

# Taller de Programación y Robótica en CMM BML 2023T2 – CL21s - 8 Noviembre 2023

[R] Conexión **Pico W modo STA** - **Web server local**  
- 1ros pasos#4 – 30'

1. [R] Modos de conexión de un uC y capas OSI
2. [R] Pico W modo STA - servidor TCP/IP con sockets
  - **Web botones para control Led + información cambiante:** temperatura Pico-5'

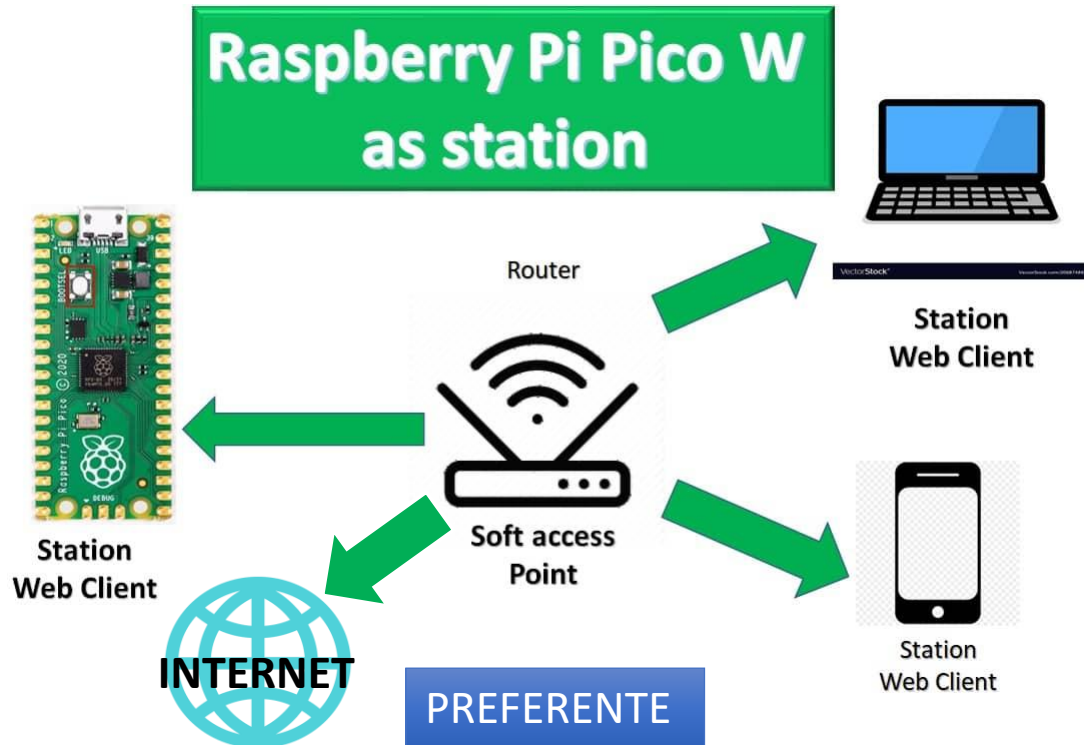
[R] Conexión **Pico W modo STA** - **Web server global con Adafruit IoT** – 45'



Voluntario : J.C. Santamaria

# Clase 20s.1.1 – [R] Básico de conexión wifi

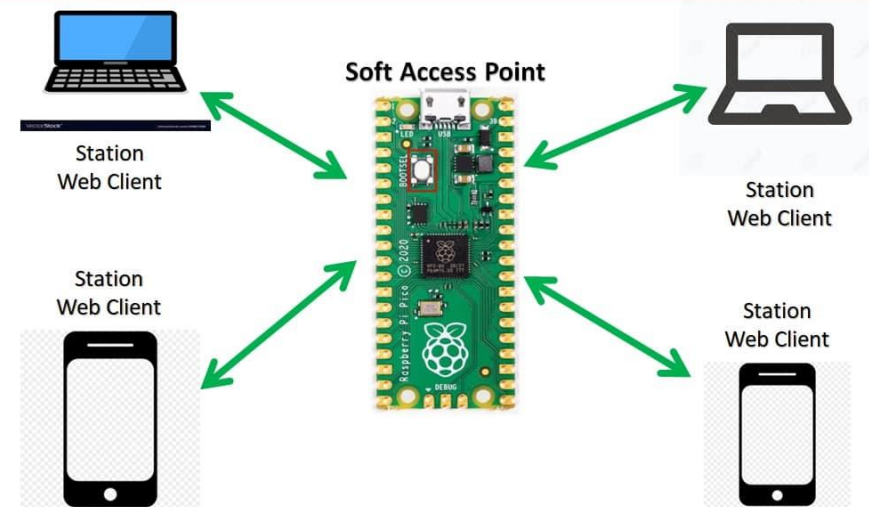
RECORDAR



Raspberry Pi Pico W se conecta a la red Wi-Fi a través de un router WIFI. Otros dispositivos conectados al mismo router (misma red local SSID) pueden “hablar” entre ellos → el router asigna una dirección IP interna a cada uno incluida la Pico W. Además, **la Pico W en este modo puede acceder a Internet** (acceder desde internet a la pico w es más complicado)

En este modo, Raspberry Pi Pico W crea su propia red Wi-Fi inalámbrica similar a un router Wi-Fi. Cualquier dispositivo, con la clave, puede conectarse a esa red WIFI, pero **no tendrá acceso a Internet**, solo a la Pico W y a otros dispositivos conectados. **Hasta 5 dispositivos pueden estar conectarse a esta red Wi-Fi privada**, cada uno con su IP de red privada

## Pico W in Soft Access Point Mode



RECORDAR

# Clase 21s.1.1 -Aclararse un poco .... niveles OSI (antes modelo TCP/IP)

El **modelo TCP/IP** es una explicación de protocolos de red creado por Vinton Cerf y Robert E. Kahn, en la década de 1970, implantado en ARPANET ver CL17

Lo veremos en algún momento de este curso

**HTTP + DNS**

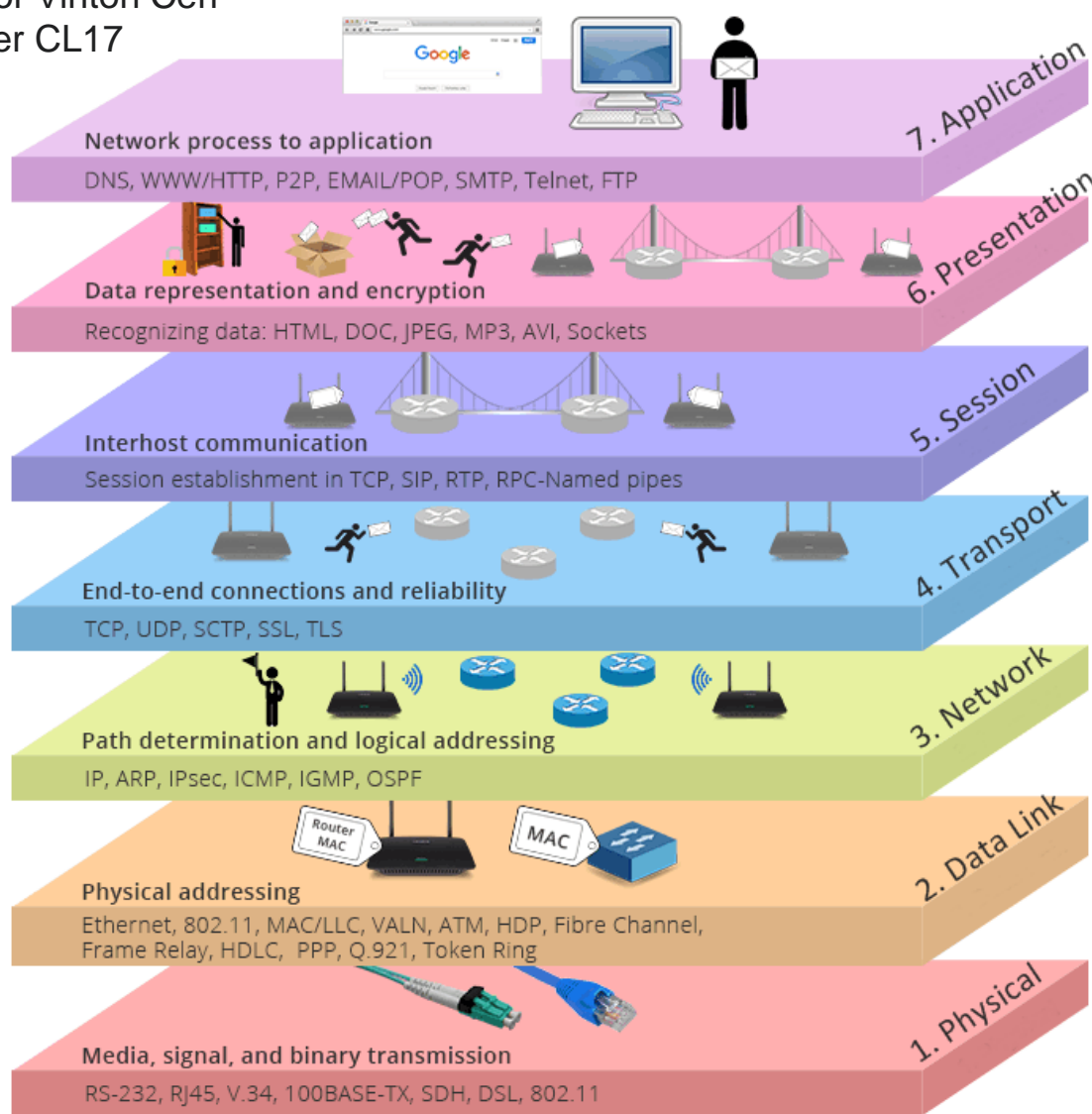
**HTML + Sockets**

**TCP + (SSL y TLS en protocolos con seguridad )**

**IP : direcciones IPV4 e IPV6**

**Ethernet / wifi + direcciones MAC**

**Cables RJ45 / wifi = 802.11**



## Clase 15 – [R-HW] Sensor NTC

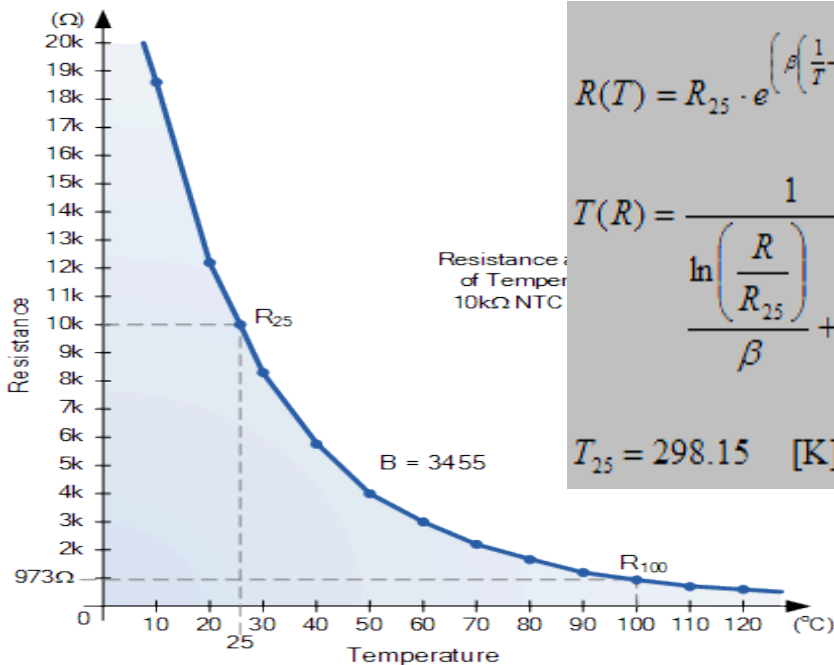
Ref info NTCs <https://youtu.be/obLccs8dRdg>

BMMR CL15 ADC NTC 1 0

Hay algunas calibraciones debidas a problemas con ADC de Pico

### ¿Cómo funciona un NTC?

Es una resistencia que cambia de valor con la temperatura. “N” significa que lo hace de forma inversa y de forma exponencial



$$R(T) = R_{25} \cdot e^{\left(\beta \left(\frac{1}{T} - \frac{1}{T_{25}}\right)\right)} \quad [\Omega]$$

$$T(R) = \frac{1}{\frac{\ln\left(\frac{R}{R_{25}}\right)}{\beta} + \frac{1}{T_{25}}} \quad [K]$$

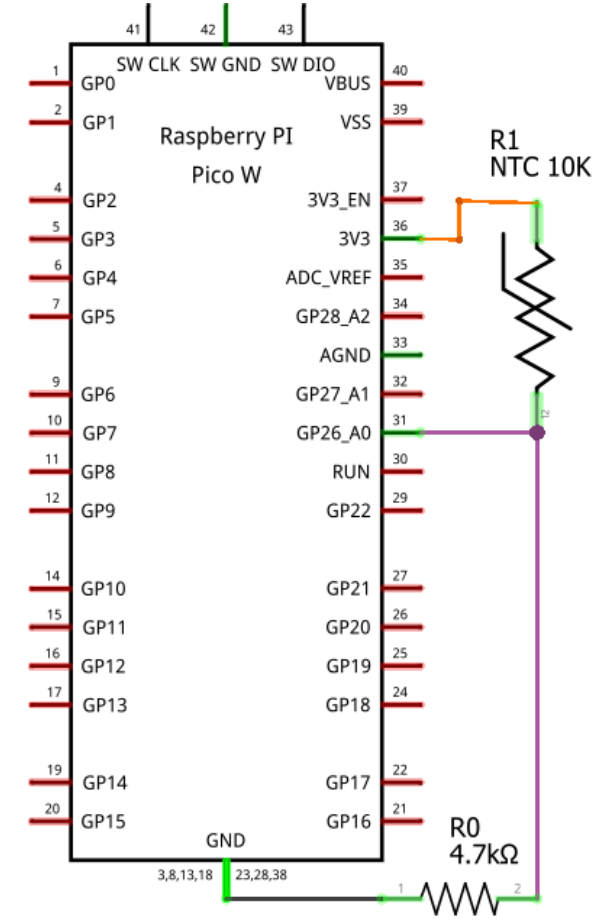
$T_{25} = 298.15 \quad [K]$

### ¿Como trabajar con NTCs?

Se suele usar un circuito divisor de tensión, con el NTC y otra resistencia de baja tolerancia => **esto es equivalente al montaje con un potenciómetro que ya hicimos así que clonamos el programa.** Pasos

1. Medir voltaje en punto intermedio
2. Calcular resistencia NTC actual
3. Con el valor Beta y la resistencia nominal a 25°C (R0) usamos la formula (más simple) de calcular la Temperatura

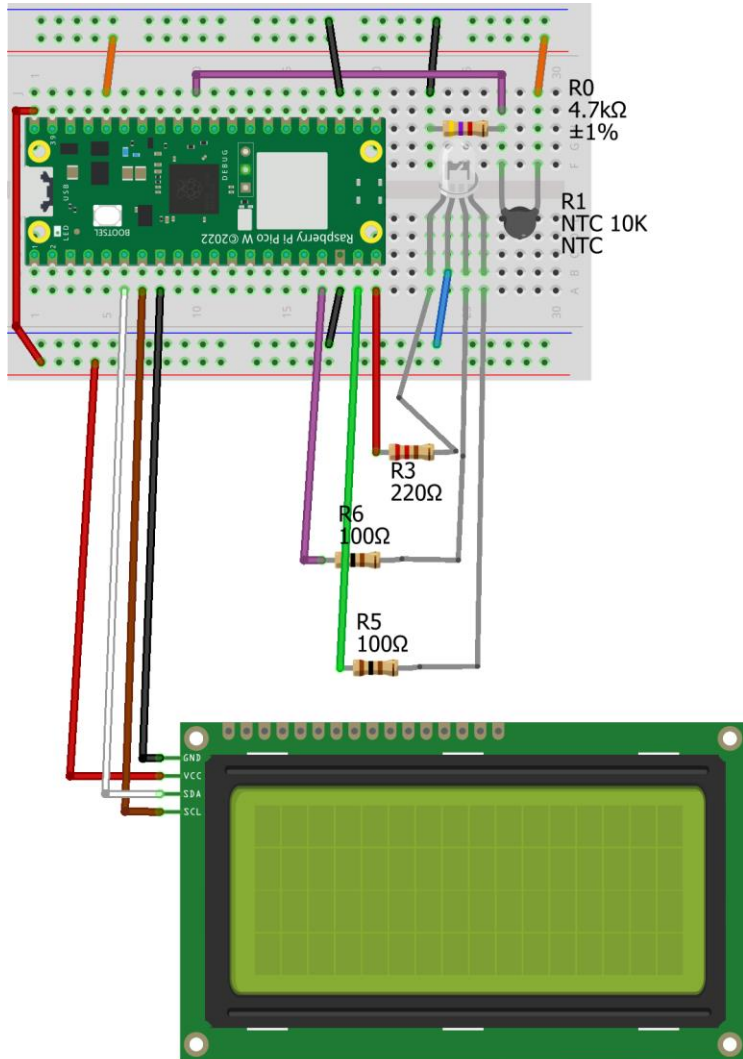
[Calculadora Resistencia NTC](#)





# Clase 21s.1.2 [R]- Pico W modo STA – Ejemplo servidor **LOCAL** web TCP/IP

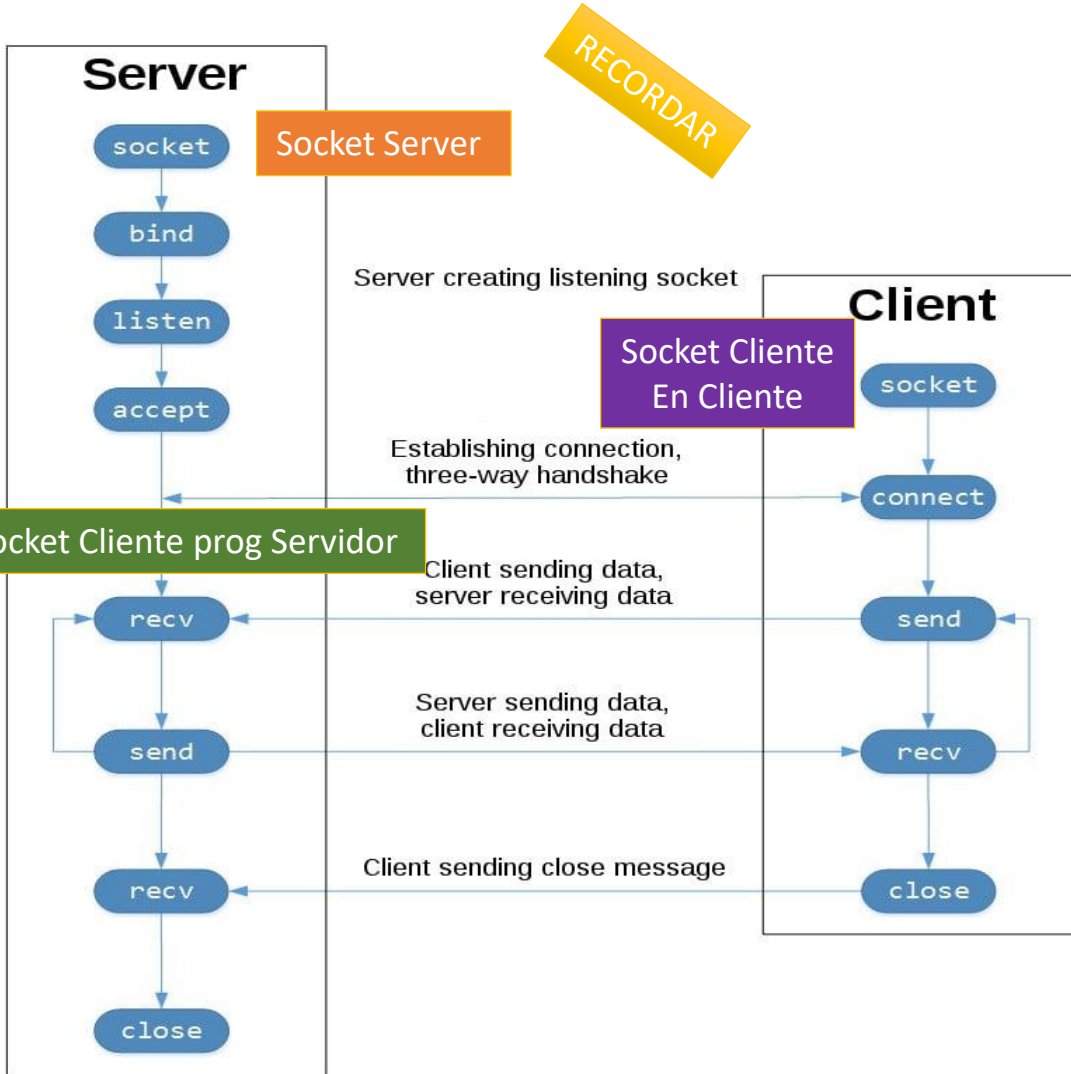
BMMR\_CL21s\_webServ\_NTC\_RGB\_2\_1.py



- # 0- Crea objetos HW : 0.1 - Crea el objeto ADC para el sensor NTC
- # 0.2- Crea el objeto rgb led y como es cátodo común lo pone a (-): red:15 / green:13/ blue:14
- # 0.3-Crea objeto LCD i2c : # I2C0 I2C(0, scl=Pin(5), sda=Pin(4), freq=400\_000)
- # 1- (Programa principal) - Presentación en LCD
- # Definición de Funciones: las funciones necesitan que los objetos como LED RGB o LCD estén creados.
- # F.1 – Temperatura NTC-> temperatura float
- # F.2 – Web (temperatura, string led estado -> crea string html
- # F.3 – Open Socket (ip) : abre socket servidor
- # F.4 – Serve (Socket servidor) : maneja peticiones cliente, Bucle infinito
- llama a Temperatura y Web
- # 2 – Bucle principal Try -Except
  - # 2.1 Intenta conexión
  - # 2.2 Si conexión Ok , Crea socket server con Open socket
  - #2.3 Servidor web con Serve
  - #2.4 Except interrupción teclado : borra display & reset Pico W



# Clase 21s.1.3– [R]- Pico W modo STA – Ej. servidor LOCAL web TCP/IP - Ejecución



```
Consola x
MicroPython v1.20.0-528-gf8e58553c on 2023-09-29; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

Microcontroller: Pico W only - Key other HW : NTC on ADC0 + RGB 4pin led common catode, r pin15, g
pin13, b pin 14 + LCD i2c 20x4
Program: Web Server STA mode - control REGB led & NTC show Temp - Version: 2.1
CONNECTED-STAT_Connected
192.168.1.43
<socket state=1 timeout=-1 incoming=0 off=0>|
/
/red?

http://192.168.1.43/red?
```

## Control de Ledx3 y visualiza temperatura-CMM Benito Martin Makers

Presiona uno de los 4 botones para cambiar el color del led

- red
- green
- blue
- off

Temperatura es de 18.46 grados Celsius

Estado del led RGB: RED on, resto off



# Clase 21s.2.1– [R]- Pico W modo STA – Ejemplo servidor web Global –con servicio **Adafruit IoT**

El principal problema del ejemplo servidor Web LOCAL para controlar un led RGB + NTC, es que **solo se puede usar dentro de la red local wifi.**

La solución (más segura) para acceder desde fuera de la red local es usar un servicio de IoT . Hay varios gratuitos

## - **Adafruit IO**

- Thingspeak : muy buena para datos de sensores
- **Blynk** : muy popular
- Cayene
- HiveMQ (bróker MQTT con tutorial en sunfounder)....

**En este proyecto vamos a usar **Adafruit IO**,**

- que permite usar HTTP (más sencillo que usar un bróker MQTT, de momento)
  - Librería requests, que ya hemos usado con acciones GET y POST
- que permite controlar HW, no solo enviar datos de sensores
- Tiene un plan gratuito suficiente
- **Tiene un “lag” considerable en algunos botones**

## **Pasos en la plataforma Adafruit IoT**

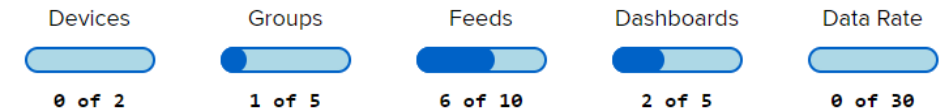
(es parecido en otras herramientas)

Tutorial en castellano, ver el hasta que cuenta el código, que es para un neopixel

[Raspberry Pi Pico W con Adafruit IO control de Neopixel - Proyecto IOT- Código en MicroPython](#)

1. Crear un usuario
2. Suscribirse plan básico. El plan básico es Gratis: permite crear hasta 10 feeds y 5 Dashboards. Los feeds son el elemento limitante, feed = elemento hw básico , por ejemplo, led de 1 color

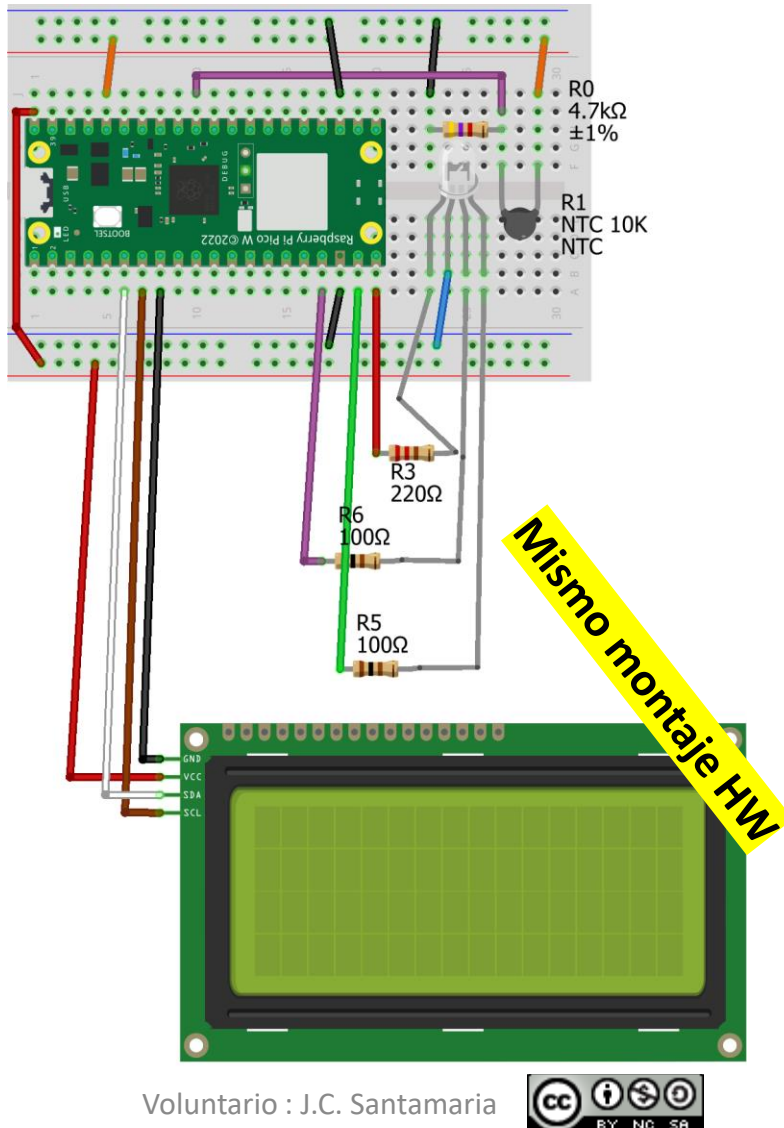
### Account Status



3. Crear los “feeds” que necesites en función del proyecto, en nuestro proyecto serán 4 = 3 para los 3 LED RGB + 1 temperatura NTC
4. Componer un Dashboard con los “feeds”, eligiendo el tipo de control o visualizador on/of, slide, aguja, etc
5. Copiar **clave, nombre de usuario y nombres de feeds**, que necesitaremos en nuestro programa

# Clase 21s.2.2– [R]- Pico W modo STA - servidor web **Global** - [Adafruit IoT](https://adafruit.io)

BMMR\_CL21s\_AfIoT\_NTC\_RGB\_2\_0.py



# 0- Crea objetos HW + ESPERA = 60 segundos

# 0.1 - Crea el objeto ADC para el sensor NTC

# 0.2- Crea el objeto rgb led y como es cátodo común lo pone a (-): red:15 / green:13/ blue:14

# RGBled representa el estado del RGB, las claves son los nombres de los feed de Adafruit

# 0.3 - Constantes para Adafruit IoT 'AFIOT\_USERNAME' 'AFIOT\_KEY' en secrets module

# 0. 4-Crea objeto LCD i2c : # I2C0 I2C(0, scl=Pin(5), sda=Pin(4), freq=400\_000)

# Definición de Funciones: las funciones necesitan que los objetos como LED RGB o LCD estén creados.

# F.1 – Temperatura NTC-> temperatura float

# F.2 – Compone URL -> url string para requests GET y POST

# 1 Bucle principal - Presentación

# 2 – Bucle principal Try -Except

# 2.1 Intenta conexión

# 2.2 Leer controles HW de Adafruit (GET) & Actualizar HW conectado a PICO

# 2.3 Escribir Sensores en Adafruitt (POST)

# 2.4 Espera con bucle que escribe puntos seguidos

# 3- Except interrupción teclado : borra display & reset Pico W



# Clase 21s.2.3– [R]- Pico W modo STA - servidor web **Global - Adafruit IoT** - Ejecución

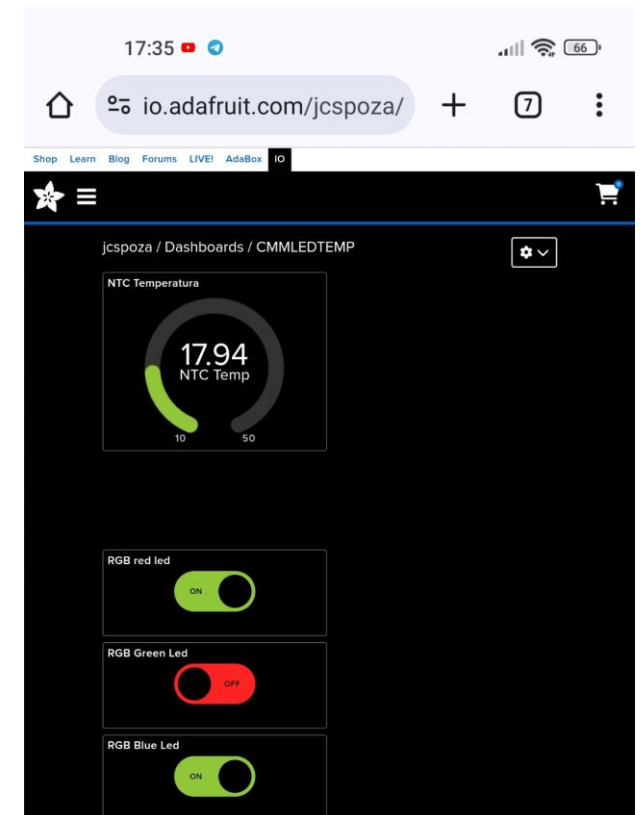
```
Consola x
Process ended with exit code 1.

MicroPython v1.20.0-528-gf5e5853c on 2023-09-29; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

Microcontroller: Pico W only - Key other HW : NTC on ADC0 + RGB 4pin led common catode, r pin15, g pin13, b
pin 14 + LCD i2c 20x4
Program: HW contorl example w. Adafruit IoT - control RGB led & NTC show Temp - Version: 1.0
CONNECTED-STAT_Connected
192.168.1.43
rgbred ON
rgbgreen OFF
rgbbblue ON
{'rgbred': 'ON', 'rgbgreen': 'OFF', 'rgbbblue': 'ON'}
Temperatura =17.94 C
{"id":"0FEHVE00BTHNQ9GKTM4ZWCPMGX","value":"17.94","feed_id":661509,"feed_key":"temperatura","created_at":"2
023-11-06T16:35:01Z","created_epoch":1699288501,"expiration":"2023-12-06T16:35:01Z"}
.....New round
rgbred ON
rgbgreen ON
rgbbblue ON
{'rgbred': 'ON', 'rgbgreen': 'ON', 'rgbbblue': 'ON'}
Temperatura =19.69 C
{"id":"0FEHVEN1GRH4QXHQBAV1ZZWS4Y","value":"19.69","feed_id":661509,"feed_key":"temperatura","created_at":"2
023-11-06T16:36:10Z","created_epoch":1699288570,"expiration":"2023-12-06T16:36:10Z"}
```

En la consola se van mostrando mensajes de debug, que habría que no mostrar en versión definitiva.

En LCD nuestro: proyecto + versión/ ip / estado leds / Temperatura



Pantallazo de Dashboard  
en pantalla de movil

# Archivo de transparencias relevantes

Taller personalizado de Programación  
y Robótica en CMM BML 2023 -2024



# [Redes] Socket Bind : un ordenador en modo servidor puede tener 2 direcciones de red

```
C:\Users\josec>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet 4:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::7631:d510:cd2c:cbf3%11
    Dirección IPv4. . . . . : 192.168.1.55
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 192.168.1.1

Adaptador de LAN inalámbrica Conexión de área local* 1:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Conexión de área local* 2:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::aa1f:6c6d:27f6:7f29%5
    Dirección IPv4. . . . . : 192.168.1.58
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 192.168.1.1

Adaptador de Ethernet Conexión de red Bluetooth:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

C:\Users\josec>
```

Un PC con cable Ethernet y wifi ambos activados tiene 2 direcciones de red, en el ejemplo tiene

192.168.1.55 -> Ethernet 4

192.168.1.58 -> Wifi

Para escuchar por todas, se configura bind como  
`socket.bind((0.0.0.0), port )` or  
`socket.bind("", port )`

usa la dirección especial `INADDR_ANY` para escuchar en todas las direcciones al mismo tiempo

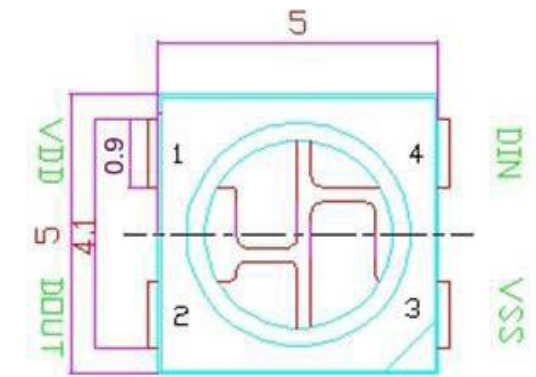
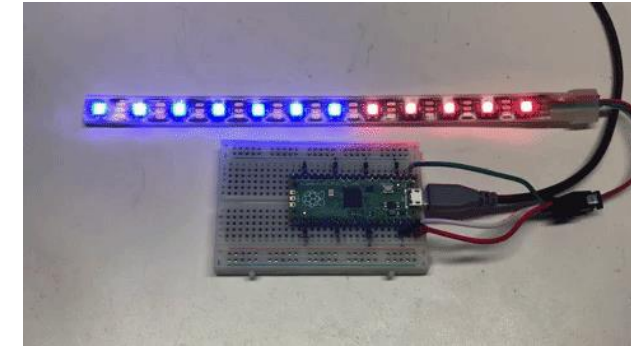
# [Robotica] Neopixel ¿Qué son y para que se usan?

## Seguiremos fuente # 3 - MicroPython for Kids

<https://www.coderdojotc.org/micropython/basics/05-neopixel/>

Los **neopixeles** son

- Dispositivos de Salida,
- Tipo led que combinan los 3 colores básicos Red Green Blue, con intensidades programables en cada color = 8bits x3 => 16 millones de colores
- Se pueden conectar “encadenados” y se puede programar el color de cada pieza de la cadena
- Solo requieren 1 hilo para su control + 2 para alimentación
  - El hilo de control “entra” y “sale” de cada neopixel
- El mas común es el compuesto por unidades de **WS2812B**
- Usaremos la librería integrada en MicroPython: desde 1.18 hay soporte integrado para NeoPixels para el microcontrolador Raspberry Pi Pico y Pico W (RP2040)
- Alimentación de módulos a 5volt con uControlador a 3.3volt -> [Ver Uso Basico](#)
- Cómo mezclar lógicas de 5volt y de 3.3 volt



NO.	Symbol	Function description
1	VDD	Power supply LED
2	DOUT	Control data signal output
3	VSS	Ground
4	DIN	Control data signal input

# [Orden] #1 Que hemos hecho hasta ahora y que sugiero hacer en los próximos meses

Clases	Fechas	Super Resumen
0,1,2,3	Feb-M	Primeros pasos , 1ros programas
4	Marzo	Entrega <b>PicoW</b> , <b>Led</b>
5,6 y 7	Ma-Ab	<b>Servos</b> , <b>POO</b> , <b>hacer librería Servo</b>
8,9	Abril	<b>Pulsadores</b> , rebotes, <b>interrupciones</b> <b>Prototipado progresivo</b> : cifra cesar <b>Buscar librerías</b>
10-11 12	Mayo	3 en ralla – <b>bloques juego, estrategias</b> <b>Neopixel</b> , volts-niveles lógicos / Trama Bucles <b>for</b> / <b>Funciones</b> / <b>Ficheros-&gt;Log</b> Entradas analógicas <b>ADC</b> , <b>Sensor Temp</b>
13+14	Mayo Junio	<b>Multímetro</b> : uso básico <b>Diccionarios</b> , <b>List comprehension</b> Sonido <b>Altavoz</b> con <b>transistor NPN</b>
15+16	Junio	<b>Proyecto Completo</b> Termómetro ( <b>NTC</b> -> <b>DHT</b> ) con Display <b>LCD I2C</b>

## Objetivos sugeridos para 4T2023- 2024

1. 'Internet no es difícil' – haremos IoT
  - Vamos a usar las capacidades Wifi del PICO W para “cacharear” con Internet de las cosas (IoT)
  - En Python del PC haremos programas que lean API's de internet
2. **Seguiremos profundizando en conceptos de Python para poder hacer mejores programas, tanto en PC como en PICOW**
  - Ejemplo: profundizar en diccionarios y datos JSON (que se usan intensivamente en API's)
3. Opcional – Avanzar aún más en Python
  - Hacer juegos con interfaz gráfica **PyGame**, con [MAKING GAMES WITH PYTHON & PYGAME](#)
  - Hacer **programas prácticos** con [AUTOMATE THE BORING STUFF WITH PYTHON](#)



# [Orden y Para casa] #2 Internet para makers

Ejemplo **Hecho** / **Pendiente**

¿Qué es internet? Selección Tutos

[How does the Internet work? 1](#)

[How does the Internet work? 2](#)

[Guía Buena algo Antigua\(ingles\)](#)  
**mucho detalle**

Video

[How the internet Works in 5 minutes](#): A 5-minute by Aaron Titus

[¿Cómo funciona Internet?](#) 10 minutos en castellano

Tipos de proyectos maker – IoT y ejemplos

1. Leer/acceder a recursos a través de API's
  - Ej. Cheerligh
  - Ej Datos meteo Openweather / AEMET
2. Publicar datos de sensores propios
  1. POST : Thinkspeak
  2. MQTT: Thinkspeak
  3. En modo AP : ej valor temperatura PICO W
3. Avisos de sensores propios a email, Telegram o WA : ej. IFTTT
4. Controlar HW propio
  1. Desde Dentro de TU red local
    - Web server LEDx3 + NTC / Control de relé
  2. Desde Fuera de la red local ( **es más complejo**)
    1. API's o MQTT:
    2. APP Movil : ej Anvil
    3. Firebase (google) .....(abrir puertos en router NO RECOMENDABLE)