# Passo 1: Conexão do Hardware

**1. Conecte o MPU-6050 ao Arduino Uno:**

1. VCC -> 5V
2. GND -> GND
3. SDA -> A4
4. SCL -> A5

**2. Conecte o sensor de vibração SW-420 ao Arduino:**

1. VCC -> 5V
2. GND -> GND
3. DO (saída digital) -> pino digital (por exemplo, D2) **3. Conecte o ESP32 ao Arduino Uno:**
4. TX do Arduino -> RX do ESP32
5. RX do Arduino -> TX do ESP32
6. GND do Arduino -> GND do ESP32

# Passo 2: Programação do Arduino

cpp

VerifyOpen In EditorEditCopy code

1#include <Wire.h> 2#include <MPU6050.h>

3

4MPU6050 mpu;

5

6const int vibrationPin = 2; // pino do sensor de vibração 7int vibrationState = 0;

8

9void setup() {

1. Serial.begin(115200);
2. Wire.begin();
3. mpu.initialize(); 13 pinMode(vibrationPin, INPUT);

14} 15

16void loop() {

1. // Lê os dados do MPU6050
2. int16\_t ax, ay, az; 19 mpu.getAcceleration(&ax, &ay, &az);

20 21 // Lê o estado do sensor de vibração 22 vibrationState = digitalRead(vibrationPin);

23

1. // Envia os dados para o ESP32
2. Serial.print("AX: ");
3. Serial.print(ax);
4. Serial.print(", AY: ");
5. Serial.print(ay);
6. Serial.print(", AZ: ");
7. Serial.print(az);
8. Serial.print(", Vibration: "); 32 Serial.println(vibrationState); 33

34 delay(500); // Ajuste conforme necessário

35}

**Passo 3: Programação do ESP32**

cpp

VerifyOpen In EditorEditCopy code

1#include <WiFi.h>

2

3const char\* ssid = "YOUR\_SSID"; // substitua pelo seu SSID 4const char\* password = "YOUR\_PASSWORD"; // substitua pela sua senha

5const char\* serverUrl = "[http://YOUR\_SERVER\_URL"](http://your_server_url/); // URL do servidor remoto

6

7void setup() {

8 Serial.begin(115200); 9 WiFi.begin(ssid, password); 10

1. while (WiFi.status() != WL\_CONNECTED) {
2. delay(1000);
3. Serial.println("Conectando ao WiFi...");
4. } 15 Serial.println("Conectado ao WiFi");

16} 17

18void loop() {

1. if (Serial.available()) {
2. String data = Serial.readStringUntil('\n'); 21 sendDataToServer(data);

22 }

23} 24

25void sendDataToServer(String data) {

1. if (WiFi.status() == WL\_CONNECTED) {
2. WiFiClient client; 28 HTTPClient http; 29

30 http.begin(client, serverUrl); 31 http.addHeader("Content-Type", "application/json");

32

33 int httpResponseCode = http.POST(data);

34

1. if (httpResponseCode > 0) {
2. String response = http.getString();
3. Serial.println(httpResponseCode);
4. Serial.println(response);
5. } else { 40 Serial.print("Erro ao enviar: ");
   1. Serial.println(httpResponseCode);
   2. }
   3. http.end();
   4. } else {
   5. Serial.println("Erro na conexão WiFi");
   6. }

47}

**Passo 4: Configuração do Servidor**

Você pode usar um servidor simples em Node.js, Python Flask ou PHP para receber os dados enviados pelo ESP32. O servidor deve escutar as requisições POST e armazenar ou processar os dados recebidos.

**Passo 5: Interface Web**

Crie uma interface web simples para exibir os dados em tempo real. Você pode usar HTML, CSS e JavaScript com AJAX para atualizar a página sem recarregá-la.

html

VerifyOpen In EditorEditRunCopy code

1<!DOCTYPE html>

2<html lang="pt-BR"> 3<head>

4 <meta charset="UTF-8"> 5 <meta name="viewport" content="width=device-width, initialscale