

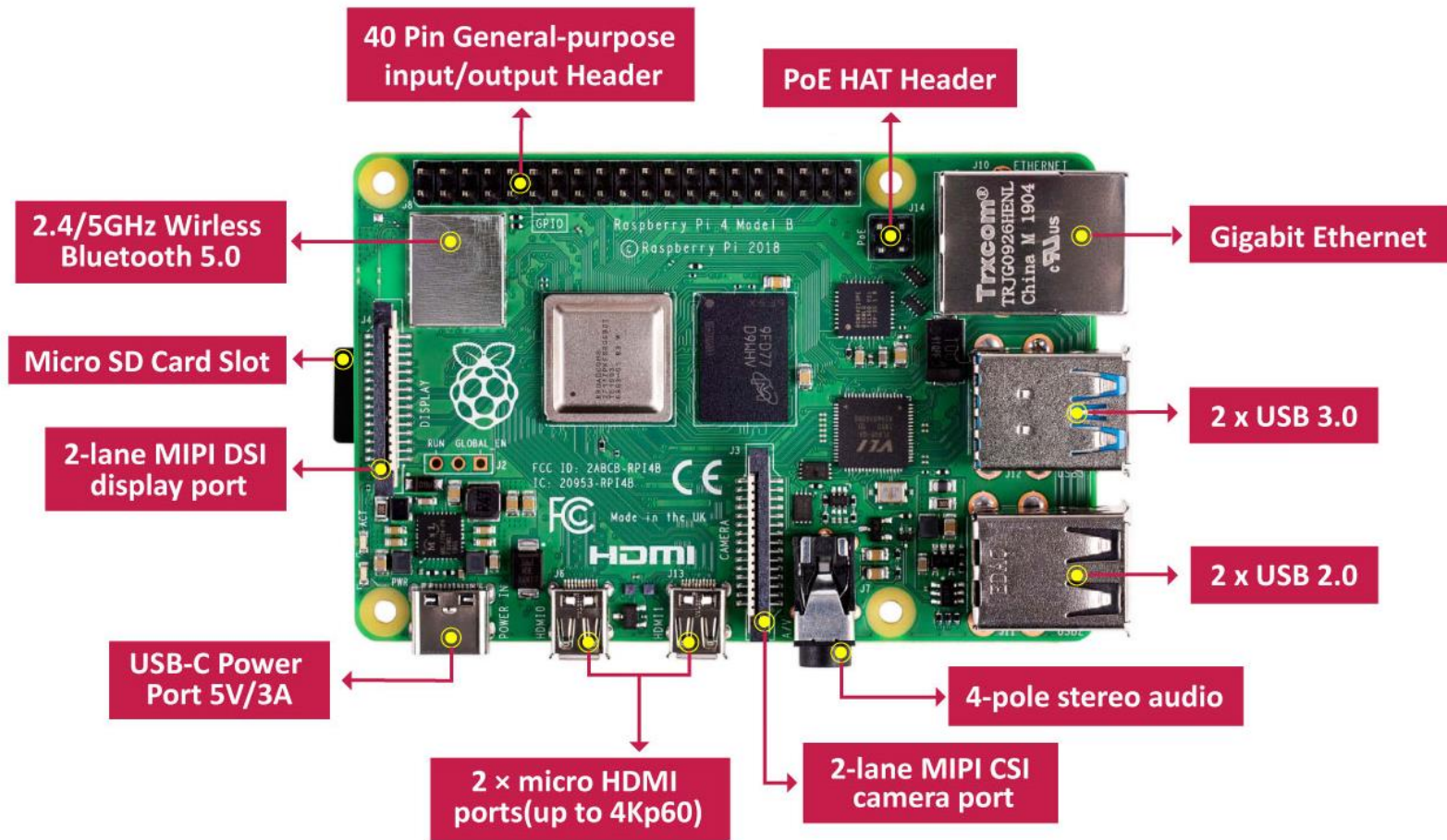


**salesianos**  
ATOCHA

# RASPBERRY PI PUERTOS GPIO

**PROFESORA: MARÍA ALBENDEA**  
[maría.albendea@salesianosatocha.es](mailto:maría.albendea@salesianosatocha.es)

# HARDWARE RASPBERRY PI



# PINES GPIO

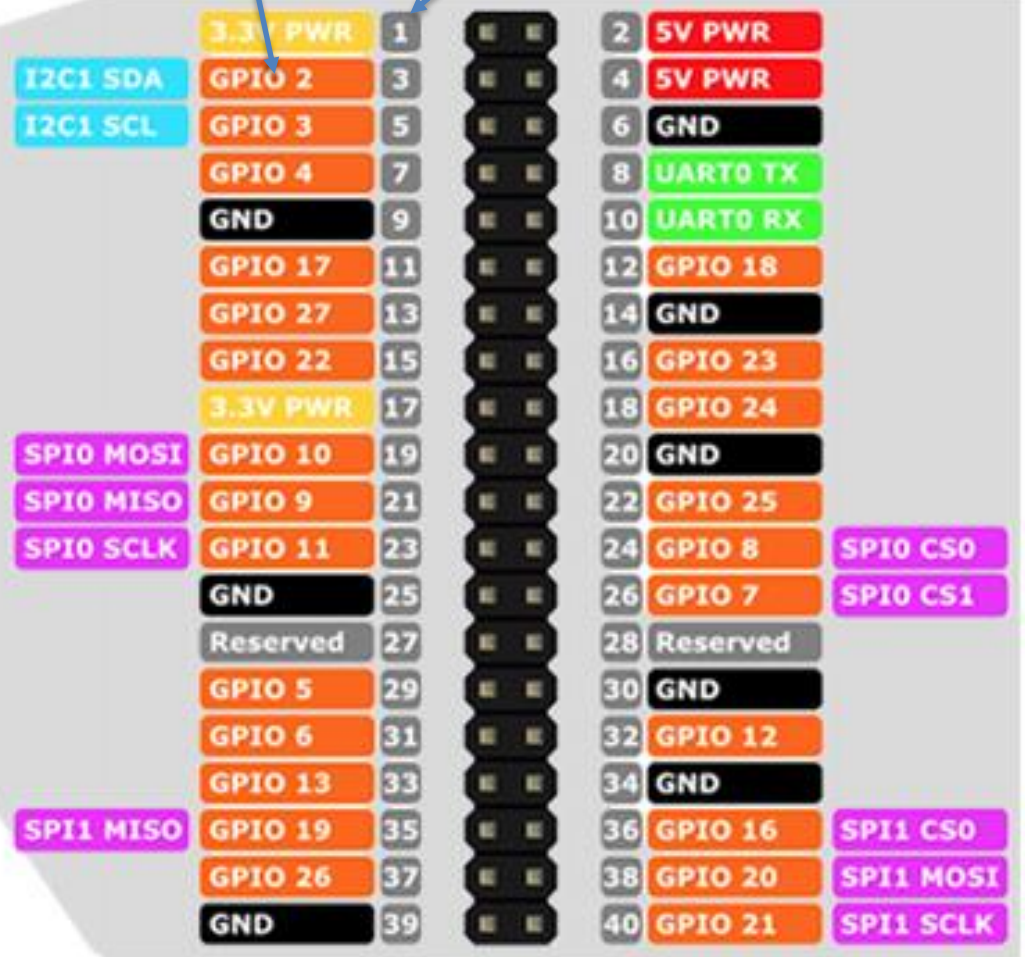
- Los pines GPIO (General Purpose Input Output) de la Raspberry pi, nos permiten **interactuar con el mundo exterior, y con la electrónica que la rodea.**
- Son pines o terminales que están pensados para **comunicarnos con los dispositivos electrónicos que deseemos controlar** o de los cuales podemos **obtener** datos e **información** en general.
- Cada puerto GPIO tiene su propósito, **varios pines trabajando en conjunto pueden formar un circuito en particular**
- Cuando un pin es una **entrada**, el programa puede **leer** si este tiene un alto o un bajo voltaje. Cuando, por el contrario, el pin es una **salida**, el software puede **generar un voltaje alto o bajo en esa patilla.**





## BCM

BOARD



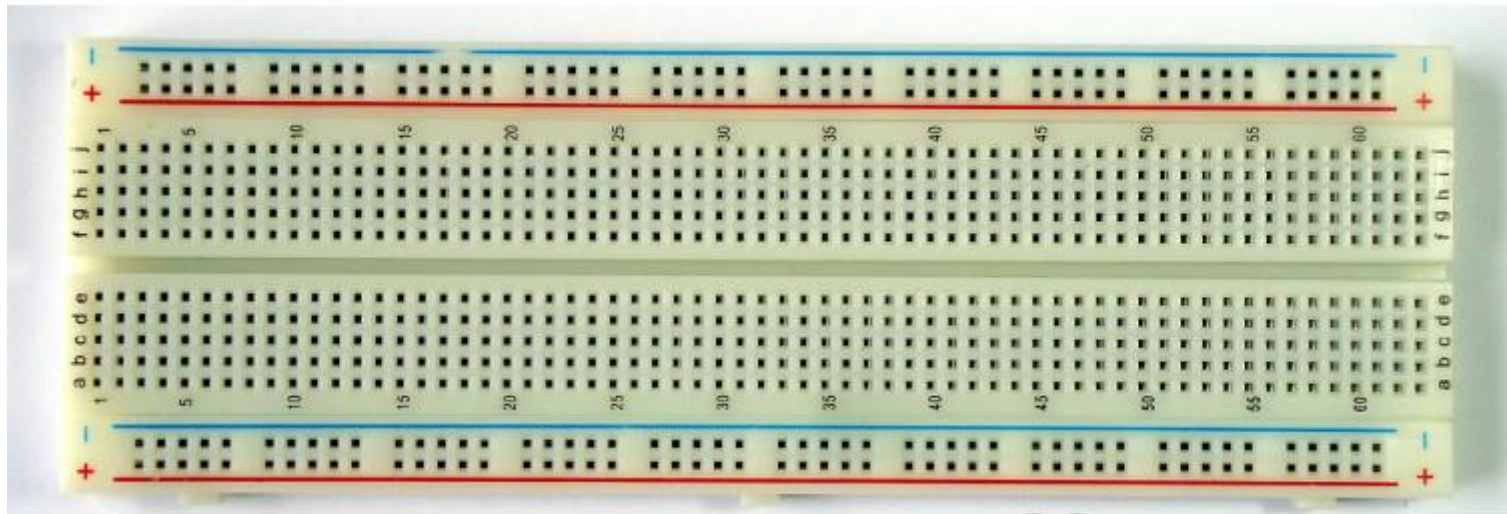
# PINES GPIO

Tal y como se ve en la imagen anterior hay dos formas de nombrar a los pines GPIO:

- **BOARD (Marcado en gris y ordenados):** Numeración que aparece en la imagen por orden **PIN#**. Cuidado que entre ellos se cuentan los voltajes y tierras y estos no se pueden definir como Inputs y Outputs
- **BCM (Marcado con colores y desordenado):** La segunda forma es con la referencia del SoC System On Chip, que son los que aparecen desordenados y solo se cuentan los GPIOs íntegramente. Pueden cambiar con diferentes versiones de la Raspberry

# PROTOBOARD

- Sirve para poder conectar componentes electrónicos y hacer circuitos de prueba, para asegurarnos de su buen funcionamiento.



# PROTOBOARD



- **Líneas rojas:** Ahí se conecta el positivo, la alimentación: 5V o 3,3V.
- **Líneas negras:** Ahí se conecta GND (tierra)
- **Líneas azules:** Aquí se conectan los diferentes componentes. Una misma línea azul es un mismo punto.

# EJEMPLO

## Programación:

Para interactuar con los pines GPIO, necesitamos importar la librería GPIO con la frase:

***import RPi.GPIO as GPIO***

Con esta línea se importa la librería para controlar los GPIOs y se le asocia una forma más corta para referirse a ella. A partir de este momento si quisiéramos ejecutar una función deberemos escribir GPIO.\_\_\_\_\_ y la función que requiramos.

A continuación, se define qué numeración de la placa se va a seguir: Board o BCM mediante la función: setmode()

***GPIO.setmode(GPIO.BCM)***

Seguimos indicando en cada pin si tenemos una entrada o una salida conectada con la función setup():

***GPIO.setup(17, GPIO.OUT)***



# EJEMPLO

## Programación:

En el caso de ser salida, queremos ponerlo al nivel alto o al bajo. Esto lo haremos con la función `output()`:

**`GPIO.output(17, True)`** *#Nivel alto, encendido*

**`GPIO.output(17, False)`** *#Nivel bajo, 0V, apagado*

*Si fuera una entrada, usaremos la función `input(pin)` para leer la entrada y almacenarla en una variable:*

**`value = GPIO.input(13)`**

*Si queremos añadir que espere un determinado tiempo entre una acción y otra, tendremos que importar la librería `time` (**`import time`**) y llamar a la función `sleep(segundos)`:*

**`time.sleep(1)`**

# EJEMPLO

## **Código final de ejemplo de encendido de un LED:**

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

pin=17
GPIO.setup(pin, GPIO.OUT)

GPIO.output(pin, True)
time.sleep(1)
GPIO.output(pin, False)
```

# EJEMPLO

**Conexionado:**

