

# Node-RED - Python

## (Agosto de 2020)

Sebastián Cardenas y Rafael Fonseca

### Resumen

Node-RED es una herramienta de código libre (Open Source) construida en Node.js y que se encuadra en la familia de herramientas "flow-based programming".

La programación basada en flujos es una forma de describir el comportamiento de una aplicación como una red de cajas negras o "nodos", como se les llama en Node-RED. Cada nodo tiene un propósito bien definido: recibe o captura algunos datos, realiza algún tratamiento con esos datos y luego los pasa a uno o varios nodos con los que se enlaza.

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

## I. INTRODUCCIÓN

### NODE-RED

Node-RED, es una herramienta de código libre (Open Source) construida en Node.js y que se encuadra en la familia de herramientas "flow-based programming (FBP) tools" ('Herramientas de programación basadas en flujos'). Originalmente desarrollado por el equipo de Emerging Technology Services de IBM, es ahora parte de la OpenJS Foundation, cuenta con una creciente comunidad de desarrollo muy activa y goza actualmente de un gran predicamento y una amplia difusión de aplicación en multitud de proyectos innovadores.

Node-RED es uno de los componentes clave de dicha arquitectura de solución. Como herramienta visual, facilita extraordinariamente la captura de eventos del mundo real, permite agregar cierto grado de inteligencia en nodos de tratamiento/transformación de datos y utilizar nodos especializados para integrar dichos eventos con todo tipo de sistemas de mensajería, como por ejemplo MQTT, AMQP o Apache Kafka. También es posible en plataformas sociales como Twitter o Facebook y sistemas de persistencia de datos como Bases de Datos Relacionales y NoSQL, como MongoDB y Redis.

Todo ello se lleva a cabo con la finalidad de crear aplicaciones que puedan reaccionar de forma ágil y precisa ante el complejo mundo existente a su alrededor, y aportar la "materia prima elaborada" a los otros componentes de la arquitectura global de tratamiento de la información.

La programación basada en flujos es una forma de describir el comportamiento de una aplicación como una red de cajas negras o "nodos", como se les llama en Node-RED. Cada nodo tiene un propósito bien definido: recibe o captura algunos datos, realiza algún tratamiento con esos datos y luego los pasa a uno o varios nodos con los que se enlaza. La red constituye la configuración de cómo estos nodos se interconectan entre sí y es responsable del mantenimiento del flujo de datos entre los nodos

### PYTHON

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.

Además, Python es gratuito, incluso para propósitos empresariales.

### Características del lenguaje

#### ➤ Propósito general

Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.

#### ➤ **Multiplataforma**

Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.

#### ➤ **Interpretado**

Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.

#### ➤ **Interactivo**

Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

#### ➤ **Orientado a Objetos**

La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

#### ➤ **Funciones y librerías**

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.

#### ➤ **Sintaxis clara**

Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave begin y end. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

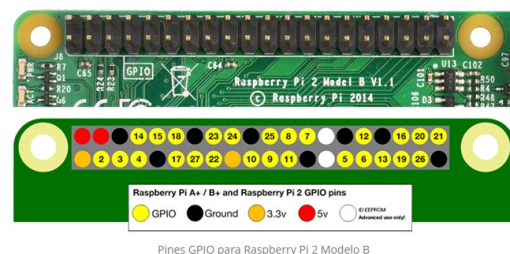


### **RASPBERRY Y PINES GPIO**

La Raspberry Pi es una computadora de bajo costo y con un tamaño compacto, del porte de una tarjeta de crédito, puede ser conectada a un monitor de computador o un TV, y usarse con un mouse y teclado estándar. Es un pequeño computador que corre un sistema operativo linux capaz de permitirle a las personas de todas las edades explorar la computación y aprender a programar lenguajes como Scratch y Python. Es capaz de hacer la mayoría de las tareas típicas de un computador de escritorio, desde navegar en internet, reproducir videos en alta resolución, manipular documentos de ofimática, hasta reproducir juegos.

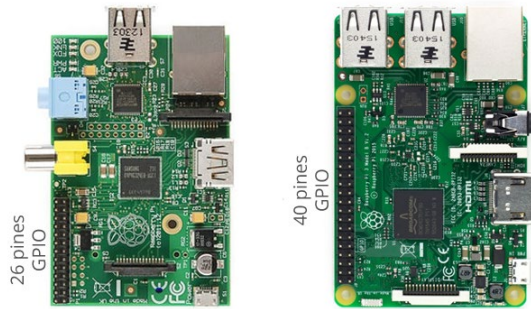
Además la Raspberry Pi tiene la habilidad de interactuar con el mundo exterior, puede ser usada en una amplia variedad de proyectos digitales, desde reproductores de música y video, detectores de padres, estaciones meteorológicas hasta cajas de aves con cámaras infrarrojas. Queremos que veas que la Raspberry Pi puede ser usada por niños y adultos por todas partes del mundo, para aprender a programar y entender cómo funcionan las computadoras.

General Purpose Input Output (GPIO) es un sistema de entrada y salida de propósito general, es decir, consta de una serie de pines o conexiones que se pueden usar como entradas o salidas para múltiples usos. Estos pines están incluidos en todos los modelos de Raspberry Pi aunque con diferencias.



Hay que tener en cuenta que dependiendo del modelo de la Raspberry Pi encontramos una cantidad de pines diferentes, por ejemplo, en la versión 1 de Raspberry Pi se tienen 26 pines GPIO mientras que a

partir de la versión 2 de Raspberry Pi el número de pines aumentó a 40. Sin embargo, la compatibilidad es total, puesto que los 26 primeros pines mantienen su función original.



Pines GPIO en Raspberry Pi Modelo A y Raspberry Pi 2 Modelo B

Como se puede observar, el número de pines pasó de 26 a 40 para tener más disponibilidad, aunque volvemos a comentar que los 26 primeros pines son comunes para todas las versiones. Los pines GPIO tienen funciones específicas (aunque algunos comparten funciones) y se pueden agrupar de la siguiente manera:

- Amarillo (2): Alimentación a 3.3V.
- Rojo (2): Alimentación a 5V.
- Naranja (26): Entradas / salidas de propósito general. Pueden configurarse como entradas o salidas. Ten presente que el nivel alto es de 3.3V y no son tolerantes a tensiones de 5V.
- Gris (2): Reservados.
- Negro (8): Conexión a GND o masa.
- Azul (2): Comunicación mediante el protocolo I2C para comunicarse con periféricos que siguen este protocolo.
- Verde (2): Destinados a conexión para UART para puerto serie convencional.
- Morado (5): Comunicación mediante el protocolo SPI para comunicarse con periféricos que siguen este protocolo.
- Existen 2 formas de numerar los pines de la Raspberry Pi, en modo GPIO o en modo BCM.
- En el modo GPIO, los pines se numeran de forma física por el lugar que ocupan en la placa (representados por el color gris) viene siendo igual para todas las versiones (comenzamos a contar desde arriba a la izquierda y finalizamos abajo a la derecha).
- En el modo BCM, los pines se numeran por la correspondencia en el chip Broadcom (que es la CPU de la Raspberry Pi).

| BOARD | GPIO                 |  | GPIO                 | BOARD |
|-------|----------------------|--|----------------------|-------|
| 01    | 3.3v DC Power        |  | DC Power 5v          | 02    |
| 03    | GPIO02 (SDA1 , PC)   |  | DC Power 5v          | 04    |
| 05    | GPIO03 (SCL1 , PC)   |  | 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 (PC ID EEPROM) |  | (PC 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    |

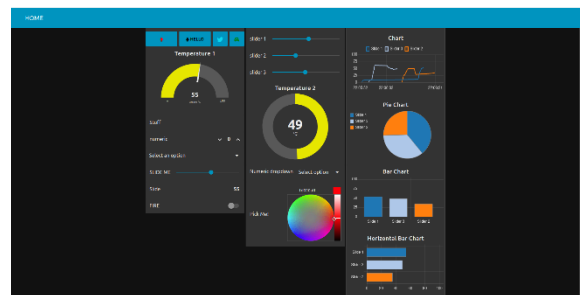
Nomenclatura GPIO y BCM

De los pines GPIO disponibles, hay una serie de pines con capacidad de PWM (como volveremos más adelante). Sin embargo no se dispone de ningún convertidor de analógico a digital. Esto quiere decir que para medir valores de sensores analógicos necesitaremos utilizar un convertidor externo o un Arduino en la mayoría de los casos.

## EXPLICACION DEL CODIGO FUENTE

### Node-RED:HMI

Para la interfaz HMI se uso diferentes nodos, como botones, ingresar texto, lista de opciones, gráficos los cuales pueden ser: barras - lineales - pastel, tambien nodos de calibre, un switch, incluso la posibilidad de escoger una tonalidad de color en una paleta de colores que esta a disposición; para poder visualizar nuestra interfaz añadimos las letras `*/ui*` a la IP que estamos trabajando en una nueva ventana.



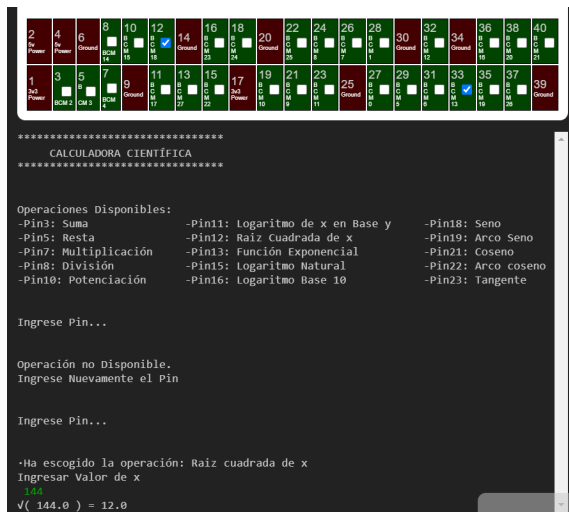
### Calculadora Científica

La calculadora usando los pines GPIO para definir la operacion que se va a realizar, se uso la libreria correspondiente para el manejo de los pines, la libreria math para las funciones matematicas, tambien una libreria time.

Al ejecutar el programa nos aparece los todos los pines GPIO en color verde, sigido de una lista el cual nos muestra la operación que realiza un pin, si el pin

seleccionado esta dentro de la lista de opciones se puede seguir con la operación que queramos hacer, pero si el pin no esta dentro de la lista, va a aparecer un mensaje de que la operación no esta disponible y que debe de ingresar nuevamente el pin.

Una vez escogido el pin dentro de las opciones disponibles, nos va a pedir un valor para x o y dependiendo la operación a realizar, al ingresar los valores nos muestra la operación que hicimos con su respectivo resultado y el program finaliza.



#### IV. Resultados

A partir del uso de simuladores se logro realizar una calculadora científica utilizando los pines GPIO de una Raspberyy Pi para esciger que tipo de operación queremos, que nos puede ayudar con diferentes tipos de operación que necesistemos y que estén disponibles y construir una interfaz Humano Mauina para que sea mas amigable al momento de interactuar con Node-RED

#### Conclusión

Se diseño un programa en NODE-RED que implemente una interfaz HMI (Interfaz Humano Máquina).

Se diseño una calculadora científica en Python empleando los conceptos de programación orientada a objetos y la librería Math

#### Recomendaciones

Como recomendaciones se tiene que ser muy cuidadoso al momento de escribir el código en python ya que si omitimos una letra o un signo, puede ocasionar errores al momento de ejecutar el simulador, el uso de puntos de control es recomendable al momento de escribir cualquier código en diferentes lenguajes de programación ya

que nos permite saber si en esa sección del código esta mal o esta bien.

#### Referencias

- [1] Pi, R. (2015). Raspberry pi 3 model b. online].(<https://www.raspberrypi.org>)
- [2] GPIO - Raspberry Pi Documentation. (s. f.). Raspberry Pi. Recuperado 2 de agosto de 2020, de <https://www.raspberrypi.org/documentation/usage/gpio/>
- [3] Python Software Foundation. (s. f.). Python.org. Recuperado 2 de agosto de 2020, de <https://www.python.org/psf/>
- [4] About : Node-RED. (s. f.). Node-RED. Recuperado 2 de agosto de 2020, de <https://nodered.org/about/>
- [5] node-red-dashboard. (s. f.). Node-RED. Recuperado 2 de agosto de 2020, de <https://flows.nodered.org/node/node-red-dashboard>