

# Sistema IoT para el análisis de calidad de aguas

Enrique Soto Campos  
Departamento Tecnología Electrónica  
Instituto de Electrónica Aplicada  
Universidad de Vigo  
Vigo, España  
esotoc@uvigo.es

Rafael Verdugo Matés  
Departamento Tecnología Electrónica  
Universidad de Vigo  
Vigo, España  
rverdugo@uvigo.es

Diego Castillo Fernández  
CTAG – Centro Tecnológico de  
Automoción de Galicia  
Vigo, España  
diegocastillofernandez@gmail.com

**Abstract**— Esta comunicación presenta un sistema de medida de la calidad de las aguas mediante una red de sensores conectados al servicio de comunicación y almacenamiento Sigfox. Este servicio constituye la forma más económica de realizar un sistema IoT de área ancha.

**Keywords**— IoT, Sigfox

## I. INTRODUCCIÓN

Esta comunicación describe el “Trabajo Fin de Grado” de un alumno del Departamento de Tecnología Electrónica que trabajó en colaboración con un grupo del departamento de Ingeniería Química.

La contaminación del agua, también conocida como contaminación hídrica, se suele definir como la acumulación de elementos externos a la naturaleza de ese medio, que en el caso objeto de este estudio es el agua. La introducción de estas nuevas sustancias, acarrea un cambio en el medio contaminado, y puede llegar a ser perjudicial para el equilibrio vital de los seres vivos que tienen alguna relación con él.

De forma general, la contaminación de las aguas es producida por actividades humanas. La industrialización y el consumo desmesurados presente en la sociedad actual, dan lugar a una gran generación de residuos. Estos desechos en gran medida no son depositados ni recogidos en un lugar adecuado para su posterior tratamiento, por lo que acaban contaminando las aguas.

Por lo general se suele hacer una clasificación en cuanto a donde se origina la contaminación del agua.

Si el foco contaminante es conocido, se dice que la fuente es puntual. Ejemplos claros de este tipo de contaminación son las tuberías que vierten residuos, un pozo de petróleo...

Sin embargo, si el foco contaminante no se puede determinar de una forma clara, decimos que la fuente es difusa. Un ejemplo claro sería un agente contaminante desplazado por un río.

Con respecto a los tipos de contaminación del agua, esta, se puede ver afectada, por compuestos minerales, los cuales pueden condicionar a las propiedades organolépticas del agua o incluso producir el desarrollo de las algas y la eutrofización.

Otras veces, la contaminación, viene de la mano de compuestos orgánicos (hidrocarburos, detergentes, etc...), compuestos microbiológicos (fenoles, bacterias, etc....) o factores que alteran la temperatura del agua.

El requisito fundamental por parte del peticionario es que el sistema elegido para las comunicaciones permita el envío de datos, desde un punto sin garantía de acceso a redes Wifi o GSM ni corriente eléctrica. Por otra parte, se especifica que no se tendrá conexión a la red eléctrica, por lo tanto, el

segundo objetivo es desarrollar un sistema de alimentación mediante baterías recargadas con energía solar. Otro de los objetivos consiste en recoger la información capturada de los sensores. Los objetivos posteriores son las configuraciones de todas las herramientas necesarias para hacer la comunicación posible. Esto incluye la programación del hardware y de las plataformas hacia las que se redirecciona la información.

Todo estos requisitos que se plantean pueden ser atendidos siguiendo los principios del IoT (Internet of Things) [1] [2] por lo que solucionaron haciendo uso de los recursos disponibles en esta tecnología. El diagrama de bloques del sistema se muestra en la figura 1.



Fig. 1. Diagrama de bloques del sistema

## II. ELEMENTOS DEL SISTEMA

### A. Sensores

Los sensores utilizados son: temperatura, pH y conductividad.

La contaminación térmica se produce por variaciones de temperatura que alteran perjudicialmente la calidad del agua, lo que supone una disminución de la solubilidad del oxígeno del agua, necesario para los seres vivos que se nutren de oxígeno. Se emplea el sensor DS18B20 que proporciona mediciones de temperatura Celsius de 9 a 12 bits. El DS18B20 se comunica a través de un bus de 1 cable que, por definición, requiere solo una línea de datos (y tierra) para la comunicación con un microprocesador central. Además, el DS18B20 puede derivar energía directamente de la línea de datos ("energía del parásito"), eliminando la necesidad de una fuente de alimentación externa [3][4].

El pH marca la concentración de iones de hidrógeno que se encuentran en una disolución. Es un indicador de acidez o alcalinidad. Un pH neutro favorece la supervivencia de microorganismos en el agua, y a su vez, la inexistencia de estos, que equivale a un aumento o disminución del pH, pueden indicar una baja calidad del agua, o que esta muestre algún tipo de anomalía. Por lo tanto, un pH alterado puede dar lugar a una situación desfavorable para la biodiversidad. Se utiliza el módulo PH-4502C que es un dispositivo que permite medir el PH con ayuda de un electrodo como el E201-BNC para el cual esta reservado un conector BNC [5].

La conductividad, es la facilidad que presenta el agua para permitir el paso de corriente eléctrica a través de ella. Esto depende directamente de la concentración de iones. En las masas de agua dulce, la conductividad está determinada principalmente por la geología del terreno que la contiene. El rango de conductividad del agua dulce varía entre 50 y 1.500  $\mu\text{S}/\text{cm}$ . Para la medición de este parámetro se utiliza el medidor de conductividad eléctrica analógico DFR0300 (con compensación de temperatura) [6].

### B. Sistema de comunicaciones

La velocidad necesaria de envío de datos es baja, ya que las magnitudes a medir cambian su valor de forma lenta, por lo que no es necesario reflejarlo a tiempo real en el punto de visualización.

Por lo tanto, se puede deducir que este sistema necesita una gran área de cobertura, que permita el envío de los datos independientemente de la zona desde la que se transmitan, y no es necesario que la velocidad del envío sea alta, lo que permite reducir el consumo de energía.

Otra característica a tener en cuenta a la hora de diseñar el sistema, es la relativa al punto de acceso a los datos recogidos. La estructura óptima es aquella que aporta flexibilidad de lectura y trabaja desde cualquier lugar en cualquier tipo de dispositivo con los valores muestreados.

La búsqueda del sistema de comunicaciones óptimo se centra en las redes IoT (Internet of Things) (figura 2). Este campo reciente dentro de las telecomunicaciones hace alusión a aquellos sistemas diseñados para intercomunicar objetos entre sí. Son tecnologías ya nacidas y diseñadas para suplir las especificaciones propias de un sistema IoT. Por lo tanto, tienen un gran alcance de cobertura, y unos consumos y precios muy bajos. Destacan dos sistemas dentro de esta clasificación, las redes LoRa [7] y las redes SigFox [8].

Ambas son consideradas redes LPWAN (Low-Power Wide-Area Network) y UNB (Ultra Narrow Band). Los dos términos hacen referencia a las características ya mencionadas con anterioridad. Por otra parte, las redes LPWAN trabajan en las bandas de radiofrecuencia ISM, que están reservadas para su uso no comercial, por lo que no es necesario un pago para utilizarlas.

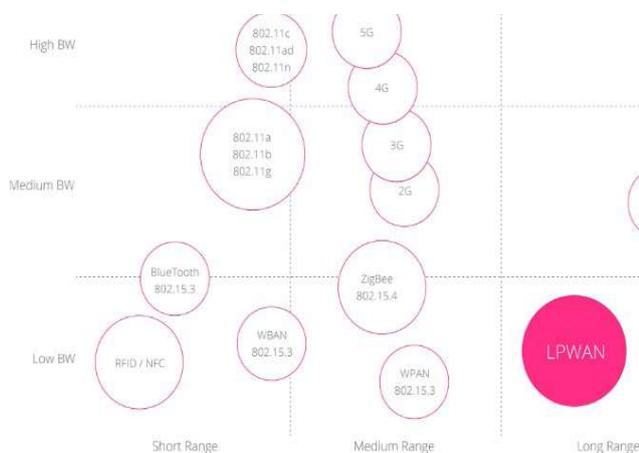


Fig. 2. Relación entre ancho de banda y alcance

La tecnología LoRa, patentada por Semtech, se encarga del nivel físico. Se utiliza la técnica del espectro ensanchado,

mediante radiofrecuencia y envía la información punto a punto entre módulos compatibles con este sistema. El envío de la información tiene un gran rango de alcance, llegando hasta los 20 km en condiciones óptimas.

Una vez que se supera la capa física, se alcanza la capa de red. Este segundo nivel recibe el nombre de LoRaWan (figura 3), cuya función es gestionar la comunicación entre los dispositivos y controlar los canales y sus parámetros: Ancho de banda, cifrado, etc... Como punto adverso, los sistemas LoRaWan emplean un concentrador para realizar la comunicación entre los sensores y el servidor en la nube. La parte negativa radica en que estos concentradores tienen un precio relativamente elevado y son complejos de fabricar.

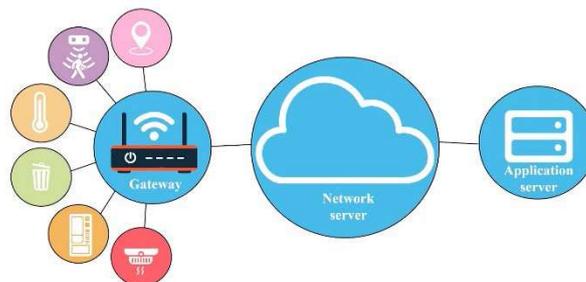


Fig. 3. Esquema de conexión mediante LoRaWan

Sigfox comparte las características propias de los sistemas LPWAN. Por lo tanto, es muy similar a LoRa en cuanto a prestaciones, pero es lo opuesto en cuanto a necesidades para su puesta en funcionamiento. SigFox no necesita de concentradores ni de aplicaciones externas a las que redirigir la información. Ya que todo esto ya lo aporta la propia compañía a cambio de una suscripción mensual de bajo coste. El gran inconveniente, además, del pago de los servicios, es que existe un número límite de mensajes que se pueden enviar diariamente. Con el pago de la suscripción a su servicio, mediante un módulo compatible con él, se pueden enviar un máximo de 140 mensajes de hasta 12 bytes cada uno. Por lo tanto, cada 10 minutos aproximadamente se pueden actualizar los valores muestreados por el sistema. Es un intervalo de tiempo suficientemente pequeño como para tener un control preciso de las magnitudes que se quieren medir, ya que estas cambian su valor lentamente respecto el tiempo

### C. Control

La gama MKR de Arduino son una serie de placas que ofrece la compañía. Destacan por reducir el tamaño y la tensión con la que trabajan buscando la eficiencia, lo que las hace ser las seleccionadas en muchas ocasiones para proyectos IoT.

El Arduino MKR FOX 1200 [9] (figura 4), está conformado principalmente por un microcontrolador de la familia SAMD21 y un módulo que proporciona la conectividad a la red de Sigfox, el ATA8520. Para que este último módulo haga el envío del mensaje a la red de SigFox, es necesario conectar a la placa una pequeña antena que está incluida con su compra. En las especificaciones que da el fabricante, se puede ver que la placa funciona a 32 bits, con una tensión de 3,3 V y a una frecuencia de reloj de 48 MHz. En cuanto a conectividad, permite comunicaciones por UART, SPI o I2C. Otras características a destacar son sus memorias Flash y SRAM de 256 kB y 32 kB respectivamente.

Uno de los puntos positivos que tiene la compra de esta placa, es que incluye una suscripción gratuita al servicio de Sigfox de 2 años, por lo que su precio queda amortizado únicamente por este ahorro, ya que el precio de la suscripción anual es de poco menos de 20 euros.

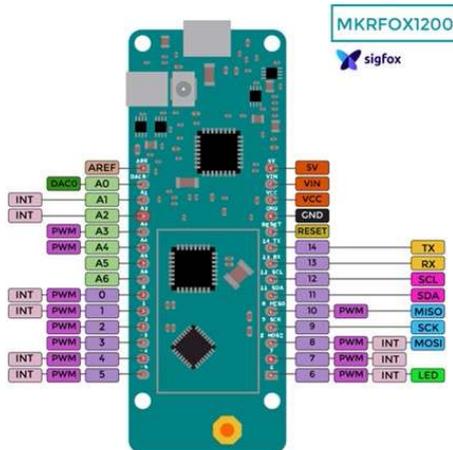


Fig. 4. Terminales del Arduino MKR FOX 1200

**D. Alimentación**

Dado que el prototipo debe funcionar de forma continua durante un largo periodo de tiempo, es necesario reducir su consumo al máximo. Con esto, se busca que el aporte de energía por parte de un panel solar, en buenas condiciones lumínicas sea superior a las demandas de corriente del sistema.

El circuito desarrollado tiene dos etapas muy diferenciadas. La primera, se puede considerar la etapa activa, ya que en ella se recoge la información y se redirige al siguiente lugar de interés. En la segunda etapa, únicamente está esperando un tiempo determinado para poder renovar nuevamente el valor de las variables medidas.

En la etapa activa, los consumos son los que soliciten los sensores para hacer las medidas, el de la placa de Arduino y el del módulo de SigFox para realizar el envío de la información. Estos gastos se optimizan mediante la programación y por la selección del sistema de comunicaciones eficiente.

En la etapa de espera, es donde realmente se puede reducir el gasto de energía, ya que, por cada 10 minutos de espera, son unos 15 segundos de envío de etapa activa.

Para conseguir estas temporizaciones se utiliza el temporizador TPL5110. Este pequeño circuito integrado, está desarrollado pensando en reducir al máximo los consumos. Para ello, mediante un MOSFET, alimenta o corta la alimentación del resto del montaje, siempre teniendo cuidado en el encendido y apagado del proyecto. Por lo tanto, en los momentos de espera el único módulo que consume energía es este temporizador y la placa sobre la que está montado.

Para dimensionar la placa solar [10] es necesario hacer unos cálculos basados en el consumo de los componentes (figura 5) y radiación solar del lugar en que se encuentre el sistema. Dichos cálculos dieron que un panel de 140 cm<sup>2</sup> cuyas especificaciones de tensión y potencia son 6V y 2W respectivamente cumplía sobradamente.

Por otra parte, para mantener la tensión en los momentos en lo que no existe luz solar como las noches o los días nublados, es necesario el uso de una batería que almacene la corriente generada durante los días soleados y la ceda en los momentos que sea necesario. Para este proyecto se decidió usar una batería tipo LiPo [11].

	Tiempo (s)	Corriente (µA)	Tensión (V)	Potencia media (W)
Etapas de temporización	720	20	3,3	-
Etapas activa	15	60000	3,3	-
Valor medio	-	1250	3,3	0,004

Fig. 5. Tabla de consumos según fase de actuación.

Este tipo de baterías se suelen usar en proyectos de electrónica por la facilidad que supone trabajar con ellas. Existen multitud de módulos preparados para su carga y descarga. Además, otras características como su gran adaptabilidad a espacios (ya que se puede llegar a deformar), o su ligereza, las hacen muy demandadas. Para el cargador se ha seleccionado el “Solar LiPo charger” de la marca dfrobot mostrado. Este cargador permite interconectar la batería LiPo, el panel solar y el circuito que se quiere alimentar, de forma que es capaz de cargar la batería mediante el panel solar a la vez que también permite darle tensión al circuito.

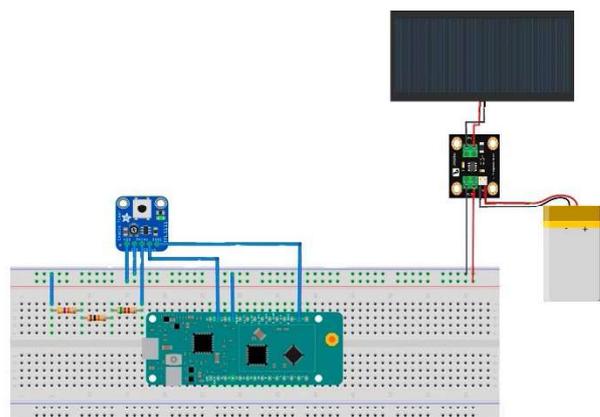


Fig. 6. Arduino MKR FOX 1200 conectado al temporizador, placa solar, cargador y batería LIPO.

**E. Montaje del equipo**

Para este proyecto se ha seleccionado una caja con un grado de protección IP 65. Lo que supone que el circuito es totalmente estanco al polvo y tiene protección contra chorros de agua que es importante a la hora de colocar el sistema en su destino final (figuras 7 y 8).

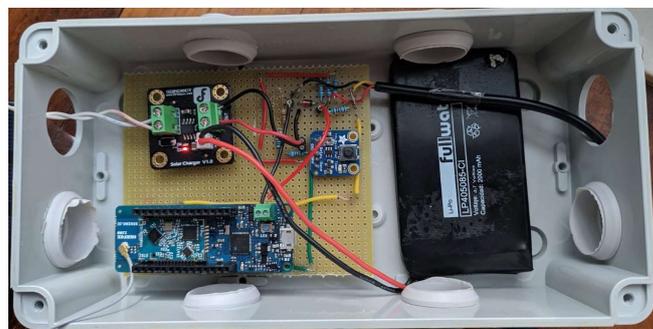


Fig. 7. Disposición componentes en caja IP65.



Fig. 8. Equipo completo.

### III. DESARROLLO DEL SISTEMA

En un primer punto se programa la placa de Arduino para que lea los sensores y envíe a través de la red de SigFox la información obtenida hasta el siguiente punto en su camino hasta llegar al usuario final (figura 9).

Una vez que la información abandona la placa, los datos viajan hasta una antena de la compañía que proporciona el servicio de telecomunicaciones, publicando los resultados recibidos en el 'back-end' de SigFox. Esta plataforma es consultable teniendo acceso a la cuenta de SigFox que se incluye con la suscripción.

Aunque los datos ya se pueden recoger en esta plataforma, la visualización de la información se hace difícil. Es esta la razón por la que los datos se redireccionan hacia un servicio de almacenamiento y tratamiento de la información especializado en IoT. En este caso se selecciona la plataforma "ThingSpeak".

Una vez los datos se encuentren disponibles en esta página, el usuario puede tener acceso a los datos a través de las aplicaciones 'front-end' diseñadas y seleccionadas. Aportando la máxima comodidad para la correcta interpretación de la información.

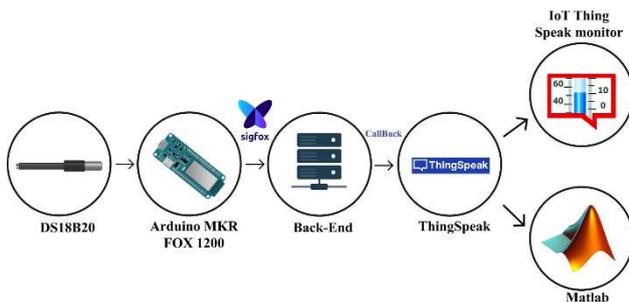


Fig. 9. Esquema general del funcionamiento del sistema.

#### A. Programación Arduino

Para facilitar la programación de este apartado, se emplean bibliotecas (library) públicas que se encargan de las tareas más complejas, como puede ser recibir la información del protocolo del sensor o enviar los datos a SigFox. En concreto se emplean:

- Biblioteca OneWire: Facilita la comunicación con el protocolo 1-wire con el que funciona el DS18B20.
- Biblioteca DallasTemperature: Se trata de una biblioteca (library) especializada en el sensor de temperatura.
- Biblioteca SigFox: Facilita los comandos para interactuar con la red de la compañía y hacer los envíos de los mensajes.

#### B. Sigfox

En primer lugar, para poder ver el valor real con decimales en las distintas zonas de redireccionamiento de la información, se ha elegido enviar cada una de las variables como 'float', esto supone que cada dato ocupa un total de 32 bits, es decir, 4 bytes. Por lo tanto, con el valor de la temperatura, el pH y la conductividad, ya se emplea el tamaño de mensaje máximo que permite enviar el sistema de SigFox.

En el propio 'back-end' ya se pueden visualizar los valores en la sección de mensajes. Sin embargo, los datos llegan en forma de una cadena de 12 bytes en hexadecimal 'little endian' (bits menos significativos primero). Esto hace que los valores sean complicados de reconocer a simple vista. Para solucionarlo, la plataforma permite personalizar los mensajes recibidos, de forma que se secciona la cadena de 12 bytes en los valores indicados. Para personalizar el mensaje recibido hay que seleccionar el tipo de dispositivo que se quiera personalizar y entrar en editar. Dentro de esa pantalla, existen múltiples opciones de edición, sin embargo, la que interesa, es la que recibe el nombre de 'Payload display'.

En este apartado, se permite especificar que se usará una estructura personalizada para dividir los mensajes recibidos. La forma en la que se seccionen los mensajes se determina siguiendo la siguiente estructura:

Nombre de la variable: Byte de inicio: Tipo de dato: Tipo de endian

Como se puede ver en la figura 10, se sigue la estructura que se acaba de mencionar para dividir los 12 bytes en 3 variables con sus respectivos nombres. De esta forma, en la sección de los mensajes recibidos se especifica el valor de cada variable en decimal bajo la cadena en hexadecimal.

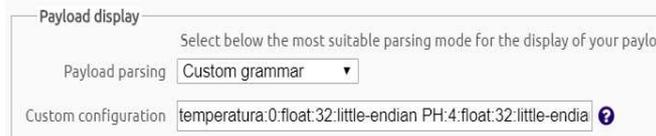


Fig. 10. Payload display.

En este momento ya se podría usar el 'back-end' (figura 10) para visualizar la información. Sin embargo, el manejo de esta página no es intuitiva ni cómoda. Por lo tanto, se decide redirigir la información a otro lugar que almacene la información y permita visualizarla y trabajar con ella con facilidad.

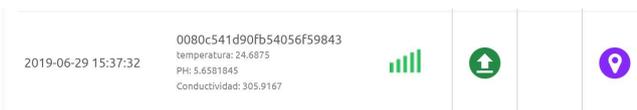


Fig. 11. Mensaje Sigfox ‘back-end’.

Para redirigir la información de forma automática cada vez que llega un mensaje, se debe configurar la sección de ‘Callbacks’ que existe en el ‘back-end’. En esta sección se especifica a donde se quiera enviar la información, además de las APIs que controlan estas acciones.

Para crear una ‘Callback’ nueva hace falta entrar en esa sección editando el tipo de dispositivo. Posteriormente se selecciona crear una ‘Custom callback’, lo que da acceso al menú para editar sus características (figura 12).

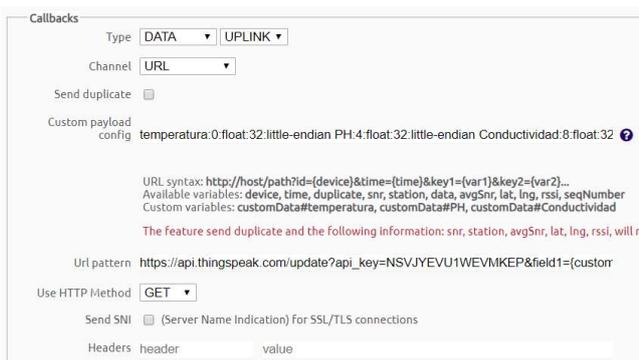


Fig. 12. Edición de ‘call-back’.

Para configurar esta sección, se tienen que completar los siguientes apartados:

- **Type:** Especifica el tipo de dato que se va a transmitir. Las opciones a escoger son:
  - **DATA:** Quiere decir que la información enviada es un dato. Se puede seleccionar si el dato es solo de subida (esta es la opción seleccionada), o bidireccional, donde se espera una respuesta por parte de la página hacia la que se redirige la información.
  - **ERROR:** Advierte de posibles fallos en la comunicación.
  - **SERVICE:** El mensaje redireccionado se trata de servicios como la geolocalización o una confirmación de que el envío fue correcto.
- **Channel:** Especifica qué tipo de canal se usa para enviar los datos, como por ejemplo por URL o por correo electrónico.
  - **Custom Payload Config:** Se configura como se secciona el mensaje en las distintas variables que contiene. Su estructura es la misma que la mostrada en la Ilustración 65.
  - **URL Pattern:** Define la llamada a la plataforma que se va a emplear para redirigir la información. Los datos a introducir en esta sección se obtienen de la página a la que se redirigirá la información.
  - **HTTP Method:** Se emplea para especificar el método HTTP usado. En este caso se emplea GET, ya que la

página a la que se redirige la información es la que se encarga de solicitarla.

A continuación hay que seleccionar el destino de la información. Teniendo en cuenta las características de este trabajo podría ser una plataforma gratuita de IoT como son Thinger.io, Arduino Cloud, Cayenne.

### C. ThingSpeak

En el caso concreto de este proyecto se ha decidido que la plataforma a la que se envíen los datos sea ThingSpeak. Se trata de la apuesta por parte de MathWorks [12] al mundo del IoT, por lo que existe una extensa documentación además de ser una de las plataformas más usadas en este tipo de proyectos IoT ya que permite visualizar la información de manera sencilla y contar con un gran número de programas externos que interaccionan con esta página, como por ejemplo el propio Matlab.

Para empezar a usar la página es necesario crear una cuenta. En caso de que el proyecto no sea comercial, como es el caso, el uso de los servicios que aporta la plataforma es totalmente gratuito, con algunas limitaciones respecto la versión de pago. Sin embargo, si se desea hacer un uso comercial del servicio es necesario pagar una suscripción a este servicio.

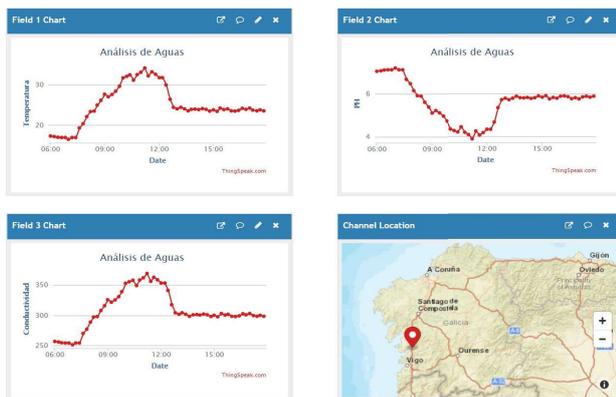
Una vez se dispone de la cuenta, se procede a crear un nuevo canal, donde se almacenan las distintas variables que se recogen del ‘back-end’. Cada canal dispone de un menú con distintos apartados, siendo los más destacados los de ajustes, los de visualización y el de las claves API.

Para poder recoger la información almacenada en el canal o para publicarla en el mismo, es necesario el uso de unas claves exclusivas para cada canal que se generan de forma automática. Para facilitar el intercambio de información entre distintos programas, en el apartado de claves API se generan los ‘API request’, siendo uno de ellos el que se usa como modelo para completar el ‘URL Pattern’ de la configuración del callback de SigFox.

En el apartado de ajustes se pueden personalizar distintas características. Por ejemplo, se permite editar el nombre y la descripción del canal, así como los campos en los que se divide la información que se almacenan en este. También se puede añadir información extra como por ejemplo la localización del origen de los datos.

En los apartados de visualización se puede interpretar la información mediante gráficas y ‘widgets’ configurables. Por cada campo configurado en los ajustes del canal se genera una nueva gráfica como las mostradas en la figura 13. En esta misma sección se permite introducir un cuadro con la localización especificada en los ajustes.

En la sección de ‘Sharing’ se puede seleccionar si la información del canal es visible por todo el mundo, si solo lo pueden ver tus seguidores o si únicamente lo puede ver el propietario de la cuenta. Es por ello que existen dos pestañas de visualización, una para lo que puede ver el propietario y otra para lo que puede ver un usuario externo.

Fig. 13. Gráficas the *ThingSpeak*.

*ThingSpeak* es una plataforma accesible desde cualquier dispositivo con conexión a internet y con una interpretación de la información directa y visual, lo que facilita que cualquier persona la consulte. Sin embargo, no da opción a consultar la información de un periodo de tiempo concreto o de comparar distintos resultados. Por otra parte, la visualización de la información en dispositivos móviles no está demasiado bien optimizada. Por todo esto, se ha decidido que la etapa final de visualización no sea directamente la página de *ThingSpeak*.

#### D. Matlab

Como la plataforma de *ThingSpeak* es de la misma compañía que Matlab, ya existen instrucciones para solicitar los datos a un canal concreto conociendo su 'ID' y su 'API Key'. Por lo tanto, resulta sencillo consultar la información almacenada en un periodo concreto de tiempo. Es por ello que se ha decidido usar este programa para crear la aplicación que usará el usuario final.

El diseño de la aplicación se hizo mediante interfaces gráficas, es decir mediante el apartado de GUI del que dispone Matlab. En este apartado dentro de Matlab se permite arrastrar hasta el área de trabajo distintos componentes, como botones, gráficas, bloques de texto... De forma que permite programar las acciones que realiza el programa una vez se interacciona con alguno de estos elementos.

Al iniciar la aplicación se despliega un menú con únicamente tres botones. Cada uno de ellos despliega una interfaz con distintas funcionalidades para adaptarse a los requerimientos del usuario: Visualizar, Comparar y Tabla de Datos.

En el programa de visualizar se despliega una interfaz donde se tienen 3 apartados diferenciados (figura 14):

- El primero es el recuadro donde se selecciona el intervalo de tiempo del que se va a demandar los datos recogidos. Este recuadro tiene dos tipos de funcionamiento, el modo 'Básico' y el modo 'Personalizado'.

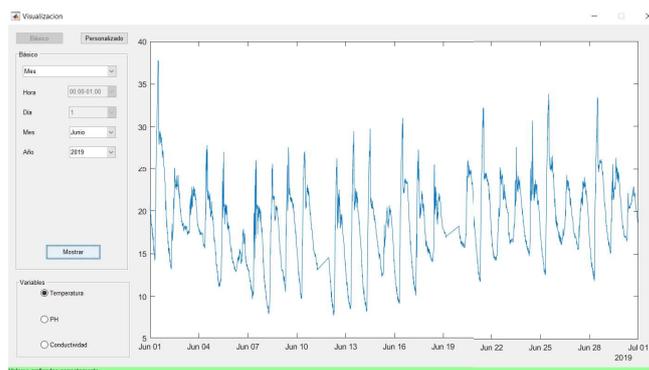
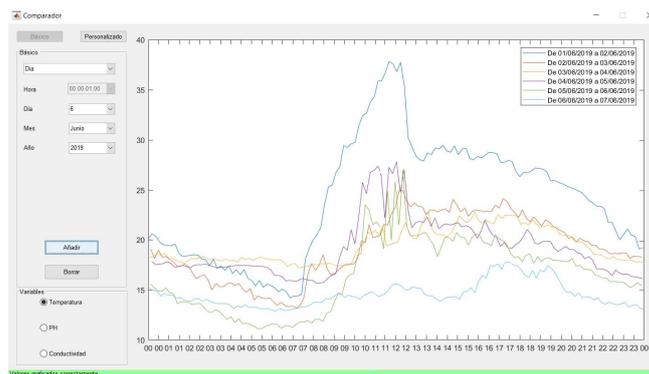
- El recuadro inferior incluye la opción de elegir que dato se desea ver gráficamente. Únicamente se permite ver una de las variables a la vez ya que para ver más de una gráfica a la vez existe otra interfaz distinta.

- El tercer recuadro es la gráfica que se dibujará con las especificaciones de los otros dos recuadros una vez se pulse el botón mostrar.

El modo comparador es similar al modo visualización, pero tal y como su nombre indica, está enfocado a tener varios valores simultáneamente, de forma que se puedan comparar a simple vista los resultados obtenidos (figura 15).

La tabla de datos (figura 16) comparte con los apartados previos únicamente el cuadro de la selección del rango de tiempo y la barra de estado. En lugar de una gráfica, cuando se pulsa el botón mostrar se pasan a un formato de tabla todos los datos recogidos durante el tiempo seleccionado. En la tabla se crea una entrada por cada medición. Una medición incluye el instante en que se realizó y los valores de las tres variables en ese instante.

Una vez que se ha muestreado la información, se activan en el cuadro de la zona inferior dos nuevos botones. En ellos se puede solicitar guardar los datos de la tabla en formato CSV, que es compatible con Excel, o bien en formato .mat que es compatible con Matlab. De esta forma el registro de valores queda a disposición absoluta del usuario de la aplicación, lo que le permite hacer con ellos lo que considere necesario.

Fig. 14. Modo Visualizador *Matlab*.Fig. 15. Modo Comparador *Matlab*.

	Fecha	Temperatura	PH	Conductividad
1	06-Jun-2019 05:03:38	13.2500	7.7932	209.6667
2	06-Jun-2019 05:14:47	13.2500	7.7932	222.6667
3	06-Jun-2019 05:25:55	13.1250	7.8166	229.8333
4	06-Jun-2019 05:37:03	13.1250	7.8166	226.8333
5	06-Jun-2019 05:48:12	13.1250	7.8166	212.8333
6	06-Jun-2019 05:59:19	13	7.8399	211.0000
7	06-Jun-2019 06:10:28	12.8750	7.8632	217.1667
8	06-Jun-2019 06:21:37	13.1250	7.8166	225.8333
9	06-Jun-2019 06:32:45	13	7.8399	229.0000
10	06-Jun-2019 06:43:53	12.9375	7.8516	207.5833
11	06-Jun-2019 06:55:00	13.0625	7.8282	215.4167
12	06-Jun-2019 07:06:09	13.1250	7.8166	222.8333
13	06-Jun-2019 07:17:17	13.1875	7.8049	211.2500
14	06-Jun-2019 07:28:25	13.3125	7.7816	207.0833
15	06-Jun-2019 07:39:33	13.3125	7.7816	230.0833
16	06-Jun-2019 07:50:42	13.5000	7.7466	228.3333
17	06-Jun-2019 08:01:50	13.7500	7.6999	233.0000
18	06-Jun-2019 08:12:58	13.7500	7.6999	212.0000
19	06-Jun-2019 08:24:06	13.7500	7.6999	211.0000
20	06-Jun-2019 08:35:14	14.1250	7.6299	226.5000
21	06-Jun-2019 08:46:23	14.0625	7.6416	215.0833
22	06-Jun-2019 08:57:31	14.3125	7.5949	241.7500
23	06-Jun-2019 09:08:39	14.5625	7.5482	217.4167
24	06-Jun-2019 09:19:47	14.5625	7.5482	217.4167
25	06-Jun-2019 09:30:56	14.2500	7.6066	216.3333
26	06-Jun-2019 09:42:03	14.5000	7.5599	236.0000
27	06-Jun-2019 09:53:11	14.3125	7.5949	240.7500

Fig. 16. Modo Tabla de datos *Matlab*.

#### IV. CONCLUSIONES

Este trabajo muy interesante explica el desarrollo de un sistema de adquisición para la medida de variables ambientales que permiten estudiar la contaminación hídrica y, en concreto, la temperatura, conductividad y pH de aguas bajo estudio (ríos, presas, etc.). El resultado es una herramienta de bajo coste y utilidad en el estudio de la contaminación.

Desde el punto de vista docente el trabajo es importante porque establece la relevancia de los sistemas de telecomunicaciones en los grados de ingeniería industrial.

#### AGRADECIMIENTOS

Esta investigación ha sido financiada por el Ministerio de Economía y Competitividad de España (MINECO) (Proyecto CTM2017-87326-R).

#### BIBLIOGRAFÍA

- [1] M. B. Andrés, Internet de la cosas, Reus, 2018.
- [2] *Plataformas IoT*: <https://secmotic.com/plataforma-iot/>.
- [3] [https://navlampmechatronics.com/blog/46\\_tutorial-sensor-de-temperatura-ds18b20.html](https://navlampmechatronics.com/blog/46_tutorial-sensor-de-temperatura-ds18b20.html)
- [4] <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
- [5] <https://www.botshop.co.za/how-to-use-a-ph-probe-and-sensor/>.
- [6] [https://wiki.dfrobot.com/Gravity\\_Analog\\_Electrical\\_Conductivity\\_Sensor\\_Meter\\_V2\\_K%3D1\\_SKU\\_DFR0300](https://wiki.dfrobot.com/Gravity_Analog_Electrical_Conductivity_Sensor_Meter_V2_K%3D1_SKU_DFR0300)
- [7] <https://lorawan.es/>
- [8] <https://www.sigfox.com/en>
- [9] <https://store.arduino.cc/arduino-mkrfox1200>
- [10] M. Boxwell, Solar Electricity Handbook – 2019 Edition: A simple, practical guide to solar energy – designing and installing solar photovoltaic systems. Greenstream Publishing; Revised edition edition.
- [11] <https://www.sparkfun.com/datasheets/Batteries/UnionBattery-2000mAh.pdf>
- [12] <https://es.mathworks.com/>