

SmartPlace

Proyecto de sitios inteligentes

con *ESP32 STEAMakers* + LoRa + 

Por: *Pedro Ruiz Fernández*

Basado en [LoRa y LoRaWAN con Arduinoblocks + ESP32 STEAMakers \(por Juanjo López\)](#)

Sumario

Objetivo.....2

Diseño 3D.....2

Componentes electrónicos.....3

Conexionado.....4

 Conexionado de placa maestra.....4

 Conexionado de cada uno de los nodos.....4

Programas.....5

 Programas par estación base o placa maestra.....5

 Programa con arduinoblocks para placa maestra con graficado en tiempo real y toma de datos (datalogger).....5

 Programa con arduinoblocks para placa maestra con envío de datos por puerto serie y grabación en sd.....6

 Programa con arduinoblocks para nodos.....6

Fuentes.....6

Autoría y Licencia.....6

Objetivo

Se trata de monitorizar a través de sensores cualquier centro, interconectando una serie de placas ESP32 STEAMakers a través de tecnología LoRa, con objeto de mantener bajo control y estudio datos como luminosidad, temperatura, humedad, presencia, ruido, etc.

Diseño 3D

Para el [diseño de los nodos \(estaciones\)](#) de los diferentes espacios, nos hemos basado en el diseño de un [alojamiento para arduino uno](#) (compatible con la forma de la ESP32 STEAMakers), al que le hemos añadido una rejilla para el atornillado de sensores y actuadores. El alojamiento sin rejilla es el que usaremos para la estación base (maestra), ya que no tiene sensores.

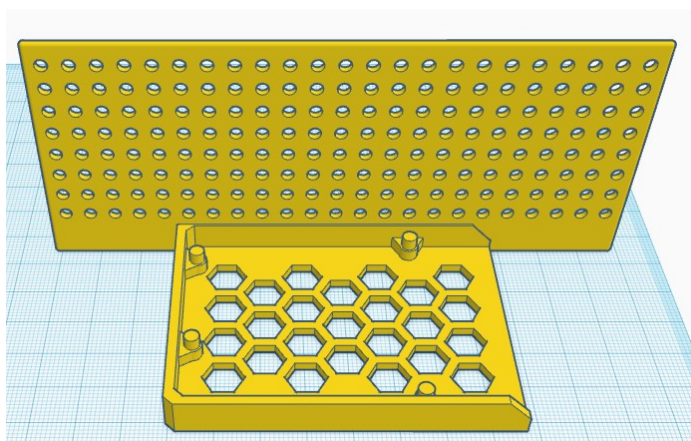


Figura 2: Diseño 3D nodo

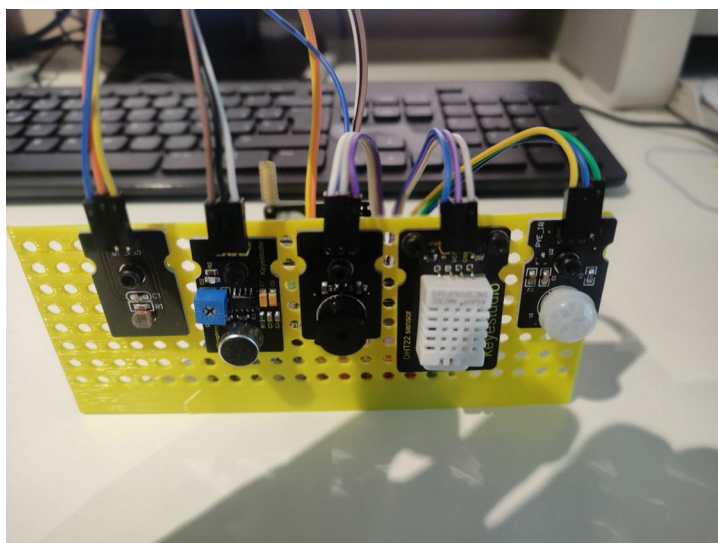


Figura 1: Foto de nodo

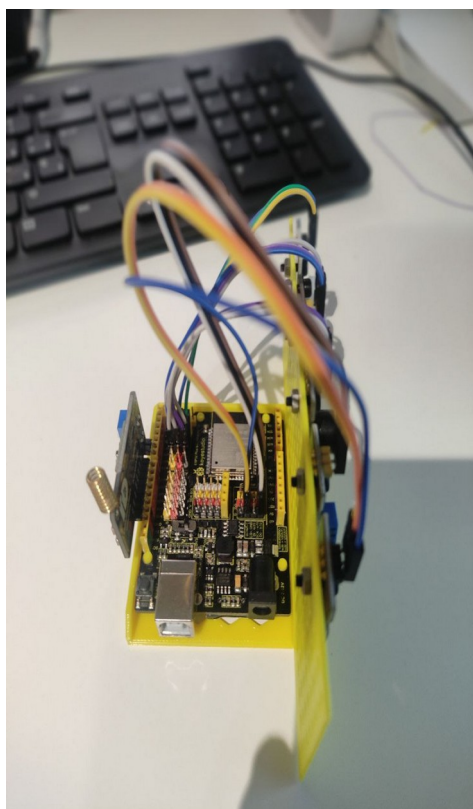


Figura 3: Foto de nodo perfil

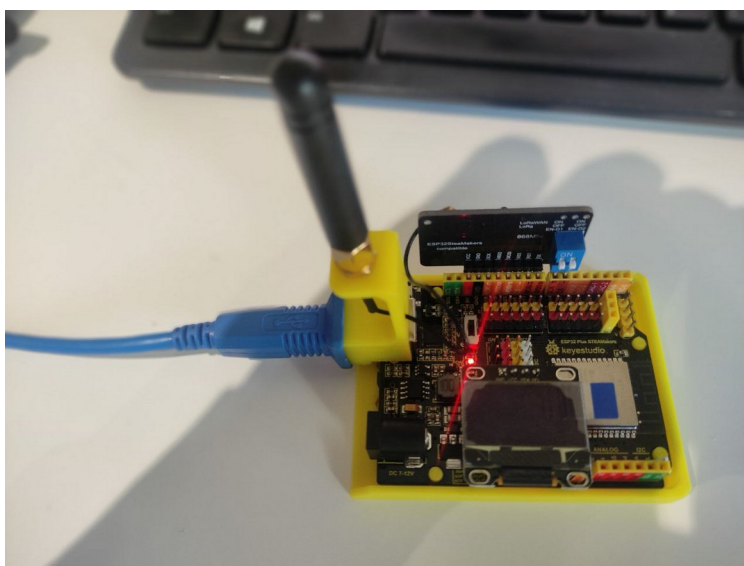
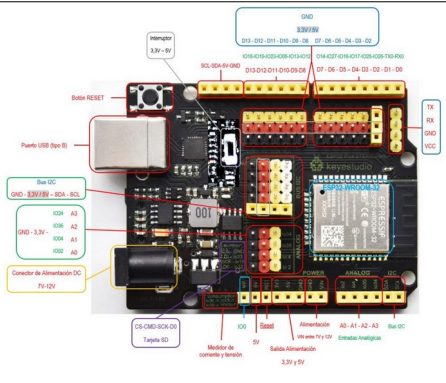

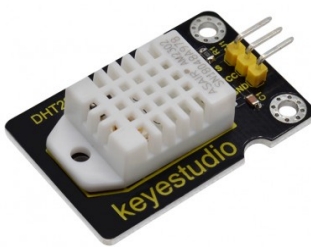

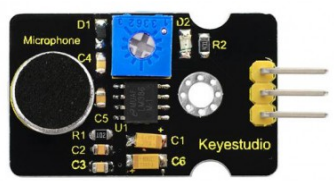


Figura 4: Estación base

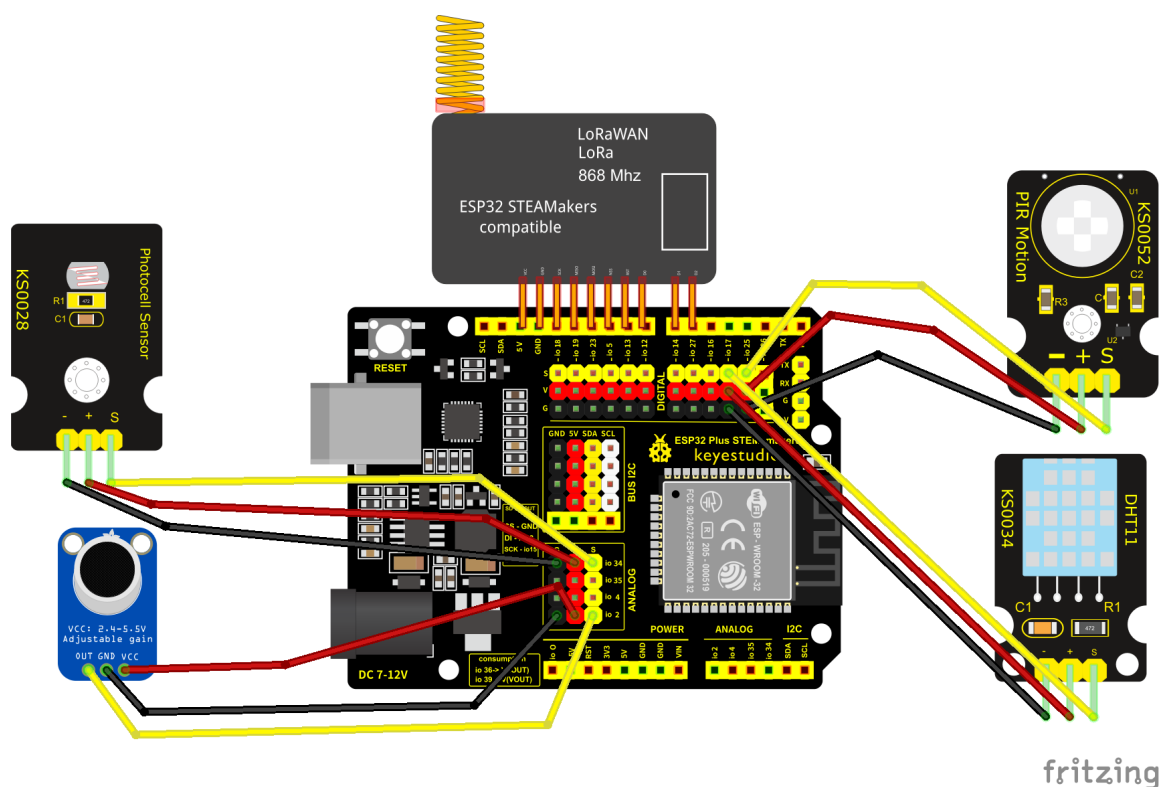
Componentes electrónicos

Elemento	Imagen	Descripción
Placa ESP32 STEAMakers		<p>Esp32 fabricada por keyestudio, con factor de forma arduino y con aditamentos como pinout macho, microsd, sensores integrados de energía, programable con arduinoblocks y con una gran documentación disponible. Se usarán una en como concentradora o maestra, y tantas como estaciones de datos quieras poner.</p>
Sensor PIR		<p>Sensor digital que detecta movimiento o no. Por tanto da valores 0 o 1, true o false.</p>
DHT 22		<p>Sensor digital que nos da valores de humedad y temperatura.</p>
LDR		<p>Sensor analógico que ofrece valores de luminosidad.</p>
Micrófono		<p>Sensor analógico que mide la cantidad de sonido ambiente</p>

Como propuesta de sensores son los anteriores, pero se pueden usar los que deseemos, lógicamente cambiando posteriormente los programas.

Conexionado de placa maestra

En los nodos (placas esclavas) vamos a tener un dispositivo LoRa, una LDR (fotoresistencia) conectada a pin analógico IO34, un micrófono conectado a pin analógico IO2, un sensor PIR (presencia) conectado a pin digital IO25 y un sensor DHT22 (humedad y temperatura) conectado a pin digital IO17.



Dispositivo	Pin
LDR	IO34
Micrófono	IO2
PIR	IO25
DHT22	IO17

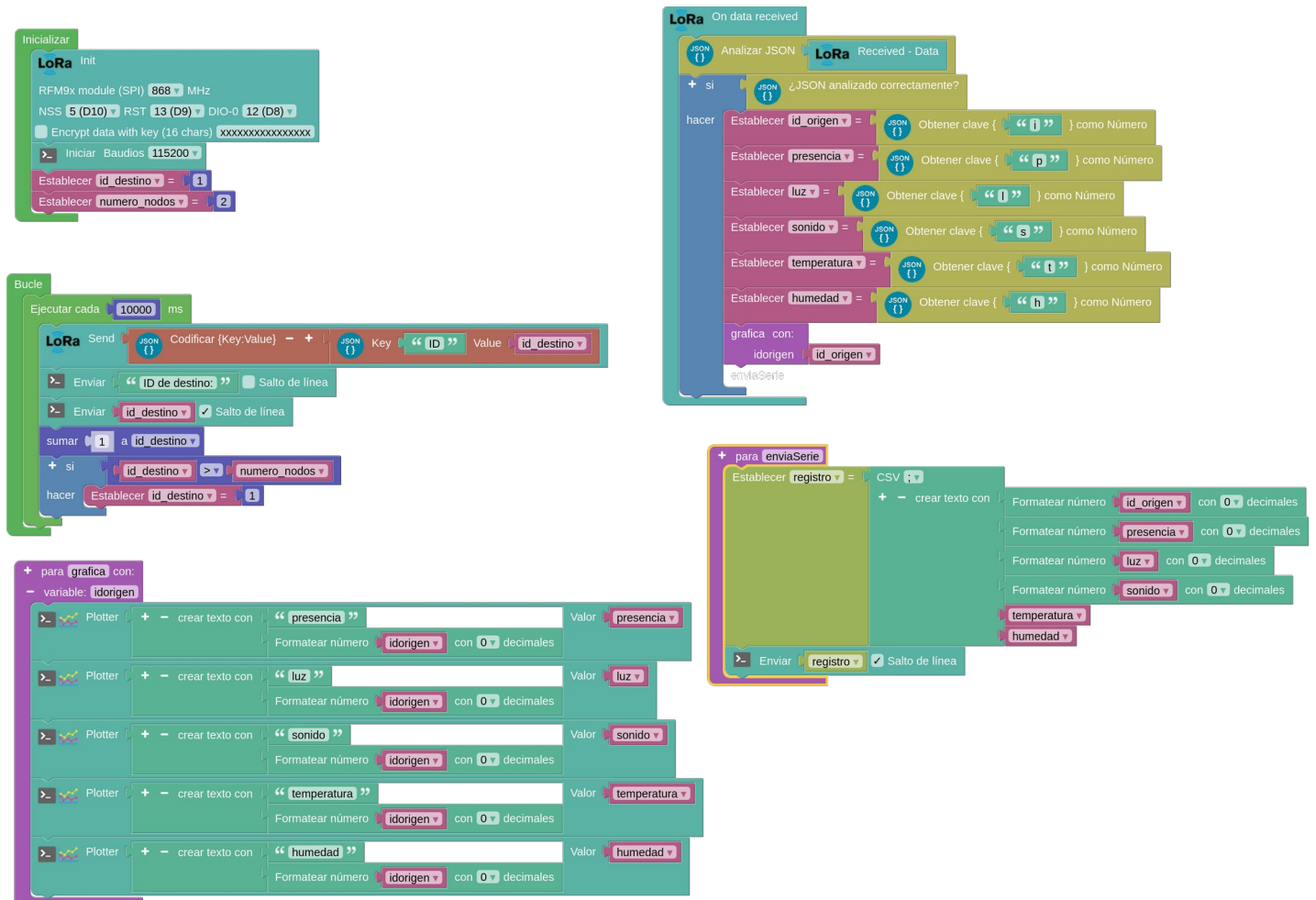
Programas

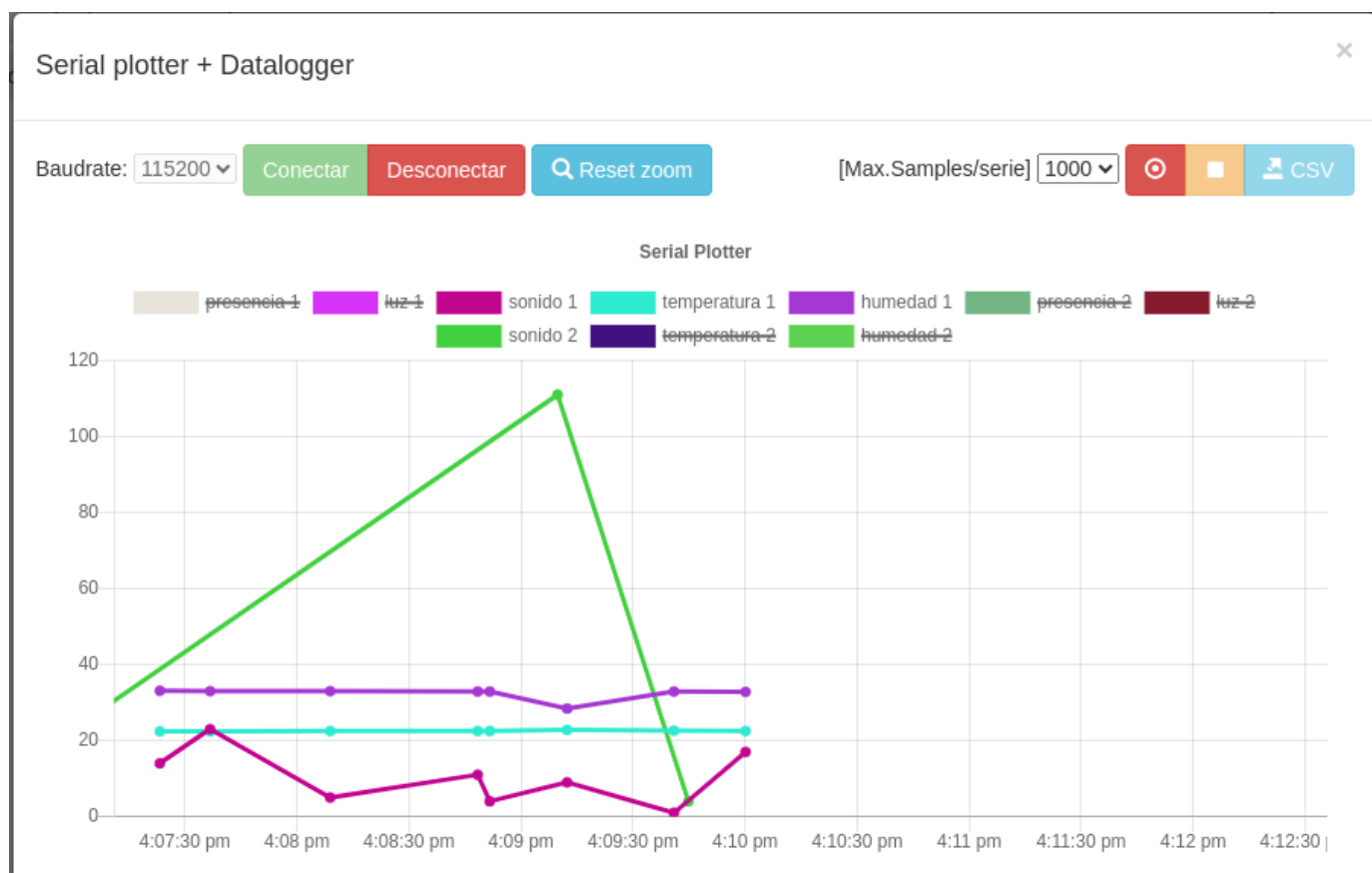
Programas par estación base o placa maestra

Programa con arduinoblocks para placa maestra con graficado en tiempo real y toma de datos (datalogger)

Este programa manda cada 10 segundos en formato JSON una identificación numérica entera, que representa el número de nodo a pedir información, empezando por el 1 e incrementándose de uno en uno. Los nodos están programados para que sólo pueda responder al que hemos mandado el identificador numérico, y responden en formato JSON con su identificador (id), si tiene presencia o no, con el nivel de luz, nivel de sonido, temperatura y humedad. Esos datos los analizamos, los asignamos a variables y los graficamos en tiempo real por puerto serie con la función “grafica” que usa la herramienta de arduinoblocks “Serial plotter”. Además podemos realizar el guardado de datos en PC con la herramienta datalogger de “Serial plotter”.

Para más señas, en [este vídeo](#) se muestra como realizar un datalogger con graficado y guardado de datos en tiempo real en arduinoblocks, similar al que estamos trabajando.





En el graficado en tiempo real aparece una gráfica por cada variable medida de cada nodo. Además podemos visualizarla o no seleccionándola con el ratón en la leyenda de la misma.

Programa con arduinoblocks para placa maestra con envío de datos por puerto serie y grabación en sd

El funcionamiento del programa es similar al anterior, se pide a los nodos información cada 10 segundos, y si el nodo coincide con el que se ha pedido información nos devuelve los datos. La diferencia es que ahora a esos datos les añadimos fecha y hora cogida de servidor NTP de internet, y toda la información la enviamos por puerto serie y a tarjeta microSD. La información por puerto serie puede ser grabada en el ordenador a través de programas de visualización del puerto (en *GNU/Linux* puede ser [GTKterm](#), y en *Windows* [Putty](#)), e incluso graficada en tiempo real con programas como [KST](#), o bien a través de hoja de cálculo. Este programa necesita de conexión a internet por parte de la placa. En el programa hemos desactivado la función “grafica”.

En estos vídeos se explica el proceso de tratamiento de datos recepcionados por puerto serie, en este caso para el proyecto cansat, pero es similar en nuestro caso:

- [Cansat recepción de datos en tierra](#)
- [Cansat graficado en tiempo real](#)
- [Cansat tratamiento de datos con hoja de cálculo](#)



Consola serie

Baudrate: 115200

Conectar

Desconectar

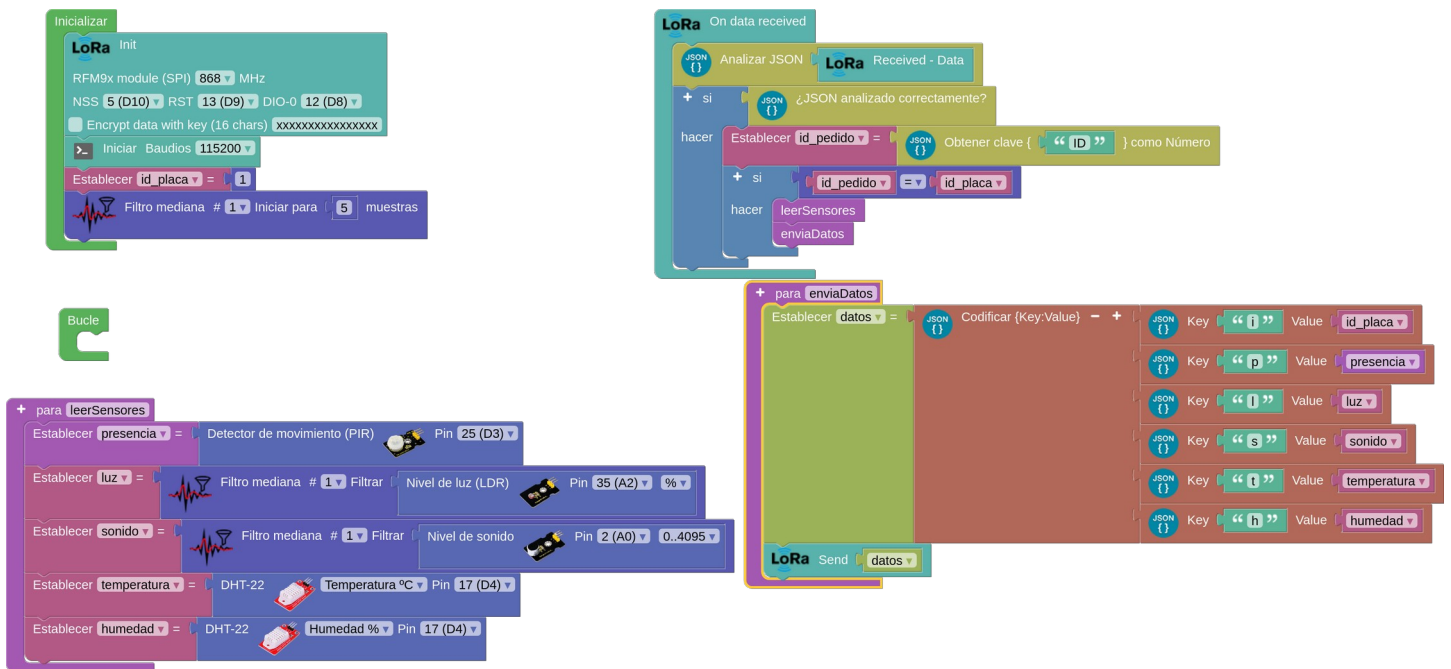
Limpiar

Enviar

```
1;2023-12-29;09:45:12; 0;23;369;19.50;37.90;  
1;2023-12-29;09:45:32; 0;23; 6;19.50;37.40;  
1;2023-12-29;09:45:52; 0;24;64;19.50;37.90;  
1;2023-12-29;09:46:12; 0;25;86;19.50;37.50;  
1;2023-12-29;09:46:32; 0;25;32;19.50;37.50;
```

Programa con arduinoblocks para nodos

En los nodos (o dispositivo esclavos), lo primero es asignarle un identificador de placa (en establecer id_placa), desde el 1 al n.º de placas que tengamos. El programa recibe un identificador de la placa maestra y si coincide con el suyo devuelve a la placa maestra los datos de los sensores en formato JSON. Hemos de reseñar que se ha usado en la lectura de la luz y del sonido el filtro mediana, que da una mediana de los últimos valores medidos (en nuestro caso 5) para evitar fluctuaciones puntuales de los valores. Ver “[Filtro mediana en Arduinoblocks](#)”.



Fuentes

- [LoRa y LoRaWAN con Arduinoblocks + ESP32 STEAMakers \(por Juanjo López\)](#)
- [Filtro mediana en Arduinoblocks](#)

Autoría y Licencia

SmartPlace by Pedro Ruiz Fernández is licensed under Attribution-ShareAlike 4.0 International. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/>