

## PRODUCTO DE UNIDAD N.2

Jhonatan Tituaña, Bryan Azuero, Javier Arteaga

<

### Resumen -

*En el presente artículo se muestran los resultados de la práctica de la asignatura Arquitectura de computadoras, en la cual se busca obtener el conocimiento acerca del uso de las interfaces gráficas HMI con Node-Red y su importancia. Diseñaremos una calculadora científica elaborada con el lenguaje de programación Python empleando los conceptos de programación orientada a objetos, cada función de la calculadora científica estará enlazada con un pin virtual de la Raspberry PI y será puesta a prueba con la ayuda del simulador with code.*

### Objetivo General

Implementar una interfaz HMI para un prototipo electrónico de en la plataforma de Node red

Desarrollar una aplicación de una calculadora científica utilizando el lenguaje de programación Python y empleando los conceptos de programación orientada a objetos, que seleccione las operaciones matemáticas por medio de la lectura de los puertos virtuales de una Raspberry PI.

### Índice de Términos –

Node-Red, IoT, HMI, FRED, Python, POO, Raspberry PI

## I. INTRODUCCION

En el presente artículo analizaremos la interfaz HMI son parte importante para todos los procesos automatización, ya que es la manera en que las personas interactúan con las máquinas, para realizar el control, monitoreo y supervisión de procesos. El software y hardware utilizado para la automatización, está dominado por tecnologías cerradas, por las que se deben pagar licencias de un valor económico elevado. Por tal motivo los procesos pequeños no pueden ser implementados, cada día se realizan modificaciones con el afán de mejorar su desempeño, desde una HMI, controlar dichas acciones, lo cual puede ser demostrado con un prototipo de simule este proceso. [1]

El lenguaje de programación Python tiene muchas ventajas para enseñar el primer curso de programación. Su sintaxis

sencilla y su modo interactivo hace que éste sea ideal para la enseñanza. Es importante que es la programación orientada a objetos, conocer el paradigma de programación que más se utiliza en la actualidad. Esta práctica está destinada a explicar el desarrollo de una calculadora científica ocupando las funciones que nos proporcionar la librería especializadas en opciones matemáticas Math. [2]. Para ejecutar nuestro programa necesitaremos la ayuda de un si simulador de la Raspberry PI, ya que en ella se establecerá una función que permita la selección de las operaciones matemáticas por medio de la lectura de los puertos virtuales.

### A. Justificación.

Implementar los conocimientos adquiridos en la asignatura Arquitectura de computadoras.

### B. Objetivos específicos

- Desarrollar una interfaz HMI para la visualización y explicar su estructura interna dentro del Node red.
- Comprender todos los conceptos más básicos de programación orientada a objetos y el uso de las librerías Math.
- Comprender, seleccionar y aplicar los conceptos fundamentales aprendidos en el lenguaje de programación en Python.

## II. RASPBERRY PI

### A. Raspberry Pi

La Raspberry Pi es una computadora en una sola tarjeta (Board Computer) creada por la Raspberry Pi Foundation para promover la enseñanza de la programación en escuelas y países en desarrollo. Python es un lenguaje de programación de alto nivel, interpretado, permite programar usando los paradigmas de programación orientada a objetos o funcional y se puede extender fácilmente por medio de módulos escritos en C o C++. La Raspberry Pi se puede usar en proyectos de electrónica y para tareas básicas que haría cualquier ordenador de sobremesa como navegar por internet, hojas de cálculo, procesador de textos, reproducir vídeo en alta definición e incluso jugar a ciertos juegos. Al utilizar Python en la Raspberry Pi tenemos la ventaja de poder utiliza los pines GPIO para conectar el mundo digital con el mundo físico mediante la electrónica y programación. [3]

## B. Puertos GPIO.

GPIO es un sistema de entrada/salida de propósito general que cuenta con interruptores que permiten activar o desactivar los 40 pines. Estos pines están incluidos en todos los modelos de Raspberry Pi. Los pines GPIO tienen funciones específicas (aunque algunos comparten funciones) y se pueden agrupar de la siguiente manera. Los GPIO representan la interfaz entre la Raspberry Pi y el mundo exterior. Y con ellos podrás hacer multitud de proyectos, desde encender un LED hasta otros mucho más sofisticados. Pero para eso debes saber sus características y como se programan. Lo primero variará en función de la revisión de placa que tengas o del modelo. [4]

## III. INTERFAZ HMI.

Interfaz Hombre – Máquina es la interfaz que interviene entre los operadores y el proceso. Es la herramienta principal con la cual los supervisores de la línea de producción coordinan y controlan los procesos industriales. Por lo general, estos sistemas consistían en paneles de control conformados por varios elementos indicadores, como pulsadores, interruptores, selectores, luces piloto, registradores, etc., que están interconectados con el proceso de producción. En la actualidad, dado que las máquinas y procesos en general incluyen dispositivos electrónicos, y herramientas informáticas dedicadas a la programación y configuración de interfaces de visualización es posible encontrar con sistemas de HMI más versátiles y eficaces con una conexión más económica y relativamente simple con un determinado proceso [1].

## IV. EJEMPLO HMI

Primero debemos tener en cuenta la creación de un grupo, para la interfaz principal, en este caso vamos a nombre Home y para que la interfaz elaborada se construya en una sola columna, dividimos este mismo grupo en tres enlazados con el principal. Home First: para la interfaz a la izquierda Home Second: para la interfaz en medio. Home Third: para la interfaz a la derecha. Cada uno de los subgrupos ubicados en una columna diferente de la interfaz final.

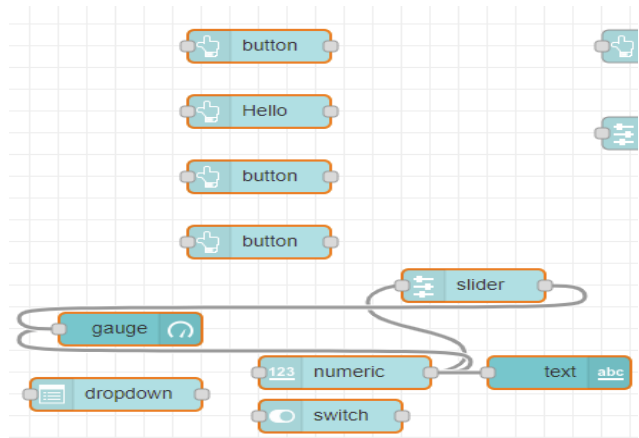
### A. Home First.

#### Nodos Utilizados

**Button:** Agrega un botón a la interfaz del usuario el ícono se puede configurar usando los íconos Material, también se puede configurar el color y el color de fondo.

**Slider:** Un control deslizante horizontal simple, con un tamaño de paso variable.

**Gauge:** Este nodo es un tipo de indicador en el tablero, tiene 4 modos: estándar (indicador simple), donut (360 ° completo), brújula y onda.



**Numeric:** Un widget de entrada numérico con botones arriba / abajo.

**Text:** Text es una salida de texto que sirve para mostrar un campo de texto no editable en la interfaz.

**Switch:** El nodo Switch permite enrutar los mensajes a diferentes ramas de un flujo mediante la evaluación de un conjunto de reglas para cada mensaje.

**Dropdown:** Un widget de selección desplegable, agrega un cuadro de varias opciones a la interfaz del usuario.

#### Explicación.

Los nodos Boton los utilizamos para mostrar en la interfaz de usuario cada uno de ellos, para poder identificarlos en la parte de la etiqueta le colocamos el nombre con el cual queremos que se muestre en la interfaz final.

El nodo slider es un control deslizable manipulable por el usuario, el rango lo podemos modificar a nuestras necesidades, para poder ver de una manera más visual el comportamiento del mismo, añadimos el nodo Gauge que es un indicador visual, los conectamos para que nos muestre la salida del nodo slider, el nodo numérico nos ayudara también a modificar el valor de gauge solo que en vez de utilizar el slider ocuparemos flechas para subir o bajar, dependiendo de este número el nodo Text de salida lo verá reflejado e imprimirá cada número seleccionado.

Para agregar un interruptor a la interfaz del usuario ocuparemos el nodo switch este nos devolverá dos estados, el nodo dropdown lo ocuparemos para desplegar un cuadro de opciones a la interfaz del usuario. El nodo dropdown no está conectado a otro nodo al igual que el nodo switch, ya que al pertenecer al nodo Home First se imprimirán en la columna correspondiente.

### B. Home Second

#### Nodos Utilizados

**Button:** Agrega un botón a la interfaz del usuario el ícono se puede configurar usando los íconos Material, también se puede configurar el color y el color de fondo.

**Debug:** El nodo de depuración se puede usar para mostrar mensajes en la barra lateral de depuración dentro del editor.

**Text:** Text es una salida de texto que sirve para mostrar un campo de texto no editable en la interfaz.

**Slider:** Un control deslizante horizontal simple, con un tamaño de paso variable.

**Dropdown:** Un widget de selección desplegable, agrega un cuadro de varias opciones a la interfaz del usuario.

**Chart:** Tiene modos de línea, barra y gráfico circular, podemos configurar el rango que necesitamos.

**Color picker:** Selector de color para un widget.

**Gauge:** Este nodo es un tipo de indicador en el tablero, tiene 4 modos: estándar (indicador simple), donut (360 ° completo), brújula y onda.

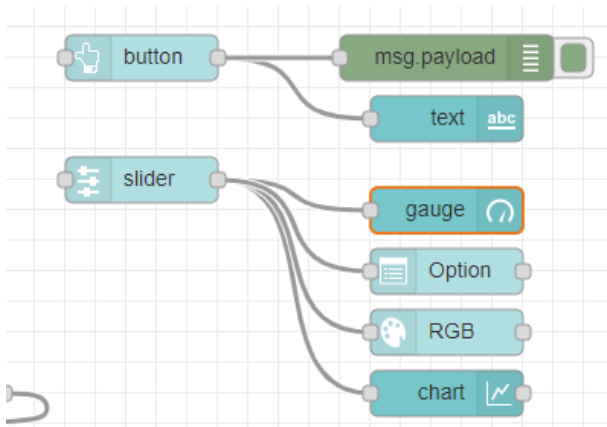
### Explicación

Para esta interfaz ocuparemos el grupo Home Second porque necesitamos que esta interfaz se coloque al lado derecho de la interfaz First

EL nodo button para que al momento de presionarlo salga un mensaje programado, para hacer esto debemos unir el nodo botón que es en donde configuraremos el mensaje con el nodo Text que es una salida de texto que sirve para mostrar un campo de texto en la interfaz.

Seguido de esto uniremos al nodo slider con los demás nodos, para que este pueda controlar las acciones de los otros. El nodo gauge es un indicador del tablero de modo grafico en el configuremos en rango que será de 0 a 200, del mismo modo el nodo chart nos devolverá una gráfica de tipo barra con el mismo rango ya definido. El nodo Dropdown nos desplegará un cuadro con tres opciones diferentes programables, y el nodo Color picker nos ayudara a seleccionar un color para nuestro widget. Todos estos nodos están enlazados al nodo principal slider que es un deslizador controlado por el usuario.

Por último, utilizaremos el nodo debug para controlar la ejecución del programa.



### C. Home Third

#### Nodos Utilizados.

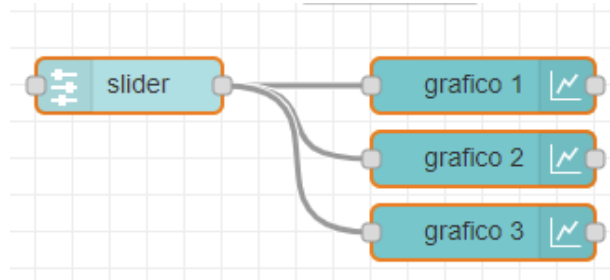
**Slider:** Un control deslizante horizontal simple, con un tamaño de paso variable.

**Chart:** Tiene modos de línea, barra y gráfico circular, podemos configurar el rango que necesitamos.

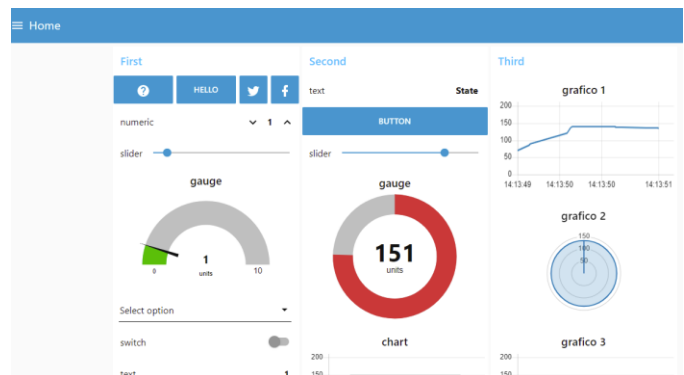
### Explicación

Para esta ultima interfaz ocuparemos el grupo Home third para que esta se ubique en una nueva columna así tendremos todas las interfaces en una distribución requerida.

Ocuparemos solo dos nodos el Slider y chart, el nodo slider tiene un control deslizante horizontal simple, esta nos servirá para controlar los nodos chart que están conectados a la slider principal. Los tres nodos chart hacen lo mismo, representan la información transmitida por la slider en forma visual, cada uno de ellos de diferente manera y en un rango igual al del slider que es de 0 min a 200 max.



### D. Resultados



### V. EJEMPLO RASBERRY

El siguiente ejemplo la realización de una calculadora científica la cual se ejecutará con puerto GPIO de la Raspberry Pi programada en Python orientado a objetos.

### A. Importación de librerías

```
mycode.py
1 #Importar modulos
2 import RPi.GPIO as GPIO
3 import math
4 GPIO.setmode(GPIO.BOARD)
```

La línea 2 será la importación de los puertos GPIO de la Raspberry.

La línea 3 línea será la importación de la librería math la cual nos proporcionará las funciones matemáticas que necesitamos.

La línea 4 será el diseño en el cual se mostrará los puertos GPIO de la Raspberry.

### B. Creación de clase

```
6 #clase
7 class Calculadora:
8 #atributos de clase
9     numero1=0
10    numero2=0
```

### C. Constructor

```
12 #Constructor
13 def __init__(self,numero1,numero2):
14     self.numero1=numero1
15     self.numero2=numero2
16 def sumar(self,numero1,numero2):
17     return (numero1+numero2)
18 def restar(self,numero1,numero2):
19     return(numero1-numero2)
20 def multiplicar(self,numero1,numero2):
21     return(numero1*numero2)
22 def dividir(self,numero1,numero2):
23     return(numero1/numero2)
24 def factorial(self,numero1):
25     return(math.factorial(numero1))
26 def raiz(self,numero1,numero2):
27     return(math.pow(numero1,(1/numero2)))
28 def pote(self,numero1,numero2):
29     return(math.pow(numero1,numero2))
```

Definiremos las funciones con las cuales va a contar nuestra calculadora.

### D. Menú

```
59 def menu(self):
60     print("MENU")
61     print("3)Suma      5)Resta      7)Multiplicación    8)División")
62     print("10)Sin      11)Asin      12)Sinh      22)Asinh")
63     print("23)Cos      24)Acos      26)Cosh      27)Acosh")
64     print("28)Tan      29)Atan      31)Tanh      32)Atanh")
65     print("13)Raiz      15)Potencia  16)Log      18)Factorial")
66     print("19)e      21)Pi")
67     print("\nSelecciones un pin:")
68 def leer(self,mensaje):
69     print(mensaje)
70     men1=int(input())
71     return men1
```

Mostraremos una guía para que el usuario puede seleccionar la opción que desea hacer.

### E. Funciones

```
73 def imprimir(self,num1,num2,resultado):
74     if GPIO.input(3) == GPIO.HIGH:
75         print(num1,"+",num2,"=",resultado)
76     elif GPIO.input(5) == GPIO.HIGH:
77         print(num1,"-",num2,"=",resultado)
78     elif GPIO.input(7) == GPIO.HIGH:
130 def operar(self,num1,num2):
131     resultado=0
132     if GPIO.input(3) == GPIO.HIGH:
133         resultado=self.sumar(num1,num2)
134     elif GPIO.input(5) == GPIO.HIGH:
135         resultado=self.restar(num1,num2)
```

Las funciones imprimir y operar serán para operaciones los cuales requieran el ingreso de dos números.

```
87 def imprimir2(self,num1,resultado):
88     if GPIO.input(18) == GPIO.HIGH:
89         print(num1,"! = ",resultado)
90     elif GPIO.input(10) == GPIO.HIGH:
91         print("El Sin de", num1,"=",resultado)
92     elif GPIO.input(11) == GPIO.HIGH:
93         print("El Asin de", num1,"=",resultado)
94     elif GPIO.input(12) == GPIO.HIGH:
95         print("El Sinh de", num1,"=",resultado)
96     elif GPIO.input(22) == GPIO.HIGH:
97         print("El Asinh de", num1,"=",resultado)
146 def operar2(self,num1):
147     if GPIO.input(10) == GPIO.HIGH:
148         resultado=self.sin(num1)
149     elif GPIO.input(11) == GPIO.HIGH:
150         resultado=self.asin(num1)
151     elif GPIO.input(12) == GPIO.HIGH:
152         resultado=self.sinh(num1)
```

Las funciones imprimir2 y operar2 serán para operaciones los cuales requieran el ingreso de un solo número.

```
117 def imprimir3(self,resultado):
118     if GPIO.input(19) == GPIO.HIGH:
119         print("El valor de e =", resultado )
120     if GPIO.input(21) == GPIO.HIGH:
121         print("El valor de pi =", resultado )
122
123 def operar3(self):
124     if GPIO.input(19) == GPIO.HIGH:
125         resultado=self.e()
126     if GPIO.input(21) == GPIO.HIGH:
127         resultado=self.pi()
128     return resultado
```

Las funciones imprimir3 y operar3 serán para operaciones los cuales no requieran de ingreso de números como es el caso al observar el resultado de "e" y de "pi".

### F. Función principal

```

175 #Funcion principal
176 calcul=Calculadora(0,0)
177 calcul.menu()
178 while(x != 0):
179     if (GPIO.input(3) == GPIO.HIGH or GPIO.i
GPIO.input(15) == GPIO.HIGH):
180         num1=calcul.leer("Ingresar Numero1")
181         num2=calcul.leer("Ingresar Numero2")
182         resultado=calcul.operar(num1,num2)
183         calcul.imprimir(num1,num2,resultado)
184         x=0
185     elif(GPIO.input(19) == GPIO.HIGH or GPIO
GPIO.input(24) == GPIO.HIGH or GPIO.input(
GPIO.input(31) == GPIO.HIGH or GPIO.input(
186         resultado=calcul.operar3()
187         calcul.imprimir3(resultado)
188         x=0
189     elif(GPIO.input(10) == GPIO.HIGH or GPIO
GPIO.input(24) == GPIO.HIGH or GPIO.input(
GPIO.input(31) == GPIO.HIGH or GPIO.input(
190         num1=calcul.leer("Ingresar Numero1")
191         resultado=calcul.operar2(num1)
192         calcul.imprimir2(num1,resultado)
193         x=0

```

Cuenta con la llamada a la clase, para poder usar todos los objetos hay presentes.

Una estructura de control while la cual permite que no muestre nada mientras no se seleccione un pin y varias estructuras if las cuales al detectar que un puerto ha sido seleccionado ejecutara o llamara a estrictos objetos que son fundamentales para su funcionamiento y una vez acabe de realizarlos termine.

### VI. CONCLUSIONES

- Node-Red te permite realizar varias interfaces HMI y una vez se domine esta forma de programación visual será podrá realizar grandes ejemplos.
- La combinación node red-raspbeery pi destaca en diversos proyectos y aplicaciones que permiten a los adentrarse usuarios y explorar temas de nueva tecnología como IoT, computación en la nube, robótica y general aspectos un poco mas avanzados de ciencias de la computación y electrónica.
- La página de Create With Code te permite practicar el lenguaje de programación Python sin la necesidad de descargar algún programa, además te permite simular otros microprocesadores y programarlos en lenguaje Python en la misma página.

### VII. RECOMENDACIONES

- Consultar a fondo cada nodo utilizado en Node-Red para poder sacarle el máximo provecho a dicho nodo.
- En el sitio web de la pagina Create With Code encontraras varios ejemplos que pueden ser de mucha ayuda.
- Asegurarse de tener como mínimo un conocimiento básico en programación antes de aventurarse a implementar un proyecto usando alguna de estas

plataformas o dispositivos porque de lo contrario se puede entorpecer el proceso de desarrollo del mismo.

### VIII. BIBLIOGRAFÍA

- [1] PDAControl, «FRED front-end Node-RED en la Nube,» 06 Junio 2017. [En línea]. Available: <http://pdacontroles.com/fred-front-end-node-red-en-la-nube/>.
- [2] R. L. Michael Blackstock, «Toward a Distributed Data Flow Platform for Edge Computing,» Columbia, octubre 2014. [En línea]. Available: [https://www.researchgate.net/publication/300757723\\_Toward\\_a\\_Distributed\\_Data\\_Flow\\_Platform\\_for\\_Edge\\_Computing](https://www.researchgate.net/publication/300757723_Toward_a_Distributed_Data_Flow_Platform_for_Edge_Computing). [Último acceso: 22 julio 2020].
- [3] Y. D.-R. A. B.-G. Ivet Challenger-Pérez, «El lenguaje de programación Python en la nube,» 01. [En línea]. Available: <https://www.redalyc.org/pdf/1815/181531232001.pdf>. [Último acceso: 25 julio 2020].
- [4] F. L. ., M. Gina Mejía, «Dashboard for Evaluating the Quality of Open Learning Courses,» 11 Mayo 2020. [En línea]. Available: <https://www.mdpi.com/journal/sustainability>.