

**FUNDAÇÃO CARMELITANA MÁRIO PALMÉRIO  
FACULDADE DE CIÊNCIAS HUMANAS E SOCIAIS  
CURSO DE ENGENHARIA CIVIL**

**RHENANN DAYVID MARINS CARDOSO**

**AUTOMAÇÃO RESIDENCIAL**

**MONTE CARMELO – MG  
DEZEMBRO / 2018**

**RHENANN DAYVID MARINS CARDOSO**

**AUTOMAÇÃO RESIDENCIAL**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Civil, da Faculdade de Ciências Humanas e Sociais da Fundação Carmelitana Mário Palmério – FUCAMP, para obtenção do grau de bacharel em Engenharia Civil.

Orientador: Prof. Rafael Fernandes Garcia

**MONTE CARMELO – MG  
DEZEMBRO / 2018**

Página reservada para a Folha de Aprovação

## **AGRADECIMENTOS**

Primeiramente à Deus por ter me dado saúde e força para superar as dificuldades;

À minha família, pelo amor, incentivo e apoio incondicional;

Aos meus amigos, que me ajudaram e me apoiaram na minha formação;

Ao meu orientador, pela ajuda na elaboração deste trabalho;

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado!

“Não importa o que aconteça, continue a nadar”  
(WALTERS, GRAHAM; Procurando Nemo, 2003).

## **RESUMO**

Este trabalho se inicia com o estudo dos componentes utilizados no sistema e as linguagens de programação mais conhecidas. Em seguida é mostrado o desenvolvimento, a comunicação e a configuração do sistema. A ideia no geral é mostrar a criação de um sistema de automação residencial de baixo custo com um sistema livre, onde este é capaz de acionar aparelhos elétricos e lâmpadas, além de monitorar a temperatura e a umidade ambiente. Tudo isso controlado pelo celular ou qualquer outro dispositivo que consiga acessar uma página da internet.

**PALAVRAS-CHAVE:** Automação residencial; NodeMcu; Arduino.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Arduino UNO. ....	14
Figura 2 - DHT11 .....	17
Figura 3 - Relé. ....	18
Figura 4 - Conexão do NodeMcu a internet. ....	20
Figura 5 - IP do site gerado, apresentado pelo monitor serial (Imagem ilustrativa). ....	21
Figura 6 - Gerando IP do endereço do site. ....	22
Figura 7 - Envio dos comandos ao relé pelo serial feito pelo NodeMcu.....	23
Figura 8 - Leitura/recebimento dos comandos do relé pelo arduino. ....	23
Figura 9 - Página de controle residencial com todos os relés desligados.....	24
Figura 10 - Página de controle residencial com todos os relés ligados. ....	24
Figura 11 - Página de controle residencial com os relés desligados (a) e ligados (b), visualizada pelo computador. ....	25
Figura 12 - Página de controle residencial com os relés desligados (a) e ligados (b), visualizada pelo celular. ....	26

## **LISTA DE TABELAS**

Tabela 1 - Quantitativo de materiais utilizados.....	19
------------------------------------------------------	----



## LISTA DE ABREVIATURAS E SIGLAS

### ABREVIATURAS

A	Ampere
V	Volt

### SIGLAS

IoT	Internet of Things (Internet das coisas)
IDE	Integrated Development Environment (Ambiente de desenvolvimento integrado)
USB	Universal Serial Bus (Barramento serial universal)
SSID	Service Set Identifier (Identificador do conjunto de serviços)
IP	Internet Protocol (Protocolo de internet)
PWM	Pulse Width Modulation (Modulação de largura de pulso)
LED	Light Emitting Diode (Diodo emissor de luz)

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	10
<b>1.1 Objetivos</b> .....	11
<i>1.1.1 Objetivos gerais</i> .....	11
<i>1.1.1 Objetivos específicos</i> .....	11
<b>2 REFERENCIAL TEÓRICO</b> .....	12
<b>2.1 Linguagens de programação mobile</b> .....	16
<b>2.2 DHT11</b> .....	17
<b>2.3 Relé</b> .....	17
<b>3 METODOLOGIA</b> .....	19
<b>3.1 Materiais utilizados</b> .....	19
<b>3.2 Procedimentos</b> .....	19
<i>3.2.1 Conexão</i> .....	19
3.2.1.1 Conectando o <i>NodeMcu</i> ao sistema de internet local .....	20
3.2.1.2 Obtendo endereço IP .....	21
<i>3.2.2 Comunicação entre o NodeMcu e o arduino</i> .....	22
<i>3.2.3 Página que é utilizada pelo usuário para o uso da automação residencial</i> .....	24
<b>4 RESULTADOS E DISCUSSÃO</b> .....	25
<b>5 CONCLUSÃO</b> .....	27
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	28
<b>APÊNDICE A</b> .....	29

## 1 INTRODUÇÃO

O intuito deste trabalho é mostrar que o cenário atual da engenharia civil no Brasil vem crescendo gradativamente em um ritmo elevado, devido a necessidade de expansão e criação de centros urbanos cada vez mais desenvolvidos. Com isso tornou-se importante ocorrer modificações de modo progressivo de tal forma a atender as necessidades da sociedade com o auxílio da tecnologia.

A tecnologia avança em larga escala conforme a demanda humana com a finalidade de tornar as tarefas diárias mais práticas e com isso deixando a vida de uma forma mais simplificada e funcional. A automação residencial é considerada como algo de alto grau tecnológico, devido a ser uma novidade e ainda pouco difundida pela população.

Segundo José Cândido Forti, presidente da AURESIDE:

Transformar casas em confortáveis refúgios capazes de oferecer segurança e economia de custos é uma das vantagens da automação residencial. O que antes parecia ser um privilégio apenas da família Jetson, começa a se difundir nos empreendimentos residenciais de alto nível, transformando o conceito de casa do futuro em casa do presente.

A automação residencial, também conhecido como domótica, que é derivada da junção da palavra latina “Domus” (que significa “casa”) com “Robótica”. Esta junção dá o sentido de facilidade em executar as tarefas cotidianas com maior praticidade, segurança e conforto para os moradores, onde o intuito é que os equipamentos estejam conectados e que “conversem” entre si por meio de uma central inteligente e programável. A central pode ser conectada a diversos tipos de equipamentos, como por exemplo: lâmpadas, sensores, temporizadores e outros equipamentos em geral.

Com o uso da IoT (*internet of things*) que em português significa “internet das coisas” é possível conectar objetos físicos com a internet, criando assim uma comunicação entre eles capaz de transmitir e coletar dados. Com a IoT é possível deixar os equipamentos acessíveis remotamente, podendo controlá-los a distância, sendo necessário apenas uma conexão com a internet.

Além da economia com energia, também haverá um grande ganho em comodidade e qualidade de vida, deixando o ambiente residencial mais prático e “na palma das mãos”.

## **1.1 Objetivos**

### ***1.1.1 Objetivos gerais***

Evidenciar a criação de um sistema de automação residencial baseada em *hardware* e *software* livre no ramo da construção civil, abordando as vantagens e desafios desta tecnologia.

### ***1.1.1 Objetivos específicos***

- Demonstrar a utilização da automação residencial para promover a comodidade;
- Desenvolver um projeto levando em conta seu custo/benefício.

## 2 REFERENCIAL TEÓRICO

Conforme dito por Bolzani (2004, p. 16), qualquer dispositivo que que consiga se comunicar com outro dispositivo, mesmo sendo uma tarefa básica, possibilitando acesso remoto e ter capacidade de tomada de decisão, pode ser considerado um dispositivo inteligente.

Apesar de não ser conhecido pela maioria das pessoas, o conceito de casas inteligentes já é bastante antigo, começando a surgir por volta dos anos 2000, fazendo assim o aparecimento de dispositivos inteligentes.

O termo “Domótica”, tem o significado de “Casa”, sendo derivado da palavra em latim “*Domus*”. Com o emprego de tecnologias atuais, pode-se criar sistemas automáticos de controle, incluindo residências, automatizando as atividades repetitivas, tornando-as inteligentes (BOLZANI, 2004, p. 7).

Segundo Bolzani (2004, p. 7), o que difere uma residência convencional de uma residência inteligente é:

- Os dispositivos utilizados, desenvolvem funções extras, com isso contribui para uma melhor gestão da residência.
- Gera uma economia muito maior de energia, pois sua construção e arquitetura inteligente reduz as taxas de utilização dos equipamentos desnecessariamente com isso, fazendo o bem ao meio ambiente, possibilitando um uso mais consciente das fontes naturais de energia.

O termo “Smart” que em tradução livre significa “Inteligente”, tem sido bastante empregado para promover os novos eletroeletrônicos e eletrodomésticos que vem saindo no mercado. Este termo passa a ideia de que são produtos com várias funcionalidades, mas muitas vezes não seria o mais indicado chama-los assim. Muitos deles não são capazes de aprender, compreender, interpretar ou desenvolver alguma forma de intelecto ou perspectiva (BOLZANI, 2004, p. 8).

Qualquer equipamento que tenha autonomia para desenvolver uma tarefa básica, efetuar troca de informações com outros dispositivos, possibilitar comando remoto e ter capacidade de tomada de decisão pode ser considerado um dispositivo inteligente (BOLZANI, 2004, p. 16).

De acordo com (McRoberts, 2011), um arduino é um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele. McRoberts (2011) também diz, o arduino é o que chamamos de plataforma

de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de *hardware* e *software*.

De acordo com (Banzi, Cuartielles, Igoe, Martino, & Mellis, 2018), o arduino é uma plataforma de prototipagem eletrônica *open-source* que se baseia em *hardware* e *software* flexíveis e fáceis de usar. É destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos.

O arduino utiliza a plataforma “*open-source*”, ou seja, uma plataforma livre, de código aberto, onde o autor fornece os direitos, sendo possível modificar e distribuir o *software* de forma livre, sem nenhum custo.

Como dito anteriormente, o arduino é um pequeno computador. Por isso, para que ele possa funcionar é preciso que o mesmo seja programado. A programação nada mais é que listar as instruções e a sequência de atos que ele deve interpretar e tomar de acordo com a circunstância preestabelecida. Para que seja executado como esperado, o código deve ser escrito com uma sequência lógica para tomada de decisão.

Para programar o arduino é necessário utilizar a sua *open-source* IDE (*Integrated Development Environment*) que em tradução livre significa “Ambiente de Desenvolvimento Integrado”, que pode ser baixado gratuitamente no próprio site do arduino ([www.arduino.cc](http://www.arduino.cc)).

O arduino IDE, é o *software* onde iremos ensinar o arduino a desempenhar as funcionalidade e atividades que pretendemos executar, para isso deve ser escrito na linguagem de programação entendida pelo arduino, que no caso é a C/C++.

De acordo com (Gotardo, 2015):

Uma linguagem de programação é um método padronizado que usamos para expressar as instruções de um programa a um computador programável. Ela segue um conjunto de regras sintáticas e semânticas para definir um programa de computador. Regras sintáticas dizem respeito à forma de escrita e regras semânticas ao conteúdo.

Ao usar uma linguagem de programação, você cria o chamado “Código Fonte”. Um código fonte é um conjunto de palavras escritas de acordo com as regras sintáticas e semânticas de uma linguagem (GOTARDO, 2015).

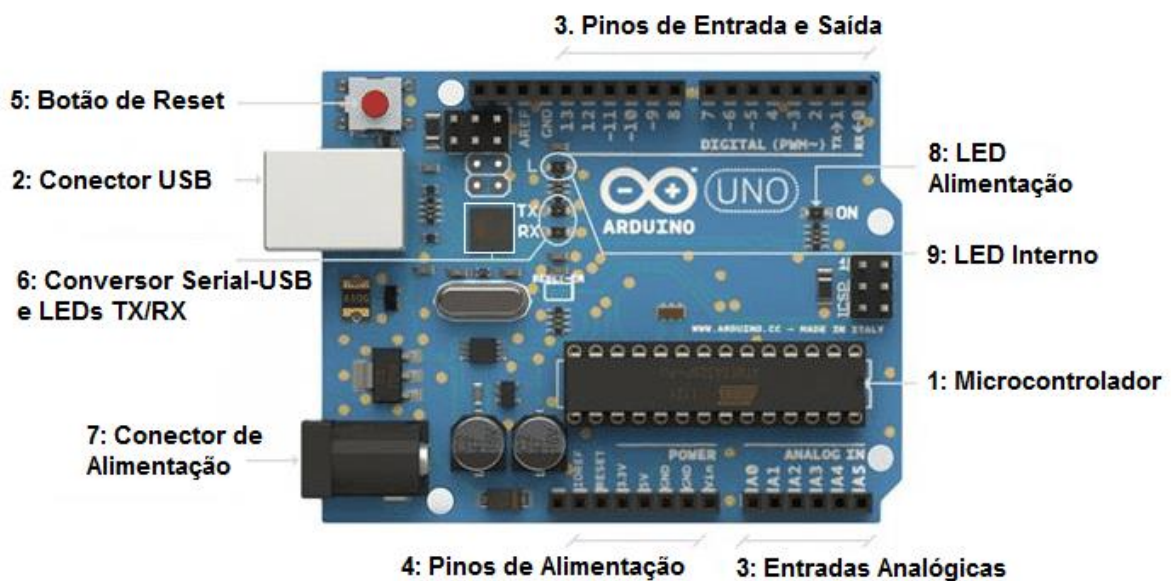
Podemos ainda dizer, segundo (Gotardo, 2015), que código fonte não é executado pelo computador, o computador apenas entende código binário. Assim, é necessário traduzir o código fonte para o formato que o computador entenda, O formato que o computador entende é chamado de “código de máquina”. O processo de “tradução” é chamado de “compilação”.

Compiladores são programas especiais que traduzem códigos fontes em códigos alvos (ou código objeto). Trata-se de um código que é executável num computador (GOTARDO, 2015).

Após a compilação do código, caso não tenha nenhum erro de contexto, ele é transferido para a placa já em formato compreensível pelo arduino.

Na Figura 1 pode-se observar o arduino UNO, a placa mais famosa entre a família dos arduino.

Figura 1 - Arduino UNO.



Fonte: Mota (2017).

Na figura também estão listados os principais componentes da placa. Uma breve explicação seria:

**1. Microcontrolador:** esse é o cérebro do arduino. Um computador inteiro dentro de um pequeno chip. Este é o dispositivo programável que roda o código que enviamos à placa (MOTA, 2017).

Segundo Angnes (2003), em um microcontrolador podemos encontrar memória, CPU, entradas e saídas. Alguns ainda possuem periféricos como conversores A/D e D/A, comparadores. O microcontrolador chega a custar muitas vezes mais barato do que um transistor.

Mota (2017) ainda diz: “Nessas placas foram adotados os microcontroladores da Microchip, que inicialmente eram produzidos pela Atmel, mais especificamente a linha ATmega. O modelo UNO, por exemplo, usa o microcontrolador ATmega328”.

**2. Conector USB:** o conector USB (*Universal Serial Bus*), segundo a Wikipédia (2018), é um tipo de conexão "ligar e usar" que permite a fácil conexão de periféricos sem a necessidade de desligar o computador.

De acordo com Mota (2017), conecta a placa ao computador. É por onde o computador e o arduino se comunicam com o auxílio de um cabo USB, além de ser uma opção de alimentação da placa.

**3. Pinos de entrada e saída:** pinos que podem ser programados para agirem como entradas ou saídas fazendo com que o arduino interaja com o meio externo. O UNO R3 possui 14 portas digitais (I/O), 6 pinos de entrada analógica e 6 saídas analógicas (PWM) (MOTA, 2017).

PWM (*Pulse Width Modulation*) em tradução livre seria “Modulação de Largura de Pulso”, segundo Silveira, s.d., refere-se ao conceito de pulsar rapidamente um sinal digital em um condutor. Além de várias outras aplicações, esta técnica de modulação pode ser utilizada para simular uma tensão estática variável.

**4. Pinos de alimentação:** fornecem diversos valores de tensão que podem ser utilizados para energizar os componentes do seu projeto. Devem ser usados com cuidado, para que não sejam forçados a fornecer valores de corrente superiores ao suportado pela placa (MOTA, 2017).

**5. Botão reset:** É o botão responsável por reiniciar a placa. Quando ele é pressionado, todo o ciclo do arduino volta para estado inicial.

**6. Conversor Serial-USB e LEDs TX/RX:** para que o computador e o microcontrolador conversem, é necessário que exista um chip que traduza as informações vindas de um para o outro. Os LEDs TX e RX acendem quando o arduino está transmitindo e recebendo dados pela porta serial (MOTA, 2017).

O RX é o responsável pela recepção de dados, enquanto o TX é o responsável pela transmissão dos dados para o computador.

**7. Conector de alimentação:** de acordo com Mota (2017):

Responsável por receber a energia de alimentação externa, que pode ter uma tensão de no mínimo 7 Volts e no máximo 20 Volts e uma corrente mínima de 300mA.



Recomendamos 9V, com um pino redondo de 2,1mm e centro positivo. Caso a placa também esteja sendo alimentada pelo cabo USB, ele dará preferência à fonte externa automaticamente.

**8. Led de alimentação:** quando ela está acesa, significa que a placa está energizada.

**9. Led interno:** este led é conectado ao pino digital 13.

## 2.1 Linguagens de programação mobile

**Java:** Totalcross (2015) diz que é a linguagem mais difundida no mundo e é capaz de utilizar multiplataforma sendo muito utilizado para sistemas *Android* e *Windows*.

**C:** desenvolvida nos anos 70, é como uma língua-mãe; dela se originou Java, C# e outras. Sendo bastante complexa, exige muitas linhas de comando, enquanto seria possível fazer em apenas uma linha utilizando outra linguagem (TATALCROSS, 2015).

**C++:** de acordo com Totalcross (2015), ela é derivada do C e é muito utilizada pelo fácil manuseio e por ser multiplataforma, também já foi utilizado para criação de softwares grandiosos, por exemplo o *photoshop*.

**Python:** excelente para iniciantes por utilizar uma sintaxe clara e didática, mas é bem robusta, demonstra resultados sofisticados. Muito utilizado em apps populares como o Instagram, *Pinterest* e outros (TOTALCROSS, 2015).

**Objective-C:** segundo Totalcross (2015), essa linguagem é bastante utilizada pela *Apple*, com mais foco em funções gráficas e de exibição, sendo a base de aplicativos feitos para *iOS* e *Mac OS X*.

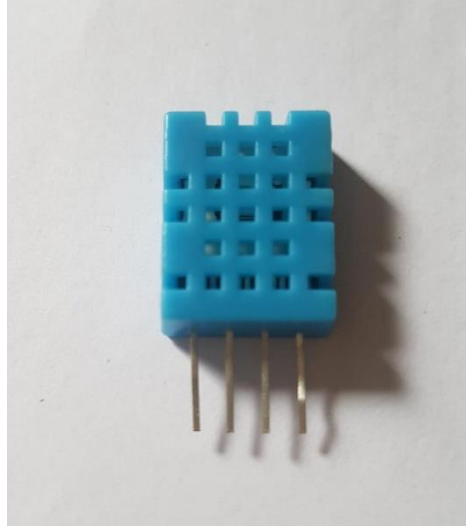
**Swift:** é a nova linguagem da *Apple* que está substituindo o *Objective-C*. Também é a melhor opção de linguagem para *iOS*, além de ter sido anunciada que será transformada em uma linguagem de código aberto, tendo o intuito de aumentar os interessados nela, e com isso, transformá-la em uma linguagem mais popular (TOTALCROSS, 2015).

**C#:** conforme Totalcross (2015) esclareceu, é a linguagem da *Microsoft* não muito difundida no mundo de linguagens *mobile*, não podendo ser descartada pelo fato da credibilidade passada pelos produtos do *Windows* no Brasil.

## 2.2 DHT11

Segundo Queiroz et al. (2015), o DHT11 (Figura 2) é um sensor digital de temperatura e umidade que permite medir temperaturas de 0 a 50 graus Celsius com precisão de 2,0° Celsius para mais ou para menos, e umidade na faixa de 20 a 90% com precisão de mais ou menos 5%.

Figura 2 - DHT11



Fonte: O autor (2018).

O sensor envia os dados para o microcontrolador utilizando apenas um pino. Os outros dois são **Vcc** e **GND**, sendo que o terceiro pino não é utilizado (THOMSEN, 2015).

## 2.3 Relé

Os relés (Figura 3) basicamente são dispositivos elétricos que tem como função produzir modificações súbitas, porém predeterminadas em um ou mais circuitos elétricos de saída (ELÉTRICA, 2018).

Figura 3 - Relé.



Fonte: O autor (2018).

### 3 METODOLOGIA

Este trabalho apresenta uma pesquisa exploratória, procurando se aprofundar no problema proposto, que seria a criação de um sistema de automação residencial de baixo custo, levando assim a aplicação do conhecimento levantado a partir do processo de investigação.

As técnicas de pesquisa utilizados na elaboração deste trabalho foram o método bibliográfico, o método documental e com abordagem qualitativa.

#### 3.1 Materiais utilizados

Todos os materiais necessários para a instalação da automação na residência se encontra no quantitativo mostrado na Tabela 1. Os valores de custo demonstrados na Tabela 1, refere-se ao valor pago pelos produtos na data de criação deste artigo.

Tabela 1 - Quantitativo de materiais utilizados.

Quantidade	Item	Valor
1	NodeMcu WiFi ESP8266	R\$39,90
1	Arduino Nano	R\$24,90
1	DTH11- Sensor de Umidade e temperatura	R\$9,90
1	Módulo Relé 8 canais 5v	R\$39,90
1	Fonte de alimentação 9V 2A	R\$12,90
1	Protoboard 830 pontos	R\$16,90
1	Kit jumpers 140 conectores	R\$18,90
Total		R\$ 163,30

Fonte: O autor (2018).

#### 3.2 Procedimentos

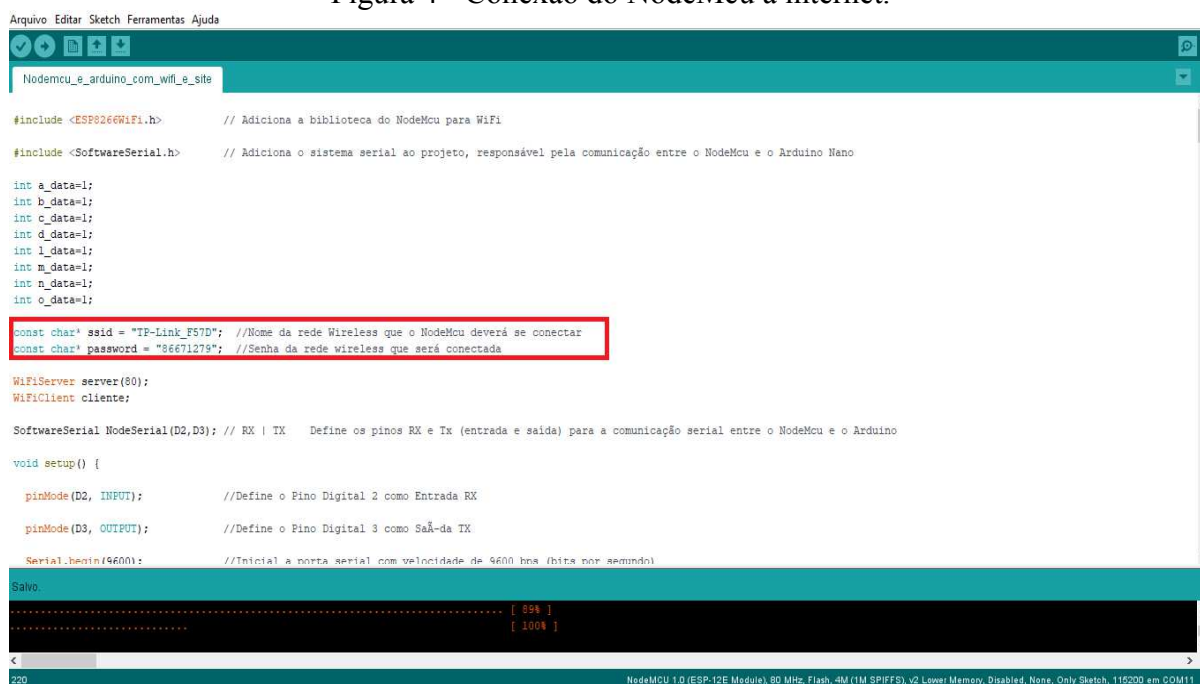
##### 3.2.1 Conexão

Para a conexão do projeto, o método escolhido foi o Wi-Fi. Um dos fatores que influenciaram nesta escolha foi o seu alcance, que está disponível em toda a área da residência. O dispositivo responsável pela conexão com a internet residencial é o *NodeMcu*, por ele já vir com um chip *wireless* embutido (chip ESP8266). Por motivos de segurança, foi definido que a rede de automação residencial só poderia ser controlada localmente, ou seja, apenas os dispositivos que estiverem conectados na mesma rede doméstica que o *NodeMcu* poderão acessá-lo e realizar o controle do sistema de automação.

### 3.2.1.1 Conectando o *NodeMcu* ao sistema de internet local

Primeiramente tem que se definir em qual rede o sistema de automação será conectado. Após a escolha, é necessário acessar a configuração do *Nodemcu*. O programa utilizado para a programação do mesmo foi o Arduino IDE, o mesmo utilizado para a programação do arduino, que será utilizado mais à frente. A conexão do *NodeMcu* à internet é mostrada na Figura 4.

Figura 4 - Conexão do *NodeMcu* a internet.



```

Arquivo Editar Sketch Ferramentas Ajuda
NodeMcu_e_arduino_com_wifi_e_site

#include <ESP8266WiFi.h> // Adiciona a biblioteca do NodeMcu para WiFi
#include <SoftwareSerial.h> // Adiciona o sistema serial ao projeto, responsável pela comunicação entre o NodeMcu e o Arduino Nano

int a_data=1;
int b_data=1;
int c_data=1;
int d_data=1;
int l_data=1;
int m_data=1;
int n_data=1;
int o_data=1;

const char* ssid = "TP-Link_F57D"; //Nome da rede Wireless que o NodeMcu deverá se conectar
const char* password = "86671279"; //Senha da rede wireless que será conectada

WiFiServer server(80);
WiFiClient cliente;

SoftwareSerial NodeSerial(D2,D3); // RX | TX Define os pinos RX e TX (entrada e saída) para a comunicação serial entre o NodeMcu e o Arduino

void setup() {

  pinMode(D2, INPUT); //Define o Pino Digital 2 como Entrada RX
  pinMode(D3, OUTPUT); //Define o Pino Digital 3 como Saída TX

  Serial.begin(9600); //Inicial a porta serial com velocidade de 9600 bps (bits por segundo)
}

Salvo.
..... [ 89% ]
..... [ 100% ]

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 em COM11

```

Fonte: O autor (2018).

Para se definir a rede que será utilizada pelo sistema, é preciso colocá-los na programação do *NodeMcu*, a configuração segue da seguinte maneira:

O nome da rede em que se deseja conectar deve ser informado no campo “SSID” que significa “*service set identifier*” (identificador do conjunto de serviços, em tradução livre), ou seja, é o nome que se dá a rede *wireless* para que ela seja visível ao usuário. Lembrando que o SSID deve ser adicionado dentro de aspas duplas, no modelo utilizado neste trabalho, por exemplo, foi “TP-Link\_F57D”. Como mostrado na Figura 4.

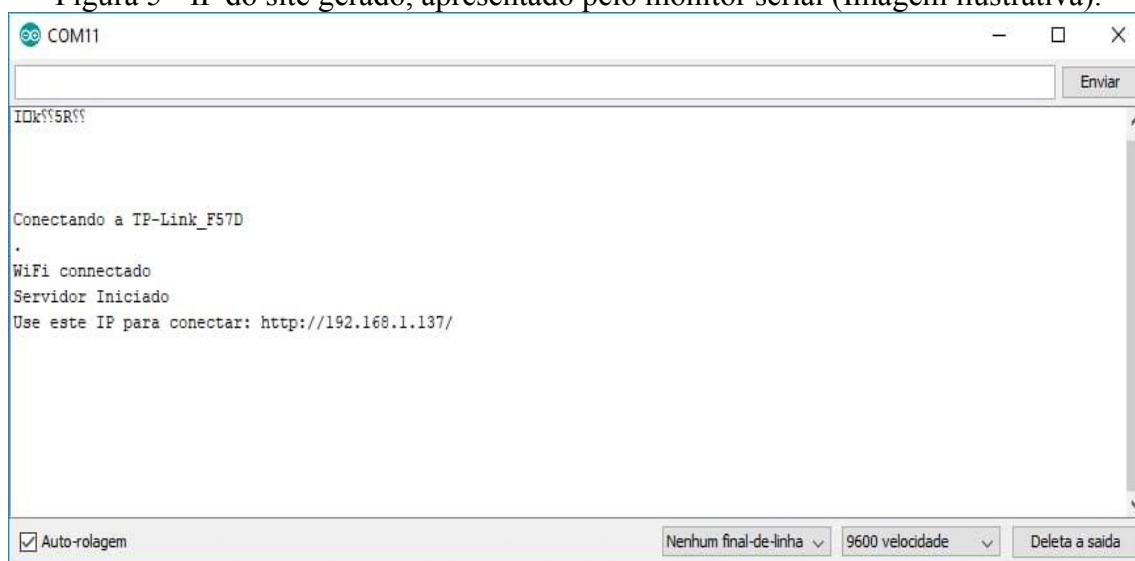
A seguir deve se adicionar a senha, que deve ser a mesma utilizada para a conexão dos dispositivos a rede sem fio, ela deverá ser digitada no campo “*Password*” (senha, em tradução livre), no caso deste trabalho foi utilizado a senha “86671279”, também mostrada na Figura 4.

### 3.2.1.2 Obtendo endereço IP

O método de controle dos dispositivos que utilizam a interface da automação residencial é realizado por meio de um site da internet, que utiliza qualquer navegador para acessá-lo, mas como todo site, precisa de um endereço para ser acessado, esse endereço é chamado de IP. No caso deste site em específico, está armazenado no *Nodemcu*, mesmo assim também precisa de um IP para ser acessado. Lembrado que só poderá ser acessado localmente, ou seja, dentro dos limites de alcance da rede Wi-Fi doméstica.

Neste caso, o IP é gerado de acordo com a disposição de endereços locais da rede local, mas por padrão ele sempre será localizado em 192.168.1.137. Caso tenha necessidade da verificação do endereço IP, basta abrir o monitor serial do próprio Arduino IDE e ele será mostrado, como na Figura 5 mostrada logo abaixo.

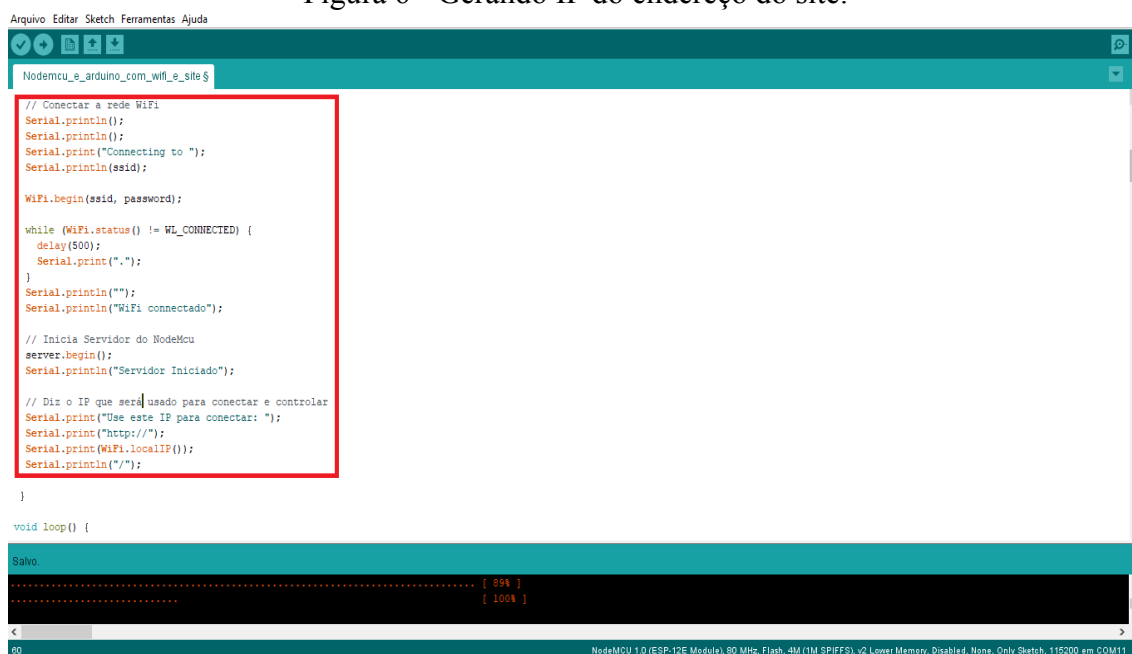
Figura 5 - IP do site gerado, apresentado pelo monitor serial (Imagem ilustrativa).



Fonte: O autor (2018).

A parte do código de programação responsável por conectar o sistema de automação residencial na rede local e gerar o IP é o demonstrado na Figura 6.

Figura 6 - Gerando IP do endereço do site.



```

Arquivo Editar Sketch Ferramentas Ajuda
Nodemcu_e_arduino_com_wifi_e_site $

// Conectar a rede WiFi
Serial.println();
Serial.println();
Serial.print("Conectando to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi conectado");

// Inicia Servidor do NodeMcu
server.begin();
Serial.println("Servidor Iniciado");

// Diz o IP que será usado para conectar e controlar
Serial.print("Use este IP para conectar: ");
Serial.print("http://");
Serial.print(WiFi.localIP());
Serial.println("/");

}

void loop() {

Salvo
..... [ 099 ]
..... [ 1009 ]

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 em COM11

```

Fonte: O autor (2018).

Após ser executado essa parte do código demonstrado na Figura 6, é então gerado o IP de conexão com o site, a parte que demonstra o IP propriamente dito seria a linha “Serial.print(WiFi.localIP());”.

### 3.2.2 Comunicação entre o *NodeMcu* e o arduino

Por questões de praticidade, foi definido que seria utilizado o *NodeMcu* juntamente com o arduino no sistema de automação residencial. O *NodeMcu* ficando responsável pela conexão do sistema com a rede *wireless* e também com a hospedagem do site que o usuário irá se conectar para realizar o controle residencial. Já o arduino, ficou com a parte de controle dos periféricos, sendo eles, o módulo relé e DHT11. Essa função foi atribuída ao arduino por ele já trabalhar com 5V nativos em suas portas programáveis, que no caso é a voltagem de trabalho do módulo relé. Ele ainda tem uma saída de 5V que pode ser utilizada para energizar outros periféricos, no projeto foi utilizado para energizar o DHT11.

Para a comunicação entre os dois dispositivos, foi utilizada a comunicação serial, conectando a porta D2 do *NodeMcu* com a porta D2 do arduino, sendo utilizada como TX e também conectada a porta D3 do *NodeMcu* com a porta D3 do arduino, funcionando como RX.

Para que se haja comunicação, é preciso que os dois dispositivos consigam mandar e receber os dados, nas Figuras 7 e 8 mostra como ficou o trecho do código em que se realiza a comunicação dos dois dispositivos para o acionamento do relé.

Figura 7 - Envio dos comandos ao relé pelo serial feito pelo NodeMcu.

```

Arquivo Editar Sketch Ferramentas Ajuda
NodeMcu_e_arduino_com_wifi_e_sita
-----
//Serial.print(a_data); Serial.print("\t");
// Serial.println(b_data);
NodeSerial.print(a_data); NodeSerial.print(" ");
NodeSerial.print(b_data); NodeSerial.print("\n");
NodeSerial.print(c_data); NodeSerial.print(" ");
NodeSerial.print(d_data); NodeSerial.print("\n");
NodeSerial.print(l_data); NodeSerial.print(" ");
NodeSerial.print(m_data); NodeSerial.print("\n");
NodeSerial.print(n_data); NodeSerial.print(" ");
NodeSerial.print(o_data); NodeSerial.print("\n");

delay(100);

// Verifica se alguém se conectou
WiFiClient client = server.available();
if (!client) {
  return;
}

Salvo.
..... [ 89% ]
..... [ 100% ]

381 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 em COM11

```

Fonte: O autor (2018).

Figura 8 - Leitura/recebimento dos comandos do relé pelo arduino.

```

Arquivo Editar Sketch Ferramentas Ajuda
Comunicacao_Arduino_NodeMCU_2$
-----
delay(100);

while (NanoSerial.available() > 0)
{
  int a_data = NanoSerial.parseInt();
  int b_data = NanoSerial.parseInt();
  int c_data = NanoSerial.parseInt();
  int d_data = NanoSerial.parseInt();
  int l_data = NanoSerial.parseInt();
  int m_data = NanoSerial.parseInt();
  int n_data = NanoSerial.parseInt();
  int o_data = NanoSerial.parseInt();

  if (NanoSerial.read() == '\n')
  {
    //Serial.print("NodeMCU or ESP8266"); Serial.print(" : ");
  }
}

Carregado
O sketch usa 5970 bytes (15%) de espaço de armazenamento para programas. O máximo são 30720 bytes.
Variáveis globais usam 330 bytes (16%) de memória dinâmica, deixando 1718 bytes para variáveis locais. O máximo são 2048 bytes.

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 em COM11

```

Fonte: O autor (2018).

Na Figura 7, é mostrado o código contido no *NodeMcu* que é responsável pelo envio dos comandos de acionamento dos relés pela serial. Já na Figura 8 está mostrando o código presente no arduino, que faz a leitura da serial e encaminha cada comando recebido para seu respectivo relé. Com essa comunicação estabelecida, os dois dispositivos estão conversando e sendo assim possível que cada um faça sua parte.



### 3.2.3 Página que é utilizada pelo usuário para o uso da automação residencial

Como dito anteriormente, para poder se controlar a residência, foi escolhido o uso de uma página de internet que pode ser rodada em qualquer navegador. Essa página está armazenada no *NodeMcu*, sendo assim possível o controle residencial mesmo sem a presença de conexão com a internet, mas lembrando que a conexão com o Wi-Fi local é indispensável.

Para a criação desta página foi utilizado o HTML e um pouco de CSS, apresentado em Apêndice A.

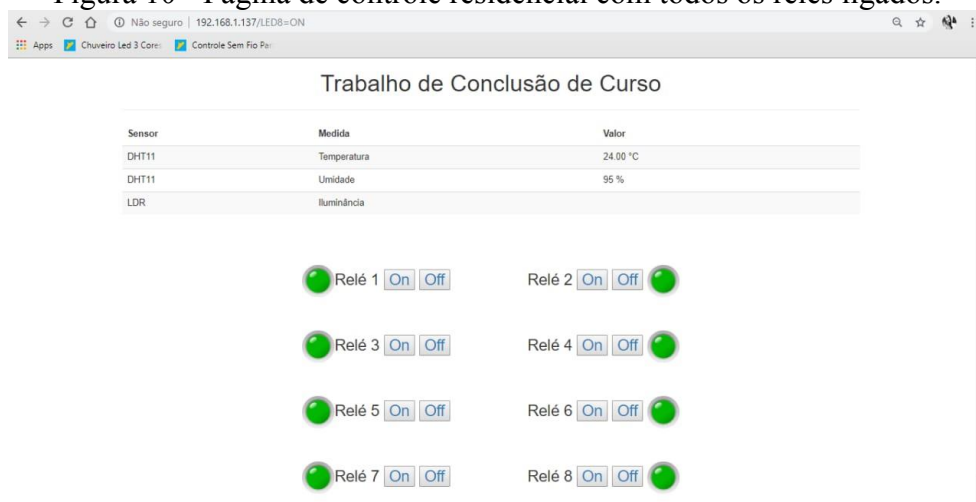
A página do código descrita no Apêndice A, quando aberto no navegador, ficou conforme mostrado na Figuras 9 e 10.

*Figura 9 - Página de controle residencial com todos os relés desligados.*



Fonte: O autor (2018).

*Figura 10 - Página de controle residencial com todos os relés ligados.*



Fonte: O autor (2018).

## 4 RESULTADOS E DISCUSSÃO

Os testes resultaram em uma comunicação bem precisa entre o sistema, as respostas foram satisfatórias, funcionando conforme o esperado. A distância de alcance para controle do sistema foi ampla, sendo a distância oferecida pelo roteador Wi-Fi, tornando possível o controle por toda a residência de teste.

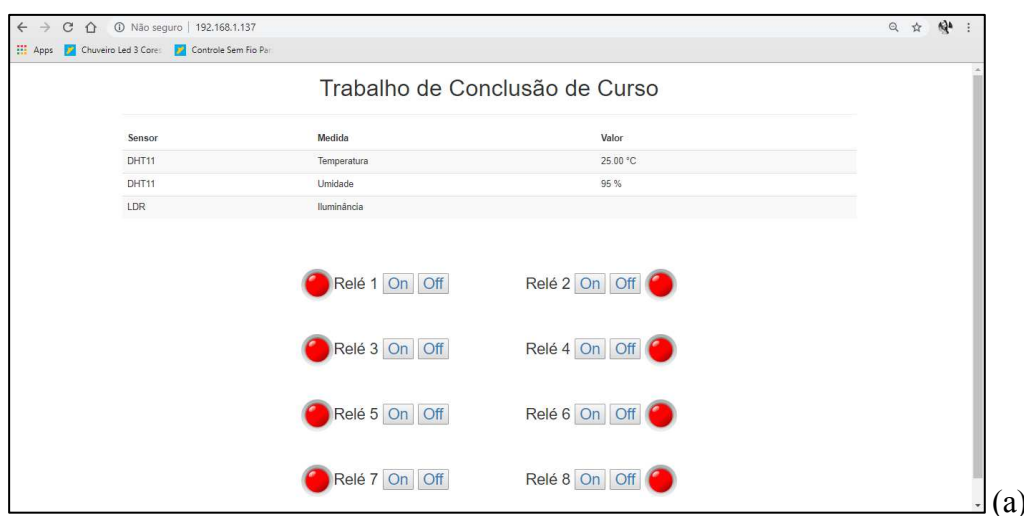
O sistema conta com 8 saídas nomeadas como “Relé”, onde por elas podem ser ligados qualquer aparelho eletrônico ou lâmpadas. O nome “Relé” pode ser facilmente alterado para o nome do dispositivo conectado, facilitando a identificação.

Os sensores adicionados (de temperatura e de umidade), ficam sempre monitorando o ambiente, podendo fazer leituras a qualquer momento, bastando atualizar a página para a geração de uma leitura do momento atual, sendo assim mais precisa.

O site, utilizado como sistema de controle, pode ser conectado por vários dispositivos simultaneamente, com isso, várias pessoas podem comandar a parte da residência que necessitarem, sem interferir nas demais partes. No caso da residência de teste, foi possível as pessoas comandarem cômodos diferentes da casa sem que atrapalhassem umas às outras.

A aparência final da página de controle residencial ficou como a demonstrada abaixo, na Figura 11 mostra a página de controle residencial com os relés desligados (a) e ligados (b), visualizada pelo computador. Na Figura 12 mostra a página de controle residencial com os relés desligados (a) e ligados (b), visualizada pelo celular.

Figura 11 - Página de controle residencial com os relés desligados (a) e ligados (b), visualizada pelo computador





Fonte: O autor (2018).

Figura 12 - Página de controle residencial com os relés desligados (a) e ligados (b), visualizada pelo celular.



Fonte: O autor (2018).

## 5 CONCLUSÃO

Conclui-se que este trabalho correspondeu ao tema central proposto, que foi como se pode criar um sistema de automação residencial de baixo custo, se comparado com o sistema similar encontrado no mercado, utilizando sistemas livres, levando assim ao aumento na facilidade da vida das pessoas e ajudando o meio ambiente de certa forma. Por ser um sistema livre, isso ajudou com que o projeto ficasse mais personalizável e confiável, além da economia final do projeto.

Também pode-se concluir que se cumpre com o papel de ser de simples implementação, podendo também ser facilmente instalado em residências já existentes. O mesmo também pode ser facilmente utilizado como base para adicionar novas funcionalidades futuras ou realizar modificações nas existentes.

Por meio deste, é possível notar que a tecnologia está cada vez mais presente na vida das pessoas, podendo dar mais conforto e praticidade até nas coisas mais corriqueiras do cotidiano residencial. Por ele foi projetado um sistema bem mais acessível à população, além de ser um sistema bastante simples e de fácil acesso aos componentes utilizados.

## REFERÊNCIAS BIBLIOGRÁFICAS

ANGNES, Derli Luís. **Introdução ao Microcontrolador: BASIC Step**. 2013. Disponível em: <<http://www.radiopoint.com.br/MICROCONTROLADORES/MicrocontroladoresBASICStep.pdf>>. Acesso em: 5 jun. 2018.

BANZI, Massimo et al. **The Arduino Playground**. 2018. Disponível em: <<https://playground.arduino.cc/Portugues/HomePage>>. Acesso em: 5 jun. 2018.

BENFICA, Alex; MATTEDE, Henrique. **O que é relé? Como funciona um relé?** Disponível em: <<https://www.mundodaeletrica.com.br/o-que-e-rele-como-funciona-um-rele/>>. Acesso em: 9 set. 2018.

BOLZANI, Caio. **Residências Inteligentes**. São Paulo: Livraria da Física, 2004.

GOTARDO, Reginaldo Aparecido. **Linguagem de Programação**. Rio de Janeiro: Sesus, 2015. 200 p.

MCROBERTS, Michael. **Arduino Básico**. São Paulo: Novatec, 2011. 453 p.

MOTA, Allan. **O que é Arduino e como funciona?** 2017. Disponível em: <<https://portal.vidadesilicio.com.br/o-que-e-arduino-e-como-funciona/>>. Acesso em: 14 maio 2018.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico**. 2. ed. Novo Hamburgo, Rs: Feevale, 2013. 276 p.

QUEIROZ, Tiago A. F. et al. **Sistema Embarcado Linux para Análise de Sensores de Temperatura DHT11 e LM35**. 2015. Disponível em: <<http://www.eripi.com.br/2016/anais2016/2472/153319.pdf>>. Acesso em: 14 jun. 2018.

SANTOS, J. V. T. **A construção da viagem inversa**. Cadernos de Sociologia, ensaio sobre a investigação nas ciências sociais, Porto Alegre, v. 3, n. 3, p. 55-88, jan. /Jul. 1991.

SILVEIRA, Cristiano Bertulucci. **O que é PWM e Para que Serve?** Disponível em: <<https://www.citisystems.com.br/pwm/>>. Acesso em: 6 jun. 2018.

THOMSEN, Adilson. **Monitorando Temperatura com DHT11 e Raspberry Pi**. 2015. Disponível em: <<https://www.filipeflop.com/blog/temperatura-umidade-dht11-com-raspberry-pi/>>. Acesso em: 5 set. 2018.

TOTALCROSS. **As 7 principais linguagens de programação usadas em desenvolvimento mobile**. 2015. Disponível em: <<https://blog.totalcross.com/pt/2015/07/31/as-7-principais-linguagens-de-programacao-usadas-em-desenvolvimento-mobile/>>. Acesso em: 6 jun. 2018.

WIKIPÉDIA. **Universal Serial Bus**. Disponível em: <[https://pt.wikipedia.org/wiki/Universal\\_Serial\\_Bus](https://pt.wikipedia.org/wiki/Universal_Serial_Bus)>. Acesso em: 27 jun. 2018.

## APÊNDICE A – CÓDIGO DE PROGRAMAÇÃO DA PÁGINA DE CONTROLE RESIDENCIAL PRESENTE NO NODEMCU

```

client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head>");
client.println("<title>TCC Rhenann</title>");
client.println("<meta http-equiv='Content-Type' content='text/html; charset=utf-8'>");
client.println("<!-- Latest compiled and minified CSS -->");
client.println("<link rel='stylesheet'
ref='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css'>");
client.println("</head>");
client.println("<body>");
client.println("<div class='container'>");
client.println("<h1><center>Trabalho de Conclusão de Curso</center></h1>");
client.println("<hr />");
client.println("<table class='table table-striped'>");
client.println("<thead>");
client.println("<tr>");
client.println("<th>Sensor</th>");
client.println("<th>Medida</th>");
client.println("<th>Valor</th>");
client.println("<th></th>");
client.println("</tr>");
client.println("</thead>");
client.println("<tbody>");
client.println("<tr>");
client.println("<td scope='row'> DHT11 </td>");
client.println("<td> Temperatura</td> ");
client.println("<td>");
client.println(f_data);
client.println("&deg;C </td>");

```









