



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Node-RED & Calculadora Científica

Alumnos:

Kevin Topón

Andrés Paspuel

Henry Simba

Arquitectura de Computadoras

Docente:

Ing. Darwin Omar Alulema Flores



CONTENIDO

Título: Node-RED & Calculadora Científica	3
1.- Planteamiento del Problema.....	3
2.- Objetivos	3
Objetivo General	3
Objetivos Específicos	3
3.- Estado del Arte	4
4.- Marco Teórico.....	5
5.- Diagramación.....	10
6.- Lista de Componentes.....	10
7.- Mapa de variable	10
9.- Descripción de pre-requisitos y	13
10.- Aportaciones.....	13
11.- Conclusiones.....	14
12.- Recomendaciones.....	15
13.- Cronograma	16

Título: Node-RED & Calculadora Científica

1.- Planteamiento del Problema

Node-red

Cómo avanza el tiempo la tecnología crece de una manera inmensa y por ello van saliendo diferentes tipos de programación como por ejemplo en este caso NODE-RED maneja una programación mediante flujos por eso es necesario nosotros como futuros ingenieros en TICS ir teniendo los conocimientos adecuados para ser unos buenos profesionales .

Calculadora Científica

Construir una calculadora científica utilizando un lenguaje de programación Python mediante la librería MATH que desarrolle operaciones básicas y científicas.

2.- Objetivos

Objetivo General

Node-red

Analizar e interactuar la página de Node-Red como un lenguaje de herramientas basado en flujos para implementar una interfaz hombre-máquina.

Calculadora Científica

Implementar lo aprendido de Programación en Python desarrollando un ejemplo de una Calculadora donde desarrolle operaciones con una Librería Específica utilizando una Pagina de Programación llamada withcode.uk donde simularemos con los puertos de la Raspberry Pi.

Objetivos Específicos

Node-red

- Identificar los parámetros y su funcionalidad de Node -Red para tener conocimiento de otro nuevo lenguaje de programación mediante Flujos.
- Diseñar un programa o ejemplo en NODE-RED que implemente una interfaz HMI (Interfaz Humano Máquina)

Calculadora Científica

- Implementar los conocimientos adquiridos en las clases para realizar una calculadora científica desarrollada en <https://create.withcode.uk/> mediante entradas GPIO.
- Implementar la librería MATH para desarrollar la calculadora

3.- Estado del Arte

Node-red

- Anoja Rajalakshmi y Hamid Shahnasser en el año 2017 en California se realizó una investigación que era sobre el Internet de las cosas usando Node-Red y Alexa, cuyo objetivo era la mejora hasta qué punto IoT puede conectar dispositivos en diferentes plataformas. Los resultados obtenidos son que el proyecto proporciona la elasticidad de alterar el proceso de responder a un evento. Como ejemplo, permite al usuario cambiar una notificación de enviar un correo electrónico a un SMS o activar fuera de la luz para reducir el brillo, además de que con la ayuda de node red la tarea se facilita.
- Salas Cifuentes, Eduardo Arturo, Santana Hidalgo y Sergio Ricardo en el año 2019 en la ciudad de Guayaquil se realizó una investigación sobre Implementación de un sistema con interfaz HMI para la adquisición de datos en pruebas de transformadores que pretende que el sistema obtenga datos en la menor cantidad de tiempo y procesarlos de mejor manera y sus resultados fueron de que con la interfaz HMI fue de gran ayuda en recolección de datos en circuitos abiertos.
- Rachid Benhamadi, Mounir Bouhedda, Billel Bengherbia, Hamza Benyezza y Omar Benzineb en el año 2019 en Argelia se realizó un trabajo que consiste en el diseño y la realización de un sistema de monitoreo y control del centro de transmisión basado en Internet de las cosas (IoT) como resultado se obtiene que se valida con un prototipo realizado donde la supervisión y el control se realizan utilizando una Interfaz hombre-máquina (HMI) diseñada bajo Node-Red de IBM.

Calculadora Científica

- Sergio Rojas, Héctor Fernandez y Juan Ruiz en el año 2016 en la ciudad de Caracas se hizo un artículo sobre Aprendiendo a programar en python, entonces los que plantea es que hay adquirir las destrezas que sean suficientes para escribir programas de fácil mantenimiento y de eficiencia razonable.
- Gabriel Via Echezarreta en el año 2017 en la ciudad de Santander realizó un trabajo que es: Entorno de desarrollo para la realización de prácticas de sistemas empujados basado en Raspberry Pi, en este proyecto es la creación de una librería que sirva de interfaz para el desarrollo de aplicaciones que hagan uso del Explorer Phat, y como resultados se obtuvo Mejora del entorno de desarrollo, Librería phat.h: y Explorer Phat.
- Valiente Lerma, Arturo José en el año 2017 en la ciudad de Valencia se realizó un trabajo que consiste en una Alarma para el hogar basada en Raspberry Pi. cuyos resultado fueron conseguir integrar todos los dispositivos de entrada y salida que se han propuesto en la parte de hardware utilizado, además de conseguir implantar un sistema de seguridad para evitar que toda persona ajena pueda controlar nuestra alarma.

4.- Marco Teórico

Node-Red

Según (Lekić & Gardašević, 2018), Node-RED es una herramienta de desarrollo basada en flujo de código abierto para la integración de dispositivos de hardware IoT, API(interfaces de programación de aplicaciones) y servicios en línea desarrollados por IBM.

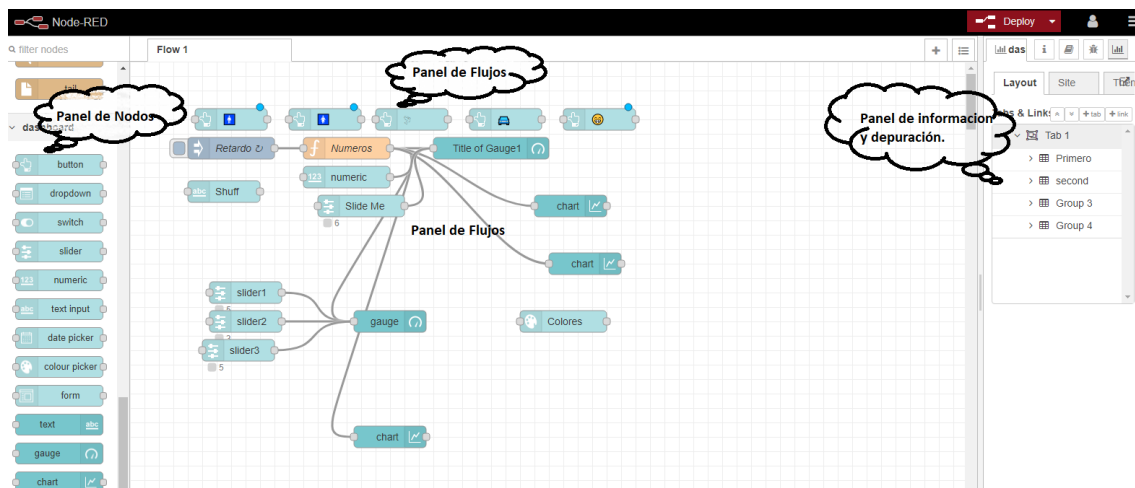
Node-RED es una herramienta gratuita basada en JavaScript, construida en la plataforma Node.js que proporciona un editor visual de flujo basado en un navegador. El sistema contiene nodos que están representados por los iconos apropiados. Y funciona de dos maneras: arrastrar, soltar y conectar nodos, o importar código JavaScript.

El nodo proporciona diferentes funciones, como monitorear el flujo como depurar nodo, o para leer y escribir con pines GPIO de Raspberry Pi, Los flujos creados se almacenan utilizando JSON (Notación de objetos JavaScript). Node-Red Permite a los

desarrolladores conectarse a la entrada y salida además de procesar nodos para crear flujos para el procesamiento de datos, controlar cosas o enviar alertas funciona con el principio de permitir la conexión de servicios web o personalizar nodos entre sí o con otras cosas, para realizar funciones como enviar datos de sensores por correo electrónico o servicios como el de realizar análisis complejos.

Node-RED contiene tres componentes básicos:

1. Panel de nodos.
2. Panel de flujos
3. Panel de información y depuración.



Node-RED es una herramienta poderosa y flexible que se utiliza para crear prototipos, siendo la esencia de esta herramienta permitir a los ingenieros y técnicos simplemente crear y configurar aplicaciones en tiempo real en dispositivos finales.

Además, (Coen-Porisini, 2018) propone que Node-RED es una herramienta de programación basada en flujos y basada en eventos que pueden comunicarse entre sí y regular el flujo de la información dentro del sistema diseñado. Es una representación visual basada en el navegador ayudando a los desarrolladores a comprender mejor las interacciones que ocurren dentro de toda la red IoT interactuando con entidades tanto de hardware como Sensores y de software como servicios, de igual manera destaca sobre la interfaz que nos brinda Node-RED además de permitir la integración de diferentes tecnologías como:

MongoDB: que es un motor de base de datos no relacional debido a que su mayor eficiencia en responder en muy poco tiempo

Java: es utilizado aquí debido a su amplia adopción en implementaciones reales.

Transporte de telemetría de la cola de mensajes (MQTT): se utiliza como un método ligero de publicación y suscripción para compartir información y notificaciones del sistema además de controlar el flujo de datos que proporciona la lógica de las aplicaciones IoT.

Python

Según (Rossum, 1995) Python es un lenguaje de programación simple pero potente e interpretado diferente a la programación en C y Shell ideal para la programación desechable y la creación rápida de prototipos. Su sintaxis se elabora a partir de construcciones tomadas de una variedad de otros idiomas, el intérprete de Python se amplía fácilmente con nuevas funciones y tipos de datos implementados en C. Python de igual manera es adecuado como un lenguaje de extensión para aplicaciones C altamente personalizables para varios sistemas operativos como por ejemplos UNIX (incluido Linux), el sistema operativo Apple Macintosh, MSDOS describiéndolo a esta sintaxis y la semántica central del lenguaje como consiso pero intenta ser exacto y completo.

De igual manera (Duque, 2009) argumenta que Python es un lenguaje de programación creado por Guido Van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible.

Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos.

La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo, los lenguajes interpretados son más flexibles y más portables. Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. En Python, como en Java y muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio llamado bytecode la primera

vez que se ejecuta, generando archivos .pyc o .pyo (bytecode optimizado), que son los que se ejecutarán en sucesivas ocasiones.

Tipado dinámico

La característica de tipado dinámico se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo del valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo.

Orientado a objetos

La orientación a objetos es un paradigma de programación en el que los conceptos del mundo real relevantes para nuestro problema se trasladan a clases y objetos en nuestro programa. La ejecución del programa consiste en una serie de interacciones entre los objetos. Python también permite la programación imperativa, programación funcional y programación orientada a aspectos.

Python es un lenguaje que todo el mundo debería conocer. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo

Raspberry Pi

Según (Aguilar, 2014) la Raspberry Pi es una computadora en una sola tarjeta (Single-Board Computer) creada por la Raspberry Pi Foundation para promover la enseñanza de la programación en escuelas y países en desarrollo.

Biblioteca para acceso a GPIO

```
>>> import RPi.GPIO as GPIO #importa la librería de GPIO
```

```
#usar número de terminal no de GPIO
```

```
>>>GPIO.setmode(GPIO.BOARD)
```

```
#configura como salida en bajo a la terminal 11
```

```
nf>>>GPIO.setup(11,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)
```



```
>>>GPIO.output(11,GPIO.HIGH) #pone la salida en alto
```

Estructuras de control básicas

Las estructuras de control más comunes son if, if - else y while, que son controladas por una condición lógica, separada del bloque controlado por : El inicio y el fin del bloque controlado depende únicamente de la indentación

Tips básicos de Python

- Importante: Python es sensible a mayúsculas y minúsculas
- Numerico: int (1 45 -678), float (12.234 -43.56), complex (-1.23+34.9j 56.1-156j)
- Cadenas de caracteres: “ Hola”, ‘Mundo’
- // división de punto flotante: 3/2=1.5
- // división entera: 3//2=1
- type(variable) regresa el tipo de variable

Funciones

Una función de un bloque de código organizado, probado y reutilizable. Permiten que los programas sean modulares.

Python incluye muchas funciones internas, tal como printf, pero también se puede construir funciones propias personalizadas, llamadas funciones definidas por el usuario

La definición de una función comienza por la palabra clave def seguida por el nombre de la función y paréntesis Los parámetros de entrada se colocan entre los paréntesis-

5.- Diagramación

Diagrama de la calculadora Científica

Calculadora
-numero1: int -numero2: int
+Calculadora(int numero1, int numero2) +sumar(int numero1, int numero2) +restar(int numero1, int numero2) +multiplicar(int numero1, int numero2) +dividir(int numero1, int numero2) +logaritmo_n(int numero1, int numero2) +potencia(int numero1, int numero2) +seno(int numero1) +coseno(int numero1) +tangente(int numero1) +rad_grad(int numero1) +grad_rad(int numero1) +logaritmo10(int numero1) +exponencial(int numero1) +raiz(int numero1) +menu() +leer(String mensaje) +imprimir(float num1, float num2, float resultado, int operacion) +operar(int opcion, float num1, float num2) +reinicio_apagado()

6.- Lista de Componentes

Node-red	Calculadora Científica
<ul style="list-style-type: none"> • Node-Red (instalado en la laptop) • Laptop • Conocimiento de Programación • Conexión a Internet 	<ul style="list-style-type: none"> • Conocimiento en lenguaje python • https://create.withcode.uk/

7.- Mapa de variable

Node-red

- Números Randómicos: el término random es para generar número aleatorios.
- parseInt(): comprueba el primer argumento, una cadena, e intenta devolver un entero de la base especificada.
- return: es el valor que nos va a devolver.
- msg.topic y msg.payload: propiedad que contiene el cuerpo del mensaje.

Calculadora Científica

- import: se utiliza para importar un módulo en este caso fue Math y import RPi.GPIO as GPIO.
- self :sirve para acceder a un atributo dentro del objeto (clase) en sí.
- GPIO: son los puertos que emulan a la de una raspberry.
- setup: se define uno de los pines como entrada o salida.
- class: tipo de dato definido por el usuario, y la crear instancias de una clase hace relación a la creación de objetos.
- def: función usada para crear objetos funciones definidas por el usuario
- __init__: su función es establecer un estado inicial en el objeto nada más instanciarlo, es decir, inicializar los atributos.
- input:es para introducir datos de distintos tipos desde la entrada estándar.
- return: es el valor que nos va a devolver.
- if/elif/else: se utiliza para condiciones que sean más de una.
- print: es lo que se va a presentar en pantalla.
- while: repetir la condición hasta que la misma se cumpla.

8.- Descripción código fuente

Node-red

Nuestro diseño que se diseñó con una interfaz HMI_

1. Vamos a tener que crear nuestra”pestaña” a la que va a estar asociada cualquier punto al que vayamos a unir y estas se unirán mediante líneas.
2. En este trabajo se agrega un código con números randómicos que este se ejecutará en nuestra interfaz y así se podrá observar valores aleatorios y este a su vez interactúa con los demás diagramas que se implementó.

Calculadora Científica

En el caso de la calculadora hay un sin número de páginas y programas en el cual realizar una calculadora científica pero que también emule los puertos GPIO de una raspberry, pero ya entrando en detalle vamos a ver que es lo que nos ofrece la calculadora científica en nuestro programa entonces esta inicia desde una simple suma, resta, multiplicación, etc hasta encontrar el coseno de un número pero para que llegue a interpretar lo que

deseamos se tuvo que utilizar los pines GPIO y para ello tuvimos que configurar cada uno pero sin olvidar que emulaba a una raspberry así que debíamos respetar cada una de sus partes que la componen.

1. GPIO.setup(3, GPIO.IN): designamos los pines de entrada de la raspberry.
2. class Calculadora: numero1=0, numero2=0 aquí llamamos a nuestra clase y a los atributos que esta tendrá.
3. def __init__(self,numero1,numero2): aquí definimos nuestro constructor.
4. def sumar(self,numero1,numero2):

```
    return (numero1+numero2)
```

Se define cada función que va a tener nuestra calculadora por ejemplo como vemos la suma de 2 números.

5. while opcion==0:
 if GPIO.input(3): #sumar
 opcion=1

Aquí va a repetirse la veces que sean necesarias hasta que “opcion” sea igual a un número asignado.

6. def imprimir(self,num1,num2,resultado,operacion):
 if operacion == 1:
 print(num1,"+",num2,"=",resultado)

En este paso se define como se va a presentar en pantalla el mensaje e la operación

7. def operar(self,opcion,num1,num2):
 resultado=0
 if opcion==1: #sumar
 resultado=self.sumar(num1,num2)

Se tiene que crear una variable donde esta va a estar almacenada pero claro que vamos a llamar a la función sumar en este caso para que ejecute esa operación.

8. def reinicio_apagado(self):
 print("\nQue desea realizar? (presione un pin en la raspberry): ")
 print("38) Realizar otra operacion")
 print("40) Apagar")

Después de realizar x operación nos va a preguntar si se desea hacer otra operación o aquí es donde termina.

9.- Descripción de pre-requisitos y

Node-red

Instalación de node-red en laptop o computadora de escritorio y seguir los pasos siguientes:

1. El paso principal es el de instalar node red y para ello nos dirigimos a la página <https://nodejs.org/> y buscamos cual es para nosotros.
2. Abrir nuestro node-red y digitar “node-red” y darle a enter.
3. Nos desplegará una dirección, la copiamos y la pegamos en nuestro navegador.
4. Se nos abrirá una ventana para comenzar a crear nuestra interfaz HMI.

Calculadora Científica

Para utilizar una las raspberry necesitamos las librerías siguientes antes de programar nuestra calculadora:

- import Math: sirve para dar acceso a funciones hiperbólicas, trigonométricas y logarítmicas para números reales.
- import RPi.GPIO as GPIO: tenemos que importar la librería que nos permite manejar los GPIO.

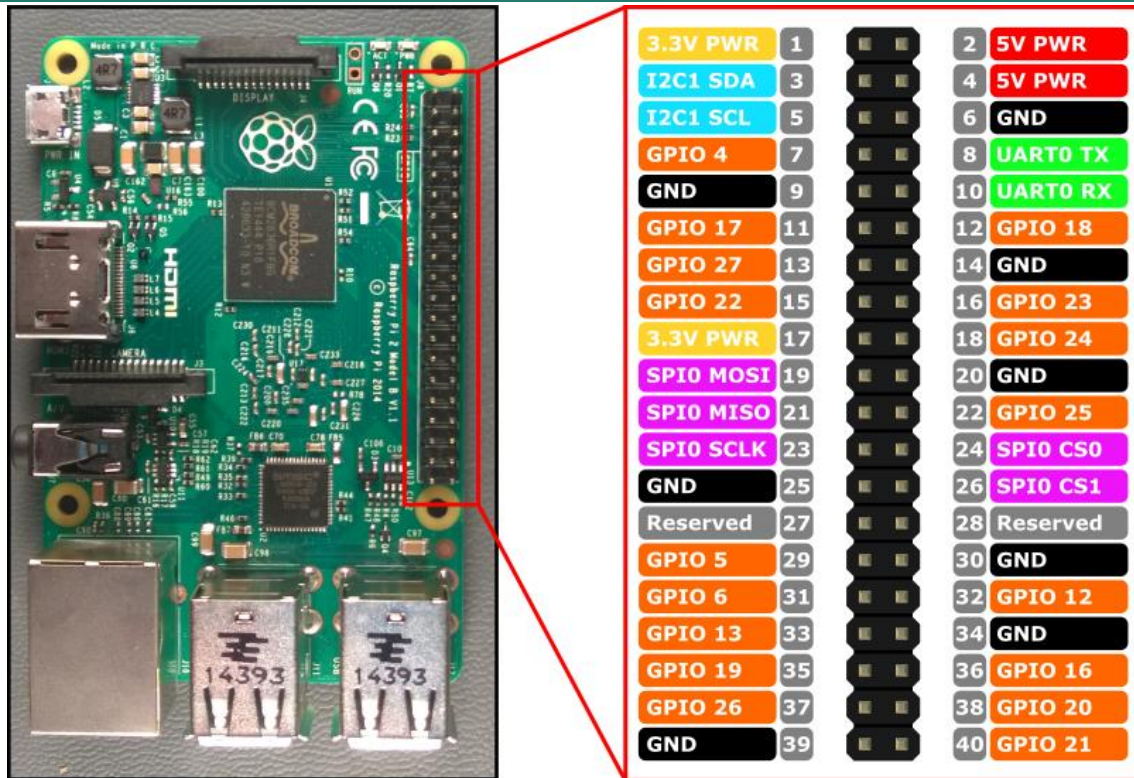
10.- Aportaciones

Node-red

Se utilizó una función que es de datos randomicos que prácticamente esto generaría datos aleatorios y por ende nuestra interfaz trabajaría de manera automática y esto serviría para las personas que quieran un interfaz similar y allí podríamos indicarles como funciona la interfaz, pero ya ingresando datos reales en la interfaz.

Calculadora Científica

En este caso para poder utilizar las librerías que se tienen digitar esta palabra “import” que significa incluir esta librería y lo que le sigue en este caso fue de Math para las funciones como “SENO” y RPi.GPIO as GPIO para poder utilizar los pines de una raspberry que son los siguientes como se muestra en la imagen:



11.- Conclusiones

Node-red

- El manejo del nodo red y de como crear nuestra interfaz es muy comprensible debido a que todos los elementos y componentes están prácticamente a la vista por lo que es amigable.
- Node red al momento de utilizar es fácil de comunicar cada elemento de nuestra interfaz que estamos creando y se volvería más fácil comprender el diagrama que se está elaborando.
- Node-RED ha demostrado ser ampliamente útil para representar la interfaz fácil de usar así como todos sus componentes Dashboard.

Calculadora Científica

- El implementar algo nuevo como son las librerías Math y de la raspberry debido a que se llegan a utilizar en temas puntuales como por ejemplo designar una GPIO si es suma, resta o cualquier otra operación y la de Math en las funciones como seno, coseno para utilizar.

- Los pines GPIO son prácticamente lo mismo que la de una raspberry y es importante saber porque no todos los pines son los que se tienen acceso y se pueden programarlos ya que esos van a estar de un color rojo.
- en este trabajo ya se unieron todos los temas que se vieron respecto a programación de python y que se logro el objetivo de diseñar nuestra calculadora científica.

12.- Recomendaciones

Node-red

- Tener una introducción sobre lo que es Node-red y de los parámetros que estos los componen debido a que por más amigable que sea Node-red si no conocemos la funcionalidad de cada elemento es imposible utilizarlo.
- Seguir los pasos de instalación de Node-red debido a que si no hacemos no podremos utilizar en la página de node-red porque ahí es donde se nos genera una dirección para poder conectarnos.

Calculadora Científica

- Llevar un orden adecuado porque puede ser fácil confundirse al momento de querer utilizar alguna función.
- Tener una programación básica en objetos porque ya llegamos a utilizar métodos, clases, constructores.
- Conocer como esta estructurada los GPIO de la raspberry porque eso es esencial momento de designar que va hacer cada una de ellas

13.- Cronograma

Columna1	Columna2	Columna3	Columna4	Columna5	Columna6	Columna7	Columna8
PRODUCTO UNIDAD	25/07/2020	26/07/2020	27/07/2020	28/07/2020	29/07/2020	30/07/2020	31/07/2020
PLANTEAMIENTO DEL PROBLEMA							
OBJETIVOS							
ESTADO DEL ARTE							
MARCO TEÓRICO							
DIAGRAMAS							
LISTA DE COMPONENTES							
MAPA DE VARIABLES							
EXPLICACIÓN DEL CÓDIGO FUENTE							
DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN							
APORTACIONES							
CONCLUSIONES							
RECOMENDACIONES							
CRONOGRAMA							
BIBLIOGRAFÍA							
ANEXOS							
MANUAL DE USUARIO							
HOJAS TÉCNICAS							

14.- Bibliografía

- Aguilar, M. F. (2014). Programación en Python en la. Obtenido de <http://sagitario.itmorelia.edu.mx/mfraga/materias/soemb/python.pdf>
- Coen-Porisini, A. R. (2018). Smart transport and logistics: A Node-RED implementation. *Check for updates Wiley*.
- Duque, R. G. (2009). Python para todos. España. Obtenido de http://www.utic.edu.py/citil/images/Manuales/Python_para_todos.pdf
- Lekić, M., & Gardašević, G. (2018). IoT sensor integration to Node-RED platform. *XVII Simposio Internacional INFOTEH*, 5.
- OLIPHANT, T. E. (15 de 05 de 2007). Python for Scientific Computing. CiSE Computational Physics.
- Rossum, G. (1995). Manual de referencia de Python. *CWI (Centro de Matemáticas e Informática)*.
- Chazallet, S. (2016). *Python 3: los fundamentos del lenguaje*. Ediciones ENI.