

API:

```
from flask import Flask, jsonify, request
```

```
app = Flask(__name__)
```

```
# Variáveis globais para armazenar os dados
```

```
gas_value = 0
```

```
status = "Seguro"
```

```
@app.route('/')
def home():
```

```
    return "<h1>Bem-vindo à API do Sensor de Gás!</h1><p>Use a rota /api/sensor para  
    acessar os dados do sensor.</p>"
```

```
@app.route('/api/sensor', methods=['GET', 'POST'])
```

```
def sensor():
```

```
    global gas_value, status
```

```
    if request.method == 'POST':
```

```
        # Recebe dados do sensor (exemplo: {"gas_value": 450})
```

```
        data = request.json
```

```
        if 'gas_value' in data:
```

```
            gas_value = data['gas_value']
```

```
            # Atualiza o status baseado no valor do gás
```

```
            status = "Perigoso" if gas_value >= 450 else "Seguro"
```

```
            return jsonify({"message": "Dados recebidos"}), 200
```

```
        return jsonify({"error": "Dados inválidos"}), 400
```

```
    else:
```

```
        # Retorna os dados atuais do sensor
```

```
        return jsonify({"gas_value": gas_value, "status": status})
```

```
if __name__ == '__main__':
```

```
    app.run(host='0.0.0.0', port=5000)
```

Banco De Dados:

```
CREATE DATABASE Sensor_SAG  
GO
```

```
USE Sensor_SAG  
GO
```

```
CREATE TABLE Sensors (  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    SensorName NVARCHAR(50) NOT NULL,  
    SensorType NVARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE GasSensorReadings (  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    GasLevel NVARCHAR(10) NOT NULL,  
    Timestamp DATETIME NOT NULL DEFAULT GETDATE(),  
    SensorId INT NOT NULL,  
    FOREIGN KEY (SensorId) REFERENCES Sensors(Id)  
);
```

Tinkercard:

```
#include <ESP8266WiFi.h>
//PEGAR AS SENHAS
const char* ssid = "YOUR_SSID";    // Nome da sua rede Wi-Fi
const char* password = "YOUR_PASSWORD"; // Senha da sua rede Wi-Fi
const char* server = "http://192.168.1.x:5000/api/sensor"; // Substitua pelo seu IP

const int gasPin = A0;           // Pino do sensor de gás
const int threshold = 450;       // Limite para o sensor de gás

void setup() {
  Serial.begin(115200);
  pinMode(gasPin, INPUT);

  // Conexão Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando ao WiFi...");
  }
  Serial.println("Conectado ao WiFi");
}

void loop() {
  int gasValue = analogRead(gasPin); // Lê o valor do gás
  Serial.print("Intensidade do gás: ");
  Serial.println(gasValue);

  // Envia dados para a API
  WiFiClient client;
  if (client.connect("192.168.1.x", 5000)) { // Use seu IP aqui
    String json = String("{\"gas_value\": " + gasValue + "}");

    client.println("POST /api/sensor HTTP/1.1");
    client.println("Host: 192.168.1.x");
    client.println("Content-Type: application/json");
    client.print("Content-Length: ");
    client.println(json.length());
    client.println();
    client.println(json);

    // Lê a resposta da API
    while (client.connected() || client.available()) {
      if (client.available()) {
        String line = client.readStringUntil('\n');
        Serial.println(line);
      }
    }
  }
}
```

```
    }  
    client.stop();  
  } else {  
    Serial.println("Falha ao conectar ao servidor");  
  }  
  
  delay(10000); // Aguarda 10 segundos antes de enviar novamente  
}
```