

# ***Arduino com Shield Ethernet – WebServer JSON***

## **Sumário**

Exercício 1: Configurando o WebServer Básico no Arduino.....	3
Objetivo: Configurar um WebServer simples no Arduino utilizando uma Shield Ethernet.....	3
Código de exemplo.....	3
Exercício 2: Integrando Sensor DHT ao WebServer.....	4
Objetivo: Incorporar um sensor de temperatura e umidade (DHT) ao WebServer, enviando dados em formato JSON.....	4
Código de Exemplo:.....	4
Exercício 3: Controle Remoto de LEDs com Node-RED.....	5
Objetivo: Criar um controle remoto no Node-RED para acender ou apagar LEDs conectados ao Arduino.....	5
Código de Exemplo:.....	5
Exercício 4: Gráficos de Temperatura e Umidade no Node-RED.....	6
Objetivo: Utilizar o Node-RED para criar gráficos dinâmicos das leituras do sensor DHT.....	6
Exercício 5: Adicionando Alarmes e Notificações.....	7
Objetivo: Configurar o Node-RED para enviar notificações quando a temperatura atingir níveis específicos.....	7
Questionário sobre os Exercícios Propostos.....	8
Questão 1: Configurando o WebServer Básico no Arduino.....	8
Questão 2: Integrando Sensor DHT ao WebServer.....	8
Questão 3: Controle Remoto de LEDs com Node-RED.....	8
Questão 4: Gráficos de Temperatura e Umidade no Node-RED.....	8
Questão 5: Adicionando Alarmes e Notificações.....	8
Questão 6: Endereços MAC:.....	8
Questão 7: Node-RED e Arduino: Integração.....	9

Questão 8: WebServer e Navegador.....	9
Questão 9: Sensor DHT.....	9
Questão 10: Controle Remoto de LEDs.....	9

## Exercício 1: Configurando o WebServer Básico no Arduino Objetivo:

Configurar um WebServer simples no Arduino utilizando uma Shield Ethernet.

Instruções:

- Conecte a Shield Ethernet ao Arduino.
- Escreva um código para configurar um WebServer que exiba uma mensagem básica no navegador quando acessado.
- Teste o WebServer acessando o endereço IP do Arduino a partir de um navegador.

### Código de exemplo

```
#include <SPI.h>
#include <Ethernet.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // Endereço MAC

EthernetServer server(80);

void setup() {
  // Inicia a comunicação Ethernet e o servidor na porta 80
  Ethernet.begin(mac);
  server.begin();
  Serial.begin(9600);
}

void loop() {
  // Verifica se há clientes conectados
  EthernetClient client = server.available();
  if (client) {
    Serial.println("Novo cliente conectado!");

    // Envia um cabeçalho HTTP padrão
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: application/json");
    client.println();

    // Cria um JSON simples como resposta do servidor
    String jsonData = "{\"mensagem\":\"Bem-vindo ao WebServer Arduino!\"}";

    // Envia o JSON para o cliente
    client.print(jsonData);

    // Encerra a conexão com o cliente
    client.stop();
    Serial.println("Cliente desconectado");
  }
}
```

## Exercício 2: Integrando Sensor DHT ao WebServer

Objetivo: Incorporar um sensor de temperatura e umidade (DHT) ao WebServer, enviando

dados em formato JSON.

Instruções:

- Conecte o sensor DHT ao Arduino.
- Modifique o código do WebServer para incluir leituras do sensor e envie os dados em formato JSON como resposta.
- Teste o WebServer acessando o endereço IP do Arduino e observe os dados do sensor no formato JSON no navegador.

Código de Exemplo:

```
#include <SPI.h>
#include <Ethernet.h>
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT11

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // Endereço MAC

DHT dht(DHTPIN, DHTTYPE);
EthernetServer server(80);

void setup() {
  Ethernet.begin(mac);
  server.begin();
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  EthernetClient client = server.available();
  if (client) {
    Serial.println("Novo cliente conectado!");

    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: application/json");
    client.println();

    float temp = dht.readTemperature();
    float hum = dht.readHumidity();

    String jsonData = "{\"temperatura\":\"" + String(temp) + "\",\"umidade\":\"" +
    String(hum) + "\"}";
    client.print(jsonData);

    client.stop();
    Serial.println("Cliente desconectado");
  }
}
```

## Exercício 3: Controle Remoto de LEDs com Node-RED

Objetivo: Criar um controle remoto no Node-RED para acender ou apagar LEDs conectados ao Arduino.

Instruções:

- Configure um fluxo no Node-RED para se comunicar com o Arduino WebServer.
- Adicione botões no dashboard do Node-RED para controlar LEDs remotamente.
- Teste o controle remoto acessando o dashboard e observando as mudanças nos LEDs.

### Código de Exemplo:

```
// Código do WebServer com LEDs do Exercício 3
// (Inclua as partes relevantes do código do Exercício 1)

void loop() {
  EthernetClient client = server.available();
  if (client) {
    Serial.println("Novo cliente conectado!");

    // ... (Código anterior)

    // Adicione lógica para controlar LEDs com base nas requisições recebidas
    if (client.available()) {
      char c = client.read();
      if (c == 'A') {
        // Ligar LED
        digitalWrite(LED_PIN, HIGH);
      } else if (c == 'B') {
        // Desligar LED
        digitalWrite(LED_PIN, LOW);
      }
    }

    // ... (Código anterior)
  }
}
```

## Exercício 4: Gráficos de Temperatura e Umidade no Node-RED

**Objetivo:** Utilizar o Node-RED para criar gráficos dinâmicos das leituras do sensor DHT.

**Instruções:**

- Modifique o fluxo no Node-RED para incluir gráficos que mostrem a variação de temperatura e umidade ao longo do tempo.
- Ajuste a frequência de atualização dos gráficos para exibir dados mais recentes.

## Exercício 5: Adicionando Alarmes e Notificações

**Objetivo:** Configurar o Node-RED para enviar notificações quando a temperatura atingir níveis específicos.

**Instruções:**

- Acrescente lógica ao fluxo no Node-RED para verificar as leituras do sensor DHT. • Se a temperatura atingir um limite predefinido, envie uma notificação via Node-RED Dashboard.
- Teste o funcionamento, observando como as notificações são acionadas em diferentes condições.

Estes códigos de exemplo e exercícios proporcionarão uma experiência prática e progressiva no desenvolvimento de

projetos Arduino com WebServer, Node-RED e sensores DHT.