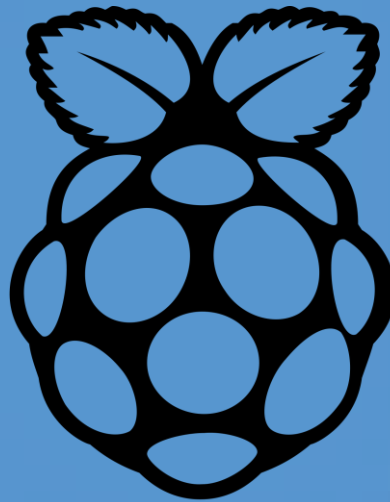


# CURSO DE RASPBERRY PI



## Clase 2

# ¿Linux?

El núcleo de Linux fue creado por Linus Torvalds a principios de los 90. Y se pensó basándose en Unix.

(Sistema operativo portable, multitarea y multiusuario; desarrollado en 1969 por un grupo de empleados de los laboratorios Bell de AT&T).

A partir de los 2000 se empezaron a crear muchas distribuciones basadas en Linux.

El desarrollo de Linux es uno de los ejemplos más claros de desarrollo de software libre. Hay una comunidad dispersa de programadores y cualquiera puede usar el sistema operativo, estudiarlo y modificarlo. Estos derechos están protegidos por la licencia GPL (GNU General Public License).

Linux cuenta con distribuciones en sistemas operativos para servidores y para clientes.

# Conceptos

*Usuario Estándar:* Permisos Limitados

*Usuario Root (superusuario):* Control total sobre el sistema operativo  
Llamar " root " a un Super Usuario tiene origen en que son los usuarios que tienen acceso a la raíz del sistema, es decir, a /root.

*Shell:* En Linux se refiere a un programa que se utiliza para enviar comandos al sistema operativo. La analogía más cercana en Windows es la línea de comandos DOS.

*Interfaz grafica (GUI, de Graphic User Interfase):* tipo de visualización que permite al usuario elegir comandos, iniciar programas, ver listas de archivos, otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú.

# Superusuario

`sudo`  
(SUperuser DO) lo ejecuta un usuario normal y toma permisos para usar ciertos comandos. Sudo, requiere que los usuarios se autentifiquen a sí mismos a través de su contraseña para permitirles el control "root".

`sudo su`  
Otra posibilidad para conseguir permisos de super usuario seria usar el comando  
" `sudo su` ". Esto nos otorga permisos de super usuario hasta cerrar sesión, o hasta que que ejecutemos el comando " `exit` "

# Sistema de administración de paquetes.

Un sistema de gestión de paquetes, es una colección de herramientas que sirven para automatizar el proceso de instalación, actualización, configuración y eliminación de software.

En los sistemas basados en Linux, el software se distribuye en forma de paquetes, frecuentemente encapsulado en un solo fichero. Estos paquetes incluyen otra información importante, además del software mismo, como pueden ser el nombre completo, una descripción de su funcionalidad, el número de versión, el distribuidor del software, la suma de verificación y una lista de otros paquetes requeridos para el correcto funcionamiento del software. Esta información se introduce normalmente en una base de datos local.

# Comandos Generales.

**apt-get update:** Actualiza tu versión de Raspbian.

**apt-get upgrade:** Actualiza todos los paquetes que tienes instalados en el sistema.

**clear:** limpia la ventana del terminal.

**date:** Muestra la fecha actual.

**find / -name prueba.txt:** Busca en todo el sistema el archivo prueba.txt y genera una lista de todos los directorios que contienen el archivo.

**nano prueba.txt:** Abre el archivo prueba.txt en “Nano”, el editor de texto de Linux.

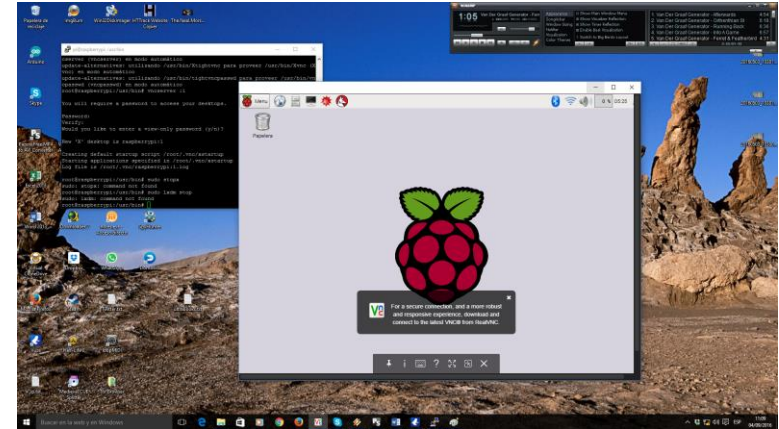
**poweroff:** Apaga el sistema de forma inmediata.

**raspi-config:** Abre el menú de configuración.

**reboot:** Reinicia el sistema de forma inmediata.

# VNC

VNC son las siglas en inglés de Virtual Network Computing. VNC es un programa de software libre basado en una estructura cliente-servidor que permite observar las acciones de un servidor remotamente a través de un cliente.



En las distribuciones modernas de Raspbian viene un VNC preinstalado pero no activado por defecto.

Los pasos son sencillos y vamos poder ver de forma remota nuestro escritorio de las Raspberry. Con lo cual no tendremos que tener un monitor exclusivo para la misma.

# Configurar servidor VNC en Raspberry

El ordenador al que nos vamos a conectar, en este caso nuestro Raspberry Pi, tiene que tener instalado el programa servidor VNC (En nuestro caso usamos Real VNC). En las distribuciones modernas de Raspbian Real VNC viene preinstalado por defecto, pero no activado.

Para activar el servidor VNC en Raspberry Pi deberemos meternos en la configuración de Raspberry Pi y activar la opción correspondiente.

Por otro lado, resulta conveniente tener configurada una IP estática, para que no cambie cada vez que reiniciemos la Raspberry Pi.

Una vez configurada la VNC, en principio, funcionara únicamente en la red local. Si queremos conectarnos desde fuera (a través de Internet) deberemos configurar los puertos correspondientes en el router y asegurarnos de que la conexión sea segura.



# Configurar cliente VNC

Para conectarnos a la Raspberry Pi deberemos instalar el programa visualizador de Real VNC en el cliente, que puede ser una notebook, una tablet, un Smartphone, etc. Disponemos de clientes para Windows, Linux, Mac y Android.

Descargamos e instalamos el visor de RealVNC adecuado para nuestro sistema operativo desde <https://www.realvnc.com>.

En el visor indicaremos la IP del sistema al que queremos conectarnos.

Al iniciar la conexión se nos pedirá el usuario y contraseña. Recordemos que por defecto es " pi " como usuario y " raspberry " como contraseña.

# Distribución GPIO y características.

## Tensiones y estados.

Aunque el puerto GPIO de la " RasPi " ofrece un suministro de 5V, proveniente de la directo de la fuente usb, sobre el Pin 2, el funcionamiento interno de la RasPi se basa en la lógica de 3.3 V. Esto significa que los componentes en la RasPi funcionan con un tensión de 3.3V. Si está planeando crear un circuito que haga de interfaz con la RasPi a través del puerto GPIO, asegúrese de utilizar componentes compatibles con lógica de 3.3V o que el circuito está pasando a través de un *adaptador de niveles lógicos* antes de llegar a la RasPi.

Entonces estos pines pueden ser conmutados entre dos estados: *alto (high)*, donde los pines proporcionan una tensión de 3.3 V; y *bajo (low)*, donde los pines son 0V (iguales referencialmente a GND). Esto equivale al 1 y 0 de la lógico, y puede utilizarse para cambiar el estado de otros componentes de activos a inactivos o viceversa.

# Distribución GPIO y características.

## Tensiones y estados.

El puerto GPIO de la " RasPi " ofrece un suministro de 5V, proveniente directamente de la fuente usb, sobre el Pin 2 y 4.

Pero la RasPi se basa en la lógica de 3.3 V. Entonces los pines digitales conmutan entre dos estados: *alto (high)*, donde los pines proporcionan una tensión de 3.3 V; y *bajo (low)*, donde los pines son 0V (iguales referencialmente a GND). Esto equivale al 1 y 0 lógico, y puede utilizarse para cambiar el estado de otros componentes digitales.

Además estos pines pueden comportarse también como entradas digitales, leyendo el valor que tengan los pines digitales. Raspberry no cuenta con pines de entrada analógicos.

## Puertos de comunicación

Un puerto UART (serie) , un puerto I2C y dos puertos SPI.

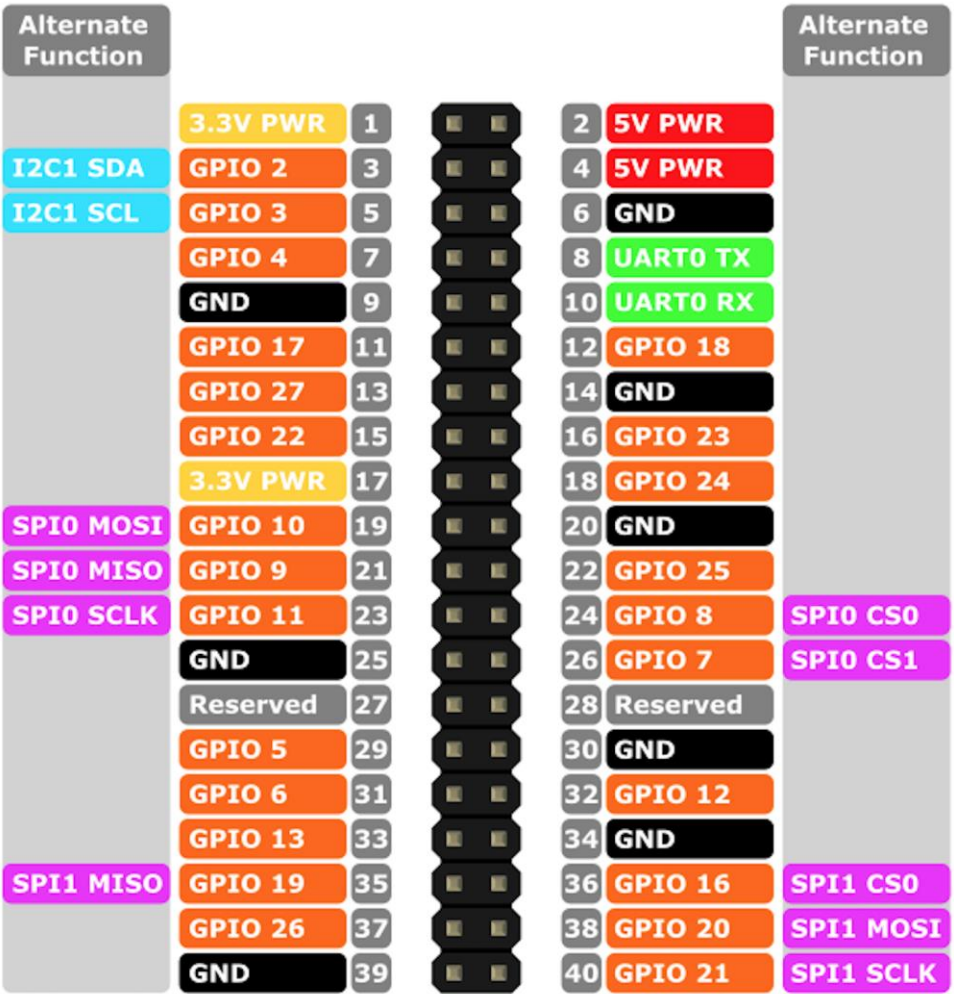
# Identificación de los GPIO.

Hay 2 sistemas de numeración de los pines GPIO estas son BCM y BOARD.

El sistema BCM usa el número de pin GPIO correspondiente.

El sistema BOARD se basa en la numeración del orden del pin en la placa.

En esta imagen se aprecia mejor la diferencia entre los dos sistemas.



# Desarrollo en “C” o en “Python”.

Podemos programar la Raspy en lenguajes como C, Python, entre otros. Este curso desarrollaremos varios ejemplos en Python siendo este último un lenguaje fácil de entender por aquellos que alguna vez programaron en C o en algún otro lenguaje. Además en la web encontraremos con mayor facilidad ejemplos y recursos en este lenguaje.

De esta forma nos enfocamos a manejar un idioma fácil, para poder ser manipulado por un gran número de alumnos.

En la actualidad se usa Python como primer lenguaje, a la hora de aprender programación.

Si igualmente quisieras experimentar, y programar los GPIO en lenguaje C++, te recomendamos usar la biblioteca WiringPi.

# Parpadeo de un LED

Como primer ejemplo, vamos a construir un circuito simple que consiste en un LED con su respectiva resistencia limitadora. El LED proveerá una confirmación visual de que el puerto GPIO hace lo que programamos en Python.

Utilizaremos el IDE Geany ya que como charlamos en la primer clase vimos que viene pre-instalado y es bastante completo.

Para hacer que el LED parpadee, creamos un nuevo proyecto de Python, y necesitaremos importar una librería dentro de nuestro script de Python. Para hacerlo, agregamos dicho módulo que nos permitirá comandar los pines como *import RPi.GPIO as GPIO*. Además necesitaremos gestionar el tiempo entre encendido y apagado , también importamos la librería *time*.

Podemos usar la nomenclatura BCM (Broadcom número de GPIO) o podemos usar Board para identificar a los pines por su posición en la placa. Queda a criterio de cada uno de nosotros.

# Circuito.

BOARD	GPIO	GPIO	BOARD
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I2C)	DC Power 5v	04
05	GPIO03 (SCL1 , I2C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)	(I2C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Primer circuito, si contamos con el cable tipo T, las conexiones se harán de una manera más fácil en el protoboard.



# Programa.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(7, GPIO.OUT)

while True:
    GPIO.output(7, True)
    time.sleep(1)
    GPIO.output(7, False)
    time.sleep(1)
```

Indicamos el modo con el cual nos vamos a dirigir al pin a utilizar, ya que podemos dirigirnos por el número en la placa (*GPIO.BOARD*), o por el canal al cual está conectado en el Chip Broadcom (*GPIO.BCM*), como se vio anteriormente.

En nuestro caso optamos por la primera opción, es decir, indicaremos que vamos a utilizar el pin situado en la posición 7 de la placa (osea *GPIO4*). La siguiente instrucción será la inicialización del pin, es decir, los pines pueden ser utilizados como entradas o salidas digitales.

Por último, solamente nos quedará activar (*True*) o desactivar (*False*) el pin seleccionado. Para ello, se va a situar el código dentro de un bucle infinito ( tener en cuenta la indentación del código).