

**UNIVERSIDADE DO MINDELO**

**DEPARTAMENTO DE ENGENHARIA E CIÊNCIAS DO MAR**

**CURSO DE LICENCIATURA EM**

**ENGENHARIA INFORMÁTICA E SISTEMA COPUTACIONAIS**

**RELATORIO**

**Mindelo, 2023**

**Tema: Controle de temperatura e Humidade em Ambientes Críticos Computacionais usando (ESP8266) & WhatsApp**

**Autor:Anifa Pinheiro N.º 5062**

**Professor: Frederico Soares**

**universidadE dO MINDELo**

**DEPARTAMENTO de ENGENHARIA E CIÊNCIAS DO MAR**

**curso de licenciatura em**

**ENGENHARIA INFORMÁTICA E SISTEMAS COMPUTACIONAIS**

**Relatorio**

**Tema: Controle de temperatura e Humidade em Ambientes Críticos Computacionais usando (ESP8266) & WhatsApp**

**Autor: Anifa Pinheiro, Nº 5062**

**Professor: Frederico Soares**

**Mindelo, 2023**

**Resumo**

**O presente trabalho tem como objetivo o monitoramento das grandezas de temperatura e umidade relativa em ambientes computacionais críticos, semelhante a data centers, e identificar a presença de pessoa no ambiente. Através de um sistema para monitoramento de variáveis ambientais é possível tomadas de decisão mais assertivas. Este projeto conta com pesquisas nas áreas de API e em Internet das Coisas para o desenvolvimento da solução. O sistema desenvolvido utiliza ferramentas open source e de baixo custo, possui precisão dos dados monitorados e opera em tempo real, permitindo reação rápida e assertiva às variações no ambiente.**

**Palavras-chave: Temperatura. Umidade Relativa. Presença. Internet das Coisas. API.**

**Introdução**

A IoT pode ser definida como uma rede aberta que inclui dispositivos inteligentes conectados pela Internet e que têm a capacidade de se auto-organizar para compartilhar informações, dados e recursos, e interagir com o ambiente (MADAKAM et al., 2015).

Um ambiente computacional crítico é aquele que oferece serviço de Tecnologia da Informação (TI) de forma ininterrupta e contém servidores e computadores em geral Para que esses equipamentos não sofram perda de desempenho e nem

tenham sua integridade comprometida, necessita-se que um conjunto de fatores estejam

alinhados. A norma ANSI/TIA-942 (Telecommunications Infrastructure Standard for Data Center) orienta o projeto e a instalação de um data center. Uma das orientações refere-se ao controle da temperatura que deve estar entre 20° e 25° C e da umidade entre 40% e 55% (ANSI/TIA, 2012). O controle de umidade visa equilibrar a quantidade de vapor d’água naturalmente presente no ar, para que não fique nem tão seco nem tão úmido. O objetivo do projeto é monitorar local e remotamente a temperatura, umidade.

Para isso, será necessário a execução das seguintes etapas:

* Programação de sensores de temperatura e umidade (Sketch).
* Bibliotecas necessárias para utilização do sensor DHT11 ( DHT11 Sensor Library eAdafruit Unified Sensor Lib).

O ESP8266 é uma placa de desenvolvimento com conectividade Wi-Fi, o que permite o envio de informações em tempo real para um aplicativo como o WhatsApp. Utilizando o ESP8266, é possível obter leituras precisas de temperatura e umidade através de sensores específicos, como o DHT11, o ESP8266 pode ser programado para realizar leituras periódicas dos sensores e enviar os dados para o WhatsApp através de uma conexão Wi-Fi. Assim, é possível receber notificações em tempo real sobre qualquer alteração nas condições de temperatura e umidade no ambiente crítico, além disso, é possível programar o ESP8266 para tomar ações automatizadas, como acionar um sistema de ar condicionado ou um umidificador, para regular a temperatura e umidade do ambiente. Isso é especialmente importante em ambientes onde a temperatura e umidade excessivas podem causar danos aos equipamentos.

O controle de temperatura e umidade em ambientes críticos computacionais utilizando o ESP8266 e o WhatsApp permite que os responsáveis pelo ambiente estejam constantemente informados sobre as condições, podendo agir rapidamente para evitar problemas relacionados ao superaquecimento ou à umidade excessiva.

Materiais utelizados:

1 x NODEMCU ESP8266;

1 x Cabo micro USB;

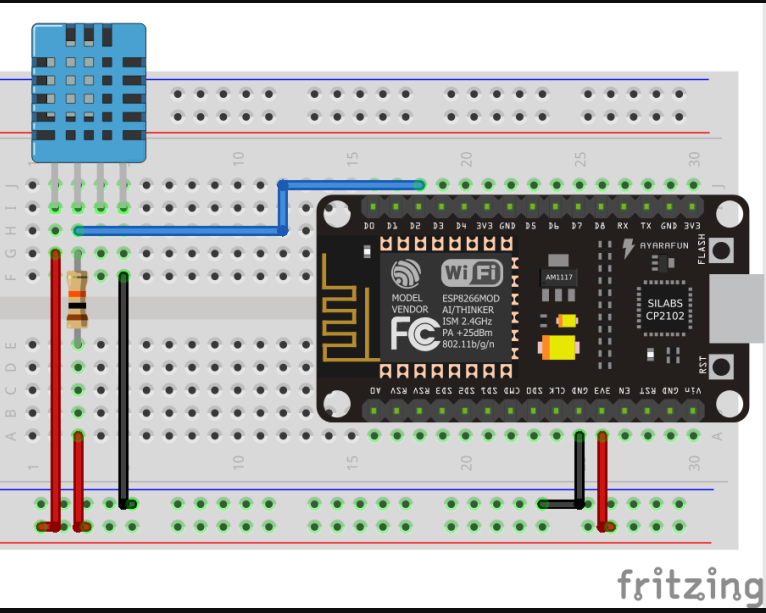
1 x Sensor DHT11;

1 x Resistor de 10 kΩ;

1 x Protoboard;

Fios jumper;

**ESQUEMÁTICA DE LIGAÇÃO DOS COMPONENTE**



Descrição:

* O sensor DHT11 possui 3 terminais. O primeiro (VCC) é o terminal de alimentação, que conforme datasheet do fabricante pode ser entre 3 V e 5,5 V.
* O segundo (DATA) é o pino de comunicação de dados, será através dele que o valor da temperatura e umidade serão comunicados.
* Por fim, o último (GND) é o terminal de terra do sensor;

O pino 1 do sensor DHT11 será ligado a alimentação de 3,3 V da placa NODEMCU ESP8266. O pino 2 deve ser conectado à porta digital D3 (GPIO4);

No pino 2 vamos conectar um resistor de 10 kΩ pull-up (resistor conectado ao VCC). O pino 3 deve ser conectado ao GND (terra).

A forma mais comum de coletar dados em IoT é usando sensores, geralmente conectados em outros objetos e posicionados em campo. Após coletar os dados eles podem ser analisados, armazenados, compartilhados ou qualquer outra ação de interesse, Porém, para compartilhar dados na rede os dados coletados, é necessário uso de protocolos que padronizam a comunicação de ponta a ponta. A IoT pode ser formada por objetos diferentes para entregar serviços específicos e por causa dessa diferença entre os objetos, é muito importante padronizar a comunicação através de um protocolo.

API para enviar mensagem whatsApp :

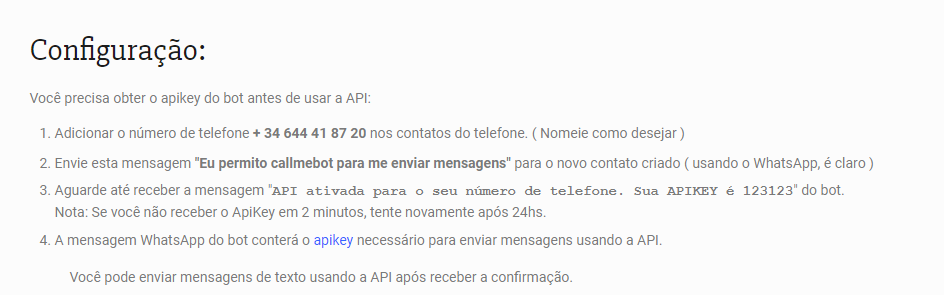


Figura configuração com WhatsApp boot

<https://www.callmebot.com/blog/free-api-whatsapp-messages/>

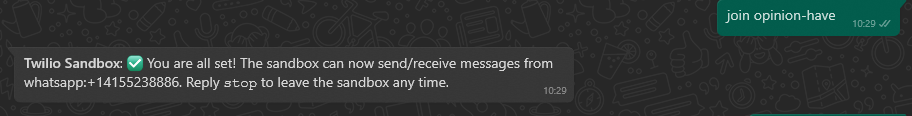


Figura Exemplo de comunicação com WhatsApp boot

Utilizei o twilio, para enviar e receber mensagens de texto e realizar outras funções de comunicação usando suas APIs de serviço da web, seguidamente ele gera um credencial que permite comunicar com o thigESp

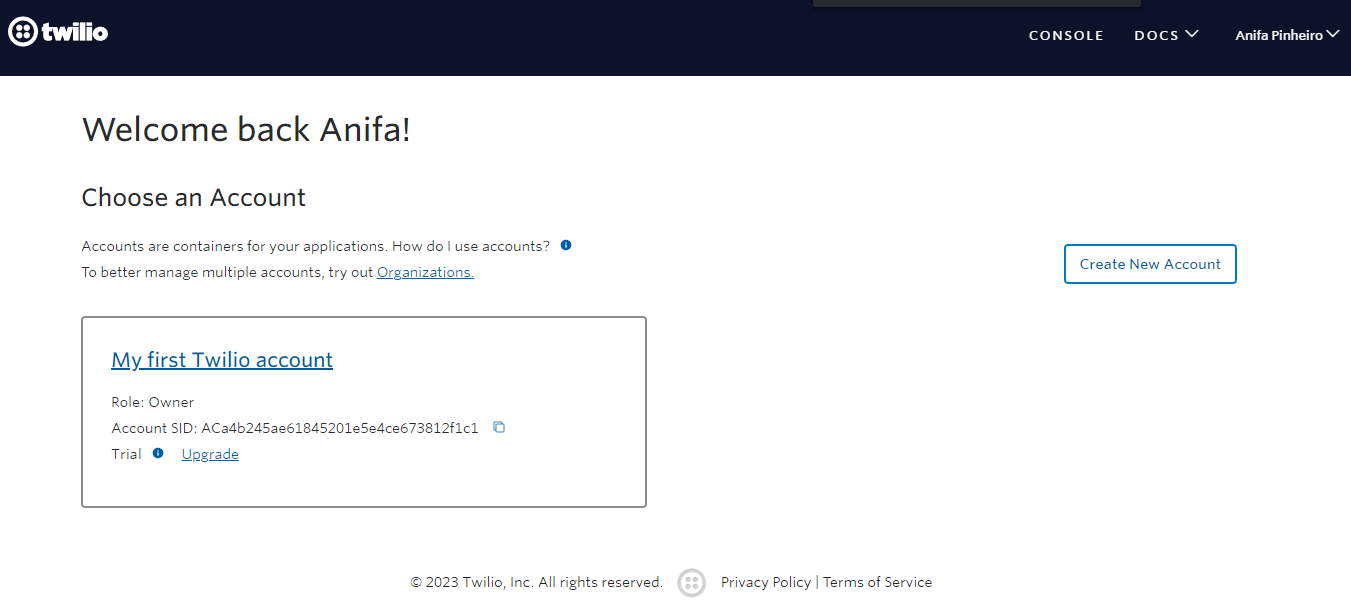


Figura Criação de containers no twilio

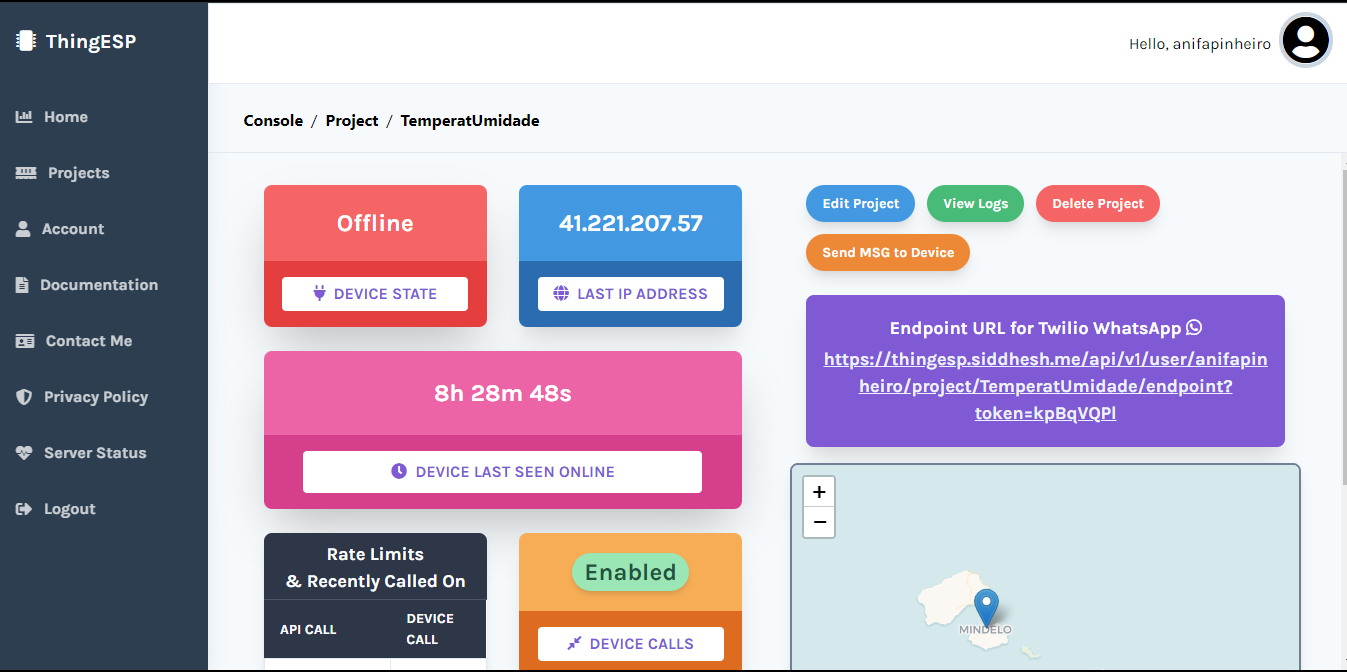
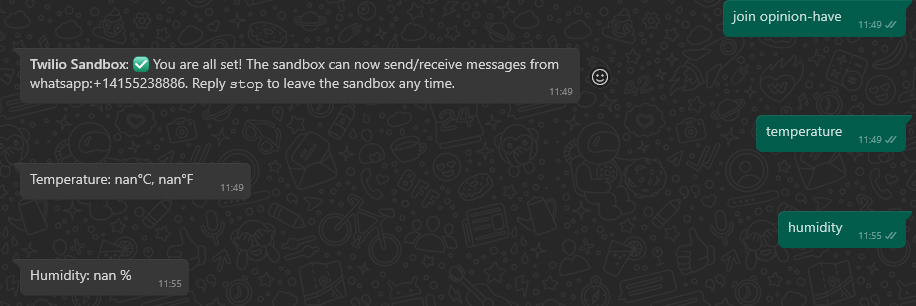


Figura configuração de comunicação ThingESP

Resultados de teste feito:



Infelismente não consegui detetar o problema de o sensor apresentar NAN(Not a Number) pretendo trabalhar nele ate o dia da apresentação.

Codigo fonte:

#include <ESP8266WiFi.h>

#include <ThingESP.h>

#include <DHT.h>

#define DHTPIN 3

#define DHTTYPE DHT11     // DHT 11

DHT dht(DHTPIN, DHTTYPE);

ThingESP8266 thing("anifapinheiro", "TemperatUmidade", "12345678anifa911011");

unsigned long previousMillis = 0;

const long INTERVAL = 6000;

void setup()

{

  Serial.begin(115200);

  thing.SetWiFi("TP\_LINK", "LINK\*2022\*");

  thing.initDevice();

}

String HandleResponse(String query)

{

 float h = dht.readHumidity();

  // ler temperatura  Celsius ( default)

  float t = dht.readTemperature();

  // ler temperatura Fahrenheit (isFahrenheit = true)

  float f = dht.readTemperature(true);

  Serial.println("Temperature :");

  Serial.print(t);

  Serial.println("Humidity");

  Serial.print(h);

  String humid = "Humidity: " + String(h) + " % \n";

  String temp = "Temperature: " + String(t) + "°C, " + String(f) + "°F \n";

  if (query == "temperature") {

    return temp;

  }

  else if (query == "humidity") {

    return humid;

  }

  else return "Conta invalida..";

}

void loop()

{

  thing.Handle();

}

Referencias:

[https://www.circuitar.com.br/tutoriais/configurando-o-arduino-no-windows/#4-instalar-os-drivers](https://www.circuitar.com.br/tutoriais/configurando-o-arduino-no-windows/%234-instalar-os-drivers)

<https://iotdesignpro.com/projects/temperature-humidity-monitoring-over-thingspeak-using-arduino-esp8266>

<https://bitbucket.org/MarcoRabelo/esp8266/src/master/>

<https://www.callmebot.com/blog/free-api-whatsapp-messages/>

<https://www.blogdarobotica.com/2020/08/11/monitor-de-temperatura-e-umidade-com-nodemcu-esp8266-como-servidor-web/>