Embedded Developer Coding Interview - CINTELINK

Por María Eugenia Bonillo

Solución a transacciones de combustibles

Se propone el siguiente sistema a implementar. Fig_1

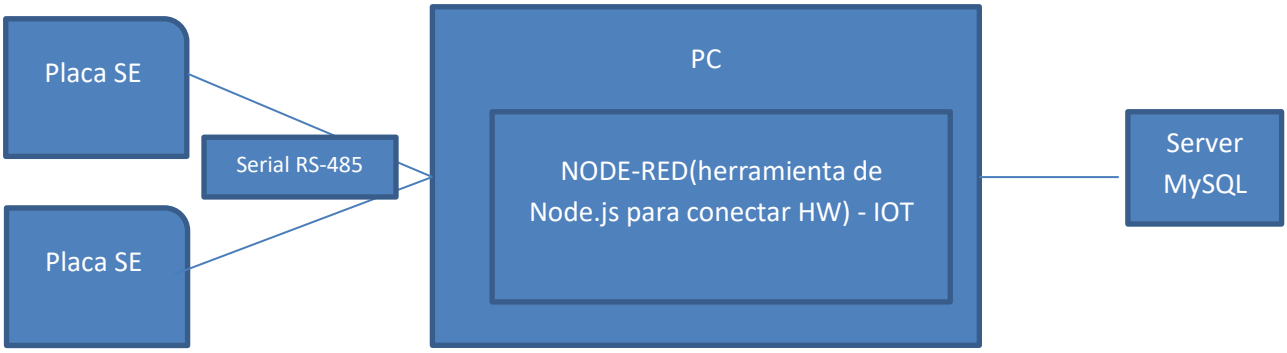


Fig_1

La PC recibe desde las placas electrónicas, tramas de 34 bytes donde es procesada en tiempo real por Node-Red y luego guardada en una base de datos por ejemplo MySQL.

Node-Red puede ser instalado en una raspberry pi, se conecta a servidores de Base de Datos, MQTT. Puede mostrar datos procesados con un dashboar.

Desarrollo:

a) Dashboard: (Fig_2)

- Parte Izquierda de se puede observar un pulsador que simula la trama de 34 bytes.
- Parte derecha muestra los datos ya procesados y listo para guardar en base de datos.

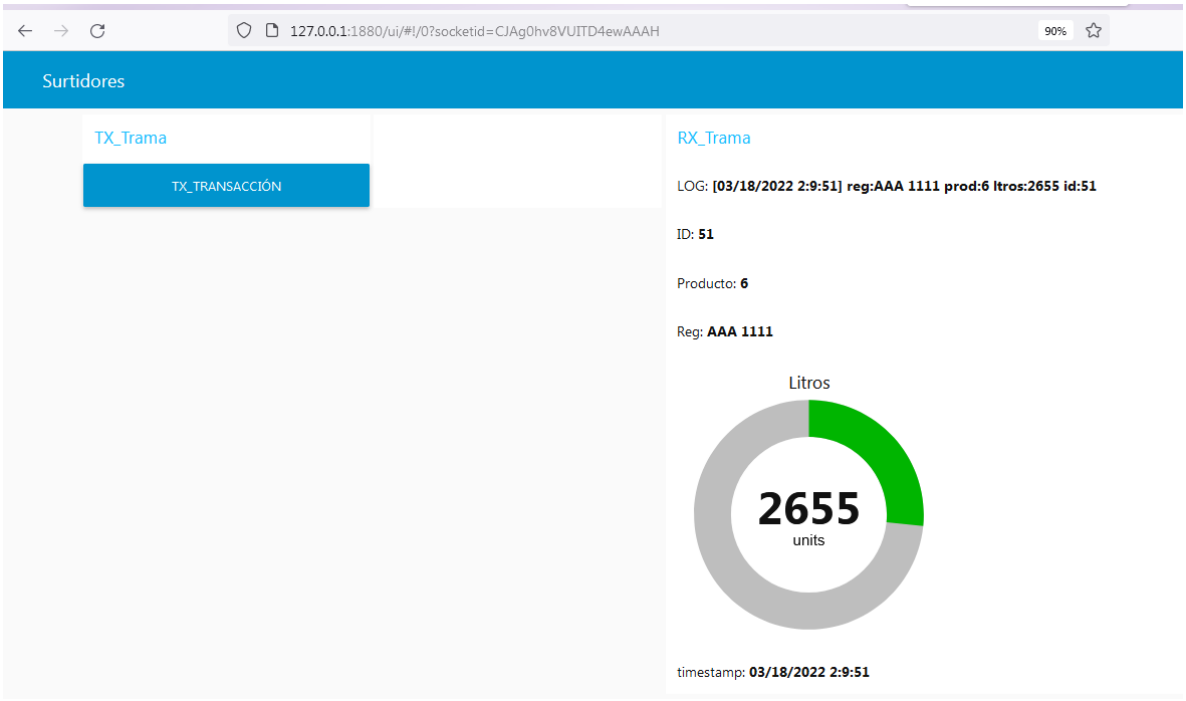
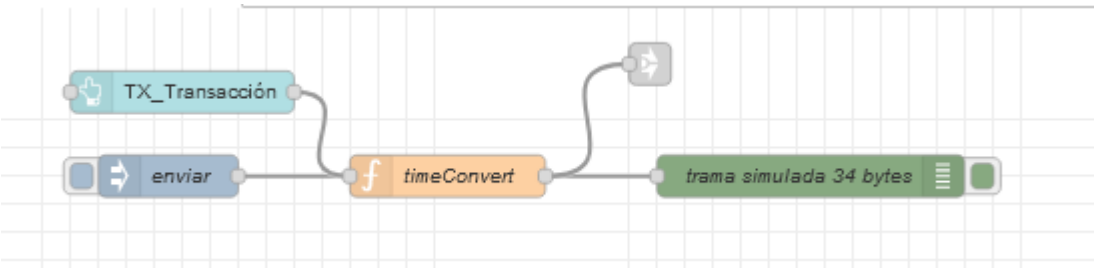


Fig_2

b) Generación de trama de 34 Bytes_ TX:

Mediante un pulsador se genera la trama en tiempo real con la hora exacta de la pc más los datos (id, reg, ltros, prod) para ser enviada al Receptor de datos.

- Trama simulada por node-red



La función ***timeConvert*** tiene el siguiente código:

```
if ( !msg.timestamp ) msg.timestamp = Math.round(+new Date());

var dt = new Date(msg.timestamp);
var mes_0=0;

/**Mes**/
if(mes_0.payload == isNaN)
        {
         mes_0={payload:Number((dt.getMonth() + 1).toString()[1])+48};
        }
else
        {
          mes_0={payload:48};
        }
var mes_1={payload:Number((dt.getMonth() + 1).toString()[0])+48};

/**Dia**/
var dia_0={payload:Number((dt.getDate()).toString()[0])+48};
var dia_1={payload:Number((dt.getDate()).toString()[1])+48};


/**Año**/
var ano_0={payload:Number(dt.getFullYear().toString()[0])+48};
var ano_1={payload:Number(dt.getFullYear().toString()[1])+48};
var ano_2={payload:Number(dt.getFullYear().toString()[2])+48};
var ano_3={payload:Number(dt.getFullYear().toString()[3])+48};

/**Hora**/
var hora_0={payload:Number(dt.getHours().toString()[0])+48};
var hora_1={payload:Number(dt.getHours().toString()[1])+48};

/**minuto**/
var min_0={payload:Number(dt.getMinutes().toString()[0])+48};
var min_1={payload:Number(dt.getMinutes().toString()[1])+48};

/**seg**/
var seg_0={payload:Number(dt.getSeconds().toString()[0])+48} ;
var seg_1={payload:Number(dt.getSeconds().toString()[1])+48};

//**Mililiters**//
var mili_0={payload:Number(dt.getMilliseconds().toString()[0])+48};
var mili_1={payload:Number(dt.getMilliseconds().toString()[2])+48};
var mili_2={payload:Number(dt.getMilliseconds().toString()[1])+48};
var mili_3={payload:Number(dt.getMilliseconds().toString()[1])+48};
```

```
/**transaccion_ID**/
var id_0={payload:Number(dt.getSeconds().toString()[0])+48};
var id_1={payload:Number(dt.getSeconds().toString()[1])+48};


var vector =
Buffer.from([mes_0.payload,mes_1.payload,'47',dia_0.payload,dia_1.payload,'47',a
no_0.payload,ano_1.payload,ano_2.payload,ano_3.payload,'32',hora_0.payload,ho
ra_1.payload,'58',min_0.payload,min_1.payload,'58',seg_0.payload,seg_1.payload,
'65','65','65','32','49','49','49','49','54',mili_0.payload,mili_1.payload,mili_2.payload,m
ili_3.payload,id_0.payload,id_1.payload]);  // aquí se genera el Buffer de 34 Bytes

var TramaOut = {payload: vector};
return TramaOut ;
////////////////////////////////////////////////////////////////////////////
```

En la siguiente figura muestra el Buffer de datos de 34 Bytes en la ventana debug:

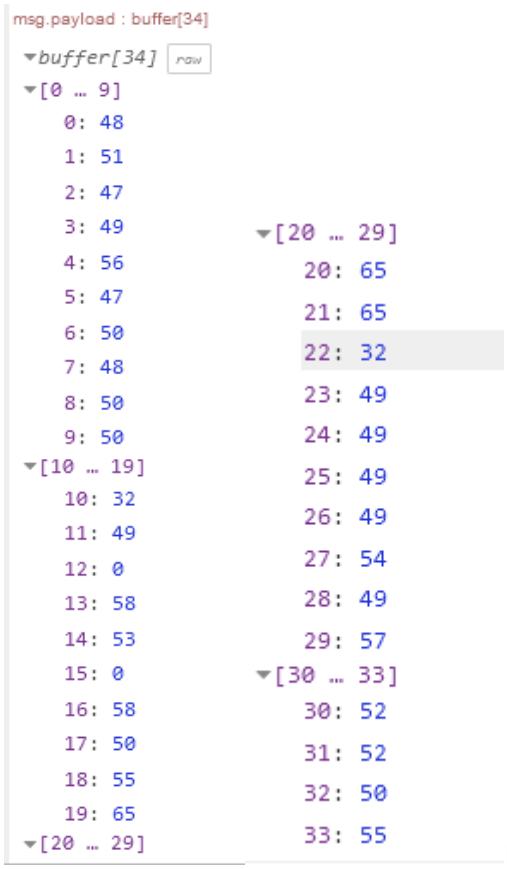


Fig_3

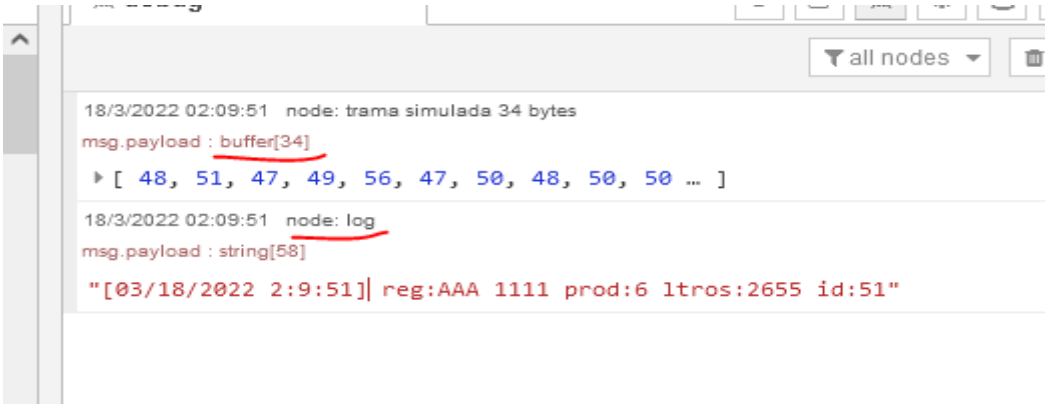


Fig_4: Ventana Debug

c) Procesamiento de Trama

Los datos recibidos se los toma como un vector de 34 valores. Este vector es procesado y convertido en formato de datos String o int que luego será enviado a la base de datos, para un futuro análisis.

En la FIg_4 muestra el log de la trama

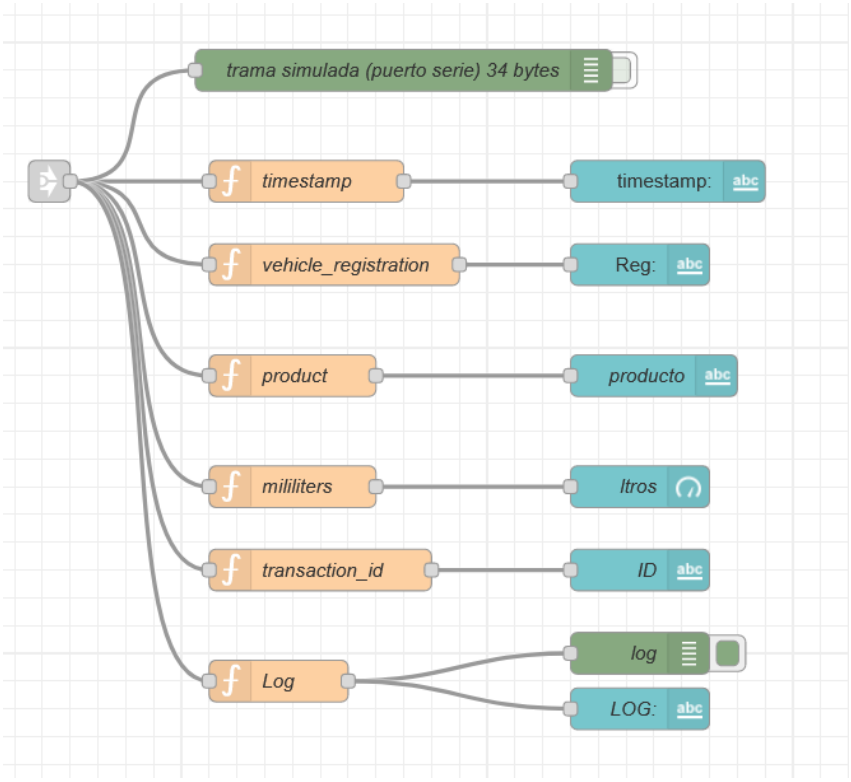


Fig_5: RX_funciones

Cada función de la Fig_5 extrae los datos de cada campo (timestamp, id, reg, product, ltros.) para ser mostrado y guardado en pantalla y base de datos.

- **Código timestamp: primeros 19 bytes de datos**

```
var mes= {payload:String.fromCharCode(msg.payload[0])+String.fromCharCode(msg.payload[1])};

var dia= {payload:String.fromCharCode(msg.payload[3])+String.fromCharCode(msg.payload[4])};

var año=
{payload:String.fromCharCode(msg.payload[6])+String.fromCharCode(msg.payload[7])+String.from
CharCode(msg.payload[8])+String.fromCharCode(msg.payload[9])}

var hora=
{payload:String.fromCharCode(msg.payload[11])+String.fromCharCode(msg.payload[12])};

var min=
{payload:String.fromCharCode(msg.payload[14])+String.fromCharCode(msg.payload[15])};

var seg=
{payload:String.fromCharCode(msg.payload[17])+String.fromCharCode(msg.payload[18])};

var timesTamp = {payload:mes.payload + "/"+ dia.payload + "/"+ año.payload+ " "+ hora.payload+
":"+min.payload+":"+seg.payload};

return timesTamp;
```

- **Código vehicle_registration**

```
var vehicule_parte1=
{payload:String.fromCharCode(msg.payload[19])+String.fromCharCode(msg.payload[20])
+String.fromCharCode(msg.payload[21])+" "};

var vehicule_parte2=
{payload:String.fromCharCode(msg.payload[23])+String.fromCharCode(msg.payload[24])
+String.fromCharCode(msg.payload[25])+String.fromCharCode(msg.payload[26])};

var vehicle_reg = {payload:vehicule_parte1.payload+vehicule_parte2.payload};

return vehicle_reg;
```

- **Código product**

```
var product= {payload:String.fromCharCode(msg.payload[27])};

return product;
```

- **Código mililiters**

```
var mililiters=
{payload:String.fromCharCode(msg.payload[28])+String.fromCharCode(msg.payload[29])
+String.fromCharCode(msg.payload[30])+String.fromCharCode(msg.payload[31])};

return mililiters;
```

- **Código transaction_id**

```
var transaction_id=
{payload:String.fromCharCode(msg.payload[32])+String.fromCharCode(msg.payload[33])}

return  transaction_id;
```

- **Código Log: es la unión de todos los datos recibidos en formato String**

d) Base de datos: aquí se plantea la solución pero no se llegó a implementar.

Cada transacción se guardará con un INSERT a la base de datos ejemplo:

> ***INSERT into tabla_surtidores SET id=?, Id_vehiculo=dato1, producto=dato2,etc***

Para mostrar los datos de forma ordenada de las últimas 100 transacciones, se utiliza las query a la base de datos. Por ejemplo:

*SELECT * FROM tabla_surtidores ORDER BY fecha_y_hora DESC LIMIT 100;*

- *con el asterisco muestra todos los campos de la tabla.*

ID	Id_vehiculo	producto	reg	litros	Fecha_y_hora
1	111	3	AAA 1111	1999	18/03/2022 18:23:07
2	123	5	AAB 4444	3000	18/03/2022 19:43:07

Fig_6: tabla en base de datos

Para más información se deja la url de la herramienta utilizada: **https://nodered.org/**

Video de la prueba técnica

https://www.youtube.com/watch?v=623EdDhWQwU

https://www.youtube.com/watch?v=aw5D6NTiyO8