

## CONFIGURANDO A API DE ENVIO DE D SENSORES - PARTE 1 SEM CONEXÃO BANCO – Versão SPTECH 2022-1

Contribuições: Carrera & Marise

Primeiramente, após ter acessado o moodle nas matérias de Arquitetura Computacional, faça o download do arquivo .zip (**arduino\_api\_v3.7z**) e do script **"FIVE\_SENSORS\_V04.ino"** que está listado e identificado na matéria.

11/04/2022 - PREPARATÓRIO SPRINT 2 - FIVE SENSORS AND GRAPHICS

API FIVE SENSORS WITH GRAPHICS

Como iniciar um serviço de API com Node

Baixe na máquina local, em uma pasta o diretório, o arquivo zipado da API em NodDescompacte dentro do diretório

Abra o cmd no local do diretório, use comandos de cd (choose directory) para subir e descer os níveis hierárquicos do sistema de arquivos

Com o CMD aberto po demos executar -

**npm install** (instala os pacotes necessários)

**npm start** (inicia o servidor node)

**Ctrl + C** (interrompe o processo)

-----Com o servidor rodando podemos executar-----

**netstat -ano** (Comando **netstat** exibe estatísticas de protocolo e conexões de redes TCP/IP atuais. Parâmetro **-a**: Exibe todas as conexões e portas de escuta. Parâmetro **-n**: Exibe endereços e número de portas no formato numérico. Parâmetro **-o**: Exibe a ID do processo proprietário associado à cada conexão)

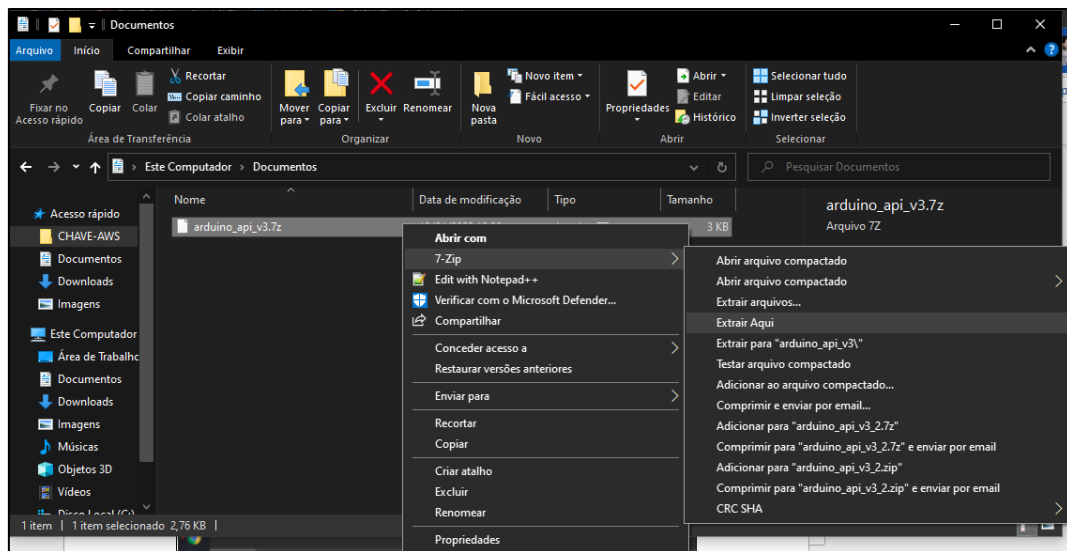
**taskkill /PID <nº ID> /F** (**taskkill** finalizar tarefas por identificação de processo (PID) ou nome de imagem. Parâmetro **/PID**: especifica o PID do processo a ser finalizado. Parâmetro **/F**: Especifica a finalização forçada do(s) processo(s))

07/03/2022 AULA PRÁTICA - Arquitetura Computacional com Arduino

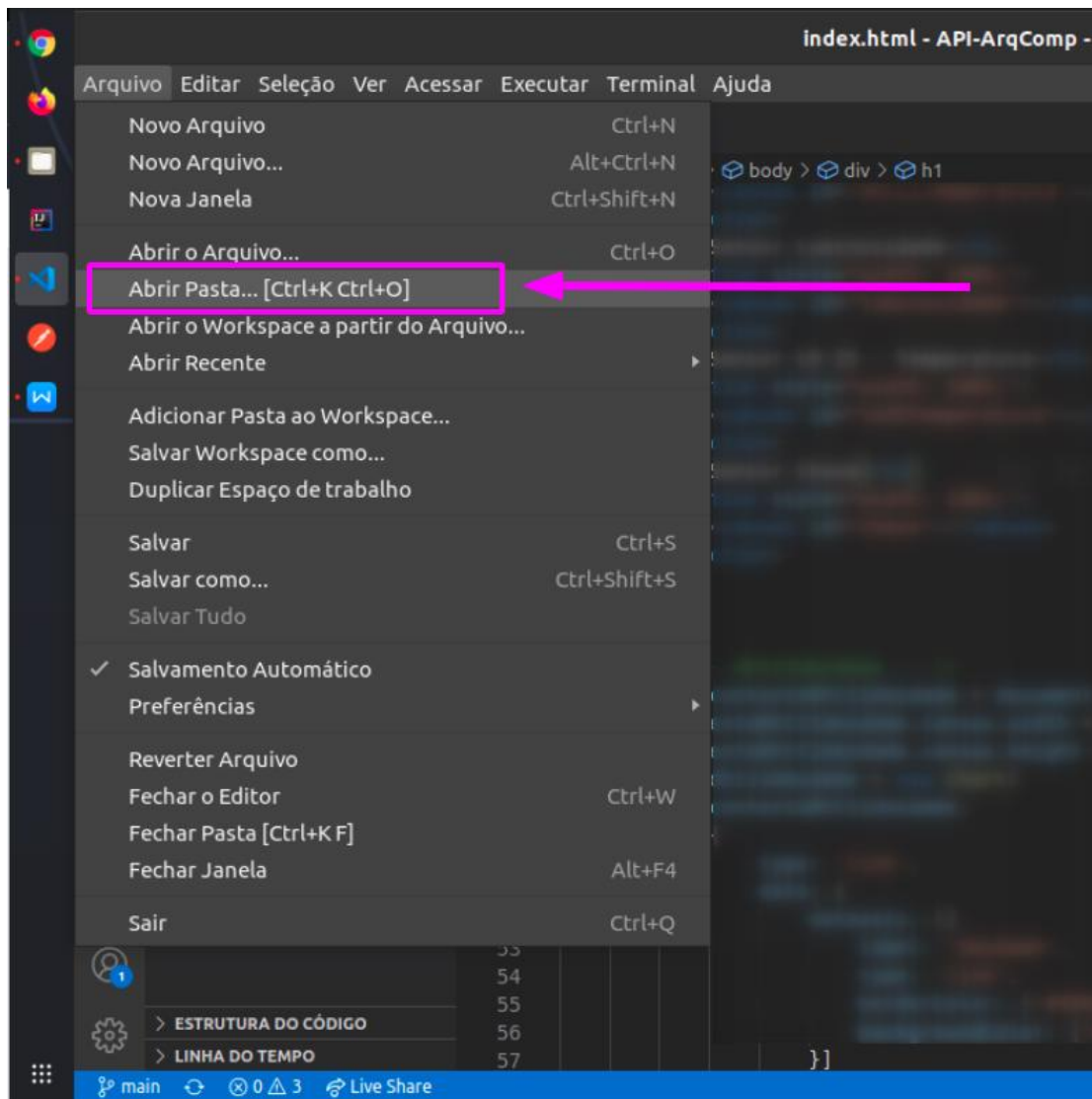
Código arduino Five Sensores v04

Código Ino

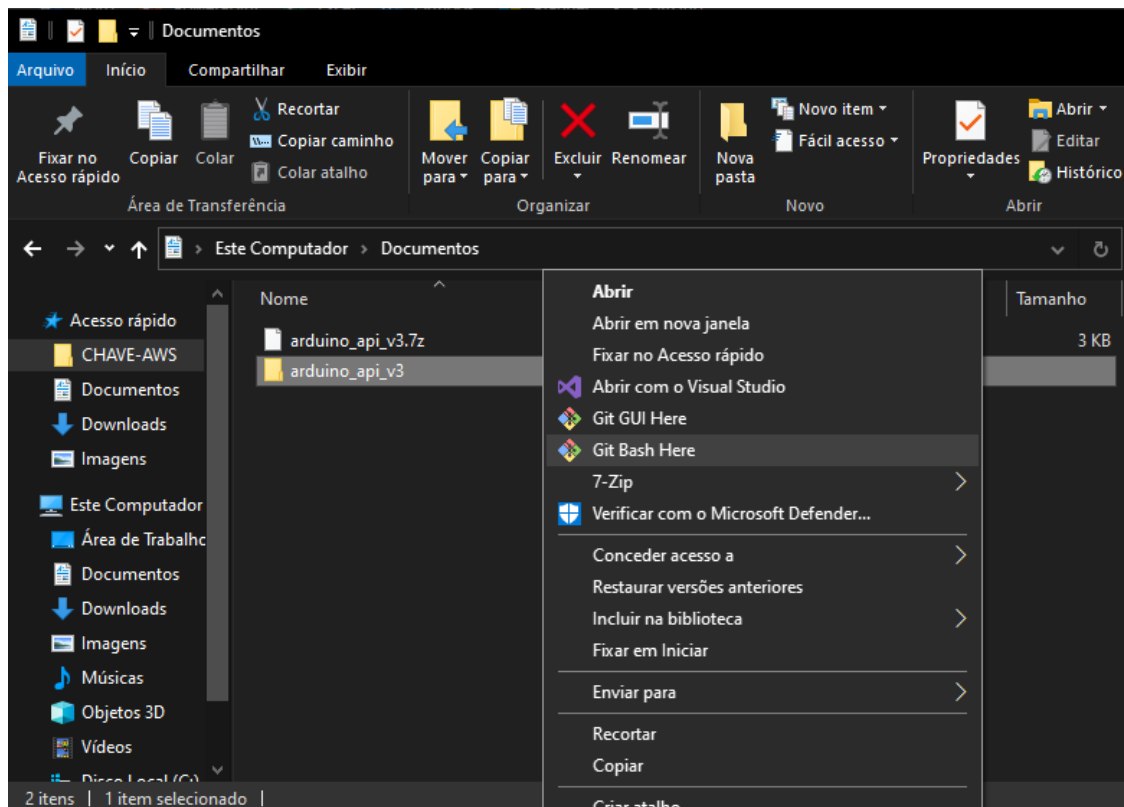
Após o download, certifique-se de colocar os arquivos **"arduino\_api\_v3.7z"** e **"FIVE\_SENSORS\_V04.ino"** dentro do diretório **"Documents"** clique com o botão direito do mouse sobre o arquivo **"arduino\_api\_v3.7z"** e selecione a opção **"extrair aqui"**:



Agora, abra o **“Visual Studio Code (VScode)”** e clique em **“Arquivo”** no canto superior esquerdo, selecione a opção **“Open Folder (Abrir Pasta)”**, uma janela aparecerá para que selecione o diretório que acabamos de extrair:



Em sequência, abra o diretório **"Documents"** e clique com o botão direito do mouse sobre o diretório que acabamos de extrair e selecione a opção **"Git Bash Here"**:



Em seguida digite o comando **npm install** no terminal, o tempo de instalação desse processo pode variar dependendo da conexão, ***não se desespere!***

```
MINGW64:/c:/Users/Aluno/Documents/arduino_api_v3
Aluno@SPHDK2648 MINGW64 ~/Documents/arduino_api_v3
$ npm install

added 89 packages, and audited 90 packages in 13s

14 packages are looking for funding
  run `npm fund` for details

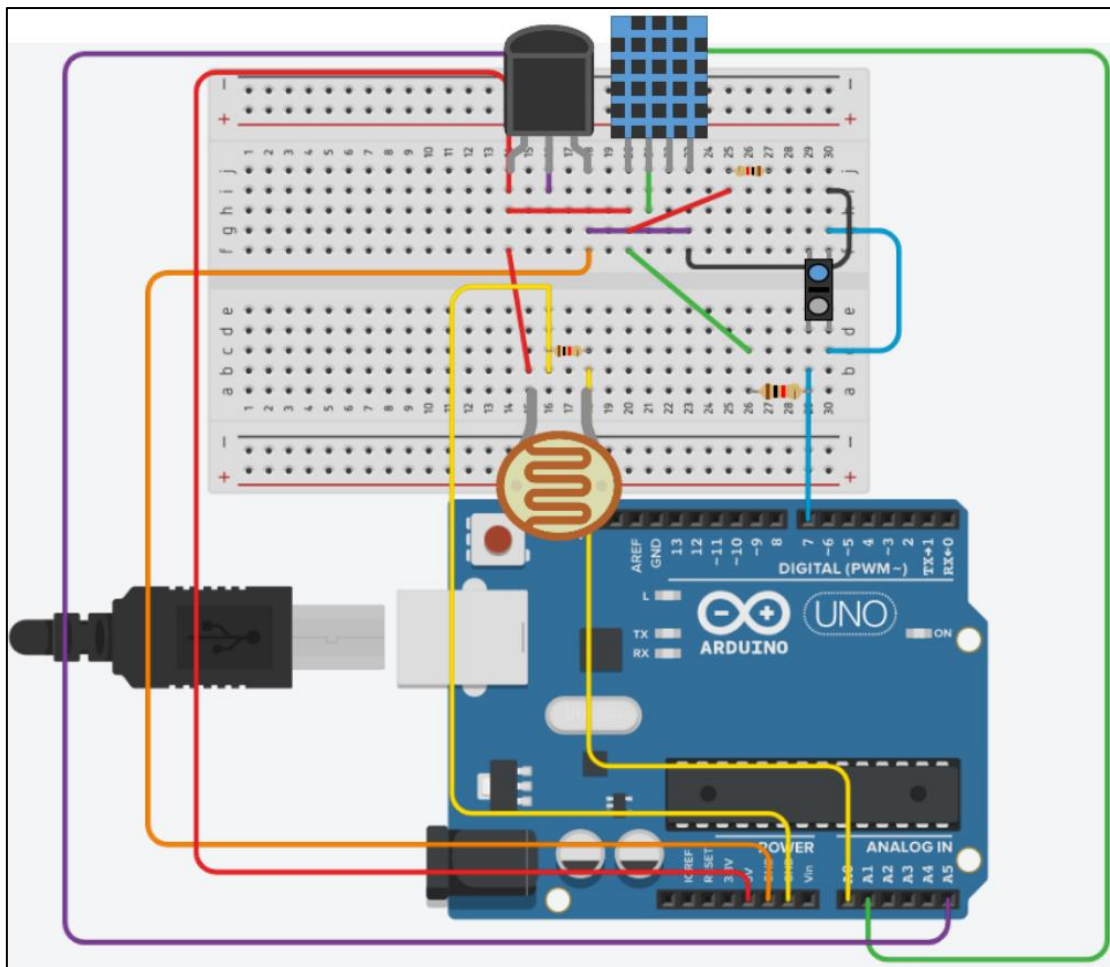
found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 8.1.2 -> 8.6.0
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v8.6.0>
npm notice Run `npm install -g npm@8.6.0` to update!
npm notice








Aluno@SPHDK2648 MINGW64 ~/Documents/arduino_api_v3
$ |
```

**\*\* Caso o terminal mostre um erro no comando, certifique-se que tenha o **NodeJS** instalado, você precisará instalar o mesmo em sua máquina \***

## PASSOS IMPORTANTÍSSIMOS:

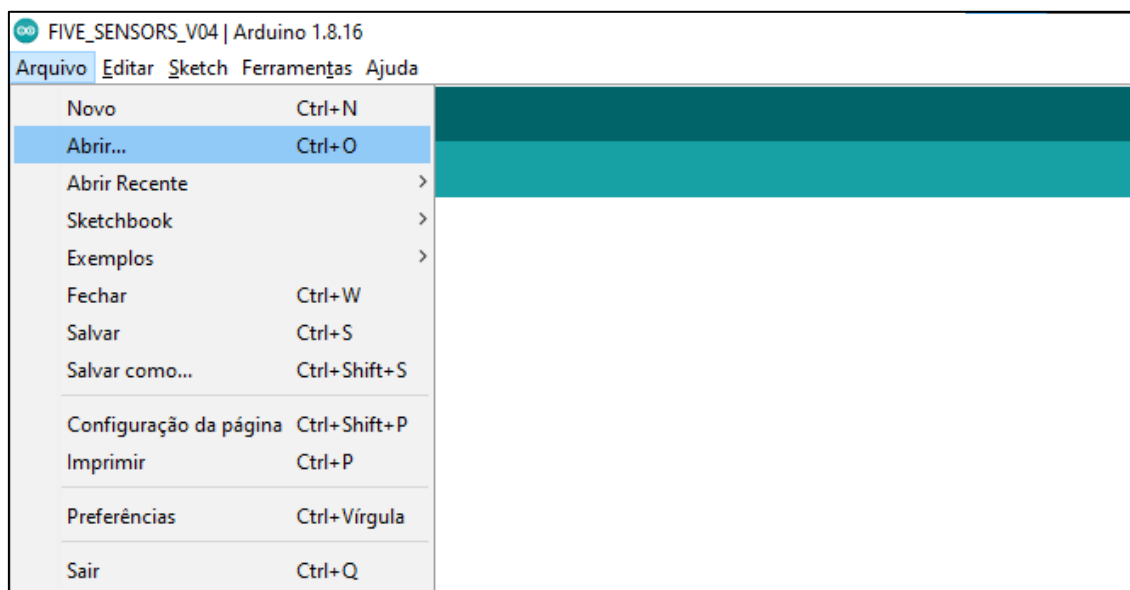
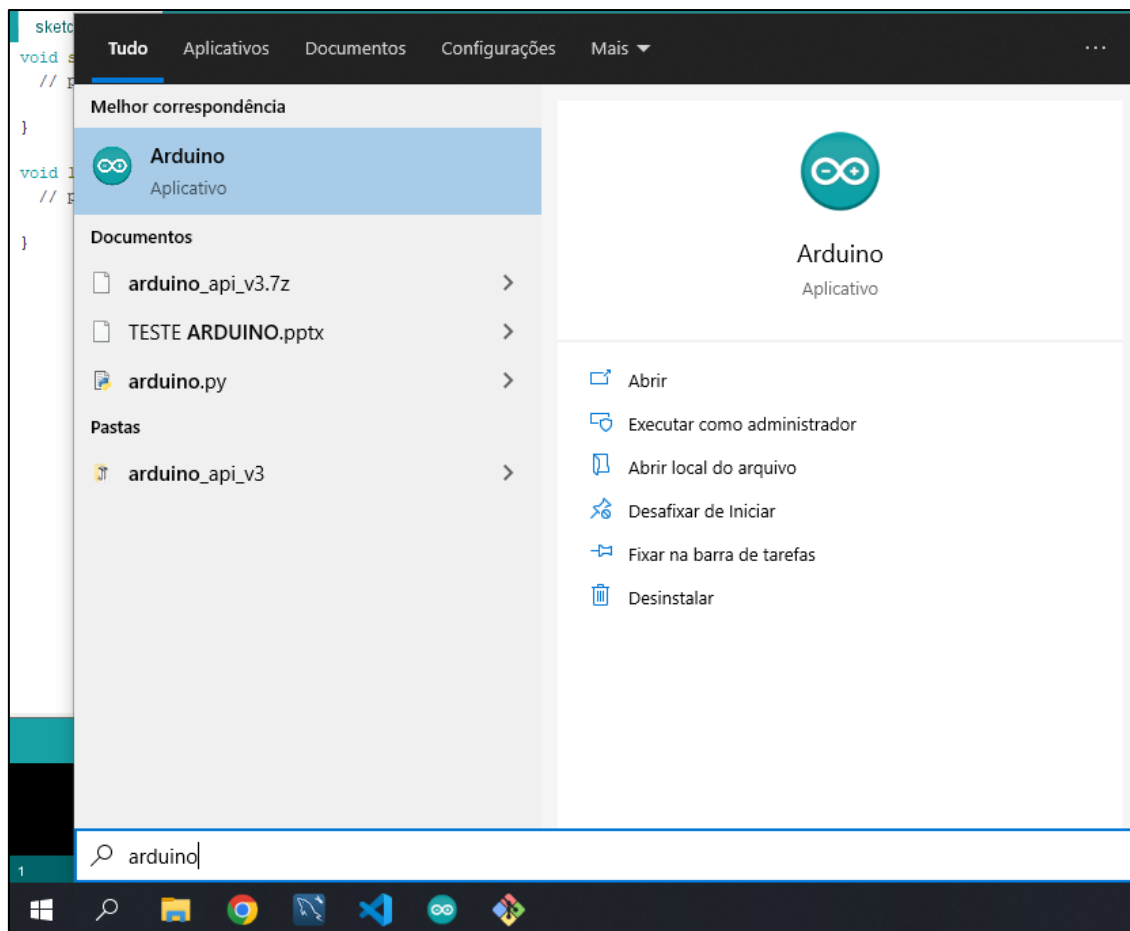
*Você já sabe como que vai funcionar a montagem do sensor do seu grupo?* Confira abaixo como será a montagem do arduino com os **CINCO SENSORES**:

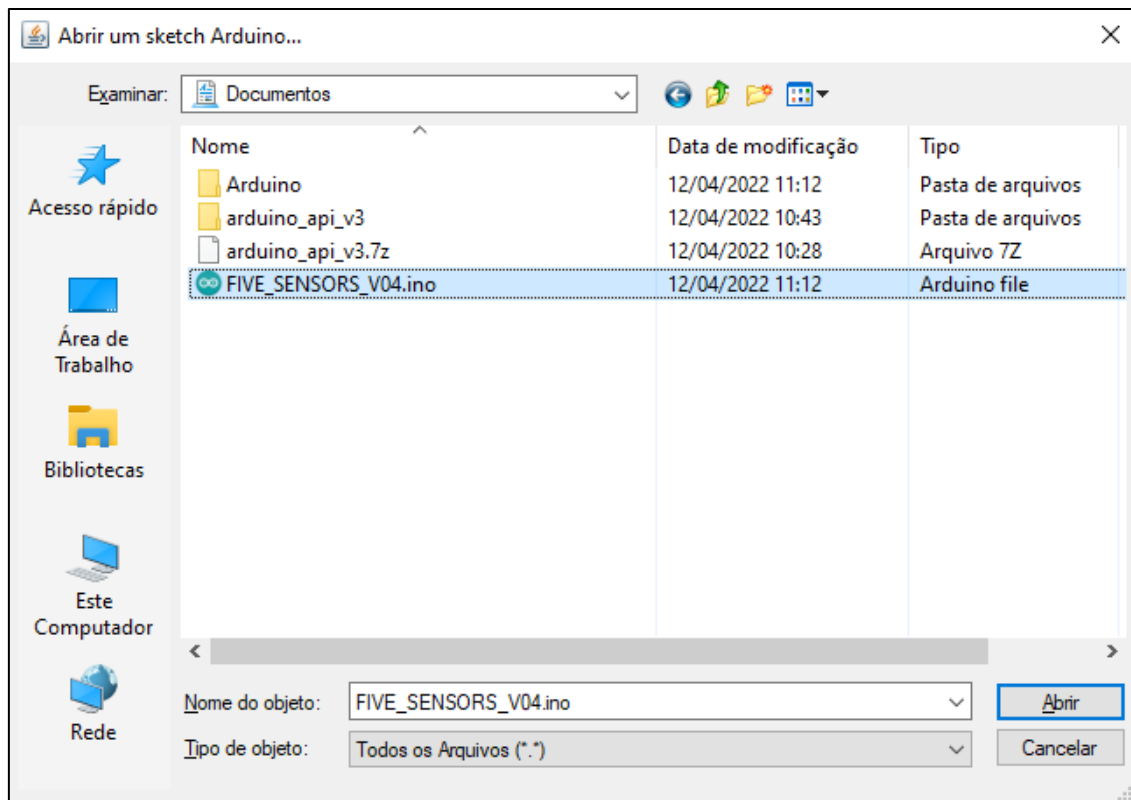


-  SENSOR TCRT5000
-  SENSOR DHT11
-  SENSOR LM35
-  SENSOR TCR
-  VOLTAGEM
-  GND
-  RESISTOR

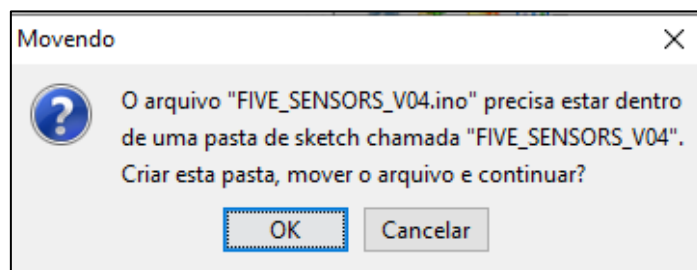
Importante ressaltar que os fios (jumpers) não precisam necessariamente ser das mesmas cores da legenda, mas para que você os identifique com maior facilidade repita conforme acima. Assim que concluir a montagem, chame um professor para que faça a validação e então possamos aos testes a seguir.

Abra a **IDE do Arduino**, clique em **"Arquivo"** e **"Abrir"**, selecione em **"Documents"** o arquivo **"FIVE\_SENSORS\_V04.ino"** que baixamos no moodle:

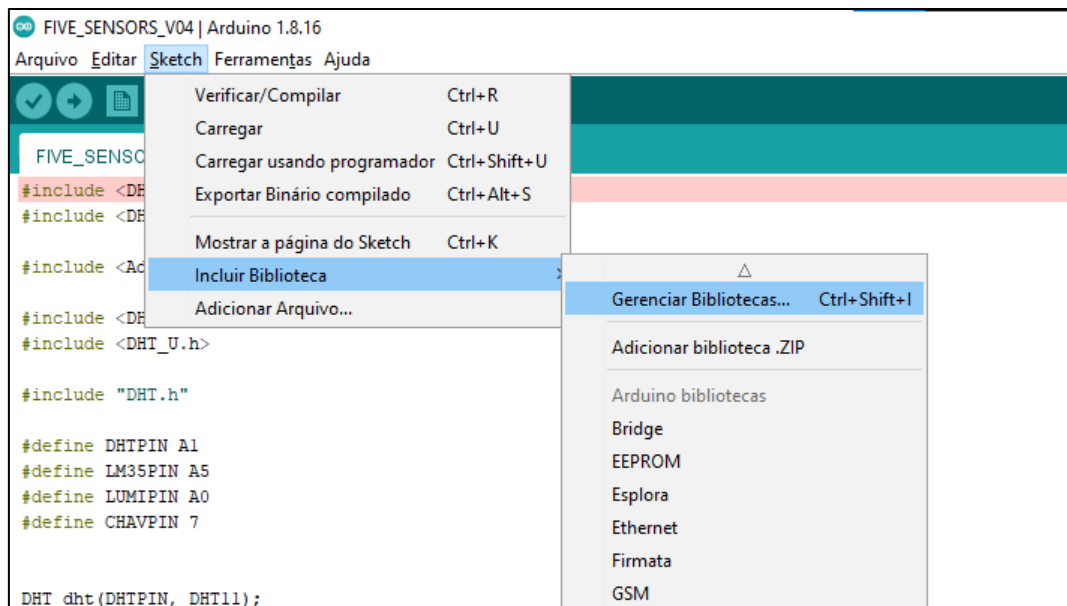




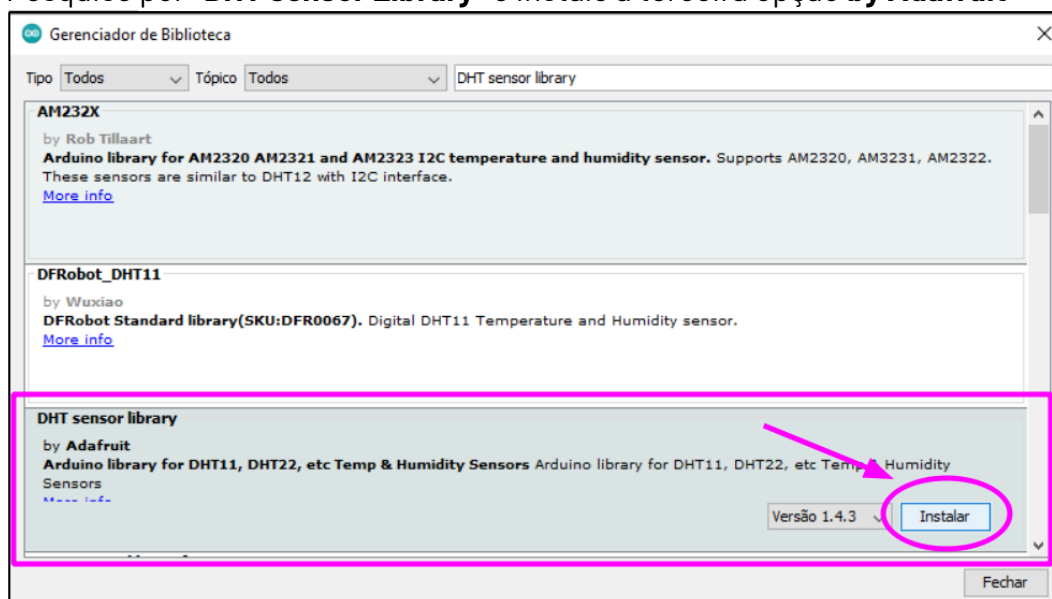
Aparecerá uma janela pedindo para criar uma pasta para o arquivo, clique em "OK":



Quando a **IDE Arduino** abrir o **script**, é importante que seja feito o download das bibliotecas que serão utilizadas. Abra "**Sketch**", passe o mouse sobre "**Incluir Biblioteca**" e acesse "**Gerenciar Biblioteca**":

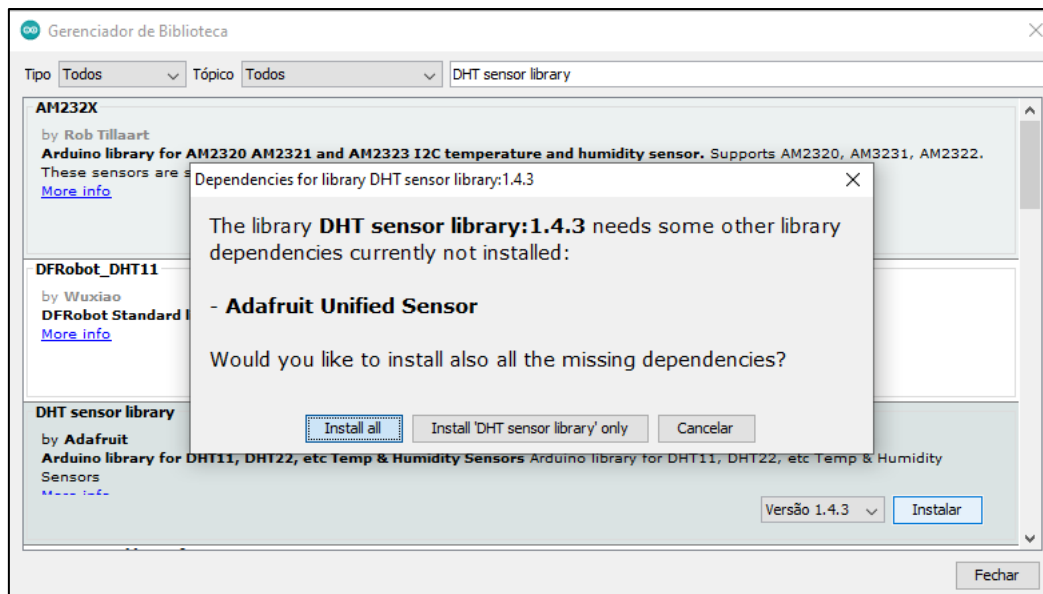


Pesquise por **"DHT sensor Library"** e instale a terceira opção **by Adafruit**:



Clique em **"Install All"** para vir todas as dependências dessa biblioteca:





Em seguida clique no **botão de verificação**  para que o **script** seja executado e a **IDE Arduino** certifique que não contém nenhum erro na programação:

```
FIVE_SENSORS_V04
#include <DHT.h>
#include <DHT_U.h>

#include <Adafruit_Sensor.h>

#include <DHT.h>
#include <DHT_U.h>

#include "DHT.h"

#define DHTPIN A1
#define LM35PIN A5
#define LUMIPIN A0
#define CHAVPIN 7

DHT dht(DHTPIN, DHT11);

void setup()
{
  pinMode(DHTPIN, INPUT);
  pinMode(CHAVPIN, INPUT);
  Serial.begin(9600);
  dht.begin();
}

void loop()
{
  float dht11 umidade = dht.readHumidity();
}
```

Quando esta mensagem aparecer no terminal abaixo, **chame o professor para validação**:

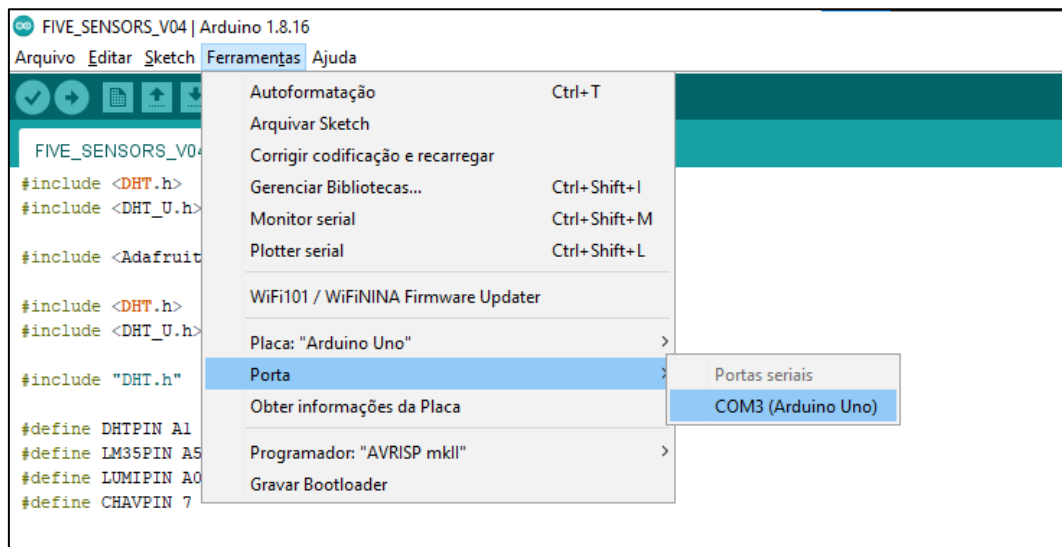
```
Compilação terminada.

O sketch usa 5236 bytes (16%) de espaço de armazenamento para programas. O máximo são 32256 bytes.
Variáveis globais usam 225 bytes (10%) de memória dinâmica, deixando 1823 bytes para variáveis locais. O máximo são 2048 bytes.

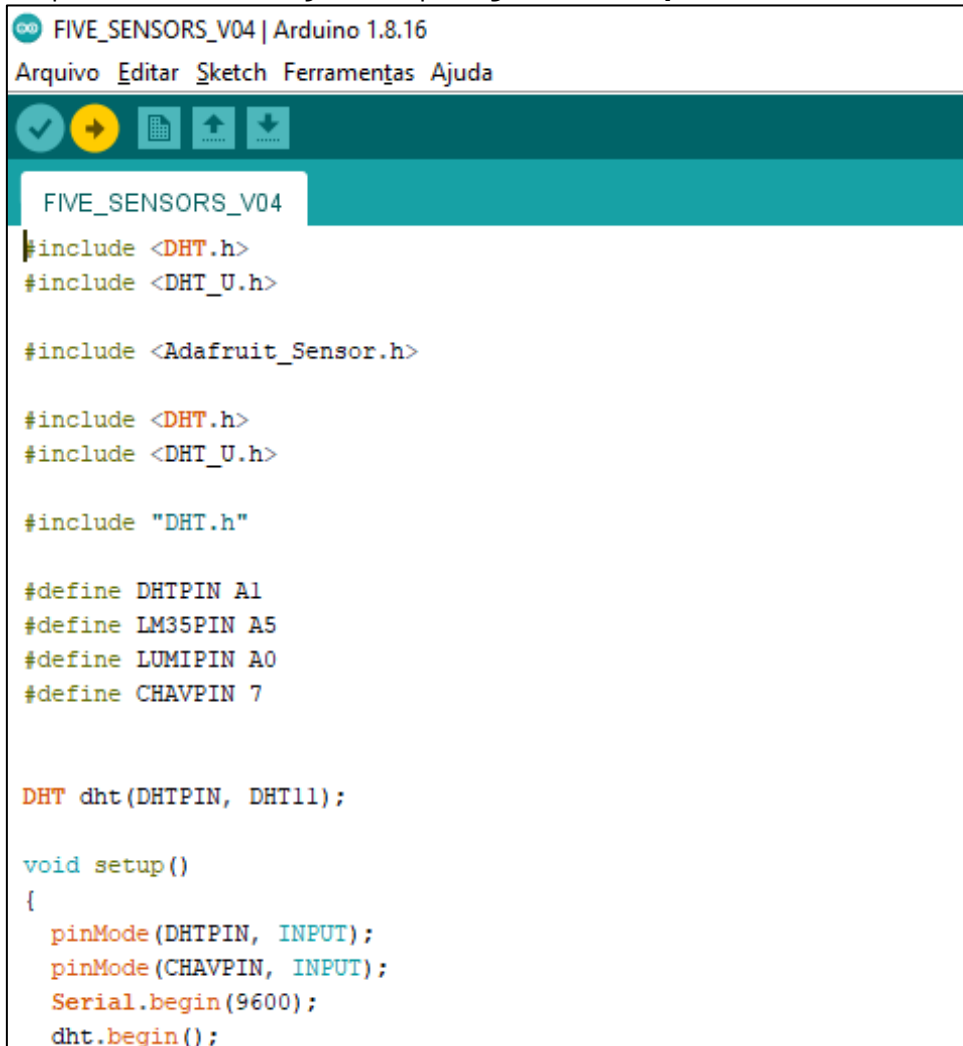
1
```

Após o mesmo validar, insira o arduino em uma porta **USB**, para que a **IDE Arduino** identifique a placa. Clique em **"Ferramentas"**, passe o mouse sobre **"Porta"** e selecione a porta disponível, neste caso é a porta serial

### COM3 (Arduino Uno):



Clique no botão carregar , para **gravar** o **script** dentro do arduino:



```
FIVE_SENSORS_V04 | Arduino 1.8.16
Arquivo Editar Sketch Ferramentas Ajuda

FIVE_SENSORS_V04
#include <DHT.h>
#include <DHT_U.h>

#include <Adafruit_Sensor.h>

#include <DHT.h>
#include <DHT_U.h>

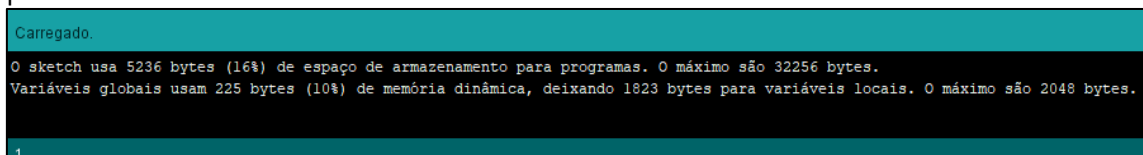
#include "DHT.h"

#define DHTPIN A1
#define LM35PIN A5
#define LUMIPIN A0
#define CHAVPIN 7

DHT dht(DHTPIN, DHT11);

void setup()
{
  pinMode(DHTPIN, INPUT);
  pinMode(CHAVPIN, INPUT);
  Serial.begin(9600);
  dht.begin();
}
```

Quando a mensagem abaixo surgir na parte inferior da **IDE Arduino**, vamos para o passo final:



```
Carregado.
O sketch usa 5236 bytes (16%) de espaço de armazenamento para programas. O máximo são 32256 bytes.
Variáveis globais usam 225 bytes (10%) de memória dinâmica, deixando 1823 bytes para variáveis locais. O máximo são 2048 bytes.
```

No **VSCo**de existe uma **ÚNICA** palavra que dará o toque final, a cereja do bolo para que a nossa **API** insira os dados no banco de dados, preste atenção.

Dentro do arquivo **"main.js"**, existe uma linha de código exatamente assim **"const HABILITAR\_OPERACAO\_INSERT = false"**:

```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  main.js - arduino_api_v3 - Visual Studio Code

EXPLORADOR
ARDUINO_API_V3
  > node_modules
  < index.html 3
  JS main.js
  {} package-lock.json
  {} package.json

JS main.js > [0] HABILITAR_OPERACAO_INSERIR
1  const serialport = require('serialport');
2  const express = require('express');
3  const mysql = require('mysql2');
4
5  const SERIAL_BAUD_RATE = 9600;
6  const SERVIDOR_PORTA = 3000;
7  const HABILITAR_OPERACAO_INSERIR = false;
8
9  const serial = async (
10     valoresDht11Umidade,
11     valoresDht11Temperatura,
12     valoresLuminosidade,
13     valoresLm35Temperatura,
14     valoresChave
15   ) => {
16     const poolBancoDados = mysql.createPool({
```

Substitua o valor dessa linha de **"false"** para **"true"**:

```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  main.js - arduino_api_v3 - Visual Stud

EXPLORADOR
ARDUINO_API_V3
  > node_modules
  < index.html 3
  JS main.js
  {} package-lock.json
  {} package.json

JS main.js > ...
1  const serialport = require('serialport');
2  const express = require('express');
3  const mysql = require('mysql2');
4
5  const SERIAL_BAUD_RATE = 9600;
6  const SERVIDOR_PORTA = 3000;
7  const HABILITAR_OPERACAO_INSERIR = true;
8
9  const serial = async (
10     valoresDht11Umidade,
11     valoresDht11Temperatura,
12     valoresLuminosidade,
13     valoresLm35Temperatura,
14     valoresChave
15   ) => {
```

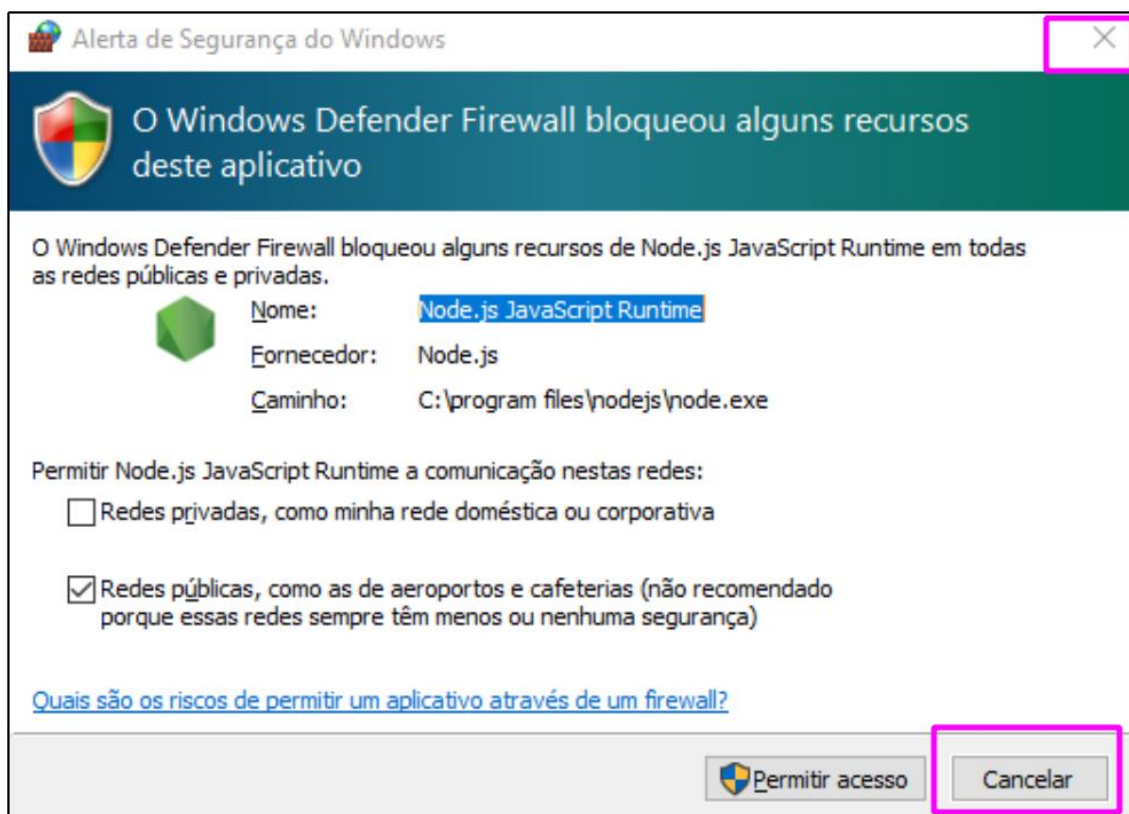
Abra o terminal do **GitBash** e insira o comando **npm start**, o Windows pedirá permissão de acesso, feche a janela ou clique em cancelar:

```
MINGW64:/c/Users/Aluno/Documents/arduino_api_v3

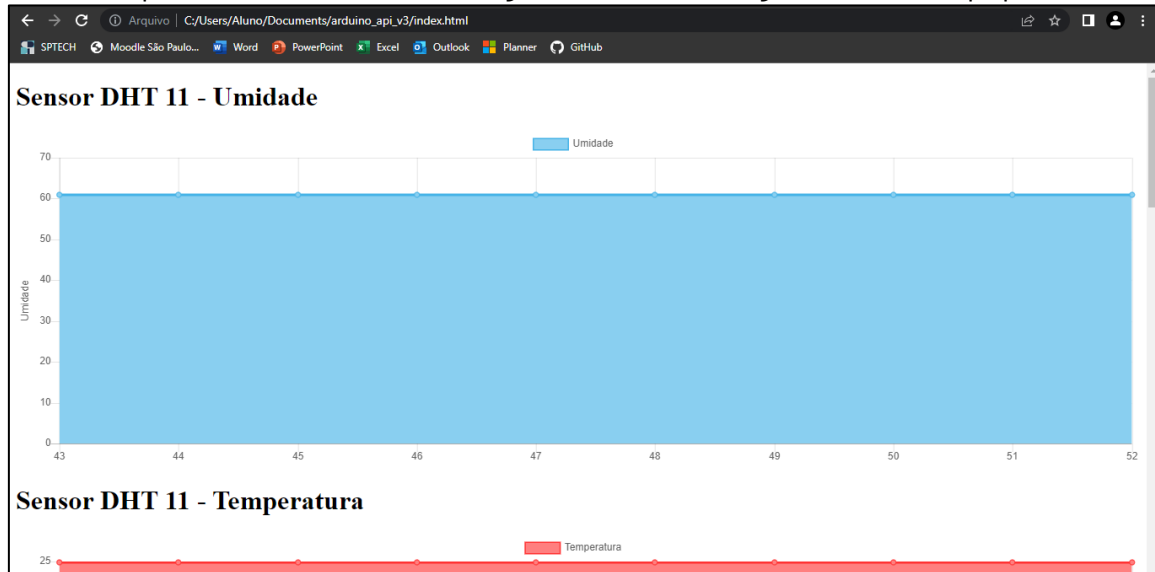
Aluno@SPHDK2648 MINGW64 ~/Documents/arduino_api_v3
$ npm start

> arduino-api@1.0.0 start
> node main.js

API executada com sucesso na porta 3000
A leitura do arduino foi iniciada na porta COM3 utilizando Baud Rate de 9600
```



Abra o arquivo **"index.html"** no navegador e confira os gráficos sendo populados:



Depois abra o **Workbench MySQL**, faça um **SELECT** da tabela para trazer os dados que estão sendo inseridos:

The screenshot shows the MySQL Workbench interface. The query editor at the top contains the query: `SELECT * FROM sensores;`. Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The results are shown in a table with 5 columns: `dht11_umidade`, `dht11_temperatura`, `luminosidade`, `lm35_temperatura`, and `chave`. There are 7 rows of data.

	dht11_umidade	dht11_temperatura	luminosidade	lm35_temperatura	chave
	61	24.9	886	44.9	0
	61	24.9	895	44.9	0
	61	24.9	886	44.9	0
	61	24.9	895	45.38	0
	61	24.9	886	44.41	0
	61	24.9	895	45.38	0
	61	24.9	888	45.38	0

## DICA DE OURO

QUER ENCERRAR A API CORRETAMENTE? NÃO FECHÉ O TERMINAL, CASO CONTRÁRIO QUANDO REINICIAR A API O TERMINAL APONTARÁ ERRO DE PORTA, POIS NÃO FOI FECHADA DA MANEIRA IDEAL.

APERTE AS TECLAS **CTRL + C** NO TERMINAL DO GITBASH.



