

# Alarma de Variación de Temperatura para un Data Center

## (Julio de 2021)

Autor, Mateo Arenas Rivera  
Fundación universitaria del área andina  
Marenas13@estudiantes.areandina.edu.co

**Resumen** - Por medio de la siguiente actividad se procederá a aplicar el conocimiento y herramientas aprendidas en el diplomado de Python - devnet para lo cual se propone una solución electrónica con el fin de supervisar el nivel de humedad de un data center en una institución educativa, evitando posibles afectaciones en el correcto funcionamiento de los servidores y la misión institucional por sobrecalentamiento de los equipos de cómputo.

**abstract** - Through the following activity, the knowledge and tools learned in the Python - devnet diploma will be applied in an electronic solution for the supervision of the temperature and humidity level of a data center in an educational institution, avoiding possible effects on the correct operation of the servers and the institutional mission due to overheating of computer equipment.

**Palabras Claves** – Applets, Bot, Data Center, Esp32

### I. INTRODUCCIÓN

El adecuado manejo de la temperatura en los equipos informáticos permite alargar su tiempo de vida, aprovechando al máximo sus prestaciones antes que empiecen a presentar fallos por el constante uso, un inadecuado control de la temperatura reduce el tiempo de vida de los componentes como los discos duros, procesadores y fuentes de poder siendo estos los más afectados por sobrecalentamiento, ocasionando el daño de las partes y la afectación del servicio informático.

La institución educativa donde se desea aplicar el proyecto cuenta con un data center donde reposan los servidores que manejan el Core del negocio, el cual tiene un aire acondicionado funcionando 24 horas de día y los 7 días de la semana, en ocasiones se han presentado fallos en el sistema de refrigeración por causas externas como la pérdida del flujo eléctrico cuando no se enciende la planta de energía de respaldo o fallas propias del equipo de refrigeración, afectando los equipos informáticos.

Aplicando el conocimiento adquirido en el diplomado se plantea una solución electrónica utilizando el módulo de desarrollo esp32 y el sensor de temperatura DHT11, los cuales interactúan con el lenguaje Python para la interpretación de los valores recibidos en el sensor y en caso de presentar un valor por fuera del rango programado se le notificara a la persona responsable del data center por medio de cuatro canales de comunicación.

El primer medio de alerta es directamente en el módulo desarrollado, donde por medio de un arreglo de leds y una pantalla oled se podrá visualizar si la temperatura y humedad se encuentran fuera del rango adecuado.

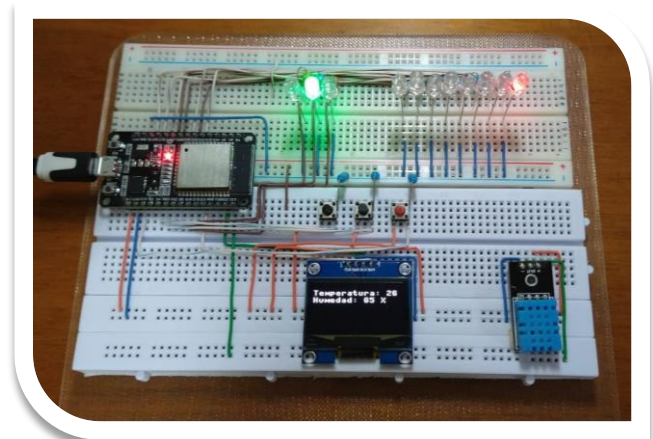


Fig 1. Arreglo de leds y pantalla oled

Como segundo medio de alerta se utilizan los Applets proporcionados por la página web IFTTT <https://www.ifttt.com> los cuales son pequeñas aplicaciones que cumplen una función determinada, una de esas funciones es ejecuta un disparador para el envío de un correo electrónico previamente configurado en la página web.

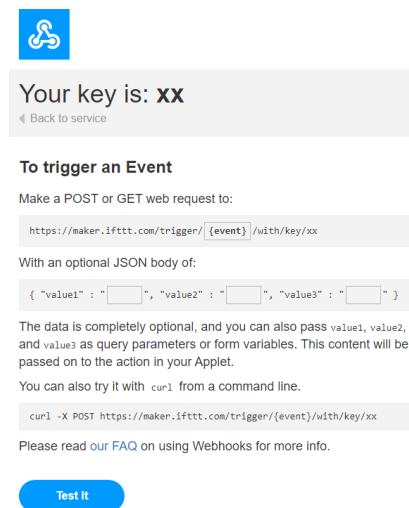


Fig 2. Configuración web request

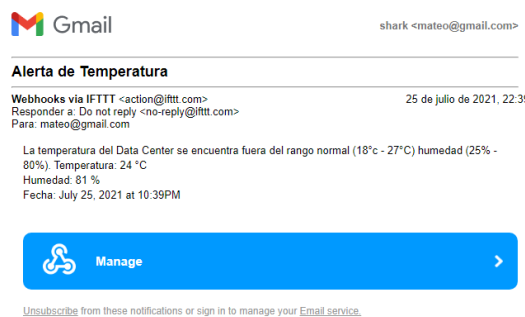
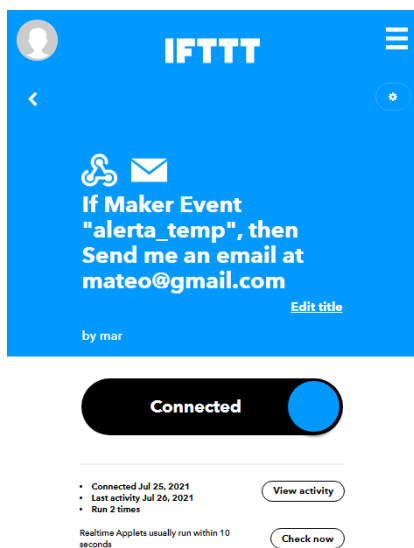


Fig 4. Email de alerta

Para el tercer medio de alerta se utiliza el applet de notificaciones para celulares, el cual despliega el mensaje configurado en la barra de notificaciones, la app IFTTT debe estar previamente instalada.



Fig 5. Notificación en el celular

El cuarto medio de alerta integra un applet y un bot proporcionado por la página web <https://callmebot.com> el cual permite ejecutar un disparador en IFTTT y posteriormente realizar un web request para el envío de un mensaje de alerta al chat de WhatsApp previamente registrado.

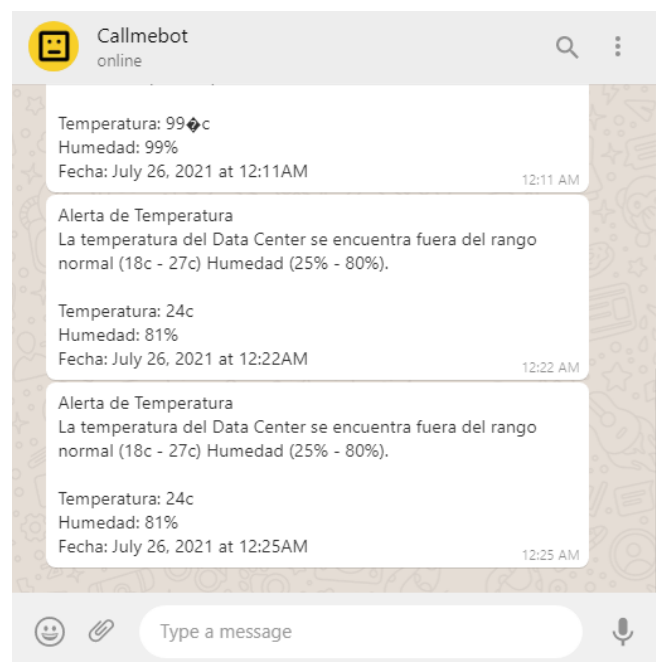


Fig 6. Alerta enviada por whatsapp

## II. MANUAL DE USUARIO

### A. Distribución de las zonas de notificación

El dispositivo electrónico se ha dividido en tres zonas de notificaciones las cuales se explicarán a continuación.

#### 1. Leds Indicadores de Humedad



Fig 7. Indicadores de humedad

En el área superior izquierda se encuentran ubicados tres leds de color azul, verde y amarillo los cuales dependiendo de los valores capturados de humedad se encienden.

A continuación, se presenta el rango en el cual se encienden los leds indicadores.

HUMEDAD	COLOR LED
< 25 %	Blue
[ 25 % : 80 % ]	Green
> 80 %	Yellow

Tabla 1. Rango de Humedad

#### 2. Leds indicadores de Temperatura



Fig 8. Indicadores de Temperatura

En el área superior derecha se presenta el arreglo de leds indicadores para la temperatura, los cuales se encienden según la configuración de la siguiente tabla.

TEMPERATURA	COLOR LED
< 18°C	Blue
18°C	Blue
19°C	Blue
20°C	Blue
21°C	Green
22°C	Green
23°C	Yellow
24°C	Yellow
25°C	Orange
26°C	Orange
27°C	Orange
> 27°C	Orange

Tabla 2. Rango de Temperatura

#### 3. Display oled de notificaciones

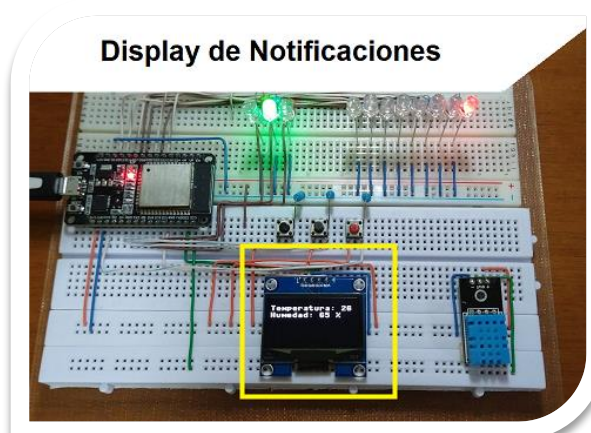


Fig. 9 Display de Notificaciones

Para la zona inferior central se encuentra ubicado el display oled de notificaciones en el cual constantemente se visualiza los valores capturados por el sensor DHT11 de temperatura y humedad.

### B. Configuración inicial

Para el correcto funcionamiento de la alarma y el envío de las notificaciones se deben de tener en cuenta los siguientes puntos.

1. El dispositivo esp32 requiere para su funcionamiento una alimentación de voltaje de 5v por medio de un puerto micro usb.
2. Antes de instalar el dispositivo en el área de trabajo se debe de configurar los parámetros de la red wifi en el dispositivo ESP32.

- La cuenta de correo donde llegara la alerta debe de estar previamente registrada en el applet de la página web IFTTT .

<https://ifttt.com/>

- Para recibir la alerta en la barra de notificaciones del dispositivo celular se debe de descargar el aplicativo IFTTT en la siguiente dirección web.

<https://play.google.com/store/apps/details?id=com.ifttt.ifttt>

- El nro. celular donde se recibe la alerta por medio del aplicativo WhatsApp debe de estar previamente registrado en el bot callmebot.

<https://www.callmebot.com/blog/whatsapp-messages-from-ifttt/>

luego de realizadas estas configuraciones iniciales el dispositivo está listo para funcionar.

### III. MANUAL DEL PROGRAMADOR

#### A. Librerías Utilizadas para el desarrollo del proyecto

```
from machine import Pin, SoftI2C
from dht import DHT11
import sh1106 # libreria para utilizar el display de 128 x 64
import utime
import network
import urequests as requests
```

- Machine: Encargada de los puertos del ESP32
- Dht: Permite el uso del sensor DHT11
- Sh1106: Maneja el modulo oled de 128 x 64
- Utime: funciones para la implementación de las pausas
- Network. Permite la conexión wifi
- Urequest: funciones para el uso de web request

#### B. Inicialización de los pines del ESP32

```
# Leds Temperatura
led_t1 = Pin(32,Pin.OUT)
led_t2 = Pin(33,Pin.OUT)
led_t3 = Pin(25,Pin.OUT)
led_t4 = Pin(26,Pin.OUT)
led_t5 = Pin(27,Pin.OUT)
led_t6 = Pin(14,Pin.OUT)
led_t7 = Pin(12,Pin.OUT)
led_t8 = Pin(13,Pin.OUT)

# Leds Humedad
led_h1 = Pin(5,Pin.OUT)
led_h2 = Pin(18,Pin.OUT)
led_h3 = Pin(19,Pin.OUT)
```

#### C. Inicialización de la pantalla oled

```
# inicializa la pantalla oled
i2c = SoftI2C(scl=Pin(22), sda=Pin(21), freq=400000)
oled = sh1106.SH1106_I2C(128, 64, i2c, Pin(16), 0x3c)
oled.sleep(False)
oled.fill(0)
oled.contrast(0xFF)
oled.flip()
```

#### D. Configuración de la red wifi

```
# inicializa la red wifi
wifi_ssid = "xxxxxxxx"
wifi_password = "xxxxxxxxxx"
```

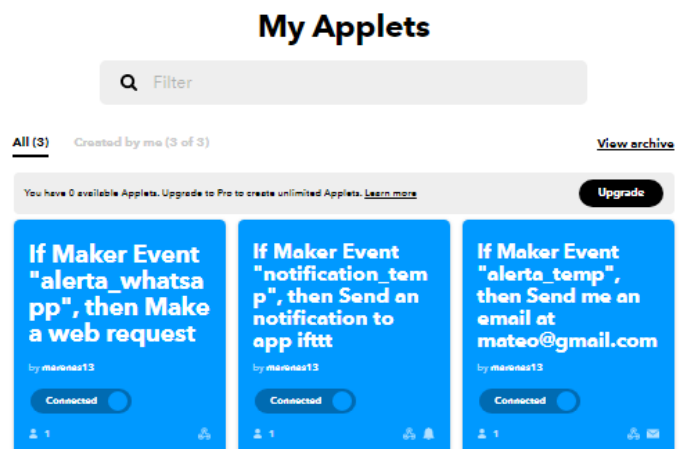
Los valores de las variables se remplazan con el nombre de la red wifi (wifi\_ssid) y la clave de acceso (wifi\_password).

#### E. Url de los applets configurados en IFTTT

```
url_alerta_temp = "https://maker.ifttt.com/trigger/alerta_temp/with/key/x"
url_notif_temp = "https://maker.ifttt.com/trigger/notification_temp/with/key/x"
url_whatsapp = "https://maker.ifttt.com/trigger/alerta_whatsapp/with/key/x"
```

Cada applets creado cuenta con una url diferente que activa el disparador del evento configurado en la web IFTTT para desplegar las alertas de temperatura y humedad.

los Applets utilizados para el proyecto son los siguientes:



- If Maker Event “alerta whatsapp” then Make a Web request.**

Al ejecutarse el disparador realiza un web request a la página callmebot.com la cual permite enviar mensajes de whatsapp.



## 2. If Maker Event “notification\_temp” then send an notification to app ifttt.

Cuando se ejecuta envía una alerta a la barra de notificaciones para un teléfono celular que tenga instalada la app IFTTT.

## 3. If Maker Event “alerta temp” then send me an email.

Al ejecutarse envía un correo electrónico a la cuenta establecida.

### F. Despliegue de la temperatura en la pantalla oled

```
# muestra en la pantalla la medición
oled.fill(0)
dh11.measure()
t = dh11.temperature()
h = dh11.humidity()
oled.text("Temperatura: " + str(t) + " c",0,10) # Temperatura
oled.text("Humedad: " + str(h) + " %",0,20) # Humedad
oled.show()
```

### G. Verifica los niveles de humedad y muestra el rango por medio de los leds.

```
# verifica los niveles de humedad
if h < 25 : # humedad muy baja
    led_h1.on()
    alerta = 1
elif h > 80: # humedad muy alta
    led_h3.on()
    alerta = 1
else:
    led_h2.on()
```

### H. Verifica la temperatura y muestra el rango por medio de los leds.

```
# verifica la temperatura
if t < 18 :
    led_t8.on()
    alerta = 1
elif t == 18 or t == 19 :
    led_t8.on()
elif t == 20 :
    led_t7.on()
elif t == 21 :
    led_t6.on()
elif t == 22 :
    led_t5.on()
elif t == 23 :
    led_t4.on()
elif t == 24 :
    led_t3.on()
elif t == 25 :
    led_t2.on()
elif t == 26 or t == 27 :
    led_t1.on()
elif t > 27 :
    led_t1.on()
    alerta = 1
```

I. Si la temperatura sobrepasa el rango programado realiza el web request para enviar la alerta por medio del whatsapp.

```
if alerta == 1 :
    url_1 = url_whatsapp + "?value1=" + "Alerta%20d
e%20Temperatura%0A%20temperatura%20del%20Data%20Center%20se%20e
ncuentra%20fuera%20del%20rango%20normal%20%2818c%20-%2027c%29%
20Humedad%20%2825%25%20-%2080%25%29.%0A%0ATemperatura%3A%20" +
    str(t) + "c%0AHumedad%3A%20" + str(h) + "%25%0AFecha%3A"
    try:
        r = requests.get(url_1)
        print(r.text)
    except Exception as e1:
        print(e1)
```

J. El segundo web request activa el envío de correo electrónico

```
url_2 = url_alerta_temp + "?value1=" + str(t) + "&value2=" + str(h)
try:
    r2 = requests.get(url_2)
    print(r2.text)
except Exception as e2:
    print(e2)
```

K. El ultimo Web request activa el envío de la alerta de temperatura por medio de la barra de notificaciones.

```
try:
    r3 = requests.get(url_3)
    print(r3.text)
except Exception as e3:
    print(e3)
```

## IV. REFERENCIAS

[1] Radomir. Dopieralski, “MicroPython SH1106 OLED driver, I2C and SPI interfaces” <https://github.com/robert-hh/SH1106/blob/master/sh1106.py> (consultada 27/07/2021).

[2] Bot de Whatsapp “callmebot” <https://www.callmebot.com/> (Consultada 27/07/2021)

[3] Creación de Applets “if this, then that” <https://ifttt.com/> (Consultada 27/07/2021)