



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Máster en Ingeniería de Computadores y Redes

Trabajo Fin de Máster

Programación "flow-based" para IoT

Autor: *GUO YILIANG*

Director(es): *Manzoni, Pietro*

<7/2021>

Resumen

La programación basada en flujo (PBF) es un paradigma de programación que define las aplicaciones como redes de procesos de "caja negra", los cuales intercambian datos a través de conexiones predefinidas por pasos de mensajes, en donde las conexiones están especificadas externamente a los procesos. Estos procesos de caja negra pueden ser conectados un número ilimitado de veces para formar diferentes aplicaciones sin tener que ser modificados internamente.

En el IoT se está empezando a utilizar este paradigma con plataformas como NodeRED, Flowhub, NoFlo, MsgFlo, imgflo y MicroFlo. En particular NodeRED permite integrar dispositivos con electrónica embebidos como RaspberryPi, Odroid, jetson nano etc.

En este proyecto se quiere evaluar las efectivas posibilidades de este paradigma diseñando y evaluando un prototipo de una aplicación a definir por el estudiante que permita ofrecer unas líneas guías para otros desarrolladores sobre el uso de estas herramientas.

Palabras clave: raspberry , node-red, IOT, sensores, . PBF

Abstract

Flow-based programming (PBF) is a programming paradigm that defines applications as "black box" process networks, which exchange data through predefined connections by message steps, where the connections are specified externally to the processes. . These black box processes can be connected an unlimited number of times to form different applications without having to be modified internally.

In the IoT, this paradigm is beginning to be used with platforms such as NodeRED, Flowhub, NoFlo, MsgFlo, imgflo and MicroFlo. In particular, NodeRED allows the integration of embedded devices such as RaspberryPi, Odroid, jetson nano etc.

In this project we want to evaluate the effective possibilities of this paradigm by designing and evaluating a prototype of an application to be defined by the student that allows offering guidelines for other developers on the use of these tools.

Keywords: raspberry, node-red, IOT, sensores, . PBF

Resum

La programació basada en flux (*PBF) és un paradigma de programació que defineix les aplicacions com a xarxes de processos de "caixa negra", els quals intercanvien dades a través de connexions predefinides per passos de missatges, on les connexions estan especificades externament als processos. Aquests processos de caixa negra poden ser connectats un nombre il·limitat de vegades per a formar diferents aplicacions sense haver de ser modificats internament.

En el IoT s'està començant a utilitzar aquest paradigma amb plataformes com NodeRED, Flowhub, NoFlo, MsgFlo, imgflo i MicroFlo. En particular NodeRED permet integrar dispositius amb electrònica embeguts com RaspberryPi, Odroid, jetson nano etc.

En aquest projecte es vol avaluar les efectives possibilitats d'aquest paradigma dissenyant i avaluant un prototip d'una aplicació a definir per l'estudiant que permeti oferir unes línies guies per a altres desenvolupadors sobre l'ús d'aquestes eines.

Paraules clau: raspberry, node-xarxa, IOT, sensors,. PBF

Contenido

Resumen	4
Abstract	5
Resum	6
1. Introducción	9
1.1 Información abierta	10
1.2. CKAN	11
1.3. Node-red.....	12
1.3.1. Abra y use Node-red.....	13
1.3.2. Funciones de varios nodos.....	15
1.3.3. Visualización de datos.....	16
1.4. Raspberry pi	20
1.5. Objetivos.....	21
2. Diseño de la solución	24
2.1. Arquitectura del Sistema.....	24
2.2. Diseño Detallado.....	24
3. Implementación.....	27
3.1 Operación y depuración de Node-Red	27
3.2. Sistema operativo del Raspberry Pi.....	33
4. Pruebas.....	39
5. Conclusiones	47
6. Referencias.....	48

1. Introducción

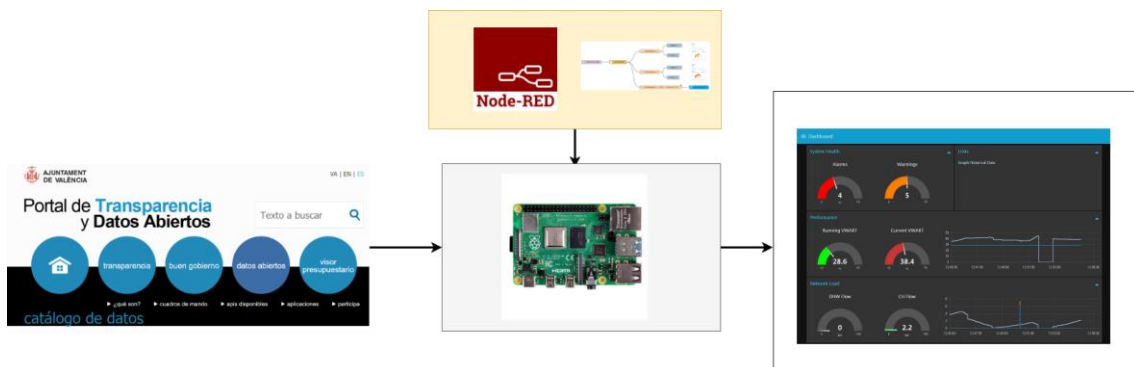


Figura 1.1. Arquitectura del sistema.

En el mundo del IoT, el desarrollo de la tecnología se ha visto impulsado debido al gran crecimiento de la cantidad de datos generados por los dispositivos conectados a la red, desde sensores para controlar el tráfico, hasta el monitoreo de signos vitales en tiempo real.

El concepto de Internet de las cosas apareció por primera vez en el libro de Bill Gates de 1995 "El camino hacia el futuro" [1], y más tarde en 1998, el Instituto de Tecnología de Massachusetts, en los Estados Unidos propuso la idea del "Internet de las cosas" llamado sistema EPC en ese momento. Por último, el 17 de noviembre de 2005, en la Cumbre Mundial sobre la Sociedad de la Información (CMSI) en Túnez, la Unión Internacional de Telecomunicaciones (UIT) publicó el "Informe de la UIT sobre Internet 2005: Internet de las cosas", que presentó oficialmente el concepto de "Internet de las Cosas". El Internet de las cosas (IoT) es una red extendida y expandida basada en Internet, en otras palabras, es una red enorme formada por la combinación de varios dispositivos de detección de información con la red. Se puede realizar en cualquier momento, en cualquier lugar, humano, máquina e interconexión de cosas.

Según la situación actual, la tasa de utilización de Internet de las cosas está creciendo rápidamente y, en el campo empresarial, el Internet de las cosas parece ser bastante popular. Los responsables de la toma de decisiones comerciales, los responsables de la toma de decisiones de TI y los desarrolladores de empresas están integrando rápidamente el Internet de las cosas en sus negocios, y la mayoría de las empresas están satisfechas con los resultados comerciales. Por lo tanto, las empresas están aún más entusiasmadas con el Internet de las cosas.

En este documento, intercambiaremos información de red a través de una conexión predefinida. Se utiliza la plataforma node-red, la plataforma de datos abiertos CKAN y la

plataforma hardware Raspberry Pi. En este proceso, la plataforma de datos abiertos CKAN actúa como proveedor de datos. Los datos de la plataforma CKAN se pueden acceder y utilizar de forma gratuita. La plataforma node-red es una herramienta de programación, las operaciones que se realicen en este proyecto se realizarán en la plataforma node-red, y en la página de visualización generada por los datos de texto se puede observar en cualquier momento para que sea más rápido cuando se produce un error. El proceso de creación de un "flujo" es más fácil y la implementación del servicio es más rápida. Como dispositivo integrado de node-red, la plataforma hardware Raspberry Pi está conectado con node-red de manera ilimitada. El raspberry pi tiene un sistema completo, tamaño pequeño, bajo costo y fácil de usar.

Por supuesto, también podemos utilizar algún otro software en este proyecto, como: "Rango de IP" para encontrar la dirección IP de la red inalámbrica. Este software puede buscar la dirección IP del dispositivo Raspberry Pi.

1.1 Información abierta

En cuanto a la parte de recogida de información, es principalmente a través del ayuntamiento de Valencia, que es un sitio web de información pública, donde seleccionas la información adecuada y eliges el formato que utilizas para descargar.

En esta parte, nuestro objetivo principal es ver los tipos de varios datos de texto, como: tráfico, entorno, tipos de texto como: CSV, JSON, TXV, XML, etc. En este documento, se usa principalmente el tipo de texto CSV y los diferentes formatos de texto se usan de diferentes maneras. Y esta plataforma tiene una gran cantidad de datos a los que se puede acceder y utilizar de forma gratuita, lo cual es muy conveniente. Y existen tres métodos alternativos de uso de API, tales como: API CKAN, API RTOD, API Dades Georeferenciades, y la plataforma CKAN seleccionada en este artículo, la operación específica se realizará a continuación.



Figura 1.1.1. Información de datos en la página de la generalidad Valenciana.

Open Data es una filosofía y práctica que proporciona la mayor cantidad de información posible para que los ciudadanos la utilicen de forma gratuita. Es para todos, y la información que brinda el sector público incluye: geografía, clima, turismo, economía y otros tipos, con características completas, confiables y de alta calidad, y uso ilimitado. Hay varios formatos de archivo en estos datos para una fácil descarga y uso, como: csv, json, GML, WFS, KML, etc.

1.2. CKAN

CKAN es una herramienta para crear sitios web de datos abiertos. Puede ayudarnos a administrar y publicar datos. Una vez publicados, los usuarios pueden utilizar su función de búsqueda multifacética para explorar y encontrar los datos que necesitan, y utilizar mapas, gráficos y tablas para obtener una vista previa.

Con respecto a la publicación y el uso de datos, primero debemos registrar una cuenta de ckan. Haga clic en "DEMO CKAN" para ingresar a la página de consulta de datos y registrarse o en iniciar sesión en la ubicación de inicio de sesión en la esquina superior derecha.

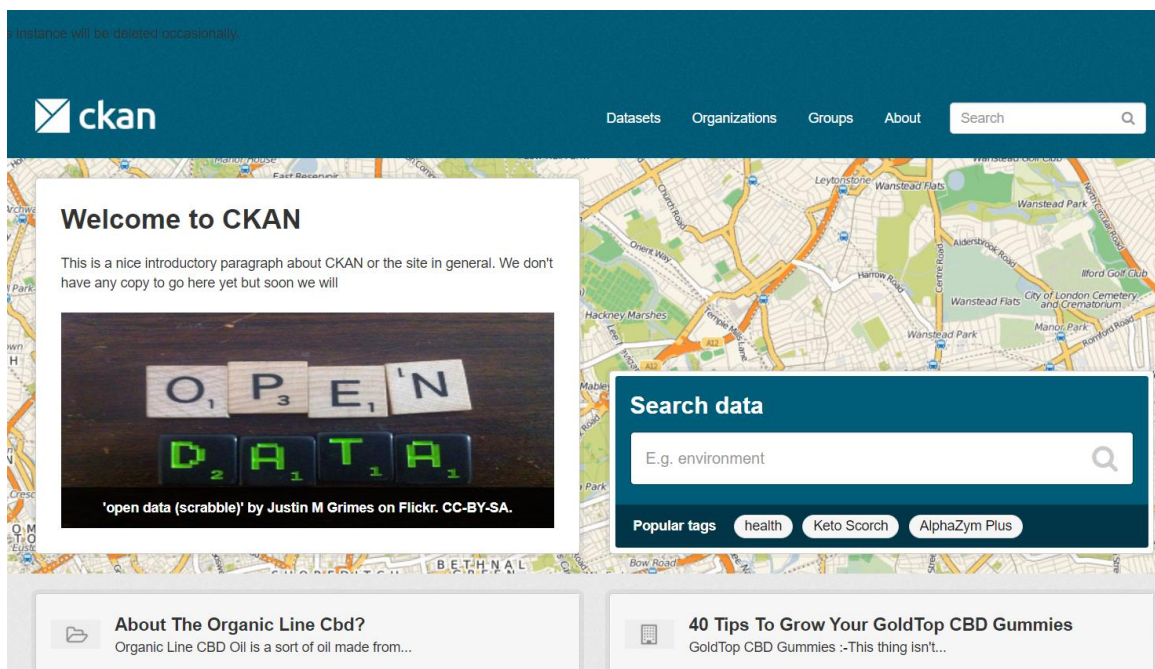


Figura 1.2.1 Página principal de CKAN.

Una vez que el registro es exitoso, lo siguiente que debemos hacer es agregar datos, como se muestra en la figura 1.2.2.

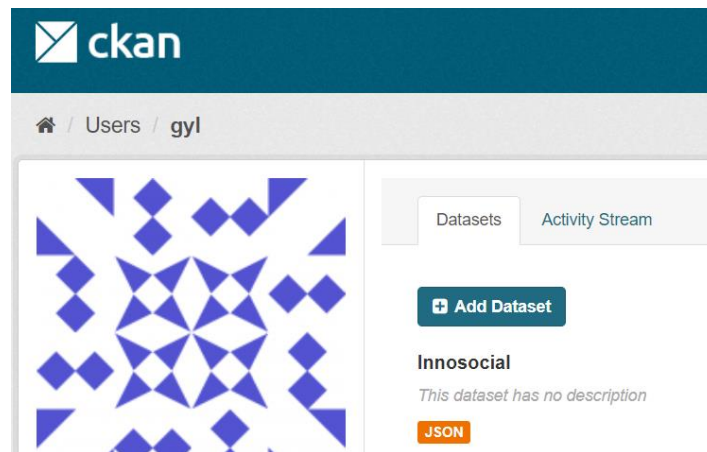



Figura 1.2.2 Perfil creado en la plataforma CKAN.

Estos datos se pueden agregar a través del enlace de la URL o cargarlos directamente de forma local. Por ejemplo, cargamos los datos "Datos diarios del último mes de las estaciones de ruido", esperamos a que se hayan cargado correctamente, para proceder a visualizarlos, ver figura 1.2.3.

Datos diarios del último mes de las estaciones de ruido.

Data and Resources


 ruido.csv [Explore](#)

Additional Info

Field	Value
State	active
Last Updated	February 6, 2021, 1:49 PM (UTC+01:00)
Created	February 6, 2021, 1:42 PM (UTC+01:00)

0 Comments CKAN [Disqus' Privacy Policy](#) [Login](#)

[Recommend](#) [Tweet](#) [Share](#) Sort by Best

 Start the discussion...

LOG IN WITH OR SIGN UP WITH DISQUS

[D](#) [f](#) [t](#) [G](#)

Figura 1.2.3 Muestra de datos cargados correctamente.

1.3. Node-red

Node-red es una herramienta de programación para el Internet de las cosas, esta herramienta nos proporciona un editor de procesos basado en navegador. Como se puede observar en la figura 1.3.1, los nodos están agrupados por módulos, cada módulo contiene varios nodos relacionados con la función que cumplen, así como algunos nodos de uso general.

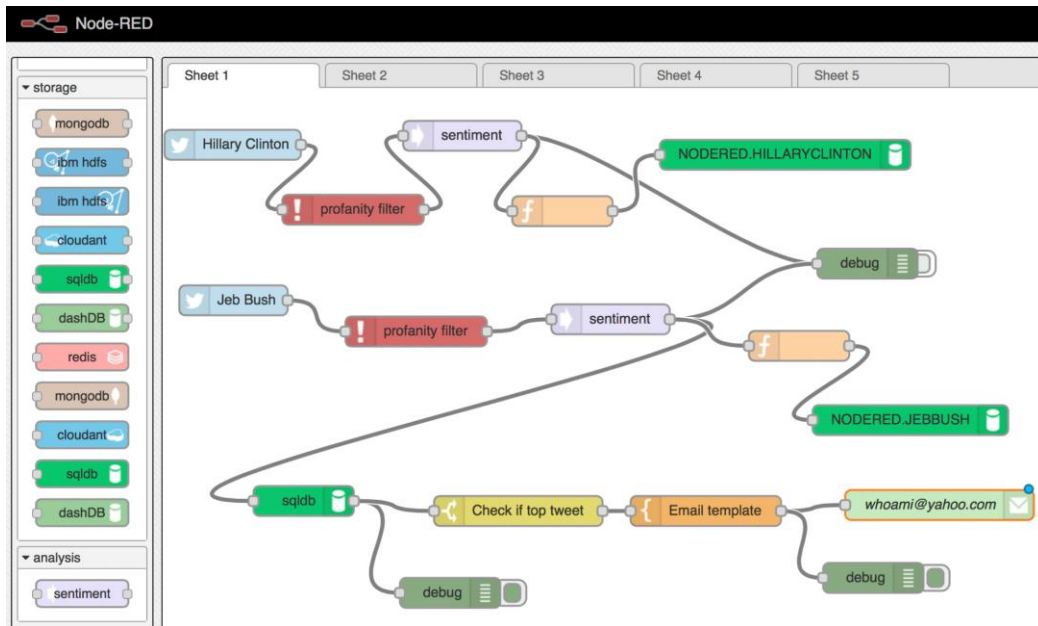


Figura 1.3.1 Ejemplo de la estructura de programación de la herramienta Node-red.

Al ser una herramienta de programación visual, los desarrolladores conectan varios nodos para construir un "flujo", por ejemplo: un nodo de entrada (inyectar), un nodo de procesamiento (función), un nodo de salida (depuración), están conectados entre sí para formar un "flujo" simple.

De acuerdo con las diferentes necesidades, los desarrolladores necesitan seleccionar nodos con diferentes funciones, arrastrarlos al panel de conexión y conectar cada nodo por turno. El funcionamiento es simple y la función del nodo es clara, lo que facilita el proceso de desarrollo.

Node-Red fue originalmente un proyecto de código abierto desarrollado por IBM a finales de 2013 para satisfacer sus necesidades de conectar rápidamente hardware y dispositivos a servicios web y otro software; como un pegamento para Internet de las cosas, se convirtió rápidamente en una especie de herramienta de programación universal para Internet de las cosas. En la actualidad, Node-red se ha convertido en una herramienta importante para los desarrolladores debido a su rápido desarrollo.

1.3.1. Abra y use Node-red

Una vez completada la descarga se abre la ventana cmd como administrador, ingrese node-red, use este comando para abrir el servidor local en el puerto 1880 y luego copie: `http://localhost:1880`, a continuación, en la figura 1.3.1.1 se muestra el proceso en consola.

```
node-red
Microsoft Windows [版本 10.0.18363.1256]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Windows\system32>node-red
16 Dec 22:09:14 - [info]

Welcome to Node-RED
=====

16 Dec 22:09:14 - [info] Node-RED version: v1.2.2
16 Dec 22:09:14 - [info] Node.js version: v12.19.0
16 Dec 22:09:14 - [info] Windows_NT 10.0.18363 x64 LE
16 Dec 22:09:16 - [info] Loading palette nodes
16 Dec 22:09:19 - [info] Settings file : C:\Users\15853\.node-red\settings.js
16 Dec 22:09:19 - [info] Context store : 'default' [module=memory]
16 Dec 22:09:19 - [info] User directory : \Users\15853\.node-red
16 Dec 22:09:19 - [warn] Projects disabled : editorTheme.projects.enabled=false
16 Dec 22:09:19 - [info] Flows file : \Users\15853\.node-red\flows_DESKTOP-156H63H.json
16 Dec 22:09:19 - [info] Server now running at http://127.0.0.1:1880/
16 Dec 22:09:19 - [warn]

-----

Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.
```

Figura 1.3.1.1 Inicio de node-red desde consola.

Copie la URL y pégala en el navegador y ejecútese, al cargarse esta URL en el navegador aparecerá la página de operación Node-red (ver figura 1.3.1.2).

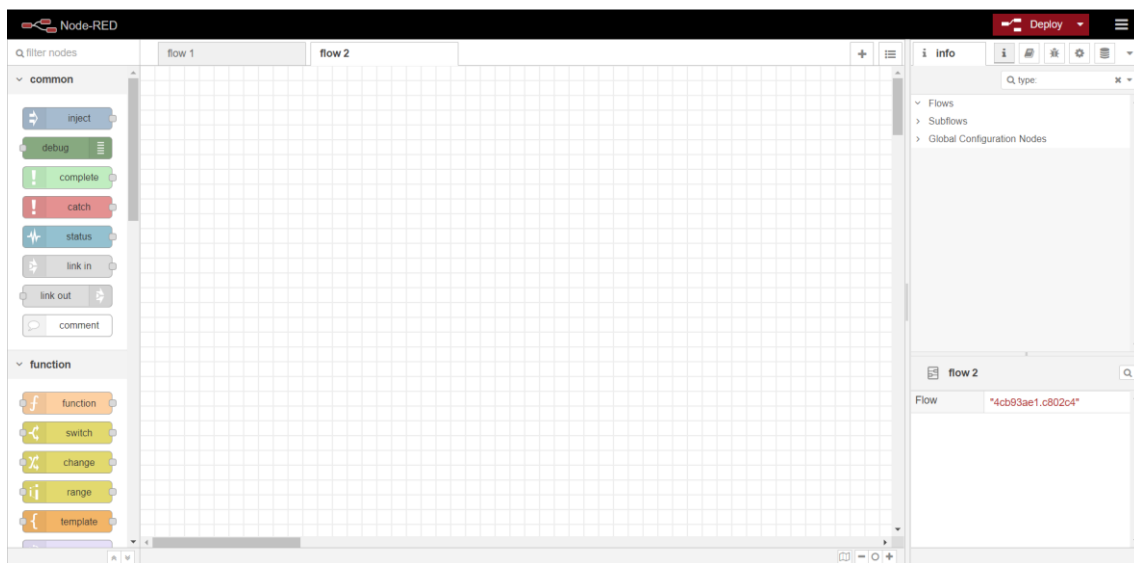


Figura 1.3.1.2 Pantalla inicial de de node-red.

La figura 1.3.1.2 representa la página de operación de node-red, en el lado izquierdo, hay varias categorías como propósito general, procesamiento, etc. Cada categoría contiene múltiples nodos, en la zona central con forma de cuadrícula se puede arrastrar el nodo de la izquierda hacia él para la conexión o el procesamiento en paralelo. Para realizar la conexión de nodos, el extremo posterior del nodo anterior se conecta al extremo frontal del siguiente nodo, para facilitar la transmisión e intercambio de mensajes.



En el lado derecho específicamente en la parte superior está "Implementar", haga clic en este botón después de completar todo el flujo, los siguientes botones son: información, configuración, depuración, ayuda. Haga clic en el nodo, observará la información del nodo en la ventana de información, complete todo el flujo luego observe los atributos de la información en la ventana de depuración.

1.3.2. Funciones de varios nodos

- **Nodo de entrada(inject)** : Permite ingresar mensajes en la secuencia, este proceso se puede activar manualmente haciendo clic en el botón de nodo, o se puede activar periódicamente estableciendo un intervalo de tiempo o un tiempo específico.

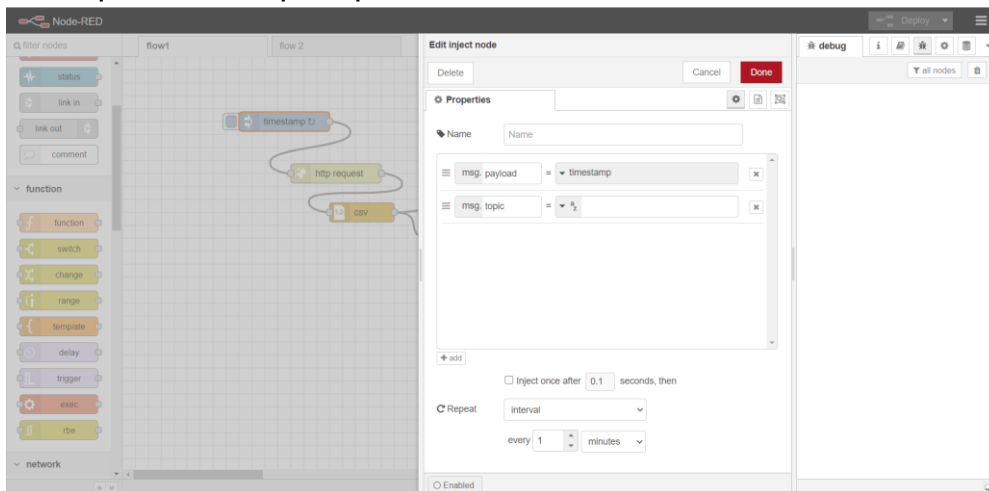


Figura 1.3.2.1 Funciones de varios nodos

Nodo de procesamiento (function) : Permite el procesamiento de mensajes inyectados en el flujo, procesamiento de mensajes de recepción a través de puertos de entrada, edición de código (código JavaScript) y luego enviar un mensaje o varios mensajes a la salida correspondiente.

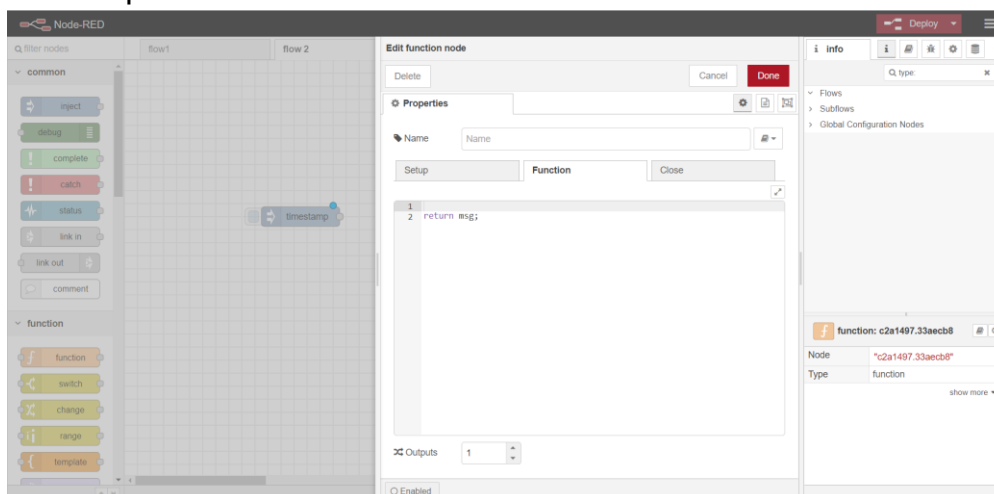


Figura 1.3.2.2 Nodo de procesamiento (function)

Nodo de salida (debug) : Enviará el mensaje completo, observará la información de todos los mensajes a través de la ventana "Debug" u observará seleccionando todos los nodos o un solo nodo.

Nodo leyendo el archivo (file in) : Permite leer el contenido del archivo, ingresar la ubicación del archivo en el lugar de edición y enviarlo en forma de cadena o búfer binario. En la salida de edición, puede seleccionar un mensaje por línea o dividir el archivo en bloques de búfer más pequeños.

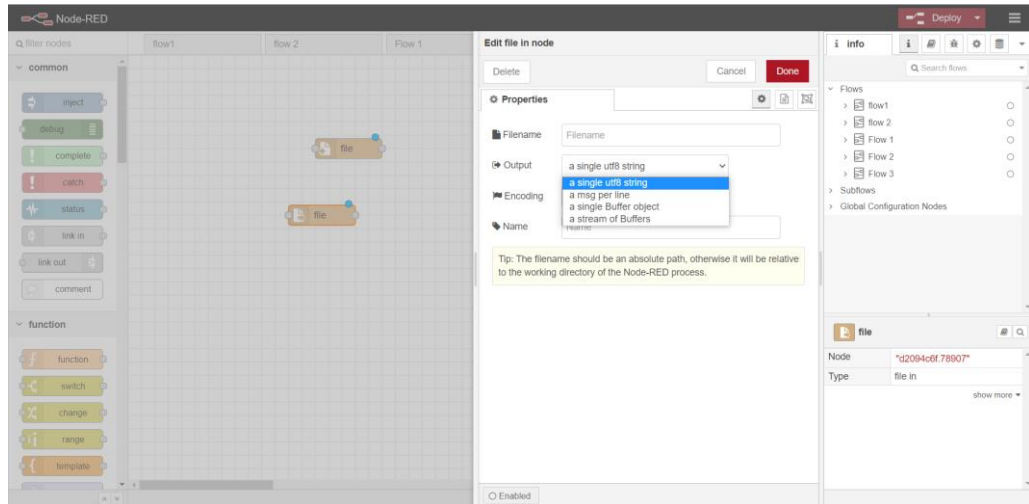


Figura 1.3.2.3 Nodo leyendo el archivo (file in)

1.3.3. Visualización de datos

Cuando utilice node-red para la programación de rutas, puede aparecer el módulo de nodo que desea, que no se puede encontrar a la izquierda. Luego se necesita que agreguemos y seleccionamos "manage palette" en el lado de la derecha inferior.

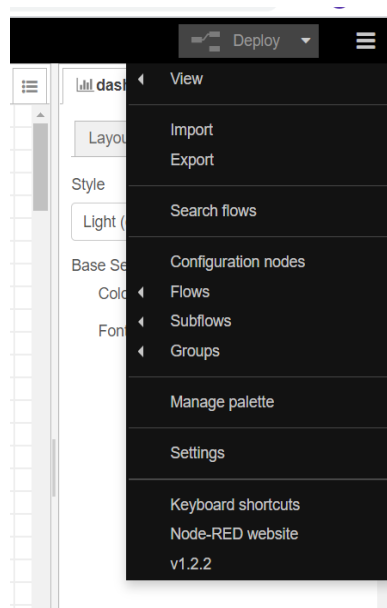


Figura 1.3.3.1 Configuración de atributos de node-red

Después se realiza un clic para abrir la ventana, se busca y se agrega en la paleta la opción de instalar, agregue el módulo del nodo del tablero, haga clic en instalar en la parte inferior derecha para tener éxito, después del éxito, podemos encontrar el módulo del tablero a la izquierda.

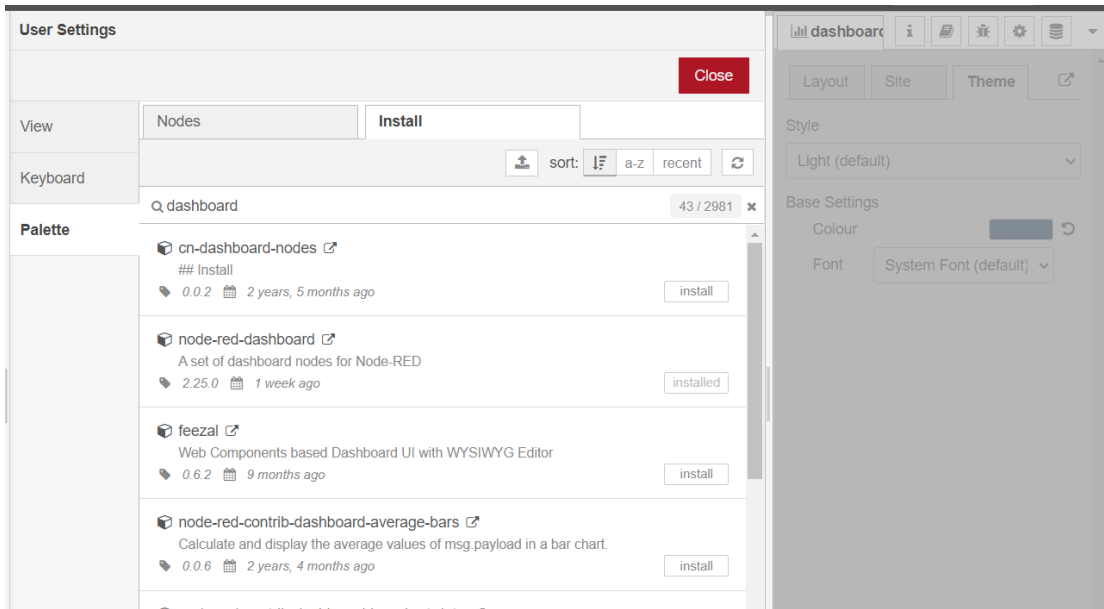


Figura 1.3.3.2 Inicie el módulo de nodo

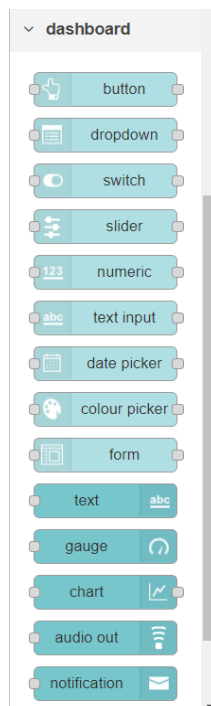


Figura 1.3.3.3 Nodo del módulo dashboard

En estos nodos se puede seleccionar los módulos necesarios, a continuación, se detalla lo que se puede realizar en cada nodo. Por ejemplo, tenemos:

Nodo Slider : Se puede agregar un control deslizante para controlar el valor en la página de datos y ajustar el valor deslizando hacia la izquierda y hacia la derecha, en la página de configuración se puede establecer los valores máximos y mínimos.

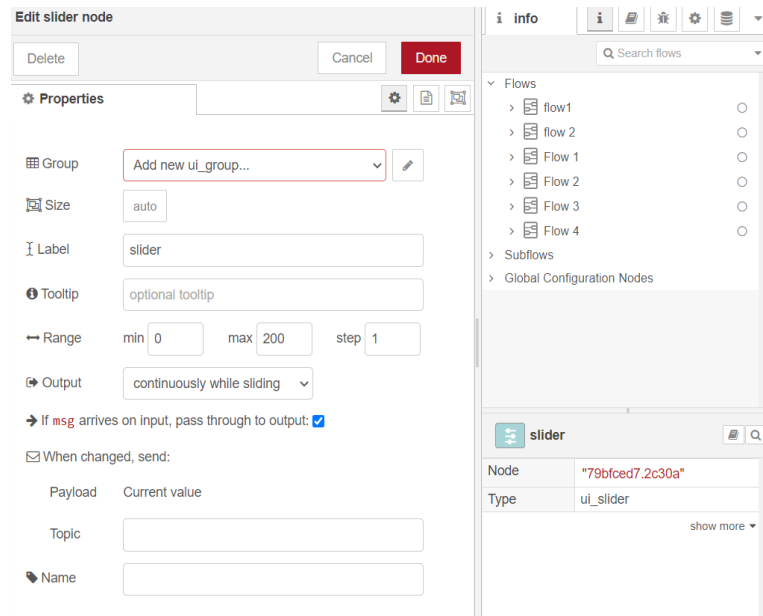


Figura 1.3.3.4 Propiedades del nodo slider

Nodo Gauge : Genera un módulo de tipo de medidor en la interfaz de datos, por ejemplo, se puede vincular con el nodo del control deslizante, y el control deslizante generado por el nodo del control deslizante se deslizará hacia la izquierda y hacia la derecha, y el valor del medidor girará en consecuencia. Los valores máximos y mínimos se pueden configurar en el área de configuración, y el color correspondiente se puede configurar de acuerdo con el rango del valor.

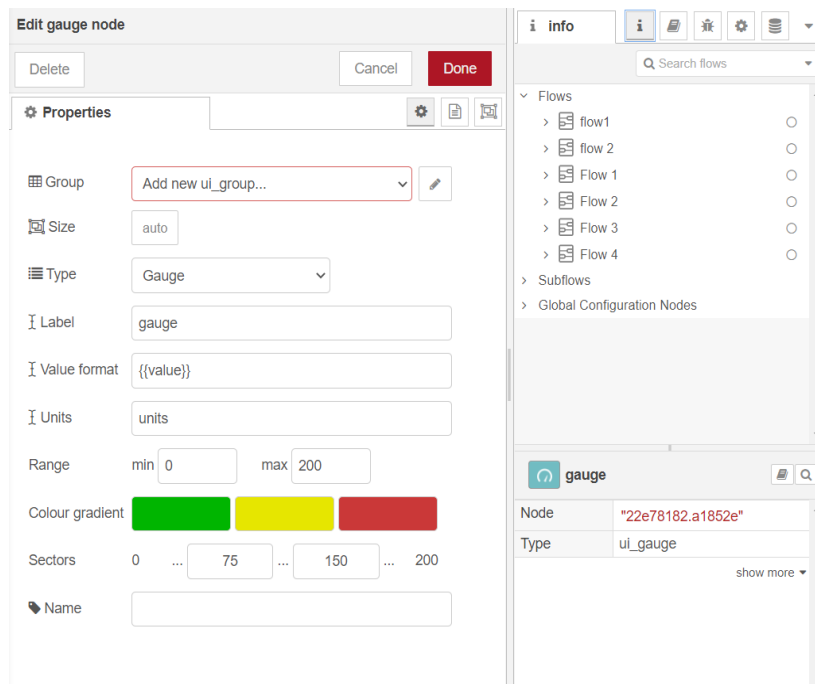


Figura 1.3.3.5 Propiedades del nodo gauge

Nodo Chart : Según el tamaño del valor de entrada, cambia con el tiempo para generar un gráfico (puede ser un gráfico de líneas, un gráfico de barras, etc.).

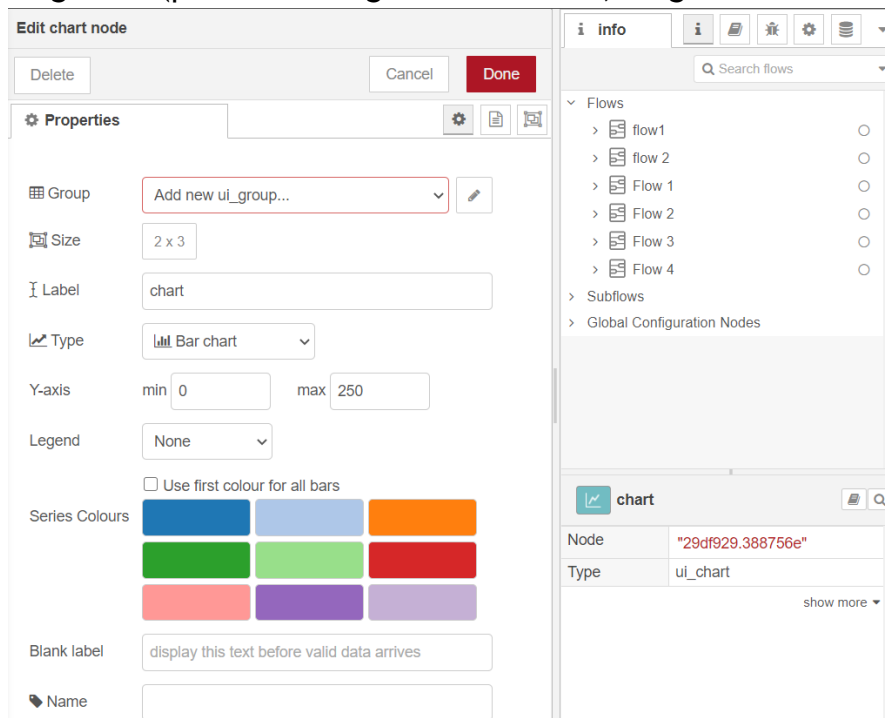


Figura 1.3.3.6 Propiedades del Nodo Chart

Finalmente, simplemente cree un "flujo", abra una nueva ventana del navegador e ingrese: <http://localhost:1880/ui>. Puede generar la página de datos como se muestra en la figura 1.3.3.7.

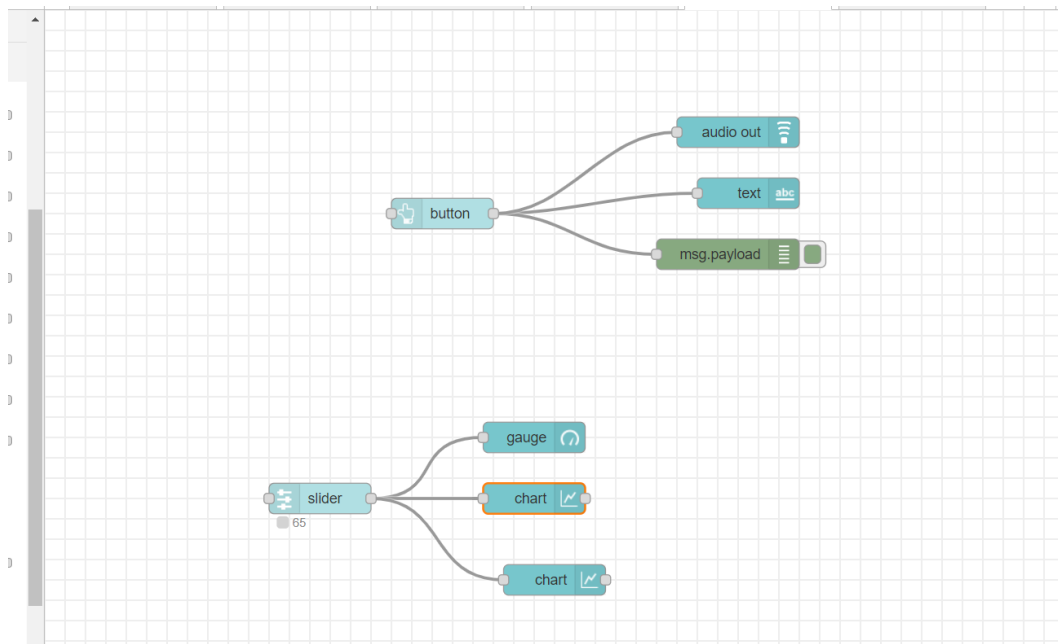


Figura 1.3.3.7 Conexión de nodo simple

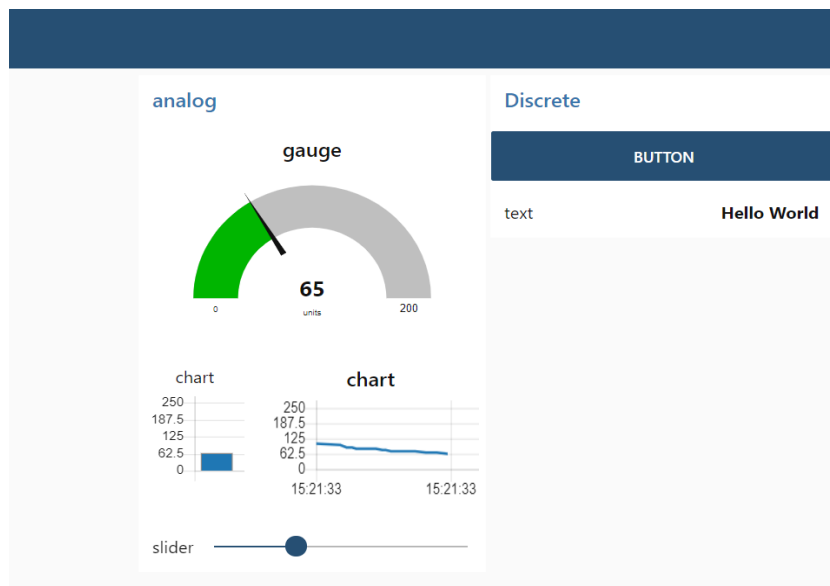


Figura 1.3.3.8 Mostrar el gráfico de datos de la página.

1.4. Raspberry pi

Raspberry Pi es una microcomputadora del tamaño de una tarjeta de crédito, puede ejecutar un sistema operativo completo, lo que significa que puede utilizar varios lenguajes (como Python, java, etc.) y bases de datos para el desarrollo, incluso si ejecuta varios procesos. Esta microcomputadora integra interfaces como cualquier otro

computador, como por ejemplo integra puertos USB-host, RJ45, HDMI, lector de tarjetas SD y otras interfaces comunes.

Aunque la Raspberry Pi es pequeña, tiene funciones completas, fue desarrollada por la Fundación Británica Raspberry Pi para promover la educación básica en informática en las escuelas con hardware de bajo costo y software libre. En este proyecto, estamos utilizando el modelo B de Raspberry Pi 3, que se lanzó en febrero de 2016 y reemplazó el modelo B de Raspberry Pi 2 con el dispositivo Raspberry Pi de tercera generación.



Figura 1.4.1 Plataforma hardware “Raspberry pi”

A continuación, se detallan las funciones principales:

- CPU de 64 bits Broadcom BCM2837 de cuatro núcleos a 1,2 GHz
- 1 GB de RAM
- LAN inalámbrica BCM43438 integrada y Bluetooth de baja energía (BLE)
- 100 base Ethernet
- GPIO extendido de 40 pines
- 4 puertos USB 2
- Salida estéreo de 4 polos y puerto de video compuesto
- HDMI de tamaño completo
- Puerto de cámara CSI, utilizado para conectar la cámara Raspberry Pi
- Puerto de pantalla DSI, utilizado para conectar la pantalla táctil Raspberry Pi
- Puerto micro SD, utilizado para cargar el sistema operativo y almacenar datos
- Interruptor actualizado Fuente de alimentación Micro USB hasta 2A.

1.5. Objetivos

El alcance de este proyecto está definido por los siguientes objetivos:

- Conocer el nodo-rojo y las ventajas y desventajas de su uso.
- Ejecutar algunos "flujos" básicos a través de algunas operaciones simples.

- Revisar y aprender sobre el módulo dashboard de node-red, y algunos usos simples.
- Adquirir texto de datos y utilizar el API a través de la plataforma de datos abiertos CKAN.
- Utilizar código JavaScript sobre node-red y generar datos.
- Conectar los datos de texto adquiridos con node-red, y editarlos mediante código JavaScript y mostrar los datos de texto en la página de visualización del navegador.
- Dar soluciones a los problemas encontrados a lo largo de este proceso.
- Analizar los datos de la interfaz de visualización.



2. Diseño de la solución

En el diseño de este proyecto, se divide principalmente en cuatro módulos, datos de texto CKAN, plataforma de programación node-red, la plataforma hardware Raspberry Pi y módulo de observación de última generación, necesitamos conectar las tres primeras partes juntas. El problema que hay que afrontar en este proceso es cómo combinar perfectamente los tres primeros módulos para que el proyecto se pueda completar con éxito.

2.1. Arquitectura del Sistema

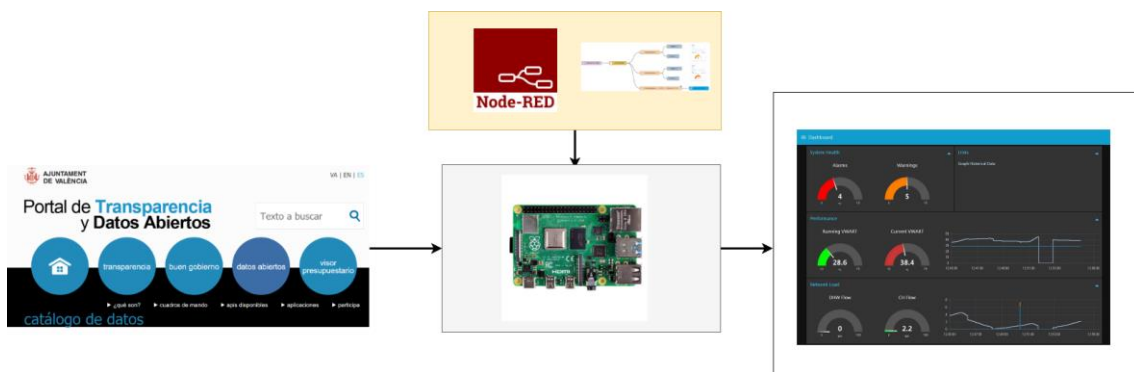


Figura 2.1.1 Arquitectura del Sistema

Como se muestra en la figura 2.2.1, necesitamos completar todo el proyecto a través de tres módulos y todo el proceso se completará utilizando la herramienta node-red, que es la parte más importante. Como primer paso se conecta la Raspberry Pi a la plataforma de programación node-red de manera ilimitada, después se transfiere los datos de texto a node-red utilizando la API, la cual se programará y desplegará en la plataforma node-red para generar la cuarta parte final.

2.2. Diseño Detallado

En todo el módulo, como paso principal debemos seleccionar los datos e incluir el tipo de datos y el formato del texto, y luego seleccionar la API adecuada. En este proyecto, hemos seleccionado la API CKAN. El CKAN puede acceder a la plataforma de datos de forma gratuita, además de utilizar datos y tener una gran cantidad de datos disponibles. Después del registro gratuito, cargue los datos localmente a través del sitio web y podrá obtener los atributos detallados de los datos de texto.

El segundo paso es utilizar código JavaScript para procesar datos de texto en node-red y establecer un "flujo" completo a través de cada módulo. En este proceso, se debe

seleccionar el módulo apropiado para el nodo, incluido el nodo de red que se conecta a los datos de la API y el nodo de función que procesa los datos de texto. Por supuesto, en este proceso también se deben configurar las propiedades del nodo, que se describe en detalle en la parte de la introducción.

El tercer paso también es un paso importante, en este paso necesitamos conectar la plataforma hardware Raspberry Pi a la plataforma node-red de forma inalámbrica. Se producirán varios problemas en este proceso, por ejemplo, el Raspberry Pi no se puede conectar al node-red debido a un error de comando o bien, se produjo un error de conexión, lo que provocó que node-red no pudiera iniciar sesión normalmente. Este paso es un paso propenso a errores en el proceso y también es un paso que requiere un cuidado especial. Y para conectarse a la operación específica, habrá un proceso detallado en la siguiente parte.

El cuarto paso es la parte de generación final, en este paso los primeros tres pasos se preparan para el cuarto paso. En esta parte operaremos principalmente en node-red, y el cuarto paso usará node-red. El módulo dashboard, este módulo puede ayudarnos a obtener gráficos de datos, gráficos de líneas o gráficos de barras. Se puede configurar de acuerdo con la cantidad de datos.

En este proceso es posible que el gráfico de líneas no se pueda observar claramente debido a la gran cantidad de datos, o puede haber un problema con la configuración de atributos del nodo de la tabla, lo que hace que los datos tabla en la página de visualización sea muy grande, etc. Con respecto al problema de mostrar la página, esto se puede configurar en las propiedades del nodo, como las configuraciones izquierda y derecha y las configuraciones arriba y abajo de la tabla.





3. Implementación

3.1 Operación y depuración de Node-Red

Como primer punto es aprender mucho sobre la programación en la herramienta node-red, como segundo punto es descargarlo a una computadora a través del sitio web proporcionado por el profesor. En esta parte, se ha utilizado el sistema operativo Windows 10, en este proyecto se ha descargado y trabajado con la última versión de esta herramienta, proporcionada en el sitio web <https://nodered.org/>. Finalmente se ha realizado una implementación muy simple, formando un "flujo" a través de algunos nodos simples como se observa en la figura 3.1.1.

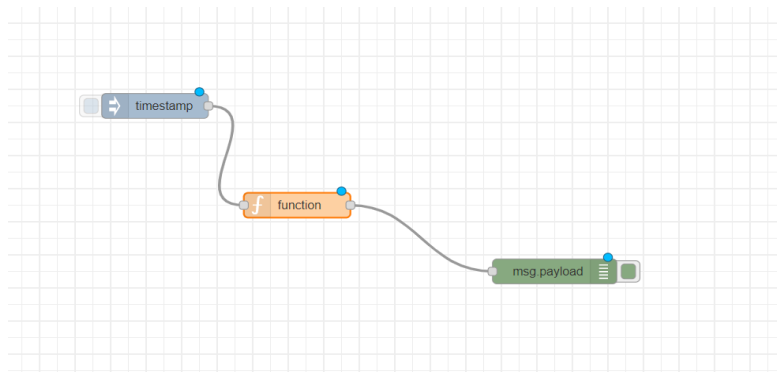


Figura 3.1.1 Flujo simple

Este es el "flujo" más básico, que incluye nodos de entrada, nodos de salida y nodos de procesamiento, los datos se procesan en los nodos de procesamiento, por ejemplo: la escritura de código JavaScript.

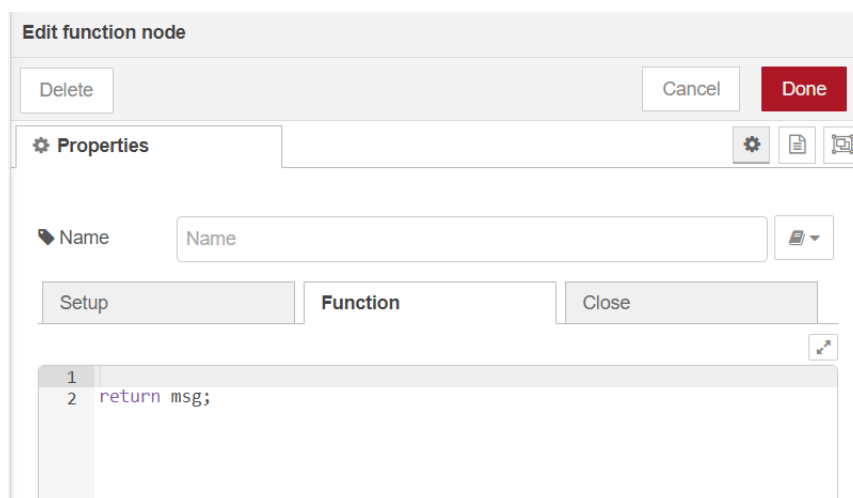


Figura 3.1.2 Nodo de procesamiento

Una vez finalizado el paso anterior, se hace clic en la opción "deploy" en la esquina superior derecha para completar la configuración. Después se realiza una descarga de datos en el sitio web abierto y luego programamos a través de node-red. En este proceso los nodos que usamos son: nodo de inject, nodo de debug, nodo de función, nodo "file in", nodo CSV y el nodo de chart. En la siguiente figura 3.1.3 se muestra la estructura y conexión entre nodos.

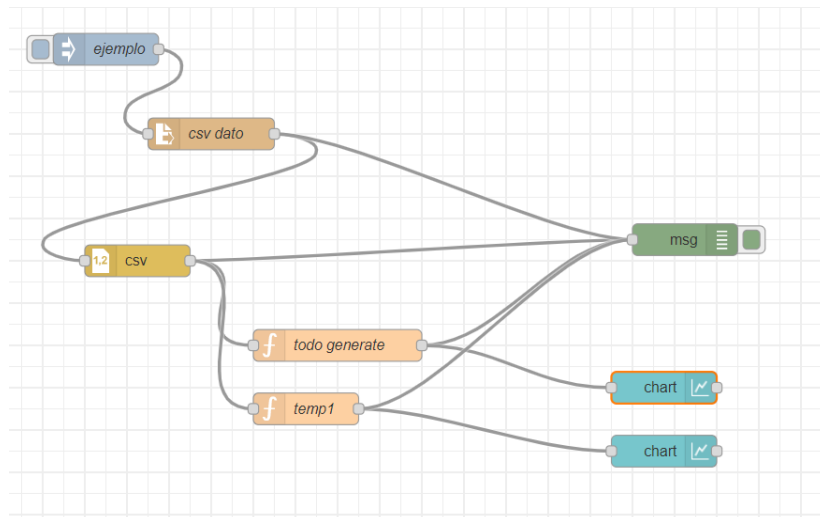


Figura 3.1.3 Procesar los datos de texto adquiridos

En cuanto a la configuración de atributos del nodo, específicamente del nodo "file in", como primer punto tenemos que seleccionar el archivo descargado y agregar la ubicación del archivo y seleccionar "a single utf8 string" para "Output", además seleccionamos la opción "default" para "Encoding", y finalmente ingresamos el nombre del nodo, mediante el botón " Done", se puede hacer (ver la figura 3.1.4).

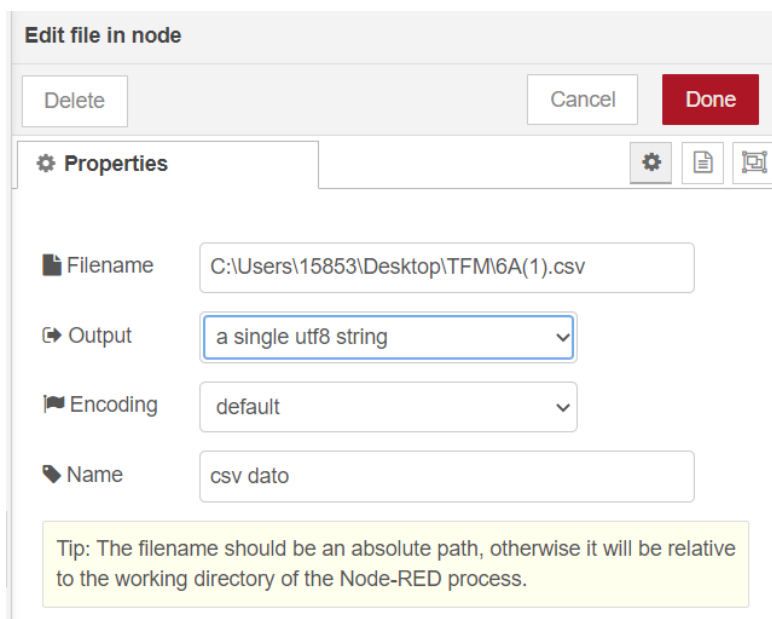


Figura 3.1.4 Atributos de la fuente del texto

En cuanto a la configuración de atributos del nodo csv, se detalla lo siguiente. En la opción "Separator" se ha elegido la configuración "semicolon", en la ventana de opciones de CSV to Object options, en el apartado de "Input" se ha seleccionado dos opciones "first row contains column names" y "parse numerical values", y en el apartado de "Output" se ha seleccionado la opción "a single message[array]". Otras opciones dentro de este nodo no cambian, una vez finalizado toda la configuración se hace clic en el botón "Done"(observar la figura 3.1.5).

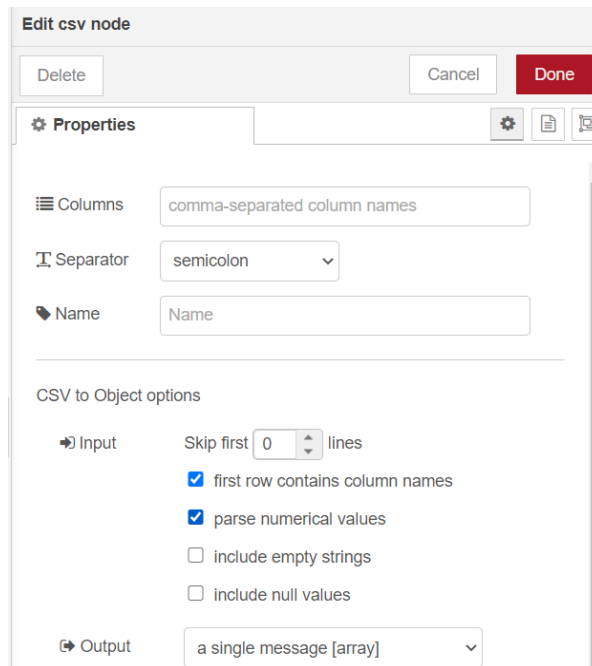


Figura 3.1.5 Configuración del nodo csv

De igual manera se realiza la configuración de atributos del nodo del chart: El primer paso es agregar la opción "group" en el botón Add y rellenar con "Default" en la opción de "Name", en la opción de "Tab" se ha seleccionado la configuración "Add new ui_tab" y en la opción "Width" se ha escrito un valor numérico que depende de la cantidad de datos. Cuantos más datos, mayor es el valor (ver figura 3.1.6).

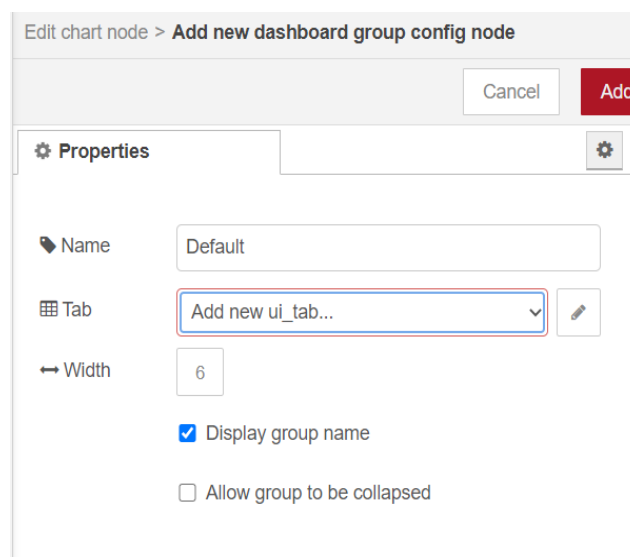


Figura 3.1.6 Agregar nuevo nodo de grupo

Edit chart node > Add new dashboard group config node > Add new dashboard tab config node

Cancel Add

Properties

Name Home

Icon dashboard

State Enabled

Nav. Menu Visible

The **Icon** field can be either a [Material Design icon](#) (e.g. 'check', 'close') or a [Font Awesome icon](#) (e.g. 'fa-fire'), or a [Weather icon](#) (e.g. 'wi-wu-sunny').

You can use the full set of google material icons if you add 'mi-' to the icon name. e.g. 'mi-videogame_asset'.

Figura 3.1.7 Agregar nueva configuración de nodo de pestaña

Después de que la nueva “Tab” en el “Group” se haya creado correctamente, se ha seleccionado el formato de gráfico que se necesita. En estos datos, se ha elegido el gráfico de líneas y se ha seleccionado "Gráfico de líneas" en el campo "Type". Para el eje X, se ha elegido el tiempo como la abscisa, y se ha seleccionado "Year-Mouth-Dat" en la X-axis Label. Para la opción “points” se ha rellenado con un valor de 5000, ya que la cantidad de datos es 3000(ver figura 3.1.8).

Edit chart node

Delete Cancel Done

Properties

Group [Home] Default

Size auto

Label chart

Type Line chart enlarge points

X-axis last 2 weeks OR 5000 points

X-axis Label Year-Month-Date as UTC

Y-axis min max

Legend Show Interpolate linear

Series Colours

Blank label display this text before valid data arrives

Figura 3.1.8 Configuración del nodo de chart

El siguiente paso es la configuración del nodo de función, escribimos el código en él. Aquí hay múltiples nodos de función, para el primer nodo de función se selecciona todos los datos, el valor que representa fecha es la abscisa. Otros tipos Datos, elija la ordenada para escribir el código:

```
function toTimestamp(strDate){
    var datum = Date.parse(strDate);
    return datum;
}
var payload=msg.payload;
var data_out=[];
var a1=[];
var a2=[];
var a3=[];
var a4=[];
var a5=[];
var a6=[];
for(let i=0;i<payload.length;i++)
{
    var Fecha= payload[i].Fecha.split("/");
    var t=Fecha[2]+"/"+Fecha[1]+"/"+Fecha[0];
    var timestamp=toTimestamp(t);
    var temp1={ "x": timestamp, "y":parseInt( payload[i]["PM2.5 (µg/m3)"] ) };
    var temp2={ "x": timestamp, "y":parseInt( payload[i]["SO2 (µg/m3)"] ) };
    var temp3={ "x": timestamp, "y": parseInt( payload[i]["CO (mg/m3)"] ) };
    var temp4={ "x": timestamp, "y": parseInt( payload[i]["PM10 (µg/m3)"] ) };
    var temp5={ "x": timestamp, "y": parseInt( payload[i]["Ozono (µg/m3)"] ) };
    var temp6={ "x": timestamp, "y": parseInt( payload[i]["Veloc. (m/s)"] ) };
    a1.push(temp1);
    a2.push(temp2);
    a3.push(temp3);
    a4.push(temp4);
    a5.push(temp5);
    a6.push(temp6);
    //
}
data_out=[a1,a2,a3,a4,a5,a6];
msg.payload={
    "series": [ "PM2.5 (µg/m3) ", "SO2 (µg/m3) ", "CO (mg/m3) ", "PM10 (µg/m3) ", "Ozono (µg/m3) ", "Veloc. (m/s) "],
    "data":data_out,
    "labels": [ "PM2.5 (µg/m3) ", "SO2 (µg/m3) ", "CO (mg/m3) ", "PM10 (µg/m3) ", "Ozono (µg/m3) ", "Veloc. (m/s) " ]
};
return msg;
```

Figura 3.1.9 Código para procesar datos de texto

El nodo Función procesa los datos, por lo tanto, una vez completada la configuración del nodo, haga clic en "deploy" para configurar, y luego haga clic en el nodo de inject, puede ver la salida de datos a la derecha, como se muestra en la figura 3.1.10.

```
▼ object
  _msgid: "19ca62c3.54534d"
  ▼ payload: array[1]
    ▼ 0: object
      series: array[6]
        0: "PM2.5 (µg/m³)"
        1: "SO2 (µg/m³)"
        2: "CO (mg/m³)"
        3: "PM10 (µg/m³)"
        4: "Ozono (µg/m³)"
        5: "Veloc. (m/s)"
      data: array[6]
        0: array[2465]
        1: array[2465]
        2: array[2465]
        3: array[2465]
        4: array[2465]
        5: array[2465]
      labels: array[6]
      topic: ""
      filename:
        "C:\Users\15853\Desktop\TFM\6A(1).c
        sv"
```

Figura 3.1.10 Datos resultantes

En el navegador Google Chrome, se añade una nueva ventana y se ingresa la URL: <http://127.0.0.1:1880/ui>. Puede ver el cuadro de datos generado por node-red, como se muestra en la siguiente figura 3.1.11.

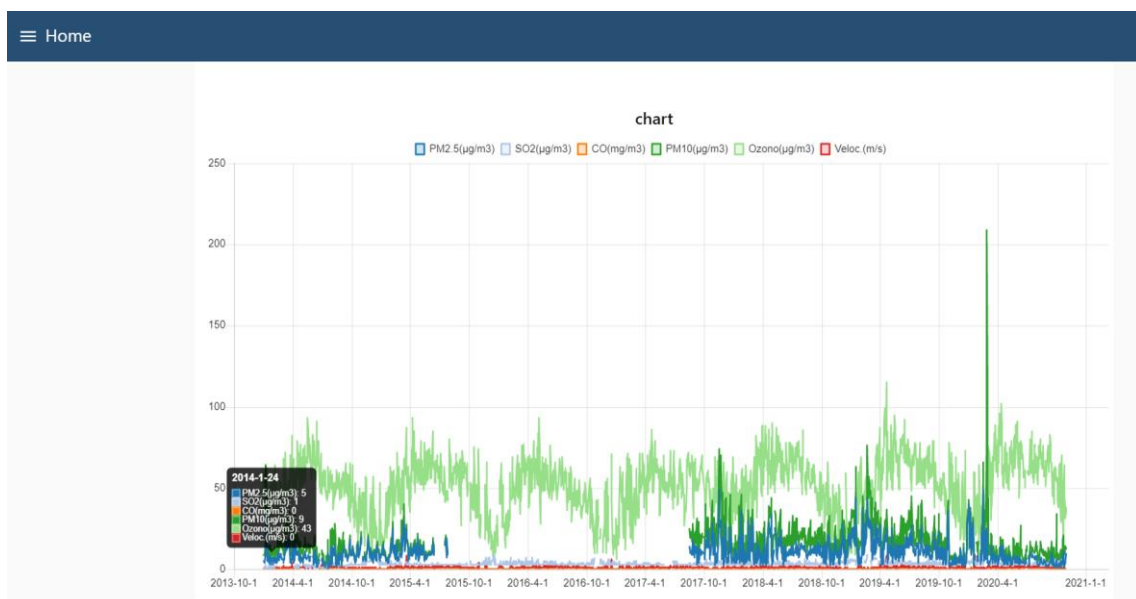


Figura 3.1.11 Mostrar los datos de la página

Por supuesto, también podemos procesar el nodo de función, modificar el código, seleccionar uno de los tipos de datos y generar un gráfico de datos. Por ejemplo, el tiempo de generación es la abscisa y “pm2.5” en la ordenada. Observar en la figura 3.1.12 el código correspondiente a este nodo.




```

1 function toTimestamp(strDate) {
2     var datum = Date.parse(strDate);
3     return datum;
4 }
5 var payload=msg.payload;
6 var data_out=[];
7 var a1=[];
8 for(let i=0;i<payload.length;i++)
9 {
10 var Fecha= payload[i].Fecha.split("/");
11 var t=Fecha[2]+"/"+Fecha[1]+"/"+Fecha[0];
12 var timestamp=toTimestamp(t);
13 var temp1={ "x": timestamp, "y":parseInt( payload[i]["PM2.5(µg/m3)"]) };
14 a1.push(temp1);
15 //
16 }
17 data_out=a1;
18 msg.payload=[{
19 "series": ["PM2.5 (µg/m3)"],
20 "data":data_out,
21 "labels": ["PM2.5 (µg/m3)"]
22 }];
23 return msg;

```

Figura 3.1.12 Código de procesamiento de datos PM2.5

Una vez escrito el código, se hace clic en el botón "depoly" para re-configurar y actualizar el sitio web: <http://127.0.0.1:1880/ui>, y luego podrá ver el gráfico de líneas con solo datos de "PM2.5" como se muestra en la figura 3.1.13.

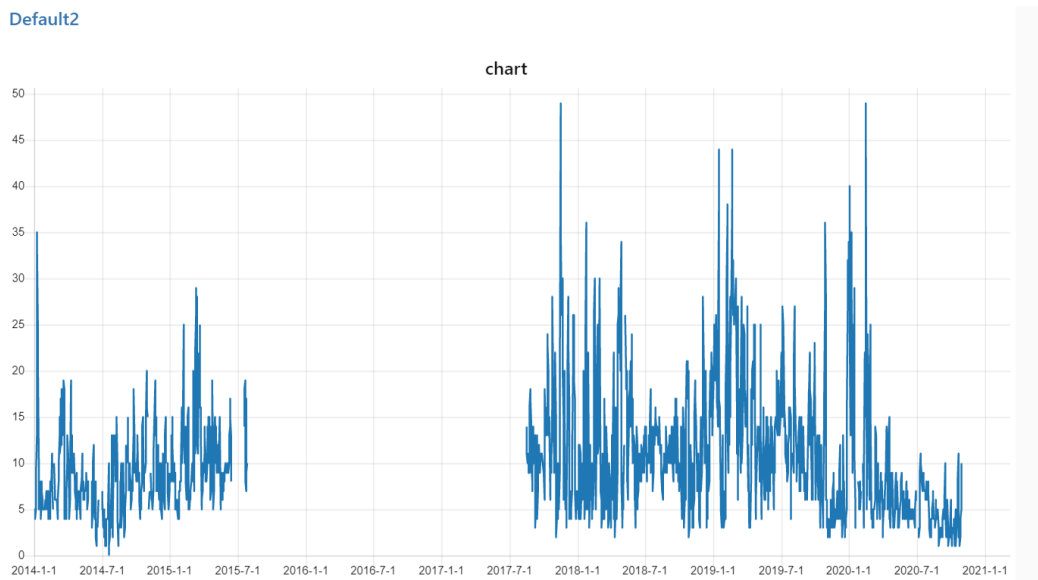


Figura 3.1.13 Datos generados

3.2. Sistema operativo del Raspberry Pi

Como primer paso es completar el aprendizaje de node-red y realizar la práctica de algunos ejemplos. En el segundo paso, nos conectaremos en la plataforma hardware Raspberry Pi. Se ha utilizado todo el material de los laboratorios.

Primero, se necesita sacar la tarjeta de memoria de la Raspberry Pi. La tarjeta de memoria tiene una capacidad de 16G, por lo se ha descargado el software a través del sitio web oficial de Raspberry Pi "<https://www.raspberrypi.org/software/>". Después se realiza la instalación del sistema operativo en la Raspberry Pi, ver figura 3.2.1.

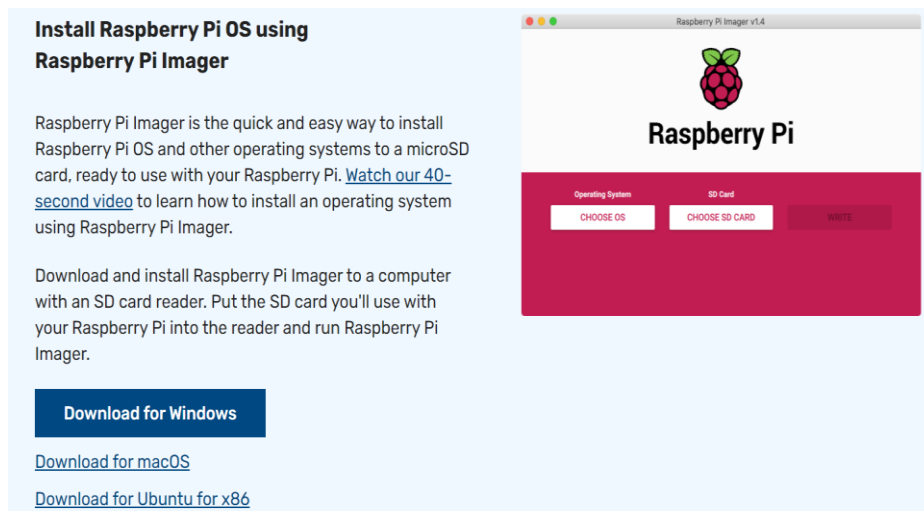


Figura 3.2.1 Descarga el sistema a la Raspberry Pi

El sistema operativo que actualmente disponemos es el sistema Windows 10, así que se hace clic en "Download for Windows". Una vez que la descarga se haya realizado correctamente, se procede a instalar el sistema, conectando la tarjeta SD extraída del computador. (ver figura 3.2.2)

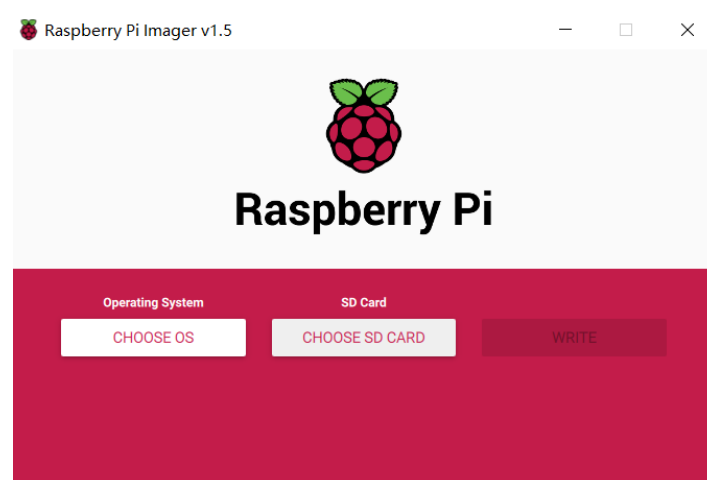


Figura 3.2.2 Imagen del SO. para el Raspberry Pi

Primero se debe seleccionar el sistema operativo, seleccionar el sistema operativo que desea instalar en "CHOOSE OS", luego seleccionar la ubicación de la tarjeta SD y

finalmente hacer clic en "WRITE" para instalar. El proceso de instalación tomará algún tiempo. Durante este tiempo, descargamos algunas herramientas auxiliares, como: Angry IP Scanner (Angry IP Scanner es un escáner de puertos y direcciones IP muy rápido), descargue desde su sitio web oficial "https://angryip.org/download/#linux".

Una vez que el sistema operativo se encuentre en la tarjeta SD y se haya instalado correctamente, saque la tarjeta SD y cargue en la Raspberry Pi. En este proceso, necesitamos un cable de red para realizar algunas conexiones simples, como se muestra en la figura 3.2.3.



Figura 3.2.3 Raspberry Pi conectado al enrutador

Conecte la fuente de alimentación, conecte el cable de red y listo para usar, podemos observar que la luz indicadora en la fuente de alimentación se enciende y los indicadores rojo y verde se muestran en la interfaz del cable de red. Luego debe usar la herramienta auxiliar que descargamos anteriormente: Angry IP Scanner, hacer clic para abrir y modificar la configuración: El "Rango de IP" se establece en "192.168.1.0", en "192.168.1.255" y luego haga clic en "Comenzar" para escanear. Ver figura 3.2.4.

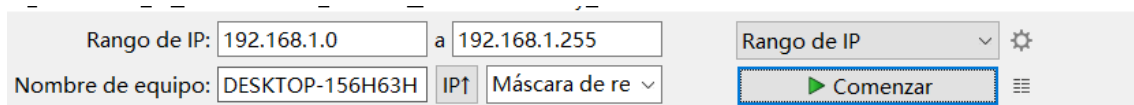


Figura 3.2.4 Configuración de la propiedad de rango de IP

Además, en la siguiente figura 3.2.5 como muestra cómo se puede encontrar información sobre Raspberry Pi

192.168.1.127	[n/a]	[n/s]	[n/s]
192.168.1.128	2 ms	raspberrypi	[n/a]
192.168.1.129	[n/a]	[n/s]	[n/s]

Figura 3.2.5 Encuentra la dirección de Raspberry Pi

Se tiene que registrar la dirección IP de la Raspberry Pi, a continuación, utilizaremos la herramienta Putty para conectarnos a la Raspberry Pi de forma remota. Esta herramienta se puede descargar del sitio web oficial "<https://www.putty.org/>". Después de que la descarga y la instalación sean exitosas, debemos ingresar la dirección IP de la Raspberry Pi en "HostName", en la opción "Connection type" seleccionamos "SSH", y luego el otro sin cambios, se hace clic en el botón "Open", como se muestra en la figura 3.2.6.

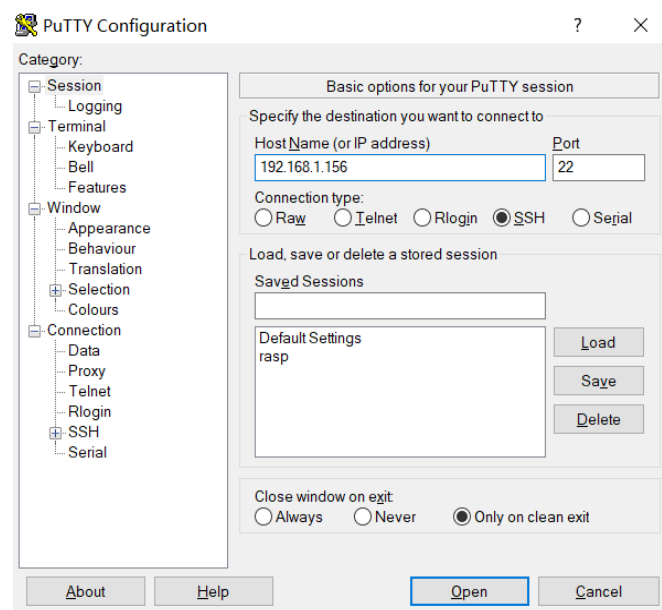


Figura 3.2.6 Configuración remota hacia la Raspberry Pi

Después de ingresar, debe ingresar el nombre de usuario y la contraseña, e iniciar sesión usando la conexión Raspberry Pi, generalmente, tanto el nombre de usuario y la contraseña son las siguientes:

Nombre de usuario: pi

Contraseña: raspberry

Para iniciar sesión correctamente, necesitamos realizar algunas operaciones de instrucción en él, como: descargar node-red, usando instrucciones en raspberry, a continuación, se detalla el comando en consola.

```
"sudo apt-get install build-essential tk-dev libncurses5-dev libncursesw5-dev libreadline6-dev libdb5.3-dev libgdbm-dev libsqlite3-dev libssl-dev libbz2-dev libexpat1-dev liblzma-dev zlib1g-dev libffi-dev -y"
```

Después se utilizará las instrucciones del script para instalar Node.js, npm y Node-RED en la Raspberry Pi, con la siguiente instrucción se puede descargar la nueva versión.

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installer/s/master/deb/update-nodejs-and-nodered)
```

Figura 3.2.7 Instrucción para instalar Node.js y NodeRed

Una vez completado, ingrese el comando "node-red" en consola y ejecute, a continuación, en la figura 3.2.8 se observa más detallado todo este proceso, donde se observa la URL generada.

```
pi@raspberrypi:~ $ node-red
8 Feb 18:43:47 - [info]

Welcome to Node-RED
=====

8 Feb 18:43:47 - [info] Node-RED version: v1.0.6
8 Feb 18:43:47 - [info] Node.js version: v10.23.1
8 Feb 18:43:47 - [info] Linux 5.4.83-v7+ arm LE
8 Feb 18:43:52 - [info] Loading palette nodes
8 Feb 18:43:58 - [info] Dashboard version 2.27.0 started at /ui
8 Feb 18:43:59 - [info] Settings file : /home/pi/.node-red/settings.js
8 Feb 18:43:59 - [info] Context store : 'default' [module=memory]
8 Feb 18:43:59 - [info] User directory : /home/pi/.node-red
8 Feb 18:43:59 - [warn] Projects disabled : editorTheme.projects.enabled=false
8 Feb 18:43:59 - [info] Flows file : /home/pi/.node-red/flows_raspberrypi.json
8 Feb 18:43:59 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

8 Feb 18:43:59 - [info] Starting flows
8 Feb 18:43:59 - [info] Started flows
8 Feb 18:43:59 - [info] Server now running at http://127.0.0.1:1880/
```

Figura 3.2.8 Conéctese a la página de node-red

En este proceso copiamos la URL y modificamos "127.0.0.1" a la dirección IP de Raspberry Pi, se abre en el navegador y se mostrará la interfaz node-red. La visualización exitosa de la interfaz indicará que node-red se ejecuta correctamente en la Raspberry Pi.

En esta parte, realizaremos la instalación y puesta en marcha del sistema, así como las pruebas. Lo principal es la prueba y depuración de node-red, la instalación y la depuración del sistema de Raspberry Pi. A través de las operaciones anteriores, se han resuelto todos los problemas encontrados, tales como: la conexión inalámbrica de la plataforma hardware Raspberry Pi y Node-Red, nodo. Además, se ha utilizado el módulo dashboard en rojo para generar con éxito los datos de la interfaz de pantalla. Todo el proceso de puesta en servicio se completó con éxito, lo que nos indica que el proyecto va en un buen camino para su finalización.



4. Pruebas

Para realizar las pruebas primero se abre CKAN y agregaremos los datos respectivos. Puede descargar los datos utilizados localmente o cargarlos directamente desde el sitio web. Los datos que se han seleccionado para este proyecto inicialmente son sobre el medio ambiente, como se muestra en la figura 4.1.1 los datos se cargan correctamente.

<http://mapas.valencia.es/WebsMunicipales/uploads/atmosferica/1A.csv>

Data and Resources

1A.csv

Explore

Preview
Download
Edit

Field	Value
State	active
Last Updated	February 8, 2021, 3:30 AM (UTC+01:00)
Created	February 8, 2021, 3:30 AM (UTC+01:00)

Figura 4.1.1 Atributos de los datos de texto

A continuación, se hace clic en el botón “Explore” y la opción “Preview” a la derecha para ver el contenido específico de los datos, ver la figura 4.1.2.

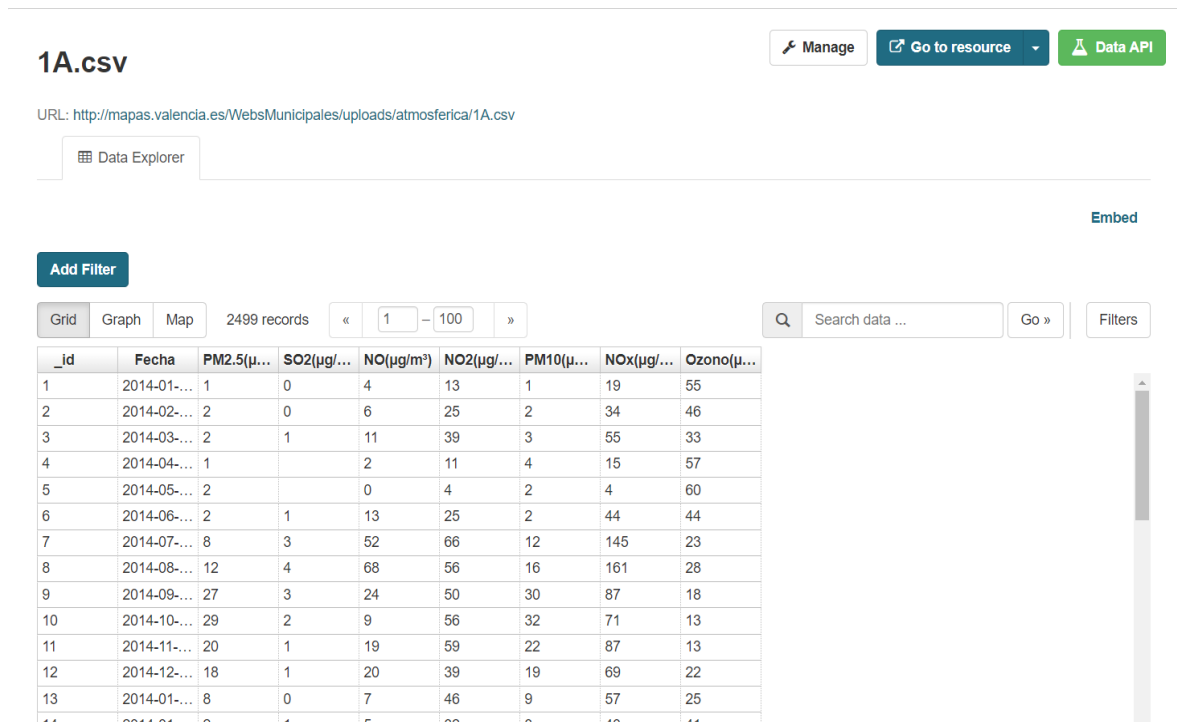


Figura 4.1.2 Datos de texto

Más adelante se utilizarán estos datos en node-red, debido a la gran cantidad de datos se han seleccionado algunos de ellos para utilizarlos, para ello se hace clic en el "Data API" a la derecha para seleccionar el ejemplo "https://demo.ckan.org/api/3/action/datastore_search?resource_id=2ea43afd-e5d0-4689-a0fc-28da18b03aed&limit=5", cambie el valor del límite a 50, como se muestra en la figura 4.1.3.

Access resource data via a web API with powerful query support. Further information in the [main CKAN Data API and DataStore documentation](#).

Endpoints »

The Data API can be accessed via the following actions of the CKAN action API.

Create	<code>https://demo.ckan.org/api/3/action/datastore_create</code>
Update / Insert	<code>https://demo.ckan.org/api/3/action/datastore_upsert</code>
Query	<code>https://demo.ckan.org/api/3/action/datastore_search</code>
Query (via SQL)	<code>https://demo.ckan.org/api/3/action/datastore_search_sql</code>

Querying »

Query example (first 5 results)
`https://demo.ckan.org/api/3/action/datastore_search?resource_id=2ea43afd-e5d0-4689-a0fc-28da18b03aed&limit=5`

Query example (results containing 'jones')
`https://demo.ckan.org/api/3/action/datastore_search?q=jones&resource_id=2ea43afd-e5d0-4689-a0fc-28da18b03aed`

Query example (via SQL statement)
`https://demo.ckan.org/api/3/action/datastore_search_sql?sql=SELECT * from "2ea43afd-e5d0-4689-a0fc-28da18b03aed" WHERE title LIKE 'jones'`

Figura 4.1.3 API de datos de Ckan

Como siguiente paso, se agrega el URL al nodo "http request" para su respectivo uso. En el nodo "http request" cambiamos el contenido de Return a "a parsed JSON object" y establecemos el nombre en "1A".

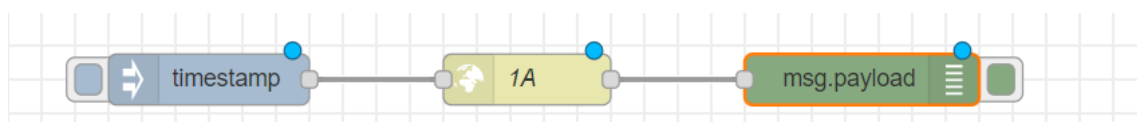


Figura 4.1.4 Obtenga un flujo de datos de texto

Después se hace clic en el botón "deploy" para configurar y luego haga clic en el nodo de inyección para generar datos y realizar el procesamiento de código basado en los datos, ver figura 4.1.5.

```

2021/2/9上午1:18:39 node: 9665082b.b5e3a8
msg.payload: Object
  object
    help:
      "https://demo.ckan.org/api/3/action/help_show?name=datastore_search"
    success: true
    result: object
      include_total: true
      resource_id: "2ea43afd-e5d0-4689-a0fc-28da18b03aed"
      fields: array[9]
      records_format: "objects"
      records: array[50]
        [0 ... 9]
          0: object
            _id: 1
            Fecha: "2014-01-01T00:00:00"
            PM2.5(µg/m³): 1
            SO2(µg/m³): 0
            NO(µg/m³): 4
            NO2(µg/m³): 13
            PM10(µg/m³): 1
            NOx(µg/m³): 19
            Ozono(µg/m³): 55

```

Figura 4.1.5 Datos recibidos

Los datos generados incluyen tiempo, PM2.5, NO, SO2 y otros tipos de datos, así como el valor de cada tipo en un tiempo específico. Se los ha clasificado incluyendo tres tipos: el tipo de PM cambia según el tiempo, el tipo de NO según cambios de hora y otros tipos de contaminación del aire. Entonces se los muestra en forma de tabla a través del nodo "table", posteriormente se conecta el nodo "función" para procesar, y luego se usa el nodo "table" para mostrarlo en forma de tabla. Ver figura 4.1.6.

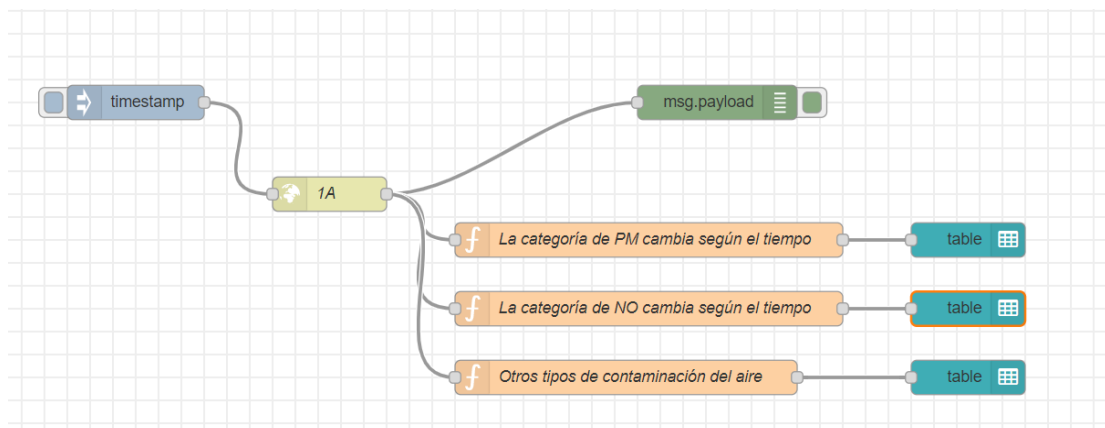


Figura 4.1.6 Procesamiento de tipos de datos

A continuación, se escribe el procesamiento de código en "función" y luego se cambia el nombre, por ejemplo, el código de procesamiento de la clase PM se muestra en la siguiente figura 4.1.7.

```

1 payload = msg.payload.result.records;
2 data = [];
3 for(var i=0;i<payload.length;i++){
4     element = {
5         "Fecha":payload[i]["Fecha"],
6         "PM2(µg/m³)":payload[i]["PM2.5(µg/m³)"],
7         "PM10(µg/m³)":payload[i]["PM10(µg/m³)"]
8     }
9     data.push(element);
10 }
11 msg.payload = data;
12 return msg;

```

Figura 4.1.7 Procesamiento de código de PM2 y pm10

Después de que la configuración sea exitosa, escriba el siguiente URL. "http://192.168.1.156:1880/ui" en una ventana del navegador abra la página del gráfico de visualización y verá tres gráficos que cambian según el tiempo, como se muestra en la figura 4.1.8.

DefaultA			DefaultB			DefaultC			
Fecha	PM2(µg/...	PM10(µ...	Fecha	NO(...	NO...	NOx...	Fecha	SO2(µg/...	Ozono(...
2014-01-01T...	1	1	2014-01-...	4	13	19	2014-01-01T...	0	55
2014-02-01T...	2	2	2014-02-...	6	25	34	2014-02-01T...	0	46
2014-03-01T...	2	3	2014-03-...	11	39	55	2014-03-01T...	1	33
2014-04-01T...	1	4	2014-04-...	2	11	15	2014-04-01T...		57
2014-05-01T...	2	2	2014-05-...	0	4	4	2014-05-01T...		60
2014-06-01T...	2	2	2014-06-...	13	25	44	2014-06-01T...	1	44
2014-07-01T...	8	12	2014-07-...	52	66	145	2014-07-01T...	3	23
2014-08-01T...	12	16	2014-08-...	68	56	161	2014-08-01T...	4	28
2014-09-01T...	27	30	2014-09-...	24	50	87	2014-09-01T...	3	18
2014-10-01T...	29	32	2014-10-...	9	56	71	2014-10-01T...	2	13
2014-11-01T...	20	22	2014-11-...	19	59	87	2014-11-01T...	1	13

Figura 4.1.8 Mostrar los datos de la página

Anteriormente, la cantidad de datos que seleccionamos era 50, puede ver los datos completos deslizando hacia arriba y hacia abajo a través de la "barra azul" en el lado derecho de cada gráfico. A continuación, crearemos un gráfico de líneas basado en las tres categorías que fue creado antes del nodo, el nodo no cambiará por lo que continúe agregando el nodo "función" y después del nodo "http request", porque tenemos tres categorías, continuamos creando tres gráficas de línea, luego tres "función" nodos y tres "chart" nodos, por ejemplo, código PM. Ver figura 4.1.9.

```

1 ▾ function getFecha(strDate){
2     var datum = Date.parse(strDate);
3     return datum;
4 ▾ }
5 payload = msg.payload.result.records;
6 data1 = [];
7 data2 = [];
8 datas = [];
9 for(let i=0;i<payload.length;i++)
10 ▾ {
11     var newFecha = payload[i]["Fecha"].split("T")[0];
12     var Data= newFecha.split("-");
13     var t=Data[2]+"/"+Data[1]+"/"+Data[0];
14     var fecha=getFecha(t);
15     var element1={ "x": fecha, "y":parseInt(payload[i]["PM2.5(µg/m³)"]) };
16     var element2={ "x": fecha, "y":parseInt(payload[i]["PM10(µg/m³)"]) };
17     data1.push(element1);
18     data2.push(element2);
19 ▾ }
20 datas.push(data1);
21 datas.push(data2);
22 ▾ msg.payload={
23     "series": ["PM2.5", "PM10"],
24     "data":datas,
25     "labels": ["PM2.5", "PM10"]
26 ▾ };
27 return msg;

```

Figura 4.1.9 Los códigos de PM2.5 y PM10 cambian con el tiempo

Una vez completado el nodo "función" debemos configurar el nodo "chart", para crear una página con el gráfico anterior, primero debemos seleccionar la agrupación correspondiente al gráfico anterior, y luego debemos seleccionar el tiempo, el tiempo en los datos tiene el formato "año-mes-día", por lo que debemos cambiar el contenido de "X-axis Label" y seleccionar "Year-Month-Date", los demás no cambiarán.

En este proceso también se necesita verificar si el código es correcto, luego, debemos ver si los datos de salida son correctos, posteriormente debemos agregar un nodo de "debug" conectado a la parte posterior de cada nodo de "función", entonces se hace clic en "deploy" después del éxito haga clic en el nodo "inject" para ver si los datos se muestran correctamente en la pantalla correcta, hay problemas como espacios en blanco en los datos, pero si los datos se muestran correctamente entonces continuamos, después debemos abrir la página de visualización anterior para observar la generación de datos, ver la siguiente figura 4.1.10.

Al ver la generación de esta página, nos indica que hemos utilizado con éxito los datos para realizar operaciones visuales en node-red. El registro de operación en node-red se puede ver en el Raspberry Pi conectado a putty. Ver figura 4.1.11.

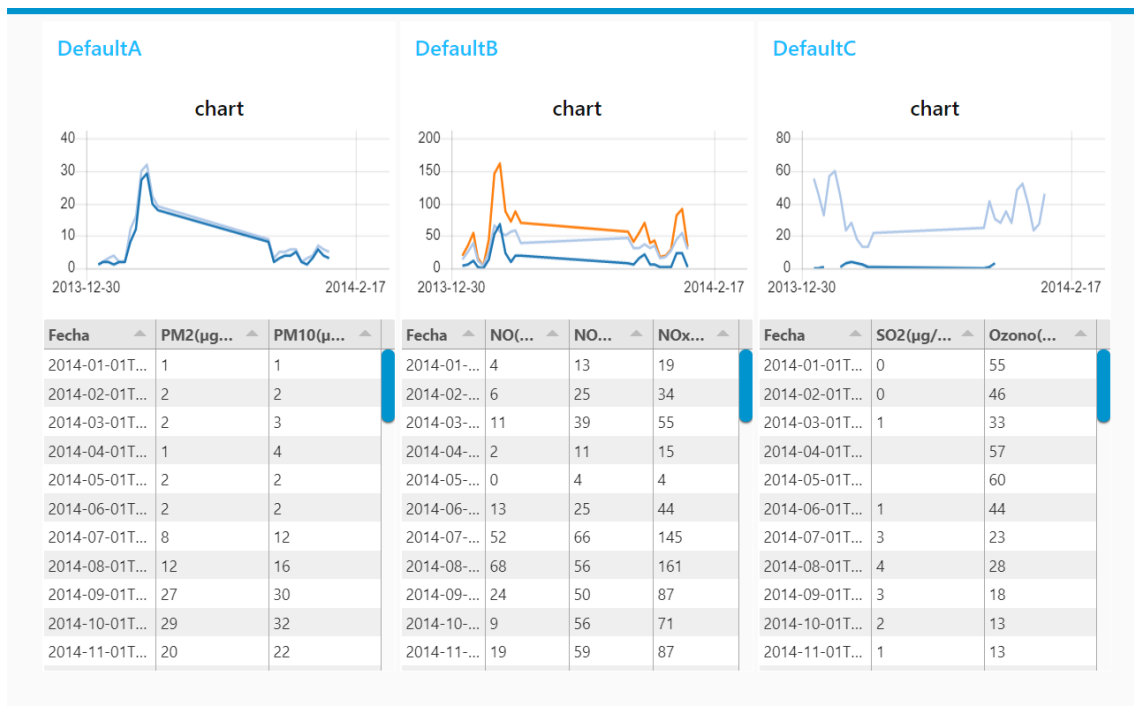


Figura 4.1.10 La página de datos final

```

pi@raspberrypi: ~
8 Feb 23:56:55 - [info] Starting flows
8 Feb 23:56:55 - [info] Started flows
8 Feb 23:56:55 - [info] Server now running at http://127.0.0.1:1880/
8 Feb 23:57:38 - [info] Stopping flows
8 Feb 23:57:38 - [info] Stopped flows
8 Feb 23:57:38 - [info] Starting flows
8 Feb 23:57:39 - [info] Started flows
9 Feb 00:17:35 - [info] Stopping flows
9 Feb 00:17:35 - [info] Stopped flows
9 Feb 00:17:35 - [info] Starting flows
9 Feb 00:17:36 - [info] Started flows
9 Feb 00:18:33 - [info] Stopping flows
9 Feb 00:18:33 - [info] Stopped flows
9 Feb 00:18:33 - [info] Starting flows
9 Feb 00:18:34 - [info] Started flows
9 Feb 00:20:00 - [info] Stopping flows
9 Feb 00:20:00 - [info] Stopped flows
9 Feb 00:20:00 - [info] Starting flows
9 Feb 00:20:01 - [info] Started flows
9 Feb 00:51:39 - [info] Stopping flows
9 Feb 00:51:39 - [info] Stopped flows
9 Feb 00:51:39 - [info] Starting flows
9 Feb 00:51:39 - [info] Started flows

```

Figura 4.1.11 Registro de operación de node-red

Los datos generados con éxito en la figura 4.1.10 muestran el éxito de este proyecto y también demuestran el correcto funcionamiento del esquema de verificación. En la interfaz de visualización de datos se puede observar claramente los datos de texto, así como su gráfico de líneas.



5. Conclusiones

- Durante todo este proceso nos encontramos con varios problemas, por ejemplo, el node-red de varios procesos se bloqueó y no se pudo usar correctamente, la conexión de la API a la plataforma node-red no pudo responder normalmente y la plataforma hardware Raspberry Pi no pudo funcionar normalmente. Al pedir ayuda a los profesores y compañeros de clase para resolver los problemas paso a paso, podemos completar todo el proyecto. La finalización de todo el proyecto se refleja principalmente en el cuarto paso. Podemos observar claramente la confianza de los datos, observar a través de gráficos de líneas y de los gráficos de barras.
- Mediante la realización de la cuarta parte, se han logrado todos los objetivos planteados para este proyecto, también se han descubierto una serie de problemas a lo largo del proceso. Durante el proceso de conexión de la plataforma hardware Raspberry Pi, se ha tenido múltiples errores de instrucción, este proceso mostró confusión en la memoria de instrucciones y mostraba un comportamiento erróneo. En el proceso de programación de código, el conocimiento básico aprendido no es suficiente, por lo que se requieren un aprendizaje y una comprensión más profundos. En el proceso de uso del Raspberry Pi se ratifica que esta plataforma hardware es muy completa, en el proceso de uso de node-red, la construcción simple y conveniente de varios procesos lo convierte en el módulo más importante de todo el proyecto.



6. Referencias

- [1] J. Carter, "Bill Gates," *Smithsonian*. 2005.
- [2] <https://nodered.org/>
- [3] <https://www.valencia.es/dadesobertes/va/>
- [4] <https://ckan.org/>
- [5] https://blog.csdn.net/qq_42936527/article/details/84862249
- [6] <https://randomnerdtutorials.com/getting-started-with-node-red-dashboard/>
- [7] <https://aprendiendoarduino.wordpress.com/2020/03/10/dashboard-en-node-red/>
- [8] <http://m.elecfans.com/article/805638.html>
- [9] <https://www.raspberrypi.org/>
- [10] <https://shumeipai.nxez.com/2014/01/03/advantages-and-disadvantages-of-arduino-raspberry-pi.html>
- [11] <https://flows.nodered.org/node/node-red-dashboard>
- [12] <https://flows.nodered.org/flow/25af5334a79d6c02d6413968a8bff572>
- [13] <https://www.raspberrypi.org/software/>
- [14] <https://angryip.org/download/#linux>
- [15] <https://www.putty.org/>

