

Interfaz HMI Node-red y Calculadora Científica en Raspberry pi (Julio de 2020)

Chanchay Randy

rfchanchay@espe.edu.ec

Zurita Dilan

dezurita4@espe.edu.ec

Lema Erick

ejlema2@espe.edu.ec

Resumen – El siguiente documento presenta aspectos generales de una interfaz HMI en Nod-Red y la programación orientada a objetos en Python mediante una calculadora científica implementada en raspberry.

Palabras Clave: Programación orientada a objetos, raspberry, node-red, dashboard, HMI

ABSTRACT: The following document presents the creation of both an HMI interface in Nod-Red and a scientific calculator in Raspberry Pi using flowchart and Poo programming respectively.

Keywords: Object oriented programming, raspberry, node-red, dashboard, HMI

1. Introduccion

Raspberry y node-red son herramientas de aprendizaje muy amplias que nos permiten desarrollar conocimiento en distintas áreas, una de ellas es la programación.

Para raspberry dependiendo del sistema operativo que le ingresemos, nos permite trabajar con Python un lenguaje de programación de sintaxis sencilla y que nos permite trabajar con POO.

Node-red por su parte nos permite crear programas mediante java script de una manera mas intuitiva ya que se basa en nodos de conexión.

2. Node-Red y HMI

2.1. ¿Qué es una interfaz HMI?

HMI son las siglas de human-machine interface y se refieren a un panel que permite a un usuario comunicarse con una máquina, software o sistema. Técnicamente, se puede referir a cualquier pantalla que se use para interactuar con un equipo, pero se utiliza normalmente para las de entornos industriales. Las HMI muestran datos en tiempo real y permiten al usuario controlar las máquinas con una interfaz gráfica de usuario.[1]

2.2. ¿Cómo diseñar una interfaz HMI en Node RED?

Para poder diseñar una interfaz hombre máquina que sea atractiva visualmente y que cumpla con su proceso de interacción correcto, podemos utilizar un dashboard en la herramienta de programación de Node-RED, la cual nos permitirá ir creando una interfaz por medio de Widgets manipulables a antojo del programador para que finalmente llegue a ser intuitivo para el usuario que vaya hacer uso de esta interfaz[1]

2.3. ¿Que es Node-RED dashboard?

Este módulo proporciona un conjunto de nodos en Node-RED para crear rápidamente un panel de datos en vivo.

Estos nodos requieren node.js versión 8. La última versión para admitir el nodo v6 fue 2.9.5.

Desde la versión 2.10.0, puede crear e instalar nodos de widget como otros nodos Node-RED.[1]

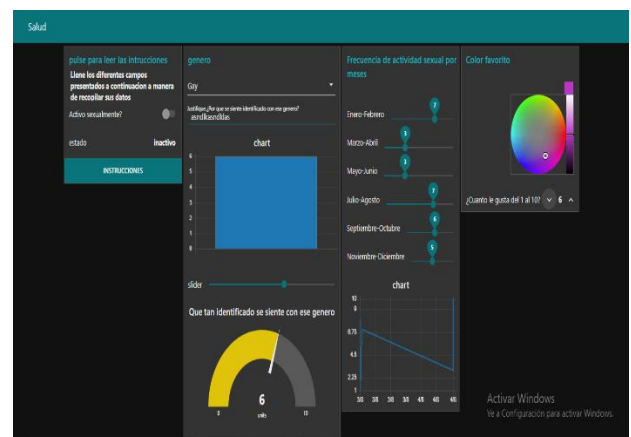


Ilustración 1 HMI Visual

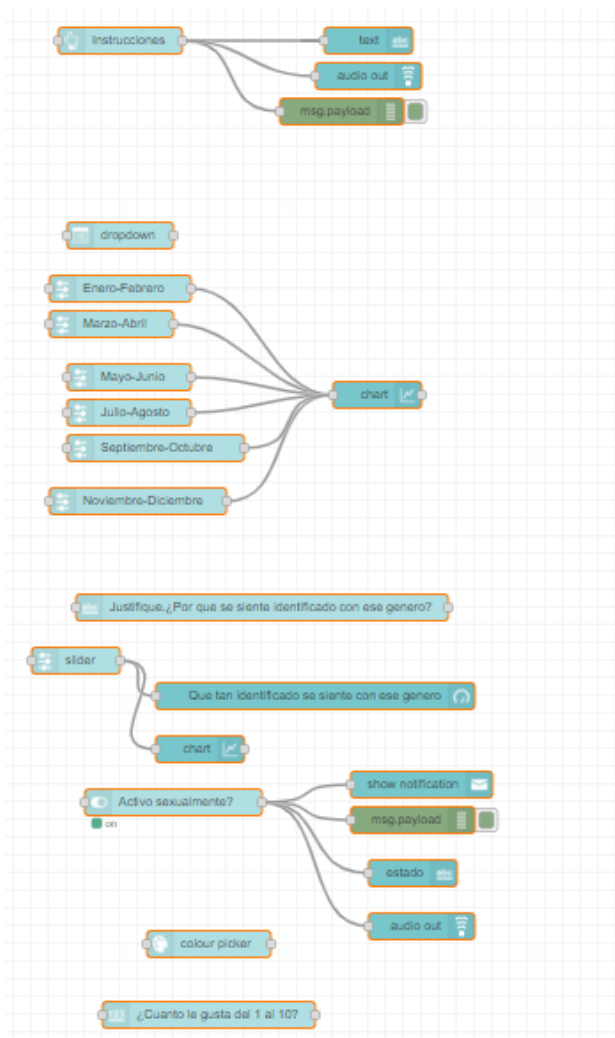


Ilustración 2 HMI en Node-Red

El programa cuenta con un dashboard dividido en grupos los cuales son: instrucciones, genero, frecuencia de actividad sexual, color favorito.

Al dividir el Dashboard en 4 podemos mostrar los datos en 4 secciones diferentes con distintos widgets internos con acciones distintas.

3. Raspberry pi y POO

3.1. Programación orientada a objetos

La programación orientada a objetos es un paradigma de programación en que los programas son vistos como formados por entidades llamadas objetos que recuerdan su propio estado y que se comunican entre sí mediante el paso de mensajes que se intercambian con la finalidad de: cambiar sus estados internos, compartir información, solicitar a otros objetos el procesamiento de dicha información [2]

3.2. Conceptos básicos de POO

- Clase

Definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

- Objeto

Instancia de una clase. Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), los mismos que consecuentemente reaccionan a eventos. Se corresponden con los objetos reales del mundo que nos rodea, o con objetos internos del sistema (del programa). Es una instancia a una clase.

- Método

Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un “mensaje”. Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un “evento” con un nuevo mensaje para otro objeto del sistema.

- Mensaje

Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó

- Comportamiento

Está definido por los métodos o mensajes a los que sabe responder dicho objeto, es decir, qué operaciones se pueden realizar con él.

- Evento

Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento la reacción que puede desencadenar un objeto; es decir, la acción que genera.

- Atributos

Características que tiene la clase

- Propiedad o atributo

Contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

- Estado interno

Es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase.

- Componentes de un objeto

Atributos, identidad, relaciones y métodos.

- Identificación de un objeto

Un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes.[2]

3.3. POO en python

En python la POO se expresa de manera simple y fácil de escribir pero debes tener en cuenta que para programar debes entender cómo funciona la teoría de POO y aplicarla al código[3].

La teoría de la POO nos dice que todos los objetos deben pertenecer a una clase, ya que esta es la base para diferenciarse unos de otros teniendo atributos y comportamientos que los distingan de otros objetos que pertenezcan a otras clases, para crear clases en python lo hacemos de la siguiente manera:

```
class Auto():
```

Para definir un atributo simplemente creamos una variable con total normalidad y un valor cualquiera por dar:

```
class Auto():
```

```
    ruedas=4
```

Para definir un método lo hacemos igual como lo hacemos con una función con la palabra por defecto **def** y el nombre de dicho método pero para diferenciar un método de una función lo hacemos escribiendo dentro de sus paréntesis el parámetro **self**:

```
def desplazamiento(self):
```

```
    pass
```

La palabra **self** hace referencia a los objetos que pertenezcan a la clase y la palabra **pass** que colocamos dentro del método le indica a el intérprete de python que todavía no le hemos definido ningún funcionamiento a ese método.

Cuando tenemos nuestra clase lista ya podemos empezar a crear objetos que pertenezcan a esa clase, para crear objetos lo hacemos de la siguiente manera:

```
miVehiculo=Auto()
```

Para mostrar atributos:

```
miObjeto.atributo
```

Para mostrar métodos:

```
miObjeto.metodo()
```

Constructor `__init__()`

El método `__init__()` es un método especial, el cual se ejecuta al momento de instanciar un objeto. El comportamiento de `__init__()` es muy similar a los “constructores” en otros lenguajes. Los argumentos que se utilizan en la definición de `__init__()` corresponden a los parámetros que se deben ingresar al instanciar un objeto.

```
def __init__(self, cedula, nombre, apellido, sexo):
```

```
    """Constructor de clase Persona"""
```

```
    self.cedula = cedula
```

```
    self.nombre = nombre
```

```
    self.apellido = apellido    self.sexo = sexo
```

3.4. Calculadora y POO mediante Raspberry

El siguiente diagrama muestra el esquema de conexión de una calculadora científica en Raspberry pi, esta esta calculadora contará con distintos los siguientes cálculos:

Operaciones Básicas

1. Funciones trigonométricas
2. Raíces, Potencias y Logaritmos

3. Valores Absolutos
4. Factorial
5. Operaciones Basicas
6. Funciones hiperbólicas

La calculadora solamente podrá efectuar una operación a la vez y para el caso de las operaciones básicas únicamente puede operar 2 números a la vez. La calculadora contará con la opción de seleccionar que tipo de operación quiera realizar el usuario según el interruptor activo que el menú del programa le indique al usuario.

El esquema del Gpio con la calculadora es el siguiente:

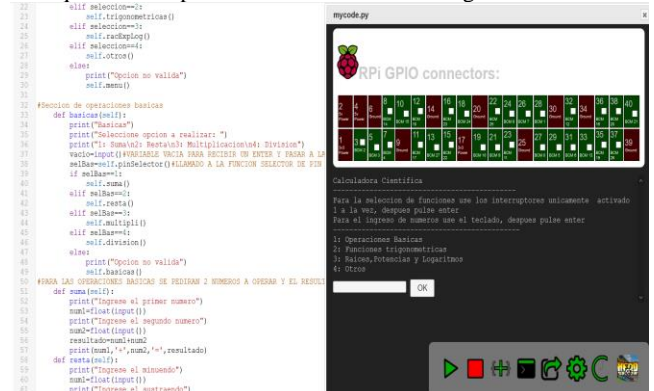


Ilustración 3 Calculadora Final

El circuito contará con 6 interruptores selectores conectados a los pines GPIO: 3,5,7,11,13,15. Cabe aclarar que para seleccionar la operación de la calculadora solo se debe seleccionar una operación e indicarla en el interruptor correspondiente a esa función, si el usuario escoge más de 2 operaciones el programa le indicará un error. Para el ingreso de números será necesario un teclado.

El programa contara con una clase llamada calculadora científica y con un objeto que la inicie llamando a sus funciones internas para realizar las operaciones.

4. Conclusiones

Una vez finalizado nuestro trabajo, hemos podido llevar a cabo programas perfectamente funcionales tanto para la plataforma de Node-RED como para Raspberry Pi, donde dichos proyectos fueron desarrollados en entornos extremadamente diferentes, para lo cual se tuvo que dedicar mayor dedicación en la investigación de ambos entornos de trabajo, dando como resultado una plataforma HMI funcional, y una calculadora científica bastante completa la cual se puede accionar mediante entradas de GPIO de nuestro Raspberry Pi

5. Recomendaciones

En cuanto a recomendaciones se trata vamos a dividir las segun los proyectos realizados.

Node-Red

Una de las grandes recomendaciones es siempre verificar que los nodos que se vayan añadiendo al programa, esten ubicados de manera correcta dentro del grupo donde se los quiera poner a funcionar, ya que si esto se pasa por alto, podemos tener complicaciones de descuadre en nuestra interfaz HMI

Calculadora

En cuanto a la calculadora se debe recomendar que se tenga una buena organización ya que el código es realmente extenso y si no se sigue un esquema o una estructura estricta podemos tener complicaciones a la hora de ponerla en funcionamiento

6. Bibliografía y fuentes.

- [1] “¿Qué significa HMI? Interfaz humano-máquina | COPA-DATA.” [Online]. Available: <https://www.copadata.com/es/productos/zenon-software-platform/visualizacion-control/que-significa-hmi-interfaz-humano-maquina-copa-data/>. [Accessed: 04-Aug-2020].
- [2] “9.3. Programación orientada a objetos — Materiales del entrenamiento de programación en Python - Nivel básico.” [Online]. Available: <https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion9/poo.html>. [Accessed: 31-Jul-2020].
- [3] “POO Programación Orientada a Objetos en Python - ▷ Cursos de Programación de 0 a Experto © Garantizados.” [Online]. Available: <https://unipython.com/programacion-orientada-objetos-python/>. [Accessed: 31-Jul-2020].