

NODE-RED PARA EL DISEÑO DE DASHBOARD

K. Peña, E. Simbaña, MJ. Vizuet

Resumen

Los nodos del panel de Node-RED proporcionan un conjunto integral de componentes de la interfaz de usuario para construir paneles básicos adecuados para Internet de las cosas (IoT), que ofrecen gráficos, indicadores, texto básico, así como controles deslizantes y entradas. Sin embargo, siempre habrá situaciones en las que necesite algo personalizado. El nodo de plantilla es la solución a dichas necesidades.

I.INTRODUCCIÓN

La finalidad principal de esta investigación es identificar y describir de manera general los parámetros de configuración de los principales componentes disponibles en Node-Red para el diseño de Dashboard a través del análisis de información.

A partir de la investigación se busca explicar más a fondo los componentes para su comprensión y posterior implementación en el desarrollo de ejemplos básicos donde se evidencie el uso de los widgets propios del Dashboard.

II.MARCO TEÓRICO

A.¿Qué es un dashboard?

El diseño del dashboard debe considerarse como una cuadrícula. Cada elemento del grupo tiene un ancho: de forma predeterminada, 6 'unidades' (una unidad tiene 48 píxeles de ancho de forma predeterminada con un espacio de 6 píxeles).

Cada widget del grupo también tiene un ancho: de forma predeterminada, 'auto', lo que significa que ocupará el ancho del grupo en el que se encuentra, pero puede establecerlo en un número fijo de unidades.

El algoritmo de diseño del tablero siempre trata de colocar elementos tan altos y a la izquierda como sea posible dentro de su contenedor; esto se aplica a cómo se posicionan los grupos en la página, así como a cómo se colocan los widgets en un grupo. Dado un grupo con un ancho de 6, si agrega seis widgets, cada uno con un ancho de 2, se colocarán en dos filas, tres widgets en cada uno.

Si agrega dos grupos de ancho 6, siempre que la ventana de su navegador sea lo suficientemente ancha, se sentarán uno al lado del otro. Si encoge el navegador, en algún momento el segundo grupo cambiará para estar debajo del primero, en una columna.

Es posible utilizar múltiples grupos si es posible, en lugar de un gran grupo, para que la página pueda cambiar de tamaño dinámicamente en pantallas más pequeñas.[1]

B.Nodos principales del dashboard

Es importante mencionar que en varios de los nodos van a existir los parámetros:

-Group/ grupo: Aquí se va a contener el nodo dentro del dashboard, si se va a implementar distintos nodos para un mismo objetivo, lo mejor sería colocarlos en el mismo grupo.

-Label/ Etiqueta: Se asigna el nombre que va a tener cada uno de los nodos al ser mostrados.

-Tooltip/ Información sobre herramientas: Se escribe una indicación al usuario, esta aparecerá al acercar el cursor al componente.

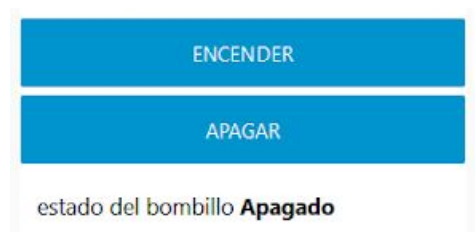


-Name: Se sitúa el nombre que se va a visualizar solo en la terminal. [1]

1)Button:

Agrega un botón a la interfaz del usuario, el cual al ser presionado debe dar como resultado lo que se indique

- Payload: Se coloca lo que se devuelve al dar click en el botón, estos datos pueden ser de tipo string, number, boolean, JSON, buffer, timestamp, flow, global.(Krishnamurthi, R.,2018)



2)Dropdown:

Agrega un cuadro desplegable a la interfaz del usuario.

- Placeholder/Marcador de posición : Se escribe un texto que indique lo que debe seleccionar entre las opciones. También es posible escoger si se puede responder una sola opción o todas.

- Options: En el primer recuadro se agrega el tipo de dato que se va a ingresar, puede ser de tipo number, string o

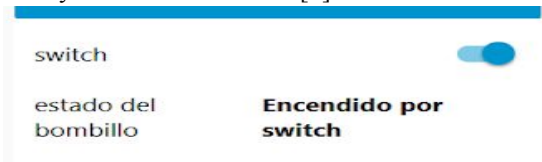
boolean, se añade el número de opciones que se quiera y en label se ingresa la opción.



3)Switch:

Agrega un interruptor a la interfaz del usuario.

- On payload/ en carga útil: Incluye dos botones en su configuración, los cuales van a retornar dos estados en caso de ser presionados, estos estados al igual que los botones pueden ser de tipo entero, string, boolean, y entre otros ya antes mencionados.[3]



4)Slider:

Agrega un control deslizante manipulable a la interfaz de usuario. Una de sus aplicaciones se puede observar conectado a un gauge, el cual es un widget de tipo indicador que se agrega a la interfaz del usuario. Con el deslizador se puede observar cómo se va a mover el indicador del gauge.

- Range: Se define el inicio, fin y paso que va a tener el nodo. Una de las particularidades en node red, es que el nodo slider puede conectarse al texto y devolver valores, transformando los datos de string a entero. (Shahnasser, H. 2017)

- Output: Se puede seleccionar si se quiere que se muestre en la terminal el valor en el que se quedó el deslizador en la interfaz.



5)Numeric:

Agrega un widget de entrada numérico a la interfaz de usuario

- Range/ Rango: Se pone el valor de inicio y fin que será seleccionado por medio de flechas.

Si se quiere que una vez llegado al número final se retorna al inicio marcar Valor de ajuste de max a min y min a max.[2]

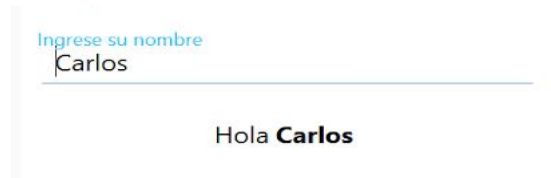


6)Text input:

Agrega un campo de entrada de texto a la interfaz del usuario, el cual puede tener distintos modos.

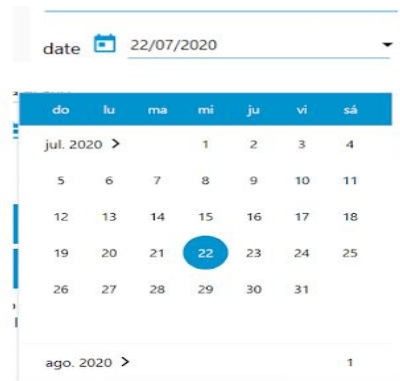
-Label: Se coloca lo que se quiere que el usuario ingrese, y se puede visualizar con el nodo text.

-Mode: Aquí se selecciona el tipo de entrada, la cual puede ser de texto, dirección de email, contraseña, número, teléfono, selector de color, selector de tiempo, selección de semana y mes.



7)Date picker:

Agrega un widget de selector de fecha a la interfaz del usuario.



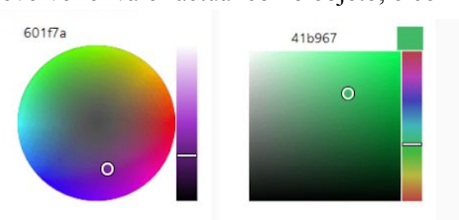
8)Colour picker:

Agrega un selector de color al tablero.

- Format: Se puede escoger entre hx, hx8, hsl, hsv, rgb.

- Send: Aquí se puede seleccionar entre enviar múltiples valores durante la edición o un valor cuando se haya cerrado.

- Payload: Existen dos opciones, la primera es devolver el valor actual como objeto, o como una cadena.[4]



9)Form:

Agrega un formulario a la interfaz del usuario.

-Form elements: Se muestra una serie de elementos a ingresar, se escoge el tipo que puede ser: texto, multilínea, número, e-mail, contraseña, switch o fecha. También se puede seleccionar si es o no requerido el ingreso de este elemento.

-Buttons: Se muestran dos opciones después de haber ingresado lo que se pide en form elements.[5]

nombre1

nombre2 *

nombre3 *

SUBMIT

CANCEL

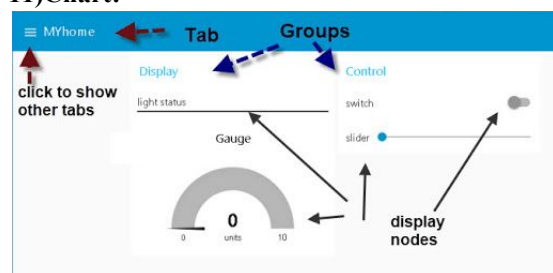
10)Text:

El componente text es una salida de texto que sirve para mostrar un campo de texto no editable en la interfaz.

- Label/ Etiqueta: colocamos lo que queremos que nos aparezca en la pantalla, y en el diseño se puede escoger entre las opciones que se encuentran.[6]

estado del bombillo Apagado

11)Chart:



Tiene modos de línea, barra y gráfico circular. Además, las etiquetas del eje X se pueden personalizar utilizando una cadena de formateador de fecha. El gráfico a su vez utiliza valores enviados a través de msg.payload para su visualización. Los gráficos admiten hasta nueve series diferentes de valores.[7]

Se puede elegir el tipo diseño entre dos opciones, es decir, que este puede ser lineal o radial. (Braun, R, 2017)

-Modo

Opción para activar el modo diferencial. Con esta opción seleccionada, la pista coloreada tiene un punto central desde el cual se muestra el valor. El valor del punto central se puede ajustar. Si el valor central no está definido, el punto central está exactamente entre el mínimo configurado y el máximo.

-Rango: Configure los valores mínimos y máximos esperados y ajuste el color de la línea. Agregue segmentos y configure colores para ellos si es necesario. Tenga en cuenta que las opciones MIN y MAX no se pueden eliminar. Se puede dibujar un pequeño punto con color de segmento en el punto del segmento. Para mostrar el punto por segmento, configure el tamaño para el punto mayor que cero.

-Color de la pista: Se puede configurar el color de la línea de fondo de la pista. Por defecto, los colores del sitio utilizados.

-Marcas de verificación: Los valores mínimos y máximos se pueden mostrar cerca de la pista. Para el modo diferencial, también se muestra el valor central. Con el diseño radial, las marcas de verificación solo se pueden mostrar si el tamaño del diseño es de 3x3 unidades o más.

-Unidad: La unidad se muestra cerca del campo de valor. La unidad puede ser cualquier cadena.

-Decimales: El valor siempre se redondea según los decimales configurados. El valor predeterminado es cero, por lo que el valor se presenta como entero.

12)Audio out:



Un widget que le permitirá reproducir audio (wav o mp3) o enviar texto a voz (TTS) al cliente. Este nodo reproducirá audio de texto a voz de los mensajes enviados a través del objeto msg.payload. El audio se reproducirá incluso cuando el tablero esté minimizado.[8]

-Group/Grupo: Es el espacio en el que se va a contener el nodo dentro del dashboard.

-TTS Voice: TTS estas siglas significan text to speech, es decir, texto para hablar.

Esta opción envía una cadena de texto al motor de texto a voz de Google para que se convierta en audio de voz. El servicio TTS de Google tiene un límite de longitud de cadena de entrada de 200 caracteres. Si el texto a traducir tiene más de 200 caracteres, se dividirá de manera inteligente en segmentos y la salida consistirá en una matriz de URL que se vinculan a archivos de audio secuencias que codifican cada segmento.

-Name/Nombre: Será el identificador para reconocer a ese widget.

13)Notification:

La notificación, que crea alertas para el usuario, puede ser una ventana emergente de pan tostado o un cuadro de alerta descartable. La alerta puede estar dirigida a un solo usuario.[9] Para que en el Dashboard quede constancia de todas las acciones realizadas, mediante las notificaciones de Node-Red, salen unos avisos en el lado derecho arriba sobre las acciones pulsadas o realizadas en cada momento. Es decir que este nodo muestra notificaciones como ventanas

emergentes en la interfaz del tablero. msg.payload será el mensaje emergente, msg.topic será el título y msg.highlight establecerá el color de resaltado del borde opcional.

-Layout/Diseño: Se refiere a la parte en la que se presentará la notificación.

-Timeout/Tiempo de salida: Tiempo que será mostrado el mensaje en el dashboard.

-Border/Borde: Hace referencia al borde del recuadro de notificación.

-Topic/Tema: se selecciona el tema que puede ser opcional.

-Name: Será el identificador con el que se reconocerá.

14)Template:

El nodo de plantilla permite al usuario especificar y crear sus propios widgets dentro del marco utilizando HTML, Javascript. Este es un widget Angular.js. También puede usar esto para anular los estilos CSS incorporados. Hay dos nodos de plantilla en nodo-rojo. Son el nodo de plantilla HTML y el nodo de plantilla de panel. El nodo acepta entradas en el objeto msg.payload. El campo llamado propiedad establece el nombre del objeto de salida del nodo. Normalmente esto es msg.payload pero puede cambiarlo a lo que quiera. por ejemplo, msg.options. Incluso puede enviarlo directamente a un flujo o variable global.

Si la plantilla contiene datos JSON o datos YAML, lo cual es común cuando se usa la plantilla como un nodo de configuración, puede generar los datos como un objeto JSON analizado o un objeto YAML analizado, es decir, JavaScript.

-Template type: se refiere a en qué grupo se encuentra este widget.

-Group/ grupo: Aquí se va a contener el nodo dentro del dashboard, si se va a implementar distintos nodos para un mismo objetivo.

-Size: Dirá la medida que puede ser elegida por el usuario, caso contrario permanecerá con la medida autodesignada.

15)Gauge:

Este nodo muestra un widget de tipo de indicador en el tablero. Formateará los números pasados a través de msg.payload en un widget de estilo de indicador. El medidor contiene diferentes sectores y la aguja se moverá a diferentes sectores según el Rango y el valor pasado al widget a través de msg.payload. (Toc, S. I., & Korodi, A., 2018).

-Group/ grupo: Aquí se va a contener el nodo dentro del dashboard, si se va a implementar distintos nodos para un mismo objetivo, lo mejor sería colocarlos en el mismo grupo.

-Size: Se selecciona el tamaño del gráfico.

-Type/tipo: Selecciona el tipo de gráfico a mostrar. Tiene 4 modos: estándar (calibre simple), rosquilla (completa 360 °), brújula y onda. También puede especificar la gama de colores de los medidores estándar y de rosquilla.

-Label/ Etiqueta: Se asigna el nombre que va a tener cada uno de los nodos al ser mostrados en la pantalla.

-Rango: Configura los valores mínimos y máximos esperados.

-Unidad: se muestra cerca del campo de valor. La unidad puede ser cualquier cadena

-Sectors: Pueden variar conforme datos numéricos.

-Colour gradient: En esta opción podemos modificar el color a visualizar en el tipo de gráfico seleccionado.

III.EXPLICACIÓN DE CÓDIGO FUENTE

C.Explicación código "Audio Output"

Como primer punto usamos el nodo inject para inicializar nuestra variable msg a usar, conectado a este irá nuestro nodo input en el cual podremos ingresar palabras para que nuestro programa lo procese, anterior a este usamos una conexión entre un nodo delay y un nodo function con los cuales podremos generar un retraso estos permitirán que nuestra variable ingresada en el nodo input se limpiará mediante una codificación en JS

```
text=msg.payload;
if(text !== ""){
```



```

    msg.payload = "";
    return msg;
}
y dé espacio a poder escribir otra cadena de texto,
consecuente a esto conectamos un nuevo nodo function el
cual cambiará la palabra antigua por la nueva
text=msg.payload;
if(text !== ""){
    msg.payload = text;
    return msg;
}
un nodo de salida de audio estará conectado al final de este
para poder escuchar la palabra impuesta, con el método base
establecido empleamos nodos button para que mediante un
link se conectan al nodo input y de esta manera tener
palabras preestablecidas alojadas dentro del botón por
último conectamos un nodo notification el cual nos permite
a la par que escuchamos la palabra visualizarla como si de
un mensaje se tratara

```

D. Explicación código “Char / Gauge / Notification”

<p>Mediante un nodo inject inicializamos nuestra variable msg que node red usa por defecto al mismo tiempo que le incluimos un intervalo de 2 segundo este nodo lo conectaremos a un nodo function en el cual crearemos nuestro número aleatorio <p/>

```

ran =parseInt(Math.random() *10);
msg.topic = "Random1";
msg.payload = ran;
return msg;

```

el cual nos servirá como base para nuestro nodo char y nuestros nodos gauge los cuales genera gráficos en un espacio x, y con un intervalo de 2 segundos establecidos anteriormente el mismo método lo usamos para generar un número aleatorio que lo muestre un nodo notification

E. Explicación código “Forms”

Usamos directamente el nodo forms el cual irá conectado a un nodo debug el cual revisa errores el nodo forms es configurable y en este podemos ingresar datos por teclado podemos decidir cuántos valores ingresar, si estos valores son obligatorios o no para la persona que ejecuten el programa y descargando un nodo cloudant mediante escribir npm install node-red-node-cf-cloudant en la consola de node-Js podremos usar una base de datos que la generemos con anterioridad por ejemplo en IBM cloud.

F. Explicación código “Template”

Para el uso el nodo template basta con conectarlo a un nodo inject una vez conectado el nodo template nos otorga de un espacio programable con estructura HTML en el cual podemos programar de forma libre el espacio y llenarlo de nuestra información en nuestro caso se genera una plantilla basado en estructura JS y CSS llamando a los lenguajes mediante las etiquetas <script> y <style> respectivamente

IV. METODOLOGÍA

Para el desarrollo de esta investigación se utilizó el método de revisión bibliográfica puesto que nos enfocamos en el análisis de documentos cuyo contenido tenía relación con el tema abordado en este estudio, esto no permitió profundizar los conceptos y definiciones necesarios para el desarrollo del código de los programas. La investigación se basó en herramientas tecnológicas como: Node.js/Node-Red los

cuales nos permitieron programar y ejecutar el código, el navegador en el cual ejecutamos los programas, la herramienta Github .

V. APORTACIONES

Para la implementación de los programas solicitados se agrego el uso de nodos diferentes como botones, links y funciones los cuales permitieron generar una interfaz más interactiva con el usuario de la misma manera se propuso una plantilla para llenar el nodo template basado en una estructura de etiquetado HTML con la ayuda de etiquetas Style y Script para mostrar una plantilla agradable

VI. CONCLUSIONES

Node-Red brinda una interfaz gráfica amigable con el usuario, para el desarrollo y diseño de Dashboard, ya que cuenta con distintos widgets los cuales han sido definidos a lo largo de la investigación conforme lo planteado, a su vez se revisó los parámetros de configuración de cada uno.

La herramienta de Node-red es muy potente ya que sirve para comunicar hardware y servicios de una forma muy rápida y sencilla. Además simplifica de gran manera la tarea de programar gracias a la programación visual.

Se cumplió nuestro objetivo acerca de desarrollar ejemplos básicos donde se evidencie el uso del dashboard de Node-red , esto debido a las investigaciones realizadas referentes al uso de la herramienta de programación, su funcionamiento y características.

VII. RECOMENDACIONES

Se recomienda que antes de intentar generar una interfaz el usuario cuente con la instalación de todos los nodos posibles si va usar una base de datos instalar el nodo cloudant o si va a usar una salida de audio con voz en español descargar el nodo play audio.

También se recomienda conocer los parámetros de configuración de cada nodo a fin de evitar fallas en el desarrollo de los programas.

VII. BIBLIOGRAFÍA

- [1]García Jiménez, S. (2019). Desarrollo de paneles de control para redes IoT basados en NodeRed (Doctoral dissertation).
- [2] OpenJS Foundation. (Julio de 2020). *Node-RED*. Obtenido de <https://flows.nodered.org/node/node-red-dashboard>
- [3] (20 de 11 de 2018). Obtenido de Aprendiendo Arduino: <https://aprendiendoarduino.wordpress.com/2018/11/20/node-red>
- [4] Sicari, S., Rizzardi, A., & Coen-Porisini, A. (2019). Smart transport and logistics: A Node-RED implementation. Internet Technology Letters, 2(2), e88.

[5] García Jiménez, S. (2019). Desarrollo de paneles de control para redes IoT basados en NodeRed (Doctoral dissertation).

[6] Lutvi, L. (2019). PERANCANGAN DAN IMPLEMENTASI DASHBOARD BERBASIS NODE-RED UNTUK MONITORING TEGANGAN, ARUS, DAYA, DAN SUHU DENGAN MEMANFAATKAN KOMUNIKASI ARDUINO DAN LORA. Program Studi Teknik Elektro.

[7] Dashboard Nodes. (2018). Hewlett Packard Enterprise.
<https://techlibrary.hpe.com/docs/otlink-wo/Dashboard-Nodes.html#DashboardNodes-ChartNode>

[8] *Los nodos centrales*. (2020). Node-RED.
<https://nodered.org/docs/user-guide/nodes#template>

[9]Notificaciones Node-Red. (2016). HEIMDALL-EYE.
<http://heimdall-eye.es/notificaciones-node-red/>