

# Interfaz humano-máquina y Python, programación orientada a objetos con puertos virtuales Raspberry Pi

K. Peña, E. Simbaña, MJ. Vizuet

## Resumen

La Interacción Humano-Computadora se convirtió oficialmente en una disciplina con el advenimiento de la computadora personal. Con el primer Macintosh, IBM PC 5150 y Commodore 64 utilizados en la oficina, la gente empezó a darse cuenta de cómo esta transición afectará no sólo a su trabajo sino a sus vidas en general, por lo que el desarrollo de interfaces que faciliten esta interacción deberían ser primordiales en su desarrollo.

## I. INTRODUCCIÓN

El objeto de investigación de este artículo es comprender el modo en que se realiza una interfaz humano-máquina a través de un dashboard elaborado en Node-Red a fin de que el usuario pueda interactuar directamente con el tablero. Además conocer el paradigma de programación orientado a objetos en el lenguaje Python, para su posterior implementación en el desarrollo de una calculadora científica con entradas de puertos virtuales correspondientes a una Raspberry pi.

## II. MARCO TEÓRICO

### A. Node-RED

Node-RED es una herramienta de programación para conectar dispositivos de hardware, API y servicios en línea de formas nuevas e interesantes. Proporciona un editor basado en navegador que facilita la conexión de flujos utilizando la amplia gama de nodos en la paleta que se pueden implementar en su tiempo de ejecución con un solo clic[1]. Además cuenta con un dashboard o tablero de instrumentos que permite crear y visualizar varios widgets para interactuar con los dispositivos del internet de las cosas.

### 1) Uso del panel de control Node-Red

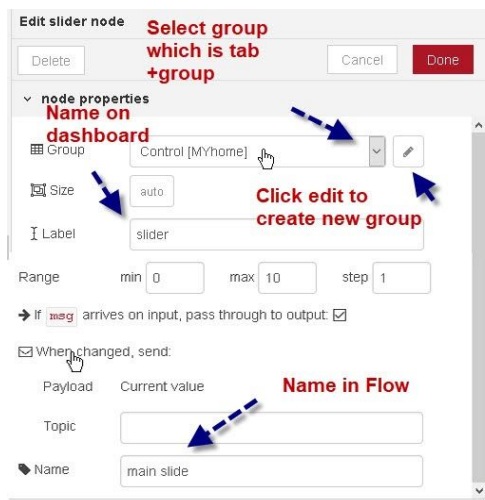
El diseño de la pantalla de la interfaz de usuario se puede controlar en la parte izquierda del tablero, donde se pueden encontrar los siguientes nodos:



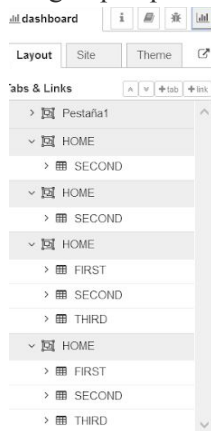
Para que en un programa existan varias pestañas se debe llamar a los nodos en diferentes grupos, además para generar subdivisiones dentro de estos, se nombran en distintas etiquetas. Con esto se logra una mejor visualización[2].

Cada nodo del tablero tiene tres configuraciones importantes las cuales son:

- Group/grupo: Aquí se va a contener el nodo dentro del dashboard, si se va a implementar distintos nodos para un mismo objetivo, lo mejor sería colocarlos en el mismo grupo.
- Label/ Etiqueta: Se asigna el nombre que va a tener cada uno de los nodos al ser mostrados.
- Tooltip/ Información sobre herramientas: Se escribe una indicación al usuario, esta aparecerá al acercar el cursor al componente.
- Nombre : Nombre que aparecerá en el trabajo de flujo.



En la parte derecha de la interfaz, podemos encontrar la organización de los grupos y sus respectivas etiquetas, con esto se puede cambiar el orden y elegir a nuestro gusto el widget que queremos visualizar primero.



## 2)Nodos más importantes en la dashboard:

- Button: Agrega un botón a la interfaz del usuario, el cual al ser presionado debe dar como resultado lo que se indique. Los datos que se van a devolver pueden ser de tipo string, number, boolean, JSON, buffer, timestamp, flow o global.
- Gauge: Este nodo muestra un widget de tipo de indicador en el tablero. Formateará los números pasados a través de msg.payload en un widget de estilo indicador. El medidor contiene diferentes sectores y la aguja se moverá en distintas direcciones según el rango y el valor pasado al widget a través de msg.payload.
- Numeric: Agrega un widget de entrada numérico a la interfaz de usuario, aquí se puede escoger el rango de números que se desee.

- Slider: Agrega un control deslizante manipulable a la interfaz de usuario.
- Switch: Agrega un interruptor a la interfaz del usuario. Este incluye dos botones en su configuración, los cuales van a retornar dos estados en caso de ser presionados. Pueden ser de tipo entero, string, boolean, y entre otros.
- Dropdown: Agrega un cuadro desplegable a la interfaz del usuario. Se escribe un texto que indique lo que debe seleccionar entre las opciones. También es posible escoger si se puede responder una sola opción o todas. Se escoge el tipo de dato que se va a ingresar, puede ser de tipo number, string o boolean, se añade el número de opciones que se quiera y en label se ingresa la opción.
- Colour picker: Agrega un selector de color al tablero. Entre sus formatos se puede escoger entre hx, hx8, hsl, hsv, rgb.
- Chart: Este nodo agrega un gráfico al tablero. El gráfico utiliza valores enviados a través de msg.payload para su visualización

## B.¿Qué es Raspberry Pi?

Es un ordenador del tamaño de una tarjeta de crédito de bajo coste y tamaño reducido, creada con el objetivo de estimular la enseñanza de la informática en las escuelas, aunque no empezó su comercialización hasta el año 2012.

Consta de una placa base sobre la que se monta un procesador, un chip gráfico y memoria RAM. Esta placa que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal.

Su concepto es el de un ordenador desnudo de todos los accesorios que se pueden eliminar sin que afecte al funcionamiento básico.[3]

## 3)Los pines GPIO (General Purpose Input Output):

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.

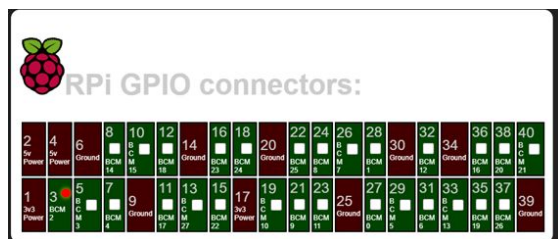
Estos pines son digitales, lo que significa que sólo pueden tener dos estados, apagado o

encendido. Pueden tener una dirección para recibir o enviar corriente (entrada, salida respectivamente) y todo esto es totalmente controlable por lenguajes de programación como Python, JavaScript, node-RED y otros. Los pines trabajan con una tensión de 3,3 V y un consumo máximo de corriente de 16 mA. Esto significa que podemos suministrar energía de forma segura desde un solo pin GPIO a través de una resistencia y uno o dos LEDs.

Para controlar GPIO con Python, lo primero es importar una librería de código escrito previamente. El más común y difundido es el RPi.GPIO, utilizado para crear miles de proyectos desde los primeros días de la Raspberry Pi.

En cuanto a la nomenclatura de los pines GPIO, existen dos maneras de identificarlos; el método más simple es definitivamente el que se refiere a su ubicación física (BOARD). Empezando por la parte superior izquierda del GPIO, tenemos el pin físico 1 que proporciona alimentación 3v3. A la derecha de ese pin se encuentra el pin físico de 2 que proporciona 5v de potencia. El número de pines seguirá creciendo a medida que se desciende por las columnas. A la izquierda se encontrarán todos los pines con números impares y a la derecha los pares.

Existe otro método utilizado, y es la numeración de pines de Broadcom (BCM), el cual se refiere a los alfileres GPIO que están conectados directamente al SoC del Raspberry Pi. Básicamente son conexiones directas al cerebro de nuestra Pi para conectar sensores y componentes para usar en nuestros diseños.



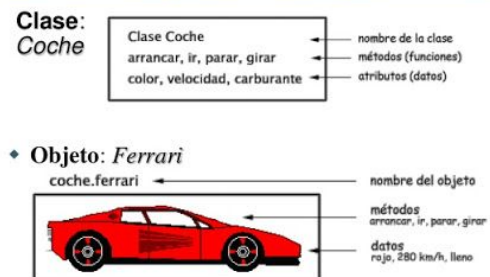
## C. PROGRAMACIÓN ORIENTADA A OBJETOS

La programación orientada a objetos se define como un paradigma de la programación, una manera de programar específica, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones (López, L.2006).

La programación orientada a objetos (POO) es una forma especial de programar, más cercana a cómo expresamos las cosas en la vida real que otros tipos de programación[4].

En este paradigma se tienen distintos elementos que se describen a continuación:

## EJEMPLO DE CLASES Y OBJETOS



### 4) Clases

Una clase es la descripción de un conjunto de objetos similares; consta de métodos y de datos que resumen las características comunes de dicho conjunto. En un lenguaje de programación orientada a objetos se pueden definir muchos objetos de la misma clase de la misma forma que, en la vida real, haríamos galletas (objeto) con el mismo molde (clase) solo que, para entenderlo mejor, cada galleta tendría igual forma pero es posible que tenga distinto sabor, textura, olor, color, etc.

Dicho de otro modo, una clase es la declaración de un tipo de objeto. Las clases son similares a los tipos abstractos de datos y equivalen a modelos que describen cómo se construyen ciertos tipos de objetos. Cada vez que se construye un objeto a partir de una clase estamos creando lo que se llama una instancia de esa clase.

### 5) Objeto

Se trata de un ente abstracto usado en programación que permite separar los diferentes componentes de un programa, simplificando así su elaboración, depuración y posteriores mejoras.(Mundo, H.1994)

Los objetos se integran, a diferencia de los métodos procedurales, tanto los procedimientos como las variables y datos referentes al objeto.

A los objetos se les otorga ciertas características en la vida real. Cada parte del programa que se desea realizar es tratado como objeto, siendo así estas partes independientes las unas de las otras. Los objetos se componen de 3 partes fundamentales: métodos, eventos y atributos.

## 6) Métodos

Son aquellas funciones que permiten efectuar el objeto y que nos rinden algún tipo de servicio durante el transcurso del programa. Determinan a su vez como va a responder el objeto cuando recibe un mensaje.

## 7) Eventos

Son aquellas acciones mediante las cuales el objeto reconoce que se está interactuando con él.

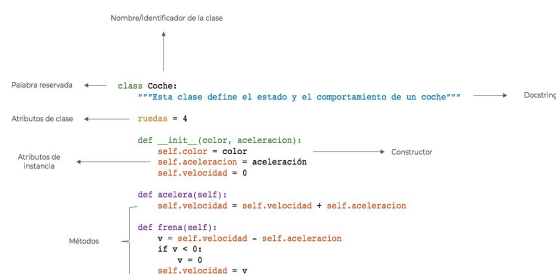
De esta forma el objeto se activa y responde al evento según lo programado en su código.

## 8) Atributos

Características que aplican al objeto solo en el caso en que el sea visible en pantalla por el usuario; entonces sus atributos son el aspecto que refleja, tanto en color, tamaño, posición, si está o no habilitado.

## D. PYTHON

Python es un lenguaje de programación de alto nivel, interpretado y multipropósito, que en los últimos años ha tenido un auge considerable ya que constituye uno de los lenguajes más empleados en el desarrollo de software (Fernández, A. 2013). Python puede ser utilizado en diversas plataformas y sistemas operativos, entre los que se puede destacar los más populares, como Windows, Mac OS X y Linux. [5] Este lenguaje nos permite trabajar en el paradigma de programación orientado a objetos, siguiendo la estructura que se muestra en la imagen:



## E. Librería Math

La librería **math** es proveída por Python como parte de su "librería estándar" y ofrece funciones matemáticas para uso en el campo de los números reales[6]. Algunas de las funciones ofrecidas son:

### 9) Funciones numéricas

- `math.ceil(x)`: Devuelve el entero más próximo mayor o igual que x.
- `math.floor(x)`: Devuelve el entero más próximo menor o igual que x.

- `math.gcd(a, b)`: Devuelve el máximo común divisor ("greatest common divisor") de los números a y b.
- `math.isnan(x)`: Devuelve el booleano True si x es un NaN ("Not a Number").

### 10) Funciones de potencia y logarítmicas

- `math.exp(x)`: Devuelve  $e^x$ .
- `math.log(x, [base])`: Devuelve el logaritmo neperiano de x. Si se incluye el segundo argumento, devuelve el logaritmo de x en la base indicada.
- `math.log2(x)`: Devuelve el logaritmo en base 2 de x.
- `math.log10(x)`: Devuelve el logaritmo en base 10 de x.
- `math.pow(x, y)`: Devuelve  $x^y$ .
- `math.sqrt(x)`: Devuelve la raíz cuadrada de x.

### 11) Funciones trigonométricas y de conversión de ángulos

- `math.cos(x)`: Devuelve el coseno de x.
- `math.sin(x)`: Devuelve el seno de x.
- `math.tan(x)`: Devuelve la tangente de x.
- `math.degrees(x)`: Convierte un ángulo de grados sexagesimales a radianes.
- `math.radians(x)`: Convierte un ángulo de radianes a grados sexagesimales.

### 12) Constantes

- `math.pi`: Número pi.
- `math.e`: Número e.
- `math.nan`: Valor equivalente a "no es un número".

Para poder acceder y hacer uso de esta librería, al iniciar el código debemos ingresar la siguiente línea:

**import math**

```
Python 3.6.0 (v3.6.0:41df79263a11, 1
D64)] on win32
Type "copyright", "credits" or "lic
>>> import math
>>> math.pow(2, 5)
32.0
>>> math.sqrt(2)
1.4142135623730951
>>> math.hypot(3, 4)
5.0
>>> math.cos(2*math.pi)
1.0
>>> |
```

## III. SOFTWARE

### 13) Plataforma:

La plataforma Created with code.uk, es una herramienta gratuita donde se permite escribir,



ejecutar, depurar y compartir programas de Python en el navegador web. Su ventaja es que no se necesita descargar ni instalar ningún archivo, es así que los programas de Python no pueden acceder a los archivos o dañar la computadora, por lo que es una forma segura de aprender a crear con código.

#### IV. METODOLOGÍA

La revisión bibliográfica fue el principal método empleado en la realización de este artículo, ya que nos centramos en el análisis de material bibliográfico y de consulta de información respecto al tema estudiado, lo que nos permitió afianzar los conceptos necesarios para el desarrollo de los ejemplos. La investigación fue posible por medio de herramientas utilizadas, así como el emulador empleado para la ejecución de la calculadora científica, la herramienta Github y del mismo modo la información recolectada nos permitió encontrar las múltiples estructuras a estudiar.

#### V. RESULTADOS

A partir del análisis y revisión de información se logró desarrollar y ejecutar a través de la plataforma `with.code.uk` y `node-red`, una calculadora científica con entradas desde los puertos virtuales de una Raspberry pi y una Interfaz humano-máquina; en dichos programas se visualizan diversas estructuras correspondientes al lenguaje de programación Python y varios nodos en el caso del Dashboard. El emulador `with.code.uk` descrito representan una gran ventaja ya que permite importar directamente para trabajar con una Raspberry.

#### VI. CONCLUSIONES

- Node-Red brinda una interfaz gráfica que se observa en pantalla como un dashboard, en el cual se podrá visualizar varias pestañas mediante una llamada a los nodos en diferentes grupos que a su vez se encuentra en subdivisiones creadas, al darnos esta posibilidad el usuario podrá interactuar con varias pestañas en un mismo tablero.
- La implementación de una interfaz gráfica basada en flujos brinda un ambiente más amigable para los desarrolladores en el aprendizaje de las nuevas tecnologías. Se debe promover esta metodología que permite al usuario observar de forma más detallada la interacción que existe entre software y hardware.

#### VII. RECOMENDACIONES

-Es recomendable conocer las librerías existentes en el programa a usar a fin de evitar que se realicen funciones innecesarias, mismas que ya pueden existir dentro del lenguaje de programación; en este caso Python cuenta con la librería `math` que nos ofrece una variedad de funciones que pueden emplearse en la solución a problemas de índole matemático.

- No olvidar las reglas sintácticas que tiene el lenguaje Python, ya que este se caracteriza por ser estricto en el caso de las indentaciones y las debidas puntuaciones. Conocer las palabras reservadas, los identificadores (los cuales determinan una variable, función, clase, módulo u objeto). Python también diferencia entre mayúsculas y minúsculas y no admite caracteres de puntuación como `@`, `$` o `%`.

- Se recomienda inicializar la entrada de pin en alto debido a que de otra forma no será seleccionado.

- En el caso del Dashboard de `node-Red` se recomienda conocer los nodos que lo contienen y cada uno de los parámetros de configuración de estos a fin de una mejor manipulación y resultados.

#### VIII. BIBLIOGRAFÍA

[1]OpenJS Foundation. (s.f.). *Node-red*. Obtenido de Node-red: <https://nodered.org>

[2]OpenJS Foundation. (Julio de 2020). *Node-RED*. Obtenido de <https://flows.nodered.org/node/node-red-dashboard>

[3]Sum, P. E. (2017, 26 diciembre). *¿Qué es GPIO? - Control de GPIO con Python en Raspberry Pi*. <https://www.programoergosum.com>. <https://www.programoergosum.com/cursos-online/raspberry-pi/238-control-de-gpio-con-python-en-raspberry-pi/que-es-gpio>

[4]*Qué es la programación orientada a objetos*. (2019, 11 diciembre). Desarrollo Web. <https://desarrolloweb.com/articulos/499.php>

[5]Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, 20(2), 1-13.

[6]*LA LIBRERÍA MATH*. (s. f.). Interactive Chaos. <https://www.interactivechaos.com/manual/tutorial-de-python/la-libreria-math>