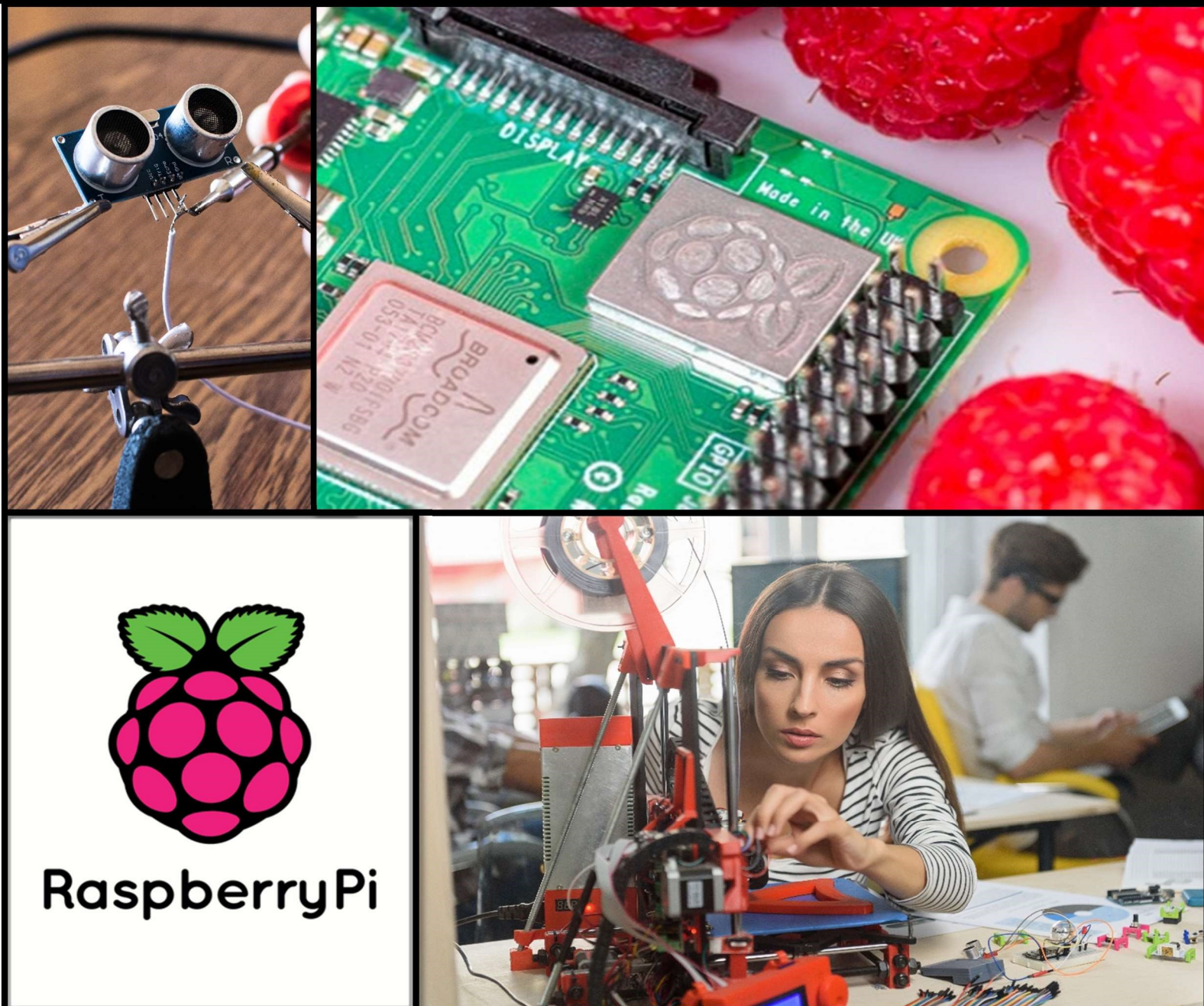


Raspberry Pi - Fundamentos



Índice

1. Datos generales	1
2. Introducción	1
2.1. Objetivos	1
2.2. Descripción del curso	1
3. Introducción	3
3.1. ¿Qué es Raspberry Pi?.....	3
3.2. Descripción de los puertos	3
3.3. Raspberry Pi 3 Modelo B+	4
3.4. Materiales necesarios	5
4. Instalación del sistema operativo.....	7
4.1. Software disponible	7
4.2. Formateo de la Micro SD.....	8
4.3. Descarga del sistema operativo	9
4.4. Instalación de Raspbian con NOOBS.....	11
4.5. Configuración de Raspbian.....	13
4.6. ¿Qué podemos hacer en Raspbian?.....	13
5. Los puertos GPIO	15
6. Programando con Scratch	16
6.1. Introducción	16
6.2. Añadir extensiones.....	17
6.3. Circuito: Encendido de un LED	18
6.4. Control GPIO con Scratch	19
7. Programando con Python.....	21
7.1. Introducción	21
7.2. Entorno de desarrollo	22
7.3. Control GPIO con Python	25

1. Datos generales

- Título: RASPBERRY PI - FUNDAMENTOS
- Organiza: DIGITALCODESIGN – KREITEK – COLEGIO NURYANA
- Fecha: 13/09/19 – 16/09/19
- Horario: 17:00 – 19:30
- Horas lectivas: 5
- Plazas: 20
- Lugar: MAKERSPACE KREITEK – COLEGIO NURYANA
- Ponente: MIQUEAS FORTES GONZÁLEZ
DANIEL ÁLVAREZ EUSEBIO
- Categoría: ELECTRÓNICA Y PROGRAMACIÓN

2. Introducción

2.1. Objetivos

El presente curso tiene por objeto el fomento de las vocaciones científicas, la creatividad y la actitud emprendedora entre alumnos, profesores y emprendedores, promoviendo un nuevo modelo de trabajo y aprendizaje basado en proyectos, también denominado Educación STEAM (Ciencia, Tecnología, Ingeniería, Arte y Matemáticas) o Educación 4.0, caracterizado por el empleo de las TIC (Tecnologías de la Información y la Comunicación) como recurso para la formación y adquisición de nuevas competencias tecnológicas.

2.2. Descripción del curso

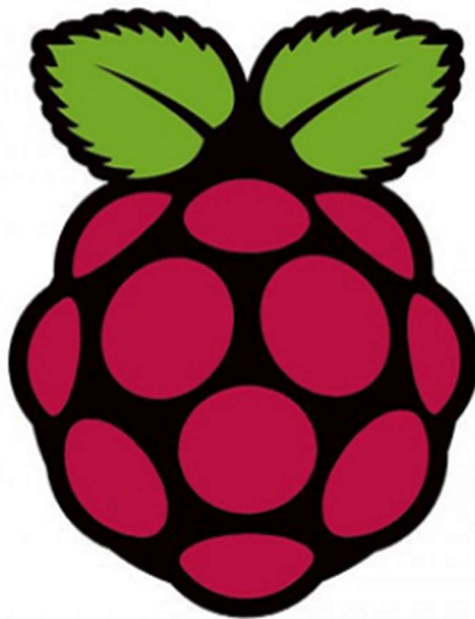
Aprender los conceptos básicos sobre el uso de la Raspberry Pi y sus aplicaciones en el ámbito de la educación.

Raspberry Pi es un ordenador de placa reducida al que podemos instalar un sistema operativo, conectar periféricos y dejarlo funcionando como un ordenador totalmente operativo, con la particular ventaja de que con su pequeño tamaño, su capacidad para conectarse físicamente con otros componentes electrónicos y dispositivos, su fácil programación, su inmensa comunidad y su reducido precio, la hace ideal para desarrollo de proyectos tanto educativos como profesionales.

A nivel de proyectos las posibilidades de Raspberry Pi son inmensas. A nivel básico, intermedio y avanzado, podemos trabajar en el aula las ciencias de la computación, aprender a usar diferentes sistemas operativos alternativos y de código abierto como GNU/Linux, aprender programación con lenguajes como Python y Scratch, construir servidores y desarrollar diversos proyectos de electrónica, robótica y domótica.

En este curso aprenderemos los aspectos básicos de Raspberry Pi. Veremos de dónde obtener los recursos necesarios para comenzar a utilizar esta tecnología, cómo instalar un sistema operativo, a conocer Raspbian y las aplicaciones con las que trabaja, a trabajar desde el terminal de Linux, a generar programas mediante lenguaje de programación Python y Scratch, así como a interactuar con componentes conectados físicamente a los puertos GPIO de la Raspberry.

El curso será eminentemente práctico y con aplicación inmediata de todos los conceptos teóricos que conforman el temario del curso. Para ello no solo se pondrá la herramienta a disposición de los alumnos desde el inicio de la clase, sino que además será explicado su funcionamiento y manejo ayudándonos de componentes y circuitos electrónicos que asimismo deberán montar los alumnos por sí solos.



3. Introducción

3.1. ¿Qué es Raspberry Pi?

Raspberry Pi es un ordenador de placa reducida al que podemos instalar un sistema operativo, conectar periféricos y dejarlo funcionando como un ordenador totalmente operativo, con la particular ventaja de que con su pequeño tamaño, su capacidad para conectarse físicamente con otros componentes electrónicos y dispositivos, su fácil programación, su inmensa comunidad y su reducido precio, la hace ideal para desarrollo de proyectos tanto educativos como profesionales.

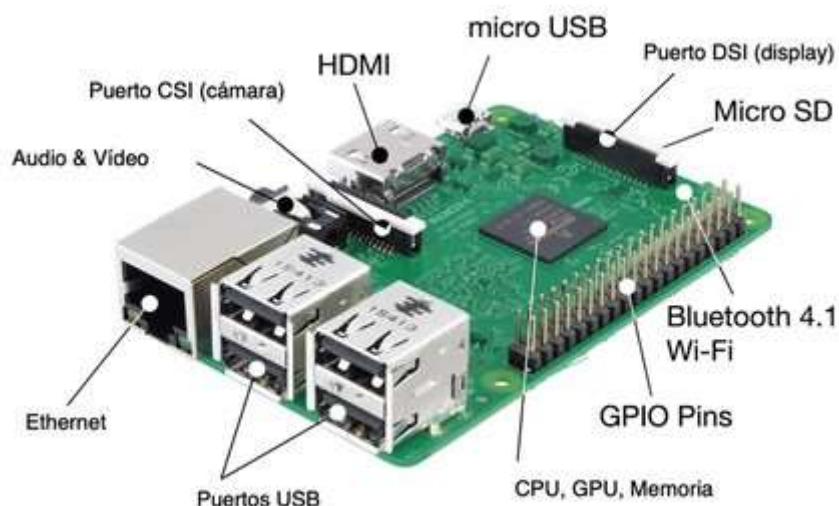


3.2. Descripción de los puertos

La siguiente imagen muestra la placa Raspberry Pi 3 Modelo B, con descripción de los puertos y elementos principales de la placa:

- **Puertos USB:** Podemos conectar ratón, teclado, pendrive, disco duro, mandos, etc...
- **Puerto Ethernet:** Para conectar la placa a internet o a una red LAN mediante cable de red.
- **Ranura tarjeta micro SD:** Para alojar la tarjeta micro SD que llevará instalado el sistema operativo.
- **Puerto HDMI:** Para conectar un monitor o una pantalla portátil.
- **Conector audio Jack:** Para conectar auriculares o altavoces.
- **Conector micro USB:** Para alimentar la Raspberry Pi por medio de una fuente de alimentación.

- **Puertos GPIO:** Para conectar otros módulos electrónicos útiles para nuestros proyectos.



3.3. Raspberry Pi 3 Modelo B+

Existen varias placas Raspberry Pi, no obstante la que tenemos previsto utilizar para el presente curso es la Raspberry Pi 3 Modelo B+, una versión mejorada de la Raspberry Pi 3 Modelo B, con las siguientes características:

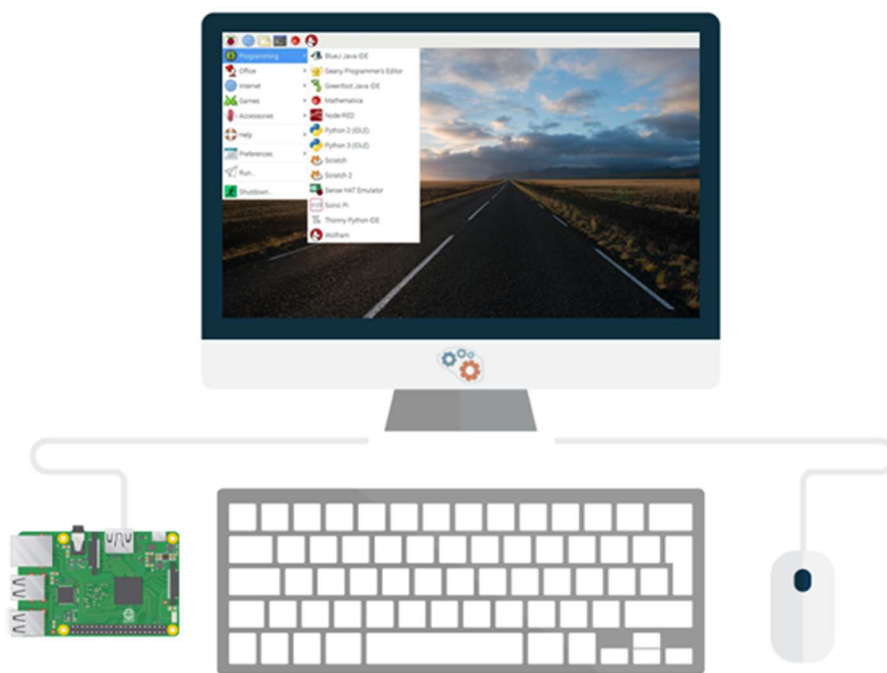
- Procesador: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
- Frecuencia de reloj: 1,4 GHz
- GPU: VideoCore IV 400 MHz
- Memoria: 1GB LPDDR2 SDRAM
- Wi-Fi: Wi-Fi a doble banda 2.4 Ghz y 5 Ghz
- Bluetooth: Bluetooth 4.2, BLE
- Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)
- GPIO de 40 pines
- Salida HDMI
- 4 puertos USB 2.0
- Puerto CSI para conectar una cámara.
- Puerto DSI para conectar una pantalla táctil
- Salida de audio estéreo y vídeo compuesto

- Micro-SD
- Power-over-Ethernet (PoE)



3.4. Materiales necesarios




Para comenzar a utilizar nuestra Raspberry Pi como un ordenador necesitaremos los siguientes elementos:




PC con Raspberry Pi

- Un Kit de iniciación a Raspberry Pi
- Un monitor
- Un teclado
- Un ratón
- Un cable de red Ethernet
- Un convertidor HDMI-VGA (solo si el monitor no dispone de entrada HDMI)

Existen diversos Kits de iniciación a Raspberry Pi disponibles en el mercado, no obstante el que nosotros recomendamos es el siguiente u otro de características similares:

KIT DE INICIACIÓN RASPBERRY PI		
Imagen	Ud.	Descripción
	1	Raspberry Pi 3 modelo B+
	1	Carcasa resistente con acceso a puertos y pines GPIO + ventilador y set de disipadores
	1	Fuente de alimentación original Raspberry Pi 5.1V 2.5A

	1	Tarjeta micro SD 16 Gb clase 10 con adaptador
---	---	---

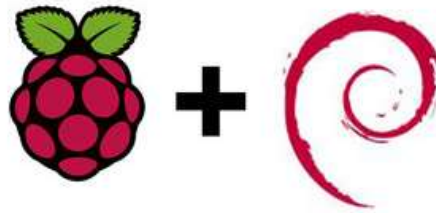
4. Instalación del sistema operativo

4.1. Software disponible

Raspberry Pi es compatible con diversos sistemas operativos, la mayoría de ellos basados en sistemas GNU/Linux.



El sistema operativo más utilizado, y por el que más se recomienda comenzar, es Raspbian, una distribución GNU/Linux optimizada para el hardware de Raspberry Pi, creada para fomentar la enseñanza de informática en las aulas.



Raspbian

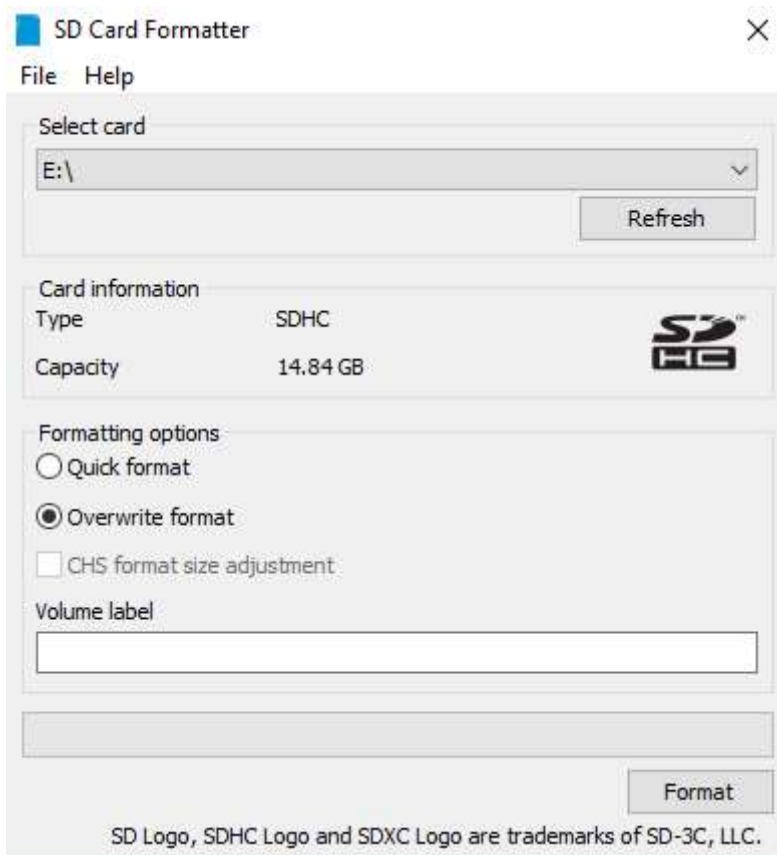
4.2. Formateo de la Micro SD

En caso de que la tarjeta Micro SD que vayamos a utilizar contenga datos, es necesario realizar un formateo, incluso aunque la tarjeta la veamos vacía es recomendable formatearla si lleva tiempo sin usarse y no estamos seguros de si la última vez se llegó a formatear o si simplemente se borraron los datos.

Para formatear una tarjeta Micro SD se puede realizar por línea de comandos desde el terminal, o bien mediante programas especializados como **SD Memory Card Formatter**:

https://www.sdcard.org/downloads/formatter_4/index.html

Se recomienda utilizar programas específicos como el arriba indicado, en lugar de ciertas herramientas genéricas de formateo que suelen venir incluidas en todos los sistemas operativos, dado que éstas pueden no estar optimizados para formateo de tarjetas SD/SDHC/SDXC.

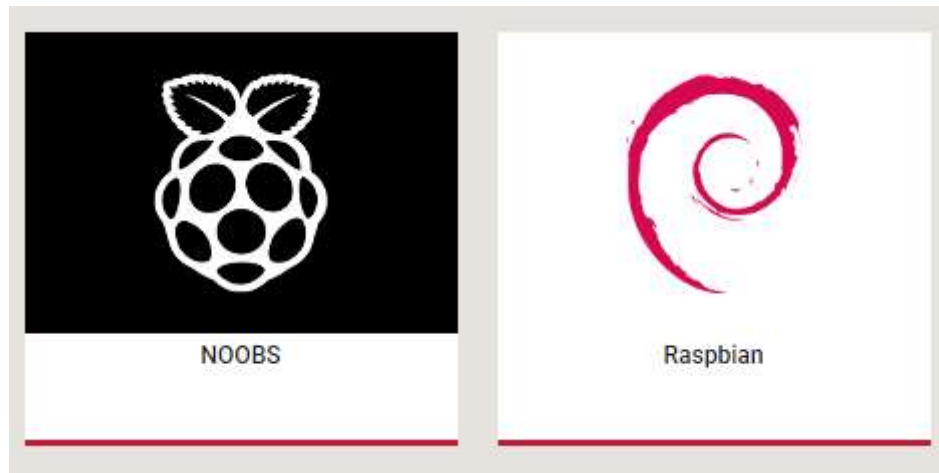


4.3. Descarga del sistema operativo

Podemos descargar el sistema operativo Raspbian desde el apartado de descargas de la web oficial de la Fundación Raspberry Pi:

<https://www.raspberrypi.org/downloads/>

Al acceder nos encontraremos con dos opciones, NOOBS y Raspbian:



En cualquiera de los dos casos el sistema operativo que se va a instalar es Raspbian, lo único que va a cambiar es el método de instalación.

- **Instalación mediante NOOBS:** NOOBS es una aplicación disponible para Raspberry Pi, pensada para facilitar el proceso de descarga e instalación del sistema operativo. Para instalar NOOBS es tan sencillo como copiar en la tarjeta Micro SD el contenido de la carpeta comprimida que obtenemos tras realizar la descarga. Al arrancar la Raspberry Pi la aplicación NOOBS se iniciará, apareciendo un asistente que nos guiará durante el proceso de instalación del sistema operativo.
- **Instalación directa de Raspbian:** Esta opción nos permite descargar directamente la imagen del sistema operativo Raspbian que debe instalarse en la tarjeta Micro SD. También es sencillo pero requiere el uso de un programa adicional que nos permita cargar imágenes de sistemas.

Eligiendo NOOBS se nos presentarán las siguientes opciones:

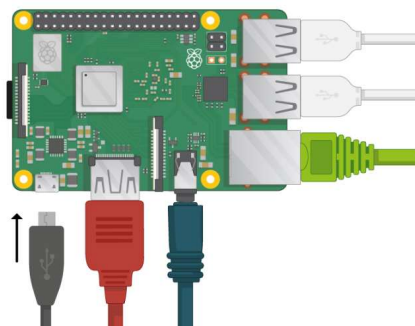


- **NOOBS:** La versión normal de NOOBS viene con el Raspbian precargado, de modo que cuando aparezca el asistente de instalación nos ofrecerá instalar Raspbian directamente sin necesidad de descargarlo. Esta es la mejor opción si lo que queremos es instalar Raspbian en la Raspberry Pi.
- **NOOBS Lite:** La versión Lite viene únicamente con los archivos necesarios para iniciar NOOBS sin ningún sistema operativo precargado, por tanto la carpeta tiene un tamaño mucho menor. Cuando se ejecute el asistente, aparecerán diversas opciones de sistemas operativos, entre ellos Raspbian, para descargar e instalar, pero claramente tendremos que tener la Raspberry Pi conectada a internet para poder realizar la descarga. Esta opción es recomendable solo si tenemos pensado instalar un sistema operativo distinto de Raspbian.

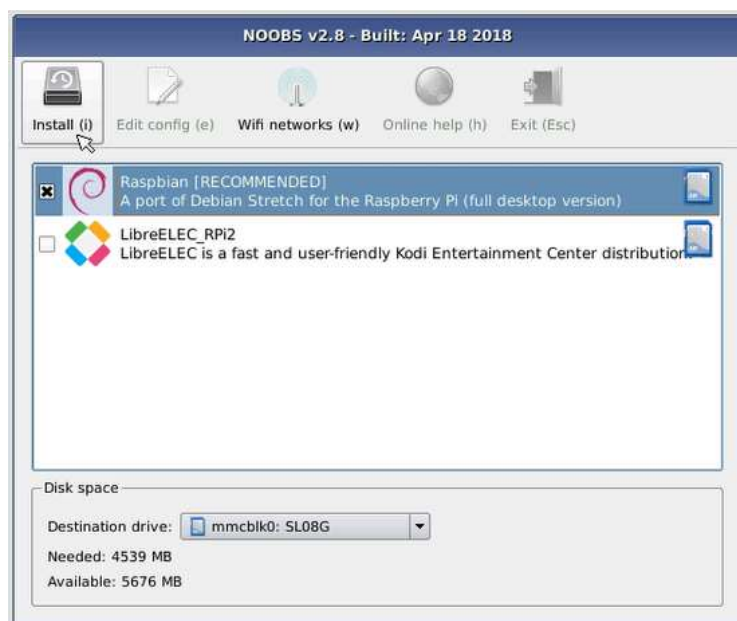
4.4. Instalación de Raspbian con NOOBS

Suponiendo que hemos descargado la versión normal de NOOBS con el propósito de instalar Raspbian en nuestra Raspberry Pi, los demás pasos a seguir serán los que se indican a continuación:

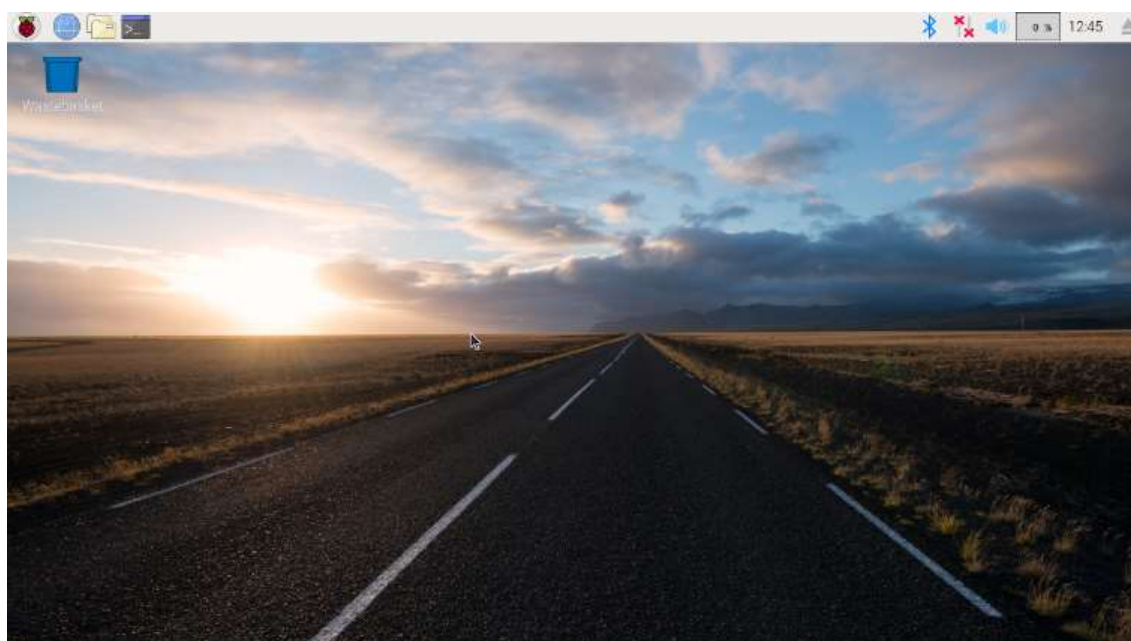
- Descomprimir la carpeta descargada que contiene la aplicación NOOBS.
- Transferir los archivos de dicha carpeta a la tarjeta Micro SD.
- Insertar la tarjeta Micro SD en la Raspberry Pi
- Conectar todos los periféricos a la Raspberry Pi: Monitor, teclado y ratón.
- Después de haber conectado todo lo necesario a la Raspberry Pi (nunca antes), ya podemos alimentar la placa conectando la fuente de alimentación.



- En este momento la Raspberry debería estar arrancando, así que esperamos hasta que nos aparezca el asistente de instalación:



- Seleccionamos el sistema operativo recomendado, Raspbian, y seleccionamos instalar siguiendo los pasos que se nos indiquen a continuación.
- El proceso de instalación puede tardar varios minutos. Una vez finalizado la Raspberry Pi se reiniciará y cuando arranque de nuevo ya podremos disfrutar de nuestro nuevo ordenador con Raspbian instalado.



4.5. Configuración de Raspbian

Como suele ocurrir con cualquier ordenador y sistema operativo, la primera vez que arrancamos el Raspbian después de haber sido instalado, aparecerá un mensaje de bienvenida y un asistente de configuración iniciales del sistema:



Las configuraciones básicas a realizar son muy sencillas:

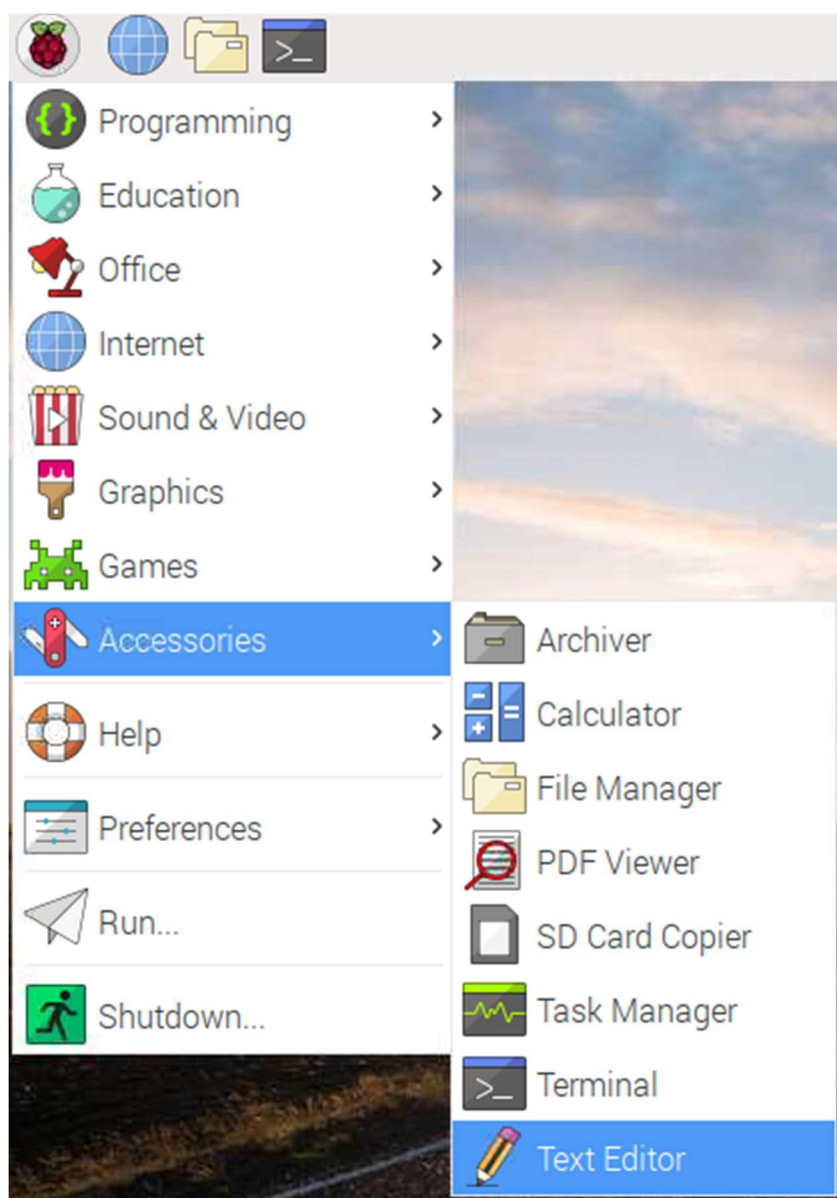
- Selección de país, lenguaje y zona horaria
- Introducir una nueva contraseña
- Buscar y conectar la Raspberry Pi a una red Wi-Fi
- Búsqueda e instalación de actualizaciones

4.6. ¿Qué podemos hacer en Raspbian?

Raspbian es un sistema operativo con una interfaz bastante simplificada, sencilla e intuitiva. En el menú superior izquierda verás cuatro iconos:

- **Menú de inicio:** La frambuesa equivale al botón de inicio de Windows. Al pulsar sobre este icono se despliega un menú vertical desde el que podemos ver y ejecutar los programas instalados, así como acceder a las preferencias del equipo y apagar el sistema.
- **Navegador:** Para abrir el navegador Chromium.

- **Gestor de archivos:** Para abrir el gestor de archivos y acceder a los directorios de carpetas.
- **Terminal:** Para usuarios avanzados, puedes abrir el terminal y controlar el ordenador mediante línea de comandos.



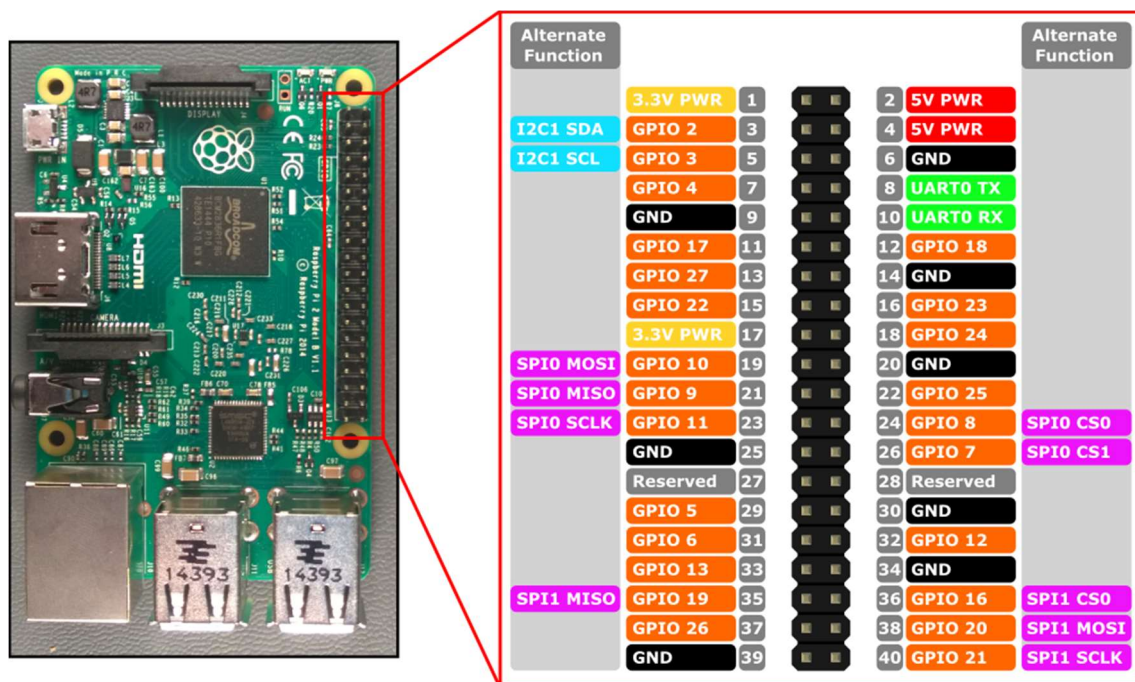
Respecto a los programas que vienen preinstalados, podemos encontrar una variedad bastante significativa de aplicaciones bien organizadas por categorías, con todo lo básico para comenzar a utilizar nuestra Raspberry Pi como cualquier ordenador. Entre estos programas destacan:

- El navegador Chromium
- El paquete de office LibreOffice
- El reproductor multimedia VLC
- El visor de documentos PDF Viewer
- El editor de texto Text Editor

Además, y dado que la Raspberry Pi ha sido especialmente creada para la enseñanza de las ciencias de la computación y para el desarrollo de proyectos de electrónica educativos, Raspbian incluye una selección de programas para aprender y trabajar con programación entre los que destacan **Python** y **Scratch**.

5. Los puertos GPIO

Los GPIO son la ristra de 40 pines ubicados en un lateral de la placa. El termino GPIO (General Purpose Input Output) se refiere a que se trata de puertos de entrada y salida de propósito general, es decir que en principio pueden ser utilizados para cualquier proyecto que queramos, teniendo cuidado siempre de respetar los requisitos y métodos de uso de los dispositivos conectados y de la propia Raspberry Pi.



En la imagen anterior se muestra una agrupación por colores de los distintos pines que conforman el GPIO dependiendo de su función específica:

- **Amarillo:** Pines de alimentación a 3,3V.

- **Rojo:** Pines de alimentación a 5V.
- **Negro:** Pines de masa
- **Naranja:** Pines de propósito general. Pueden ser configurados como entradas y como salidas. Muy importante tener en cuenta que el nivel alto de estos pines es de 3,3V, por lo que no los podemos usar a 5V como podríamos estar acostumbrados a hacer con Arduino, ya que de lo contrario podríamos dañar la Raspberry Pi.
- **Azul:** Pines para comunicación serie mediante protocolo I2C.
- **Verde:** Pines para comunicación serie convencional UART.
- **Violeta:** Pines para comunicación serie mediante protocolo SPI.
- **Gris:** Pines reservados.

Como puede verse en el esquema, algunos de los pines tienen una función alternativa además de su función por defecto.

Otro aspecto a destacar de los pines GPIO es su sistema de numeración, para lo cual existen dos criterios diferentes:

- **Numeración GPIO:** Los pines se enumeran según el orden en el que se encuentran ubicados en la placa, tal y como se muestran representados en el esquema anterior mediante recuadros grises.
- **Numeración BCM:** La enumeración de los pines del GPIO se corresponde con la de los pines del procesador Broadcom.

Es muy importante conocer ambas numeraciones para evitar confundirlas cuando revisemos información de proyectos de terceros.

Finalmente es importante conocer la limitación en corriente de los siguientes pines GPIO para saber lo que podemos y lo que nos podemos alimentar desde los pines:

- **Pines 3,3V:** Máximo 50 mA.
- **Pines de propósito general:** 3 mA.
- **Pines 5V:** La capacidad de la fuente menos lo que consuma la Raspberry Pi. Normalmente podremos obtener en torno a 1,5 A.

6. Programando con Scratch

6.1. Introducción

Recordemos que Scratch es un programa ideado para que alumnos de primaria, secundaria y principiantes en general, puedan aprender a programar de una forma sencilla e intuitiva, por medio de un lenguaje de programación por bloques.



Scratch se emplea principalmente para el diseño de animaciones y videojuegos. Sin embargo existen versiones adaptadas para ser compatibles con hardware externo, como Arduino, Raspberry Pi, mBot y Makey Makey entre otros, de forma que también podemos usar Scratch para programar microcontroladores y ordenadores.

6.2. Añadir extensiones

Antes de nada abrimos el programa Scratch 2.0, que ya viene preinstalado en Raspbian y que podrás encontrar en el menú de Programación.

Una vez abierto, en la sección de Programas encontraremos diversas categorías de bloques disponibles para programar. Sin embargo y aunque por defecto hay muchísimos bloques, ninguno de ellos nos permite controlar los pines GPIO de la Raspberry Pi, de modo que necesitaremos instalar extensiones específicas para ello.



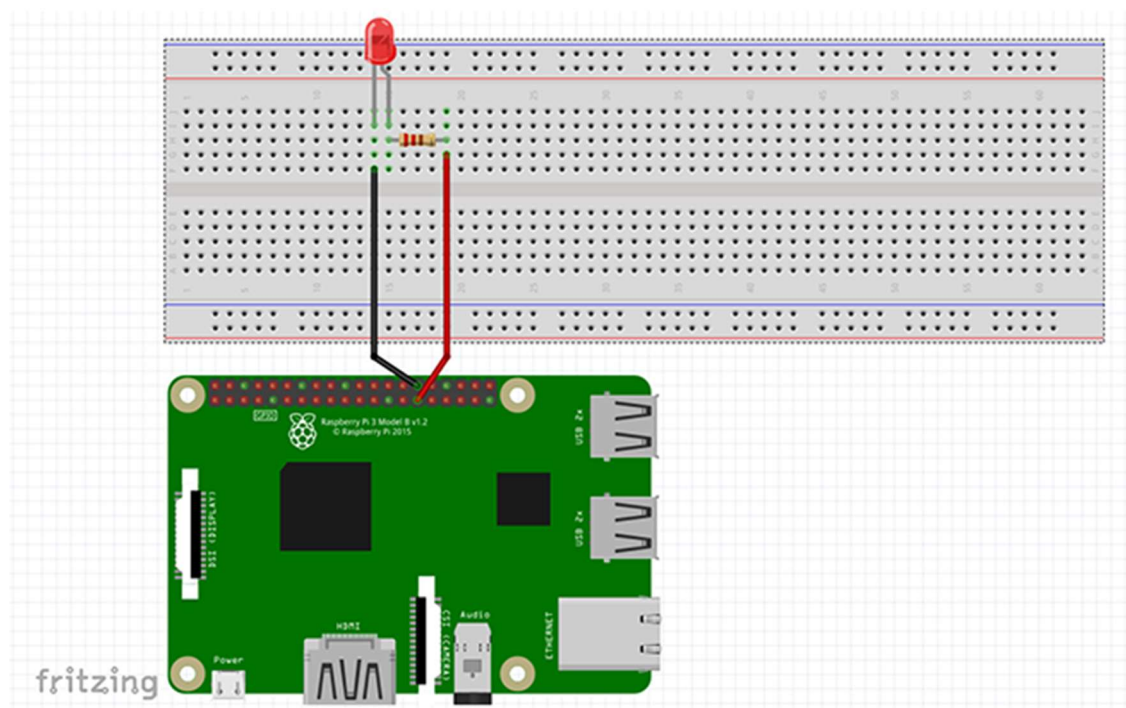
Lo que haremos será seleccionar **Más Bloques** y **Añadir una Extensión**. Automáticamente se nos abrirá la siguiente ventana que nos ofrece dos extensiones específicas para Raspberry Pi, la primera para poder controlar los **GPIO** y la segunda para manejar el módulo **Pi SenseHat**, y puesto que ambas son fantásticas instalamos las dos:



Hecho esto deberíamos de disponer de nuevos bloques dentro de la categoría **Más Bloques**.

6.3. Circuito: Encendido de un LED

Para familiarizarnos con el control de los pines GPIO desde Scratch vamos a ver un ejemplo sencillo, hacer que un LED se apague y se encienda de forma intermitente. El esquema del circuito electrónico sería el que se muestra a continuación:



Como puede verse en el esquema, únicamente necesitamos dos pines de la placa, uno cualquiera de GND de entre los 8 que hay disponibles para lo cual en este caso hemos código el pin 30, y un pin digital para controlar el encendido y el apagado del LED, es este caso hemos tomado el pin 29 (GPIO 5).

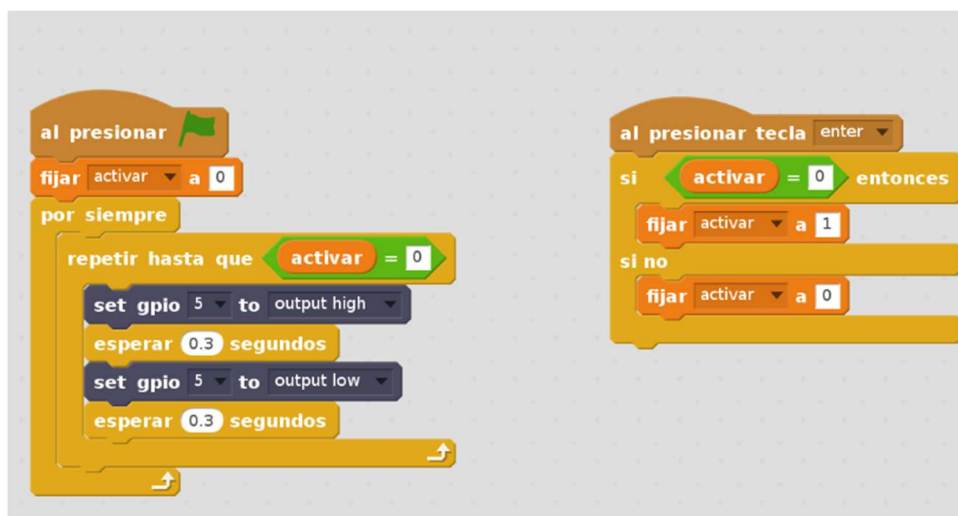
Es importante recordar que en general no podremos utilizar los pines digitales para controlar y alimentar los dispositivos conectados, dado que la capacidad de alimentación de los GPIO es de apenas 3 mA, suficiente para encender un LED, pero insuficiente para la mayoría de los casos. Generalmente utilizaremos los pines GPIO solo para control y los pines de 5V para alimentación. De necesitar aún más corriente de la que nos pueden entregar los pines de 5V, tendremos que utilizar una fuente de alimentación externa.

6.4. Control GPIO con Scratch

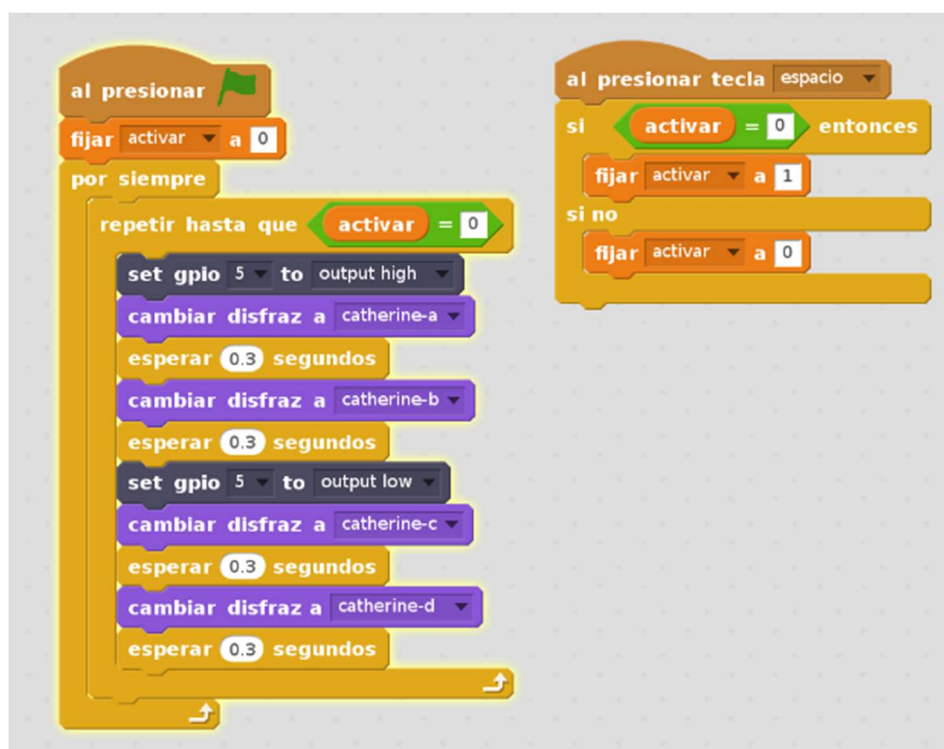
Empezaremos con un código que permita encender y apagar un LED de forma intermitente según los intervalos de tiempo que le indiquemos, y de manera ininterrumpida cada vez que ejecutemos el programa (botón con bandera verde). El código correcto sería el siguiente:



Para complicarlo un poco, vamos a realizar una modificación que consista en activar y desactivar el funcionamiento del LED anterior, cada vez que pulsemos la tecla ENTER del teclado. El código quedaría de la siguiente manera:



Ahora vamos a añadir algo divertido. Modificaremos el fondo por un escenario un poco más colorido y añadiremos un personaje que disponga de disfraces suficientes para crear una animación en la que se vea bailando. Combinaremos esto con lo que ya teníamos del LED, para que a la vez que el LED funcione de forma intermitente cada vez que pulsemos la tecla ENTER, también el personaje se ponga a bailar. Cuando volvamos a pulsar la tecla el LED dejará de parpadear y el personaje cesará de bailar. Un ejemplo de código para esto sería el siguiente:



7. Programando con Python

7.1. Introducción

Python es el lenguaje de programación más utilizado en Raspberry Pi, y de hecho Raspbian incluye por defecto la última versión instalada del software, el Python 3.

La principal característica de Python y en lo que basa su filosofía, es que la sintaxis del código es fácilmente legible, simplificada y sencilla de aprender.

A diferencia de otros lenguajes populares para aprender a programar, Python tiene la gran ventaja de que además de ser un lenguaje muy recomendable para aprender, también es uno de los más utilizados a nivel profesional, comunmente para programación web, desarrollo de aplicaciones, programación de hardware, automatización de tareas, inteligencia artificial, desarrollo de videojuegos, etc...



Programming for the Raspberry Pi

Por tanto, como lenguaje de programación Python no solo es el principio sino también el fin, muy indicado para aprender y al mismo tiempo uno de los lenguajes más prometedores profesionalmente para el presente y para el futuro.

Otra ventaja de Python es su comunidad de usuarios y la inmensa cantidad de recursos y librerías disponibles, que nos ayudan a realizar tareas complejas sin necesidad de programar todo desde cero.

En el caso particular de Raspberry Pi, la principal utilidad de Python es la capacidad que nos ofrece de controlar los dispositivos conectados a los pines GPIO, pudiendo conectar así el medio físico con el mundo digital.

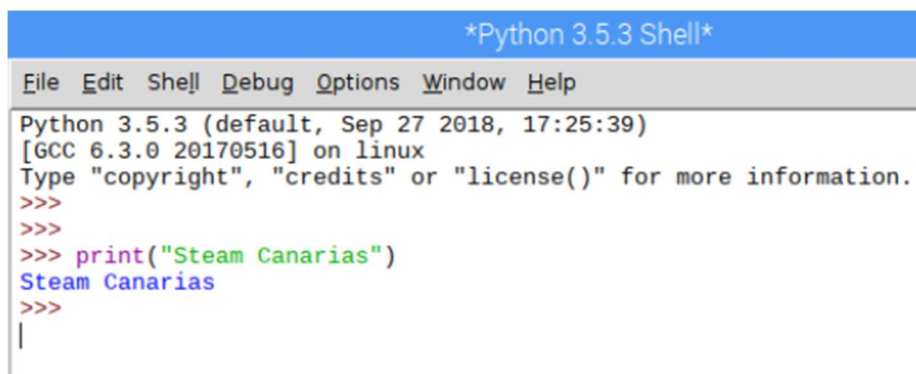
7.2. Entorno de desarrollo

Existen múltiples entornos de desarrollo disponibles para programar en Python. Nosotros veremos los dos que vienen preinstalados por defecto en Raspbian:

- **Python 3 (IDLE)**
- **Thonny Python IDE**

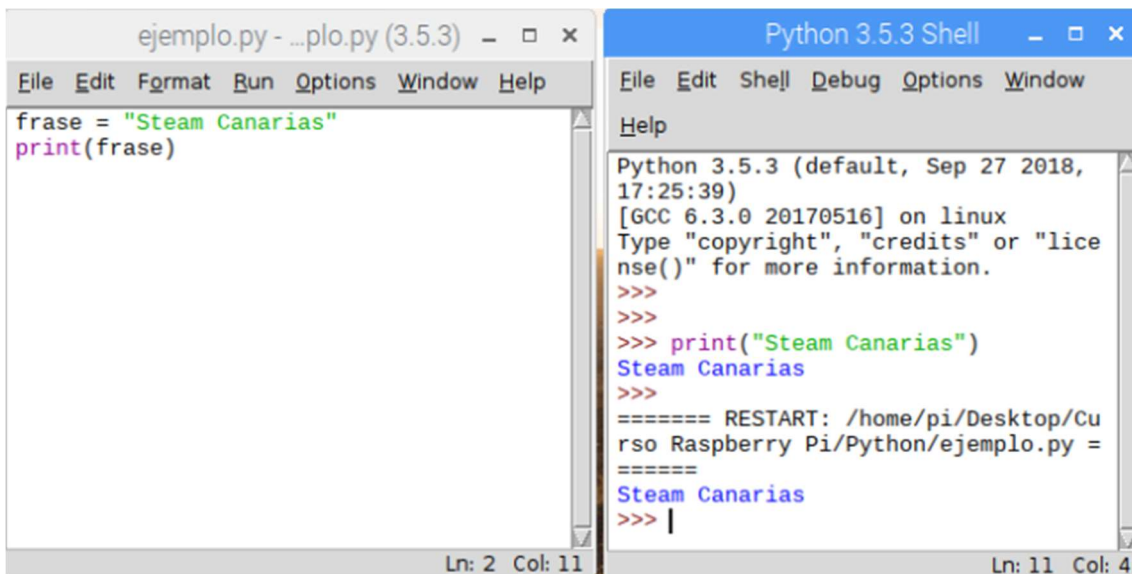
Para familiarizarnos con ambos programas vamos a empezar realizando el ejemplo más sencillo que se puede hacer, imprimir una frase por pantalla.

El **Python 3 (IDLE)** por defecto permite ejecutar una a una cada línea de código que se introduzca en el editor. A continuación se ve como efectivamente cuando ejecutamos la instrucción “**print**” para imprimir por pantalla el contenido de la función, se muestra la frase indicada:



```
*Python 3.5.3 Shell*
File Edit Shell Debug Options Window Help
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>> print("Steam Canarias")
Steam Canarias
>>>
|
```

Ejecutar líneas de código una a una puede estar bien para aprender y practicar, pero por lo general lo interesante es generar archivos de código o scripts en Python, es decir, pequeños programas que luego podamos utilizar y ejecutar de una sola vez. Para hacer esto desde Python 3 (IDLE), tenemos que abrir el editor de texto desde **File – New File**. Generamos el código que queramos, lo guardamos y lo ejecutamos pulsando sobre la opción **Run**. Podremos ver el resultado desde la línea de comandos:



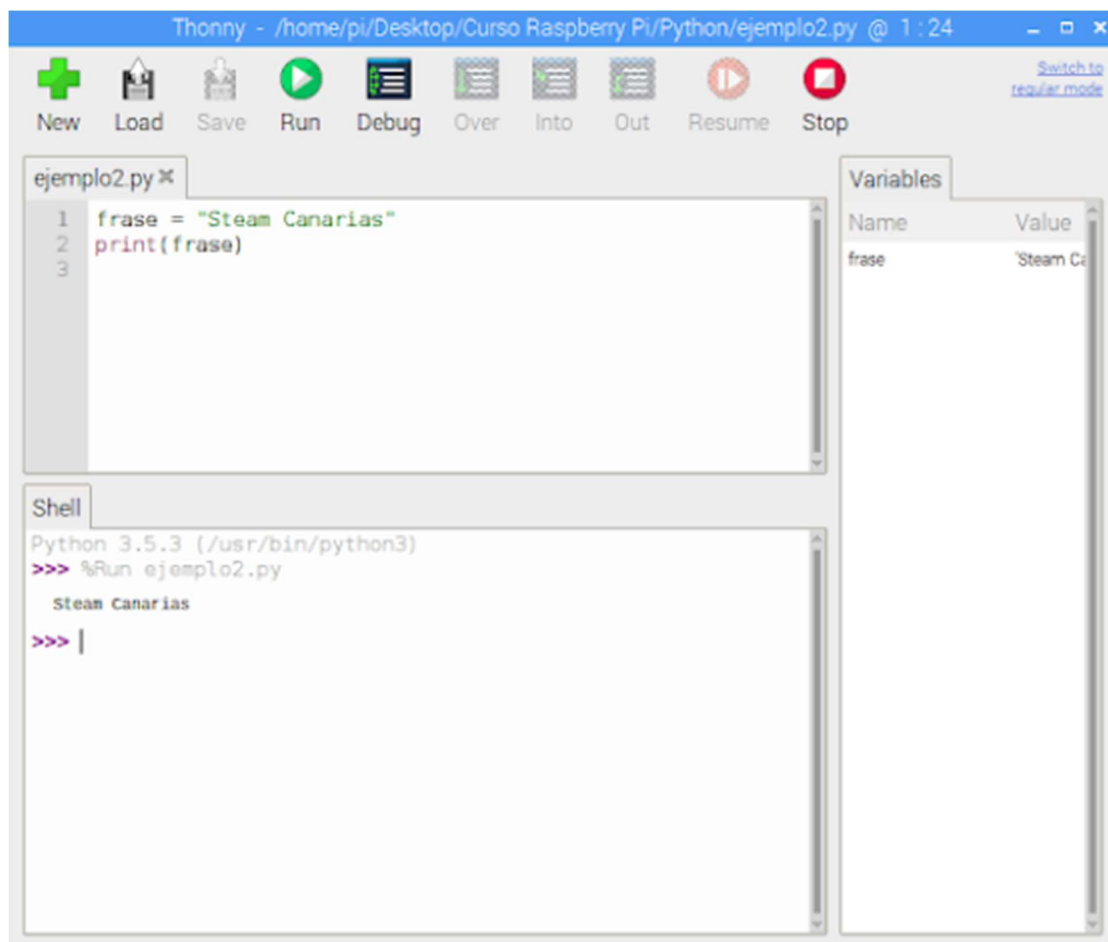
The image shows two windows from the Thonny Python IDE. The left window, titled 'ejemplo.py - ...plo.py (3.5.3)', contains the following code:

```
frase = "Steam Canarias"
print(frase)
```

The right window, titled 'Python 3.5.3 Shell', shows the execution output:

```
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
>>> print("Steam Canarias")
Steam Canarias
>>>
===== RESTART: /home/pi/Desktop/Curso Raspberry Pi/Python/ejemplo.py =====
>>> print("Steam Canarias")
Steam Canarias
>>> |
```

Para hacer lo mismo desde la aplicación **Thonny Python IDE**, es aún más sencillo ya que por defecto nos aparece el editor de texto en una ventana superior y la línea de comandos en la ventana inferior. Introducimos el mismo programita anterior para mostrar una frase por pantalla, guardamos el archivo, ejecutamos el código pulsando el botón verde **Run**, y veremos como el resultado aparece en la ventana inferior:



Antes de finalizar este apartado vamos a ver también como ejecutar scripts de Python desde el terminal. Con el terminal abierto, lo primero es buscar el directorio en el que se encuentra el script de Python que deseamos ejecutar. Lo siguiente es ejecutar la siguiente instrucción:

python3 nombre_archivo.py

En la siguiente imagen se muestra cómo ejecutar el mismo ejemplo que hemos visto con las aplicaciones anteriores, directamente desde el terminal:

```

pi@raspberrypi: ~/Desktop/Curso Raspberry Pi/Python
Archivo  Editar  Pestañas  Ayuda
pi@raspberrypi:~ $ ls
2019-03-27-184835_1824x984_scrot.png  Desktop  MagPi  Public
2019-03-27-185155_1824x984_scrot.png  Documents Music  Templates
2019-03-27-185345_1824x984_scrot.png  Downloads Pictures Videos
pi@raspberrypi:~ $ cd D
Desktop/ Documents/ Downloads/
pi@raspberrypi:~ $ cd Desktop/
pi@raspberrypi:~/Desktop $ ls
Curso Raspberry Pi
pi@raspberrypi:~/Desktop $ cd Curso\ Raspberry\ Pi/
pi@raspberrypi:~/Desktop/Curso Raspberry Pi $ ls
Python Scratch
pi@raspberrypi:~/Desktop/Curso Raspberry Pi $ cd Python/
pi@raspberrypi:~/Desktop/Curso Raspberry Pi/Python $ ls
ejemplo2.py ejemplo.py
pi@raspberrypi:~/Desktop/Curso Raspberry Pi/Python $ python3 ejemplo.py
Steam Canarias
pi@raspberrypi:~/Desktop/Curso Raspberry Pi/Python $

```

7.3. Control GPIO con Python

Vamos a ver el mismo ejemplo del encendido y apagado del LED que hicimos con Scratch, pero mediante un programita realizado en Python. Se puede utilizar cualquiera de los entornos de desarrollo vistos anteriormente. En este caso nosotros vamos a comenzar utilizando la aplicación de **Thonny Python IDE**:

```
LED_simple.py x
1 import RPi.GPIO as GPIO      # Importamos el módulo (librería) GPIO del
2                               # paquete RPi y le damos el nombre GPIO al
3                               # conjunto RPi.GPIO. Este módulo nos permite
4                               # controlar el puerto GPIO de la Raspberry Pi
5 import time                  # Importamos el módulo "time" para hacer uso
6                               # de las funciones de tiempo
7 LED = 29                     # Declaramos una variable LED con valor
8                               # entero 29
9
10 GPIO.setmode(GPIO.BOARD)     # Función "setmode()" del módulo GPIO para
11                               # configurar el puerto GPIO en modo BOARD
12 GPIO.setup(LED, GPIO.OUT)     # Función "setup" del módulo GPIO para
13                               # Configurar el pin LED en modo salida
14
15 while True:                  # Bucle infinito
16     GPIO.output(LED, True)    # Pin LED en HIGH
17     time.sleep(1)             # Esperar 1 segundo
18     GPIO.output(LED, False)   # Pin LED en LOW
19     time.sleep(1)             # Esperar 1 segundo
20
```

El ejemplo anterior ejecuta un bucle infinito de encendido y apagado de un LED a intervalos de un segundo.

Como no hemos previsto en el código que el programa deba finalizar mediante alguna condición, en el momento en el que intentemos ejecutar el mismo código de nuevo u otro diferente, nos saldrá un “Warning” avisando de que los GPIO están en uso.

Para evitar lo anterior, la solución sería prever que la ejecución del programa pueda finalizar de alguna manera, y antes de terminar ejecutar la instrucción **GPIO.cleanup()**, que resetea los pines GPIO devolviéndolos a su configuración por defecto.

Una posible forma de hacer esto lo vemos en el siguiente ejemplo, en el que hemos sustituido el bucle infinito por un bucle de seis iteraciones, con un **cleanup()** antes de finalizar el programa:

```

LED_simple.py x LED_simple_2.py x LED_simple_3.py x
1  import RPi.GPIO as GPIO      # Importamos el módulo (librería) GPIO del
2                                # paquete RPi y le damos el nombre GPIO al
3                                # conjunto RPi.GPIO. Este módulo nos permite
4                                # controlar el puerto GPIO de la Raspberry Pi
5  import time                  # Importamos el módulo "time" para hacer uso
6                                # de las funciones de tiempo
7  LED = 29                     # Declaramos una variable LED con valor
8                                # entero 29
9
10 GPIO.setmode(GPIO.BOARD)     # Función "setmode()" del módulo GPIO para
11                               # configurar el puerto GPIO en modo BOARD
12 GPIO.setup(LED, GPIO.OUT)     # Función "setup" del módulo GPIO para
13                               # Configurar el pin LED en modo salida
14
15 for i in range(0, 5):        # Bucle de 0 a 5
16     GPIO.output(LED, True)    # Pin LED en HIGH
17     time.sleep(1)             # Esperar 1 segundo
18     GPIO.output(LED, False)   # Pin LED en LOW
19     time.sleep(1)             # Esperar 1 segundo
20
21 GPIO.cleanup()               # Devuelve la configuración de los pines GPIO
22                               # a su estado original por defecto

```

Antes de terminar con el ejercicio vamos a ver un último caso para aprender a realizar lecturas de teclado. Con la función **input()** podemos hacer que el programa se detenga para solicitar una pulsación de teclado.

En el siguiente código de ejemplo hemos aprovechado esta función para que el programa nos pregunte si queremos o no volver a encender el LED después de cada iteración. En caso de pulsar la 's' de Sí entonces el programa se repite, en caso contrario el programa termina ejecutando un **cleanup()**.

LED_simple.py ✖	LED_simple_2.py ✖	LED_simple_3.py ✖
<pre> 1 import RPi.GPIO as GPIO 2 3 4 5 import time 6 7 LED = 29 8 9 10 GPIO.setmode(GPIO.BOARD) 11 12 GPIO.setup(LED, GPIO.OUT) 13 14 15 tecla = 's' 16 17 while tecla == 's': 18 GPIO.output(LED, True) 19 time.sleep(1) 20 GPIO.output(LED, False) 21 time.sleep(1) 22 print("¿Volver a encender?") 23 tecla = input() 24 25 GPIO.cleanup() 26 </pre>		<pre> # Importamos el módulo (librería) GPIO del # paquete RPi y le damos el nombre GPIO al # conjunto RPi.GPIO. Este módulo nos permite # controlar el puerto GPIO de la Raspberry Pi # Importamos el módulo "time" para hacer uso # de las funciones de tiempo # Declaramos una variable LED con valor # entero 29 # Función "setmode()" del módulo GPIO para # configurar el puerto GPIO en modo BOARD # Función "setup" del módulo GPIO para # Configurar el pin LED en modo salida # Se repite el bucle mientras se pulse 's' # Pin LED en HIGH # Esperar 1 segundo # Pin LED en LOW # Esperar 1 segundo # Imprime por pantalla el texto indicado # Solicita un valor de teclado # Devuelve la configuración de los pines GPIO # a su estado original por defecto </pre>