



Universidad
Zaragoza

Trabajo Fin de Grado

Manager de Energía
basado en Tecnologías IoT

‘IoT Energy Manager’

Autor

Jorge Ramos Guillen

Director

Roberto Casas Nebra

Universidad de Zaragoza

Escuela de Ingeniería y Arquitectura

Departamento de Ingeniería Electrónica

Año 2020

RESUMEN

En el momento en el que nos encontramos el mundo está viviendo una nueva revolución industrial, la denominada revolución 4.0, liderada por el IoT o Internet de las cosas, un concepto que hace referencia a la capacidad de prácticamente todas las cosas de conectarse a internet para mejorar su rendimiento, compartiendo sus datos y descargando otros. Esta nueva revolución tiene uno de sus puntos principales en la denominada **medición inteligente**, poder monitorizar el consumo de recursos (agua, electricidad, combustible...) de manera precisa para saber cuánto se consume y para qué, facilitando una gestión mucho más eficaz y reduciendo el coste operativo.

La anterior revolución, la 3.0 o Tercera revolución industrial, como todas las anteriores, se dio como una convergencia entre las tecnologías de comunicación y los nuevos sistemas de generación energética. En esta revolución una de las consecuencias más notables fue un cambio en la mayor utilización de **energías renovables**, así como un acercamiento a una red eléctrica "inteligente" (Smart grid).

De la unión de ambas o más bien, de la mejora que aporta la primera sobre la segunda, surge la idea que da forma a este trabajo, tener la capacidad de gestionar, monitorizar y mostrar de manera sencilla tanto la producción de energía como su consumo en las distintas cargas de un edificio.

El diseño se basará en un dispositivo **Pigeon**, un sistema diseñado para el uso en sistemas de control y automatización, que será el encargado de recibir los datos de las distintas fuentes (caldera, generador de energía solar, consumo de las cargas...) la transformación mediante software **Python** a datos utilizables y el envío a la plataforma **ThingsBoard** para su visualización por parte del usuario.

ÍNDICE

| | |
|--|----|
| 1.INTRODUCCIÓN | 4 |
| 1.1. OBJETIVOS | 4 |
| 1.2. HERRAMIENTAS | 5 |
| 1.3. METODOLOGÍA | 6 |
| 2.ESTADO DE LA CUESTION | 8 |
| 2.1. MODBUS | 9 |
| 2.2. VBUS | 10 |
| 2.3. ACOMETIDA A LA RED ELÉCTRICA | 11 |
| 2.4. HOME CONNECT | 12 |
| 2.5. IFTTT | 13 |
| 3. DISEÑO DE LA ARQUITECTURA | 15 |
| 3.1. PIGEON | 15 |
| 3.2. DELTASOL | 16 |
| 3.3. CONTADOR CONTAZARA | 17 |
| 3.4. SONDA DS18B20 | 18 |
| 3.5. INVERSOR KOSTAL | 19 |
| 3.6. THINGSBOARD Y EL PROTOCOLO MQTT | 20 |
| 4. DESARROLLO DEL PROYECTO | 21 |
| 4.2. INSTALACION Y RECOLECCION DE DATOS | 21 |
| 4.2.1. GESTOR DELTASOL CS4 | 22 |
| 4.2.2. CONTADOR DE AGUA CONTAZARA | 25 |
| 4.2.3. SENSOR DE TEMPERATURA | 27 |
| 4.2.4. INVERSOR KOSTAL | 28 |
| 4.3. PANEL DE THINGSBOARD | 28 |
| 5. INSTALACION REAL | 33 |
| 5.1. TEMPERATURAS Y GESTION DEL AGUA | 34 |
| 5.2. PANEL FOTOVOLTAICO E INVERSOR | 36 |
| 5.3. ENERGÍAS | 37 |
| 6. CONCLUSIONES | 40 |
| 6.1. CONCLUSIONES DEL PROYECTO | 40 |
| 6.2. TRABAJO FUTURO | 41 |
| 6.3. CONCLUSIONES PERSONALES | 43 |
| 7. AGRADECIMIENTOS | 45 |
| 8. BIBLIOGRAFIA | 46 |
| 9. ANEXOS | |

TABLA DE ILUSTRACIONES

| | |
|--|----|
| <i>Ilustración 1 (Esquema del Pigeon)</i> | 5 |
| <i>Ilustración 2 (PyCharm)</i> | 5 |
| <i>Ilustración 3 (Ejemplo de panel de ThingsBoard)</i> | 6 |
| <i>Ilustración 4 (Gestor de energía Resol)</i> | 7 |
| <i>Ilustración 5 (Red MODBUS)</i> | 9 |
| <i>Ilustración 6 (Sistema VBUS)</i> | 10 |
| <i>Ilustración 7 (PZEM-004 Picefair)</i> | 11 |
| <i>Ilustración 8 (Factura en la aplicación de Iberdrola)</i> | 12 |
| <i>Ilustración 9 (HomeConnect)</i> | 13 |
| <i>Ilustración 10 (IFTTT)</i> | 14 |
| <i>Ilustración 11 (Diagrama de Bloques: Diseño de la Arquitectura)</i> | 15 |
| <i>Ilustración 12 (Información del Display de Contador)</i> | 17 |
| <i>Ilustración 13 (Señales Sel y Sdata de Contazara)</i> | 17 |
| <i>Ilustración 14 (Trama Contazara)</i> | 18 |
| <i>Ilustración 15 (Tiempos del protocolo 1-wire)</i> | 19 |
| <i>Ilustración 16 (Esquema MQTT)</i> | 20 |
| <i>Ilustración 17 (Conexión Vbus/USB)</i> | 22 |
| <i>Ilustración 18 (Conexión Resol-Pigeon)</i> | 22 |
| <i>Ilustración 19 (Diagrama de Flujo: Gestor de Resol)</i> | 23 |
| <i>Ilustración 20 (Json)</i> | 24 |
| <i>Ilustración 21 (Conexión Contazara-Pigeon)</i> | 25 |
| <i>Ilustración 22 (Diagrama de Flujo: Contazara)</i> | 26 |
| <i>Ilustración 23 (Montaje para la realización de pruebas en el ordenador)</i> | 26 |
| <i>Ilustración 24 (Conexión DS18B20-Pigeon)</i> | 27 |
| <i>Ilustración 25 (Diagrama de Flujo: Sonda DS1820)</i> | 27 |
| <i>Ilustración 26 (Diagrama de Flujo: Inversor Kostal)</i> | 28 |
| <i>Ilustración 27 (Diagrama de Bloques: Panel Thingsboard)</i> | 29 |
| <i>Ilustración 28 (Creación de dispositivo en ThingsBoard)</i> | 30 |
| <i>Ilustración 29 (Ultima telemetría)</i> | 31 |
| <i>Ilustración 30 (Creación de panel)</i> | 31 |
| <i>Ilustración 31 (Creación de widget)</i> | 32 |
| <i>Ilustración 32 (Postprocesamiento de variables)</i> | 32 |
| <i>Ilustración 33 (Diagrama de Bloques: Instalación Real)</i> | 33 |
| <i>Ilustración 34 (Gráfica: temperaturas)</i> | 35 |
| <i>Ilustración 35 (Gráfica: temperaturas 2)</i> | 36 |
| <i>Ilustración 36 (Gráfica: energía fotovoltaica)</i> | 37 |
| <i>Ilustración 37 (Diagrama de Barras: Energías)</i> | 38 |
| <i>Ilustración 38 (Panel completo)</i> | 39 |

1.INTRODUCCIÓN

En este apartado hablaremos de tres pilares básicos de este proyecto: objetivos, herramientas y metodología.

Trataremos tanto el objetivo principal como los secundarios necesarios para alcanzarlo.

También contaremos que herramientas generales que hemos utilizado para la recolección de los datos, su monitorización y su puesta a disposición del usuario.

Por ultimo comentaremos la metodología que hemos empleado para conseguir los objetivos de una manera organizada y eficiente.

1.1. OBJETIVOS

El objetivo principal de este proyecto es desarrollar un sistema de gestión energética integral que centralice todos los procesos en un solo dispositivo controlador, siendo capaz de mostrar en una interfaz sencilla toda la información al usuario. Para ello se necesitarán conseguir una serie de objetivos secundarios que detallamos a continuación:

- Controlar y monitorizar la producción de energía por parte de un panel solar.
- Controlar y monitorizar el consumo de la energía por parte de un sistema con varias cargas.
- Controlar y monitorizar el consumo de otros recursos como pueden ser el gas o el agua.
- Ofrecer esta información a un usuario a través de una interfaz sencilla y completa.
- Ser capaz de optimizar los consumos de todos los recursos utilizados por el sistema, tanto de manera neta como de manera económica fijándonos en la información de consumo el coste en diferentes tarifas u horas.
- Que todo esto se consiga desde un mismo dispositivo que reciba y redirija toda la información.

1.2. HERRAMIENTAS

Pigeon

El dispositivo que usaremos como unidad de control y automatización será un Pigeon RB300 de la empresa Kristech, se trata de un sistema basado en Raspberry Pi con capacidad para recibir y mandar señales digitales y analógicas, con puertos USB, Ethernet, CAN, RS-232, RS-485 y 1-Wire. Todas estas comunicaciones nos dan una flexibilidad y un potencial muy altos para el control de todo lo necesario para la gestión energética.

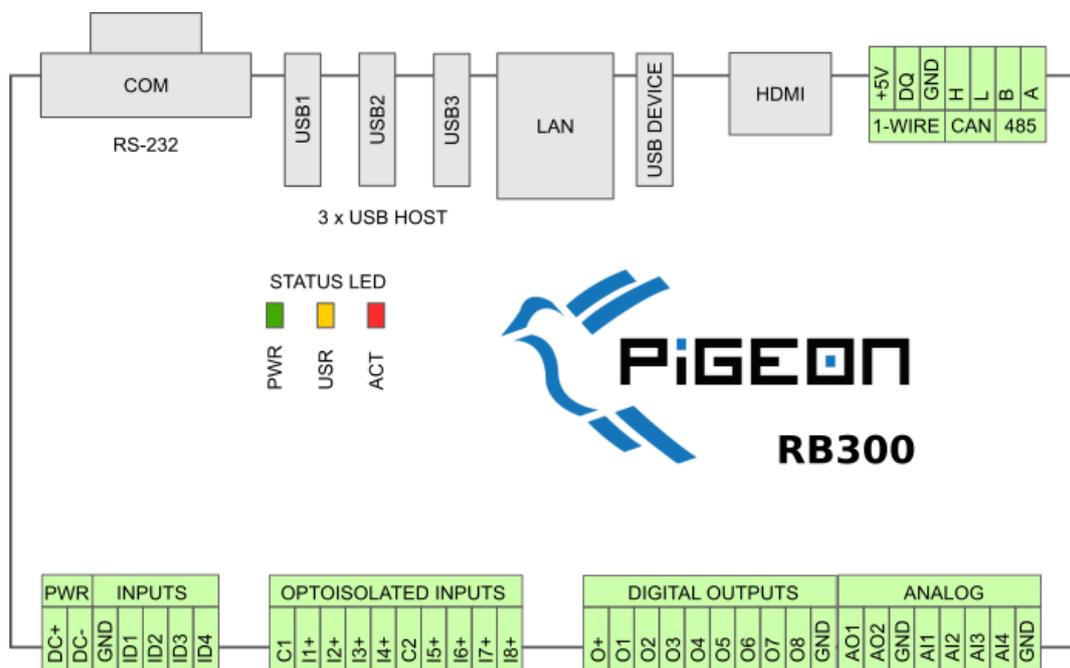


Ilustración 1 (Esquema del Pigeon)

Python-PyCharm

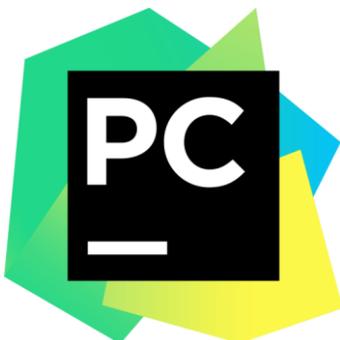


Ilustración 2 (PyCharm)

Para el software utilizaremos el lenguaje de programación Python con el entorno PyCharm. El principal motivo para el uso de este entorno es la capacidad que nos da para ejecutar el programa dentro del Pigeon monitorizándolo desde un ordenador lo que resulta óptimo para depurar errores y observar el comportamiento del programa durante su ejecución.

Thingsboard

La plataforma que usaremos para mostrar los datos y la interacción con el usuario será Thingsboard ya que nos ofrece una interfaz muy intuitiva para que cualquier usuario sea capaz de entender que está pasando en su edificio y ejecutar las acciones que desee.

El modo en el que nos comunicaremos con la plataforma será mediante un protocolo MQTT que explicaremos más adelante.

La sencillez a la hora de crear un servidor que podamos utilizar como puente entre el control y el usuario también es un factor a favor del uso de esta plataforma.



Ilustración 3 (Ejemplo de panel de ThingsBoard)

1.3.METODOLOGÍA

Para alcanzar los objetivos descritos en el apartado anterior el método de trabajo elegido consiste en lograr cada uno de ellos de una manera individual. Para ello ejecutaremos una metodología similar para cada proceso, empezaremos con la lectura de la documentación y repaso de códigos ya existentes para después empezar con pequeñas pruebas, comprobando el correcto funcionamiento de cada elemento, complicando y añadiendo matices en cada prueba hasta alcanzar el funcionamiento completo de cada objetivo, a modo de ejemplo utilizaremos el caso del gestor solar.

En primer lugar, recopilamos la documentación de la página web del fabricante de todos los dispositivos que utilizamos, del gestor DeltaSol CS4 y del adaptador VBus/USB, también repasamos códigos utilizados para la recepción de datos del gestor.

Una vez estamos familiarizados con el gestor, lo primero que comprobamos es si por el puerto USB somos capaces de recibir la información. todas estas primeras pruebas las realizamos desde un ordenador, ya que tanto el trabajo como la resolución de posibles problemas es más intuitiva. Conseguido esto, el siguiente paso es tratar esos datos en forma de bytes para convertirlos en información manejable, con la ayuda de la documentación previa y adaptándola a nuestras necesidades particulares conseguimos recibir la información, descodificarla y almacenarla de manera ordenada.



Ilustración 4 (Gestor de energía Resol)

Ahora nos centramos en conseguir mandar esta información a la plataforma Thingsboard y que ésta la muestre de manera clara y sencilla. Una vez conseguimos esto en el ordenador, hacemos funcionar el programa en el dispositivo Pigeon y tras hacer ajustes logramos que el sistema funcione.

Nuestra metodología consiste en seguir estos pasos con cada una de las funciones que deseamos y finalmente poner a trabajar todas juntas, consiguiendo en un solo dispositivo la gestión energética integral que nos marcamos como objetivo principal del proyecto.

2.ESTADO DE LA CUESTION

En este apartado vamos a analizar las soluciones actuales para obtener los distintos objetivos que en este trabajo nos hemos propuesto resolver, analizaremos distintos protocolos de comunicación, modos en los que el usuario puede acceder a esos datos, distintas plataformas que integran lo anterior en una misma solución, tipos de sensores y actuadores que se utilizan en el mercado actual, formas de conectarse a la red eléctrica general y otros temas similares.

Como visión general del asunto cabe destacar que la mayoría de soluciones no funcionan como un solo ente sino más bien como el trabajo en conjunto de varios de ellos, por eso el objetivo de este proyecto es poder integrar en una solución única todos los retos que propone la gestión energética de un edificio.

A modo de resumen vamos a tratar protocolos que se utilizan en este tipo de sistemas como pueden ser MODBUS o VBUS, trataremos también métodos que existen para acometer la red eléctrica de manera que podamos tener una monitorización de los consumos de la forma más rápida y fiable posible y para acabar trataremos tecnologías nuevas para conectar los dispositivos de una casa, tanto productos de determinadas marcas que conectan todos sus dispositivos como portales que te permiten configurar todo aquello que se pueda conectar a Internet.

2.1. MODBUS

MODBUS es un protocolo de transmisión de datos que se utiliza con mucha asiduidad en sistemas de generación fotovoltaica, está posicionado en el nivel 7 en el modelo OSI, y está basado en una arquitectura cliente/servidor(TCP/IP) o maestro/esclavo(RTU). Fue diseñado en 1979 por Modicon con el fin de servir en sus controladores lógicos programables (PLC).

Entre sus ventajas cabe destacar su sencillez a la hora de implementarlo, la capacidad de manejar bloques de datos sin suponer restricciones, que sea público y gratuito y su enfoque desde el principio para aplicaciones industriales.

En una red MODBUS cada dispositivo conectado tiene una dirección única y todos ellos pueden enviar ordenes, aunque lo habitual es permitir que solo lo haga uno de ellos que actúe como maestro. Los mensajes son recibidos por todos los dispositivos, pero la orden tan solo es ejecutada por el dispositivo objetivo de la misma.

En la ilustración 5 podemos observar un pequeño esquema de una red MODBUS en la que existen dos maestros y 4 esclavos, que podrían ser sensores de temperatura, humedad o actuadores en las persianas por nombrar algunas posibilidades. Como hemos dicho, la orden que se mande desde cualquiera de los dos maestros llegara a todos los esclavos, pero solo uno responderá o ejecutara la orden en cuestión.

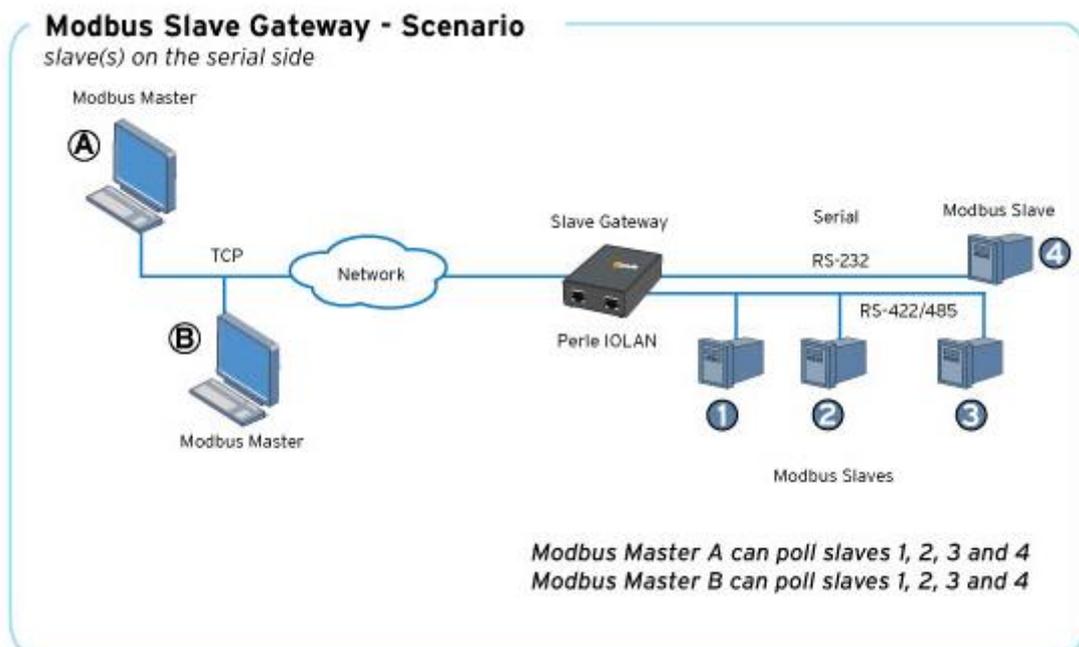


Ilustración 5 (Red MODBUS)

Como hemos dicho antes, ese protocolo es en el que se basan muchos sistemas de generación fotovoltaica y de gestión energética en general, tanto por su sencillez como por su eficacia y estandarización en el mercado.

2.2. VBUS

La marca comercial RESOL, una empresa alemana especializada en tecnología de control, utiliza un sistema llamado VBUS para controlar instalaciones de generación y distribución de energía en instalaciones tanto industriales como particulares. Para ello, la empresa comercializa adaptadores que pueden ser conectados a sus productos y hacer de puente entre el dispositivo e Internet u otro dispositivo de control.

En la ilustración 6 se ve el aspecto que tiene la interfaz VBus, en la que el usuario puede observar temperaturas y otros datos del sistema en tiempo real.

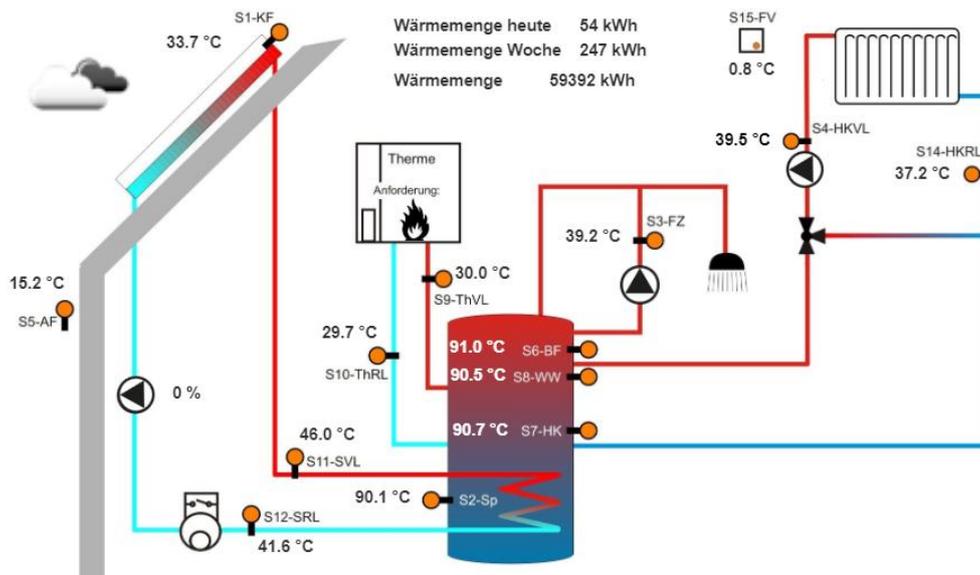


Ilustración 6 (Sistema VBUS)

Tras probar la interfaz y las posibilidades que otorga esta tecnología llegamos a la conclusión de que, si bien es un sistema útil y con posibilidades, la información y las opciones que daba al usuario estaban poco explotadas, además de ser difícilmente integrable en un sistema más grande, con productos de otras marcas o incluso con el contador de la luz del edificio.

Por estos motivos en el proyecto trabajaremos en parte con este método, ya que usaremos un regulador solar de esta marca, y lo conectaremos a nuestro dispositivo de

control mediante un adaptador de interfaz VBus/USB para extraer los datos y poder trabajar con ellos.

2.3. ACOMETIDA A LA RED ELÉCTRICA

Uno de los objetivos del proyecto es el control del consumo de la red eléctrica general y para ello es necesario que tengamos la información del consumo instantáneo de la instalación del edificio. Esto, unido al conocimiento del precio de la luz en determinadas horas y días, ayudará a reducir el gasto de la instalación y facilitará las decisiones acerca de cuándo efectuar las actividades que más consumo generen.

Para este fin hablaremos de dos métodos, el primero consiste en la instalación de un contador con salida de datos en el cuadro de luces y el segundo, el uso de contadores telegestionados como los del grupo Iberdrola o los contadores inteligentes de Endesa.

Sobre los primeros existen numerosas posibilidades en el mercado, como ejemplo hemos encontrado el siguiente dispositivo.



Ilustración 7 (PZEM-004 Pacefair)

Se trata de un producto de la marca Peacefair capaz de medir potencias de hasta 22Kw. Estos dispositivos pueden medir el consumo de cargas individuales o de toda la instalación, aunque para este fin necesita una implementación algo más compleja. Como vemos en la imagen tiene salida USB con la que podemos recibir los datos y aunque tiene

su propio software podemos usar estos datos con aplicaciones propias. Otra característica positiva de este método es su bajo coste, menos de 15 €.

En el segundo caso, el uso de contadores inteligente o telegestionados, los datos debemos recibirlos de la página que nos ofrece la compañía o mediante aplicación para dispositivos móviles. La adquisición de datos de esta manera parece más difícil, ya que, aunque para la simple consulta de facturas o datos de consumo sea una buena opción, no existe una manera de descargar los datos de manera instantánea y continua para poder trabajar con ellos en cualquier tipo de software propio.



Ilustración 8 (Factura en la aplicación de Iberdrola)

2.4. HOME CONNECT

Otra solución existente se basa en los sistemas que conectan los electrodomésticos de la casa a una aplicación accesible desde cualquier aparato con internet como puede ser un móvil, una tablet o un ordenador personal. Muchas marcas de electrodomésticos están desarrollando ideas similares que buscan homogeneizar los productos de un hogar bajo un mismo sistema de control. Aquí vamos a analizar **Home Connect**, el producto con el que BSH trabaja para ofrecer este servicio.

Home Connect ofrece la posibilidad de conectar a la red prácticamente todos los electrodomésticos de sus marcas. De hecho, desde enero de 2020 todos los aparatos que producen tienen la capacidad de estar conectados a la red y por tanto a Home Connect. Desde la lavadora hasta las lámparas pasando por el aspirador o la cafetera todo se puede conectar para intentar optimizar la planificación del día, la dieta o el gasto energético.

Como ejemplos, en la página web de BSH nos menciona la posibilidad de elaborar recetas con ayuda y guía, poder controlar lavadora y secadora para determinar a qué hora necesitamos la ropa seca o controlar el tipo de tejido, la elección del programa del lavavajillas de forma automática o las posibilidades que ofrece para la conciliación familiar.

También tiene la posibilidad de ser controlado mediante Alexa, el asistente virtual de Amazon además de integrarse en IFTTT (If This Then That), del que hablaremos en el siguiente apartado.

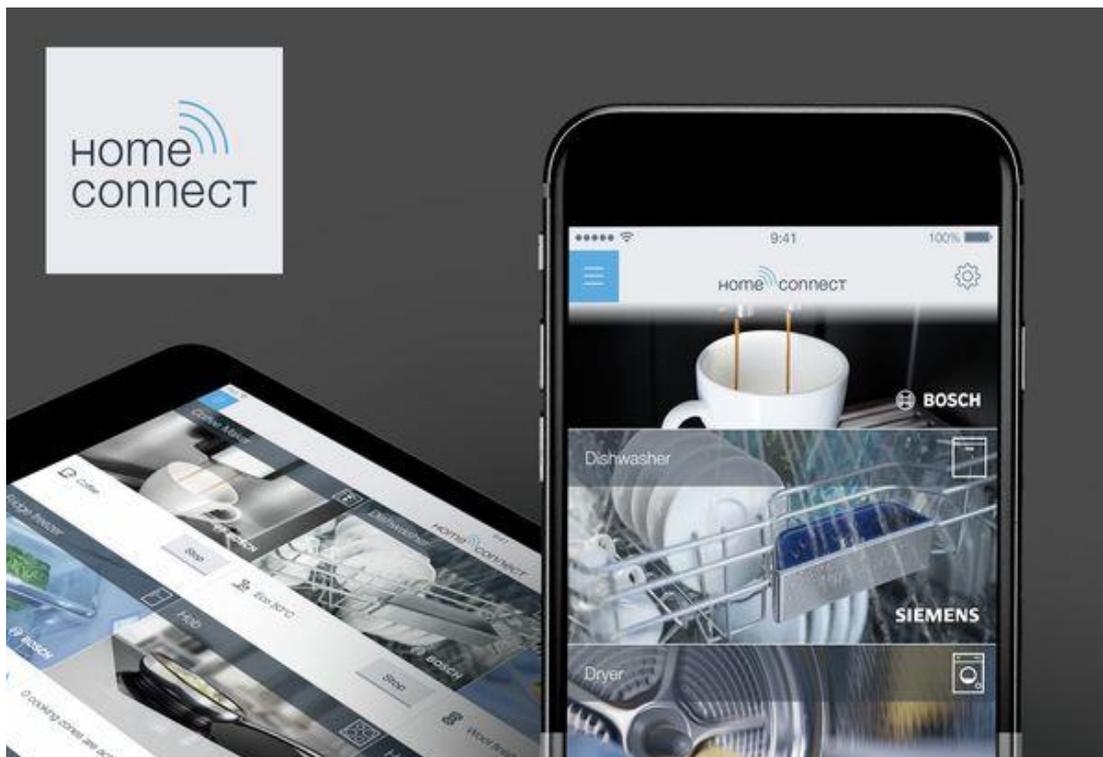


Ilustración 9 (HomeConnect)

Una de las desventajas de este sistema es que requiere que todos los electrodomésticos sean de la marca BSH además de que sean actuales y tengan la capacidad de conectarse a internet de manera autónoma. Otra es que la conexión de este programa con el consumo eléctrico de la casa en general y de la producción mediante energía solar tampoco es directa, lo que obliga a realizar otro tipo de solución para hacerlos funcionar juntos.

2.5. IFTTT

El servicio IFTTT se trata de un portal en el que puedes conectar prácticamente todo, desde la hora a la que llegará la comida a domicilio que has pedido, que parpadee una lámpara de tu salón, una alarma en tu reloj o saber cuándo te etiqueten en una foto en Facebook. Este servicio está basado en la idea de “If This Then That”, es decir “Si esto entonces aquello”, que quiere decir que cuando ocurra un suceso (This) se desencadene un proceso (Then) que active una acción (That).

Existen numerosos ámbitos de aplicación del IFTTT, desde unificar la información de las redes sociales, a controlar la calefacción o las luces del garaje, pero requiere que ya exista una aplicación dentro del portal, llamada receta, para controlar lo que queremos, o desarrollar nosotros una solución para problemas más concretos, que abarque todo lo que necesitamos en una sola receta, lo que aumenta la dificultad del desarrollo.



Ilustración 10 (IFTTT)

Si bien es una solución que podría servir, el hecho de que este muy orientado a acciones simples y no tanto a la recolección y el tratamiento de datos, que es lo que requiere el control de la generación y consumo de energía, hace que no sea trate de una solución óptima para el problema que nos planteamos.

3.DISEÑO DE LA ARQUITECTURA

En este apartado explicaremos la parte técnica de cada parte del proyecto, hablaremos del protocolo que utilizan para comunicarse, el software que viene integrado etc...

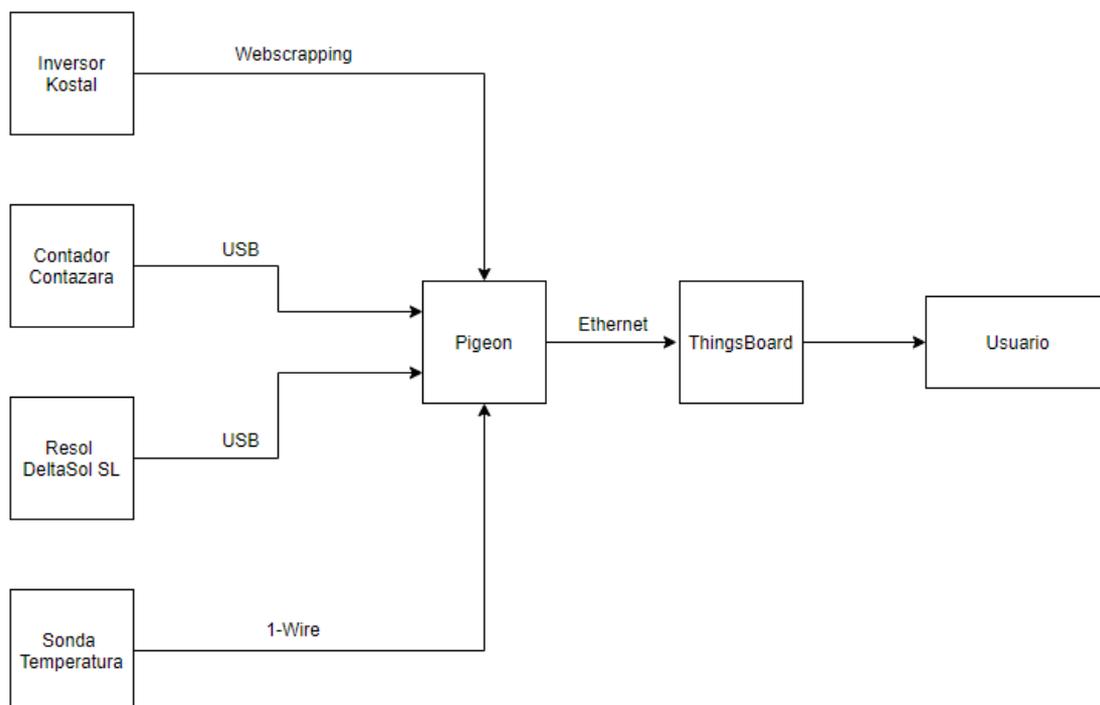


Ilustración 11 (Diagrama de Bloques: Diseño de la Arquitectura)

3.1. PIGEON

Pigeon RB300 es una computadora diseñada para su uso en sistemas de control y automatización. Funciona con un módulo de computo Raspberry Pi y con un sistema operativo de Linux.

Raspberry Pi se trata de un ordenador que tiene el tamaño aproximado de una tarjeta de crédito, consta de una placa base en la cual se monta un procesador, un chip gráfico y una memoria RAM. Lleva en el mercado desde 2006 cuando fue lanzado por la Fundación Raspberry Pi. Tenía como objetivo fomentar la enseñanza de informática en los colegios de todo el mundo.

En nuestro caso el modulo tiene una CPU 700MHz ARM1176JSF-S, una GPU Broadcom VideoCore IV, 512 MB de memoria RAM y 4GB de eMMC Flash.

El Pigeon utiliza esta Raspberry como base y amplía sus posibilidades agregándole circuitos de adecuación que nos permite usar directamente un conjunto muy rico de interfaces, entradas y salidas, tanto analógicas como digitales, un reloj en tiempo real o entrada de HDMI. Además de todo esto añade la carcasa que nos proporciona un diseño

robusto que cumple con los requisitos de EN 61326-1:2013 para funcionar en entornos electromagnéticos básicos e industriales.

En cuanto al software, el dispositivo viene con una versión de Linux basada en Raspbian Lite, que da un soporte completo a todas las interfaces y permite instalar la mayoría de lenguajes de programación como pueden ser: C/C ++, Python, Java...

En cuanto a las interfaces anteriormente mencionadas, encontramos 3 puertos USB, entrada para Ethernet, CAN, RS-232, RS-485 y 1-Wire. Cuenta también con 8 entradas digitales opto aisladas, 4 entradas *dry contact*, 8 salidas *open drain*, 4 entradas analógicas 0-10V y 2 salidas analógicas 0-10V.

También, como sistema basado en Raspberry Pi, cabe destacar el bajo consumo de energía y su alta eficiencia, características básicas para el proyecto.

3.2. DELTASOL

El siguiente dispositivo que vamos a desarrollar es el gestor de Resol: DeltaSol, este dispositivo de la empresa alemana es el encargado de controlar lo relativo a la generación energética del panel solar, sensores de temperatura, así como varios relees, que pueden conectarse a bombas de agua o a sistemas de refrigeración.

Los reguladores DeltaSol se utilizan para el control de velocidad de las bombas de alta eficiencia en instalaciones solares y de calefacción estándar pequeñas. Los reguladores incluyen una salida PWM y una entrada para sensores Grundfos Direct Sensor™ VFD, los cuales permiten realizar balances térmicos precisos.

Este dispositivo viene instalado con un software que posee 3 sistemas básicos seleccionables, y menú de puesta en marcha. Además de esto también posee una función anti hielo para el circuito solar, una limitación mínima y máxima del captador, y desconexión de seguridad del captador.

Tiene entrada para un captador y un acumulador, y está preparado para funcionar con un caudal fijo.

En cuanto a su protocolo de comunicación, utiliza un protocolo propio de la marca denominado Vbus, protocolo que introdujimos en la sección 2. ESTADO DEL ARTE.

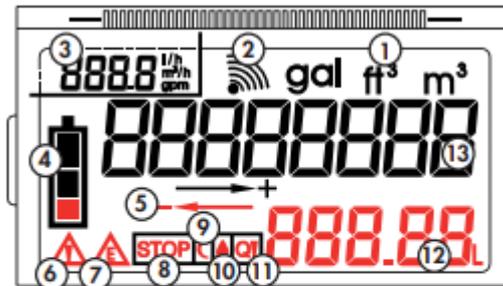
Este protocolo envía mensajes desde cada uno de los dispositivos solares a la red donde están conectados todos los dispositivos de la marca, con un encabezado en el que indica la dirección a la que se envía y el protocolo que rige el mensaje, dentro de tres posibilidades que tiene el Vbus. Heredado del protocolo de comunicación Modbus, el sistema Vbus diferencia entre maestros y esclavos, e incluye esta información en el encabezado del mensaje, tras este encabezado aparece la trama o *payload* donde se encuentra la información relativa a temperaturas, estados de relees, generación fotovoltaica etc...

3.3. CONTADOR CONTAZARA

El contador de Contazara se trata de un producto pensado para la gestión del agua, combinando electrónica avanzada y fiabilidad, es instalable en cualquier posición o ángulo, tiene una pérdida de carga baja, cuenta con un visor LCD donde muestra la información y posee telelectura, característica por la cual lo hemos elegido para el proyecto.

Información en LCD

1. Unidades de medida
2. Comunicación activada
3. Caudal instantáneo
4. Nivel de batería
5. Sentido del flujo del agua
6. Activación otras alarmas
7. Error interno
8. Alarma por defecto de consumo
9. Alarma de consumo nocturno



10. Alarma por detección de fugas
11. Alarma de superación de Q_{máx}
12. Consumo acumulado (l)
13. Consumo acumulado (m³)

Ilustración 12 (Información del Display de Contador)

La comunicación se produce a través de un protocolo que cuenta con un canal de petición de envío de datos y el canal por el que se envía la trama, también cuenta con un canal de tierra.

Para iniciar la comunicación debemos enviar una señal por el canal de petición que se mantenga alta 0.75 segundos y que una vez baje se mantenga en estado bajo otro mínimo de 2 segundos una vez finalice el envío antes de realizar una nueva petición, durante el envío de datos la señal de petición puede estar alta o baja indistintamente.

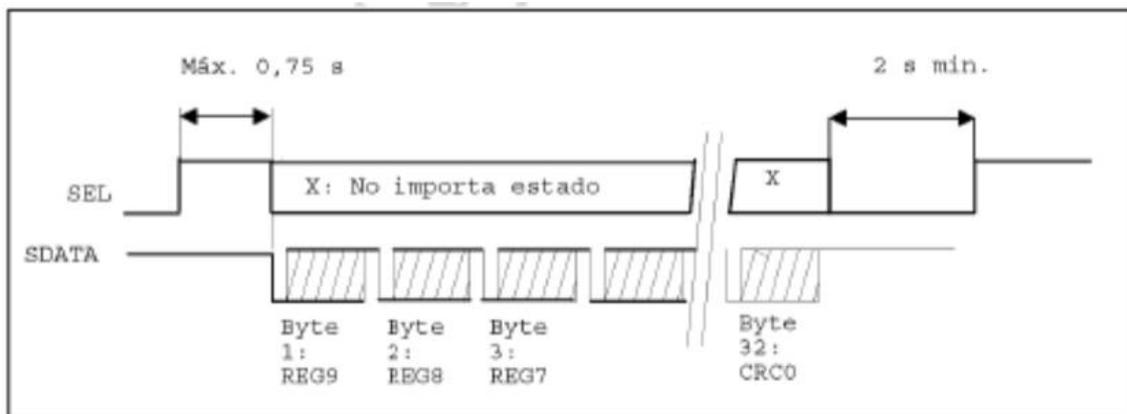


Ilustración 13 (Señales Sel y Sdata de Contazara)

La trama que envía está definida en la siguiente ilustración.

| <i>Campos de la TRAMA A+</i> | | | |
|------------------------------|-----------------------|-----------------------|---|
| Campo | Longitud | Obligatoriedad | Descripción |
| V | 1 | | Carácter "V" mayúscula (ASCII 56h) |
| Campo S | De 2 a 18 MSB primero | Obligatorio | Fabricante y número de serie. Debe ser el primer campo. |
| TAB | 1 | Obligatorio | Carácter tabulador (ASCII 09h), utilizado para separar campos. |
| Campo R | Sin especificar | Obligatorio | Lectura. Incluye valor, unidad, factor de potencia y unidad de tiempo. |
| Campo X | 4 | Obligatorio | Información de las capacidades de comunicación en tramas C. |
| Campo A | De 1 a 16 | Opcional | Código de diagnósticos y error, específico del fabricante, de 1 a 16 caracteres ASCII. Cada uno de ellos se forma a partir de un byte 0xxxxxb, al que se le suma 30h en el momento de la transmisión, donde cada x representa un bit. |
| Campo F | Sin especificar | Opcional | Versión, específico de cada fabricante. |
| Campo N | Sin especificar | Opcional | Identificador del modelo, en texto libre. |
| Campo Q | Sin especificar | Opcional | Estado del dispositivo. |
| Campo B | Sin especificar | Opcional | Identificador de cliente, usuario o N° Serie Ampliado. Opcional. |
| Campo J | Sin especificar | Opcional | Texto libre. |
| Campo C | 4 MSB primero | Obligatorio | CRC16 de la trama, sin incluir el propio campo C, ni el primer carácter "C" de este campo. El primer carácter de la trama "V" queda incluido en el cálculo. Este campo debe ser el último. |

Ilustración 14 (Trama Contazara)

3.4. SONDA DS18B20

Como sonda térmica usaremos el modelo DS18B20, el protocolo de comunicación será sobre 1-Wire.

Este protocolo está basado en un bus, un maestro y varios esclavos que se conectan en una sola línea de datos, aunque en nuestro caso solo utilizaremos uno.

La manera de comunicarnos con este protocolo está basada en los tiempos que cada dispositivo baja a 0 la línea de datos. El maestro envía un 1 lógico si la baja entre 1 y 15 microsegundos y envía un 0 lógico bajándola durante 60 microsegundos. Los esclavos se comunican tras una petición y actúan de manera similar, para enviar un 1 mantienen la señal a 5 voltios, o la bajan a 0 durante 60 microsegundos para enviar un 0. Los datos son empaquetados siempre en grupos de 8 bits. Antes de cada comunicación debe reiniciarse el bus lo que implica mantener la señal en 0 voltios durante 480 microsegundos.

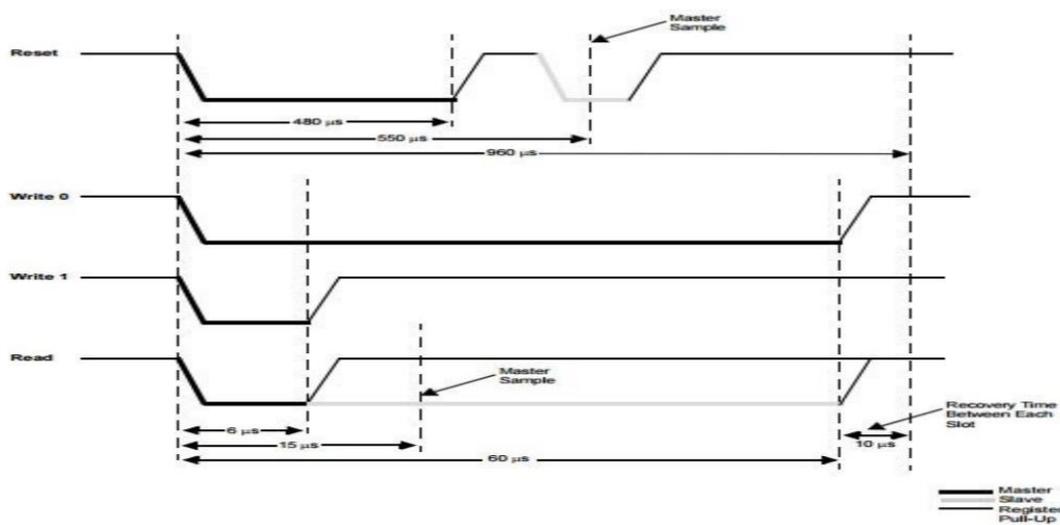


Ilustración 15 (Tiempos del protocolo 1-wire)

3.5. INVERSOR KOSTAL

Para el inversor necesario en la instalación la opción elegida ha sido un producto de la marca Kostal, en concreto el modelo PIKO MP plus. Se trata de un inversor monofásico adecuado para instalaciones pequeñas, por lo tanto, adecuado para nuestro proyecto. Cuenta también con un amplio rango de tensión de entrada. También tiene la posibilidad de ser ampliado para usarse como acumulador de energía sin necesidad de inversor adicional.

Otras características destacables son la posibilidad de integrarle contadores de energía, la capacidad del control dinámico de la potencia activa las 24h mediante un *webserver* interno del cual nos aprovecharemos para extraer la información que necesitamos. Su instalación es muy sencilla, tiene alimentación monofásica y no es necesario abrir el equipo, tiene un menú que guía durante el primero uso para su configuración y también tiene una protección frente al polvo y el agua para poder ser utilizado en exteriores (protección IP 65).

La comunicación, como hemos comentado, la haremos a través del *webserver* que el sistema nos ofrece, nos descargaremos la información y la extraeremos mediante *Web Scraping*.

3.6. THINGSBOARD Y EL PROTOCOLO MQTT

La comunicación entre el dispositivo Pigeon, que recibe la información de los otros dispositivos cada uno de una manera diferente, y la plataforma ThingsBoard se rige por un protocolo llamado MQTT. Por ello en este apartado describiremos en que consiste este protocolo y como lo utilizaremos nosotros. Mencionar también que este protocolo se ha convertido en uno de los más usados dentro de los sistemas que trabajan con la tecnología IoT, entre otras cosas por su sencillez y ligereza, lo que permite que dispositivos con baja potencia sean capaces de funcionar con él.

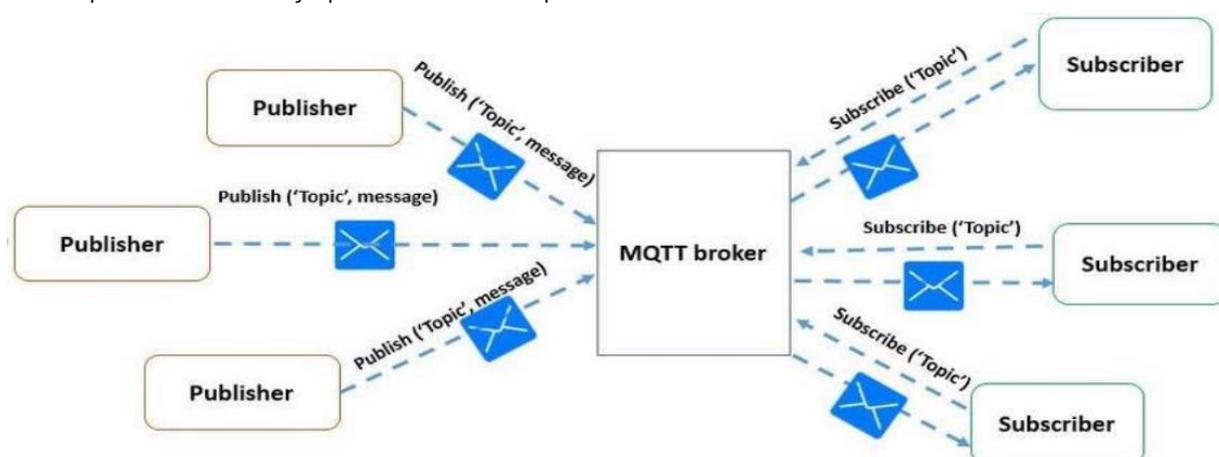


Ilustración 16 (Esquema MQTT)

El protocolo MQTT se basa en la existencia de temas o *Topics* a los que cada dispositivo que forma el sistema puede estar suscrito o no y también puede tener la capacidad de publicar mensajes en ese *Topic*. Estos temas tienen una estructura jerárquica en la que unos temas están dentro de otros, por ejemplo, dentro del tema “garaje” pueden existir dos temas que se llamen, “luz” y “puerta”. Cualquier dispositivo suscrito al tema “garaje” será capaz de recibir los mensajes publicados en “luz” y “puerta” pero un dispositivo suscrito a “luz” tan solo podrá recibir mensajes publicados en este *Topic*.

Aquí observamos la primera diferencia con otros protocolos como el HTTP donde cada información requiere que se abra y se cierre el canal por el que se transmite mientras que en MQTT el canal permanece abierto y se reutiliza de manera constante, lo que favorece, comentado anteriormente, su sencillez y su ligereza.

En nuestro caso, ThingsBoard ofrece la posibilidad de publicar la información comunicándonos desde el Pigeon al servidor con este protocolo. Así pues, nosotros publicaremos toda la información en un *Topic* al que el servidor está suscrito y de esta forma podremos transmitir a la plataforma los datos que recibimos en el Pigeon del resto de dispositivos.

Como comentaremos más adelante en el apartado en el que hablamos de posibles vías de trabajo futuro, este protocolo nos permitiría desarrollar comunicación en los dos sentidos, de manera que podríamos controlar actuadores desde la misma plataforma de ThingsBoard.

4.DESARROLLO DEL PROYECTO

Nuestro objetivo principal es desarrollar un gestor energético capaz de controlar y monitorizar cada una de las cargas de las que dispongamos en una casa, un edificio o un almacén.

A continuación, describiremos detalladamente cada una de las partes que integra el proyecto, repasando como hemos desarrollado cada elemento y su funcionamiento completo, incluyendo la subida de datos a ThingsBoard.

Crearemos también todo lo necesario para que ThingsBoard muestre los datos de una manera intuitiva y sencilla.

Finalmente desarrollaremos el trabajo en conjunto para producir la subida de todos los datos disponibles y la configuración del panel final.

4.2. INSTALACION Y RECOLECCION DE DATOS

En este apartado trataremos cómo hemos instalado y desarrollado cada uno de los dispositivos encargados de la recolección de datos, desde la conexión electrónica con el Pigeon hasta la descripción de la forma de funcionar del código.

Incluiremos en cada caso un esquema eléctrico de la conexión y un diagrama de flujo que nos servirá de soporte a la hora de explicar el funcionamiento de cada herramienta.

Cada uno de los dispositivos que entran en comunicación tiene asociado un script que hemos programado con Python 3 que se ejecutaran en el sistema operativo Linux que el propio dispositivo Pigeon lleva instalado.

La conexión con el dispositivo la hacemos a través de una conexión Secure SHell o SSH, logrando de esta manera que podemos acceder de manera remota desde cualquier ordenador. El protocolo SSH establece conexiones seguras entre los dos sistemas, utilizando la arquitectura cliente/servidor y se encarga de encriptar todas las sesiones, así que resulta imposible que alguien pueda acceder a contraseñas o datos sensibles.

Con esta conexión, y a partir de la consola podemos manejar y proceder con la programación del sistema. La idea principal, como ya hemos comentado, es programar un script para cada dispositivo y finalmente otro que ira ejecutando el resto de manera secuencial, depurando si existiera algún error y realizando alguna tarea de procesamiento de datos extra que fuera necesaria.

Este el ultimo script lo asignaremos a un servicio de la propia Pigeon que se ejecute en el momento en que el dispositivo se encienda, de esta manera ante cualquier reinicio o corte de luz, el sistema seguirá monitorizando en el momento que se reestablezca sin necesidad de ninguna acción externa.

4.2.1. GESTOR DELTASOL CS4

En primer lugar, vamos a tratar el gestor de Resol que se trata del componente que más información aporta del sistema, reportando datos de bombas de agua, múltiples sensores de temperatura, horas de funcionamiento o generación de energía por parte del panel solar.

Para la conexión con el dispositivo Pigeon necesitamos un adaptador que la propia empresa facilita. Este adaptador VBus/USB permite, como su nombre indica, conectar los datos procedentes del VBus a un USB del dispositivo.

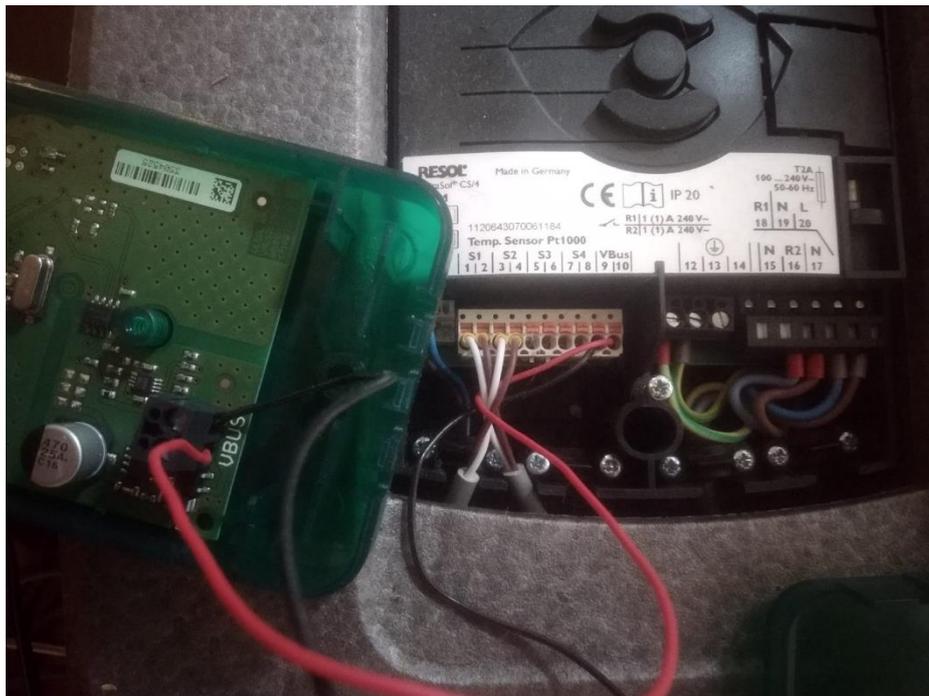


Ilustración 17 (Conexión Vbus/USB)

De esta forma la conexión es bastante sencilla, sin necesidad de añadir ningún elemento extra.

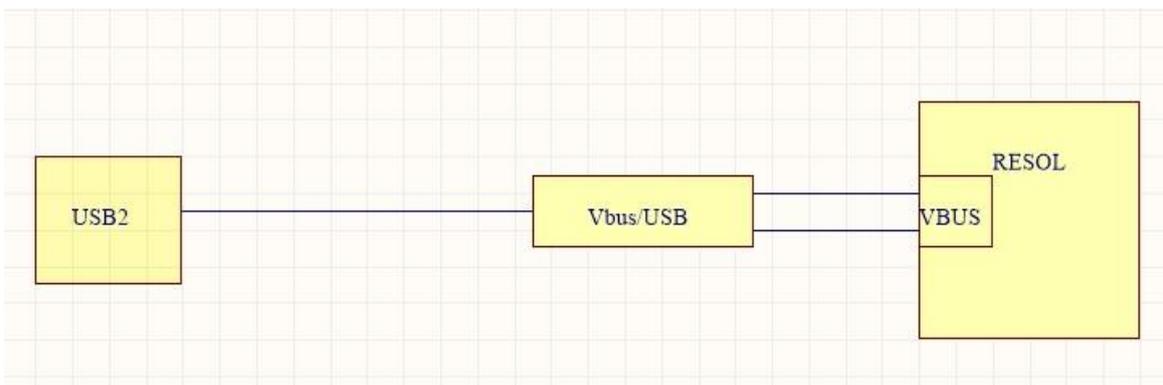


Ilustración 18 (Conexión Resol-Pigeon)

Una vez realizada la conexión queremos recibir los datos y para ello debemos conocer la forma en la que el gestor nos envía la información. El dispositivo manda periódicamente tres mensajes a través del Vbus, en ellos, además de la información anteriormente descrita, nos informa del identificador de quien manda el mensaje, el identificador del objetivo de dicho mensaje y el protocolo que lo rige. De estos tres mensajes a nosotros nos interesa el que posee la información que deseamos mostrar al usuario, por lo que en primer lugar debemos recibir los datos, agruparlos en los tres mensajes e identificar el mensaje que nos interesa analizar para la extracción de la información.

Antes de empezar a trabajar en el código, utilizamos el software que la propia marca nos facilita para identificar la forma que tendrá la trama que recibiremos y la distribución de las distintas informaciones que contiene. Con esto, confirmamos los tres mensajes, uno de ellos más corto, con protocolo distinto y dos más largos, con varios datos duplicados. Analizando estos mensajes seleccionamos aquel que nos transmite toda la información que necesitamos mostrar, que en este caso será aquel que se rija por el protocolo 1 y que tenga como destinatario el software de Resol que muestra por pantalla los datos.

Una vez tenemos nuestros objetivos claros, nos disponemos a recibir el mensaje, transformar la trama en bruto en datos con el formato adecuado para poder trabajar con ellos, separarlo en los tres mensajes individuales, seleccionar el que nos interesa y preparar nuevamente los datos en un formato que podamos enviar a ThingsBoard.

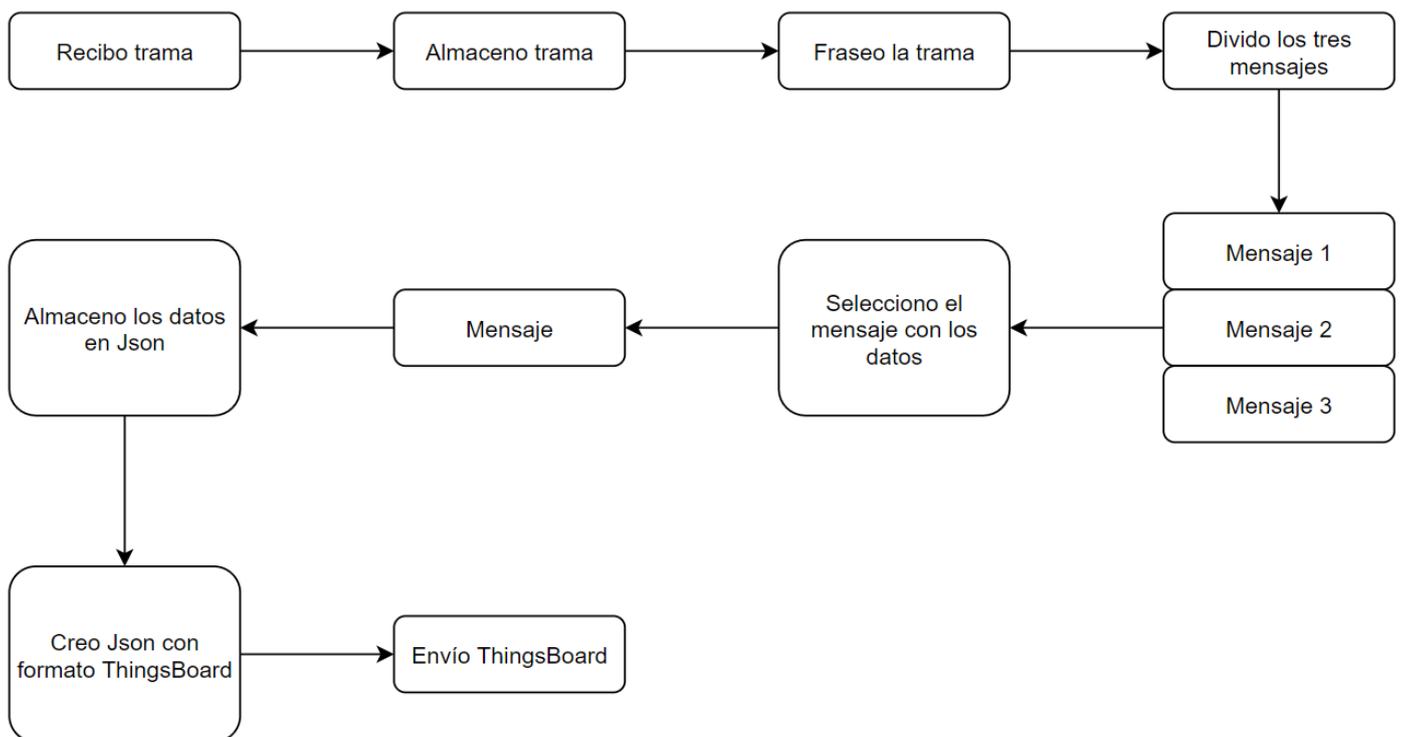


Ilustración 19 (Diagrama de Flujo: Gestor de Resol)

La recepción de datos varía dependiendo del modo de conexión con el gestor, en nuestro caso lo hacemos mediante el adaptador de VBus/USB a diferencia de gran parte de la documentación existente que los recibe a través de un adaptador Vbus/LAN. La mayor diferencia entre ambos métodos reside en que en nuestro sistema los bytes llegan de uno en uno, por lo que tenemos que leerlos de este modo. Por lo demás, ambos métodos son bastante similares.

Una vez tenemos la trama completa pasamos a transformarla en datos, que incluye pasos como transformar a String los bytes u obtener los valores numéricos de los bytes que están en complemento a dos.

Conseguido este paso, ya tenemos los datos de manera que podemos acceder a ellos de forma sencilla. Recordemos que aún tenemos los tres mensajes juntos, por lo que en este momento pasamos a separarlos en los tres que sabemos que existen, esto lo hacemos gracias a que todos los mensajes empiezan de la misma manera, identificando así el inicio de un mensaje individual.

Separados ya los mensajes pasamos a localizar el que nos interesa por contener los datos y a proceder con la extracción de los mismos.

Para el almacenamiento de estos datos vamos a utilizar un Json construido previamente, en el que hemos incluido el nombre de cada dato, su posición en la trama, su tamaño bits, el factor por el que debemos multiplicar para obtener su valor real y las unidades en las que está representado. De esta forma conseguimos que el programa acuda al Json a consultar esta información y sea capaz de extraer la información del *payload* de manera automática.

Gracias al modo en el que extraemos los datos, este proceso se vuelve automático y cuenta con otra gran ventaja, la adaptabilidad a cualquier otro gestor de Resol, ya que cambiando tan solo el Json al correspondiente al modelo que se utilice, el resto del proceso es idéntico.

Una vez tenemos este Json completado el único paso que nos falta es el envío a Thingsboard de la información.

Para este envío tenemos que tener presente que el formato necesario para que dicha información sea legible por ThingsBoard es muy concreto, y es el siguiente:

```
{"temperature": "22"}
```



```
{
  "offset": "6",
  "name": "Sensor Temperatura 4",
  "bitSize": "15",
  "factor": "0.1",
  "unit": " °C"
},
{
  "offset": "8",
  "name": "Velocidad de la bomba 1",
  "bitSize": "7",
  "factor": "1",
  "unit": " %"
},
{
  "offset": "10",
  "name": "Horas de funcionamiento Bomba 1",
  "bitSize": "15",
  "factor": "1",
  "unit": " h"
},
}
```

Ilustración 20 (Json)

Es decir, debemos iniciar el mensaje con una llave, el nombre del dato que enviamos entrecomillado, dos puntos, el valor del dato (preferiblemente entrecomillado) y cerrar la llave. Para esto utilizamos otro proceso automático, creando un Json en el que vamos construyendo la información contenida en el anterior Json, pero con el formato adecuado. Nótese la necesidad de la existencia del primero para automatizar el proceso de la extracción de los datos del mensaje.

Una vez tenemos este segundo Json completo lo publicamos en ThingsBoard

4.2.2. CONTADOR DE AGUA CONTAZARA

Aquí desarrollaremos la manera en que conectamos el contador al dispositivo Pigeon y como extraemos la lectura del mismo.

Contazara es una empresa aragonesa dedicada a la gestión eficiente del agua, especializada en contadores electrónicos capaces de transmitir la información de manera instantánea. Para nuestro proyecto usaremos su modelo CZ3000, el cual conectaremos mediante un adaptador al puerto USB de nuestro dispositivo, también necesitaremos conectar una de las salidas digitales al contador, que será la encargada de la petición de envío de la información. Además, el canal con la información requiere de una resistencia conectada a una fuente de 5V, nosotros usaremos la que la propia Pigeon nos facilita. Esta resistencia puede variar, pero el valor óptimo es de 470 ohmios. Por último debemos tener en cuenta que las salidas digitales del Pigeon son *OpenDrain* (están conectadas a la base de un transistor) por lo que debemos añadir una resistencia más y conectar esta por un lado a la fuente de 5V y por otro a la salida del Pigeon y la señal asociada a la petición de envío de información del contador.

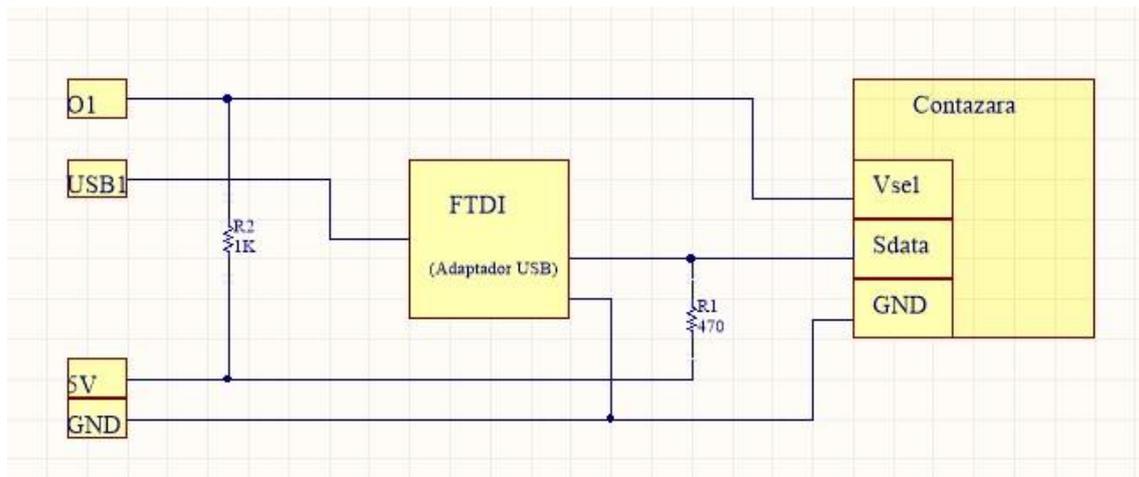


Ilustración 21 (Conexión Contazara-Pigeon)

Una vez completamos la conexión ya estamos preparados para recibir la lectura del contador.

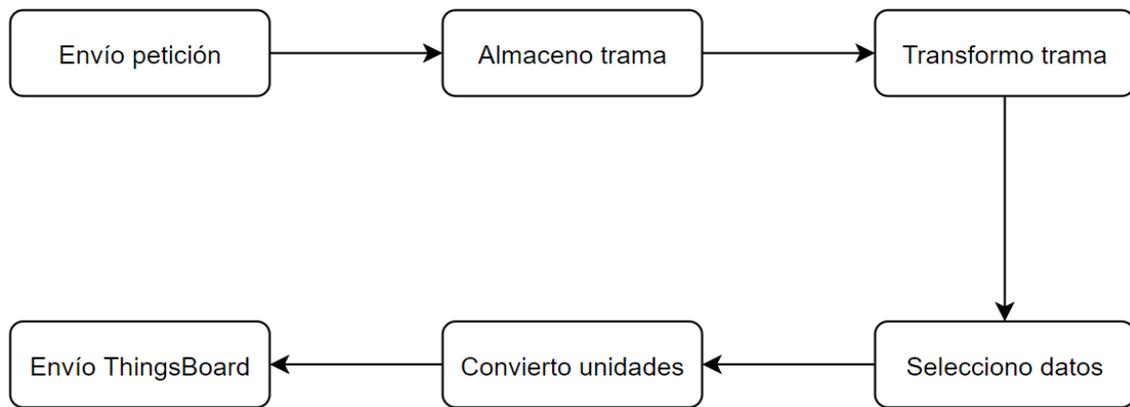


Ilustración 22 (Diagrama de Flujo: Contazara)

Como hemos dicho antes, el contador no manda la información de manera constante si no que la envía como respuesta a una señal mandada por el canal Vsel, y lo hace siempre que recibe la parte alta de un pulso. Por ello lo primero que debemos hacer es controlar ese pulso, respetando los tiempos que el fabricante nos indica.

Una vez enviamos el pulso y recibimos la trama la almacenamos en un array de bytes. Algo que tenemos que tener en cuenta en este momento es que en la trama que recibimos los bytes tienen un formato *8 bits 1 stop* y nos interesa transformarlos a *7 bits 2 stop*, por lo que el siguiente paso es realizar dicha transformación.

Completada la transformación el siguiente paso consiste en localizar el tramo de la traza que contiene la lectura y las unidades en las que esta expresada.

Para ello conocemos que la lectura comienza con los caracteres “\tRC” por lo que buscamos dicha secuencia y, conociendo que su longitud, almacenamos los bytes posteriores necesarios. También conocemos los bytes que nos informan de la unidad utilizada en la medida, por lo que buscamos dichos bytes y procedemos, si es necesario, a multiplicar la medida por el valor adecuado para obtener la unidad que deseamos, en nuestro caso litros.



Ilustración 23 (Montaje para la realización de pruebas en el ordenador)

Finalmente, ya con la medida tomada y en la unidad correcta, construimos el mensaje en el formato que ThingsBoard requiere y procedemos al envío de la lectura.

4.2.3. SENSOR DE TEMPERATURA

Además de los sensores de temperatura que el gestor de Resol es capaz de controlar, nos parecía necesario ser capaces de monitorizar la temperatura de cualquier habitación, cámara o exterior que existiera en la instalación, ya que saber la temperatura en cualquiera de estos casos es imprescindible para un buen control de la energía y el gasto de recursos de cualquier edificio.

Para ello vamos a utilizar una sonda de temperatura que conectaremos mediante una conexión 1-Wire al dispositivo, aprovechando que el mismo tiene una entrada compatible., en nuestro caso una sonda DS18B20, común en el mercado.

Para la conexión con el dispositivo tan solo necesitamos una resistencia *Pull-up* de 4K7 ohmios, indicado por el fabricante, y conectar cada cable a su correspondiente entrada, indicada en el propio Pigeon.

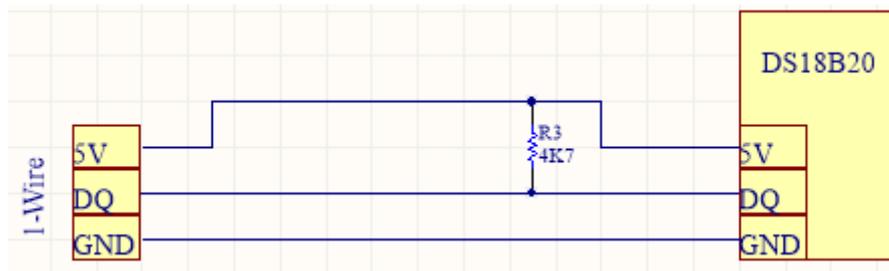


Ilustración 24 (Conexión DS18B20-Pigeon)

Con la conexión realizada la toma de datos es bastante sencilla, tan solo tenemos que abrir un canal con la ubicación en la que el dispositivo registra la temperatura, realizar una lectura y darle el formato que requerimos para acabar enviándola a ThingsBoard.

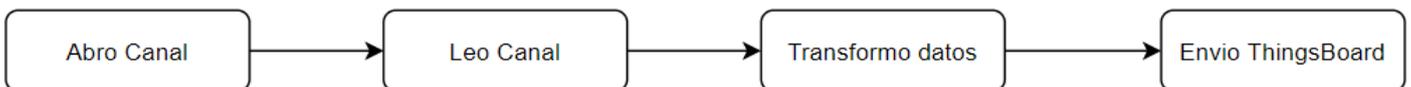


Ilustración 25 (Diagrama de Flujo: Sonda DS1820)

4.2.4. INVERSOR KOSTAL

Asociada a toda generación fotoeléctrica aparece la figura necesaria del inversor, pieza que resulta fundamental para el aprovechamiento de la energía solar. En nuestro caso vamos a utilizar un inversor de la marca Kostal PIKO MP plus, del cual nos interesa recabar cuanta más información con el fin de controlar el correcto funcionamiento de toda la instalación.

La manera en la que abordaremos la recolección de datos de este dispositivo va a tener un enfoque distinto. Aprovechando que el propio inversor es capaz de crear una web dentro de la red local en la que muestra los datos que nos interesan, vamos a descargar la información directamente de esa web, con un proceso que suele recibir el nombre de *Web Scraping*.

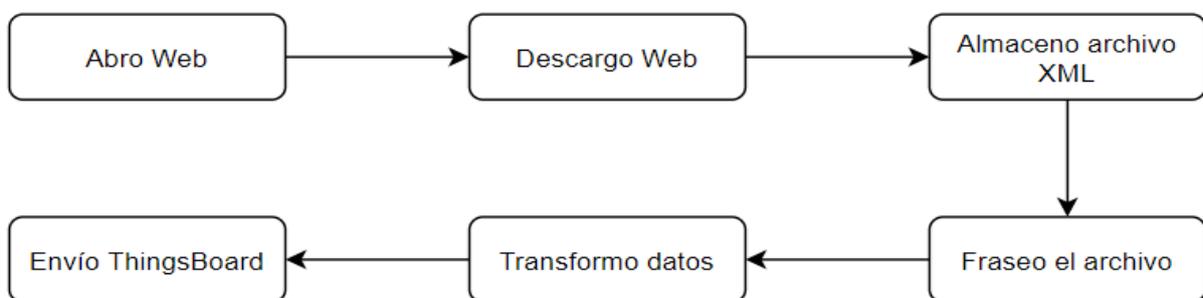


Ilustración 26 (Diagrama de Flujo: Inversor Kostal)

Para ello accederemos a la parte donde se almacenan los datos, en este caso alojada en <http://localhost:8000/measurements.xml>, y descargaremos el texto, almacenándolo en un archivo XML.

Una vez descargado, debemos realizar el fraseo para extraer los datos. Nuevamente automatizamos el proceso de tal manera que el mismo sistema podría ser aplicado para funcionar con archivos XML similares que pudiera generar otros modelos de la misma marca. De esta manera los cambios necesarios para funcionar con otros dispositivos serían mínimos.

Una vez tenemos los datos de forma que el acceso a ellos es sencillo, rellenamos un Json con el formato adecuado para que ThingsBoard tenga la capacidad de almacenarlos y los enviamos a la plataforma.

4.3. PANEL DE THINGSBOARD

En esta sección vamos a desarrollar el panel que mostrara la información que recogemos de cada uno de los dispositivos disponibles. La función que debe cumplir este panel es doble, por una parte, debe ser intuitivo, de manera que en un vistazo rápido seamos capaces de formarnos una idea clara de la situación actual de la instalación,

asegurándonos de que todo está funcionando de una manera correcta y esperada. Por otro lado, el panel debe mostrar suficiente información y suficientemente detallada como para facilitarnos la tarea de observar en que momentos la instalación gasta más o requiere más cantidad de recursos, de forma que, con la información que nos pueda facilitar las distintas empresas suministradoras, seamos capaces de seleccionar la opción más favorable, económica y energéticamente hablando.

Para esto lo primero que haremos será un diagrama de bloques en el que expresaremos, acercándonos lo máximo posible a la solución final, qué deseamos mostrar y de qué manera. Elegiremos de qué modo expresaremos cada gasto o consumo, o información del sistema.

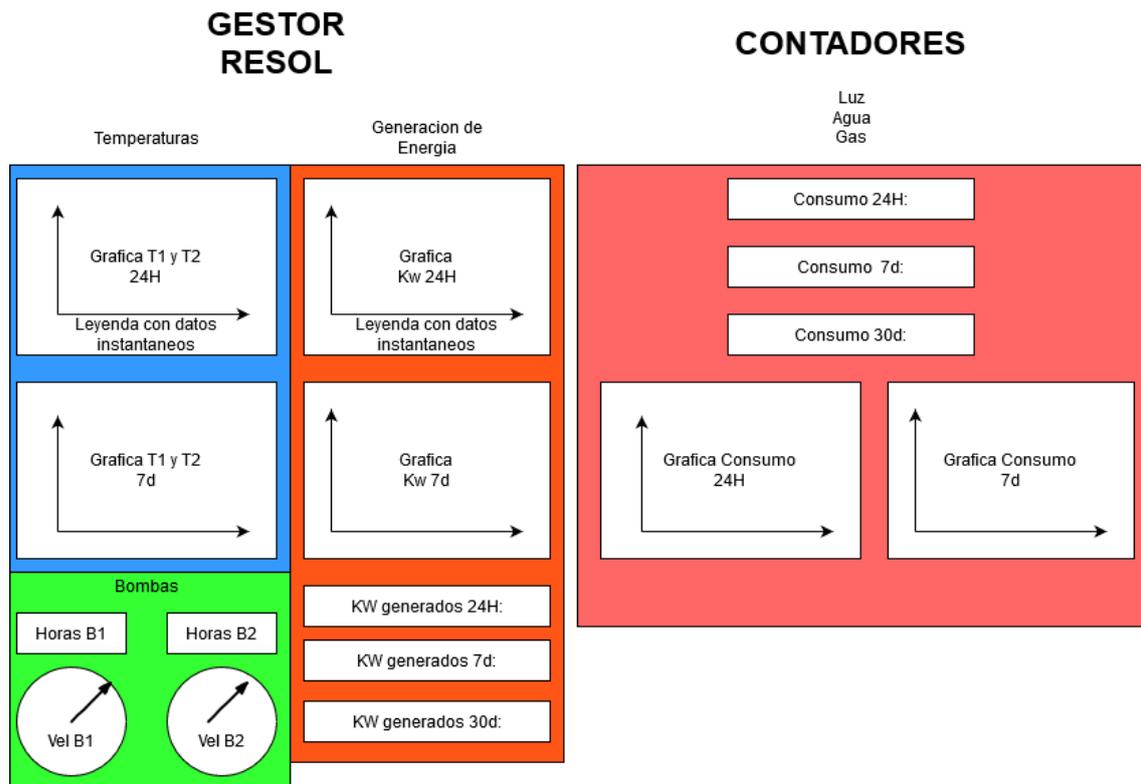


Ilustración 27 (Diagrama de Bloques: Panel Thingsboard)

4.3.1. SELECCION VARIABLES

En este momento lo que debemos hacer es selección aquellas variables que deseamos mostrar en las gráficas y asignarlas a los distintos *Widgets* que nos proporciona el propio Thingsboard. En algunos casos nos interesara mostrar la variable sin modificar y en otro quizás nos interese más mostrar un acumulado o una nueva variable que podamos extraer de la modificación de la que recibimos por parte del dispositivo recolector correspondiente.

Cuando la cantidad de datos que recibimos es ingente la selección de los datos que deseamos mostrar es importante. Nuevamente tenemos que tener equilibrio entre ser suficientemente claros sin dejar de mostrar todos los datos que sean interesantes para conocer el estado de la instalación en cada momento.

4.3.2. SELECCIÓN DE ELEMENTOS

Ahora es importante la selección de los elementos que mostraran cada información de la que disponemos, elegir bien nos permitirá que el panel final cumpla las dos funciones que antes hemos mencionado, claridad sin renunciar al detalle.

Por ejemplo, elegiremos una gráfica para mostrar el consumo de un determinado recurso de forma que podamos observar dicho consumo en diferentes etapas temporales; diario o semanal. Para mostrar temperaturas utilizaremos también gráficas, añadiendo que en su leyenda puedan mostrar los datos de ese instante. En el caso de las velocidades de las distintas bombas de la instalación, que son controladas por el gestor de Resol, usaremos indicadores con forma de cuentakilómetros. Aplicando estos criterios de claridad sin dejar de mostrar toda la información elegiremos cada uno de los elementos que conformaran el panel final.

4.3.3. CREACIÓN DEL PANEL

Una vez hemos seleccionado las variables y como mostrarlas debemos realizar el proceso para hacerlo posible. Lo primero debemos crear un dispositivo, que será el que reciba la información que manda el Pigeon en forma de telemetría. Lo definiremos como un dispositivo de entrada y al crearlos se le asignaran tanto una ID como un *token* de acceso que necesitaremos incluir en la conexión MQTT entre el Pigeon y Thingsboard.

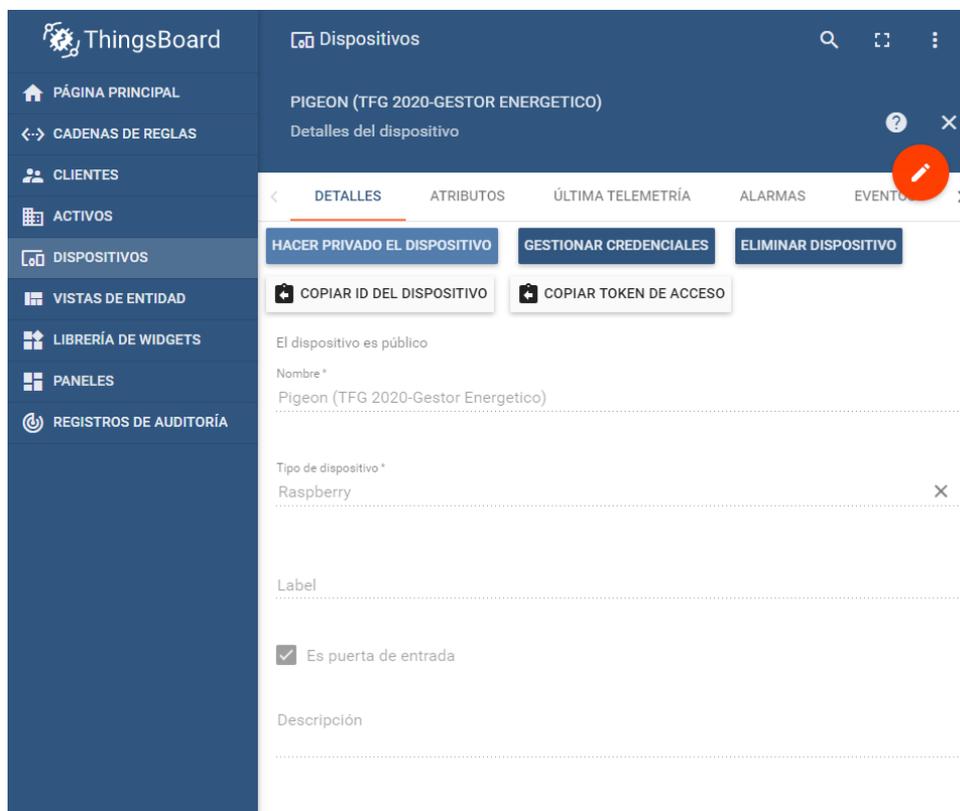


Ilustración 28 (Creación de dispositivo en ThingsBoard)

Una vez hemos creado el dispositivo e incluido el *token* en el envío de datos, empezaremos a recibir las telemetrías en la pestaña correspondiente, donde podremos observar como ThingsBoard va recibiendo los datos. En esta misma ventana tenemos la posibilidad de mostrar la telemétrica en un *widget*, aunque nosotros lo haremos de otro modo que nos permita mostrar en un solo grafico distintas variables.

Última telemetría 🔍

| <input type="checkbox"/> | Hora de la última actualización | Clave ↑ | Valor |
|--------------------------|---------------------------------|--|-------|
| <input type="checkbox"/> | 2020-06-20 23:00:19 | AC_Current | None |
| <input type="checkbox"/> | 2020-06-20 23:00:22 | AC_Frequency | 49.94 |
| <input type="checkbox"/> | 2020-06-20 23:00:20 | AC_Power | None |
| <input type="checkbox"/> | 2020-06-20 23:00:21 | AC_Power_fast | None |
| <input type="checkbox"/> | 2020-06-20 23:00:18 | AC_Voltage | 234.3 |

Page: 1 ▾ Rows per page: 5 ▾ 1 - 5 of 39 < >

Ilustración 29 (Ultima telemetría)

Una vez confirmamos que recibiendo la información vamos a crear el panel donde mostraremos los *widgets* incluyendo la información que deseemos. Para ello debemos ir a la sección de paneles y crear uno. Una vez creado debemos añadir el *widget*.

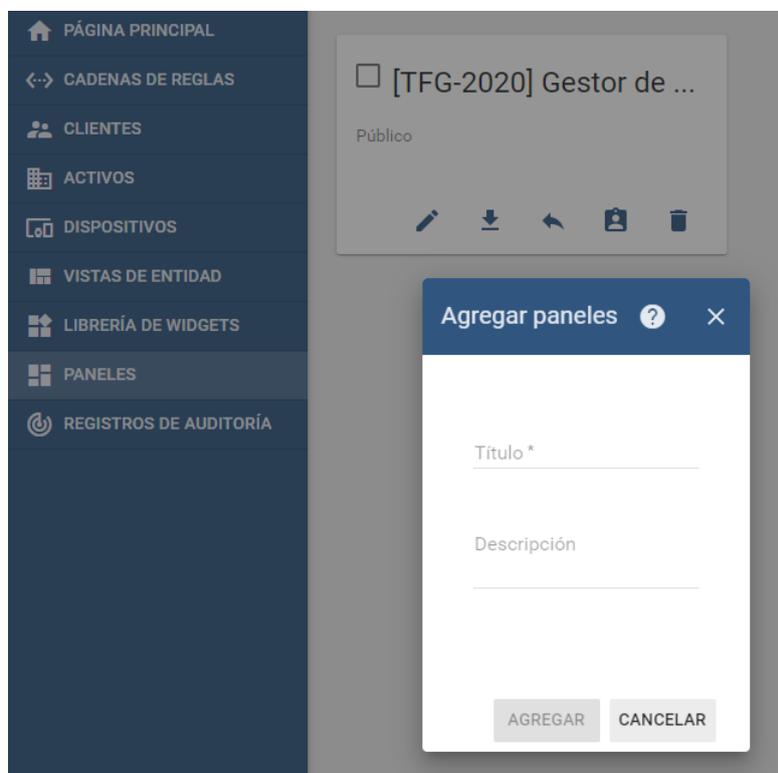


Ilustración 30 (Creación de panel)

Lo primero que debemos hacer al crear el *widget* es seleccionar el tipo de grafico que queremos utilizar, una vez seleccionado añadiremos los datos de origen que extraeremos de la telemétrica del dispositivo que hemos creado anteriormente.



Ilustración 31 (Creación de widget)

Por ultimo podemos elegir mostrar la variable tal cual no llega en la telemetría o modificarla mediante una función de postprocesamiento, en la que podemos tanto mostrar la variable en otras unidades por ejemplo o calcular una nueva variable, como puede ser la energía a través de la potencia o la temperatura del colector.

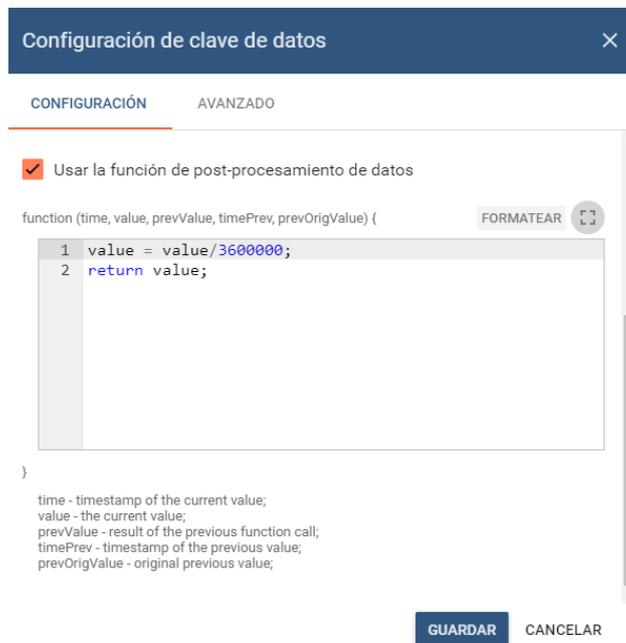


Ilustración 32 (Postprocesamiento de variables)

Ahora solo nos falta crear el resto de *widgets* y colocarnos en el panel como deseemos, guardar el panel y ya podremos empezar a monitorizar nuestro sistema.

5. INSTALACION REAL

Para poder probar nuestro proyecto hemos tenido la oportunidad de probarlo en una instalación real que cuenta con un módulo de Resol que controla la generación eléctrica de un panel fotovoltaico que se utiliza para calentar el agua de un acumulador de 150 litros. Además, contamos con un el correspondiente inversor y una sonda que nos permite medir la temperatura ambiente.

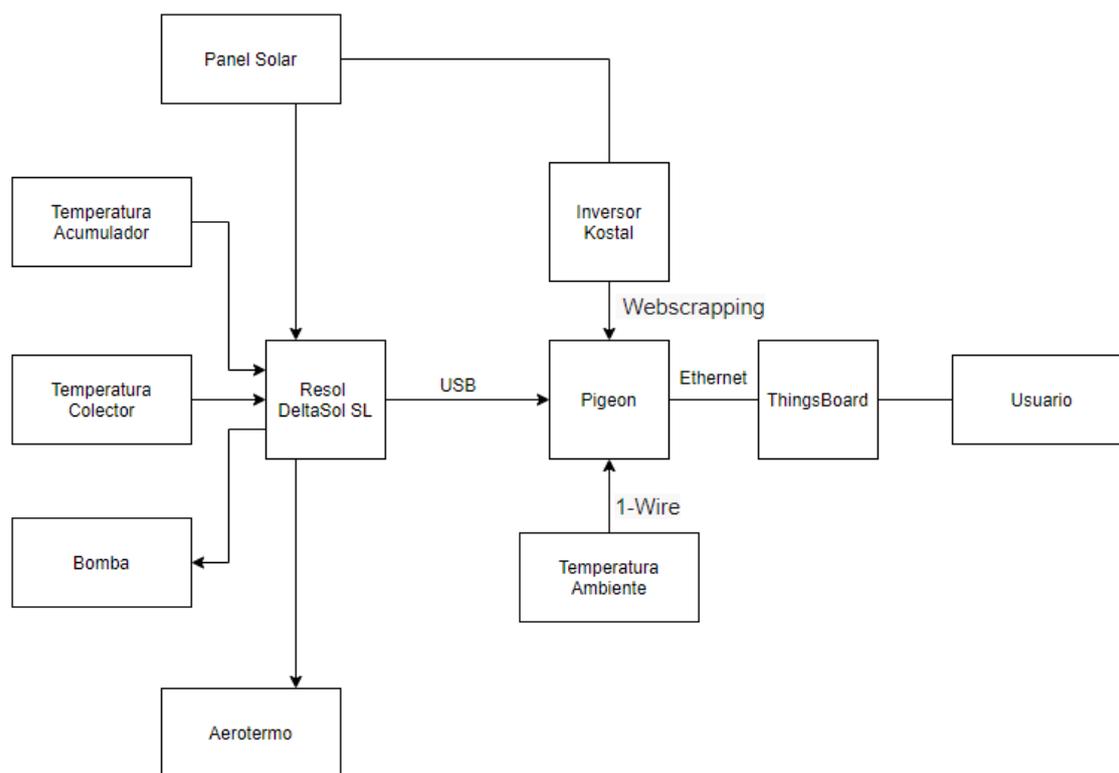


Ilustración 33 (Diagrama de Bloques: Instalación Real)

En esta instalación el gestor de Resol es el encargado de controlar dos sensores de temperatura, correspondientes a la temperatura del colector y del acumulador, una bomba de agua, la energía generada por el panel y un aerotermo encargado de enfriar las placas cuando alcanzan una determinada temperatura.

Como novedad con respecto a lo programado en el laboratorio (y más tarde por causa de la COVID en mi casa) aparece que el gestor se trata de otro modelo, que genera mayor cantidad de datos y con un formato de mensaje un tanto diferente.

La mayor diferencia entre ambos gestores está en que en este segundo la trama con en la que recibimos los datos solo cuenta con dos mensajes, pero en este acaso ambos nos reportan información interesante, uno de ellos nos informa de la cantidad de energía generada de manera instantánea, en las últimas 24 horas y en la última semana. El otro tiene un formato muy similar al que utilizábamos en el modelo del laboratorio, pero con un numero de datos significativamente mayor.

Gracias a la forma en la que afrontamos la extracción de datos en el laboratorio, procurando que el proceso fuera lo más automático y adaptable posible esto no provocó contratiempos considerables. Como comentamos en el apartado donde explicábamos el funcionamiento del gestor de Resol, con tan solo modificar el Json en el que almacenábamos los datos el proceso en el mensaje que es similar al que trabajábamos en el laboratorio se ejecuta sin problemas. Para tener en cuenta el otro mensaje tan solo tenemos que añadir una nueva criba que nos seleccione dicho mensaje y aplicar el mismo proceso que al anterior.

Una vez realizados estos cambios, la subida y tratamiento de datos en ThingsBoard se realiza de la misma manera que describimos en el apartado correspondiente.

En cuanto al inversor y la sonda de temperatura, el funcionamiento es igual a lo expuesto en los apartados anteriores.

El panel de ThingsBoard de esta instalación que vamos a desgranar a continuación se puede consultar desde cualquier dispositivo con conexión a internet y que posea un navegador en el siguiente enlace: <http://tfg.howlab.es:8080/dashboard/39e18490-b029-11ea-bcce-87a5d2a752be?publicId=3d73eef0-b2ee-11ea-bcce-87a5d2a752be>

5.1. TEMPERATURAS Y GESTION DEL AGUA

Una parte importante del control de la instalación es tener monitorizada la gestión relativa al agua y las temperaturas que están relacionadas con ella. Para ello vamos a prestar atención a diferentes valores que nos reportan los distintos dispositivos y mostrarlos en la plataforma Thingsboard de manera que podamos entender de un solo vistazo el estado de la instalación en lo relativo a esta parte.

Empezaremos por las temperaturas de las distintas partes que se ven involucradas en el proceso. Por un lado, tenemos la temperatura que se da en el colector, la parte encargada de calentar el agua aprovechando la energía solar, esta parte será la que mayor temperatura alcance durante el funcionamiento de la instalación y se enfriará hasta alcanzar la temperatura ambiente cuando el sistema este desconectado. En este punto es importante saber que cuando esta temperatura alcanza los 70°C entra en funcionamiento un sistema de refrigeración que consiste en un aerotermo, así que, para asegurarnos el correcto funcionamiento del mismo, monitorizaremos en esta parte cuando entra en funcionamiento. Además, controlaremos también cuando la bomba de agua se conecta para conocer cuando la instalación se ha puesto en marcha.

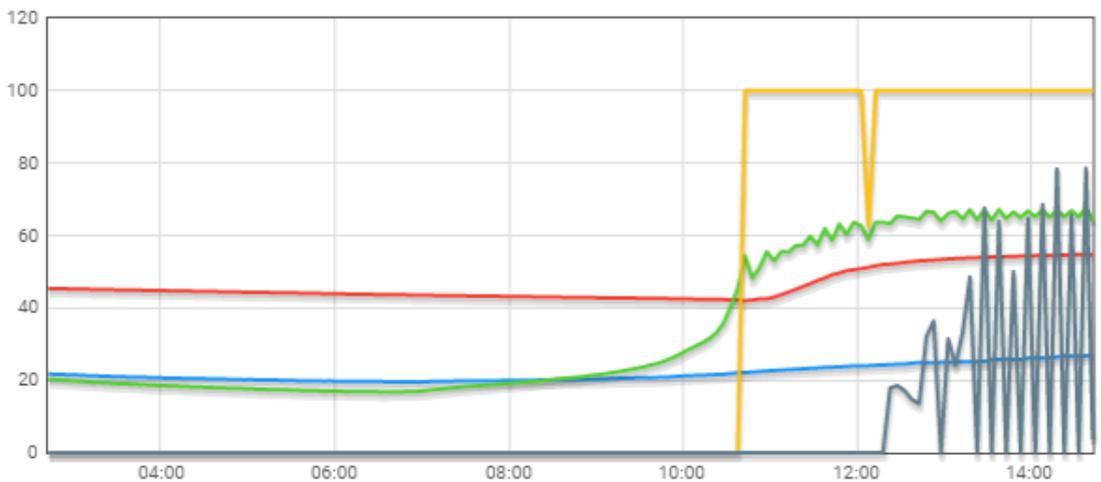
Es también imprescindible conocer la temperatura del acumulador, el lugar donde los 150 litros de la instalación se almacenan, el cambio de temperatura de este acumulador nos transmite la cantidad de energía que estamos aprovechando cuando la instalación funciona y la que perdemos cuando se apaga.

Otra temperatura importante es la que se está dando en el ambiente de la instalación. Como es evidente, no es lo mismo trabajar con una temperatura de 30°C que con una de 10°C, por lo que controlaremos esta con la sonda de la que disponemos.

Resumiendo, en este apartado necesitamos ver las temperaturas del colector, del acumulador y la del ambiente, también cuando entran en funcionamiento tanto la bomba como el aerotermo. Así pues, nos resulta la siguiente gráfica:

Temperaturas

🕒 Tiempo real - último(s) 12 horas



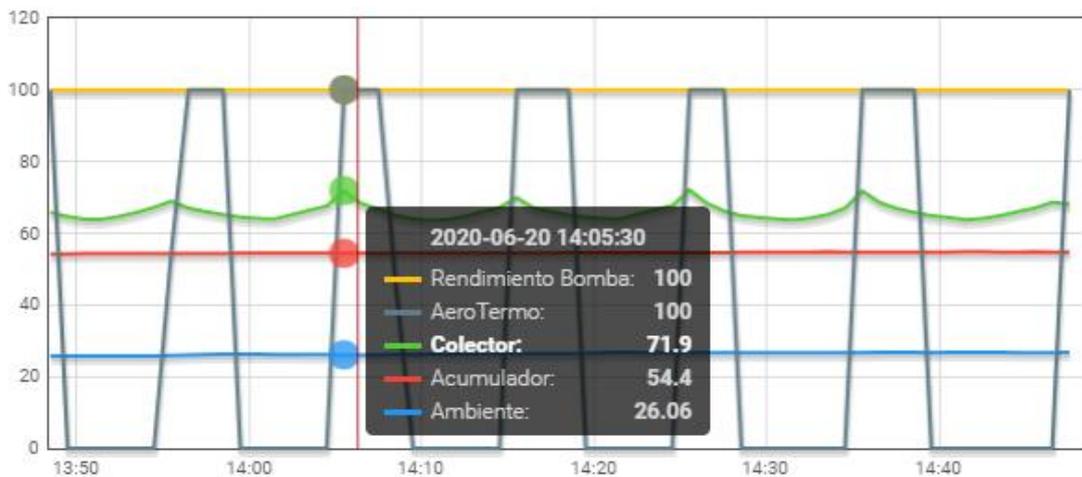
| | mínimo | máximo | promedio |
|---------------------|--------|--------|----------|
| — Ambiente | 19.49 | 26.75 | 21.83 |
| — Acumulador | 41.96 | 54.78 | 46.17 |
| — Colector | 16.75 | 67.71 | 34.63 |
| — Rendimiento Bomba | 0 | 100 | 33.53 |
| — AeroTermo | 0 | 78.57 | 5.69 |

Ilustración 34 (Gráfica: temperaturas)

Podemos observar en un solo vistazo las variables que intervienen en la gestión del agua de la instalación. Se aprecia como en el momento en el que amanece el colector empieza a calentarse, después la bomba empieza a funcionar y el agua del acumulador comienza a elevar su temperatura. En la segunda gráfica, en la que hemos hecho un zoom temporal, acción que puede realizar el usuario en el panel al que tiene acceso, observamos como entra en acción el aerotermo en el momento en que la temperatura del colector alcanza los 70°C y como provoca que baje.

Temperaturas

🕒 Tiempo real - último(s) horas



| | mínimo | máximo | promedio |
|-------------------|--------|--------|----------|
| Ambiente | 25.75 | 26.75 | 26.39 |
| Acumulador | 54.2 | 54.8 | 54.53 |
| Colector | 63.6 | 72.1 | 66.02 |
| Rendimiento Bomba | 100 | 100 | 100 |
| AeroTermo | 0 | 100 | 33.33 |

Ilustración 35 (Gráfica: temperaturas 2)

5.2. PANEL FOTOVOLTAICO E INVERSOR

Otra parte que necesitamos mantener controlada es el funcionamiento del panel fotovoltaico y del inversor. Controlar las potencias en las que se está trabajando, tanto en continua como en alterna, y la tensión que aparece en las placas son la parte más interesante a mostrar en el panel de ThingsBoard.

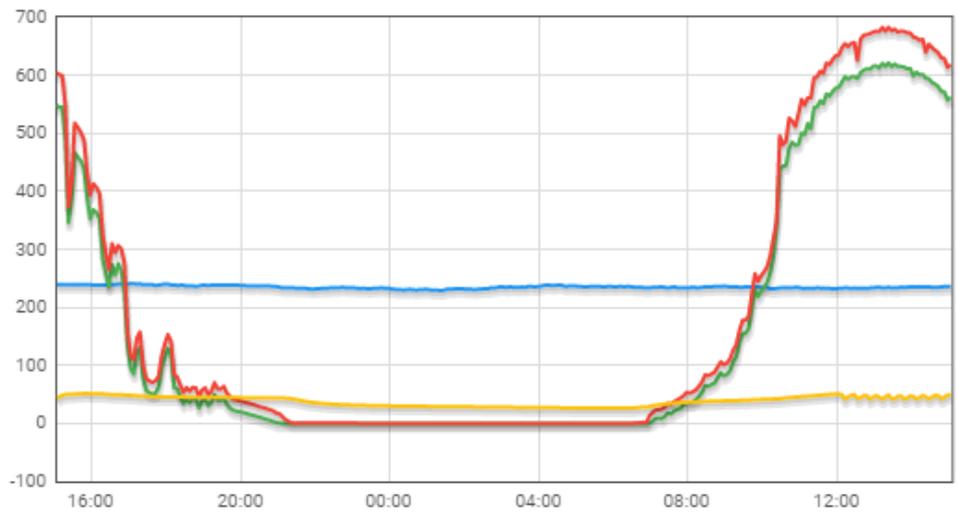
Aparte de esto, controlar la temperatura del inversor también es interesante para estar atentos a un posible sobrecalentamiento o un mal funcionamiento del mismo.

De esta manera construimos la gráfica que nos informara acerca de la generación de energía fotovoltaica.

Energía Fotovoltaica



🕒 Tiempo real - último(s) día



| | mínimo | máximo |
|----------------------|--------|--------|
| AC_Voltage | 228.75 | 240.81 |
| AC_Power | 0 | 621.28 |
| DC_Power | -0.04 | 682.62 |
| Temperatura Inversor | 26.13 | 51.46 |

Ilustración 36 (Gráfica: energía fotovoltaica)

Observamos como por la noche, como es evidente, la generación de energía se para y no comienza hasta que amanece, alrededor de las 7 a.m. La producción va creciendo hasta alcanzar su punto más alto sobre las 2 p.m. momento en el que los rayos del sol caen de manera más directa sobre el panel. Alcanzado este punto álgido, la producción decae hasta que se vuelve a detener con el atardecer y la ocultación del sol.

5.3. ENERGÍAS

La clave última de una instalación fotovoltaica es la generación de energía, por lo que la monitorización de esta es un paso obligado para el control de la instalación. En nuestro caso, aprovechando la energía solar vamos a producir energía eléctrica que calentará el depósito de agua transformándose así en energía calorífica.

Durante este proceso, y como muestran las leyes de la termodinámica, existirán pérdidas de energía que nos interesa conocer. Además, durante la noche, el enfriamiento del depósito también provocará una pérdida energética.

Por ello, vamos a mostrar la evolución de la energía eléctrica producida y de la energía almacenada en el acumulador. En ambos casos tenemos datos que podemos manipular para conocer dichas energías.

En primer lugar, para conocer la energía eléctrica producida en la instalación tan solo tenemos que integrar la potencia que tenemos en el inversor para obtener el dato, para ello obtendremos el valor de la potencia medio en un intervalo pequeño de tiempo y la multiplicaremos por ese pequeño intervalo.

$$\text{Energía} = \text{Potencia} * \text{Tiempo}$$

Para obtener la segunda, necesitamos saber:

- El calor específico de la sustancia que calentamos, en este caso agua, que le corresponde un valor de 4,186 julios/gramo °C
- La masa que calentamos, que, teniendo 150 litros de agua, y con una densidad de 1 Kg/L, sabemos que son 150 Kg.
- La diferencia de temperaturas.

De esta manera obtenemos la fórmula de la energía que buscábamos:

$$\text{Energía} = \text{Calor Especifico} * \text{Masa} * (\text{Temperatura final} - \text{Temperatura inicial})$$

Con estos dos métodos vamos a calcular las distintas energías que se producen en espacios de tiempo iguales, de esta manera podremos observar como la energía eléctrica se transforma en energía calorífica durante el día, mientras la instalación está en funcionamiento y como la perdemos durante la noche en forma de disipación calorífica.

Con esta intención construimos los *Widget* necesarios para mostrar la información. Hemos elegido una tarjeta donde actualizar los últimos datos, pudiendo elegir el usuario los intervalos temporales que desee, y un diagrama de barras que muestra ambas energías.

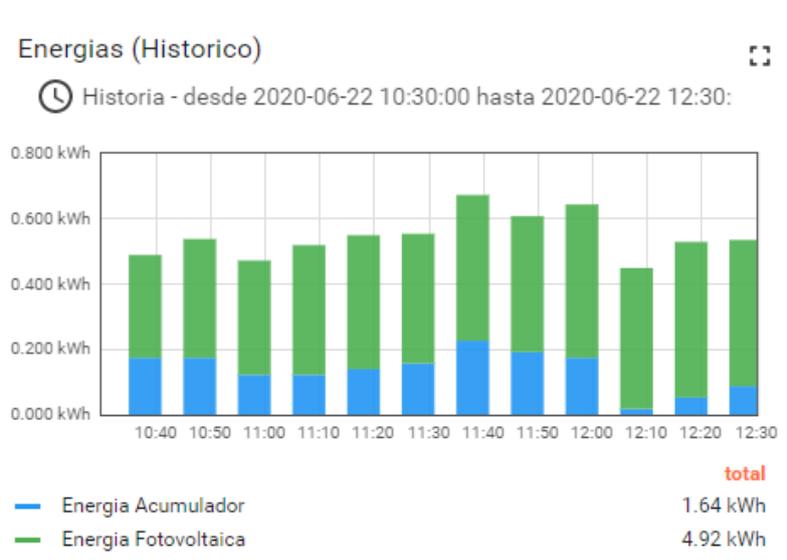


Ilustración 37 (Diagrama de Barras: Energías)

De esta manera observamos varias claves del funcionamiento. Al principio del día, cuando el sistema empieza a funcionar el agua se calienta de manera bastante rápida, hasta alcanzar los 55°C aproximadamente, pero una vez alcanza esta temperatura casi toda la energía se emplea en compensar las pérdidas por disipación del acumulador.

Una vez el sistema se apaga, el deposito pierde temperatura de manera suave pero constante y llega hasta unos 40°C en el momento de encender el sistema de nuevo. Comentar que esta monitorización se está haciendo en el mes de junio y es de esperar que este comportamiento varíe dependiendo de la temperatura exterior y por lo tanto de la época del año.

Recogiendo todas estas informaciones, tenemos acceso al estado de toda la instalación, pudiéndonos hacer una idea de su correcto funcionamiento general sin perder detalles que puedan sernos de utilidad.

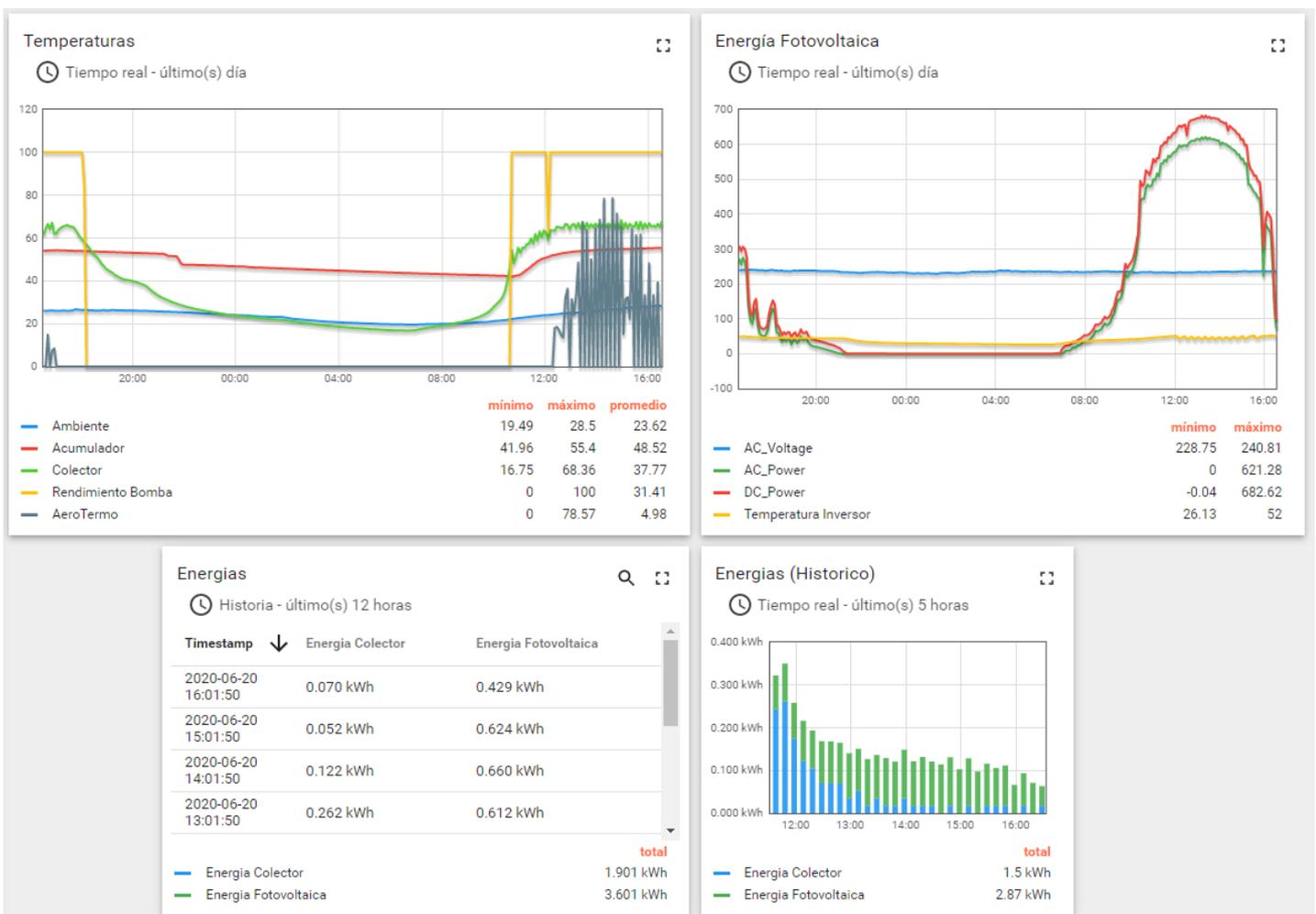


Ilustración 38 (Panel completo)

6. CONCLUSIONES

En este apartado vamos a valorar el trabajo realizado en el proyecto, posibilidad de mejoras aplicables con una posible futura continuación y unas reflexiones finales sobre el trabajo personal realizado.

6.1. CONCLUSIONES DEL PROYECTO

Para las conclusiones relacionadas con el proyecto vamos a diferenciar el análisis en tres partes, por un lado, todos los dispositivos encargados de recolectar toda la información disponible y transmitírsela al Pigeon, por otra parte, analizaremos el Pigeon como herramienta central y para finalizar trataremos el análisis sobre la plataforma ThingsBoard.

Dispositivos Recolectores

Sin duda ha resultado la parte que más trabajo y más dificultad ha tenido de todo el proyecto. Cabe destacar que cada uno de los dispositivos encargados de recolectar los datos funciona de una manera distinta, se conecta de manera distinta y emplea protocolos de comunicación distintos.

Empezamos por el Gestor de Resol, con un protocolo completamente determinado para los productos de esta marca fue quizás la parte del proyecto en la que se invirtió más recursos. También es el que más información nos facilita, teniendo dos sondas de temperatura, información de las bombas, información de la generación fotovoltaica...

El enfoque que desde un principio le dimos, con la clara intención de automatizar al máximo el proceso resultó clave a hora de aplicar el proyecto en la instalación real. No solo por facilitar el cambio si no porque, como ya contamos, en el nuevo dispositivo nos era útil no solo el mensaje que utilizamos en primera instancia, sino un segundo mensaje que nos facilitaba información acerca de la energía generada. Sin la automatización previa el trabajo que hubiese supuesto la adecuación a la nueva instalación hubiese resultado mucho mayor.

Sobre el contador de agua Contazara comentar que sirve como ejemplo claro de lo útil que resulta tener información instantánea de consumo de cualquier recurso, ya se agua o cualquier otro como puede ser gas o combustible. Disponer de esta información se muestra como una necesidad imprescindible a la hora de construir un sistema eficiente, tanto energética como económicamente.

El inversor Kostal también se presenta como una pieza fundamental en cualquier sistema que produzca energía fotovoltaica, tener constancia de cómo está funcionando esta transformación energética es indispensable para el control del sistema y la vigilancia del correcto funcionamiento del mismo.

Por último, la sonda de temperatura tan solo es un ejemplo de la enorme adaptabilidad que presenta el proyecto, proporcionada por el dispositivo Pigeon que pasamos a comentar a continuación.

Pigeon

El dispositivo Pigeon es la piedra angular del proyecto, la parte del sistema por el que circula toda la información, que la ordena y que la prepara para la visualización posterior que se producirá en la plataforma ThingsBoard.

La gran dificultad de utilizar este dispositivo ha sido la necesidad de trabajar siempre por consola, pese a que las pruebas iniciales las realizábamos en el ordenador. A cambio, y como acabamos de comentar, nos ofrece una gran versatilidad, tanto por el número de conexiones posibles que nos ofrece como por la capacidad de ejecutar en el cualquier programa con Python con todas las posibilidades que esto nos ofrece.

Si bien al comienzo del proyecto trabajar con el Pigeon era algo que conllevaba cierta dificultad, con el paso de las horas de trabajo y estando cada vez más familiarizado con sus sistemas el proceso se hizo cada vez más intuitivo y eficaz.

ThingsBoard

Para finalizar comentaremos las impresiones que nos ha dejado la plataforma ThingsBoard.

Aun con las horas dedicadas a esta herramienta queda la sensación que tan solo hemos rascado la superficie de esta plataforma. La capacidad de mostrar infinidad de datos de todas las formas imaginables provoca la sensación de que las posibilidades en las que podemos utilizar Thingsboard son innumerables.

Otra gran ventaja es que una vez los datos han sido recibidos se puede realizar operaciones relativamente sencillas para calcular otras variables, por ejemplo, calcular la potencia de salida del inversor sabiendo corriente y tensión.

La única pega que pondría a esta plataforma es lo estricta que es en cuanto al formato que deben tener los datos para que puedan ser visualizados, entendiendo que para automatizar el proceso ese tipo de formatos son necesarios, sería de gran utilidad que el usuario tuviese la posibilidad de visualizar todo lo que reciba el servidor de manera que supiéramos con certeza que el problema reside en el formato y no en cualquier otro paso del proceso del tratamiento de la información.

6.2. TRABAJO FUTURO

Las posibilidades con las que podemos ampliar este proyecto son prácticamente ilimitadas, podemos ser capaces de incluir dentro del sistemas desde la climatización total de la instalación hasta un sistema de alertas por dispositivos móviles, controlar la iluminación o la cloración de una piscina.

Precisamente creemos que este es el punto fuerte de la tecnología IoT y por ello del proyecto, la increíble versatilidad que nos ofrece, no obstante, por intentar desarrollar alguna idea vamos a desarrollar alguna de las posibilidades.

Actuadores

La opción de desarrollo siguiente que aparece de manera natural en el proyecto es la posibilidad de controlar distintos actuadores desde el propio ThingsBoard, es decir, tener la capacidad de, por ejemplo, encender o apagar una caldera, una luz o una bomba de agua desde la plataforma en internet.

Dado que la comunicación con ThingsBoard es bajo el protocolo MQTT la implementación sería similar a la existente. En el proyecto actual tenemos un servidor suscrito a un tema o *Topic* en el que el resto de dispositivos publican la información que van generando. Para ser capaces de controlar las acciones que estos dispositivos pueden ejecutar la idea sería la contraria, suscribir cada dispositivo a un *Topic* propio y publicar en él cada acción que queramos efectuar.

De esta manera, y por poner un ejemplo, si quisiéramos controlar una luz en el garaje, lo único que tendríamos que hacer es suscribir el interruptor de dicha luz a un *Topic*, Casa/Garaje/Luz, y en ahí publicar mensajes como ON u OFF y configurar el interruptor para apagarse cuando reciba el OFF y encenderse cuando reciba el ON.

Alarmas

Otra opción que creo que podría mejorar mucho el proyecto es la inclusión de alarmas dentro del sistema. Resultaría de gran utilidad ser capaces de alertar al usuario, vía móvil por ejemplo, del funcionamiento erróneo de alguna parte del sistema.

Ser capaces de avisar de una pérdida de agua en la caldera, de un sobrecalentamiento de alguna de las partes del sistema o del mal funcionamiento de la generación eléctrica sin tener que esperar a que el usuario entre en el portal de ThingsBoard a comprobar que efectivamente todo funciona bien, sería sin duda una gran contribución a la funcionalidad del sistema.

ThingsBoard permite programar reglas de cadenas que podrían utilizarse de forma interesante para desarrollar esta idea.

Deep Learning

Como ultima idea de desarrollo futuro del proyecto vamos a mencionar el *Deep Learning*.

Sin duda sería la que más dificultades nuevas podría plantearnos, pero también sería la que añadiría un potencial mayor al proyecto. Ser capaz de desarrollar una red neuronal u otro tipo de IA capaz de aprender cuando es más eficiente, energética y económicamente, calentar el agua de una caldera, o pedir o ceder electricidad a la red eléctrica general, de qué manera calentar un espacio o modificar el comportamiento del

sistema en función del tiempo meteorológico, por poner algunos ejemplos, sería añadir un potencial increíblemente potente al proyecto.

Por supuesto se trata de una tarea compleja y que requeriría el desarrollo de una solución más amplia donde nuestro sistema sería una pequeña parte de un proyecto mucho más ambicioso, pero como idea de desarrollo futuro con el que seguir ampliando un gestor de energía capaz de integrar la mayor cantidad de datos a nuestro alcance parece una de las ideas más potentes a proponer con las tecnologías actuales.

6.3. CONCLUSIONES PERSONALES

Una vez concluido el proyecto puedo decir que a nivel personal he aprendido mucho, tanto a nivel de conocimientos técnicos relacionados con la profesión como a nivel de desarrollo personal.

La gran diferencia y la gran dificultad en comparación con el resto de la titulación creo que es que por primera vez he trabajado con algo que no estaba preparado para funcionar bien, es decir, me he encontrado con problemas que no aparecían necesariamente por un mal planteamiento o una mala ejecución si no porque a veces las cosas simplemente no funcionan como uno esperaba y es necesario buscar soluciones o alternativas que nos permitan progresar en el proyecto.

Como gran ejemplo de lo expuesto en el anterior párrafo, la aparición de la pandemia de la COVID me obligo a trabajar de manera muy distinta a como estaba haciéndolo con anterioridad, pase de trabajar en un laboratorio donde tenía prácticamente todo lo que fuera a necesitar al alcance y rodeado de compañeros que podían ayudarme a resolver cualquier duda que pudiese surgirme a tener que trabajar en casa, montando un pequeño laboratorio y sin la posibilidad de comunicarme de manera directa con dichos compañeros.

Otra de las cosas que más me han gustado es la experiencia de enfrentarme a un objetivo en el que no tenía directrices que me limitaran a la hora de elegir como y con que solucionar cada uno de los retos que iban apareciendo a lo largo del proyecto, he podido decidir desde el lenguaje de programación a la manera en la que mostrar los datos.

He tenido que aprender a trabajar con herramientas con las que no estaba para nada familiarizado, por ejemplo, para programar el dispositivo Pigeon he tenido que utilizar en todo momento la consola de comandos, para navegar por los menús, para ver qué servicios estaban activos, para completar los códigos de la programación, ver si los puertos estaban conectados y funcionaban correctamente... También he aprendido a utilizar el lenguaje Python ya que en la carrera tan solo tuve un pequeño contacto con él, y gracias a esto descubrir su versatilidad y potencial.

Por último, decir que en esta memoria espero haber expresado bien la cantidad de tiempo que se ha invertido en cada uno de los aspectos que se han reflejado aquí, lo que en esta memoria ocupa quizás tres carillas explicando por ejemplo como programar el

contador Contazara detrás conlleva horas y horas de ensayos, pruebas, errores y más pruebas. Eso también me ha enseñado a valorar el trabajo que subyace tras las cosas que a simple vista parecen sencillas, pensar cuantas horas habrá detrás de las conclusiones y soluciones finales que se exponen cuando leemos un artículo o un proyecto.

A modo de resumen creo que he aprendido mucho a la hora de plantear un proyecto de manera que conocidos unos objetivos, constituir un plan para alcanzarlos, desarrollar dicho plan con una determinada metodología y llegar al final a una solución eficaz y que alcance a satisfacer dichos objetivos. Opino que haber desarrollado este proyecto, incluidas las dificultades añadidas por la pandemia, y sabiendo que aún tengo mucho por aprender, ha hecho que este mejor preparado para ejercer la profesión de Ingeniero de la mejor manera posible.

7. AGRADECIMIENTOS

A todos y cada uno de los que me han acompañado en la universidad, desde profesores y personal de cafetería a compañeros y amigos, por hacer este camino mucho más fácil. Gracias.

A mi director, D. Roberto Casas y los compañeros de *Howlab*, por lo aprendido en el laboratorio y la ayuda inestimable en la resolución de problemas y dudas que me surgieron durante el proyecto, y especialmente por las facilidades a la hora de trasladar el equipamiento necesario del laboratorio a mi casa a causa de la COVID. Gracias.

A mi familia y pareja por aguantar los días en los que era difícil aguantarme y apoyarme en este trayecto. Gracias.

A todos los grandes hombres y mujeres que se dedicaron a la ciencia y provocaron en mí el deseo de ser ingeniero (o inventor) desde muy pequeño. Gracias.

No hubiese sido posible sin todos vosotros.

8. BIBLIOGRAFIA

Modbus. (2020). Recuperado 11 de abril de 2020, de Wikipedia website:
<https://es.wikipedia.org/wiki/Modbus>

Tercer Revolución Industrial. (2020). Recuperado 11 de abril de 2020, de Wikipedia website: https://es.wikipedia.org/wiki/Tercera_revoluci%C3%B3n_industrial

Industria 4.0. (2018). Recuperado 15 de abril de 2020, de Idboxrt website:
<https://idboxrt.com/blog/industria-4-0-la-revolucion-del-iot-industrial/>

Modelo OSI. (2020). Recuperado 15 de abril de 2020, de Wikipedia website:
https://es.wikipedia.org/wiki/Modelo_OSI

Convertidores Modbus a Ethernet. (2020). Recuperado 15 de abril de 2020, de PerleSystems website:
https://www.perlesystems.es/supportfiles/modbus_gateway_technical_note.shtml

Vbus. (2020). Recuperado 15 de mayo de 2020, de Vbus website:
<https://www.vbus.net/#/>

CONTADOR TELEGESTIONADO. (2020). Recuperado 15 de mayo de 2020, de I-de website: <https://www.i-de.es/redes-inteligentes/contador-telegestionado#>

Iberdrola Área Clientes. (2020). Recuperado 15 de mayo de 2020, de Selectra website:
<https://selectra.es/energia/companias/iberdrola/clientes>

El contador inteligente. (2020). Recuperado 19 de mayo de 2020, de Endesa website:
<https://www.endesa.com/es/conoce-la-energia/energia-y-mas/contador-inteligente>

Voltímetro AC. (2020). Recuperado 19 de mayo de 2020, de AliExpress website:
https://es.aliexpress.com/item/32320321777.html?cv=12_DeepLink&af=179029&aff_platform=AAF&sk=Y7bAZbY&aff_trace_key=549a7139943342d0be50a8f25550d077-1586542829584-02947-Y7bAZbY&cn=15716&dp=12%3A%3A179029%3A%3A%3A%3A%3A1586542829&terminal_id=6b5b50543b7344cead41282df2f36275&aff_request_id=549a7139943342d0be50a8f25550d077-1586542829584-02947-Y7bAZbY

Home Connect. (2020). Recuperado 5 de abril de 2020, de Home-Connect website:
<https://www.home-connect.com/es/es/>

Qué es IFTTT y cómo lo puedes utilizar para crear automatismos en tus aplicaciones. (2019). Recuperado 4 de abril de 2020, de Xataka website:
<https://www.xataka.com/basics/que-ifttt-como-puedes-utilizar-para-crear-automatismos-tus-aplicaciones>

Referencias de Productos Resol. (2019). Recuperado 4 de febrero de 2020, de danielwippermann website: https://danielwippermann.github.io/resol-vbus/vbus-packets.html#0010_1122_

Contazara. (2020). Recuperado 8 de febrero de 2020, de Contazara website: <https://www.contazara.es/index.php/es/producto>

¿QUÉ ES MQTT? SU IMPORTANCIA COMO PROTOCOLO IOT. (2019). Recuperado 8 de junio de 2020, de Luis Llamas website: <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>

Dos millones de razones para saber qué es exactamente Raspberry Pi. (2013). Recuperado 8 de junio de 2020, de El Confidencial website: https://www.elconfidencial.com/tecnologia/2013-11-22/dos-millones-de-razones-para-saber-que-es-exactamente-raspberry-pi_56003/#:~:text=%C2%BFQu%C3%A9%20es%20Raspberry%20Pi%3F,escuelas%20de%20todo%20el%20mundo.

PIKO MP plus: el inversor monofásico flexible. (2020). Recuperado 11 de junio de 2020, de Kostal website: <https://www.kostal-solar-electric.com/es-es/products/hybrid-inverters/piko-mp-plus>