

São Paulo Tech School

## **Manual de instalação e Configuração do sistema de Monitoramento de Gás**

**Versão: 1.1 | Data: dezembro de 2024**

### **Grupo 4**

**JOÃO PEDRO FERRAZ - RA: 01242071**

**LUCAS AIELLO - RA: 01242123**

**LUCAS PEREIRA - RA: 01242038**

**MIGUEL ANGEL - RA: 01242107**

**SHELLY NADUDVARI - RA: 01242093**

**THIAGO SANCHEZ - RA: 01242006**

**São Paulo - 2024**

## **1. Introdução**

Este manual detalha o processo completo de instalação e configuração do sensor de gás MQ-2 integrado a um sistema de monitoramento que inclui coleta de dados via API, armazenamento em banco de dados e exibição em um dashboard web.

## **2. Requisitos**

### **2.1. Equipamentos Necessários**

#### **1. Hardware:**

1. Sensor de Gás MQ-2
2. Placa Arduino Uno R3
3. Fonte de alimentação para Arduino (USB conectado ao Computador)
4. Protoboard
5. Fios jumper
6. Notebook

#### **2. Software:**

1. Arduino IDE (última versão)
2. Node.js (para a API)
3. MySQL Server + (Mysql Workbench)
4. Virtual Machine (Opcional)
5. Visual Studio Code (HTML, CSS, Javascript, API)
6. GitHub (Versionamento do Projeto)

#### **3. Requisitos do Ambiente:**

1. Rede Wi-Fi estável para integração com API e dashboard
2. Computador ou servidor próximo ao local de instalação para processamento de dados
3. Entrada de gás livre para implantação do Arduino R3 (MQ2, Protoboard)

## 4. Procedimento de Instalação

### 4.1. Preparação do Local

Escolha um local elevado e limpo para instalar o sensor.

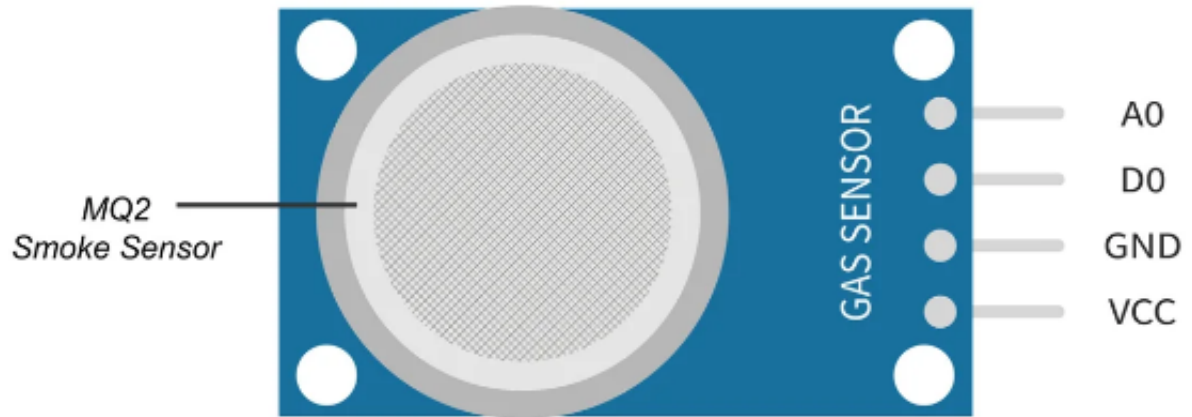


(Imagem de um exemplo de cozinha para a instalação)

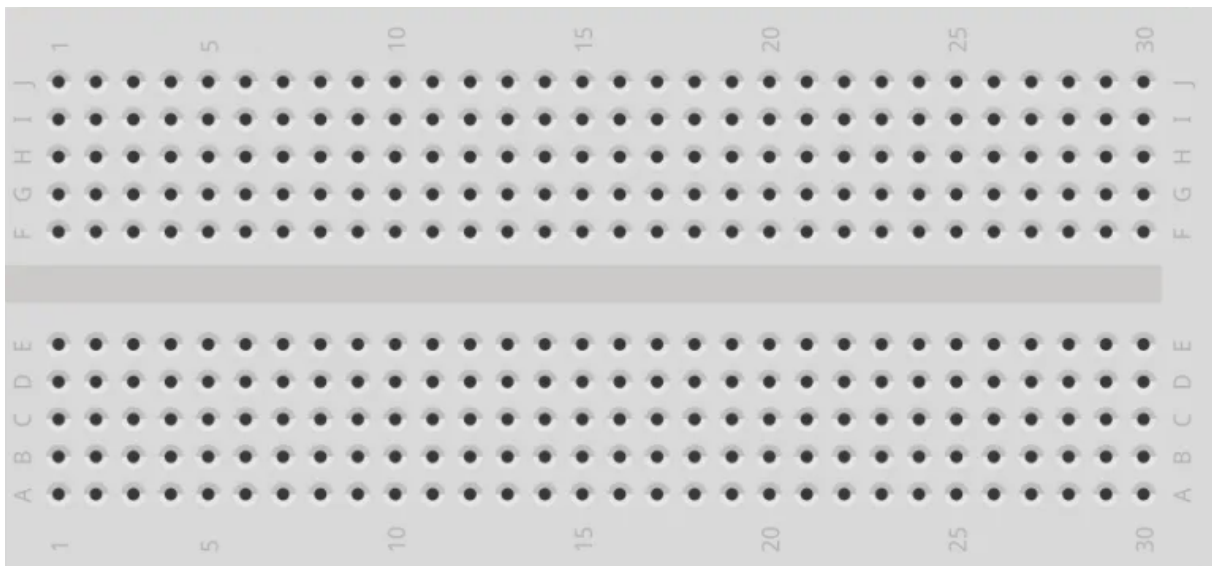
(Garanta fácil acesso ao sensor para manutenção.)

## 5. Conexões com o Sensor MQ-2

1. Conecte o sensor à protoboard.

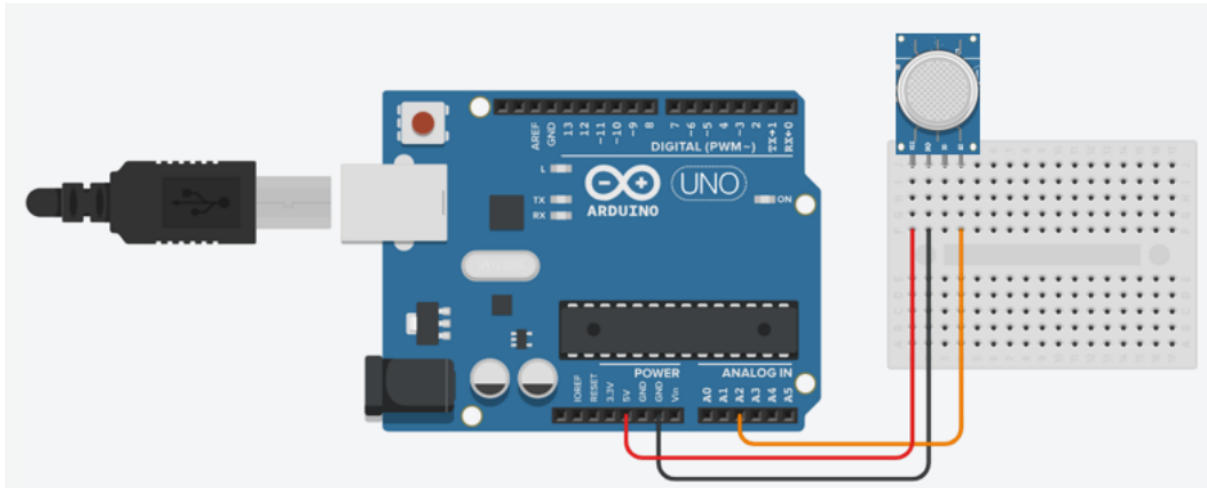


2- Protoboard



## 6. Faça as conexões dos fios (Jumpers):

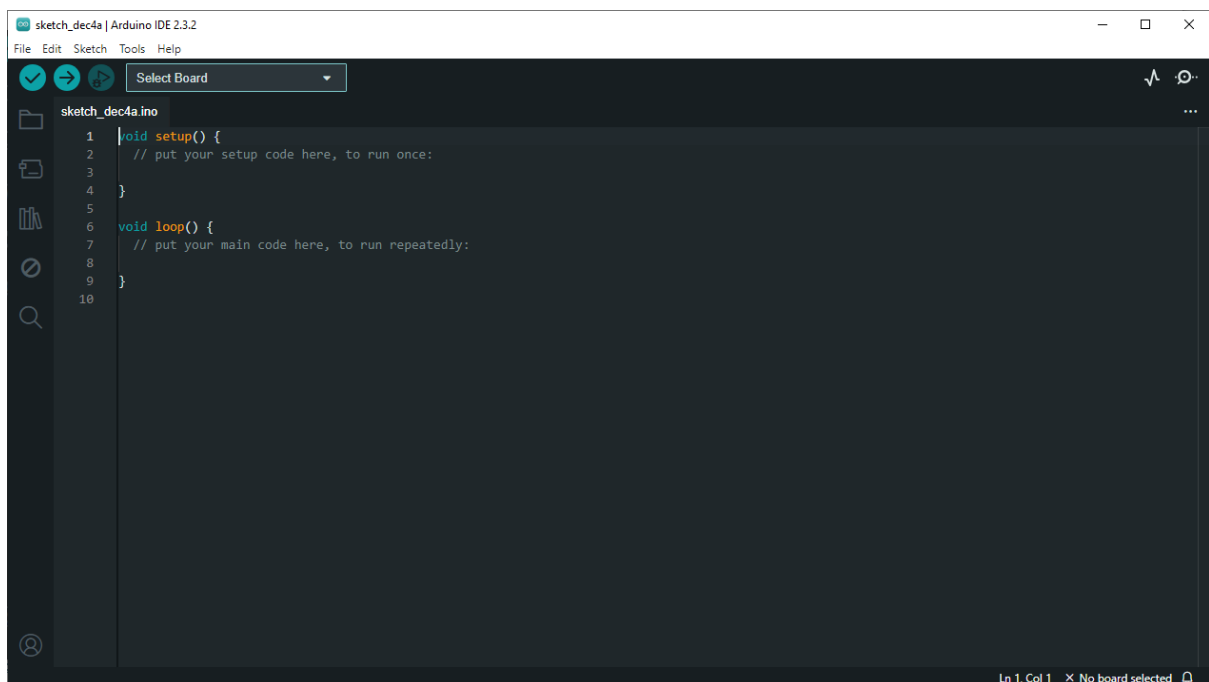
1. VCC do sensor ao 5V do Arduino. (Jumper Red)
2. GND do sensor ao GND do Arduino. (Jumper Black)
3. (A0 OUT) para a A0 do Arduino. (Jumper Blue)



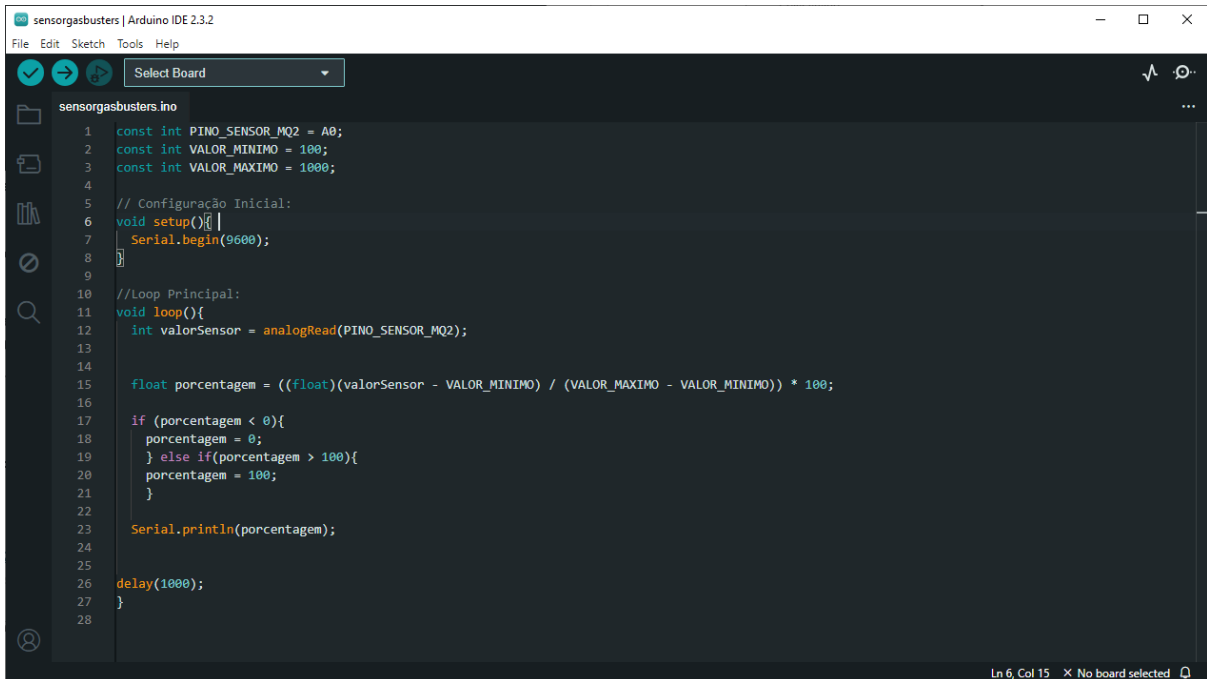
(Verifique que todas as conexões estejam corretas e firmes.)

## 7. Configuração do Código no Arduino

1. Instale a IDE (Arduino IDE) <https://www.arduino.cc/en/software>
2. Inicie a IDE



### 3. Escreva o Código e compile para verificar



```
sensorgasbusters.ino
1  const int PINO_SENSOR_MQ2 = A0;
2  const int VALOR_MINIMO = 100;
3  const int VALOR_MAXIMO = 1000;
4
5  // Configuração Inicial:
6  void setup(){
7    Serial.begin(9600);
8  }
9
10 //Loop Principal:
11 void loop(){
12   int valorSensor = analogRead(PINO_SENSOR_MQ2);
13
14   float porcentagem = ((float)(valorSensor - VALOR_MINIMO) / (VALOR_MAXIMO - VALOR_MINIMO)) * 100;
15
16   if (porcentagem < 0){
17     porcentagem = 0;
18   } else if (porcentagem > 100){
19     porcentagem = 100;
20   }
21
22   Serial.println(porcentagem);
23
24   delay(1000);
25 }
26
27
28
```

#### CODIGO

```
”    const int PINO_SENSOR_MQ2 = A0;

    const int VALOR_MINIMO = 100;
    const int VALOR_MAXIMO = 1000;

    void setup(){
        Serial.begin(9600);
    }

    void loop(){
        int valorSensor = analogRead(PINO_SENSOR_MQ2);

        float porcentagem = ((float)(valorSensor - VALOR_MINIMO) /
        (VALOR_MAXIMO - VALOR_MINIMO)) * 100;

        if (porcentagem < 0){
            porcentagem = 0;
        } else if (porcentagem > 100){
            porcentagem = 100;
        }

        Serial.println(porcentagem);

        delay(1000);
    }

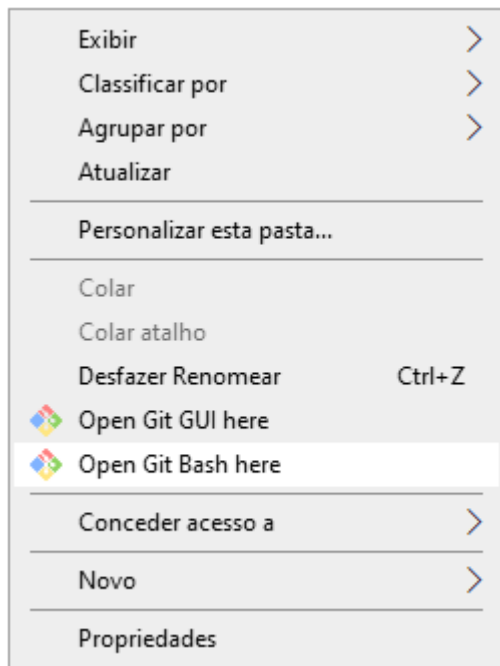
”
```

## 8. Conexão do Arduino via USB a o computador para a transferência do código

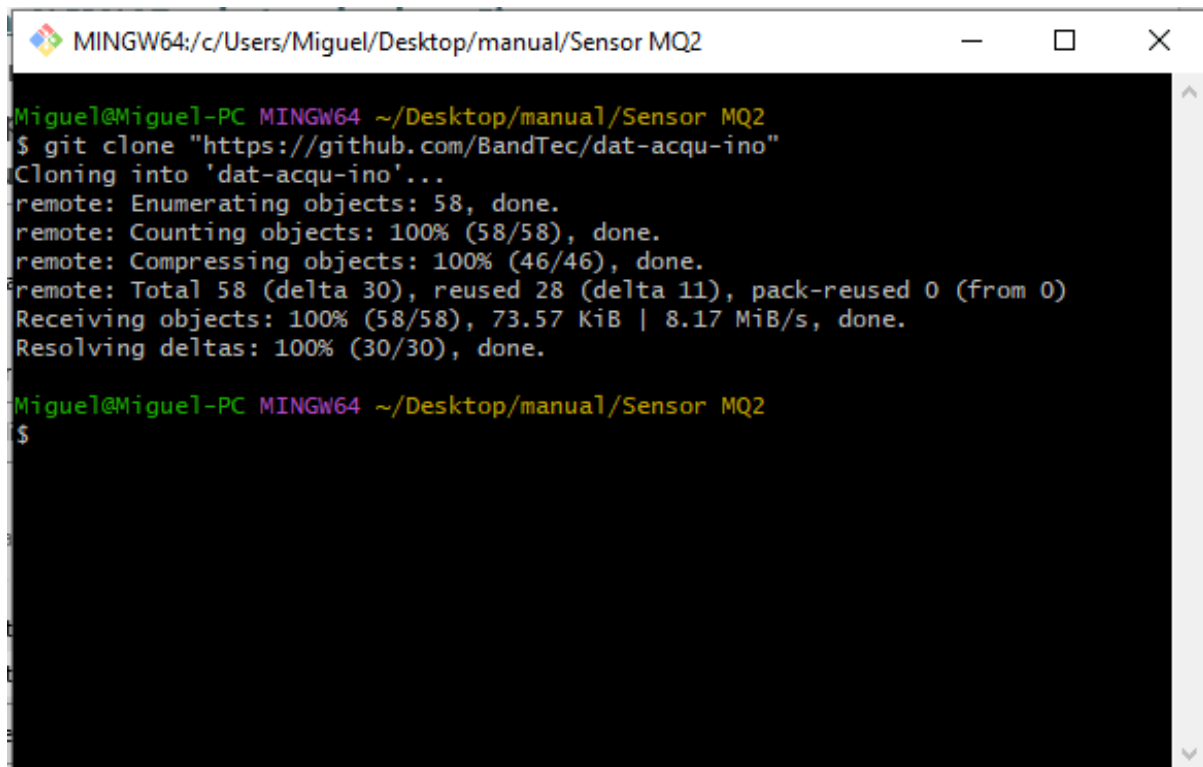
1. Conecte o arduino montado ao computador via USB
2. Transfira o código para o arduino conectado (Certifique que o código tenha sido executado antes de transferir, para evitar problemas no código).
3. Verificar se o led do sensor está ligado (led red), após a transferência do código.

## 9. Crie a API para enviar dados do sensor ao banco

1. Faça o download do Git Bash pelo link <https://git-scm.com/book/pt-br/v2/Come%C3%A7ando-Instalando-o-Git>
2. Execute e instale no notebook conectado com o Arduino. Após a instalação.
3. Crie um repositório no notebook e dentro do repositório clique com o botão esquerdo e selecione o git bash.



4. No terminal do git bash, faça o git clone do: [BandTec/dat-acqu-ino](https://github.com/BandTec/dat-acqu-ino): Data Acquisition Arduino API (API)



```
MINGW64:/c/Users/Miguel/Desktop/manual/Sensor MQ2
Miguel@Miguel-PC MINGW64 ~/Desktop/manual/Sensor MQ2
$ git clone "https://github.com/BandTec/dat-acqu-ino"
Cloning into 'dat-acqu-ino'...
remote: Enumerating objects: 58, done.
remote: Counting objects: 100% (58/58), done.
remote: Compressing objects: 100% (46/46), done.
remote: Total 58 (delta 30), reused 28 (delta 11), pack-reused 0 (from 0)
Receiving objects: 100% (58/58), 73.57 KiB | 8.17 MiB/s, done.
Resolving deltas: 100% (30/30), done.

Miguel@Miguel-PC MINGW64 ~/Desktop/manual/Sensor MQ2
$
```

5. Acesse o diretório pelo terminal usando o 'cd dat-acqu-ino'

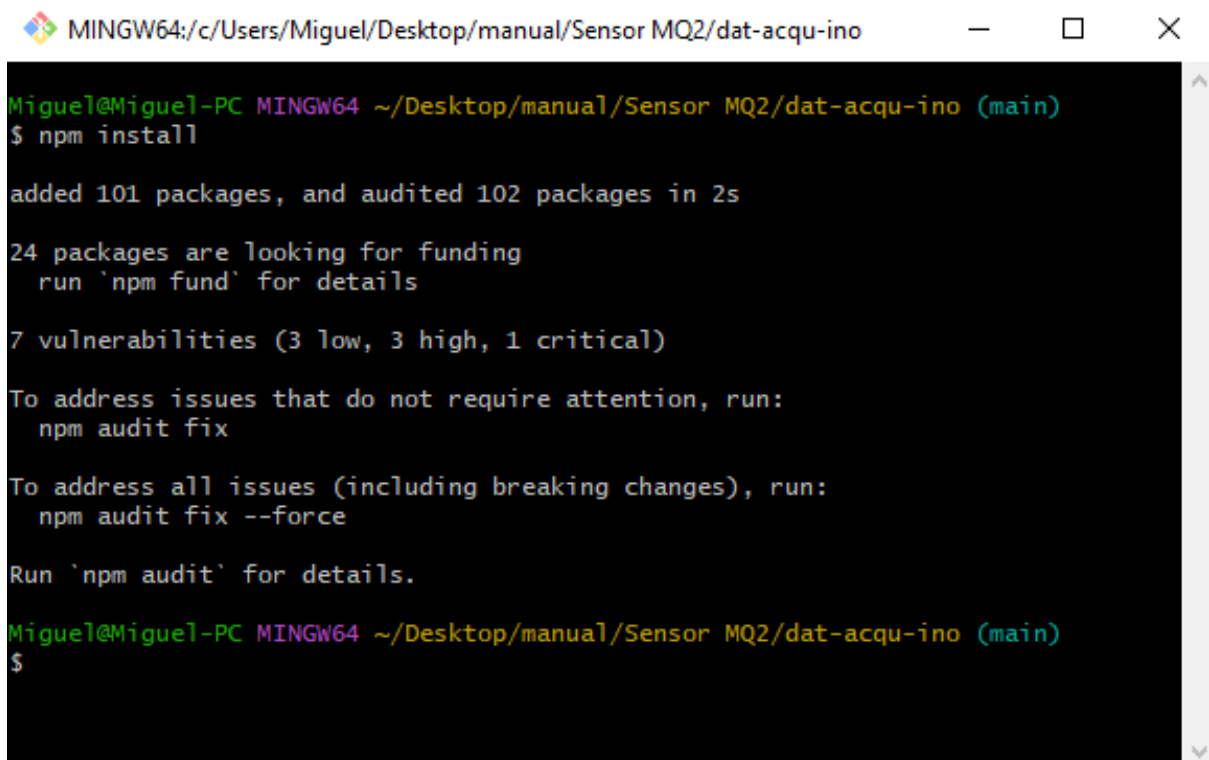


```
MINGW64:/c/Users/Miguel/Desktop/manual/Sensor MQ2/dat-acqu-ino
Miguel@Miguel-PC MINGW64 ~/Desktop/manual/Sensor MQ2
$ cd dat-acqu-ino/

Miguel@Miguel-PC MINGW64 ~/Desktop/manual/Sensor MQ2/dat-acqu-ino (main)
$ |
```



6. Faça a instalação do Modulos, usando o comando 'npm install'

A terminal window titled 'MINGW64:/c/Users/Miguel/Desktop/manual/Sensor MQ2/dat-acqu-ino' with standard window controls. The prompt is 'Miguel@Miguel-PC MINGW64 ~/Desktop/manual/Sensor MQ2/dat-acqu-ino (main)'. The command '\$ npm install' has been executed. The output shows that 101 packages were added and 102 packages were audited in 2 seconds. It also mentions 24 packages looking for funding and 7 vulnerabilities (3 low, 3 high, 1 critical). Instructions are given to run 'npm audit fix' or 'npm audit fix --force' to address issues. The prompt returns to '\$'.

## 10. Configuração do Ambiente:

1. instale o Visual Studio Code e configure o ambiente de desenvolvimento com HTML, CSS e JavaScript.
2. Atualize as Configurações da API, usando o VS Code, no diretório da API (dat-acqu-ino), localize o arquivo de configuração do banco de dados (main.js).
3. Edite o arquivo para incluir as credenciais do banco de dados, como no exemplo abaixo:

```
) => {  
  
  // conexão com o banco de dados MySQL  
  let poolBancoDados = mysql.createPool(  
    {  
      host: 'HOST_DO_BANCO',  
      user: 'USUARIO_DO_BANCO',  
      password: 'SENHA_DO_BANCO',  
      database: 'DATABASE_DO_BANCO',  
      port: 3306  
    }  
  ).promise();
```

### EXEMPLO:

“

```
host: '177.8.164.141',  
user: 'gasbusters',  
password: 'Busters#gas',  
database: 'gasbusters',  
port: 3306
```

”

4. Teste a Conexão com o Banco de Dados
5. Execute o servidor da API, no terminal com o comando:

“

```
npm start
```

“

6. Verifique no terminal se a conexão com o banco foi estabelecida com sucesso.
7. Verifique o Envio de Dados
8. Abra o navegador e acesse o endpoint da API configurado (exemplo: <http://localhost:3300/sensores>).
9. Confirme se os dados do sensor estão sendo registrados no banco de dados.

## 11. Finalização e Testes

### 1. Teste Completo do Sistema

1. Verifique se os dados do sensor estão sendo lidos corretamente.
2. Confirme que os dados estão sendo enviados à API e armazenados no banco de dados.
3. Assegure-se de que o dashboard exibe as informações em tempo real.
4. Ofereça um meio de suporte técnico para dúvidas e problemas adicionais.

## **11. Manutenção e Atualização**

1. Limpeza regular do sensor para evitar acumulação de poeira.
2. Atualizações do código e integração com o banco de dados conforme novas versões forem lançadas.

## **12. Contato para Suporte**

**E-mail: [suporte@gasbusters.com](mailto:suporte@gasbusters.com)**

**Telefone: (11) 99999-9999**