



Sistemas Embarcados 2

Trabalho Final 1

Sistema IOT com WebServer e ESP8266

Professor: Éder Alves de Moura
Aluno: Lucas Gonçalves e Silva

11811EAU016

SUMÁRIO

1.	Introdução	3
2.	Objetivos.....	4
3.	Materiais e Métodos	4
4.	Resultados.....	5
4.1	Código.....	5
4.2	Servidor Web	8
5.	Conclusão	10

1. INTRODUÇÃO

Com o avanço da tecnologia e da comunicação o IOT (Internet das Coisas) vem sendo um assunto muito tratado ao redor de todo o mundo. O poder de se conectar a objetos do dia a dia e ter o controle de tudo isso na palma da mão, seja pelo celular ou pelo computador, é um dos fatores que potencializa esse crescimento.

Ao conectar objetos com diferentes recursos a uma rede, potencializa-se o surgimento de novas aplicações. Neste sentido, conectar esses objetos à Internet significa criar a Internet das Coisas. Na IoT, os objetos podem prover comunicação entre usuários, dispositivos. Com isto emerge uma nova gama de aplicações, tais como coleta de dados e visualização de forma remota, sensoriamento de ambientes de difícil acesso, entre outras.

A Figura 1 mostra um escopo do que o IOT é composto.

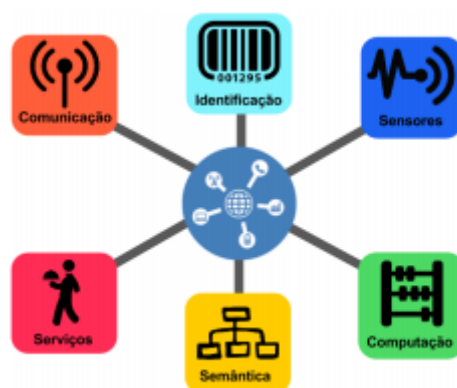


Figura 1: base do IOT.

A identificação é um dos pontos mais importantes, identificar os objetos unicamente é necessário para conectá-los à Internet. Os sensores coletam informações sobre o contexto em que os objetos se encontram e, em seguida, encaminham esses dados ao controlador. A computação inclui a unidade de processamento como, por exemplo, microcontroladores, processadores e FPGAs.



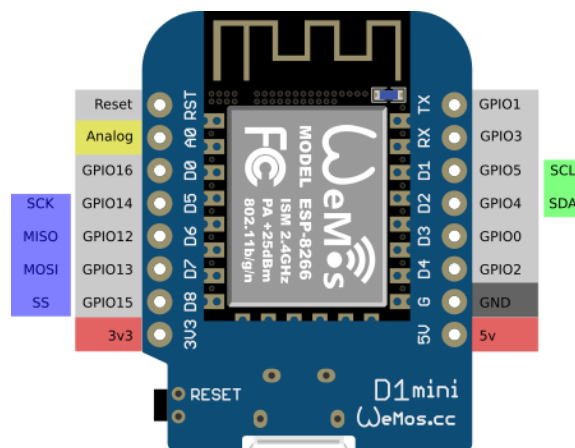
Figura 2: fluxo de dados em um IOT.

2. OBJETIVOS

O objetivo desse trabalho é desenvolver uma aplicação que funcione considerando os princípios da Internet das Coisas (IOT). De acordo com a proposta e o objetivo, foi desenvolvido uma aplicação IOT com microcontrolador capaz de ligar e desligar duas luzes, denominadas Luz Quarto e Luminária, através de um Servidor Web local.

3. MATERIAIS E MÉTODOS

Para realizar esse trabalho foi utilizado um microcontrolador que possui uma placa de Wi-Fi com o objetivo de conectar diretamente ao servidor local sem ter um cabo conectado ao computador para leitura serial. O microcontrolador que foi usado é chamado de Wemos D1 Mini, essa placa é muito indicada para realizar projetos IOT pelo seu tamanho e por já vir com um chip Wi-Fi, o ESP8266.



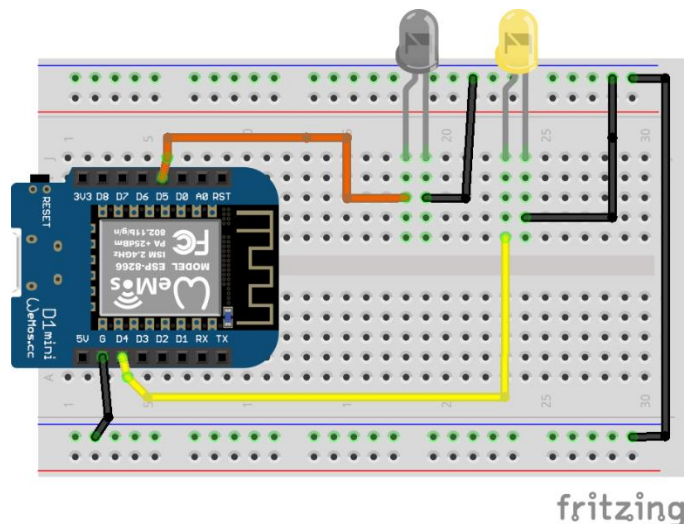


Figura 4: esquemático do sistema.

O Wemos D1 Mini é programado da mesma forma que se programa um Arduino, então o código inserido na placa também foi escrito em C++, com umas linhas que precisam de um conhecimento básico em HTML para configurar o visual da página do servidor. Todo o código foi feito na IDE do próprio Arduino.

A biblioteca usada é a “ESP8266WiFi” que faz a interação e conexão com o Wi-Fi da placa ESP8266 instalada no microcontrolador.

4. RESULTADOS

4.1 Código

Primeiramente o código vai ser exposto e explicado para que todos possam entender como o sistema funciona.

O primeiro passo é importar a biblioteca que vai ser utilizada para auxiliar no desenvolvimento do código.

```
// Biblioteca
#include <ESP8266WiFi.h>
```

Após a importação da biblioteca, foram definidas as constantes que vão ser manipuladas no decorrer do código, por exemplo a senha da rede Wi-Fi e os pinos do microcontrolador.

```
// Configurando as credenciais do Wi-Fi
const char* ssid = "nome_do_seu_wifi"; //nome da rede
const char* password = "senha"; //senha
```

```
//Configura o servidor web para a porta 80
WiFiServer server(80);
```

```
String header;

// Variaveis auxiliares para armazenar o estado dos leds
String output5State = "off";
String output4State = "off";

// Portas GPIO que vão ser utilizadas
const int output5 = D5; //LUZ do quarto
const int output4 = D4; //Luminaria

unsigned long currentTime = millis();
unsigned long previousTime = 0;
const long timeoutTime = 2000;
```

é necessário configurar os pinos e a velocidade de comunicação com porta serial, tudo isso é feito na função “setup”

```
//A função setup é rodada uma vez só no código
void setup() {
  //Configurando a comunicação serial
  Serial.begin(115200);
  // Definindo as portas GPIO como saidas
  pinMode(output5, OUTPUT);
  pinMode(output4, OUTPUT);
  // As portas vão sempre iniciar no estado LOW: desligado
  digitalWrite(output5, LOW);
  digitalWrite(output4, LOW);

  //Conexão ao Wi-Fi, se a conexão for bem sucedida ele imprime o IP do servidor
  Serial.print("Conectado ao");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Print do IP do servidor local
  Serial.println("");
  Serial.println("WiFi conectado.");
  Serial.println("Endereço de IP: ");
  Serial.println(WiFi.localIP());
  server.begin();
}
```

A função que roda o tempo todo e executa as funções enquanto o dispositivo está ligado da forma correta é chamada de “loop”.

```
//Loop é a função que vai ficar rodando enquanto o dispositivo estiver ligado
void loop(){
  WiFiClient client = server.available(); // Leitura do Cliente que entra no servidor
  // Se uma nova conexão ocorrer,
```

```

if (client) {
  Serial.println("Novo Usuario");
  String currentLine = "";
  currentTime = millis();
  previousTime = currentTime;
  while (client.connected() && currentTime - previousTime <= timeoutTime) {
    currentTime = millis();
    if (client.available()) {
      char c = client.read();
      Serial.write(c);
      header += c;
      if (c == '\n') {
        if (currentLine.length() == 0) {
          client.println("HTTP/1.1 200 OK");
          client.println("Content-type:text/html");
          client.println("Connection: close");
          client.println();

          // Verifica qual botão é pressionado e indica o estado atual da porta
          if (header.indexOf("GET /5/on") >= 0) {
            Serial.println("GPIO 5 on");
            output5State = "on";
            digitalWrite(output5, HIGH);
          } else if (header.indexOf("GET /5/off") >= 0) {
            Serial.println("GPIO 5 off");
            output5State = "off";
            digitalWrite(output5, LOW);
          } else if (header.indexOf("GET /4/on") >= 0) {
            Serial.println("GPIO 4 on");
            output4State = "on";
            digitalWrite(output4, HIGH);
          } else if (header.indexOf("GET /4/off") >= 0) {
            Serial.println("GPIO 4 off");
            output4State = "off";
            digitalWrite(output4, LOW);
          }

          // Gerando a pagina do servidor web
          client.println("<!DOCTYPE html><html>"); //mostra que esta sendo enviado um html
          client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">"); //tornando a pagina responsiva
          client.println("<link rel=\"icon\" href=\"data:;\">");
          // Estilo da pagina, configurando o tamanho dos botões e etc
          client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px
auto; text-align: center; }");
          client.println(".button { background-color: #a1ba23; border: none; color: white;
padding: 16px 40px;");
          client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer; }");
          client.println(".button2 { background-color: #c22e1d; }</style></head>");
          // Titulo da pagina
          client.println("<body><h1>IOT COM WEB SERVER E ESP8266</h1>");
          //Mostra o estado da porta D5
          client.println("<p>LUZ QUARTO - Estado " + output5State + "</p>");
          if (output5State=="off") {

```

```

        client.println("<p><a href=\"/5/on\"><button
class=\"button\">LIGAR</button></a></p>");
    } else {
        client.println("<p><a href=\"/5/off\"><button class=\"button
button2\">DESLIGAR</button></a></p>");
    }

    //Mostra o estado da porta D4
    client.println("<p>LUMINARIA - Estado " + output4State + "</p>");
    if (output4State=="off") {
        client.println("<p><a href=\"/4/on\"><button
class=\"button\">LIGAR</button></a></p>");
    } else {
        client.println("<p><a href=\"/4/off\"><button class=\"button
button2\">DESLIGAR</button></a></p>");
    }
    client.println("</body></html>");

    client.println();
    break;
} else {
    currentLine = "";
}
} else if (c != '\r') {
    currentLine += c;
}
}
}
// Limpando a variavel
header = "";
// Encerrando conexão
client.stop();
Serial.println("Conexão encerrada");
Serial.println("");
}
}

```

4.2 Servidor Web

Após fazer o upload do código para o microcontrolador e abrir o monitor serial na IDE do Arduino é possível ver o estado da conexão e o IP do servidor para realizar a conexão no navegador. A Figura 5 mostra como deve estar assim que você fizer o upload do código para a placa.

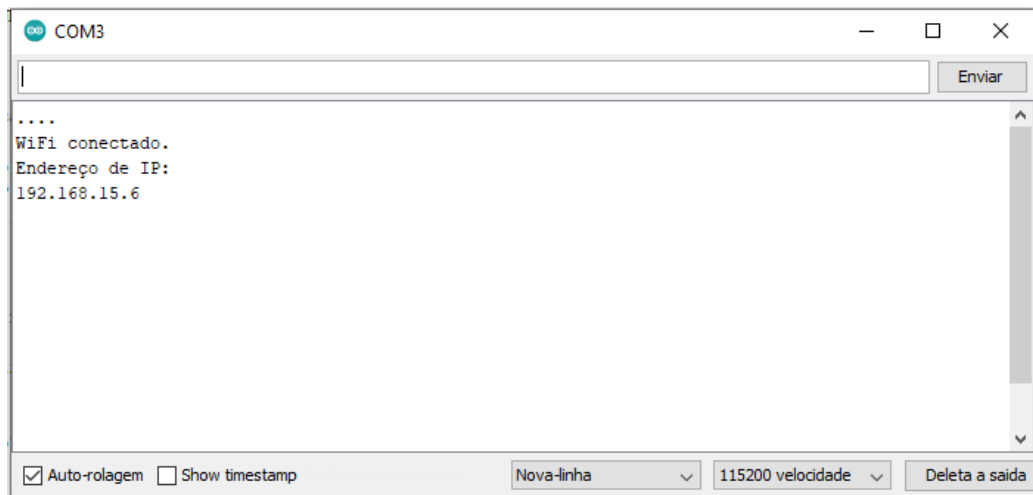


Figura 5: monitor serial da IDE.

Após obter o endereço de IP basta colocar esse endereço no navegador e você vai se deparar com a página da Figura 6.

IOT COM WEB SERVER E ESP8266

LUZ QUARTO - Estado off

LIGAR

LUMINARIA - Estado off

LIGAR

Figura 6: página do Servidor Web criado.

Nesse servidor local é possível ligar e desligar as luzes, que estão sendo representadas com os LEDs, além de verificar se elas estão ligadas ou desligadas.

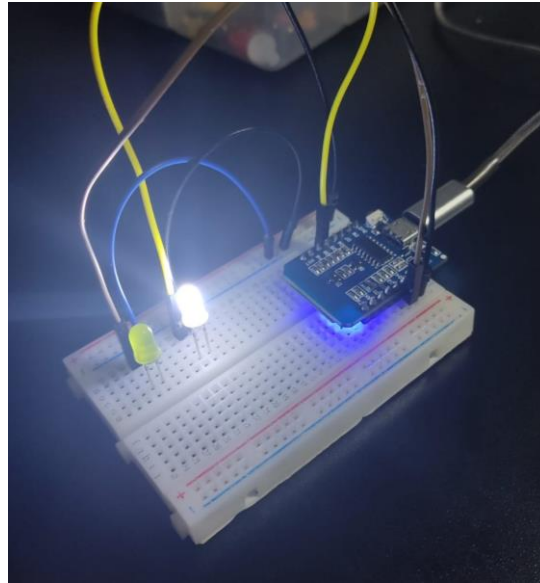


Figura 7: resultado do protótipo.

O resultado do sistema, Figura 7, pode ser visto em funcionamento acessando o link abaixo:

[Vídeo explicativo](#)

Além desse link, todo o código e material está disponibilizado no meu portfólio do GitHub que pode ser acessado por esse outro link:

[Link Diretório](#)

5. CONCLUSÃO

Ao fazer esse trabalho foi possível perceber o quanto sistemas IOT estão cada vez mais destacados na rede, a gama de projetos e artigos que contém esse assunto prova que o gerenciamento de informações e o poder de tomar decisões de forma remota só fortalece o crescimento do IOT. O objetivo principal, após finalizar esse projeto, é tentar expandir esse sistema para responder com comando de voz e adicionar sensores que possibilitam o gerenciamento de temperatura e umidade do local.