PROJETO WOKWI

<https://wokwi.com/projects/432864299840487425>

VÍDEO EXPLICATIVO

<https://youtu.be/1Kci4tR4BBo>

CÓDIGO WOKWI

#include "DHT.h"

#define DHTPIN 4          // Pino de dados do DHT22

#define DHTTYPE DHT22     // Tipo do sensor DHT

#define LDRPIN 34         // GPIO34 (VN)

DHT dht(DHTPIN, DHTTYPE);

void setup() {

**Serial**.begin(9600);

  dht.begin();

}

void loop() {

  float temp = dht.readTemperature();

  float umidade = dht.readHumidity();

  int ldrValue = analogRead(LDRPIN); // valor entre 0 e 4095

  float pressao = map(ldrValue, 0, 4095, 850, 1050); // converte p/ faixa de mB

**Serial**.println("=== LEITURA DOS SENSORES ===");

  if (!isnan(temp) && !isnan(umidade)) {

**Serial**.print("Temperatura: ");

**Serial**.print(temp);

**Serial**.println(" °C");

**Serial**.print("Umidade: ");

**Serial**.print(umidade);

**Serial**.println(" %");

**Serial**.print("Pressão simulada: ");

**Serial**.print(pressao);

**Serial**.println(" mB");

  } else {

**Serial**.println("Erro ao ler DHT22.");

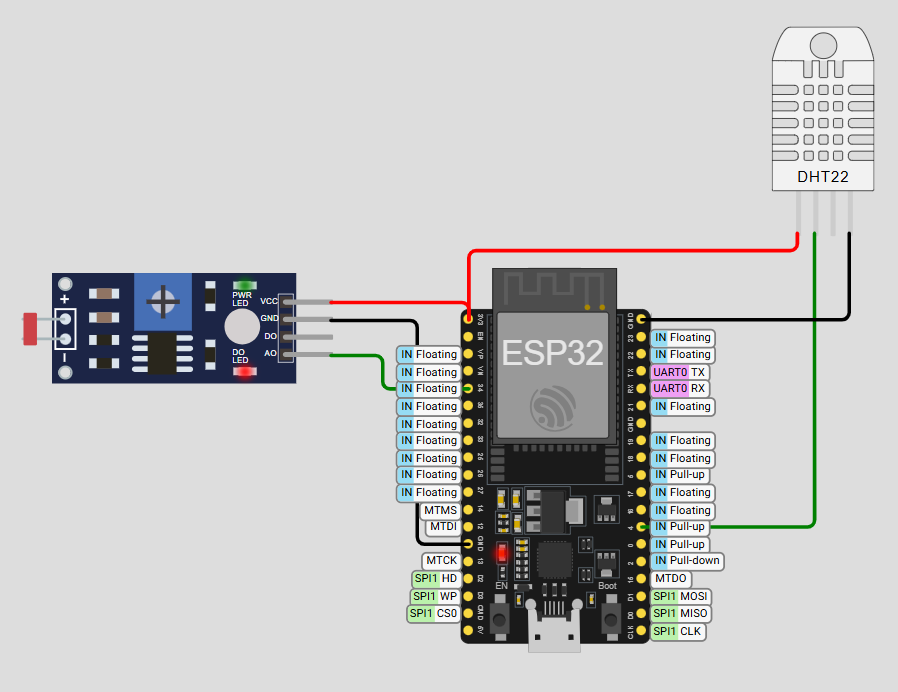
  }

**Serial**.println("============================\n");

  delay(5000); // Aguarda 5 segundos

}

CIRCUITO WOKWI



EXEMPLO DE LEITURA/PAINEL WOKWI

=== LEITURA DOS SENSORES ===

Temperatura: 11.80 °C

Umidade: 40.00 %

Pressão simulada: 898.00 mB

============================

=== LEITURA DOS SENSORES ===

Temperatura: 34.10 °C

Umidade: 60.00 %

Pressão simulada: 859.00 mB

=== LEITURA DOS SENSORES ===

Temperatura: 31.30 °C

Umidade: 34.50 %

Pressão simulada: 910.00 mB

SQL

CREATE TABLE monitoramento\_calor (

id NUMBER GENERATED BY DEFAULT AS IDENTITY,

temperatura FLOAT,

umidade FLOAT,

pressao FLOAT,

status VARCHAR2(30),

data\_insercao TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

COLAB

# 1. Instalar e importar bibliotecas

!pip install scikit-learn joblib --quiet

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_absolute\_error, r2\_score

import joblib

# 2. Carregar e preparar os dados

df = pd.read\_csv("/content/INMET\_CO\_DF\_A001\_BRASILIA\_07-05-2000\_A\_31-12-2000.CSV",

                 sep=";", encoding="latin1", skiprows=8)

df = df.rename(columns={

    'DATA (YYYY-MM-DD)': 'data',

    'TEMPERATURA MÁXIMA NA HORA ANT. (AUT) (°C)': 'temp\_max',

    'TEMPERATURA DO AR - BULBO SECO, HORARIA (°C)': 'temp\_atual',

    'UMIDADE RELATIVA DO AR, HORARIA (%)': 'umidade',

    'PRESSÃO ATMOSFERICA MAX.NA HORA ANT. (AUT) (mB)': 'pressao\_max',

    'PRESSÃO ATMOSFERICA MIN. NA HORA ANT. (AUT) (mB)': 'pressao\_min'

})

df.replace(['////', '///', '-9999'], pd.NA, inplace=True)

for col in ['temp\_max', 'temp\_atual', 'umidade', 'pressao\_max', 'pressao\_min']:

    df[col] = df[col].astype(str).str.replace(',', '.')

    df[col] = pd.to\_numeric(df[col], errors='coerce')

df['pressao'] = (df['pressao\_max'] + df['pressao\_min']) / 2

df.dropna(subset=['temp\_max', 'temp\_atual', 'umidade', 'pressao'], inplace=True)

# 3. Treinar modelo com 3 variáveis

X = df[['temp\_atual', 'umidade', 'pressao']]

y = df['temp\_max']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

modelo = RandomForestRegressor(n\_estimators=100, random\_state=42)

modelo.fit(X\_train, y\_train)

# 4. Avaliação

y\_pred = modelo.predict(X\_test)

mae = mean\_absolute\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Erro médio absoluto: {mae:.2f} °C")

print(f"Coeficiente de determinação (R²): {r2:.2f}")

# 5. Salvar modelo

joblib.dump(modelo, "modelo\_previsao\_amanha.pkl")