



**INSTITUTO FEDERAL**

Brasília  
Campus Brasília

**TECNOLOGIA EM SISTEMAS PARA INTERNET**

**Claudio Lucas de Oliveira Franco**

**Jean Pereira Ribeiro**

**Marcos Vinicius Lira de Oliveira**

**Rafael Souza Nunes**

**RELATÓRIO DE PRÁTICA INTEGRADA  
DE  
CIÊNCIA DE DADOS E INTERNET DAS COISAS**

**Brasília - DF**

**10/12/2022**

# Sumário

<b>1. Objetivos</b>	<b>3</b>
<b>2. Descrição do problema</b>	<b>4</b>
<b>3. Desenvolvimento</b>	<b>5</b>
3.1 Código implementado	6
<b>4. Considerações finais</b>	<b>9</b>
<b>Referências</b>	<b>10</b>

# 1. Objetivos

O objetivo principal deste relatório é descrever o processo de desenvolvimento da sprint 1 da Prática Integrada. O projeto escolhido pelo grupo foi o da “Caixa para controle de umidade de filamento para impressora 3D”. Para essa sprint, os objetivos foram: montar o hardware e configurá-lo para funcionar adequadamente.

## 2. Descrição do problema

Em relação ao projeto escolhido, o problema envolve a umidade no filamento que resulta em problemas de impressão e perdas do material. Para resolver esse problema, é necessário evitar ou remover a umidade no material para que se tenha uma impressão adequada (TECNOCUBO, 2022).

Enquanto aos problemas a serem resolvidos nessa etapa do projeto, os principais foram: conseguir os materiais necessários para a realização do projeto, fazer a montagem do hardware e o de configurar o código para o recebimento dos dados do sensor.

### 3. Desenvolvimento

Para a realização do projeto, seguimos o passo a passo mostrado no artigo “Sensores DHT11 e DHT22: Guia Básico dos Sensores de Umidade e Temperatura” (MURTA, 2019).

Os materiais utilizados foram:

- 1 Arduino Uno R3 + Cabo Usb para Arduino
- 1 Protoboard 400 Pontos
- 3 Jumpers – Macho/Macho de 20cm
- 1 Sensor de Umidade e Temperatura DHT11

Os materiais foram adquiridos pela internet. Após a entrega, montamos o hardware de acordo com o Diagrama Fritzing a seguir:

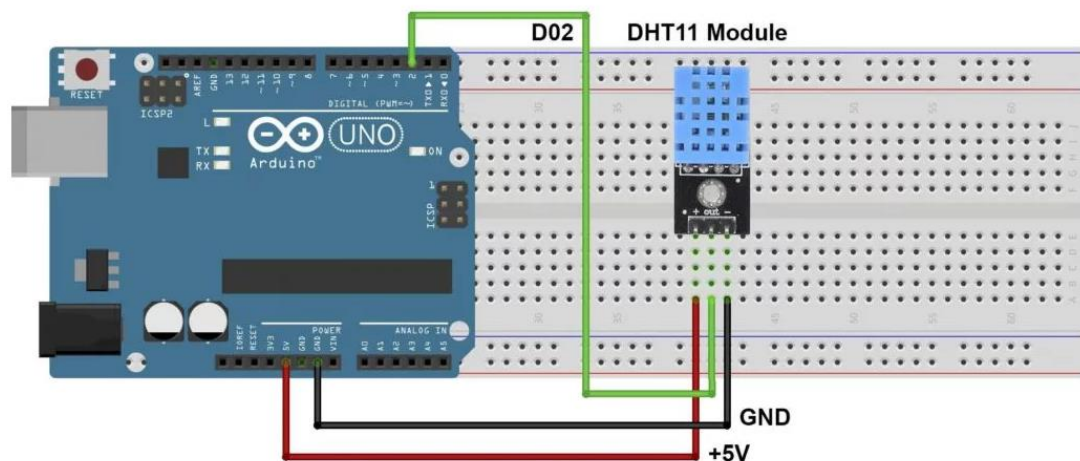
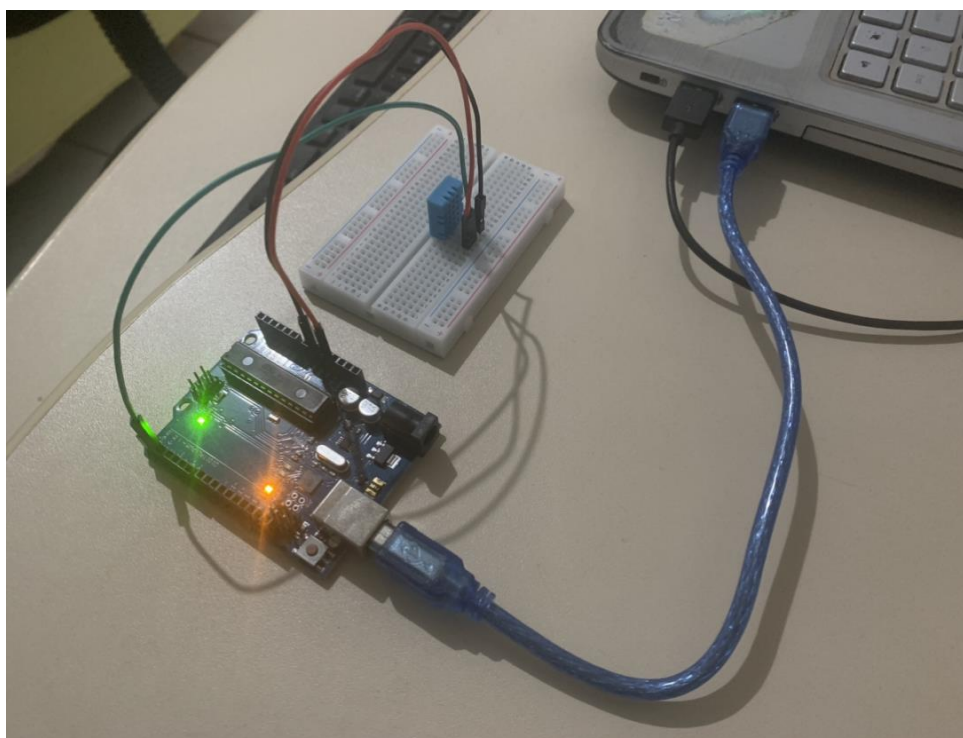
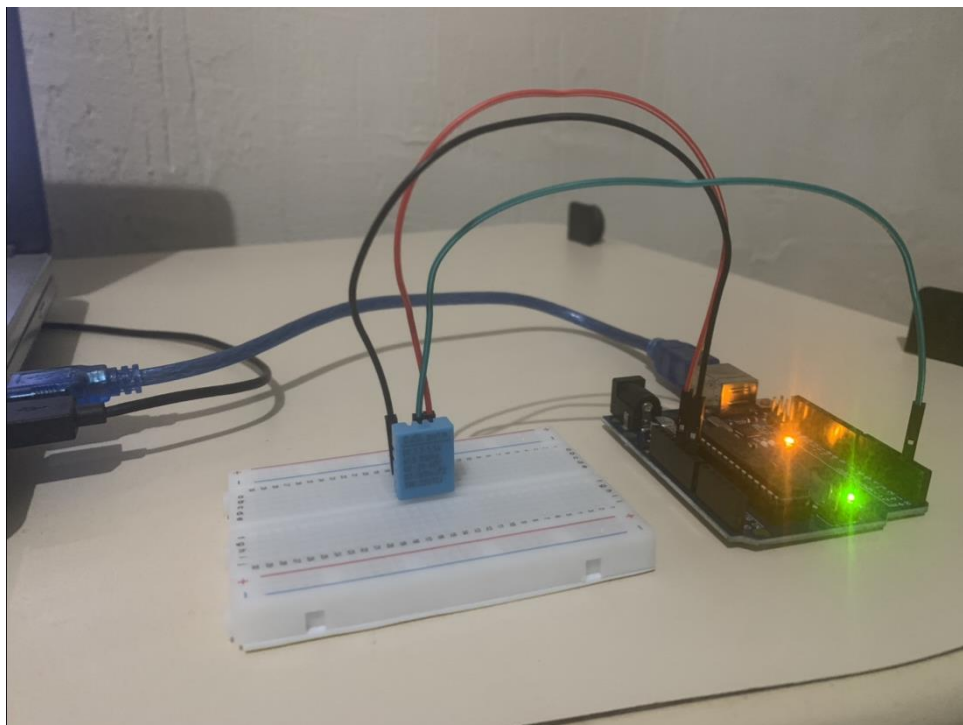


Diagrama Fritzing – Arduino e sensor DHT11

As imagens abaixo mostram o hardware montado:



### 3.1 Código implementado

O código implementado para o recebimento dos dados do sensor contendo os comentários explicando o funcionamento de cada parte será mostrado a seguir:



sketch\_dec08a\$

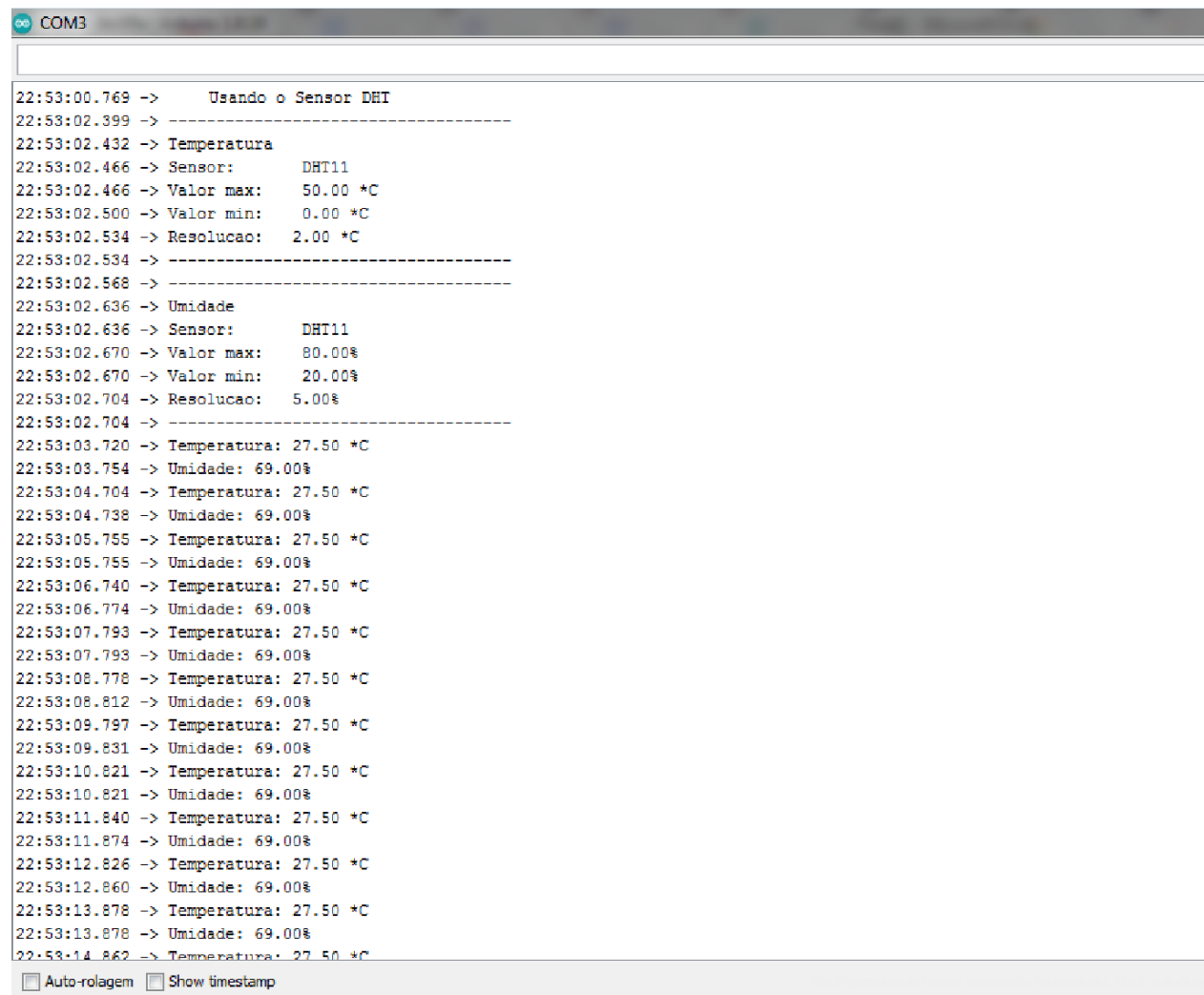
```

#include <Adafruit_Sensor.h>                                // Biblioteca DHT Sensor Adafruit
#include <DHT.h>
#include <DHT_U.h>
#define DHTTYPE      DHT11                                // Sensor DHT11
#define DHTPIN 2                                           // Pino do Arduino conectado no Sensor(Data)
DHT_Unified dht(DHTPIN, DHTTYPE);                          // configurando o Sensor DHT - pino e tipo
uint32_t delayMS;                                         // variável para atraso no tempo
void setup()
{
  Serial.begin(9600);                                       // monitor serial 9600 bps
  dht.begin();                                              // inicializa a função
  Serial.println("Usando o Sensor DHT");
  sensor_t sensor;

  dht.begin();                                              // inicializa a função
  Serial.println("Usando o Sensor DHT");
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);                    // imprime os detalhes do Sensor de Temperatura
  Serial.println("-----");
  Serial.println("Temperatura");
  Serial.print ("Sensor: "); Serial.println(sensor.name);
  Serial.print ("Valor max: "); Serial.print(sensor.max_value); Serial.println(" *C");
  Serial.print ("Valor min: "); Serial.print(sensor.min_value); Serial.println(" *C");
  Serial.print ("Resolucao: "); Serial.print(sensor.resolution); Serial.println(" *C");
  Serial.println("-----");
  dht.humidity().getSensor(&sensor);                        // imprime os detalhes do Sensor de Umidade
  Serial.println("-----");
  Serial.println("Umidade");
  Serial.print ("Sensor: "); Serial.println(sensor.name);
  Serial.print ("Valor max: "); Serial.print(sensor.max_value); Serial.println("%");
  Serial.print ("Valor min: "); Serial.print(sensor.min_value); Serial.println("%");
  Serial.print ("Resolucao: "); Serial.print(sensor.resolution); Serial.println("%");
  Serial.println("-----");
  delayMS = sensor.min_delay / 1000;                        // define o atraso entre as leituras
}
void loop()
{
  delay(delayMS);                                           // atraso entre as medições
  sensors_event_t event;                                     // inicializa o evento da Temperatura
  dht.temperature().getEvent(&event);                       // faz a leitura da Temperatura
  if (isnan(event.temperature))                             // se algum erro na leitura
  {
    Serial.println("Erro na leitura da Temperatura!");
  }
  else
  {
    Serial.print("Temperatura: ");                          // imprime a Temperatura
    Serial.print(event.temperature);
    Serial.println(" *C");
  }
  dht.humidity().getEvent(&event);                           // faz a leitura de umidade
  if (isnan(event.relative_humidity))                       // se algum erro na leitura
  {
    Serial.println("Erro na leitura da Umidade!");
  }
  else
  {
    Serial.print("Umidade: ");                              // imprime a Umidade
    Serial.print(event.relative_humidity);
    Serial.println("%");
  }
}

```

A imagem a seguir mostra a saída desse código e os dados que foram enviados pelo sensor:



```
COM3
22:53:00.769 -> Usando o Sensor DHT
22:53:02.399 -> -----
22:53:02.432 -> Temperatura
22:53:02.466 -> Sensor: DHT11
22:53:02.466 -> Valor max: 50.00 *C
22:53:02.500 -> Valor min: 0.00 *C
22:53:02.534 -> Resolucao: 2.00 *C
22:53:02.534 -> -----
22:53:02.568 -> -----
22:53:02.636 -> Umidade
22:53:02.636 -> Sensor: DHT11
22:53:02.670 -> Valor max: 80.00%
22:53:02.670 -> Valor min: 20.00%
22:53:02.704 -> Resolucao: 5.00%
22:53:02.704 -> -----
22:53:03.720 -> Temperatura: 27.50 *C
22:53:03.754 -> Umidade: 69.00%
22:53:04.704 -> Temperatura: 27.50 *C
22:53:04.738 -> Umidade: 69.00%
22:53:05.755 -> Temperatura: 27.50 *C
22:53:05.755 -> Umidade: 69.00%
22:53:06.740 -> Temperatura: 27.50 *C
22:53:06.774 -> Umidade: 69.00%
22:53:07.793 -> Temperatura: 27.50 *C
22:53:07.793 -> Umidade: 69.00%
22:53:08.778 -> Temperatura: 27.50 *C
22:53:08.812 -> Umidade: 69.00%
22:53:09.797 -> Temperatura: 27.50 *C
22:53:09.831 -> Umidade: 69.00%
22:53:10.821 -> Temperatura: 27.50 *C
22:53:10.821 -> Umidade: 69.00%
22:53:11.840 -> Temperatura: 27.50 *C
22:53:11.874 -> Umidade: 69.00%
22:53:12.826 -> Temperatura: 27.50 *C
22:53:12.860 -> Umidade: 69.00%
22:53:13.878 -> Temperatura: 27.50 *C
22:53:13.878 -> Umidade: 69.00%
22:53:14.862 -> Temperatura: 27.50 *C
☐ Auto-rolagem ☐ Show timestamp
```

O código pode ser encontrado no Github através do link:

<https://github.com/infocbra/pratica-integrada-cd-e-ic-2022-2-g2-cmrj>



## 4. Considerações finais

Para a realização dessa etapa, a principal dificuldade encontrada foi a de conseguir os materiais necessários para a realização do projeto. Fizemos um pedido pela internet e a entrega chegou poucos dias antes do fechamento da sprint, o que nos deu pouco tempo para a montagem do hardware e desenvolvimento do código.

Uma possibilidade de melhoria para as próximas etapas é a integração do Arduino com o Excel, algo que não conseguimos implementar nessa sprint, para facilitar a visualização, organização e análise dos dados obtidos.

Apesar das dificuldades encontradas, ficamos satisfeitos por atingir o objetivo dessa sprint de montar o hardware e configurá-lo para funcionar adequadamente.

## Referências

MURTA, José Gustavo Abreu. Sensores DHT11 e DHT22: Guia Básico dos Sensores de Umidade e Temperatura. **Blog Eletrogate**, 2019. Disponível em: <https://blog.eletrogate.com/sensores-dht11-dht22/>. Acesso em: 10 dez.2022.

Umidade no Filamento. **Tecnocubo**, c2022. Disponível em: <https://www.tecnocubo.com.br/pagina/umidade-no-filamento.html>. Acesso em: 10 dez.2022.