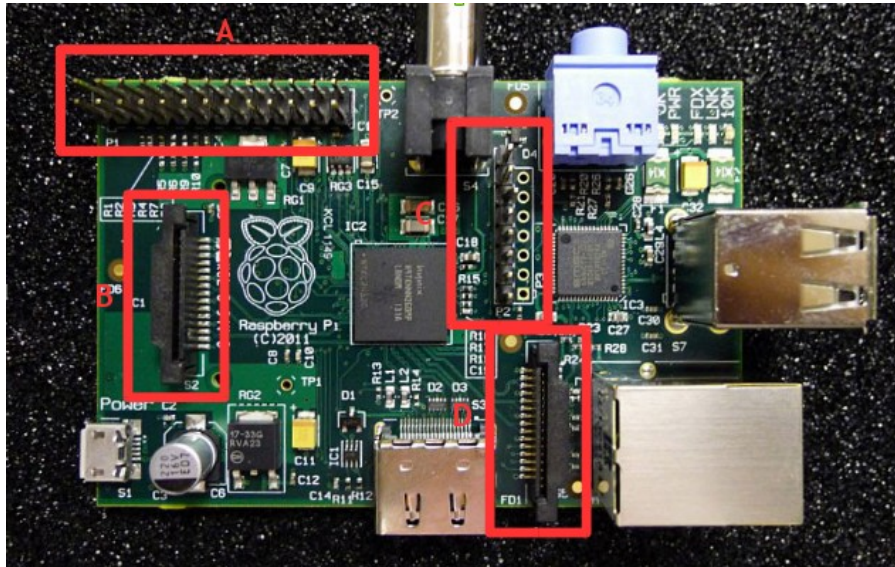


## Portos de E/S da Raspberry PI

Aparte dos conectores, a nosa placa posúe unha serie de portos de E/S.



- A) **GPIO** (General Purpose Input and Output): Permite conectarnos a unha placa Board e controlar leds, motores...
- B) **DSI** (Display Serial Interface): Permite conectarse con unha pantalla LED ou OLED.
- C) **P2 e P3**: Permiten testear o chip Broadcom (P2) e o chip de rede (P3)
- D) **CSI** (Camera Serial Interface): Permite conectar unha cámara directamente a placa

### **Modelos de raspberry dispoñibles**

Actualmente temos tres modelos distintos

**Raspberry PI - B**



**Raspberry PI - B +**



**Raspberry PI 2 - B**



## Conectándonos co mundo exterior

Empregaremos os pins da GPIO para conectar a raspberry co exterior.

Dependendo do modelo da raspberry teremos diferente número e disposición dos pins.

<b>Modelo B</b> 26 pins	
<b>Modelo B+ Raspber PI 2</b> 40 pins	

Hai algúns pins que proporcionan un voltaxe continuo e outros, os etiquetados como GPIO que poden ser programados por software.

Existen dúas codificacións para os pins do GPIO. Empregaremos sempre **GPIO.BOARD**.

<b>GPIO.setmode(GPIO.BOARD)</b>	<b>GPIO.setmode(GPIO.BCM)</b>

Para comunicarnos co GPIO empregaremos **Python**. Creamos o seguinte arquivo chamado probaPins.py

```
#!/usr/bin/env python
# Este arquivo emprega encoding: utf-8
# As dúas liñas anteriores son imprescindibles para poder empregar
acentos
# O seguinte programa fai 5 veces o seguinte
    # activa o pin 10
    # agarda 2 segundos
    # desactiva o pin 10
# Podemos medir a súa actividade con un polímetro

#importamos as librerías necesarias
import RPi.GPIO as GPIO # Empregar o GPIO
import time              # Empregar os retardos

#Establecemos o sistema de numeración dos pins
GPIO.setmode(GPIO.BOARD)

#Configuramos o pin 3 como unha saída
GPIO.setup(10, GPIO.OUT)

for i in range(0,5):
    GPIO.output(10, GPIO.HIGH)# 3.3v
    time.sleep(2)
    GPIO.output(10, GPIO.LOW) # 0v
    time.sleep(2)

GPIO.cleanup() # Devolve os pins o estado orixinal
```

Para executar o programa

```
sudo python proba1.py
```

Podemos comprobar o seu funcionamento empregando un polímetro. Cable vermello o pin 3 e cable negro o pin 6 e veremos como cambia a voltaxe.

**Importante:** A partir da versión **bookworm** de Debian, para traballar co GPIO é preciso instalar o paquete **python3-rpi.gpio**

## Conexión da raspberry PI cunha placa board

Temos tres alternativas

- GPIO Ribbon cable
- Cables femia-femia
- Empregar un cable floppy



### **Cable femia-femia**

É o máis sinxelo de empregar, só conectaremos os cables necesarios

### **GPIO Ribbon cable**

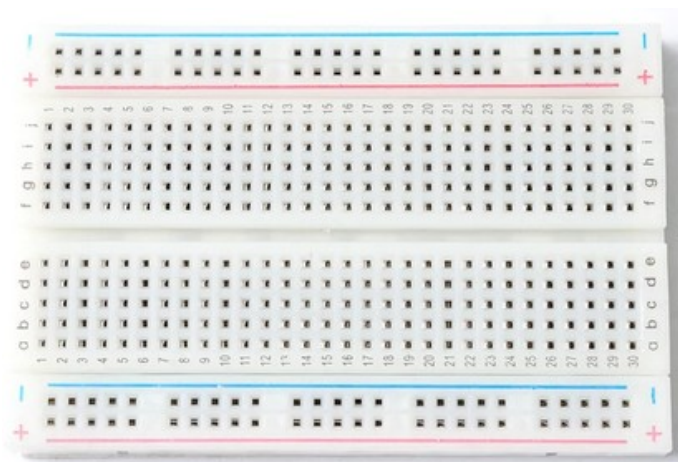
Temos que identificar os conectores. Sempre o conectaremos co pin vermello o conector 1 do GPIO



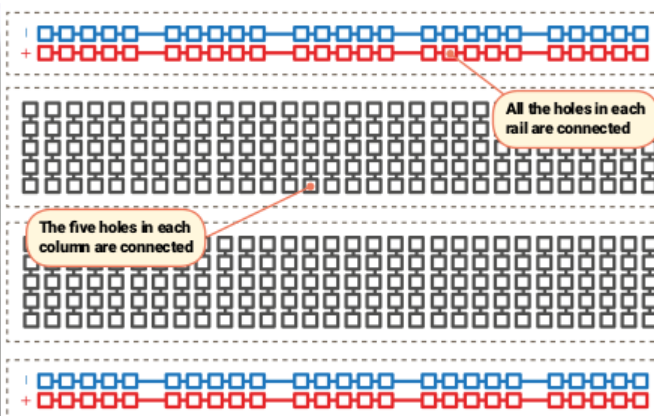
Unha **placa board (breadboard)** é unha placa onde podemos probar os nosos circuítos sen necesidade de soldaduras.

- As columnas dunha mesma sección están conectadas
- Hai filas especiais que están conectadas

### **IMAXE**



### **CONEXIÓNS INTERNAS**



Unha boa explicación do seu funcionamento podémolo atopar [aquí](#).

Para facer os diagramas e familiarizarnos podemos empregar **fritzing**, pero só permite facer os diagramas, non simulacións.



## **Identificando o modelo da Raspberry**

O comando **pinout** amosa o modelo da nosa raspberry, as súas características hardware e os pins dispoñibles.

- Instalación

```
apt install python3-gpiozero
```

- Emprego

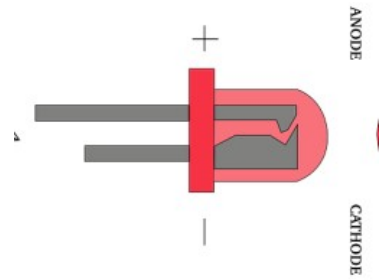


```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ pinout  
-----  
  00000000000000000000 J8  
  10000000000000000000  
-----  
Pi Model 3B V1.2  
-----  
  [D] [SoC] [C] [S] [I] [A] [V]  
  [S] [ ] [ ] [ ] [ ] [ ] [ ]  
  [I] [ ] [ ] [ ] [ ] [ ] [ ]  
-----  
  pwr HDMI I A V  
-----  
  +====+  
  | USB |  
  +====+  
  +====+  
  | USB |  
  +====+  
  +====+  
  | Net |  
  +====+  
-----  
Revision      : a02082  
SoC           : BCM2837  
RAM           : 1024Mb  
Storage       : MicroSD  
USB ports     : 4 (excluding power)  
Ethernet ports: 1  
Wi-fi        : True  
Bluetooth     : True  
Camera ports (CSI) : 1  
Display ports (DSI): 1  
  
J8:  
  3V3 (1) (2) 5V  
  GPIO2 (3) (4) 5V  
  GPIO3 (5) (6) GND  
  GPIO4 (7) (8) GPIO14  
  GND (9) (10) GPIO15  
  GPIO17 (11) (12) GPIO18  
  GPIO27 (13) (14) GND  
  GPIO22 (15) (16) GPIO23  
  3V3 (17) (18) GPIO24  
  GPIO10 (19) (20) GND  
  GPIO9 (21) (22) GPIO25  
  GPIO11 (23) (24) GPIO8  
  GND (25) (26) GPIO7  
  GPIO0 (27) (28) GPIO1  
  GPIO5 (29) (30) GND  
  GPIO6 (31) (32) GPIO12  
  GPIO13 (33) (34) GND  
  GPIO19 (35) (36) GPIO16  
  GPIO26 (37) (38) GPIO20  
  GND (39) (40) GPIO21  
  
For further information, please refer to https://pinout.xyz/  
pi@raspberrypi:~ $
```

## Dispositivos de Saída: Leds

Un Led ten dúas patillas

- **Ánodo:** A larga, é a positiva
- **Cátodo:** A pequena, negativa



Para acceder un led temos que dar corrente pola positiva e pechar o circuío pola negativa.

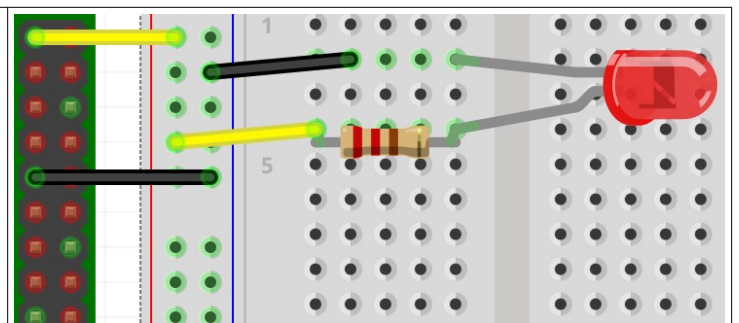
Os leds consumen 2v, como o alimentaremos con 3,3v teremos que empregar unha resistencia en serie no para evitar queimalo.

Teoricamente deberíamos conectar a resistencia á pata positiva do led, pero na práctica da igual.

Para alimentar o circuío empregaremos a nosa Raspberry Pi.



Sen Resistencia (Podemos queimar o led)



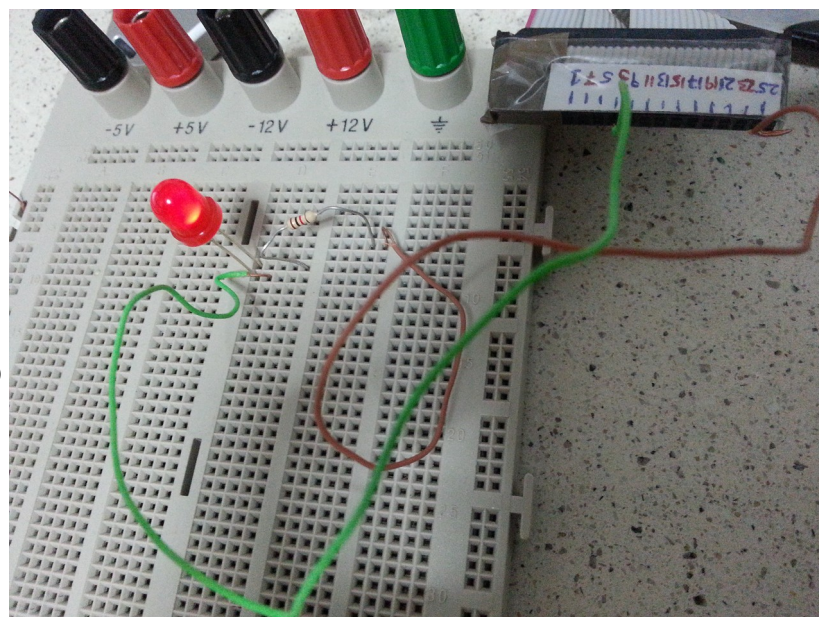
Con resistencia

O cable amarelo será o que proporcione electricidade. Para probar que funciona conectáremolo o pin1 que proporciona 3,3v. O azul será o de terra para pechar o circuío, conectáremolo o pin 25.

Se todo está correctamente conectado o led acenderá todo o tempo. Pero nos queremos controlar por programa cando acende. Para iso conectamos o cable amarelo o conector 3.

Executamos o programa python anterior

Se todo vai ben veremos como se ilumina e apaga o led 5 veces



## Exercicios con leds:

1. Facer o diagrama no fritzing para acender o led, con e sen resistencia
2. Probar circuío iluminar ler conectado a 5v e a 3v sen resistencia
3. Probar circuío iluminar ler conectado a 5v e a 3v con resistencia

4. Acender o led por programa

A partir do código de exemplo:

- Seleccionar pin como saída
- Probar a acender (3.3v), Agardar 2 seg.
- Probar a apagar (0 v)

5. Facer o programa anterior empregando **00-plantilla.py**

6. Desactivar a bomba

Comeza o programa co led acendido

- Pedir un números ata introducir 0
- Apagar led

7. Desactivar a bomba v2

- Comeza o programa co led acendido
- Xerar un número aleatorio entre 1 e 10
- Definir Tempo 5 seg
- Pedir número
  - Se se acerta o número aleatorio: Apagar led, bomba desactivada
  - Se non se acerta pedir máis números
  - Se se acaba o tempo a bomba estoupa

### Pista:

```
import time
import random
#Xeramos número aleatorio do 1 o 10
clave= random.randint(1, 10)
tempoInicial = time.time() #Obtemos o tempo inicial
tempoTranscurrido=time.time()-tempoInicial
```

8. Acender alternativamente dous leds