

Código Avaliação Arquitetura de Computadores

Código nó 1

```
#include <dht.h> // Biblioteca para o sensor DHT

dht DHT; // Declaração do sensor DHT

int Tx = 10; // Pino de transmissão serial
int Rx = 11; // Pino de recepção serial

void setup() {
  Serial.begin(9600); // Inicialização da comunicação serial
  pinMode(Tx, OUTPUT); // Configuração do pino de transmissão como saída
  pinMode(Rx, INPUT); // Configuração do pino de recepção como entrada
}

void loop() {
  int t = DHT.temperature; // Variável para armazenar a temperatura
  int u = DHT.humidity; // Variável para armazenar a umidade
  int l = analogRead(A0); // Variável para armazenar a luminosidade
  (utilizando o pino A0)

  Serial.print("t:"); // Envia o identificador "t:" para o Nó 2
  Serial.println(t); // Envia a temperatura para o Nó 2
  Serial.print("u:"); // Envia o identificador "u:" para o Nó 2
  Serial.println(u); // Envia a umidade para o Nó 2
  Serial.print("l:"); // Envia o identificador "l:" para o Nó 2
  Serial.println(l); // Envia a luminosidade para o Nó 2

  delay(10000); // Aguarda 10 segundos antes de coletar novos dados
}
```

Esse é o código para o Nó 1. Ele utiliza a biblioteca DHT para coletar dados de temperatura e umidade e a função `analogRead` para coletar a luminosidade. Em seguida, ele envia esses dados para o Nó 2 através da comunicação serial, usando identificadores "t:", "u:" e "l:" para indicar qual dado está sendo enviado. Ele aguarda 10 segundos antes de coletar novos dados.

Código Nó 2

```
int t, u, l; // Variáveis para armazenar os dados enviados pelo Nó 1
bool tVermelho, uVermelho, lVermelho; // Variáveis para armazenar se os
LEDs vermelhos estão ligados
bool tAmarelo, uAmarelo, lAmarelo; // Variáveis para armazenar se os LEDs
amarelos estão ligados
bool tVerde, uVerde, lVerde; // Variáveis para armazenar se os LEDs verdes
estão ligados
int motor1 = 5, motor2 = 6; // Pinos dos motores

void setup() {
    Serial.begin(9600); // Inicialização da comunicação serial
    pinMode(motor1, OUTPUT); // Configuração do primeiro pino de motor como
saída
    pinMode(motor2, OUTPUT); // Configuração do segundo pino de motor como
saída
}

void loop() {
    if (Serial.available()) { // Verifica se há dados disponíveis na serial
        String data = Serial.readString(); // Armazena os dados recebidos em
uma string
        int separator = data.indexOf(":"); // Encontra o índice do separador
":"
        String id = data.substring(0, separator); // Armazena o identificador
(t, u ou l)
        int value = data.substring(separator + 1).toInt(); // Armazena o valor
(temperatura, umidade ou luminosidade)

        if (id == "t") { // Se o identificador for "t", atualiza a temperatura
            t = value;
            if (t > 30) {
                digitalWrite(9, HIGH); // Liga LED vermelho
                tVermelho = true;
            } else if (t > 20) {
```

```

        digitalWrite(9, HIGH); // Liga LED amarelo
        tAmarelo = true;
    } else {
        digitalWrite(9, HIGH); // Liga LED verde
        tVerde = true;
    }
} else if (id == "u") { // Se o identificador for "u", atualiza a
umidade
    u = value;
    if (u > 60) {
        digitalWrite(10, HIGH); // Liga LED vermelho
        uVermelho = true;
    } else if (u > 30) {
        digitalWrite(10, HIGH); // Liga LED amarelo
        uAmarelo = true;
    } else {
        digitalWrite(10, HIGH); // Liga LED verde
        uVerde = true;
    }
} else if (id == "l") { // Se o identificador for "l", atualiza a
luminosidade
    l = value;
    if (l > 80) {
        digitalWrite(11, HIGH); // Liga LED vermelho
        lVermelho = true;
    } else if (l > 30) {
        digitalWrite(11, HIGH); // Liga LED amarelo

```

```

void loop() {
  if (Serial.available()) { // Verifica se há dados disponíveis na serial
    String data = Serial.readString(); // Armazena os dados recebidos em
uma string
    int separator = data.indexOf(":"); // Encontra o índice do separador
":"
    String id = data.substring(0, separator); // Armazena o identificador
(t, u ou l)
    int value = data.substring(separator + 1).toInt(); // Armazena o valor
(temperatura, umidade ou luminosidade)

    if (id == "t") { // Se o identificador for "t", atualiza a temperatura
      t = value;
      if (t > 30) {
        digitalWrite(9, HIGH); // Liga LED vermelho
        tVermelho = true;
      } else if (t > 20) {
        digitalWrite(9, HIGH); // Liga LED amarelo
        tAmarelo = true;
      } else {
        digitalWrite(9, HIGH); // Liga LED verde
        tVerde = true;
      }
    } else if (id == "u") { // Se o identificador for "u", atualiza a
umidade
      u = value;
      if (u > 60) {
        digitalWrite(10, HIGH); // Liga LED vermelho
        uVermelho = true;
      } else if (u > 30) {
        digitalWrite(10, HIGH); // Liga LED amarelo
        uAmarelo = true;
      } else {
        digitalWrite(10, HIGH); // Liga LED verde
        uVerde = true;
      }
    }
  }
}

```

```

    }
    } else if (id == "l") { // Se o identificador for "l", atualiza a
luminosidade
        l = value;
        if (l > 80) {
            digitalWrite(11, HIGH); // Liga LED vermelho
            lVermelho = true;
        } else if (l > 30) {
            digitalWrite(11, HIGH); // Liga LED amarelo
            lAmarelo = true;
        } else {
            digitalWrite(11, HIGH); // Liga LED verde
            lVerde = true;
        }
    }
    }
    if (tVermelho && uVermelho && lVermelho) { // Se todos os LEDs
vermelhos estiverem ligados
        digitalWrite(motor1, HIGH); // Liga o primeiro motor
        digitalWrite(motor2, HIGH); // Liga o segundo motor
    } else if (tAmarelo && uAmarelo && lAmarelo) { // Se todos os LEDs
amarelos estiverem ligados
        digitalWrite(motor1, HIGH); // Liga o primeiro motor
    } else if (tVerde && uVerde && lVerde) { // Se todos os LEDs verdes
estiverem ligados
        digitalWrite(motor1, LOW); // Desliga o primeiro motor
        digitalWrite(motor2, LOW); // Desliga o segundo motor
    }
}
}
}

```