

# Diseño y construcción de un sistema para adquisición y análisis del consumo energético en el hogar

Jesús Sánchez de Lechina Tejada

Junio 2020



**UNIVERSIDAD  
DE GRANADA**

# Índice

|  |           |
|--|-----------|
| <b>1. Internet de las Cosas</b>  | <b>4</b>  |
| 1.1. ¿Qué es un dispositivo IoT?   | 4         |
| <b>2. Revisión de enchufes inteligentes existentes en el mercado</b>           | <b>5</b>  |
| 2.1. ¿Qué es un enchufe? Tipos de enchufe                                      | 5         |
| 2.2. ¿Qué es un enchufe inteligente?   | 5         |
| 2.3. Funcionalidades comunes que ofertan los enchufes electrónicos actualmente | 5         |
| 2.3.1. Monitorización:   | 5         |
| 2.3.2. Control:  | 5         |
| 2.4. Tipos de enchufes   | 6         |
| 2.5. Modelos en el mercado   | 6         |
| <b>3. Arquitecturas IoT</b>  | <b>7</b>  |
| 3.1. Protocolos  | 7         |
| 3.2. Modelos de interconexión de red   | 8         |
| <b>4. Diseño de la arquitectura de nuestro sistema</b>                         | <b>9</b>  |
| 4.1. Descripción general   | 9         |
| 4.2. Modelo de red usado   | 9         |
| 4.3. Protocolo MQTT  | 9         |
| 4.4. Diseño broker, publishers, suscribers                                     | 9         |
| 4.4.1. Mosquitto   | 10        |
| 4.5. Enchufe   | 11        |
| 4.5.1. Diseño y construcción del enchufe                                       | 11        |
| 4.5.2. Programa del microcontrolador   | 12        |
| 4.6. Servidor central  | 16        |
| 4.6.1. Configuración de red  | 16        |
| 4.6.2. Gestión de la base de datos   | 16        |
| 4.7. Aplicación de escritorio  | 17        |
| <b>5. Pruebas y test</b>   | <b>18</b> |



# 1. Internet de las Cosas

Uno de los términos más recurrentes en el ámbito de la tecnología a día de hoy es el de “*Internet de las Cosas*” (en inglés, IoT, *Internet of Things*). Pero, puesto que el término puede resultar ambiguo, es conveniente dar una definición sobre esta que permita esclarecer el concepto.

El Internet de las Cosas es un paradigma tecnológico en sí mismo. Desglosando el concepto vemos que reúne dos tecnologías. Por un lado parte de “las cosas”, que abarca desde los dispositivos y electrodomésticos que podemos encontrar en nuestro día a día; que tienen una funcionalidad en sí misma (p.ej. una televisión, un frigorífico), hasta incluso podemos abstraer a personas o animales (una persona con un marcapasos, un ave con un geolocalizador u otros ejemplos[1]). A estas “cosas” se les añade el término “Internet”, que no es más que una abstracción de la conectividad que permite dotar a las cosas de comunicación con el exterior, extendiendo sus funcionalidades sin privarlas de su cometido original. Esta es su principal característica, la posibilidad de comunicación con otras “cosas” mediante internet y sin la necesidad de intervención humana.

La idea que trasciende de esto es que cualquier cosa o dispositivo que usemos habitualmente puede conectarse a una red, a internet. Haciendo que, en términos de redes, un dispositivo IoT se pueda equiparar a un ordenador convencional.

## 1.1. ¿Qué es un dispositivo IoT?

En el símil anterior hemos adelantado el concepto de dispositivo IoT sin llegar a definirlo completamente. A continuación explicaremos qué es concretamente un dispositivo IoT y qué lo diferencia de otras “cosas”.

Un dispositivo IoT es una entidad (objeto, o incluso animales o personas) que tiene una funcionalidad en sí misma y a la cual dotamos de una capacidad de conexión y telecomunicación. De modo que sea capaz de, por sí misma, comunicarse con otros dispositivos en su entorno dotados de esta misma capacidad.

Los beneficios que podemos obtener de este paradigma se pueden aplicar en muchas áreas[2], por ejemplo:

- Logística: Rastreo de envíos por correo o estado del stock en almacenes automatizando lectura de los items mediante wifi.
- Transporte: Gestión automática de rutas por GPS; captura y procesamiento de infracciones de velocidad.
- Salud: Desde una pulsera que capte tus hábitos de vida hasta la supervisión de personas ancianas que viven solas.
- Monitorización: Desde instalación de sensores en un bosque para medir datos de contaminación hasta medir en un hogar el consumo de un electrodoméstico.

Este último ejemplo es justo la motivación de nuestro trabajo. Podemos usar esta tecnología para crear un **enchufe inteligente** que nos permita saber nuestro consumo eléctrico y controlar el uso que le damos.

## **2. Revisión de enchufes inteligentes existentes en el mercado**

### **2.1. ¿Qué es un enchufe? Tipos de enchufe**

Un enchufe es una fuente de alimentación entre sistemas eléctricos. En este proyecto trataremos los enchufes dentro del ámbito doméstico o comercial frente al ámbito industrial, que tiende a trabajar con voltajes de un orden muy superior.

Un enchufe en el ámbito doméstico permite la conexión entre un dispositivo electrónico y la corriente alterna para la alimentación de este dispositivo.

Estos trabajan con unos voltajes entre 100V y 240V.[3] El tipo de enchufe, voltaje y frecuencia usados en una región geográfica vienen determinados por un convenio fijado por el gobierno de esa zona.

Por razones históricas[4] no existe una estandarización.

Trabajaremos con el tipo de enchufe usado en España, que cuenta con un potencial de 230V y una frecuencia de 50Hz[5].

### **2.2. ¿Qué es un enchufe inteligente?**

Un enchufe inteligente extiende esta definición de enchufe. Actuando como una interfaz que permite añadir un amplio abanico de funcionalidades relativas al manejo y supervisión de estos dispositivos externos. Dotándole de las ventajas de la conectividad.

Esto encaja con la definición de dispositivo IoT que mencionábamos previamente. Un objeto cotidiano que ha sido dotado de la capacidad de comunicación con otros dispositivos.

### **2.3. Funcionalidades comunes que ofertan los enchufes electrónicos actualmente**

#### **2.3.1. Monitorización:**

Supervisar el uso de energía: Esto permite establecer un control sobre su uso, realizar estimaciones sobre el coste o incluso notificar frente a anomalías como el consumo en horarios no esperados o picos de voltaje.

#### **2.3.2. Control:**

Manejar tus dispositivos de manera remota, programar el horario de funcionamiento de estos o la manera en la que funcionan son solo algunas de las opciones; destacando la posibilidad de integración con otros dispositivos que proporcionen una mayor accesibilidad (p.ej. la interfaz de voz de un asistente de móvil).

## 2.4. Tipos de enchufes

Atendiendo a sus características más distintivas podemos hacer una clasificación (no excluyente) de:

- Enchufes inteligentes (E.I.) por control remoto (wifi, bluetooth, infrarrojos u otros tipos de señales electromagnéticas).
- E.I. programables, aquellos que permiten la programación temporal de eventos.
- E.I. de regletas, aquellos que permiten tratar con varios dispositivos simultáneamente en un mismo enchufe.

## 2.5. Modelos en el mercado

La comercialización de este producto está muy extendida, proporcionando una amplia oferta de productos los cuales podemos recoger bajo la clasificación previa:

- Programación temporal: Permiten la ejecución de tareas a horas concretas del día. Por ejemplo: [Garza 400603 \(analógico\)](#), [Orbegozo PG 20 \(digital\)](#)
- Control remoto: Se pueden controlar con mando a distancia, a través de una app o con control por voz. Por ejemplo: [TP-Link HS100](#) o la [Regleta Xiaomi](#), que es un ejemplo de intersección entre E.I. de regleta y por control remoto.

### 3. Arquitecturas IoT

El paradigma de IoT viene a integrarse entre los mecanismos convencionales y el resto de redes. En sus primeros pasos se adaptaron sus diseños a las necesidades de estos sistemas. Pero según fue desarrollándose fueron naciendo algunas arquitecturas que ayudaban a definir y a trabajar mejor con los requisitos de estos nuevos sistemas.

Existen diferentes arquitecturas en la actualidad, que difieren en el enfoque que le dan a ciertos campos. Dos de los más conocidos son IoT-A e IIRA. Siendo el primero el más extendido desde su lanzamiento en 2012.[6]

Algunos campos de comparación entre arquitecturas de IoT son[7]:

- El enfoque particular a casos de negocio frente a conceptos generales.
- Orientación a Internet: Si se introducen nuevos protocolos y se adaptan los ya existentes a sus necesidades.
- Tipo de dispositivo sobre el cual se orienta (como sensores o actuadores).

Estas arquitecturas coinciden en la definición de capas de manera análoga a TCP/IP u OSI[6], pero adaptándolas a las necesidades del IoT. Ante las diversas clasificaciones en múltiples capas se puede encontrar una similitud entre ellas que permite generalizar en 3 capas principales: Percepción, asociada a los dispositivos físicos, consistente en sensores; capa de red, encargada de transmitir información entre sensores y sistemas procesadores finales; y capa de aplicación, para el proceso final de la información[8].

Para conocer cómo se produce la comunicación a distintos niveles de la arquitectura es necesario analizar sus protocolos.

#### 3.1. Protocolos

Dos ordenadores convencionales necesitan una pila de protocolos para que puedan transmitir información a las distintas capas que conforman la comunicación. Ante los requisitos que surgían con los dispositivos IoT, emergían también protocolos que daban respuesta a estos. Estableciéndose así una pila de protocolos de red especializada para IoT.

A fin de permitir estas comunicaciones de manera liviana es necesario alguna tecnología a **nivel físico** que pueda solventar el problema de la saturación del espectro de frecuencias, algunas propuestas se basan en aprovechar el ruido blanco (frecuencias no usadas en el espectro de radio)[9]. En la **capa de enlace** se encuentra 6LoWPAN[10] para conectar los dispositivos a una red IP, dicha red ha de ser del tipo IPv6 (**capa de red**) para dar soporte a la escalabilidad.

En la **capa de aplicación** existen alternativas para proporcionar un transporte de datos con baja sobrecarga en la comunicación. Algunas de estas opciones[11] son: **CoAP** (Constrained Application Protocol), usado como un protocolo web basado en *REST*[12]; **XMPP** (Extensible Messaging and Presence Protocol), usado especialmente en mensajería; **DDS** (Data Distribution Service), especializado en aplicaciones de tiempo real y con alta disponibilidad; o **MQTT** (Message Queue Telemetry Transport), para conectar dispositivos embebidos con aplicaciones y middleware.

Esta es una de las decisiones que más afecta al diseño, pues el resto del diseño de la arquitectura se ve afectado por ello.

### 3.2. Modelos de interconexión de red

De acuerdo a la topología de la red, el modo en el que se conectan los dispositivos involucrados, reconocen ocho tipos básicos de modelos de interconexión.[\[13\]](#)

- Punto a punto: Dos dispositivos finales se perciben directamente conectados entre sí.
- Cadena Margarita: Se conecta en serie cada computador al siguiente, los mensajes se retransmiten durante toda la cadena.
- Bus: Todos los dispositivos están conectados entre sí usando un cable central. Esto implica una difusión de cada mensaje a todos los nodos. Es más usual en redes locales.
- Estrella: Un conjunto de nodos periféricos se conectan con un nodo central que actúa como servidor.
- Anillo: Similar a una topología de “cadena margarita” donde existe un único bucle y la comunicación se realiza en un único sentido.
- Malla: Los dispositivos se conectan de muchos a muchos, puede ser completamente conectada o parcialmente conectada si todos los nodos se encuentran conectados entre sí o no.

El modelo de interconexión de red de un sistema no es inherente al Internet de las Cosas, es una cuestión recurrente en cualquier diseño de una red de telecomunicación y cuya decisión se realiza en tanto a las necesidades del sistema. Por ejemplo un modelo de malla consta de una mayor complejidad en las comunicaciones que una de anillo, pero este puede dificultar las comunicaciones entre distintos nodos. Una arquitectura de estrella simplifica este problema pero se vuelve dependiente de que el nodo central se encuentre siempre operativo.



## 4. Diseño de la arquitectura de nuestro sistema

### 4.1. Descripción general

Nuestro enchufe inteligente recoge mediante el sensor SCT013 la corriente que requiere el dispositivo conectado. En el módulo ESP32 se realiza el cálculo del consumo en Vatios y se emite hacia nuestro servidor, una Raspberry Pi con Mosquitto y MQTT, de donde se podrá consultar desde una aplicación externa para supervisar el consumo y manipular el enchufe.

Como mencionábamos en la sección 3, habitualmente las arquitecturas IoT se engloban bajo tres capas. La de percepción, nuestro sensor que recoge los datos; la capa de red, el uso de MQTT para gestionar las comunicaciones de manera liviana (sobre TCP/IP) y la capa de aplicación, que permite el procesamiento, manipulación y visualización por parte del usuario del sistema.

### 4.2. Modelo de red usado

La topología de red usada en este proyecto es el modelo en estrella. En el centro se encuentra nuestro servidor que se encarga de gestionar tanto la información de los sensores como de otorgar la visualización y manejo sobre los enchufes a los usuarios. En los extremos se encontrarían tanto los enchufes inteligentes como los usuarios desde sus aplicaciones.

Se ha decidido utilizar este modelo tanto por su versatilidad como simplicidad para conectar los nodos periféricos al servidor central.

### 4.3. Protocolo MQTT

De los protocolos previamente mencionados [3.1] optamos por utilizar MQTT por su facilidad de integración con Mosquitto, un broker (agente intermediario) que implementa este protocolo y cuya ligera carga le permite ejecutarse en servidores de bajas prestaciones (Raspberry Pi, en nuestro caso).

MQTT utiliza el patrón de publicación/suscripción (publish-suscribe), un publicador escribe sobre un tema (topic) del cual un suscriptor lee. Está construido sobre TCP.

El tipo de comunicación en este protocolo es indicado en cada mensaje en la cabecera del paquete. Proporciona mensajes para conexión (CONNECT), desconexión (DISCONNECT), publicar en un tema (PUBLISH), suscribirse/desuscribirse a un tema (SUBSCRIBE/UNSUBSCRIBE) y otros mensajes de control correspondientes a una respuesta ACK a los anteriormente mencionados (pues está basado en TCP).[14]

### 4.4. Diseño broker, publishers, suscribers

Aunque existen otros subtipos de patrones publicador/suscriptor basados en tipo o en contenido[15], MQTT utiliza un modelo basado en temas.

Las principales componentes de este patrón son suscriptor, publicador y el broker. Un dispositivo interesado se registra como suscriptor de un tema y el broker se encargará de informarle cuando se publique algo en el tema. El publicador actúa como un generador de información de algún tema de interés. El broker es el intermediario que transmite a los interesados, teniendo en cuenta además comprobaciones de seguridad mediante autorización de publicadores y suscriptores.[11, 16]

Por este motivo, en este patrón pueden existir múltiples publicadores y suscriptores mientras que es la entidad que actúa como broker la que gestiona el flujo de comunicación.

#### 4.4.1. Mosquitto

El broker que usamos en nuestro diseño es **Mosquitto**[\[17\]](#). Agente que implementa MQTT v3.1.1, la versión de MQTT usada en este proyecto.

Uno de los requisitos más habituales en sistemas IoT es la integración de manera ubicua de las componentes en el entorno del usuario. A menudo esto implica utilizar sistemas compactos o embebidos con pocos recursos. En nuestro caso utilizamos un servidor con recursos limitados, una Raspberry Pi, por lo que resulta beneficioso utilizar un broker tan ligero que no suponga una sobrecarga en el sistema.

## 4.5. Enchufe

De los tres subsistemas que componen el diseño de nuestro sistema podríamos atribuirle el concepto de Enchufe Inteligente a este. Pues es el encargado de realizar las lecturas del consumo energético del dispositivo conectado, transmitir estas lecturas al servidor central y manejar mediante el uso de un relé la alimentación del dispositivo conectado para regular su uso.

El enchufe está compuesto por cuatro elementos principales:

- **Módulo Wifi:** Microcontrolador ESP32[18]. Núcleo del enchufe, encargado de ejecutar el programa principal que gestiona el manejo de lecturas y del relé. Dispone de funcionalidad WiFi para conectarse vía MQTT al servidor.
- **Relé:** Tongling jqc-3ff-s-z, un relé de 5V que controla el paso de corriente por el dispositivo conectado cual interruptor y que es gestionado por el módulo ESP32.
- **Sensor de corriente:** Sensor YHDC SCT013000 CT[19]. Un sensor de corriente no invasivo (basta con colocarlo alrededor del cable a medir) que actúa como un transformador de corriente. Transforma la corriente de entrada entre 0 y 100 Amperios en otra de salida entre 0 y 50 miliAmperios que, mediante el uso de una resistencia de carga que convierta a un voltaje adecuado, el módulo es capaz de procesar.

Los cables convencionales suelen estar compuestos por dos cables internos que conducen la corriente en sentido opuesto. Para evitar que se anulen las medidas el sensor se coloca en torno a uno de ellos. Además, es posible medir el sentido de la corriente para determinar si se está consumiendo o generando electricidad.

- **Fuente de alimentación:** Una fuente de alimentación de 5 Voltios y 2 Amperios para alimentar el módulo wifi. Un conversor que toma la corriente alterna que llega al enchufe y alimenta al resto del circuito.

### 4.5.1. Diseño y construcción del enchufe

En un proceso de prueba y selección de componentes descartamos otras opciones en el diseño de nuestro enchufe:

El microcontrolador Arduino UNO carecía de funcionalidad WiFi y el uso de módulos externos requería alimentación extra. El ESP32 usado sí que disponía de WiFi y era compatible con el resto de componentes, reduciendo así la complejidad de este subsistema mientras otorgaba la misma funcionalidad.

El sensor de corriente no invasivo fue seleccionado tras descartar el sensor ACS712[20]. Este introducía ruido en las lecturas, mostrando diferencias entre el consumo esperado y el registrado para componentes eléctricas que concieramos de antemano.

Tras esta selección procedimos a probar y, posteriormente, soldar el enchufe a una placa base. A fin de minimizar el número de cruces entre cables elaboramos el diseño visible en la Figura 1. Este representa la vista superior de la placa y cómo se disponen los cables y las componentes tanto por encima como por debajo de la misma. Es conveniente indicar que cada rectángulo representa un conjunto de pines que irán conectados a alguna componente del sistema: El rectángulo GND, 5V a la fuente de alimentación, el par In, Out la entrada y salida del sensor de corriente y la tres-upla Vcc, Gnd y S a la alimentación y control del relé.

Hacemos uso del condensador  $C1 = 10\mu F$  para absorber los posibles picos de tensión, reducir el ruido en las lecturas y garantizar calidad en las mismas.

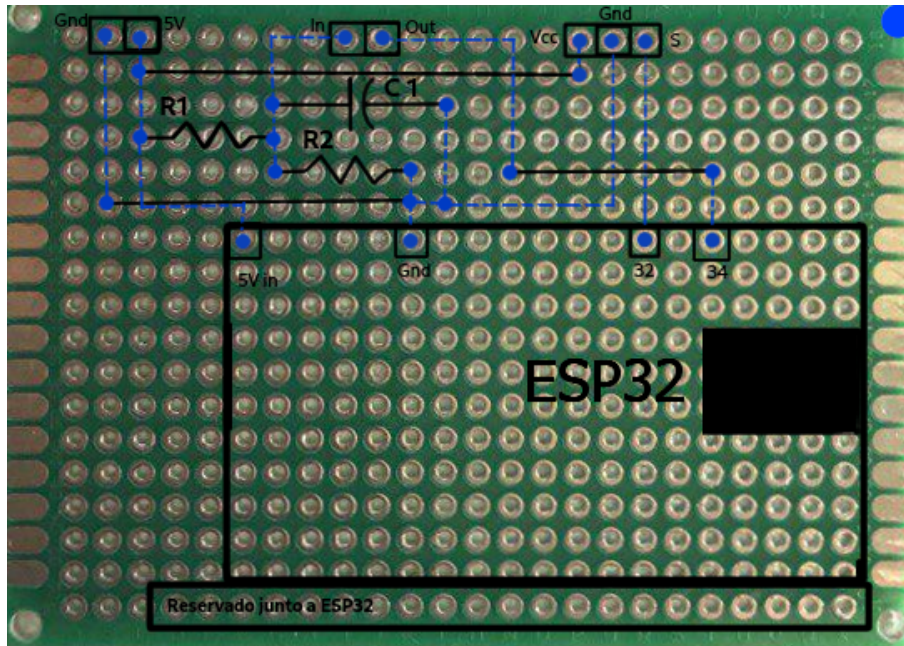


Figura 1: Vista superior de la placa, en negro componentes y cableado de la parte anterior. En azul, de la parte posterior.

Las resistencias  $R1 = R2 = 2k\Omega$  cumplen la función de un divisor de tensión y de resistencia de carga para garantizar el voltaje operativo del sensor de corriente y otorgar el rango de voltaje de salida esperado.

Para complementar esta visualización añadimos la parte posterior (trasera) de la vista trasera de la placa (volteo vertical), Figura 2, y la parte posterior de la vista superior, Figura 3.

Los adaptadores de los pines sobre los que se colocarán las otras componentes queda dispuestos de acuerdo a la Figura 4. Hacemos uso de estos adaptadores a fin de simplificar las tareas de mantenimiento y sustitución de componentes en caso de avería.

El circuito resultante es el que podemos observar en las figuras 5 y 6.

Tanto la alimentación como la medición de la corriente se produce a partir de un cable que hemos seccionado y empalmado a las componentes de nuestro circuito.

Para abstraer el diseño del enchufe y simplificar el uso del mismo al usuario lo hemos acoplado en una caja de conexión. De este modo el usuario final puede limitarse a, simplemente, conectar el dispositivo a medir a nuestro enchufe inteligente y este a una fuente de alimentación. Una conexión macho y una hembra.

#### 4.5.2. Programa del microcontrolador

Toda la lógica del enchufe, tanto lecturas como comunicaciones, son gestionadas por el programa del microcontrolador. Y resulta conveniente comprender qué elementos hacen posible esta mecánica.

Por un lado se encuentra la configuración de la comunicación. Es necesario conectarse a la red sobre la que se trabajará para poder comunicarse con el broker, así que se requiere conocer SSID y contraseña de la red en la que se trabajará. Del mismo modo la información del broker, temas MQTT de suscripción o publicación y credenciales tendrán que ser conocidos. Dada la licencia permisiva de este proyecto y la flexibilidad de programación del módulo ESP32 no habrá problema



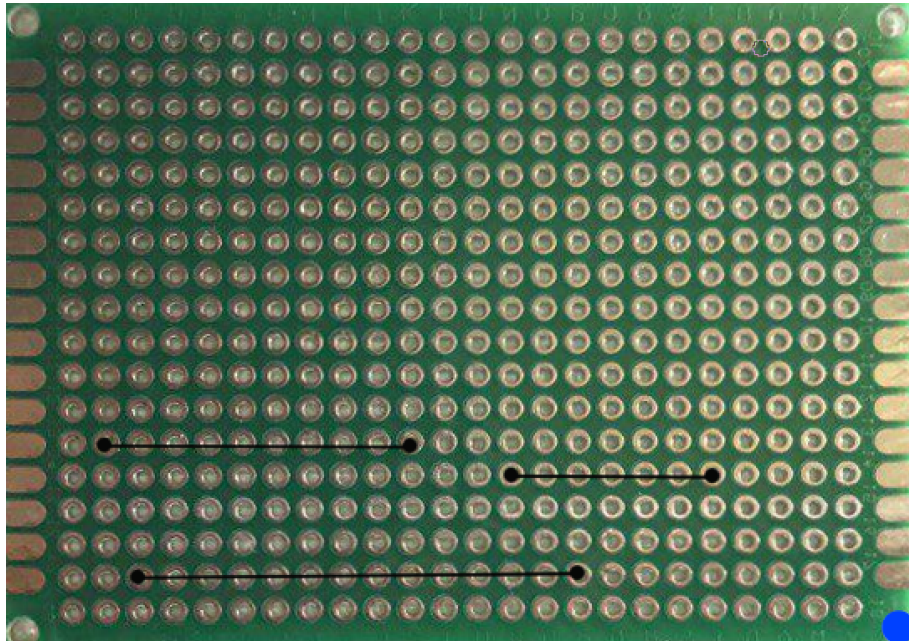


Figura 2: Vista trasera de la placa, parte posterior.

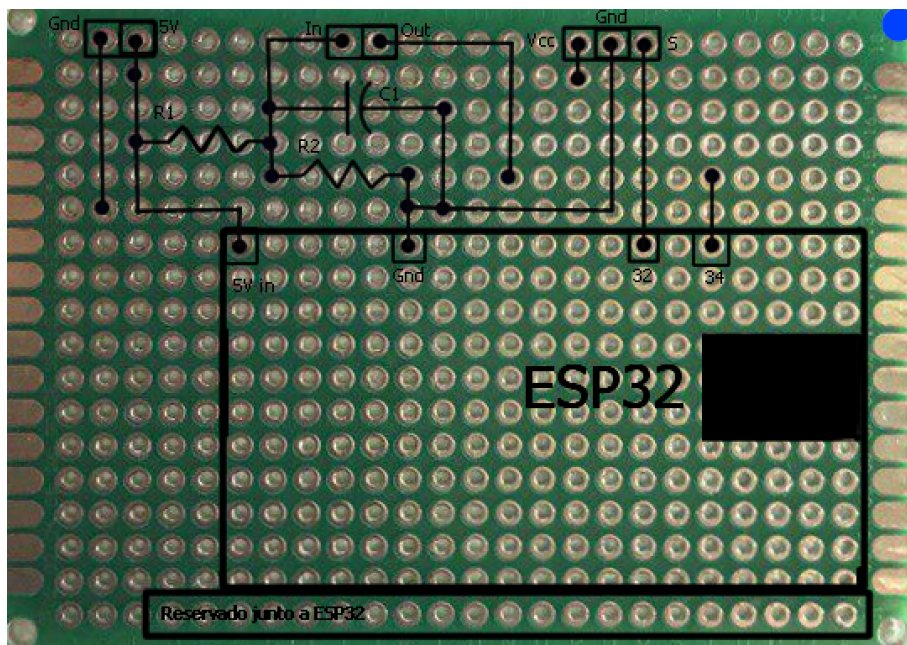


Figura 3: Vista trasera de la placa, parte posterior.

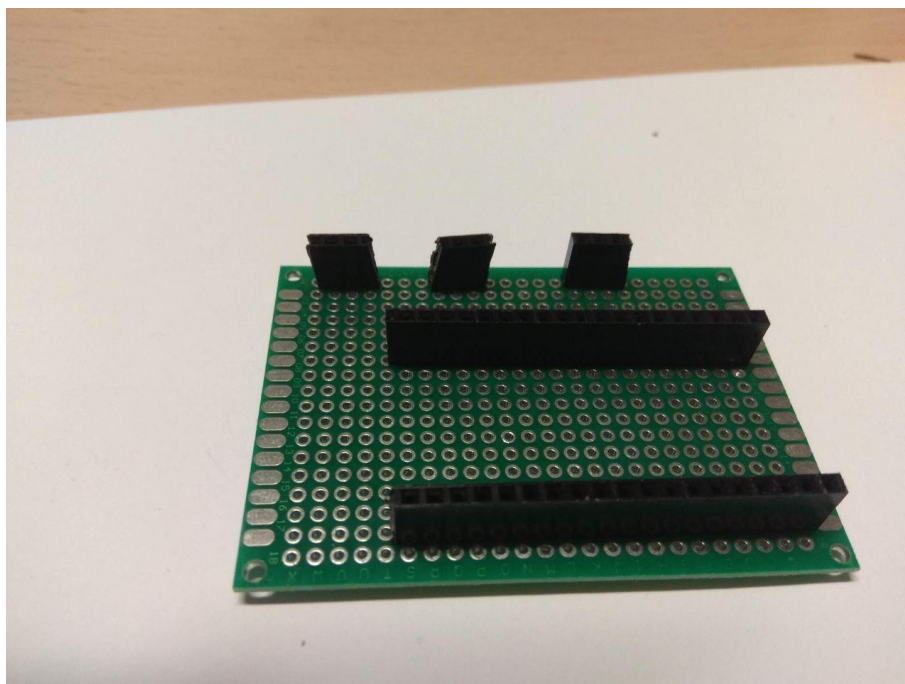


Figura 4: Adaptadores de las componentes.

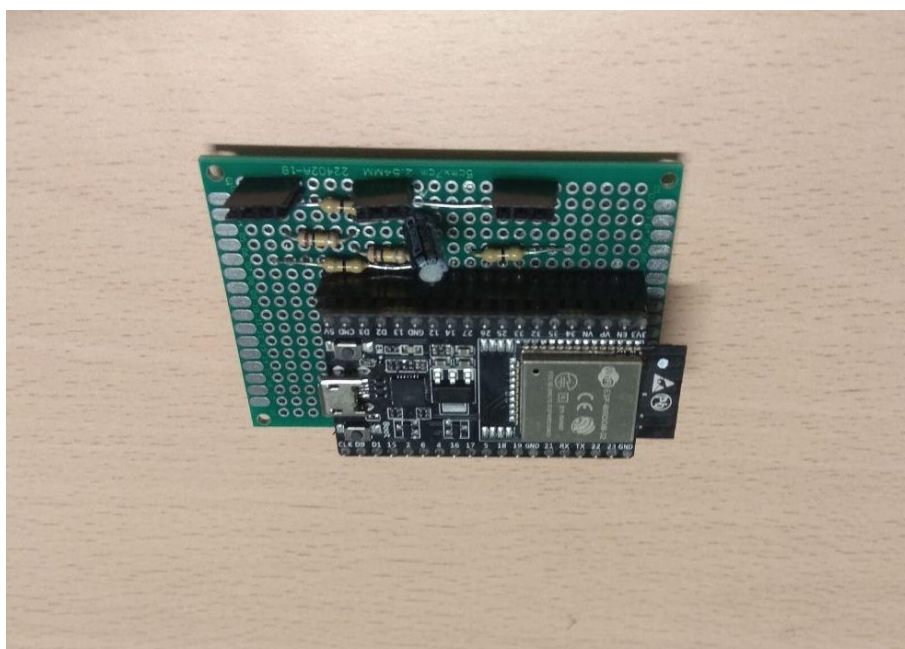


Figura 5: Soldadura parte superior placa.

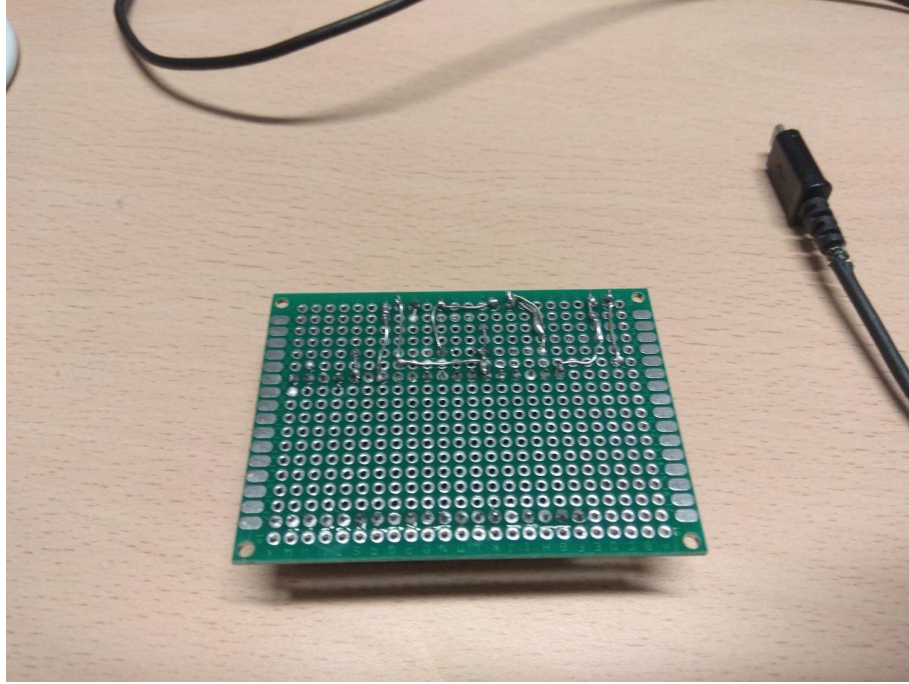


Figura 6: Soldadura parte inferior placa.

en entregar el sistema al usuario final en el estado de uso inmediato o de permitir que este lo adapte a sus necesidades.

El enchufe publica el cálculo del consumo en un tema asociado a su identificador en el broker y se suscribe a los canales de control para gestionar su funcionamiento (actualmente manejo del relé, pero puede extenderse a configuración de otros parámetros).

A fin de procesar la información del consumo cada enchufe está identificado unívocamente por un identificador que puede ser adaptado *ad-hoc* (se podrían utilizar otros identificadores como el del propio microcontrolador, pero se deja a opción del usuario para permitir flexibilidad en la nomenclatura).

En el apartado de las lecturas es destacable cómo se han realizado los cálculos.

En términos de electromagnetismo se define la potencia  $W$  (la variable que utilizamos para medir el consumo) como:

$$W = V \cdot A$$

Siendo  $V$  el voltaje del circuito, 230 voltios en España, y  $A$  la intensidad de la corriente que mide el sensor.

Con las mediciones del sensor de corriente calculamos la media cuadrática. La corriente alterna fluctúa en torno a valores positivos y negativos siguiendo un comportamiento sinusoidal. La manera de calcular la intensidad que realmente está pasando es obteniendo el valor eficaz. Este valor eficaz para la intensidad se corresponde con el valor cuadrático medio[21].

Para reducir hipotéticos picos de ruido se realiza un muestreo un intervalo de tiempo fijo y se calcula el consumo durante ese período de tiempo  $W$  tomando como valor de la corriente  $A$  la media de las medidas en el muestreo.



## 4.6. Servidor central

En el centro de nuestra arquitectura de estrella se encuentra el servidor central. Una Raspberry Pi 3 Modelo B que actúa como broker entre los publicadores y suscriptores de MQTT, como mencionábamos en 4.4.1, haciendo uso de Mosquitto para gestionar las publicaciones y suscripciones a los temas.

Es necesario que el broker esté activo y sea persistente a reinicios y caídas. Por esto hay que configurarlo para que Mosquitto se lance en cada inicio del servidor.

Este servidor supone el nodo central de nuestra topología de red, es por tanto el más determinante para el correcto funcionamiento del sistema. Si quisiéramos mejorar la disponibilidad del sistema para poder seguir utilizándolo aunque este servidor esté caído puede ser una mejora utilizar un segundo servidor similar a este al cual el resto de nodos tomaran por broker en caso de que el primero estuviera inaccesible.

### 4.6.1. Configuración de red

Para favorecer la localización del servidor central en la red es conveniente asignarle bien una IP estática (configurar alguna interfaz o modificar la configuración del cliente dhcp en Raspberry Pi) o bien utilizar algún DNS y que se actualice la dirección de la misma.

En este caso hemos asignado una IP estática al servidor dentro de nuestra red y hemos solicitado un nombre de dominio que actualiza la IP pública del router de acceso al servidor, configurando además un reenvío de puertos para que un cliente pueda conectarse desde el exterior.

### 4.6.2. Gestión de la base de datos

En el servidor central se encuentra situada una base de datos donde se recogen las lecturas de los enchufes. De este modo se puede gestionar la difusión de esta información a los clientes sin comprometer la consistencia de los datos.

La estructura de esta base de datos consiste en una tabla que incorpora para cada medición una fila con los datos del enchufe en el que se ha leído, el consumo y la hora registrada según el servidor.



## 4.7. Aplicación de escritorio

El último eslabón de nuestro sistema es el cliente de escritorio. Una aplicación que permite al usuario final interactuar con el resto del sistema.

Este cliente es en realidad una interfaz que otorga control sobre las funcionalidades del sistema: Por un lado, acceder a la información sobre el consumo eléctrico y visualizarla y, por otro lado, manipular los enchufes inteligentes para controlar su estado.

En la topología de nuestro diseño este tipo de nodo forma parte de los nodos periféricos que se conectan al servidor central. Pero a diferencia de los otros nodos periféricos, los enchufes inteligentes, estos permiten una mayor flexibilidad y libertad de configuración. Por esto se le permite al cliente modificar desde la aplicación los parámetros de conexión.

Una vez conectado al servidor la aplicación puede obtener la información del mismo. Consultar su base de datos para obtener el consumo realizado y poder visualizarlo de manera sencilla o consultar los registros en mayor profundidad.

Al obtener esta información también estamos consultando qué enchufes se encuentran conectados y qué identificadores tienen. Lo cual nos permitirá controlar el estado de los enchufes (modificando el estado de su relé) publicando en el broker en los temas de control a los que los enchufes estarán suscritos.

## 5. Pruebas y test

*Aquí se adjuntarán algunas imágenes y resultados de los tests que se realicen sobre distintas partes del sistema*

## **6. Conclusión**

Ya sabíamos que el Internet de las Cosas traía consigo innumerables posibilidades para extender la concepción habitual de los objetos que nos rodean en nuestro día a día.

Con este proyecto se pretende hacer ver que el Internet de las Cosas no es un concepto distante y que no existe una barrera económica o tecnológica que solo permita a grandes empresas trabajar con él. El Internet de las Cosas es accesible para cualquiera tanto para su uso como para su desarrollo. Apto para cualquiera sin requerir grandes gastos económicos y con unas opciones de diseño y desarrollo que existen y que cada vez están mejor documentadas.

## Referencias

- [1] IoT Agenda (website). What is internet of things (IoT)? <https://web.archive.org/web/20191107082047/https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>.
- [2] S. Vongsingthong, S.; Smachat. IoT examples: Suranaree Journal of Science and Technology. p5-17.
- [3] IEC. IEC - World Plugs: List view by potential. [http://www.iec.ch/worldplugs/list\\_byelectricpotential.htm](http://www.iec.ch/worldplugs/list_byelectricpotential.htm).
- [4] Nuevatribuna. El origen de las frecuencias eléctricas ¿Por qué 50 y 60 Hz? <https://www.nuevatribuna.es/articulo/ciencia/origen-frecuencias-electricas-50-y-60-hz/20150811110350118983.html>.
- [5] IEC - World Plugs: List view by location. [https://www.iec.ch/worldplugs/list\\_bylocation.htm](https://www.iec.ch/worldplugs/list_bylocation.htm).
- [6] Michael Weyrich and Christof Ebert. Reference Architectures for the Internet of Things. *IEEE Software*, 33:112–116, January 2016.
- [7] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805, October 2010.
- [8] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges. In *2012 10th International Conference on Frontiers of Information Technology*, pages 257–260, Islamabad, Pakistan, December 2012. IEEE.
- [9] James Temperton. TV white space will connect the internet of things. *Wired UK*, February 2015.
- [10] Christian Peter Pii Schumacher, Nandakishore Kushalnagar, and Gabriel Montenegro. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. <https://tools.ietf.org/html/rfc4919>.
- [11] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376, Fourthquarter 2015.
- [12] Web Services Architecture. <https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest>.
- [13] B. Bicsi. *Network Design Basics for Cabling Professionals*. McGraw-Hill Professional., (2002).
- [14] Andrew Banks and Rahul Gupta. MQTT Version 3.1.1. Edited by Andrew Banks and Rahul Gupta. 29 October 2014. OASIS Standard. <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>. Latest version: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>.
- [15] P.Th. Eugster, P.A. Felber, and R. Guerraoui. The Many Faces of Publish/Subscribe.
- [16] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. MQTT-S &#x2014; A publish/subscribe protocol for Wireless Sensor Networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, pages 791–798, Bangalore, India, January 2008. IEEE.
- [17] Eclipse Mosquitto. <https://mosquitto.org/>.
- [18] ESP32 Series Datasheet.
- [19] SCT-013-000 Datasheet PDF - Datasheet4U.com. <https://datasheet4u.com/datasheet-pdf/XiDiTechnology/SCT-013-000/pdf.php?id=1004703>.

- [20] ACS712 Datasheet(PDF) - Allegro MicroSystems. <https://www.alldatasheet.com/datasheet-pdf/pdf/168326/ALLEGRO/ACS712.html>.
- [21] Pablo Alcalde San Miguel. *Electrotecnia : Instalaciones Eléctricas y Automáticas p.424*. Paraninfo., Madrid, 6ª edición edition, (2014).