidiano das pessoas, proporcionando melhor qualidade de vida, além de controlar o consumo de energia. O projeto foi realizado a partir de iluminação inteligente e, outros recursos de automação residencial, foram realizados por meio de objetos inteligentes, como sensores interconectados e sistemas acessados via Internet.

Na conclusão deste artigo, elencam-se os benefícios advindos da implementação da tecnologia, objeto da pesquisa realizada e os limites encontrados na sua implementação.

## 2 REVISÃO BIBLIOGRÁFICA

Para melhor compreensão sobre o funcionamento do projeto da Automação Residencial, serão abordados conceitos e assuntos essenciais para o desenvolvimento do protótipo.

### 2.1 Trabalhos relacionados

Em um projeto de Automação

Residencial proposto por ACCARDI, EUGENI DODONOV (2012) são citados alguns elementos básicos necessários para a realização da AR. Esses elementos, tais como: controladores, sensores, atuadores, barramentos e interfaces são componentes essenciais para o projeto e são encontrados frequentemente em residências inteligentes.

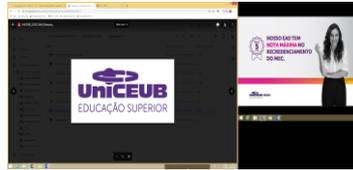
De acordo com ACCARDI e DODONOV (2012), a Automação Residencial é uma área com grande potencial de expansão e que precisa ser melhor explorada.

A tecnologia tem a finalidade de trazer e proporcionar à sociedade tranquilidade, comodidade e uma vida mais prática e econômica. Por estas razões, a automação residencial tem ganhado espaço nos dias atuais. Verifica-se que a tecnologia possui uma vertente de inclusão social, possibilitando às pessoas com deficiências utilizarem melhor os recursos disponíveis e terem uma vida com mais autonomia e independência. (TÓFOLI, 2014)

Nos dias atuais, estar conectado à internet é muito mais que uma opção. Tornou-se uma necessidade que perpassa a vida das pessoas, refletindo também no dia a dia das casas, a partir de uma busca e implementação de equipamentos e tecnologia cada vez mais inteligentes. Há uma evolução da internet comum para a internet das coisas. É possível ter a conexão de toda a casa na internet, possibilitando o melhor uso dos equipamentos e uma vida mais confortável e com o uso racional de energia. (CARDOSO, 2019).

### 2.2 Automação Residencial

Conceitua BOLZANI (2004) Automação Residencial (AR) como sendo um conjunto de tecnologias que auxiliam nas tarefas domiciliares do dia-a-dia. Estas tecnologias objetivam maior comodidade e segurança, além de maior controle sobre o consumo de energia.



## 2.3 IoT – Internet das Coisas

Segundo EVANS (2011) a Internet das Coisas mudará tudo. Este autor considera o impacto que a Internet já provocou na humanidade, como na comunicação, na educação, nos negócios e na ciência. A IoT representa uma evolução da Internet, propiciando a capacidade de coletar, analisar e distribuir dados em prol de benefícios para a humanidade.

### 2.3.1 Protocolo MQTT

Protocolo MQTT, sigla de MQ *Telemetry Transport*, pode ser definido como um protocolo de comunicação bastante usado dentro do conceito de Internet das Coisas. É um protocolo de troca de informações baseado na arquitetura publish/subscribe. Esse tipo de comunicação requer uma largura de banda baixa e trabalha com alta latência. Embora esse protocolo seja voltado para redes inseguras, a confiabilidade e a garantia de entrega são pontos positivos. (Para MARTINS, MQTT (2015))

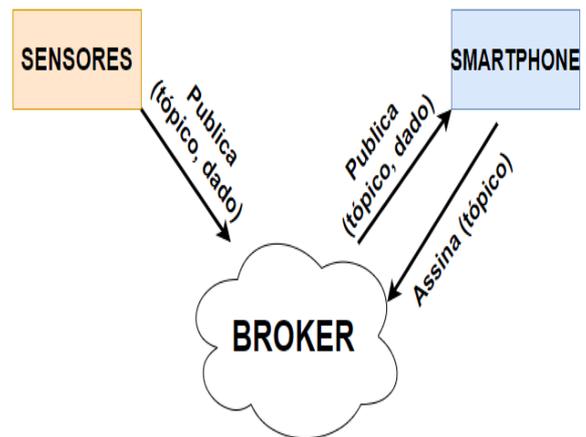
Segundo JAFFEY(2014), o protocolo desenvolvido pela IBM projetado para comunicação M2M (machine to machine) seguindo a arquitetura do modelo cliente/servidor. Cada sensor funciona como cliente que se conecta ao servidor (broker), por meio de TCP. Como o MQTT é um protocolo orientado a mensagens, toda a troca de dados é realizada via mensagens, através do elemento responsável (o broker) por gerir as publicações e subscrições deste protocolo. A transmissão das mensagens é realizada a partir da publicação de um tópico, que é um endereço semelhante a uma estrutura de diretórios, por exemplo, casa/sala/lâmpada. Já os clientes, após o tópico ser publicado, devem assinar (ou subscrever) para o mesmo. A partir disso, o cliente é capaz de receber todas as mensagens publicadas neste tópico.

Para YUAN (2017), um cliente é qualquer coisa que pode realizar a comunicação com o broker (veja a Figura 1),

seja recebendo ou enviando mensagens. Sensores, smartphones ou qualquer dispositivo conectado na internet pode ser considerado cliente.

O funcionamento da comunicação começa com a conexão do cliente ao broker. Ele assina algum tópico de mensagem no servidor. Após isso, o cliente envia a mensagem com o seu respectivo tópico ao broker através do comando de publicação (*publish*). Por fim, o broker encaminha a mensagem aos clientes que assinam o mesmo tópico já publicado. Na figura 1 é apresentado o modelo de publicação e assinatura do MQTT para clientes IoT.

Figura 1. Modelo de publicação/assinatura

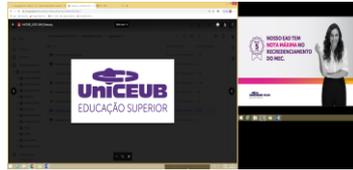


Fonte: Autor (2020)

## 2.4 Sensores

Os sensores são os dispositivos responsáveis por detectar, monitorar e responder a algum estímulo. Existem diferentes tipos de sensores para diferentes tipos de estímulos, por exemplo: temperatura, luz, umidade, som, aproximação, etc. A conversão do estímulo em um dado manipulável é realizada por sistemas computacionais, que encaminham as informações para os controladores.

Para o funcionamento da Automação Residencial foram utilizados alguns sensores. Estes serão mencionados e brevemente apresentados a seguir.



#### 2.4.1 DHT11

É o sensor responsável por medir a umidade e temperatura do ambiente, ele atua com dados precisos e possui uma resposta rápida. A comunicação pode ser realizada pela interface de barramento único ou I2C.

#### 2.4.2 HC-SR4

O sensor ultrassônico é responsável por detectar a presença de objetos a uma distância de 2 cm a 4m. Esse sensor emite uma onda sonora de alta frequência, esse sinal vai gerar um eco quando colidir em um objeto. A distância é medida a partir do intervalo de tempo em que o impulso sonoro é emitido e o eco é recebido de volta.

#### 2.4.3 Sensor de raios UV

Esse sensor é capaz de medir a incidência de raios ultravioletas no ambiente. O índice de UV é medido a partir de valores de tensão em milivolts obtidos pelo sensor. Esses valores de tensão são associados ao índice UV de acordo com valores tabelados.

A partir da tabela mostrada na figura 2 é possível verificar o índice UV de acordo com cada tensão.

Figura 2. Tabela de Índice UV a partir da tensão

UV Index (Índice UV)	0	1	2	3	4	5
Vout(mV)	<50	227	318	408	503	606
UV Index (Índice UV)	6	7	8	9	10	11 <sup>+</sup>
Vout(mV)	696	795	881	976	1079	1170+

Fonte: Flípeflop, 2015

#### 2.4.4 LDR

Sensor de luminosidade cuja resistência varia de acordo com a intensidade da luz.

Quanto maior a incidência de luz, menor será a resistência (FLIPEFLOP, 2015).

### 2.5 Servidor Broker MQTT

O broker é responsável por receber todas as mensagens dos clientes e, após isso, direciona essas mensagens para o cliente destinatário (subscriber).

O *broker* utilizado neste projeto foi o CloudMQTT, que pode ser acessado em: <<https://www.cloudmqtt.com/>>. Ele funciona como um agente de mensagens hospedado para internet das coisas. Os servidores do CloudMQTT são servidores gerenciados em nuvem.

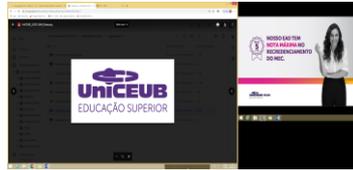
Para usufruir do serviço do CloudMQTT é necessário a criação de uma conta para obter as informações da instância, como mostra a Figura 3. O serviço de broker é pago, onde existem vários planos mensais e anuais diferentes. Para o caso deste projeto, o plano escolhido foi o “Keen Koala”, que é o plano mais simples disponível.

Figura 3. Informações da instância Instance info

Server	great-gardener.cloudmqtt.com
Region	amazon-web-services::sa-east-1
Created at	2020-11-05 14:33
User	nnjlxvdp <input type="button" value="Restart"/>
Password	u7V88w... <input type="button" value="Refresh"/>
Port	1883
SSL Port	8883
Websockets Port (TLS only)	443
Mosquitto version	1.5.7 <input type="button" value="Upgrade available"/>

Fonte: CloudMQTT (2020)

O primeiro passo, após a criação da conta e ter conectado no site, é criar uma instância. Nessa instância haverá todas as informações da conexão, conforme figura 3,



que podem ser acessadas na aba “*details*”. Estas informações serão usadas posteriormente no aplicativo do celular e no microcontrolador NodeMcu. As informações que serão utilizadas serão: servidor, usuário, senha e porta. Com essas informações, qualquer pessoa poderá controlar os dispositivos com essas configurações que estiverem conectados à Internet.

Outra funcionalidade, que é bem importante e bastante usada, é a seção “*Websocket UP*”, a qual é possível verificar a comunicação dos clientes em tempo real. A cada mensagem recebida é exibido o tópico e a mensagem que foi enviada neste tópico. Isso possível observar na Figura 4.

Figura 4. Websocket

Received messages ▼

Topic	Message
Temperatura	24.60 °C
Temperatura	24.60 °C
Temperatura	24.60 °C
CASA/QUARTO/LAMPADA	liga
CASA/QUARTO/LAMPADA	desliga
CASA/QUARTO/LAMPADA	liga
CASA/QUARTO/LAMPADA	desliga

Fonte: CloudMQTT(2020)

## 2.6 Aplicativo Smartphone

O aplicativo é responsável por monitorar e controlar objetos remotamente. É possível ligar e desligar lâmpadas ou outros periféricos, verificar informações em tempo real, como temperatura, umidade e intensidade de raios UV. A configuração é realizada de acordo com os dados obtidos na criação da instância do broker no CloudMQTT.

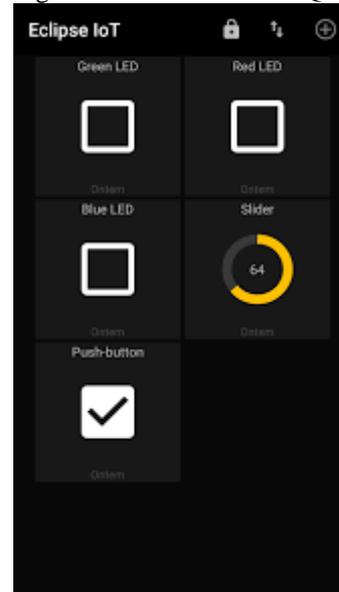
O aplicativo usado foi o MQTT Dash, um aplicativo disponível no sistema Android que permite a construção de um painel de controle a partir de informações obtidas pela comunicação entre objetos.

O MQTT Dash oferece algumas opções de entrada e saída de dados, como botões, interruptores, campo de texto, etc. Cada item

é atribuído a um tópico MQTT diferente. Logo, para cada item do painel de controle é necessário um tópico MQTT, sendo publicado/assinado no mesmo broker configurado no microcontrolador.

A figura 5 mostra como o *dashboard* do aplicativo possui um leiaute bem amigável e simples.

Figura 5. *Dashboard* do MQTT Dash Android



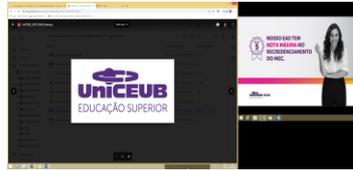
Fonte: Flípeflop, 2019

## 3 METODOLOGIA DO TRABALHO

Este artigo foi elaborado a partir da realização de uma experiência com a construção de uma maquete residencial simulando o uso da internet das coisas em uma residência real.

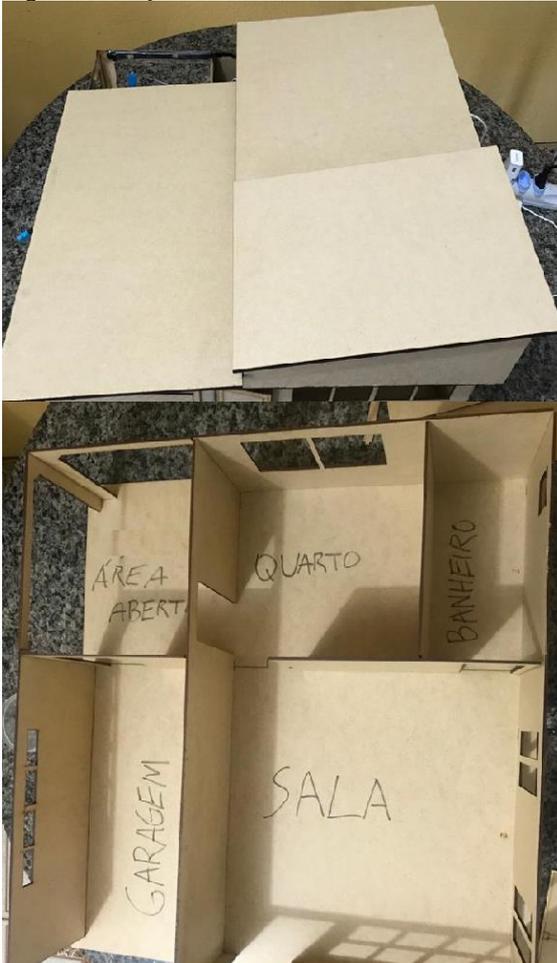
O protótipo da residência foi elaborado em algumas partes: construção da maquete em madeira mdf, simulação elétrica e programação.

Na parte da maquete, a casa foi projetada para ter os seguintes cômodos: sala de estar, quarto, banheiro, área aberta e garagem (veja a Figura 6). O foco do projeto é na área da automação da residência, a parte de construção foi realizada para facilitar no entendimento e auxiliar na parte computacional e eletrônica, que abrange Internet das Coisas e automação,



respectivamente, e a maquete para projetos educacionais com objetivo de automatização de protótipos. A casa possui 62cm de comprimento e 59cm de largura.

Figura 6. Maquete da casa



Fonte: Autor (2020)

O projeto elétrico foi construído a partir de componentes eletrônicos de fácil acesso e baixo custo.

A comunicação entre os componentes eletrônicos se deu por meio da Internet, via protocolo MQTT, que é um protocolo simples para implementação de Internet das Coisas.

Para realizar essa comunicação via Internet é necessário o uso de um **mqtt Broker**, que é um servidor na nuvem responsável por gerenciar as publicações e subscrições (publish/subscribe) do MQTT. Esse servidor funciona como intermediador da comunicação, ou seja, é ele que vai receber a

informação do celular e passar a informação para o NodeMcu.

Figura 7. Protótipo do projeto com componentes

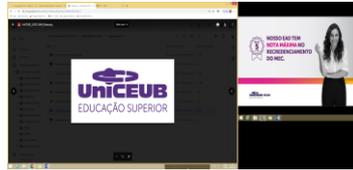


Fonte: Autor (2020)

A figura 7 mostra o protótipo do projeto com os componentes que foram utilizados na pesquisa. Pode-se observar na sala uma lâmpada inteligente e o sensor de presença HC-SR04 que irá verificar a presença de algum objeto, na área aberta haverá o DHT11 para medir temperatura e umidade relativa do ar, o LDR para verificar se há luminosidade no local e o sensor UV para medir a incidência de raios UV, as luzes do quarto e do banheiro serão controladas via aplicativo, já o NodeMCU que tem como função controlar todo esses objetos, ficará em uma central de controle, juntamente com uma protoboard e um buzzer.

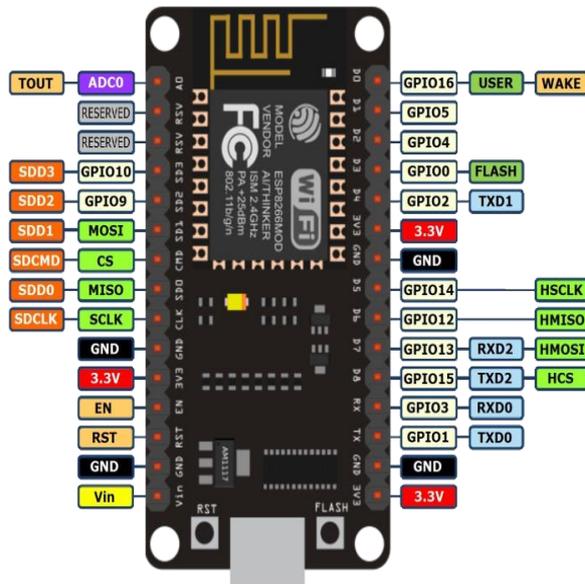
### 3.1 Dados e aquisições

O desenvolvimento do projeto teve como base a placa de desenvolvimento NodeMCU, exibida na figura 5, o qual é composto por um módulo ESP8266 e um processador de 32 bits com Wi-Fi integrado. Esse microcontrolador possui dez entradas digitais e uma analógica, além de portas de alimentação providas da conexão USB.



A figura 8 mostra a placa de desenvolvimento NodeMCU, juntamente com sua pinagem.

Figura 8. Placa NodeMCU



Fonte: Flípeflop, 2015

Para COSTA (2018), o uso da placa NodeMCU tem se tornado popular devido aos recursos oferecidos e a facilidade de desenvolvimento, além do fato de ser um objeto de baixo custo.

O NodeMCU pode ser carregado via cabo micro-usb utilizando a IDE do Arduino, o que facilita a programação devido ao grande acervo de livros e artigos que citam essa IDE. A linguagem do Arduino é baseada em C/C++.

A IDE do Arduino dispõe de diversas bibliotecas para o ESP8266 e diversos módulos de *hardware*. O desempenho é excelente devido ao uso da linguagem C++, visto que essa linguagem, normalmente, consome pouca memória devido ao poder de alocação de memória oferecido pela linguagem.

Para fins de programação, as portas de entrada e saída do NodeMCU possuem uma equivalência com o número das portas do ESP8266, de acordo com cada pino. É possível notar na tabela 1 as portas do ESP8266 que correspondem às portas do NodeMCU.

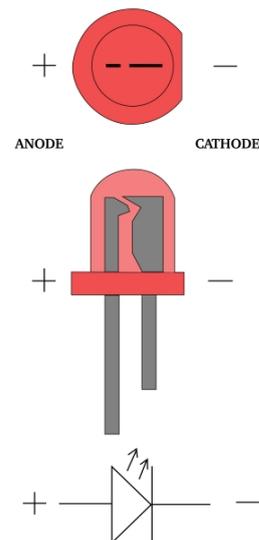
Tabela 1. Equivalência de portas NodeMCU/ESP

NodeMCU	ESP8266
D0	16
D1	5
D2	4
D3	0
D4	2
D5	14
D6	12
D7	13
D8	15
D9	3

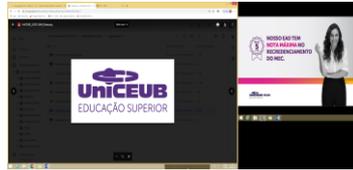
Fonte: Autor (2020)

Para fazer o papel da iluminação, serão usados diodos emissores de luz, mais conhecidos como LED. O diodo é um componente eletrônico que conduz corrente elétrica apenas em uma polaridade, por exemplo 0 ou 1. No projeto foram utilizados LEDs transparentes de luz branca e LEDs amarelos, que acendem caso estejam com valor alto (*HIGH*), conforme programado. O lado positivo do LED é conhecido como anodo e o negativo como catodo, de acordo com a Figura 9.

Figura 9. Placa NodeMCU

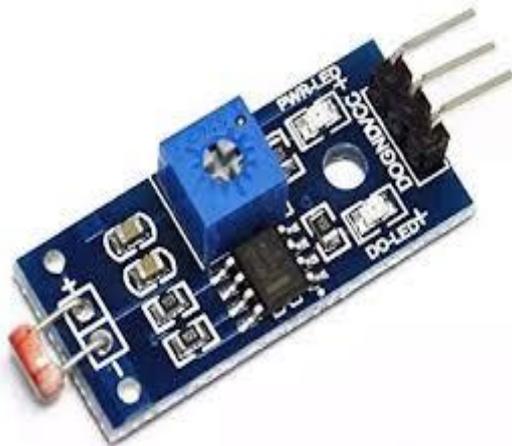


Fonte: Flípeflop, 2015



Juntamente com os LEDs, o módulo de sensor de luminosidade LDR foi usado para detectar se há luminosidade na área aberta da residência. Esse sensor irá auxiliar no acendimento automático das luzes em caso de baixa luminosidade. Conforme a figura 10 é possível notar que ele possui 3 pinos: VCC (positivo), GND (terra) e DO (saída digital de dados).

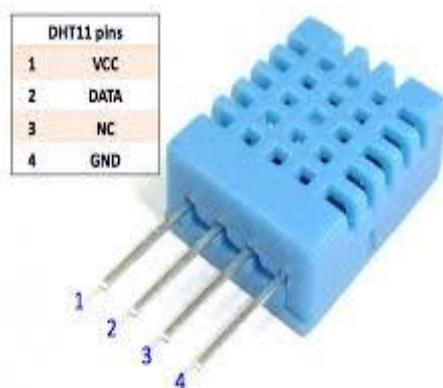
Figura 10. Módulo sensor LDR



Fonte: Eletrogate, 2020

Na medição de temperatura do ambiente externo foi utilizado o sensor DHT11. É possível verificar os pinos de alimentação e de saída digital de dados na figura 11.

Figura 11. Sensor DHT11



Fonte: FritzingLab (2016)

Com esse sensor também é possível realizar a leitura de umidade relativa do ar. Embora o DHT11 seja um periférico de baixo custo e tenha uma boa precisão, a faixa de aferição de temperatura e umidade são baixas. A precisão da temperatura é decimal, de 0,10, ou seja, a medição vai aumentar ou diminuir sempre de 0,10°C em 0,10°C. Já para umidade relativa do ar, este sensor só consegue medir umidade acima de 30%. Isso foi um problema durante boa parte do projeto, pois a umidade relativa do ar em Brasília frequentemente fica abaixo dos 30%.

Juntamente com o sensor DHT11 foi utilizado um *buzzer*, o qual possui um oscilador interno que emite um som contínuo quando ligado à corrente. Quando a temperatura, medida pelo sensor, ultrapassa 28°C e a umidade relativa do ar está abaixo de 30%, será emitido um alerta sonoro, que avisará que a temperatura e umidade do ar estão inapropriadas para ficar ao ar livre. A figura 12 apresenta o buzzer, que possui apenas dois pinos, um de entrada de dados e outro de alimentação.

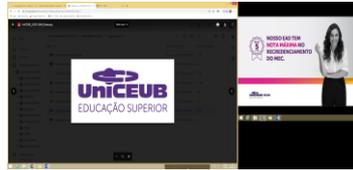
Figura 12. Buzzer 6V



Fonte: Baú da Eletrônica (2020)

### 3.2 Conectando o NodeMCU a Internet e ao servidor do *broker* CloudMQTT

A conexão dos clientes com o NodeMCU se dá a partir do protocolo MQTT. Esse protocolo, responsável pela comunicação, funciona via Internet. Logo, a placa de desenvolvimento precisa estar



conectada a Internet também. Para haver a comunicação entre objetos não é necessário que todos estejam na mesma rede, é preciso apenas estar conectado a Internet.

Figura 13 Código Declaração de dados conexão

```

1 #include <ESP8266WiFi.h>
2 #include <PubSubClient.h>
3
4 const char* ssid = "jonawifi";
5 const char* password = " ";
6 const char* mqttServer = " ";
7 const int mqttPort = 1883;
8 const char* mqttUser = " ";
9 const char* mqttPassword = " ";
10
11 WiFiClient espClient;
12 PubSubClient client(espClient);

```

Fonte: Autor (2020)

O código representado na figura 13 mostra a inclusão das bibliotecas <ESP8266WiFi.h> e <PubSubClient.h> necessárias para a conexão do NodeMCU a Internet e ao *broker* MQTT, respectivamente. Elas podem ser baixadas no próprio IDE Arduino. Os dados da conexão como rede, senha, servidor, usuário e senha são declarados no início do código. Para o funcionamento da comunicação é necessário manter as informações atualizadas.

Figura 14. Código para conexão a Internet

```

12 PubSubClient client(espClient);
13
14 void setup() {
15   Serial.begin(115200);
16   WiFi.begin(ssid, password);
17
18   while (WiFi.status() != WL_CONNECTED)
19   {
20     delay(100);
21     Serial.println("Conectando a WiFi..");
22   }
23   Serial.println("Conectado!");
24   client.setServer(mqttServer, mqttPort);
25
26   while (!client.connected()) {
27     Serial.println("Conectando ao servidor MQTT...");
28     if (client.connect("Projeto", mqttUser, mqttPassword ))
29     {
30       Serial.println("Conectado ao servidor MQTT!");
31     } else {
32       Serial.print("Falha ao conectar ");
33       Serial.print(client.state());
34       delay(2000);
35     }
36   }
37
38   client.publish("Status ", "Reiniciado!");
39   client.publish("Placa", "Em funcionamento!");
40 }

```

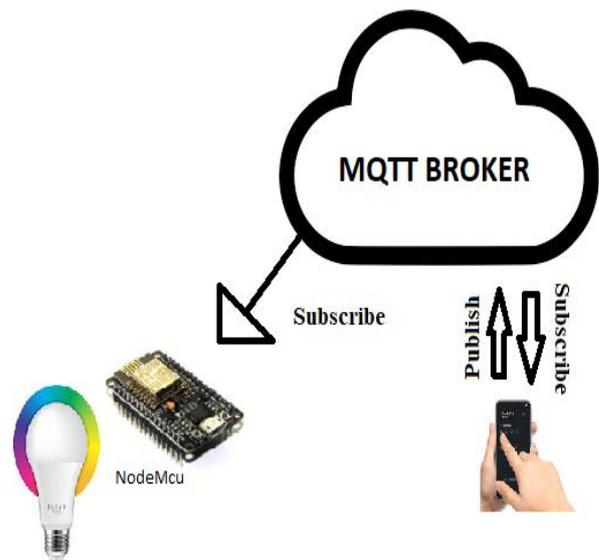
Fonte: Autor (2020)

A figura 14 exibe a função para realizar a conexão da placa a Internet e ao *broker* MQTT. Após baixar as bibliotecas do NodeMCU, a própria IDE fornece o modelo de código para realizar a conexão a Internet, sendo necessário realizar alguns ajustes nas mensagens que serão exibidas no CloudMQTT quando a conexão estiver funcionando corretamente.

### 3.2 Ligando lâmpadas pelo Smartphone

Em três cômodos da casa - quarto, banheiro e sala - será possível acender a lâmpada (LED) a partir da assinatura do tópico “CASA/LAMPADAS”. O cliente assina o tópico, que será enviado ao *broker*, que irá repassar a informação, usando também um *subscribe*, ao NodeMCU. A figura 15 representa graficamente como ocorre a comunicação entre os clientes.

Figura 15. Protótipo de comunicação lâmpada



Fonte: Autor (2020)

Os LEDs que representam as lâmpadas foram ligados nas portas D1, D2 e D3. Eles podem ser acionados por qualquer dispositivo que use protocolo MQTT, desde que possua todos os detalhes da conexão e permissão.

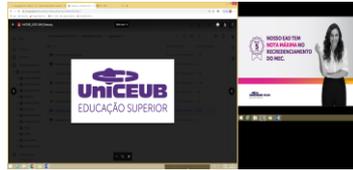
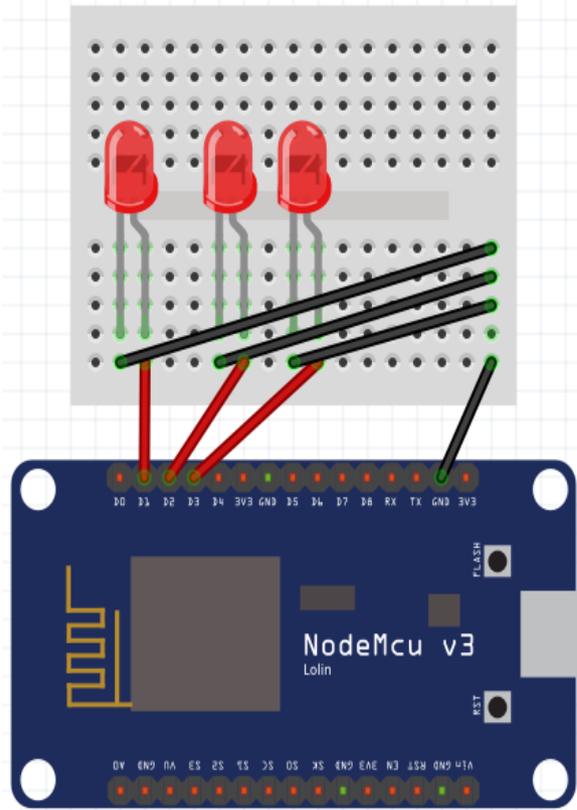


Figura 16. Esquemático LEDES



Fonte: Fritzing (2020)

O esquemático apresentado na figura 16 representa a conexão dos LEDs na placa de desenvolvimento NodedMCU. A lâmpada do quarto está conectada na porta D1, a do banheiro na D2 e a da sala na D3.

Tabela 2. Equivalência de portas NodeMCU/ESP

Comodo	Porta NodeMCU	liga	desliga
Quarto	D1	A	B
Sala	D2	C	D
Banheiro	D3	E	F

Fonte: Autor (2020)

Na tabela 2 estão as mensagens necessárias para acender/desligar as lâmpadas de acordo com cada cômodo. Para que isso ocorra, é necessário que o cliente assine o tópico “CASA/LAMPADAS” com a mensagem A,B,C,D,E ou F. Para acender a lâmpada da sala, por exemplo, é preciso assinar o tópico enviando a mensagem ‘A’ ao broker.

Figura 17. Código para acender leds

```

1  const int LED1 = 5;
2  const int LED2 = 4;
3  const int LED3 = 0;
4
5  void setup() {
6    Serial.begin(115200);
7    pinMode(LED1, OUTPUT);
8    pinMode(LED2, OUTPUT);
9    pinMode(LED3, OUTPUT);
10
11    client.setCallback(callback);
12    client.subscribe("CASA/LAMPADAS");
13  }
14
15  void callback(char* topic, byte* dados_tcp, unsigned int length) {
16    for (int i = 0; i < length; i++) {
17      }
18    switch ((char)dados_tcp[0]){
19      case 'A':
20        digitalWrite(LED1, HIGH);
21        break;
22      case 'B':
23        digitalWrite(LED1, LOW);
24        break;
25      case 'C':
26        digitalWrite(LED2, HIGH);
27        break;
28      case 'D':
29        digitalWrite(LED2, LOW);
30        break;
31      case 'E':
32        digitalWrite(LED3, HIGH);
33        break;
34      case 'F':
35        digitalWrite(LED3, LOW);
36        break;
37    }
38  }
39  void loop() {
40    client.loop();
41  }

```

Fonte: Autor (2020)

O código mostrado na figura 17 demonstra a programação realizada no IDE Arduino. Primeiramente, os LEDs 1, 2 e 3 foram declarados como pinos de saída. Após isso foi declarado o tópico “CASA/LAMPADAS” e a função que determina o acendimento das lâmpadas.

### 3.3 Acionamento automático da lâmpada de acordo com a falta de luminosidade no ambiente

A verificação da presença ou não de luz natural é de responsabilidade do sensor de luminosidade LDR. A placa do sensor permite realizar o ajuste de sensibilidade à luz através do trimpot, que é um mini potenciômetro acoplado ao módulo LDR. Após a verificação será acionado ou não uma carga elétrica na alimentação do LED em função da presença de luminosidade sobre o sensor.

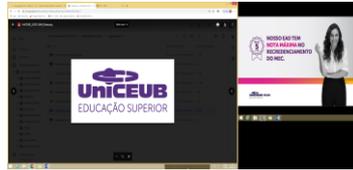
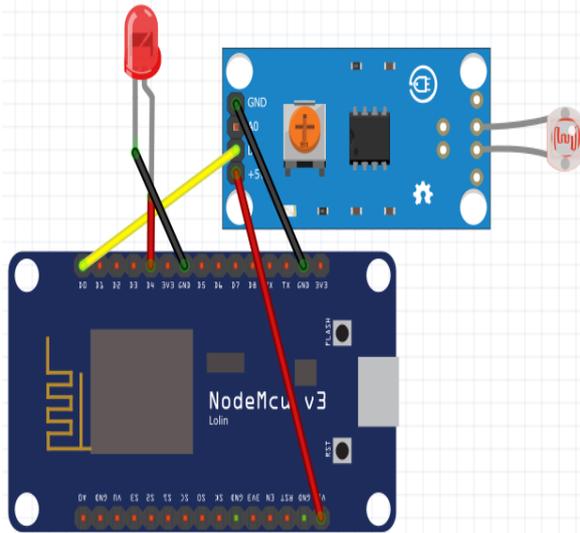


Figura 18. Esquemático LED e sensor LDR



Fonte: Fritzing(2020)

De acordo com o esquema representado na figura 18, o módulo necessita de 5V de alimentação e a entrada de dados coletados pelo sensor foi realizada pela porta D0. A saída de luz, emitida pelo LED, foi conectada na porta D4.

O LED ficou na área da casa onde há uma grande incidência de luz natural. O sensor fica localizado na parede externa da casa, a fim de melhorar a precisão da verificação de luz. A programação foi realizada em uma função simples, onde se a leitura do pino do sensor for igual a 1 (*HIGH*), o LED acende. Pode-se verificar no código contido na figura 19.

Figura 19. Código sensor de luminosidade

```

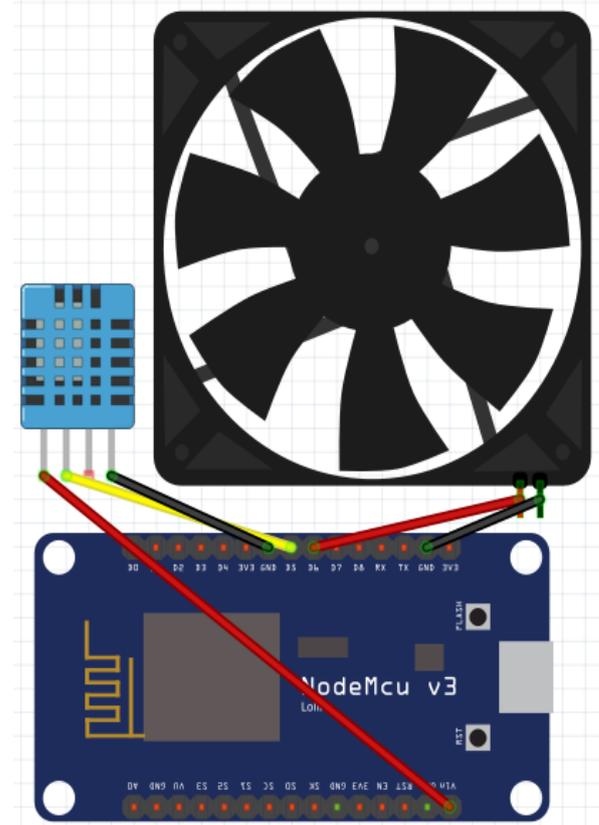
1  const int pinoSensor = 16;
2  const int pinoLed = 2;
3
4  void sensorluz(){
5
6    if(digitalRead(pinoSensor) == HIGH){
7      digitalWrite(pinoLed, HIGH);
8    }
9    else{
10     digitalWrite(pinoLed, LOW);
11    }
12  }

```

Fonte: Autor (2020)

### 3.3 Acionamento automático do sistema de ventilação de acordo com dados obtidos pelo sensor ultrassônico e de temperatura

Figura 20. Esquemático sensor DHT11 e ventilador



Fonte: Fritzing(2020)

O sistema de ventilação, observado na figura 20, está disponível na sala, pois é onde está a porta de entrada com uma grande janela e, por isso, pode haver uma grande incidência de calor solar durante o dia. O sensor está estrategicamente localizado, no meio do cômodo, onde as pessoas costumam ficar.

Conforme esquema observado na figura 20, a saída de dados do sensor DHT11 foi ligada na porta D5 (cabo amarelo), o pino VCC foi ligado na alimentação de 5V do NodeMCU e o pino GND do sensor foi ligado no GND da placa. O sistema de ventilação, representado por um cooler de 3V, foi ligado na porta D6.

O sensor ultrassônico é responsável por detectar a presença de objetos, ele possui um pino Trigger, que é usado para transmitir um pulso de ultrassom e um pino Echo, o qual recebe e detecta o pulso de ultrassom refletido por um obstáculo dentro de uma distância pré configurada na programação (ver figura 21).

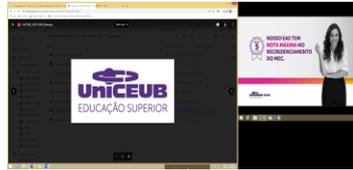
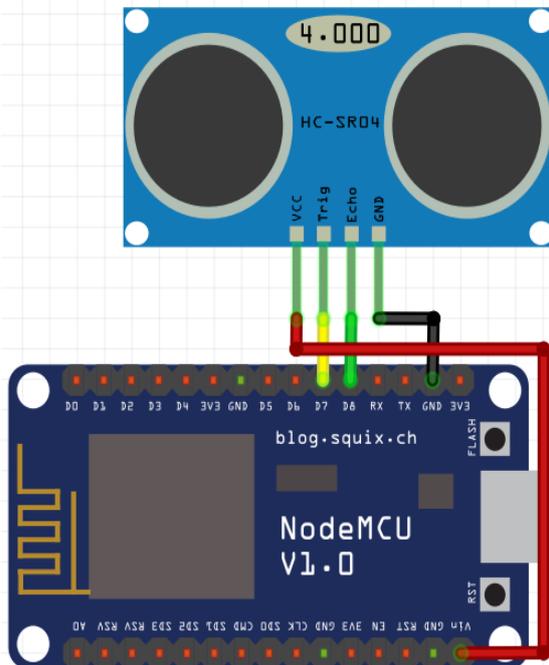


Figura 21. Esquemático sensor ultrassônico



Fonte: Fritzing(2020)

O sistema de ventilação ligará automaticamente quando o sensor de presença detectar algo em até 2cm de distância e se a temperatura aferida pelo DHT11 for maior que 27°C.V

Figura 22. Código sensor de temperatura

```

1 #include <DHT.h>
2 #define DHT11_PIN 14
3 const int ventilador = 12;
4
5 void setup() {
6   Serial.begin(115200);
7   client.subscribe("CASA/SALA/TEMPERATURA");
8 }
9
10 void tempDht(){
11   int chk = DHT.read11(DHT11_PIN);
12   String msg;
13   msg = DHT.temperature;
14   if (msg != "-999.00"){
15     msg = msg + " Â°C ";
16     char message[58];
17     msg.toCharArray(message,58);
18     client.publish("CASA/SALA/TEMPERATURA", message);
19   }
20   float temperatura = DHT.temperature;
21   if (temperatura >= 26){
22     digitalWrite(ventilador, HIGH);
23   }
24   } else{
25     digitalWrite(ventilador, LOW);
26   }
27 }

```

Fonte: Autor (2020)

Figura 23. Código sensor ultrassônico

```

1 void sensorUltra(){
2   digitalWrite(trigPin,HIGH);
3   delayMicroseconds(10);
4   digitalWrite(trigPin, LOW);
5
6   duracao = pulseIn(echoPin, HIGH ,1000000);
7   aux = duracao;
8   distancia = (aux*340/10000)/2;
9
10  if(distancia <= 2){
11    digitalWrite(VENTILADOR, HIGH);
12  }
13  else{ //SENÃOfo, FAZ
14    digitalWrite(VENTILADOR, LOW);
15  }
16
17  delay(1000);
18 }

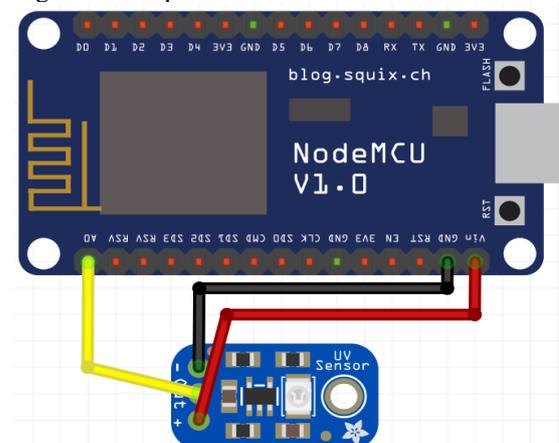
```

Fonte: Autor (2020)

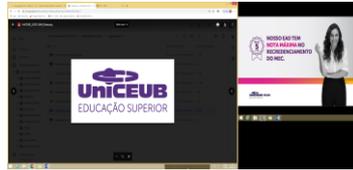
Nas figuras 22 e 23 estão representadas as funções do sensor de temperatura e de presença, respectivamente. Vale ressaltar que a imagem foi usada apenas para mostrar como funciona cada sensor. Porém, na situação prática, a estrutura de condição *if* irá abranger ambos os dados obtidos pelo sensor de temperatura e de presença, pois o ventilador só irá ativar caso ambas as condições forem verdadeiras.

O sensor UV, o único a ser conectado em uma porta analógica (ver figura 24), foi usado para medir a incidência de raios ultravioletas na área aberta da maquete.

Figura 24. Esquemático sensor UV



Fonte: Fritzing(2020)



Os dados coletados por esse sensor são exibidos para o usuário no aplicativo. A programação realizada para converter a tensão capturada pelo sensor em índice UV, além do tópico utilizado para publicação do dado, pode ser observada na figura 25.

Figura 25. Código sensor UV

```
8 void Calcula_nivel_UV()
9 {
10  valor_UV = analogRead(pino_sensor_UV);
11  //Calcula tensao em milivolts
12  int tensao = (valor_UV * (5.0 / 1023.0)) * 1000;
13  //Compara com valores tabela UV_Index
14  if (tensao > 0 && tensao <= 227)
15  {
16    UV_index = "0";
17  }
18  else if (tensao > 227 && tensao <= 318)
19  {
20    UV_index = "1";
21  }
22  else if (tensao > 318 && tensao <= 408)
23  {
24    UV_index = "2";
25  }
26  char INDICE[58];
27  UV_index.toCharArray(INDICE,58);
28  client.publish("CASA/EXTERNA/UV", INDICE);
29 }
```

Fonte: Autor (2020)

#### 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Com a realização da experiência de automação residencial, por meio da internet das coisas, na maquete residencial utilizada, pode-se verificar que é possível encontrar soluções para problemas do cotidiano das pessoas. Os materiais utilizados na maquete demonstraram-se de fácil aquisição e, desde que bem instalados, mostraram-se com bons resultados.

Com a automação residencial, pode-se compreender os processos. Ou seja, como era uma maquete, foi possível verificar a conexão existente entre os equipamentos.

Caso esta experiência seja transportada para o mundo real, na vida real, seria possível o aumento da produtividade, uma vez que as pessoas ficariam livres de atividades que estariam automatizadas, para dedicarem-se mais às questões estratégicas, tendo mais tempo, inclusive para o lazer e convivência com a própria família.

A automação residencial pode ser útil, portanto, numa gestão mais eficiente, por ser mais fácil verificar os gargalos existentes e apontar soluções mais rápidas. Ao instalar os equipamentos e realizar a conexão na maquete utilizada no projeto descrito neste artigo, é possível, por exemplo, economizar o gasto de energia, por meio dos sensores de presença, que acionam os equipamentos somente quando, de fato, há pessoas para sua utilização.

A automação residencial permite a abolição de interruptores, bem como de outros dispositivos para ligar e desligar equipamentos eletrônicos, tais como ar condicionado, ventiladores, portões de acesso, entre outros.

Além disso, os sensores, permitem os alertas automáticos que ajudam na qualidade de vida das pessoas. No caso da experiência realizada, foi utilizado um *buzzer* que apitava quando a temperatura estava acima de um determinado valor e da umidade relativa do ar baixa. Pode-se verificar no experimento que o sensor utilizado para verificar a umidade não media quando a umidade estava abaixo de 30%, o que é comum em Brasília. Isto mostra que os equipamentos a serem instalados na automação residencial precisam considerar os fatores ambientais existentes para que se alcance os objetivos propostos no Projeto de Engenharia.

Em síntese, pode-se observar que a instalação da automação residencial pode proporcionar às pessoas economia, por meio de equipamentos inteligentes; segurança, uma vez que a iluminação acionada por sensores pode inibir a presença de invasores, conveniência; controle, por permitir o acionamento dos equipamentos eletrônicos, mesmo à distância, por meio de dispositivos ou aplicativos e conforto para quem a utiliza.

##### 4.1 Sensor de temperatura e dados do INMET

A figura 26 apresenta o gráfico com a variação de temperatura do dia 5 de dezembro de 2020, de acordo com dados oficiais do INMET.

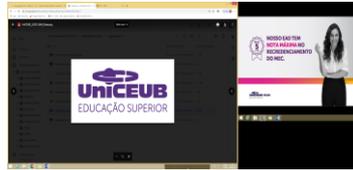
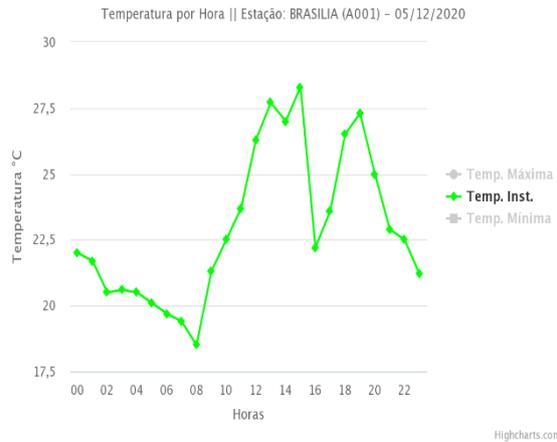


Figura 26. Gráfico de Temperatura por Hora  
Instituto Nacional de Meteorologia – INMET



Fonte: INMET (2020)

A partir dos dados recebidos do sensor de temperatura DHT11, foi realizada uma média de temperatura a cada hora do dia. Essas informações foram comparadas com os dados oficiais do INMET para se obter o erro percentual. É possível observar na tabela 3 que os dados apresentaram um erro percentual médio de 7,75%

Tabela 3.

Data/Hora Brasília	INMET (° C)	DHT11 (° C)	Erro (%)
5/12/20 8:00	18,5	19,3	4,15%
5/12/20 9:00	21,3	21,8	2,29%
5/12/20 10:00	22,5	24,1	6,64%
5/12/20 11:00	23,7	26	8,85%
5/12/20 12:00	26,3	27,5	4,36%
5/12/20 13:00	27,7	29,4	5,78%
5/12/20 14:00	27	28,5	5,26%
5/12/20 15:00	28,3	30,1	5,98%
5/12/20 16:00	22,2	26,1	14,94%
5/12/20 17:00	23,6	26,5	10,94%
5/12/20 18:00	26,5	28,7	7,67%
5/12/20 19:00	27,3	30,1	9,30%
5/12/20 20:00	25	29,5	15,25%
5/12/20 21:00	22,9	24,4	6,15%
5/12/20 22:00	22,5	24,1	6,64%
5/12/20 23:00	21,2	23,5	9,79%
<b>Média de erro percentual</b>			<b>7,75%</b>

Fonte: Autor (2020)

## 4.2 Atrasos causados após a conexão dos componentes eletrônicos

Tabela 4. Tempo de cada ação

Ação realizada pelo NodeMCU	Tempo (s)
<b>LEDS</b>	
Acionamento LEDs Via MQTT	1,48
<b>LEDS + LDR</b>	
Acionamento LEDs Via MQTT	1,81
Acionamento LED com sensor LDR	2,58
<b>LEDS + LDR + DHT11</b>	
Acionamento LEDs Via MQTT	2,11
Acionamento LED com sensor LDR	2,71
Leitura Temperatura/Umidade	1,00
<b>LEDS + LDR + DHT11 + HC-SR04</b>	
Acionamento LEDs Via MQTT	2,99
Acionamento LED com sensor LDR	2,93
Leitura Temperatura/Umidade	1,00
Acionamento da Ventilação	3,57
<b>LEDS + LDR + DHT11 + HC-SR04 + UV</b>	
Acionamento LEDs Via MQTT	3,74
Acionamento LED com sensor LDR	3,29
Leitura Temperatura/Umidade	1,00
Acionamento da Ventilação	4,12
Leitura Incidência UV	5,04

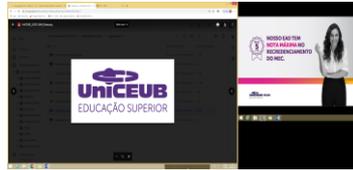
Fonte: Autor (2020)

Ao conectar objetos como sensores e LEDs ao NodeMCU, é possível verificar que o tempo entre um comando executado e a resposta do mesmo aumenta de acordo com a quantidade de componentes utilizados na placa de desenvolvimento.

Observa-se na tabela 4 o tempo gasto (em segundos) por cada ação no decorrer do desenvolvimento do projeto. O tempo de resposta aumenta de acordo com a quantidade de sensores e LEDs adicionados ao microcontrolador.

## 4.3 Visualização das informações obtidas através do aplicativo

O controle das lâmpadas, assim como o recebimento das informações obtidas pelos sensores, será realizado via aplicativo MQTT Dash. No app há uma central de controle, que possui a opção de ligar a desligar as lâmpadas



a partir de interruptores virtuais, além da possibilidade de verificar a temperatura, umidade e incidência de raios UV, conforme figura 27.

Figura 27. Controle via aplicativo MQTT Dash



Fonte: Autor (2020)

## 5 CONSIDERAÇÕES FINAIS

Foi verificado que o uso do NodeMCU é uma opção barata e eficaz para automatizar residências. Porém, durante o decorrer do projeto, pode-se notar que o microcontrolador tem um poder de processamento limitado quando utilizadas todas as suas portas de entrada/saída, resultando em um pequeno atraso entre o comando executado e a resposta obtida. De acordo com a experiência obtida, é recomendável usar mais de um NodeMCU para automações de maior porte, que utilizam muitos sensores e dispõe de muitas lâmpadas e/ou outros objetos de saída de dados.

O principal foco do projeto foi realizar a automação de baixo custo em uma residência com 4 ambientes. O projeto desenvolvido custou aproximadamente 269 reais, ignorando o preço da maquete em madeira, pois não é o foco do trabalho e serviu apenas como objeto de demonstração. Toda a automação pode ser aplicada na vida real, utilizando os mesmos componentes. A tabela 5 apresenta os gastos detalhados relacionados ao projeto.

Tabela 5.

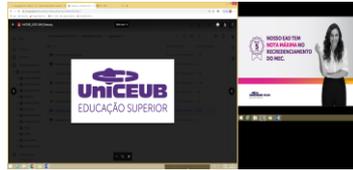
Material	Valor (R\$)
NodeMCU ESP8266	50,31
Sensor DHT11	15,21
Sensor UV	34,11
Sensor LDR	5,67
Sensor HC-SR04	9,45
Buzzer 5V	1,73
Ventoinha 5V	7,52
LED (6x)	7,50
Protoboard 830	18,81
Jumpers	30,49
Broker Dedicado	88,57
<b>Total</b>	<b>269,37</b>

Fonte: Autor (2020)

O objetivo deste artigo consistiu em demonstrar como a automação residencial por meio da Internet das Coisas pode colaborar no cotidiano das pessoas, proporcionando melhor qualidade de vida, além de controlar o consumo de energia.

Para analisar e verificar como a automação residencial colabora para a qualidade de vida das pessoas, buscou-se na literatura especializada do tema, argumentos que comprovam o quanto a automação residencial facilita o dia a dia das pessoas, possibilitando otimização do tempo, economia financeira, além de promoção da inclusão social.

Com a experiência desenvolvida com a simulação da automação residencial por meio do uso da construção da maquete, verificou-



se que é possível, por meio da internet das coisas, utilizar de modo mais eficiente e integrado os equipamentos de uma residência em prol de mais qualidade de vida das pessoas. Tudo isto decorre de um projeto de engenharia que deve otimizar o uso dos recursos tecnológicos disponíveis, aliando investimentos financeiros e praticidade.

## AGRADECIMENTOS

Agradecimentos especiais ao Professor Orientador deste artigo, Prof. MsC Francisco Javier de Obaldia Diaz, pelas orientações recebidas. Agradeço, também, a todos os professores do Curso de Engenharia de Computação do UNICEUB, por todo esforço e dedicação ao longo do curso.

## REFERÊNCIAS

ACCARDI, Adonisi; Eugeni Dodonov. Automação Residencial: **Elementos Básicos, Arquiteturas, Setores, Aplicações e Protocolos**. Disponível em: <<http://professor.pucgoias.edu.br/SiteDocente/admin/arquivosUpload/17829/material/ARTIGO02.pdf>>. Acesso em: 25 out. 2020.

ALVES, J. A.; MOTA, J. Coleção **Soluções, Casas Inteligentes**, Inova. Portugal: 2003.

BANZI, M. **Primeiros Passos com Arduino**. São Paulo: Novatec, 2012

BOLZANI, C. A. M. **Análise de Arquiteturas e Desenvolvimento de uma Plataforma para Residências Inteligentes**. 2010 155f. Tese (Doutorado) – Escola Politécnica da Universidade de São. Paulo. Departamento de Engenharia de Sistemas Eletrônicos, 2010. Disponível em: <[https://www.teses.usp.br/teses/disponiveis/3/3142/td-e-12082010-112005/publico/Tese\\_Caio\\_Augustus\\_Morais\\_Bolzani.pdf](https://www.teses.usp.br/teses/disponiveis/3/3142/td-e-12082010-112005/publico/Tese_Caio_Augustus_Morais_Bolzani.pdf)>. Acesso em: 16 nov. 2020.

CAMARGO, Valter Luís Arlindo de. **Elementos de automação**. São Paulo: Érica, 2014.

CARDOSO, Danielle Silva. **Aspectos Atuais da IOT: Características e desafios**. Ouro Preto, 2019. Disponível em: <<https://www.monografias.ufop.br/bitstream/3540000>

0/2569/1/MONOGRAFIA\_AspectosAtuaisIot>. Acesso em: 16 nov. 2020

COSTA, Eduardo Naccache Martins de - UFRJ. **Um simulador do processador Sapiens no NodeMCU**. 2018. Disponível em: <<https://pantheon.ufrj.br/bitstream/11422/5495/3/Eduardo%20Naccache.pdf>> . Acesso em: 05 de novembro de 2020.

EVANS, Dave. **A Internet das Coisas. Como a próxima evolução da Internet está mudando tudo**. Disponível em: <[https://www.cisco.com/c/dam/global/pt\\_br/assets/executives/pdf/internet\\_of\\_things\\_iiot\\_ibsg\\_0411final.pdf](https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iiot_ibsg_0411final.pdf)>. Acesso em: 17 nov. 2020.

FLIPEFLOP. Disponível em: <<https://www.flipeflop.com/>>. Acesso: 15 nov. 2020.

JAFFEY, Toby. MQTT and CoAP, IoT protocols. 2014. Disponível em: <[http://www.eclipse.org/community/eclipse\\_newsletter/2014/february/article2.php](http://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php)>. Acesso em 16 de outubro de 2020.

MARTINS, Ismael Rodrigues; ZEM, José Luís. **Estudo dos protocolos de comunicação MQTT e COaP para aplicações machine-to-machine e Internet das coisas**. Revista Tecnológica da Fatec Americana, 2015, 3.1: 24p.-24p. Disponível em: <<https://fatecbr.websiteseuro.com/revista/index.php/RTecFatecAM/article/view/41>> Acesso em : 10 nov. 2020.

SUNDMAEKER, H.; GUILLEMIN, P.; FRIESS, P.; WOELFFLÉ, S. **Vision and challenges for realising the internet of things. Cluster of European Research Projects on the Internet of Things**, European Commission, v. 3, n. 3, p. 34–36, 2010.

INMET. 2020. **Instituto Nacional de Meteorologia**. Disponível em: <https://tempo.inmet.gov.br/>. Acesso em: 05 dez. 2020.

YUAN, Michael. **Conhecendo o MQTT**. 2017. Disponível em: <<https://developer.ibm.com/br/technologies/iot/article/s/iot-mqtt-why-good-for-iot/>>. Acesso em 20 out. 2020. Acesso em: 15 nov. 2020.

TÓFOLI, Ricardo José. **Casa Inteligente – Sistema de Automação Residencial**. 2014. Disponível em: <<https://cepein.femanet.com.br/BDigital/arqTccs/1211320586.pdf>>. Acesso em: 10 nov. 2020.