

*Aprendemos Python y Robotica con micropython en paralelo,
para que los 2 aprendizajes se beneficien mutuamente*

Taller Programación y Robótica-CMM BML-2023-CL19s

Taller de Programación y Robótica en CMM BML 2023T2 – CL19s - 18 Octubre 2023

[R] Conexión internet con Pico W- 1ros
pasos#2

1. [R] Modos de conexión de un uC- 5'
2. [R] Conexión básica STA – 20'
3. [R] Conexión mejorada STA 1 & 2 – 35'



Voluntario : J.C. Santamaria

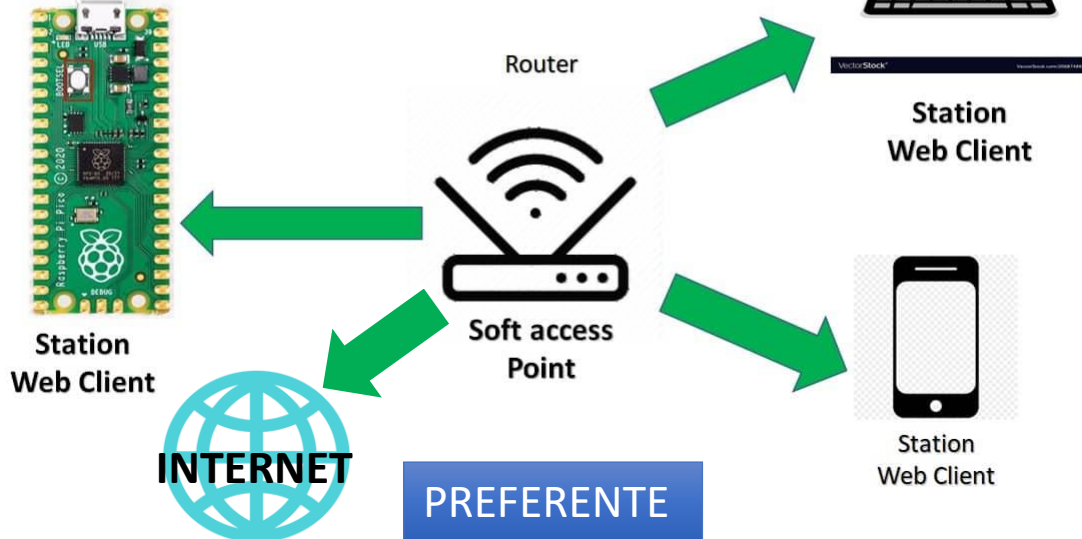
Este obra está bajo una [licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).





Clase 19s.1 – [R] Básico de conexión wifi(es = en otros uControladores como ESP32)

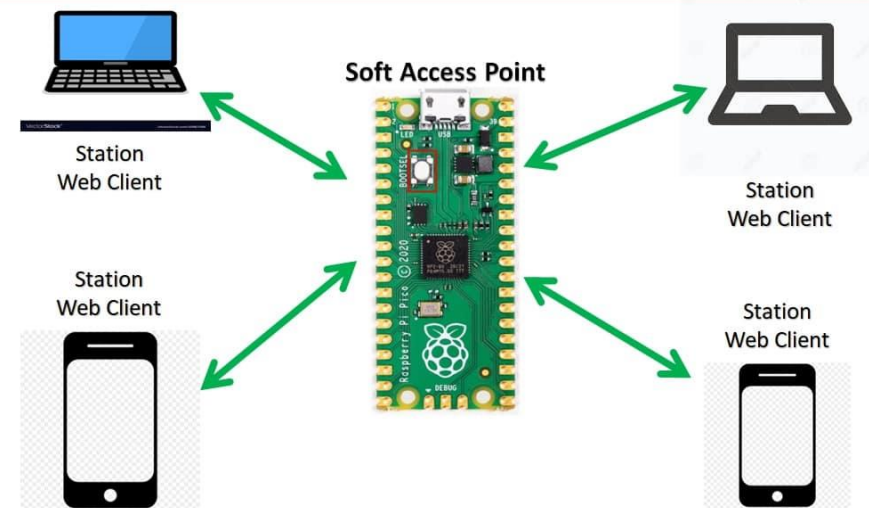
Raspberry Pi Pico W as station



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 conectados a esta red Wi-Fi privada, cada uno con su IP de red privada

Pico W in Soft Access Point Mode





Clase 19s.2.1– [R] Conexión Wifi básica => solo Pico W

[BMMR CL19s wifi connect 0 1.py](#) Solo para enseñar – NO USAR

El tutorial de sunfounder IoT Access es muy bueno. Comentaremos principalmente la parte de gestión de errores y excepciones, en la parte de **Conexión Compleja**

```
import network # varias funciones y objetos requeridos para la
conexión wifi
```

```
from secrets import * # guardados en secrets como diccionario
SSID y PASSWORD
```

```
wlan = network.WLAN(network.STA_IF)
```

```
# 1- Crea un objeto wireless local área network en modo estación
```

```
wlan.active(True)
```

```
# 2- activa el circuito wifi - tarda unos segundos
```

```
wlan.connect(secrets['ssid'], secrets['password'])
```

```
# 3- se conecta a la red wifi SSID con al PASSWORD
```

```
# Mas seguro y mantenible en fichero aparte
```

```
print(wlan.isconnected()) # 4- será = True si está conectado
```

1. `wlan = network.WLAN(network.STA_IF)`

- Crea un objeto en uPython que se corresponde con el HW del circuito wifi de la Pico w.
- `STA_IF` indica modo estación = adecuado para conectarse a routers wifi

2. `wlan.active(True)`

Activa el circuito wifi de la Pico W, que esta desactivado para ahorrar energía

3. `wlan.connect(secrets['ssid'], secrets['password'])`

Se 'loga' en la red wifi. El nombre de la red SSID y la PASSWORD se pueden escribir directamente en el código, pero es más seguro y 'mantenible' en un fichero

4. `print(wlan.isconnected())` -> Cuidado

No se creará la conexión inmediatamente, prueba a ejecutar 2 o 3 veces

Clase 19s.2.1.1– [R] Conexión Wifi básica => solo Pico W

BMMR CL19s wifi connect 0 1.py Solo para enseñar – NO USAR

Si ejecutamos el programa, devolverá False porque tarda un tiempo a conectarse

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

Microcontroller: Pico W SOLO - Key other HW : Nada
Program: Connection wifi networks - super basic - Version: 0.1
False

>>> %Run -c $EDITOR_CONTENT

Microcontroller: Pico W SOLO - Key other HW : Nada
Program: Connection wifi networks - super basic - Version: 0.1
False

>>> %Run -c $EDITOR_CONTENT

Microcontroller: Pico W SOLO - Key other HW : Nada
Program: Connection wifi networks - super basic - Version: 0.1
True

>>>
```

La identificación y password de la red wifi, las guardamos en un fichero aparte como diccionario

```
1 # Hardware all
2 # Author : JC Santamaria
3 # Date : 2023 - 2 - 19
4 # Goal : Store in a separate file wifi credential
5 # Ref :https://docs.sunfounder.com/projects/kepler-kit/en/late
6
7 secrets = {
8     'ssid': 'CMM_PIAS_Invitados',
9     'password': 'mayorescentro2022',
10    'openweather_api_key': '0cc26725d0116fea49f810212be5eb27',
11    'webhooks_key' : 'b6m1B0SPIZyvHmPV7SKN_S',
12    'aemet_api_key': "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJqY3Nwb3phQGd
13 }
14 |
```



Clase 19s.3.1– [R] Conexión Wifi básica 0.2 => solo Pico W

[BMMR CL18r wifi connect 0 2.py](#)

```
wlan = network.WLAN(network.STA_IF)
wlan.active(True) # activa el circuito wifi - tarda unos segundos
start = ticks_ms() # Inicializamos una variable para contar tiempo
if not wlan.isconnected(): # Check de si YA está conectado
    wlan.connect(secrets['ssid'], secrets['password'])
    print("Waiting for connection...")
    counter = 0
    while not wlan.isconnected(): # Check de si está conectado
        sleep(1)
        print(counter, '.', sep="", end="")
        counter += 1

delta = ticks_diff(ticks_ms(), start)
print("Connect Time:", delta, 'milliseconds')
print('IP Address:', wlan.ifconfig()[0])
```

El problema está en el tiempo de espera hasta que la conexión esta activa, que comprobamos con

wlan.isconnected() → **Solución 1:**

1. Con el **'if not wlan.isconnected()'** comprobamos si YA estuviera conectado el wifi
2. Si no, conectamos con **'wlan.connect(secrets['ssid'], secrets['password'])'**, recuerda que importamos **secrets.py** con el diccionario de claves wifi
3. Entramos en un bucle de espera con **while not wlan.isconnected()**
4. Al salir del bucle de espera , restamos los contadores de tiempo y mostramos la dirección IP

Ver sin contar el tiempo en [BMMR CL19s wifi connect 0 3.py](#)

Solución 2: más técnica, ver el status de la conexión
WLAN wlan.status()



Clase 19s.3.2 [R] Conexión Wifi MEJORADA. Solución 2 => solo Pico W

El tutorial de [sunfounder IoT Access](#) es muy bueno. Comentaremos principalmente la parte de gestión de errores y excepciones. Consultar también el tuto oficial [Connecting to the Internet with Raspberry Pi Pico W](#)

Espero hasta que se conecta **Y No Errores** (maximo 10 veces)

```
# 3- se conecta a la red wifi SSID con la clave PASSWORD,
wlan.connect(ssid, psk)

# 4 - hacer un máximo de 10 intentos de conexión espera de 1 seg
wait = 10

wStatus = wlan.status()

while wait > 0 and (wStatus >= 0 and wStatus < 3):

    # 3 posibilidades de salir del bucle: 1->numero de intentos 0
    # 2->por error wStatus = -3, -2 o -1
    # 3-> conexion con exito que es wStatus = 3

    wStatus = wlan.status()
    wait -= 1

    print(secrets['ssid'] + ' Wlan Status = ' + wStatusDef[wStatus])

    time.sleep(1)
```

salió sin Errores?

```
# 5 Compruebo si salió por error, es decir wStatus distinto de '3'
if wStatus != 3:

    raise RuntimeError('Wifi connection failed-'+ wStatusDef[wStatus])

else:

    # Conexion exitosa!!

    print('CONNECTED-'+wStatusDef[wStatus])

    ip=wlan.ifconfig()[0]

    # extrae la direccion ip de PICO W, util para conectar

    return ip
```

Mejora : convierto el programa en función “do_connect” y lo cargo como modulo

Los códigos de error, estan ‘traducidos’ en un diccionario

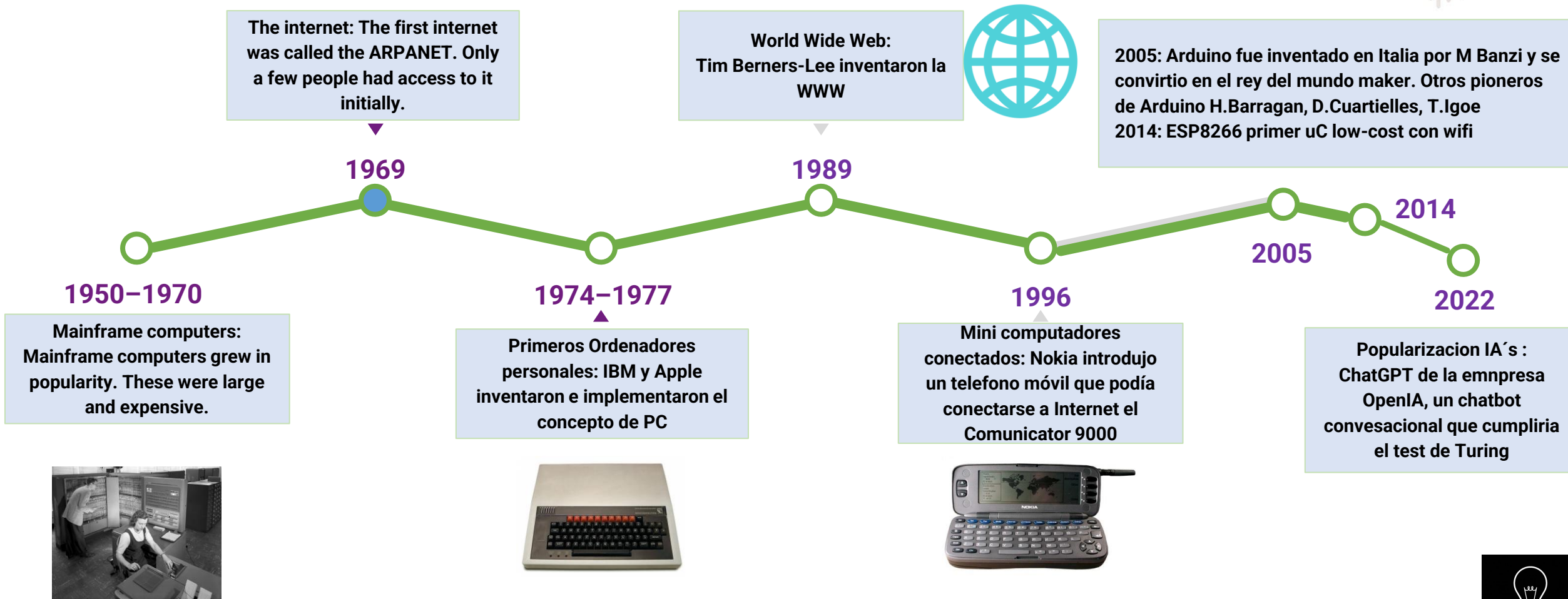
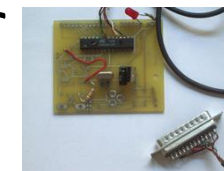
Nota : se puede ejecutar con mucha información adicional, parámetro comment = True



Clase 17.2.1 – [T] Algunos hitos históricos en relación a los ordenadores, internet y el mundo maker

Fuente : Raspberry Foundation – KS3 – Unit - Networks from semaphores to the Internet

+ Elaboración propia de 3 cuadros finales



[Teoria] 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

RECORDAR

Lo veremos en algun momento
de este curso

HTTP + DNS

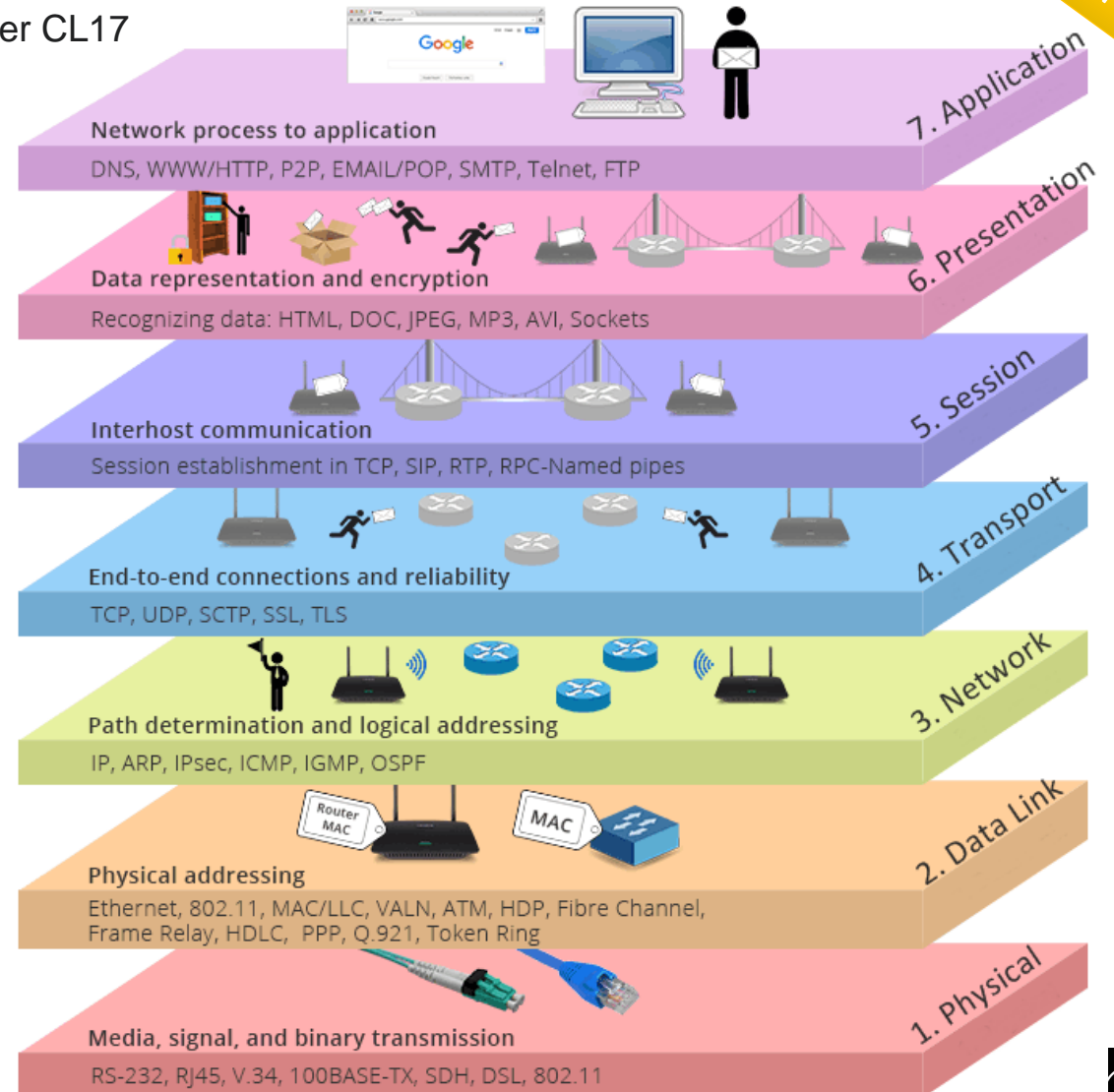
HTML + Sockets

TCP + (SSL y TLS en protocolos con seguridad)

IP : direcciones IPV4 e IPV6

Ethernet / wifi + direcciones MAC

Cables RJ45 / wifi = 802.11



Internet para makers

¿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