



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

MONITORIZACIÓN Y CONTROL DEL CONSUMO ELÉCTRICO DE SMART HOMES

Alumno: Juan Sánchez Santiago

Tutor: Antonio Cano Ortega

Co-Tutor: Francisco José Sánchez Sutil

Dpto: Departamento de Ingeniería Electrónica y
Automática

Departamento de Ingeniería Eléctrica



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Ingeniería Eléctrica

Don Antonio Cano Ortega y D. F.J. Sánchez Sutil, tutores del Trabajo Fin de Grado titulado: Monitorización y control del consumo eléctrico de smart homes, que presenta Juan Sánchez Santiago, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2021

El alumno:

Una firma manuscrita en tinta negra que parece decir "J. Sánchez".

Los tutores:

Juan Sánchez Santiago

Antonio Cano Ortega
Francisco José Sánchez Sutil

Resumen

El incremento de las tarifas eléctricas ha despertado el interés de los consumidores de energía tales como edificios comerciales y residenciales o incluso viviendas familiares a reducir su uso de energía. Este proyecto propone un dispositivo inteligente y un sistema de control central para controlar y reducir el uso de energía. El sistema está diseñado para que los enchufes inteligentes proporcionen de forma inalámbrica los datos de tensión, intensidad, potencia y consumo al usuario. Dichos datos se utilizan con el objetivo de analizar, monitorizar y controlar el consumo eléctrico de una vivienda y para generar comandos de control con los que encender o apagar los elementos conectados al dispositivo.

Abstract

The rise in electricity rates has sparked the interest of big energy consumers such as commercial and residential buildings or even family homes to reduce their energy use. This project proposes a connected device and a control system to manage and reduce the use of energy. The system is designed for smart plugs to wirelessly provide voltage, current power and consumption data to the user. These data are used in order to analyze, monitor and control the electrical consumption of a home and to generate control command to turn on or off the elements connected to the device.

Definiciones, acrónimos y abreviaciones

IoT Internet of Things

IDE Integrated Development Environment

PCB Printed Circuit Board

SNMP Simple Network Management Protocol

WSNs Wireless Sensor Networks

BMS Building Management System

HMI Human Machine Interface

RAM Random Access Memory

ROM Read Only Memory

SoC System on Chip

EPROM Erasable Programmable Read-Only Memory

PROM Programmable Read-Only Memory

EEPROM Electrically-Erasable Programmable Read-Only Memory

UAL Unidad Aritmético Lógica

NFC Near Field Communication

DIY Do It Yourself

USB Universal Serial Bus

API Interfaz de Programación de Aplicaciones

PVP Precio de venta al público

V Voltios

A Amperios

Hz Hercios

W Vatio

kWh Kilovatios-hora

mV mili-Voltios

mA mili-Amperios

MB MegaBytes

UART Universal Asynchronous Receiver Transmitter

GPIO General Purpose Input Output

GUI Graphical User Interface

AC Alternating Current / Corriente Alterna

DC Direct Current / Corriente Continua

URL Uniform Resource Locator

SDK Software Development Kit

HTTP Hypertext Transfer Protocol

MQTT Message Queue Telemetry Transport

GND Ground/Tierra

THT Through Hole

SMD Surface Mount Technology

DRC Design Rule Checking

LCD Liquid-Crystal Display

Índice

1. Introducción.....	9
1.1 . Motivación y objetivos.....	10
2. Estado del arte.....	12
2.1. Antecedentes.....	12
2.2. Microcontroladores.....	13
2.3. Tecnología actual.....	16
2.3.1 Arduino	16
2.3.2. ESP8266.....	18
2.3.2. Raspberry Pi.....	20
2.4. Sistemas de monitorización de energía y control	22
2.4.1. Engage Hub Kit de Efergy	23
2.4.2. Pack Wattio Energy de Wattio	24
2.4.3. Sistemas The OWL	25
2.4.4. FIBARO.	26
2.4.4. Leotec Smarthome Enchufe Inteligente	28
3. Requisitos.....	29
3.1. Requisitos	29
3.1.1. Requisitos Funcionales	29
3.1.2. Requisitos No Funcionales	30
4. Metodología y desarrollo.....	31
4.1. Especificaciones de Hardware Empleado	31
4.1.1. WeMos D1 Mini.....	31
4.1.2. Medidor multifunción PZEM 004T.....	34
4.1.3. Relé de 1 canal.....	37
4.1.4. Fuente de alimentación 5V.	39
4.2. Especificaciones del IDE y Software Empleados.....	39
4.2.1. Entorno de Programación de Arduino (IDE)	40
4.2.2. WiFiManager	45
4.2.3. Firebase.....	47
4.2.4. ThingSpeak.....	49

4.2.5. App Inventor	53
4.3. Desarrollo de la placa de circuito impreso (PCB)	59
4.3.1. Diseño de la PCB	60
4.3.2. Fabricación de la PCB	66
4.3.3. Montaje del prototipo.....	72
5. Pruebas Realizadas	74
5.1. Estudio con carga resistiva	74
5.2. Estudio con carga electrónica	80
5.3. Otros ensayos	82
6. Conclusiones	82
6.1. Cambios futuros.....	83
7. Bibliografía	84
8. Anexos	88
8.1. Anexo 1 – Presupuesto.....	88
8.2. Anexo 2 – Hojas de características.....	88
8.2. Anexo 3 – Esquemáticos.....	89

Lista de Figuras

Figura 1. Arquitectura Von Neuman	14
Figura 2. Arquitectura Harvard	15
Figura 3: Logotipo de Arduino.....	16
Figura 4: Plataforma de hardware libre Arduino Uno.....	17
Figura 5: Distribución de pines del ESP8266	19
Figura 6: Distribución e pines del ESP32	20
Figura 7: Raspberry Pi, marca registrada por Raspberry Pi Foundation	21
Figura 8: Raspberry Pi 4 Modelo B 4GB, lanzada en Junio de 2019	22
Figura 9: Raspberry Pi Zero 1.3	22
Figura 10: Engage Hub Kit de Efergy	23
Figura 11: Wattio Pack Energy	24
Figura 12: Familia de productos Intuition	25
Figura 13: Captura de pantalla de la plataforma web de The OWL.....	26
Figura 14: FIBARO Wall Plug.....	27
Figura 15: Software FIBARO Wall Plug	28
Figura 16: Leotec Smarthome Enchufe Inteligente.....	29
Figura 17: WeMos D1 Mini v1	32

Figura 18: WeMos D1 Mini v3.1.0.....	32
Figura 19: Pinout del WeMos D1 Mini	33
Figura 20: Medidor PZEM-004T con anillo toroidal cerrado.....	35
Figura 21: Esquema de conexiones del PZEM 004T	36
Figura 22: Pinout del SDIC SD3004 y su empaquetado físico	37
Figura 23: Esquema de contactos de un relé	38
Figura 24: Relé de un canal	38
Figura 25: Módulo fuente de alimentación de 5V	39
Figura 26: IDE de Arduino	42
Figura 27: Sketch de Arduino	43
Figura 28: Gestor de Librerías Arduino IDE	44
Figura 29: Sección de Github del ESP8266.....	44
Figura 30: Interfaz de WifiManager	46
Figura 31: Interfaz Configure Wifi de WifiManager	46
Figura 32: WifiManager en el gestor de librerías de Arduino	47
Figura 33: Logotipo Firebase	47
Figura 34: Variables de nuestra Realtime Database en Firebase.....	48
Figura 35: Localización del footprint de Firebase	49
Figura 36: Logotipo de ThingSpeak	49
Figura 37: Creación de un nuevo canal en ThingSpeak.....	51
Figura 38: API Keys de ThingSpeak.....	52
Figura 39: Logotipo de App Inventor.....	53
Figura 40: Nuevo proyecto de App Inventor.....	54
Figura 41: Configuración del componente Firebase en App Inventor	55
Figura 42: Menú de la app MIT App Inventor	56
Figura 43: Pantalla principal de nuestra aplicación	57
Figura 44: Pantalla de control del relé	58
Figura 45: Pantalla de control de reseteo del contador.....	59
Figura 46: Esquemático simple WeMos D1 Mini	60
Figura 47: Esquemático simple de los bornes y la fuente de alimentación.....	61
Figura 48: Esquemático simple del relé de un canal.....	61
Figura 49: Esquemático simple del medidor PZEM-004T	61
Figura 50: Esquemático simple de los bornes de alimentación	62
Figura 51: Esquemático total de conexionado de la placa.....	62
Figura 52: Esquemático del conexionado a tensión de red	63
Figura 53: Conectores de dos entradas para nuestra placa de circuito impreso.....	63
Figura 54: Footprints de los conectores WeMos D1 Mini y PZEM-004T respectivamente	64
Figura 55: Diseño del footprint de la fuente de alimentación de 5V con una dimensión de 30x20mm.....	64
Figura 56: Diseño del footprint del relé con una dimensión de 42x16mm.....	65
Figura 57: Diseño de nuestra placa PCB.....	66
Figura 58: Página web de JLCPCB.....	67
Figura 59: Creación del archivo GERBER en EasyEDA	68
Figura 60: Previsualización del acabado final de la PCB.....	69
Figura 61: Primera placa Top Layer	70

Figura 62: Primera placa Bottom Layer	70
Figura 63: Segunda placa Top Layer	71
Figura 64: Segunda placa Top Layer	71
Figura 65: Soldadura del prototipo	72
Figura 66: Prototipo Top Layer	73
Figura 67: Prototipo Bottom Layer	73
Figura 68: Dispositivo y multímetro conectados a un radiador eléctrico	74
Figura 69: Lecturas del dispositivo y multímetro con el radiador en nivel 1	75
Figura 70: Medidas para el nivel 1 del radiador	76
Figura 71: Lectura del dispositivo con el radiador en nivel 2	76
Figura 72: Medidas para el nivel 2 del radiador	77
Figura 73: Lectura del dispositivo con el radiador en nivel 3	78
Figura 74: Medidas para el nivel 3 del radiador	79
Figura 75: Lectura del dispositivo para la carga electrónica	80
Figura 76: Medidas para carga electrónica	80

Lista de Tablas

Tabla 1: Especificaciones Técnicas de WeMos D1 Mini	33
Tabla 2: Características para la selección de kit de desarrollo	34
Tabla 3: Especificaciones técnicas del PZEM 004T	36
Tabla 4: Carga resistiva bajo consumo	75
Tabla 5: Carga resistiva consumo medio	77
Tabla 6: Carga resistiva alto consumo	78
Tabla 7: Carga electrónica	81

1. Introducción

El presente trabajo de fin de grado toma la iniciativa de desarrollar un dispositivo para controlar el consumo eléctrico basándonos en el concepto de Smart Homes, o casas domóticas, que son una aplicación del Internet de las cosas (Internet of Things, IoT) con el fin de brindar confort, seguridad, eficiencia energética y comodidad en todo momento a sus usuarios, que acceden a la red para su control remoto, teniendo en cuenta el control y uso de las tarifas eléctricas. Esta tecnología se fundamenta en la Domotización de edificios la cual

permitirá que estos se vuelvan autónomos e inteligentes, existen otros campos interesantes en los que aplicarla como la medicina, la agricultura y el sector transportes.

Se prevé una evolución en la que cualquier cosa y objeto cotidiano están dotados de una conexión a Internet mediante procesos de inteligencia que relacionan los datos que estos dispositivos aportan con datos relativos a sus usuarios, haciendo cada vez más difusa la separación del mundo digital y el mundo físico.

Se trabajó con herramientas de desarrollo de prototipos hardware de fuentes abiertas, con las que fácilmente llevar a cabo el desarrollo de proyectos de electrónica destinadas a personas con escasos conocimientos en la materia.

1.1. Motivación y objetivos.

Hoy en día Internet es la herramienta más importante de la vida moderna, superior a cualquier otra tecnología creada por el hombre. Durante el breve período histórico que lleva instaurado en la sociedad, ha cambiado radicalmente la forma de comunicarnos entre nosotros y también, nuestra comunicación con las cosas. Es aquí donde entra el IoT, considerado como la gran evolución de internet desde que fue creado. Con el paso del tiempo aumentan más y más la cantidad de dispositivos que envían datos a la red y a su vez, se multiplican las tecnologías que aprovechan esta ingente cantidad de datos, aprendiendo de ellos y generando conocimiento que nos permite avanzar socialmente como nunca antes había ocurrido.

Las estadísticas muestran que proveer información a los usuarios sobre los usos y hábitos de consumo ayudan a reducir la gran problemática ambiental y económica que es el malgasto de energía eléctrica.

Para ayudar a resolver estos problemas, vamos a utilizar esta tecnología y llevar a cabo el desarrollo de un dispositivo de bajo coste, al cual conectaremos aparatos domésticos como por ejemplo lámparas, aire acondicionado, televisores... etc. Con el fin de reducir su tiempo de uso y mejorar su eficiencia,

reduciendo gastos y aumentando también así el tiempo de vida de los elementos, y accediendo en a él desde cualquier momento y en cualquier lugar.

El prototipo está basado en una placa de desarrollo con un módulo WifiESP8266 integrado y un PZEM004-T con el que realizaremos las medidas de consumo mediante métodos no invasivos, de manera que no necesita ningún tipo de instalación, con total autonomía. Todos los datos quedan almacenados en un servidor web.

Teniendo en cuenta esta propuesta procedemos a definir algunos objetivos a cumplir durante la realización del trabajo:

- Estudiar los conceptos asociados a las tecnologías que utilizaremos en el proyecto
- Analizar el hardware disponible en el mercado para el desarrollo de prototipos IoT, así como un estudio de mercado de los dispositivos comerciales.
- Aprendizaje del lenguaje y programación mediante el IDE de Arduino.
- Construcción e implementación de un dispositivo IoT basado en el hardware y elementos electrónicos mencionados anteriormente.
- Desarrollo de un sistema de almacenamiento de datos utilizando plataformas IoT y creación de una aplicación móvil que permita interactuar con el dispositivo a tiempo real y sea fácilmente accesible por el usuario
- Testeo del funcionamiento y las mediciones del dispositivo y comparación con las especificaciones de consumo del electrodoméstico.

- Diseño y desarrollo de la placa PCB en la que posteriormente soldaremos los distintos elementos y componentes

2. Estado del arte

2.1. Antecedentes

Desde el siglo XX, científicos como Nikola Tesla o Alan Turing ya pensaban sobre la idea de poder conectar los objetos y que estos fuesen inteligentes y autónomos.

En 1965, el investigador Lawrence G. Roberts conecta por primera vez dos ordenadores a distancia usando una línea telefónica de baja velocidad, dando lugar a la primera red de ordenadores de área amplia.

En los años 70 ya existían soluciones basadas en protocolos X-10 que permitían comunicar sensores y actuadores de una vivienda, comunicándose a través de la línea eléctrica. Con el paso del tiempo han aparecido protocolos y sistemas más complejos y más robustos, con características adecuadas para cada campo.

En 1990 John Romkey, en el evento Interop en EEUU, crea el primer objeto conectado a Internet: una tostadora podía encenderse o apagarse remotamente. La conectividad fue realizada con el protocolo TCP/IP y el control se realizó mediante SNMP

El término Internet of Things es más reciente, instaurado por Kevin Ashton en el año 2009 señaló que toda la información que manejaban las computadoras, incluido Internet dependía de los seres humanos. El IoT puede ser considerado una infraestructura de redes tanto dinámicas como globales que gestiona elementos auto configurables de una manera inteligente.

Hoy en día, la noción de IoT ha evolucionado hasta la idea de crear una infraestructura de redes interconectadas física y virtualmente e incluir otras tecnologías como Cloud Computing y Wireless Sensor Networks (WSNs).

También los sistemas de gestión de edificios conocidos como BMS están formados por grupos de sensores, actuadores, controladores, pasarelas, buses de campo y una aplicación o programa que actúa de interfaz hombre-máquina (HMI) para facilitar su gestión.

2.2. Microcontroladores

Un microcontrolador está formado por circuito integrado programable capaz de ejecutar las órdenes almacenadas en su memoria. Está compuesto por varios bloques funcionales que cumplen una tarea específica.

Su inicio proviene del desarrollo tecnológico sobre los circuitos integrados, dicho desarrollo ha hecho posible contener miles de transistores en un solo chip, un requisito para producir los microprocesadores. Las primeras computadoras fueron construidas añadiendo elementos tales como memorias, contadores, dispositivos de entrada/salida, etcétera. No obstante, el posterior aumento de la densidad de integración nos permitió fabricar circuitos integrados que contenían el procesador y los periféricos. Así se desarrolló la primera microcomputadora en un chip, llamado más tarde microcontrolador.

Los ingenieros de *Texas Instruments* Gary Boone y Michael Cochran lograron crear el primer microcontrolador, TMS 1000, en 1971, el cual fue comercializado en 1941. Este microcontrolador combinaba memoria RAM, memoria ROM y un reloj en un chip destinado a sistemas embebidos.

En ese momento la mayoría de microcontroladores tenían dos variantes. Una memoria EPROM (*Erasable Programmable Read-Only Memory*) programable pero mucho más caras que las PROM (*Programmable Read-Only Memory*), que eran programable una sola vez. Para poder borrar la EPROM es necesario exponer a luz ultravioleta la tapa de cuarzo transparente.

En 1993, se lanzó la memoria EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) en los microcontroladores (empezando por el Microchip PIC16x84), las cuales podían borrarse de manera eléctrica y sin

grandes costes, lo cual permitió el rápido desarrollo de prototipos y una programación más fácil en los microcontroladores. El mismo año la compañía Atmel lanzó el primer microcontrolador con memoria flash.

Existen dos arquitecturas básicas desde el punto de vista de arquitectura de computadores: Von Neuman y Harvard, las cuales están diferenciadas en la forma de conectarse a la memoria del procesador y en los buses que necesitan cada uno.

- **Arquitectura Von Neuman**

Desarrollada por el matemático John Von Neuman, nacido en Budapest el 28 de diciembre de 1903, establece una forma de almacenar el programa en la memoria junto con los datos. Cuenta con una memoria principal, que trabaja a mayor velocidad, constituida por memorias RAM y ROM. Una memoria auxiliar/secundaria más lenta, pero de mayor capacidad. Una Unidad Aritmético-Lógica (UAL o ALU) encargada de realizar operaciones aritméticas y lógicas y una Unidad de Control que detecta señales de estado procedentes de distintas partes del ordenador y genera señales de control dirigidas a todas las unidades para controlar el funcionamiento de la máquina.

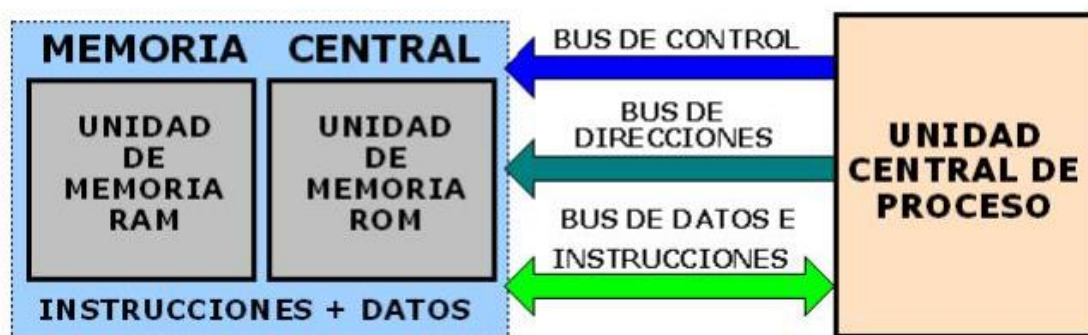


Figura 1. Arquitectura Von Neuman

En este caso, la memoria está dividida y el acceso a cada tipo de memoria depende de las instrucciones del procesador.

Las familias de microcontroladores como Intel 8051 y Z80 implementan este tipo de arquitectura, principalmente porque era la utilizada cuando surgieron los primeros microcontroladores.

- **Arquitectura Harvard**

El término proviene de la computadora Harvard Mark I, basada en relés, que almacenaba las instrucciones sobre cintas perforadas y los datos en interruptores electromecánicos.

La arquitectura utilizada en los microcontroladores y sistemas integrados en general. Además de la memoria, el procesador posee los buses segregados de tal modo que cada tipo de memoria tiene un bus de datos, uno de direcciones y uno de control.

La ventaja primordial de esta arquitectura es que permite adecuar el tamaño de los uses a las características de cada tipo de memoria.

El procesador puede acceder a cada una de ellas simultáneamente, lo cual incrementa significativamente la velocidad de procesamiento. Este tipo de procesamiento otorga a los sistemas el doble de velocidad que los sistemas similares con la arquitectura Von Neuman

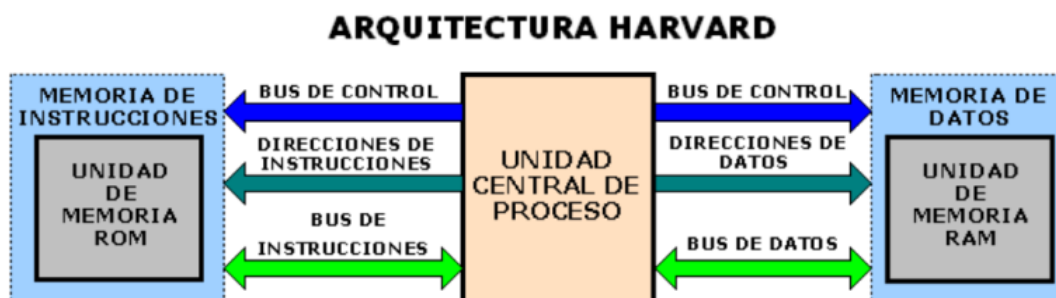


Figura 2. Arquitectura Harvard

Actualmente, el coste de los microcontroladores ha ido disminuyendo rápidamente, actualmente son muy baratos y disponibles para desarrolladores aficionados. Muchos de ellos pertenecen a grandes comunidades en línea, con multitud de librerías, ejemplos y tutoriales.

2.3. Tecnología actual

Una parte muy importante a la hora de desarrollar un producto IoT es la elección de una placa de desarrollo, es decir, el dispositivo hardware electrónico desde se implementan los diferentes sensores, actuadores y la unidad de procesamiento que se encargará de la lógica de nuestra aplicación. Se denomina dispositivo conectado o inteligente, un dispositivo generalmente conectado a otros dispositivos o redes mediante distintos protocolos como Bluetooth, NFC, Wi-Fi, 5G, X10, Xbee, LoRa o NB-IoT cuya característica principal es su funcionalidad, hasta cierto punto, de forma autónoma e interactiva.

Existen muchos tipos de dispositivos en forma de placa para llevar a cabo estas tareas, pero vamos a destacar algunas de las más conocidas.

2.3.1 Arduino

Arduino es un entorno de desarrollo gratis, libre y multiplataforma (ya que funciona en Linux, MacOS y Windows) que ha adquirido una gran popularidad en los últimos años, contando con una amplia comunidad internacional y la propia empresa encargada de diseñar y manufacturar sus propias placas de desarrollo hardware con el fin de construir dispositivos inteligentes, capaces de monitorizar y controlar objetos del mundo real de una manera sencilla y accesible para todos. Las placas *Arduino* se encuentran disponibles comercialmente en forma de placas ensambladas o también en forma de kits DIY (*Do It Yourself*).



Figura 3: Logotipo de Arduino

No obstante, cuando hablamos de “placas *Arduino*”, deberíamos especificar el modelo concreto, ya que existen varias placas *Arduino* oficiales, cada una con diferentes características (tamaño físico, número de pines-hembra ofrecidos, el modelo de microcontrolador incorporado, cantidad de memoria utilizable, etc.). Conviene conocer estas características para identificar qué placa *Arduino* es la que nos convendrá más en cada proyecto. Estas placas normalmente incluyen un microcontrolador *Atmel AVR*, sobre las cuales cabe la posibilidad de añadir placas de expansión (*Shields*) a través de los distintos puertos de entrada y salida presentes en la placa base. Dichos *Shields* complementan el funcionamiento del modelo de *Arduino* incorporando circuitería, sensores y módulos de transmisión externos a la placa principal. Suelen estar alimentados por un puerto USB y ser programados a través del puerto serie incorporado, utilizando el *Bootloader* incluido por defecto.

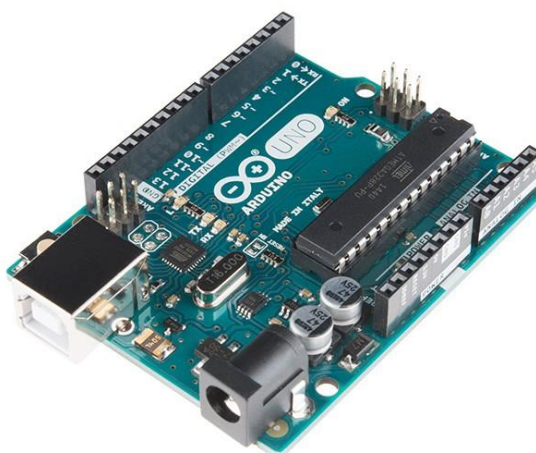


Figura 4: Plataforma de hardware libre *Arduino Uno*

Para que el trabajo con las placas sea más intuitivo, *Arduino* nos ofrece su entorno de desarrollo integrado (IDE) con el cual es posible programar sus placas fácilmente. Este IDE también permite programar la placa conectada al puerto USB, permitiéndonos descargar y agregar librerías y componentes establecidos por *Arduino* de forma oficial, también tiene incorporado su propio gestor de placa,

el cual nos permite descargar frameworks para poder programar distintas placas de desarrollo propias de *Arduino*.

En este proyecto utilizaremos la placa Arduino Wemos D1 mini, sobre la que hablaremos más adelante.

2.3.2. ESP8266

El ESP8266 es un chip lanzado al mercado en 2014 con un módulo ESP-01, pensado para que actúe como adaptador de puerto serie a *Wifi* por un módico precio. Desarrollado por *Espressif Systems*, empresa china con sede en Shanghai que consiguió una gran popularidad dándose a conocer como alternativa a los otros shields para *Arduino*, que eran de mayor coste.

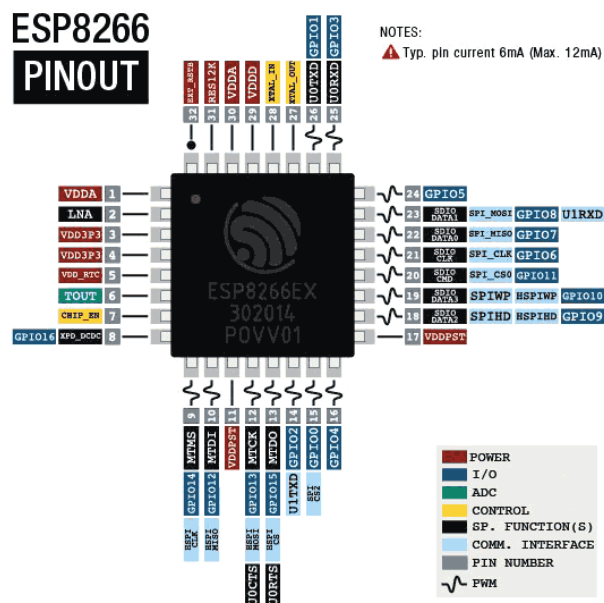
Debido a las peticiones de la comunidad de desarrolladores que creían en su potencial como MCU independiente, tras meses de su lanzamiento, la empresa fabricante proporcionó una API para desarrollar directamente sobre su microprocesador, permitiendo emplear de este modo el chip de manera independiente. Pero no terminamos aquí, la comunidad que utilizaba el *ESP8266* para sus proyectos logró adaptar su *API* para poder utilizarse en el entorno de *Arduino*, permitiendo a los desarrolladores programar en el *ESP8266* la capacidad de utilizar las librerías y el conocimiento acumulado de *Arduino*.

Debido a estas aportaciones, actualmente existen distintos *SDK* de código abierto para el *ESP8266*, entre los cuales encontramos un firmware basado en C++ para *Arduino*, permitiendo que el *MCU* del *ESP8266* y los componentes sean fácilmente programados como otro dispositivo *Arduino* a través de su *IDE*.

Tiene las siguientes características:

- Procesador RISC Tensilica Xtensa LX106 de 32-bit con una frecuencia de 80 MHz ampliable hasta 160 MHz RISC
- RAM de instrucción de 64 KB
- RAM de datos de 96 KB
- Capacidad de memoria externa flash QSPI hasta 16 MB

- Wifi estándar IEEE 801.11 b/g/n
- 16 pines GPIO
- Alimentación de 3.3V.



limitaciones que causan algunas restricciones para su uso en desarrollos más generales ya que su propósito inicial era el de servir como adaptador de serie a *WiFi* con un bajo coste. Debido a esto, Espressif comenzó en 2016 a trabajar en un producto con nuevas características debido a la popularidad que había conseguido el ESP8266. A finales de ese mismo año se anunció el lanzamiento de una versión preliminar del chip ESP32, y en septiembre de 2016 lanzaron la versión definitiva, la cual incluía una mejora en sus características lo cual, obviamente, provocó un sobrecoste con respecto al original

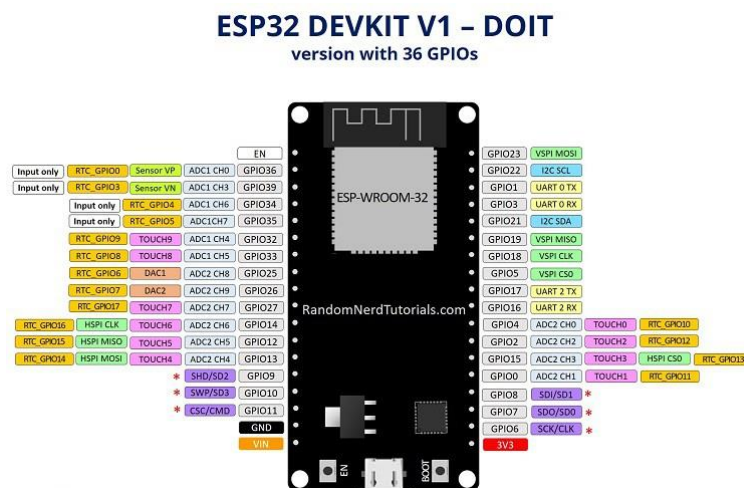


Figura 6: Distribución e pines del ESP32

2.3.2. Raspberry Pi

La Raspberri Pi es una minicomputadora diseñada y fabricada con el fin de proporcionar a los más pequeños herramientas asequibles y sencillas con las que poder aprender y divertirse a la vez que programas. Debido al enfoque dado, el dispositivo fue formado con una placa base del tamaño de una tarjeta de crédito la cual integra los componentes esenciales de un ordenador (memoria RAM, procesador, etc.) y posee conexiones de entrada/salida para conectar un ratón, un teclado y un monitor, lo que hace que su precio se más reducido. Aunque su capacidad de almacenamiento está limitada al de una tarjeta SD en la que va instalado el sistema operativo.

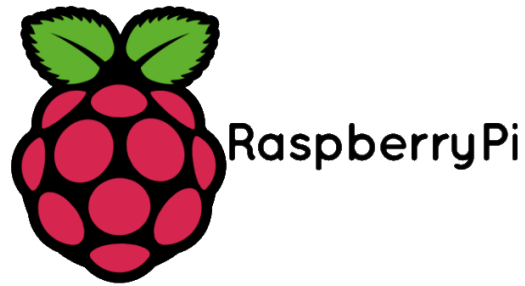


Figura 7: Raspberry Pi, marca registrada por Raspberry Pi Foundation

Otra de las múltiples ventajas ofrecidas por la Raspberry Pi es la posibilidad de instalar una gran variedad de sistemas operativos de fuentes abiertas, las cuales suelen ser distribuciones de Linux optimizadas a la plataforma. Por esto es habitual empezar a programar lenguajes como Python, debido a la sencillez de su lenguaje y a que viene preinstalado un intérprete de dicho lenguaje. Además, cuenta con conexión a Internet, terminales de entrada y salida digitales que lo convierten en un dispositivo idóneo para integrarlo en proyectos de electrónica donde se comuniquen muchos sensores y actuadores entre sí y se transmiten datos a través de la red.

Algunos modelos lanzados al mercado en los últimos años son:

- Raspberry Pi 2 Modelo B
- Raspberry Pi 3 Modelo B, B+ y A+
- Raspberry Pi 4 Modelo B
- Raspberry Pi Zero.



Figura 8: Raspberry Pi 4 Modelo B 4GB, lanzada en Junio de 2019

Las Raspberry Pi Zero son una gama de placas más pequeñas y algo menos potentes que las anteriores, especialmente indicadas para desarrollar productor IoT. En concreto los modelos Raspberry Pi Zero y Pi Zero W (Wireless) que incluyen *WiFi* y *Bluetooth*.

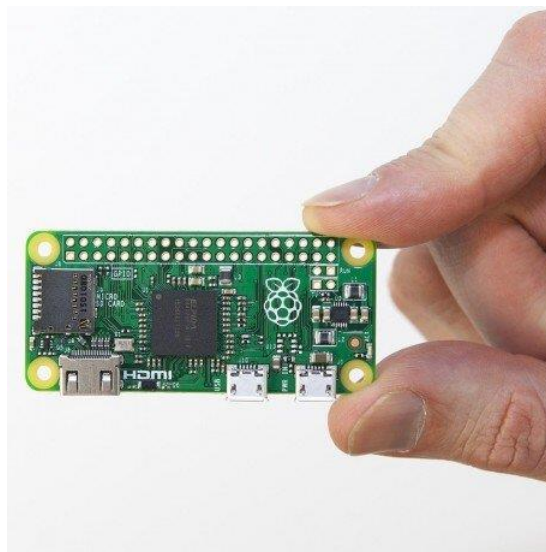


Figura 9: Raspberry Pi Zero 1.3

Como ya comentamos en el caso de Arduino, existe un gran soporte en la comunidad de usuarios unidos por el bajo precio y comercialización tan extensa que provoca que muchas personas se decanten por esta minicomputadora.

2.4. Sistemas de monitorización de energía y control

Existen gran multitud de soluciones para la problemática propuesta, soluciones enchufables (productos denominados “Plug & Play”), soluciones de enchufes empotrables a la pared, como el que vamos a diseñar, o en forma de regleta. Algunos de los más conocidos en el mercado son Open Domo, Current Cost, Circutor, Cliensol Energy, Wattio Energy y OpenEnergyMonitos.

En esta sección se van a analizar algunos productos con prestaciones similares a los requeridos por este proyecto

2.4.1. Engage Hub Kit de Efergy

Efergy son unos de los fabricantes globales de productos de ahorro energético, con el fin de reducir el consumo eléctrico, ahorrar dinero y ayudar a conservar el medio ambiente.

Su producto principal es el “Engage Hub Kit”. Producto compuesto por un sensor no invasivo conectado al cable de fase del cuadro eléctrico y al transmisor de datos en su otro extremo. El transmisor se encuentra conectado de forma inalámbrica a un receptor denominado “hub”, conectado a Internet mediante un cable Ethernet. El equipo reúne datos cada 6, 12 ó 18 segundos, según quiera el cliente. Dichos datos se suben a una web en la que es posible consultarlos y gestionarlos desde cualquier dispositivo. Tiene un precio de 99.90 euros.



Figura 10: Engage Hub Kit de Efergy

- **Ventajas:** Buen diseño web y de aplicaciones, las cuales funcionan en Android e IOS, y un sistema que permite compartir el consumo con amigos, siendo uno de los más económicos del mercado.
- **Inconvenientes:** Tiene una gran cantidad de elementos y es necesario enviar la señal a otro dispositivo, utiliza baterías en el emisor y su conexión a Internet se realiza mediante cable Ethernet. Solo es posible monitorizar una señal de consumo en todo el sistema.

2.4.2. Pack Wattio Energy de Wattio

Producto también enfocado a la monitorización de la corriente eléctrica. Este producto es uno de los más completos que puedes encontrar en el mercado. Compuesto por una serie de dispositivos de recogida de datos, gestionados por una consola domótica con pantalla táctil, o accediendo a la aplicación de *Wattio* desde un ordenador o cualquier otro dispositivo con conexión a Internet.



Figura 11: Wattio Pack Energy

Con un precio de 299,90 euros, en el pack vienen incluidos: centralita, enchufe inteligente, termostato y medidor de electricidad.

- **Ventajas:** Sistema muy completos, el enchufe inteligente mide el consumo individual de dispositivos, con medidor de electricidad de tres

circuitos para el cuadro eléctrico, centralita táctil y aplicaciones para dispositivos móviles.

- **Inconvenientes:** Necesita una centralita, la conexión entre dispositivos es de unos treinta metros y utiliza baterías para sistemas enlazados. Es un sistema de un elevado coste.

2.4.3. Sistemas The OWL

OWL es una marca registrada con la cual se comercializan en Reino Unido una gran cantidad de productos destinados al ahorro eléctrico en el ámbito doméstico. Esta empresa basa gran parte de su oferta en el desarrollo de multitud de modelos de monitores de energía, por lo que se deduce que los usuarios aprecian el tipo de información que proporcionan dichos sistemas a partir de los datos proporcionados del consumo energético.



Figura 12: Familia de productos Intuition

Prestamos especial atención en una familia de sistemas que la compañía ha producido posteriormente debido a la evolución de la tecnología en sus monitores energéticos. De tal forma, se han analizado las características del sistema *Intuition-e*, el cual incluye el mismo sistema de recogida de datos característico para la medición del consumo de la toma de corriente en las viviendas. Como novedad incluye un receptor de radio que, conectado a un router de Internet, manda los datos de consumo a una plataforma *cloud* de la

empresa, en la cual el usuario se beneficia de una mayor comodidad de acceso a la información a la vez que de una mayor configuración del servicio.



Figura 13: Captura de pantalla de la plataforma web de The OWL

Se puede apreciar una gran similitud entre este servicio y el que ofrece la empresa *Efergy* a través del mencionado *Engage Hub*. Aun así, hay disponibles diversos productos que permiten ampliar la funcionalidad de la plataforma y poder personalizarla, incluyendo información sobre el rendimiento si la instalación cuenta con paneles fotovoltaicos o integrando monitores de temperatura. Se considera que este tipo de plataformas digitales pueden experimentar un mayor crecimiento al incorporar más información y nuevas funcionalidades a los propietarios de estos sistemas de medida.

Puede encontrarse en el mercado por un precio que ronda los 110 euros.

2.4.4. FIBARO.

FIBARO es un sistema domótico completo, integra múltiples soluciones para la automatización del hogar de forma modular, de manera que los compradores pueden instalar un sistema personalizado que contenga las soluciones que más les interesen. Además, el fabricante afirma que los módulos que componen el servicio no necesitan modificar la instalación eléctrica del domicilio para ser instalado, sino que puede ser conectado sin necesidad de obra

en el hogar y consta de conexiones inalámbricas, ya que la información es transmitida mediante el estándar de comunicaciones inalámbricas Z-Wave.

El sistema central del sistema es un ordenador que recibe inalámbricamente la información de los sensores instalados, esta es almacenada y proporciona al usuario una interfaz web en la que no solo puedes monitorizar, sino también controlar el hogar. A partir de esta plataforma son desarrolladas numerosas aplicaciones domóticas como controles de iluminación, sensores de humo, sensores de movimiento, cámaras de vigilancia etc. Para este sistema se ha desarrollado un módulo con el que gestionar el consumo eléctrico que vamos a ver a continuación.



Figura 14: FIBARO Wall Plug

El FIBARO *Wall Plug* es un adaptador de enchufe al cual se conectan los aparatos eléctricos que desean ser gestionados. Uno de sus características que más llama la atención de este adaptador de corriente es un anillo luminoso que adorna la parte frontal, el cual cambia de color según la potencia instantánea que esté midiendo. Como podemos apreciar, el modo de instalación del sensor es distinto al empleado en la mayoría de sistemas revisados previamente.

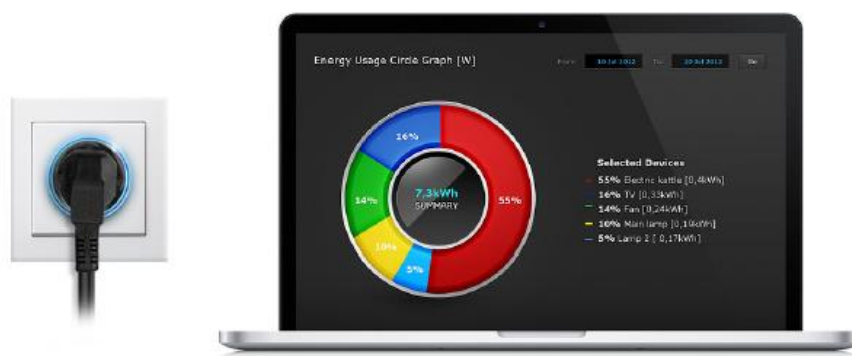


Figura 15: Software FIBARO Wall Plug

Este componente envía la información registrada al ordenador central gracias al mecanismo inalámbrico de transmisión que mencionamos anteriormente, donde combina datos provenientes de los otros adaptadores eléctricos en el hogar y proporciona al usuario una información de consumo muy variada. En este servicio se pueden consultar estadísticas de consumo discriminadas por tipo de electrodoméstico, lo cual supondría una evolución a la hora de que el usuario establezca nuevos hábitos de ahorro energético. Otro aspecto a resaltar es que los adaptadores contienen un interruptor que puede ser accionado remotamente, lo cual otorga distintas aplicaciones de control, permitiéndonos apagar y encender automáticamente ciertos electrodomésticos que superen un umbral de consumo previamente establecido.

El FIBARO Wall Plug tiene un precio alrededor de los 50 euros.

2.4.4. Leotec Smarthome Enchufe Inteligente

Es una toma de corriente que se puede controlar de forma remota desde un Smartphone o Tablet. Permite visualizar de forma gráfica el consumo energético en tiempo real del aparato que haya conectado al mismo, así como acceder a un completo historial de consumos. Además, permite programar el encendido y/o apagado de forma remota. Si el usuario se encuentra fuera de casa, le permite simular su presencia, encendiendo remotamente radio, televisión, luces, etc. Tiene un precio de 30 euros.



Figura 16: Leotec Smarthome Enchufe Inteligente

3. Requisitos

Los requisitos del proyecto muestran una clara visión de las necesidades y expectativas contenidas en este proyecto. En este apartado vamos a exponer los requisitos necesarios para poder llevar a cabo el proyecto.

3.1. Requisitos

En esta sección explicaremos los requisitos necesarios para la elaboración del proyecto. Abordaremos los requisitos fundamentales, los cuales definen una función en el sistema de software o sus componentes y los requisitos no funcionales, que son los que definen un atributo de calidad, es decir, características del funcionamiento.

3.1.1. Requisitos Funcionales

- **1. Lectura del consumo energético:** El usuario puede colocar el dispositivo en cualquier toma de corriente, y en el otro extremo, conectar el elemento del cual desea medir en tiempo real el consumo y su control de estado.
- **2. Configuración de la comunicación:** El usuario debe configurar una conexión *Wi-Fi* en el dispositivo para enviar los datos.

- **3. Creación de una cuenta en una base de datos a tiempo real:** Se deberá crear una cuenta en la base de datos a tiempo real para realizar las lecturas e interactuar con el estado del dispositivo.
- **4. Creación de una cuenta en una plataforma web IoT:** El usuario debe crear una cuenta en la plataforma web IoT utilizada para la monitorización del consumo energético.
- **5. Creación de una cuenta en un entorno de desarrollo de aplicaciones:** Tenemos que crear una cuenta en un entorno de desarrollo software para crear una aplicación móvil con la que poder interactuar con nuestro prototipo.
- **6. Envío de datos a la plataforma:** La plataforma IoT elegida permitirá el envío de un número suficiente de mensajes para la funcionalidad requerida
- **7. Consulta de consumo:** el usuario podrá consultar el consumo instantáneo en la aplicación móvil y en la plataforma web IoT cuando lo desee. Además, se proporcionará un histórico para un control más visual del consumo.

3.1.2. Requisitos No Funcionales

- **1. Seguridad en la comunicación:** El sistema posee un procedimiento de cifrado para comunicarse con la base de datos.
- **2. Seguridad en el acceso a la plataforma:** El sistema tendrá un procedimiento de seguridad para el acceso a la plataforma
- **3. Usabilidad:** Los dispositivos, la plataforma web y la aplicación móvil serán sencillos e intuitivos para el posterior uso de cualquier usuario.
- **4. Bajo coste:** Requisito primordial en el proyecto. Tanto el dispositivo desarrollado en este proyecto, incluyendo microcontrolador y los medidores, etcétera, serán de bajo coste
- **5. Fiabilidad y disponibilidad:** La base de datos tiene que ser fiable y debe estar disponible en el momento que el usuario desee.

4. Metodología y desarrollo

En este apartado abordaremos las herramientas utilizadas en el proyecto y se explicará el funcionamiento de cada una de ellas, justificando por qué han sido utilizadas en este proyecto.

4.1. Especificaciones de Hardware Empleado

Para llevar a cabo el diseño y la implementación de este dispositivo se han utilizado cuatro elementos fundamentales: la placa WeMos, cuyo microcontrolador está encargado de leer y comunicar los datos con la plataforma *Thingspeak* y la base de datos *Firebase*; el medidor de voltaje, potencia y energía denominado PZEM-004T; el relé que nos permitirá controlar el estado del dispositivo; y un convertidor de tensión que utilizaremos como fuente de alimentación para que al conectar el dispositivo a la toma de corriente, este sea totalmente autónomo.

4.1.1. WeMos D1 Mini.

Dentro de las variantes de placas que incorporan el ESP8266 se encuentra a uno de los fabricantes de este hardware libre “NodeMCU” como es LOLIN (como se conocía a los NODEMCU de 3ª generación, actualmente llamados WeMos).

El fabricante creó su propio diseño mejorado a partir de la placa de desarrollo previamente descrita, creando el NodeMCU v3. En esta placa fue reemplazado el conversor Serial-USB CP2102 por un CH340G, que establecía un puerto USB más robusto.

Las especificaciones eran iguales, con el inconveniente del tamaño de la placa, la cual era más ancha que sus predecesoras y en una placa Board no era capaz de dejar libres hileras de pines para realizar montajes.

Tras esto se redefinió la disposición de la placa para subsanar el defecto, y se creó lo que ahora conocemos como WeMos D1 Mini.

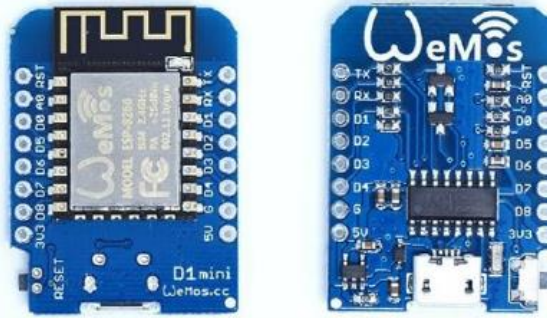


Figura 17: WeMos D1 Mini v1

Así, se consiguió diseñar una placa de desarrollo que contuviera el SoC ESP8266-12E, tan potente como el NodeMCU y que solucionara la problemática de los pines en la placa Board, así como conseguir unas dimensiones menores a las del NodeMCU.



Figura 18: WeMos D1 Mini v3.1.0

En cuanto a especificaciones técnicas contamos con 9 pines GPIO (configurables como PWM a 3,3V, comunicación con PC e interrupciones), 1 pin analógico (máximo 3,3V), conexión micro-USB, comunicación serial, botón RESET y alimentación tanto de 5V como de 3,3V, gracias a su regulador de voltaje.

Características	Valor
Modelo	D1 mini
Microcontrolador	ESP8266
Voltaje de funcionamiento	3.3V

Voltaje de entrada	5V
Entradas/salidas digitales	11
Entradas analógicas	1
Memoria Flash	4Mb
SRAM	1Kb
EEPROM	512 bytes
Velocidad de reloj	80MHz/160Mhz

Tabla 1: Especificaciones Técnicas de WeMos D1 Mini

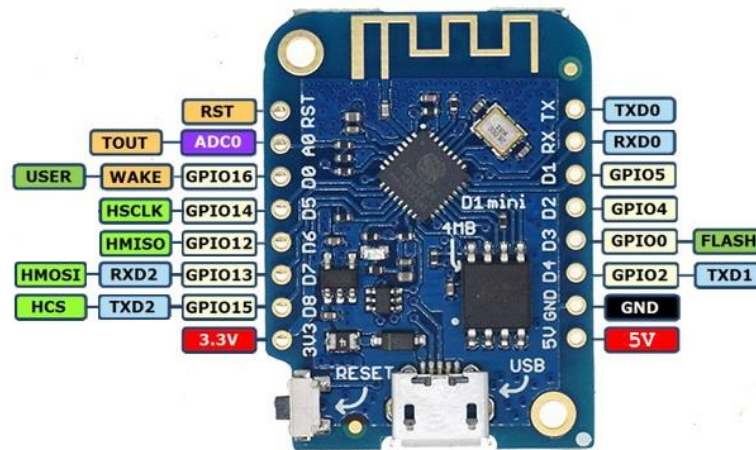


Figura 19: Pinout del WeMos D1 Mini

En la figura 18: Pinout del WeMos D1 Mini se muestra el conexionado de pines con lo que cuenta la placa de desarrollo.

Además, posee una CPU de 32 bits, una frecuencia de reloj de 80 MHz (máximo 160 MHz) y una memoria Flash externa de 4MB.

En la siguiente tabla, vamos a ilustrar las características más importantes de las placas presentadas

	Procesador	WiFi	Entradas Analógicas	Tamaño (mm)	Precio (€)
ESP82-66	32 bits	Sí	1	24 x 16	1,00 – 3,00
Arduino MRK1000	32 bits	Sí	7	61,5 x 25	31,00
NodeMCU	32 bits	Sí	1	49 x 24,5	2,00 – 4,00
WeMos D1 Mini	32 bitts	Sí	1	34,2 x 25,6	2,00 – 3,00

Tabla 2: Características para la selección de kit de desarrollo

Por tanto, tras realizar un balance, la elección del WeMos D1 Mini fue a causa de su sistema integrado, a su reducido tamaño, a su bajo coste, de unos 3,80 euros y a que tiene incorporado el chip ESP8266. WeMos D1 Mini trabaja a 5V y 3.3V, lo que es suficiente y adecuado para el resto de elementos del proyecto. Otro aspecto a resaltar para su elección es el consumo, ya que es muy bajo comparado con otras placas también basadas en el ESP8266 y es compatible con la IDE de Arduino.

4.1.2. Medidor multifunción PZEM 004T.

Este medidor tiene grandes prestaciones en aplicaciones de medición de consumo eléctrico a nivel *Do It Yourself*, perfecto para proyectos básicos de análisis de consumo eléctrico a baja escala, a nivel residencial o domiciliario, y es perfecto para integraciones con plataformas IoT, así como fácilmente programable en el entorno IDE de *Arduino*.



Figura 20: Medidor PZEM-004T con anillo toroidal cerrado

Se encuentra en dos formatos:

- Con anillo toroidal cerrado, lo que obliga a desconectar un cable para poder monitorizar el circuito, este es el que se utilizará.
- Con anillo toroidal de pinza, más cómodo, pero un poco más caro.

Estos dispositivos se comunican por el puerto serie con Arduino mediante el mismo protocolo UART con el que intercambia datos Arduino con el ordenador. Por tanto, necesitamos dos pines digitales, uno actuará como receptor de datos (Rx) y otro como transmisor (Tx).

Es importante destacar que Arduino utiliza los puertos digitales 0 y 1 como Tx y Rx para comunicar con el PC, por tanto, si conectamos otros dispositivos a estos puertos, no podremos cargarle programas. Por este motivo, se eligen otros pines distintos y mediante una librería, se les asigna la funcionalidad de puerto serie virtual.

Se pueden conectar varios medidores simultáneamente a la misma placa utilizando el puerto serie como bus de datos. Esto implica que cada dispositivo deba tener un nombre (dirección individual) único, para que Arduino pueda diferenciar de dónde procede cada dato. Pero en este caso solo utilizaremos uno.

Para su conexión, encontramos dos circuitos principales:

- Circuito de alimentación de la placa y comunicación con Arduino
- Circuito de toma de medidas eléctricas en AC



Figura 21: Esquema de conexiones del PZEM 004T

Las conexiones se realizan de igual forma que si fuéramos a realizar las mediciones con una pinza amperimétrica. Para medir tensión se conecta el PZEM en paralelo con los puntos en los que queremos realizar la medida, y para medir intensidad, se pasa uno de los cables del circuito a medir a través del anillo toroidal. Si en lugar de anillo, tenemos la pinza, no hace falta desconectar el circuito principal para meter el conductor a través del anillo, basta con abrir la pinza.

Entre sus especificaciones técnicas, nos encontramos con un medidor no invasivo en el circuito eléctrico, con un pequeño tamaño adecuado para espacios reducidos y que es capaz de medir tensión, intensidad, potencia y energía.

Función	Rango de medida	Resolución	Precisión
Voltaje	80 ~ 260V	0,1V	0,5%
Corriente	0 ~ 100 A	0,02A	0,5%
Potencia instantánea	0 ~ 23kW	0,1W	0,5%
Potencia acumulada	0 ~ 9999,99kWh	1Wh	0,5%
Frecuencia	45~65Hz	0,1Hz	0,5%
Factor de potencia	0,00 ~ 1,00	0,01	1%

Tabla 3: Especificaciones técnicas del PZEM 004T

Contiene una memoria externa EEPROM I^2C que almacena los kWh y en caso de apagar el medidor, no se reinicia el contador de consumo. El integrado principal es un SDIC SD3004 que realiza técnicamente todo: mediciones, conversiones, cálculos y comunicación serial.

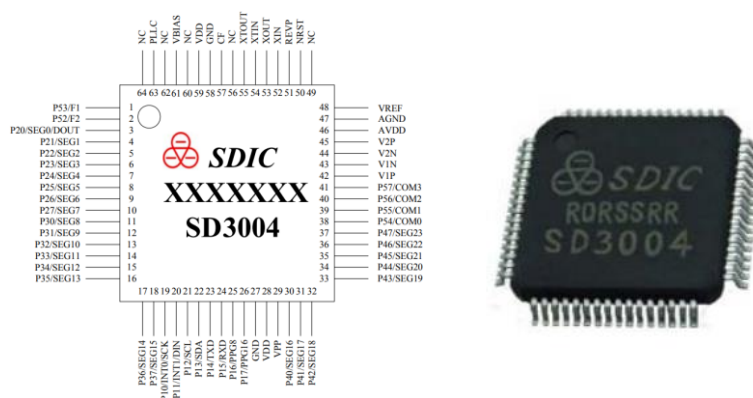


Figura 22: Pinout del SDIC SD3004 y su empaquetado físico

Como podemos observar, es un elemento muy versátil dado su precio reducido, por lo que se tomó la decisión de incorporar este elemento al proyecto, el cual se ha encontrado a un precio de 3,80 euros.

Algo muy importante a tener en cuenta es que este dispositivo está conectado a una toma de corriente de nuestro domicilio, lo cual es muy peligroso, por lo que se requiere tener un extremo cuidado durante su instalación y manejo.

4.1.3. Relé de 1 canal

Un relé es un dispositivo electromecánico, el cual nos permite controlar elementos conectados a grandes voltajes, como por ejemplo, una lámpara de casa.

Por así decirlo, un relé se comporta como un interruptor convencional, pero siendo este tipo de interruptor uno que podemos controlar tanto en tiempo real como programado.

Un relé se puede descomponer en dos circuitos (primario y secundario), el circuito primario es aquel que se conecta con la electrónica a baja tensión, para recibir la señal de encendido o apagado. Por otro lado, tenemos el circuito secundario donde tendremos la carga conectada, cuyo fin es según el estado del circuito primario, conmutar los contactos del relé.

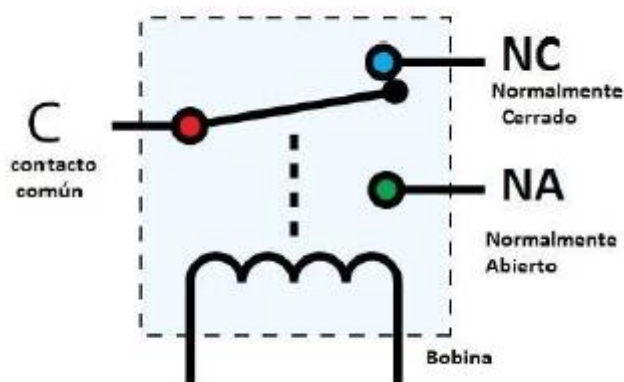


Figura 23: Esquema de contactos de un relé

Estos relés suelen ser muy habituales en automatización de procesos, también son muy baratos y fáciles de utilizar en proyectos de *IoT*.

Para tratar con la red eléctrica necesitaremos un relé que soporte hasta 230V y entre y en nuestro caso, que funcione con 3V ya que es el máximo que pueden aportar los pines GPIO de nuestro ESP8266

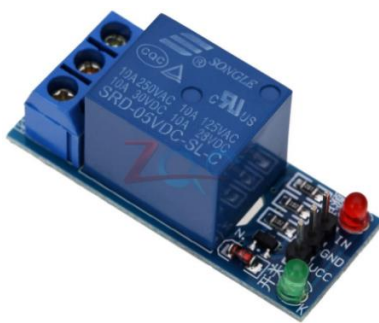


Figura 24: Relé de un canal

Estos relés, como el observado en la Figura 22: Relé de un canal, pueden ser encontrados por un precio de 0,50 – 0,90 €.

4.1.4. Fuente de alimentación 5V.

Los elementos que van a componer en su totalidad la placa de prototipo deberán ser alimentados, ya que la placa de desarrollo no va a estar siendo continuamente alimentada a través del puerto micro-USB, necesitaremos una fuente de alimentación capaz de alimentar no solo la placa de desarrollo, sino también los otros componentes del dispositivo.

Esto lo podemos conseguir gracias a un convertidor AC-DC tipo Buck que cambiará nuestra tensión de 220V AC a 5V DC con los que se podrá alimentar los dispositivos integrados en la placa. Esta fuente además nos proporciona una intensidad de 700mA, suficientes como para alimentar sin problema todos los elementos.

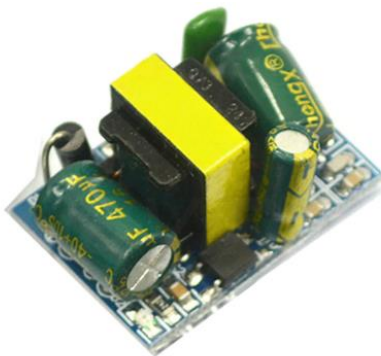


Figura 25: Módulo fuente de alimentación de 5V

Este tipo de fuente tiene un precio de unos 0,70 euros aproximadamente.

4.2. Especificaciones del IDE y Software Empleados

Para llevar a cabo el proyecto y conseguir que cumpla los objetivos establecidos, necesitaremos utilizar distintas herramientas de software. La

primera herramienta es el entorno de desarrollo *Arduino*, que será la herramienta de programación. La siguiente herramienta software será la plataforma de desarrollo de aplicaciones web *Firebase*, que consta de una base de datos a tiempo real que puede interactuar directamente con Arduino. A esto le sumamos la plataforma *Thingspeak*, plataforma web para IoT encargada de recibir los datos y poder mostrarlos al usuario. También se utilizó el entorno de desarrollo *App Inventor*, con el cual crearemos una aplicación móvil con la que poder interactuar con todas las plataformas anteriormente descritas. Y por último, el software gratuito de diseño EasyEDA con el que se llevó a cabo el diseño y la elaboración de la placa PCB.

4.2.1. Entorno de Programación de Arduino (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como una aplicación. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además, en el caso de Arduino, incorpora las herramientas necesarias para cargar el programa en la placa a través del puerto serie.

Los programas de *Arduino* están compuestos por un solo fichero con extensión “.ino”, aunque cabe la posibilidad de organizarlo en distintos ficheros para incluir, por ejemplo, librerías. El fichero principal debe encontrarse siempre en una carpeta del mismo nombre que el fichero.

Como comentamos previamente, *Arduino* es un entorno liviano que nos otorga las herramientas necesarias para subir, depurar y comunicarnos con la placa. Describiendo brevemente los menús más importantes dentro del menú *Arduino* podemos encontrar las siguientes opciones:

- **Nuevo y abrir:** Permite crear y abrir una aplicación para el dispositivo (*sketch*) respectivamente.
- **Proyecto:** permite abrir un conjunto de sketch que conforma un proyecto determinado.
- **Ejemplos:** Son sketch que nos proporcionan ejemplos sobre el uso de la placa Arduino. Hay ejemplos básicos que van desde hacer parpadear

un led hasta hacer un servidor web básico. Una herramienta muy útil para principiantes en la materia.

El menú Programa proporciona las opciones de verificar y compilar el programa, incluir librerías y mostrar la carpeta del programa. Esta opción es muy útil para buscar los sketches sin tener que navegar hasta encontrar la carpeta de los ficheros. Esta carpeta está localizada en la carpeta del usuario, dentro de una carpeta llamada ARDUINO. En esta carpeta se deberán copiar todas las librerías utilizadas.

El menú Herramientas incluye las siguientes opciones:

- **Auto formato:** Formatea el código de tal forma que sea legible.
- **Archivo de programa:** Comprime toda la carpeta del proyecto para guardarla en un fichero ZIP.
- **Reparar codificación y recargar:** Esta opción es útil para reparar ficheros con diferentes codificaciones, e.g. cuando se ha guardado un fichero con codificación UTF8 usando una codificación diferente.
- **Administrar bibliotecas:** Gestor de librerías que nos permite buscar en Internet cualquier librería que necesitemos para nuestros proyectos.
- **Monitor serie:** Es un monitor series que nos permite la comunicación con la placa. Permite tanto enviar como recibir datos a través de un puerto serie (e.g. a través del USB). Muy útil a la hora de comprobar el funcionamiento del dispositivo y ver que realiza las lecturas correctamente.
- **Puerto:** Permite seleccionar con qué placa estamos trabajando (modelo) y en qué puerto está conectada
- **Quemar bootloader:** Permite volver a cargar el bootloader (programa encargado de iniciar el chip y periféricos, así como de iniciar el programa).

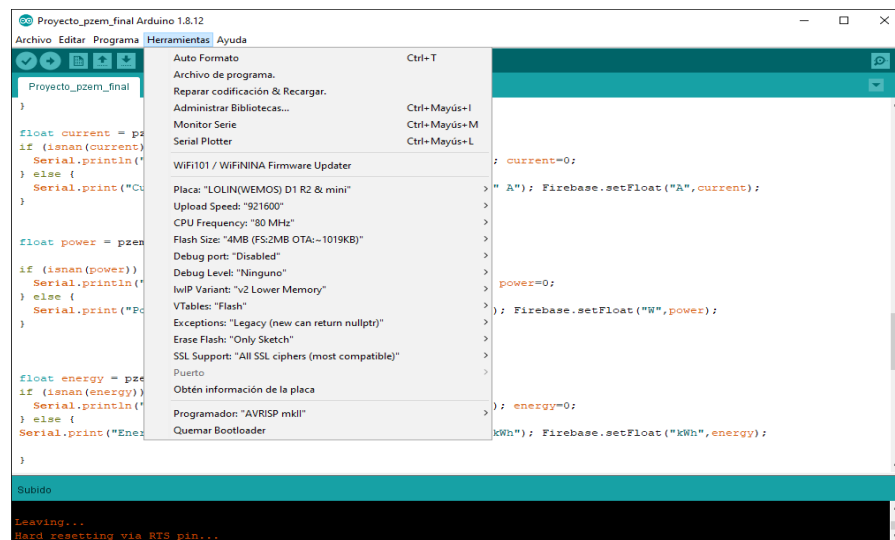


Figura 26: IDE de Arduino

Para terminar, vamos a comentar los elementos de un programa en *Arduino*. Los *sketches* en *Arduino* tienen una estructura como la de la *Figura 27: Sketch de Arduino*.

Consta de dos funciones principales:

- **Setup:** Esta función se ejecuta una solamente una vez iniciado el programa, por lo que se realizan las tareas de inicialización de variables, abrir canales de comunicación, etc. Es decir, es donde se deberá cargar la configuración y el estado inicial del proyecto.
- **Loop:** El funcionamiento de Arduino consiste en un bucle infinito. Por lo que este es el cuerpo principal del programa que se ejecutará infinitamente.

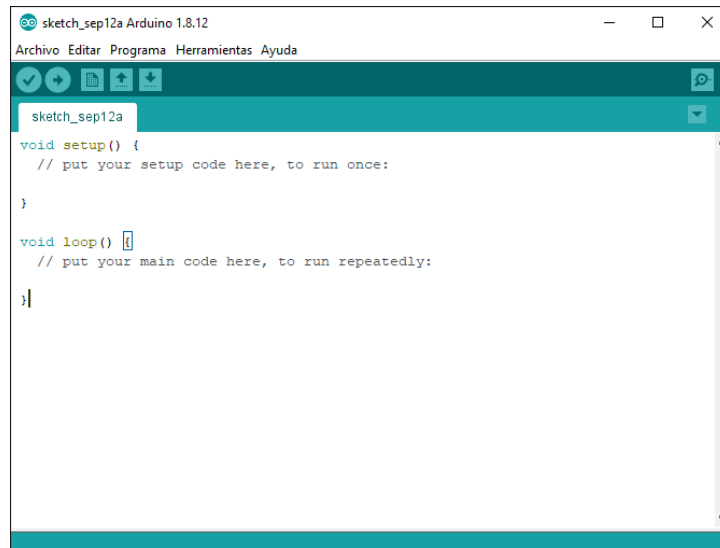


Figura 27: Sketch de Arduino

Configuración en Arduino IDE de la placa WeMos D1 Mini.

Para poder interconectar placas de desarrollo de terceros, es decir, fabricantes que no sean Arduino, tenemos tres opciones:

- **A través de la IDE de Arduino:**

Para poder comunicar el IDE con el ESP8266 vamos a dirigirnos a la pestaña de la barra de herramientas que dice “Programas”, en ella situaremos el cursor sobre “Incluir Librería” y después accederemos a “Administrar Bibliotecas...”.

Seguidamente aparecerá una ventana con el nombre “Gestor de Librerías”, aquí podemos instalar, actualizar o desinstalar librerías creadas por Arduino, proveedores o usuarios.

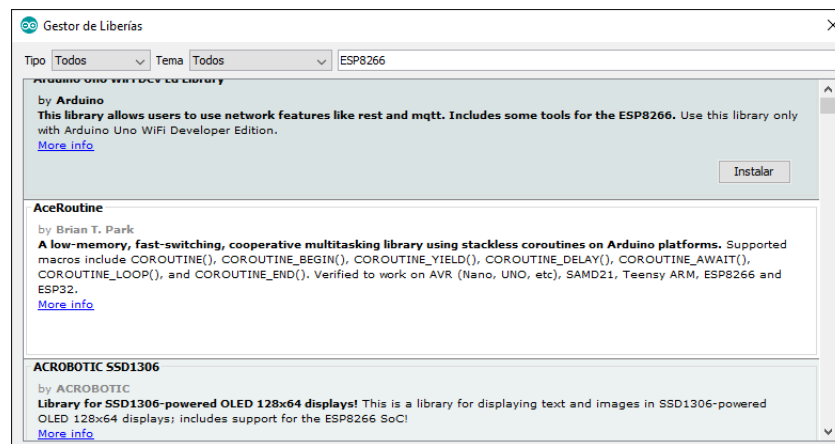


Figura 28: Gestor de Librerías Arduino IDE

En la parte superior derecha podremos ver un cuadro de entrada de texto en el que escribiremos “ESP8266” y posteriormente se procede a instalar la librería.

- **Descargando la librería e importándola:**

Si se accede al enlace <https://github.com/esp8266/Arduino> , y nos dirigimos al botón verde “Clone or download” nos descargará un archivo “.ZIP” que podremos importar en el Arduino IDE.

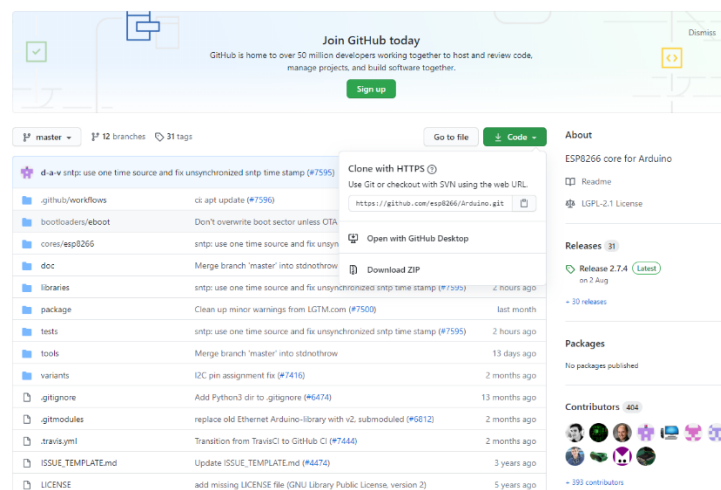


Figura 29: Sección de Github del ESP8266

En el IDE nos dirigimos a “Programa” – “Incluir Librerías” – “Añadir Biblioteca.Zip...”. Dirigiéndonos a la ruta donde tengamos el archivo, lo seleccionamos y lo importamos.

- **Incluyendo la URL adicional de tarjeta:**

De esta forma hemos incluido la librería del ESP8266.

Nos dirigimos a “Archivo” – “Preferencias”. Ya dentro, en la pestaña de “Ajustes”, se puede observar en la parte inferior una zona que señala “Gestor de URLs Adicionales de Tarjetas”, en ese cuadro de texto incluimos el siguiente enlace y pulsamos OK:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

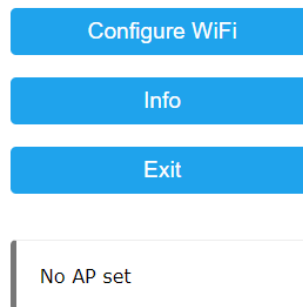
4.2.2. WiFiManager

WiFiManager es una librería de Arduino que permite gestionar las conexiones Wi-Fi del ESP8266 mediante un portal web. Cuando el microcontrolador funciona por primera vez (o no consigue conectarse a ningún punto de acceso configurado previamente), este funciona como punto de acceso. El usuario podrá conectarse usando un PC o móvil, y accederá al navegador. El dispositivo redirigirá cualquier petición HTTP a la dirección IP 192.168.4.1, donde habrá un servidor web con distintas opciones para configurar una conexión Wi-Fi.

Gracias a esta librería, nuestro dispositivo podrá funcionar en cualquier lugar siempre que haya disponible una red Wi-Fi a la que conectarse y poder enviar y recibir sus datos.

ESP8266Temp

WiFiManager



Configure WiFi

Info

Exit

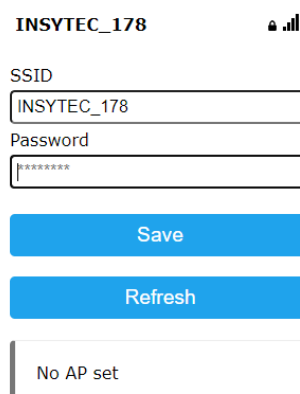
No AP set

Figura 30: Interfaz de WifiManager

En la Figura 29, se puede observar la interfaz que proporciona el portal web de WifiManager. En él se pueden observar las siguientes opciones:

- **Configure WiFi:** Nos permite escanear las redes Wi-Fi disponibles y, tras elegir una, el dispositivo se conectará a esta tras introducir la contraseña correctamente.
- **Info:** Nos proporciona información acerca del módulo Wi-Fi.
- **Reset:** Reinicia el ESP8266

Para el proyecto hemos utilizado la opción Configure WiFi y tras seleccionar nuestra red e introducir la contraseña, el microcontrolador está conectado a Internet. Podemos comprobarlo en el monitor serie de Arduino.



INSYTEC_178

SSID

INSYTEC_178

Password

Save

Refresh

No AP set

Figura 31: Interfaz Configure Wifi de WifiManager

Podemos encontrar la librería WifiManager buscando en el gestor de librerías del IDE de Arduino. Buscamos wifimanager e instalamos la librería que indica WiFiManager by tzapu.

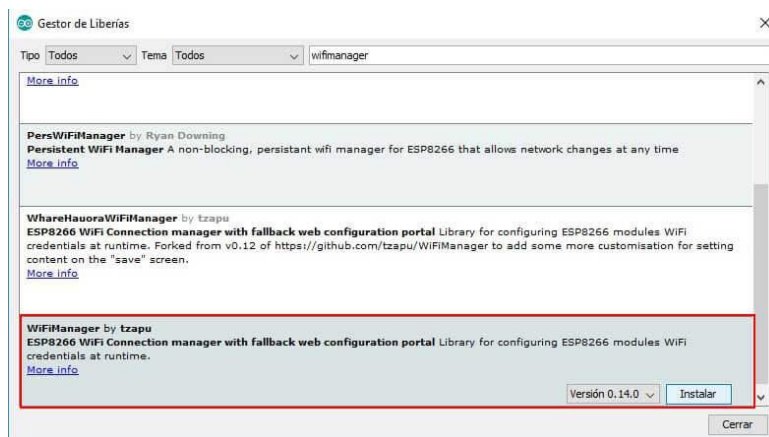


Figura 32: WifiManager en el gestor de librerías de Arduino

4.2.3. Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por Google en 2014.



Figura 33: Logotipo Firebase

Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad. Los desarrolladores tendrán una serie de ventajas al usar esta plataforma:

- Sincronizar fácilmente los datos de sus proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.
- Usa un conjunto de herramientas multiplataforma: se integra fácilmente para plataformas web como en aplicaciones móviles. Es compatible con grandes plataformas como Android, IOS, aplicaciones web, Unity, etc.

- Usa la infraestructura de Google y escala automáticamente para cualquier tipo de aplicación, desde las más pequeñas hasta las más potentes.
- Crea proyectos sin necesidad de un servidor: Las herramientas se incluyen en los SDK para los dispositivos móviles y web, por lo que no es necesario la creación de un servidor para el proyecto.

El servicio que nosotros usaremos será Realtime Database. Firebase proporciona una base de datos a tiempo real y permite que los usuarios accedan a la información de sus datos desde cualquier dispositivo en cualquier momento, compartiendo una instancia de Realtime Database, y cada vez que un usuario realice una modificación en esta, se almacena dicha información en la nube y se notifica simultáneamente al resto de dispositivos.

Una funcionalidad interesante de esta base de datos, es que, si un usuario realiza cambios y pierde a la vez su conexión a Internet, el SDK de la plataforma usa una caché local en el dispositivo donde guarda estos cambios; y una vez que vuelva a tener conexión, automáticamente se sincronizan los datos locales.

Para poder utilizar la plataforma en nuestro proyecto, hay que registrarse con un correo electrónico. Después añadimos un nuevo proyecto y acto seguido creamos una base de datos a tiempo real. Dentro establecemos una serie de variables, que gracias a Arduino y a la librería “Firebase.h”, podremos conectarlas directamente a nuestro dispositivo.

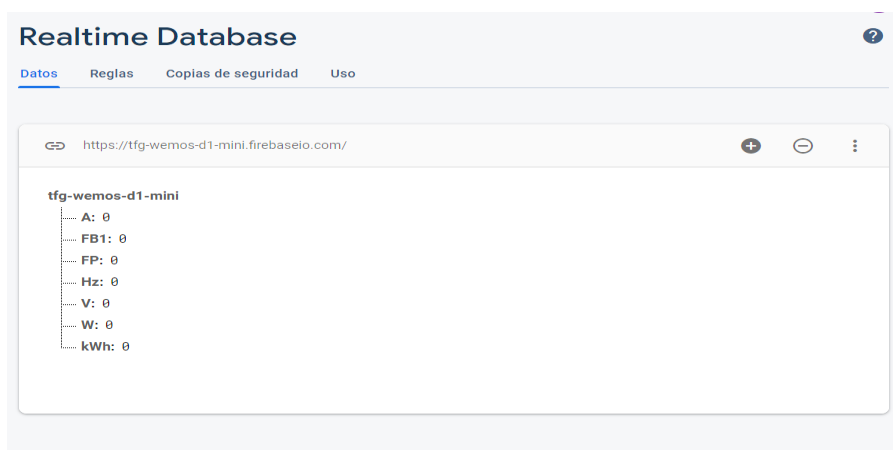


Figura 34: Variables de nuestra Realtime Database en Firebase

Una parte fundamental es la comunicación con nuestro dispositivo, para ello, cada base de datos cuenta con un footprint que es un largo código que podemos encontrar en la configuración de nuestro proyecto y al añadirlo a nuestro programa de Arduino, nos permitirá interactuar con esa base de datos en concreto.

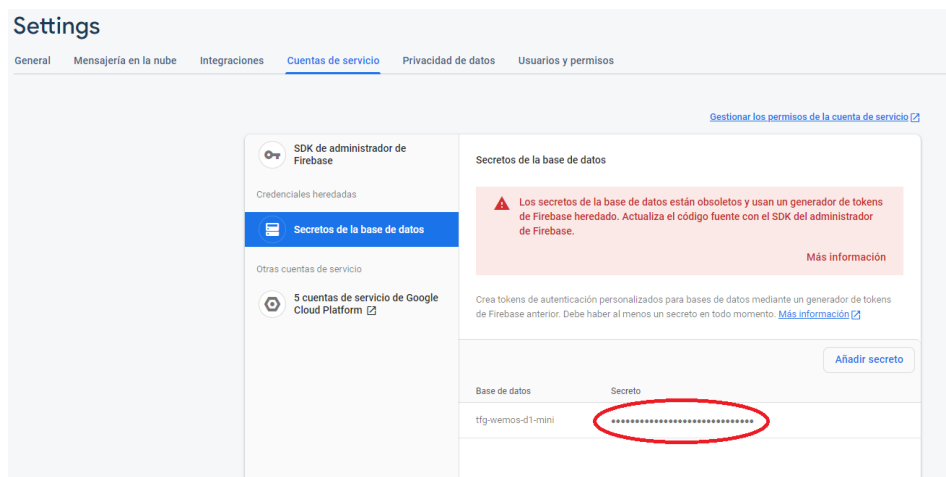


Figura 35: Localización del footprint de Firebase

Podemos encontrar la librería de Firebase para el IDE de Arduino en el siguiente enlace:

[https://github.com/FirebaseExtended/firebase-
arduino/blob/master/src/Firebase.h](https://github.com/FirebaseExtended/firebase-arduino/blob/master/src/Firebase.h)

4.2.4. ThingSpeak

Thingspeak es una aplicación de código abierto para proyectos IoT cuya función consiste en el almacenamiento y recuperación de datos a través de Internet utilizando el protocolo HTTP (*Hypertext Transfer Protocol*) o MQTT (*Message Queue Telemetry Transport*). ThingSpeak fue inicialmente lanzado por *ioBridge* como un servicio de apoyo a las aplicaciones de IoT.



Figura 36: Logotipo de ThingSpeak

Como vamos a ver en esta sección, ThingSpeak permite la creación de diferentes canales para distintos dispositivos. Un canal es el concepto que utiliza esta plataforma para enviar o recuperar datos, pudiendo almacenar valores en 8 campos distintos. Además de almacenar datos, puede representarlos gráficamente.

ThingSpeak ha integrado el soporte del software e cálculo numérico MATLAB. Esto ha permitido a los usuarios de ThingSpeak analizar y visualizar datos subidos utilizando scripts de MATLAB. Actualmente, Thingspeak pertenece a MathWorks.

ThingSpeak es una plataforma gratuita para uso personal, cuenta con una API (Application Programming Interface) abierta. Para las licencias abiertas, ThingSpeak solamente nos permite mandar tres millones de mensajes al año por usuario, cantidad suficiente para cumplir los requisitos de nuestro proyecto.

Para comenzar debemos crear una cuenta y registrarnos en la plataforma. Como se comentó anteriormente, la comunicación con la plataforma se basa en canales. Estos almacenan datos que se envían a través de Internet. Los datos almacenados constan de tres elementos:

- Ocho campos para almacenar datos de cualquier tipo.
- Tres campos para poder almacenar la latitud, altitud, longitud y elevación del dispositivo. Para esto necesitaríamos un componente hardware que nos proporcionara esta información.
- Un campo para almacenar el estado.

New Channel

Name

Description

Field 1 ☒

Field 2 ☐

Field 3 ☐

Field 4 ☐

Field 5 ☐

Field 6 ☐

Field 7 ☐

Field 8 ☐

Metadata

Tags
(Tags are comma separated)

Link to External Site

Link to GitHub

Elevation

Show Channel Location ☐

Latitude

Longitude

Show Video ☐

☒ YouTube

☐ Vimeo

Video URL

Show Status ☐

Figura 37: Creación de un nuevo canal en ThingSpeak

Como comentamos anteriormente, un requisito fundamental en este proyecto y en las comunicaciones IoT, es la seguridad de los datos en las comunicaciones. Así, se utilizará un canal privado de ThingSpeak. Este canal tendrá un ID Channel que será el identificador del canal. Asimismo, el canal tendrá dos claves, una para escribir datos y otra para poder leerlos, las cuales necesitaremos incluir en nuestro programa de *Arduino*. Así, unos usuarios o aplicaciones podrán escribir datos, y otros podrán leerlos sin posibilidad de modificarlos. ThingSpeak incorpora una opción de generación de contraseñas, si bien se puede cambiar en cualquier momento. En el proyecto se han configurado de manera privada ya que la información que contienen, a parte de

un mayor o menor consumo energético, podría indicar a otras personas si nuestro domicilio está habitado o no.

The image shows a web interface for managing API keys on ThingSpeak. It is divided into two main sections: 'Write API Key' and 'Read API Keys'. In the 'Write API Key' section, there is a text input field labeled 'Key' containing a masked string 'XXXXXXXXXXXXXXXX'. Below it is an orange button labeled 'Generate New Write API Key'. The 'Read API Keys' section contains a 'Key' input field with a masked string, a 'Note' text area, and two buttons: a green 'Save Note' button and a red 'Delete API Key' button. At the bottom of the interface is an orange button labeled 'Add New Read API Key'.

Figura 38: API Keys de ThingSpeak

Además de permitirnos crear distintos canales, ThingSpeak tiene otras funcionalidades:

- **ThingTweet:** Actúa como proxy permitiendo a los dispositivos enviar actualizaciones de status a Twitter.
- **TweetControl:** Escucha los hashtags de Twitter y luego permite ejecutar servicios web usando ThingHTTP.
- **ThingHTTP:** Permite a un microcontrolador o dispositivo a bajo nivel conectarse a un servicio web a través de Internet o la propia red.
- **React:** Ejecuta una acción cuando se cumple una cierta condición para los datos en el canal.
- **TalkBack:** Permite que cualquier dispositivo pueda actuar sobre los comandos en cola.
- **TimeControl:** Con TimeControl se puede ejecutar ThingHTTP o ThingTweet a una determinada hora, e incluso añadir un nuevo comando TalkBack.

La librería de ThingSpeak para el entorno IDE de Arduino se puede encontrar en el siguiente enlace:

<https://github.com/mathworks/thingspeak-arduino>

4.2.5. App Inventor

App Inventor es un entorno de desarrollo software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente desde la web. Las aplicaciones creadas con App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil, las cuales son suficiente para cumplir los objetivos de nuestro proyecto.



Figura 39: Logotipo de App Inventor

Al registrarnos en la web <http://ai2.appinventor.mit.edu> y al hacer click en Projects—Start New Project, se abrirá una ventana en la que se podrá nombrar el proyecto sobre el cual se va a empezar a trabajar.

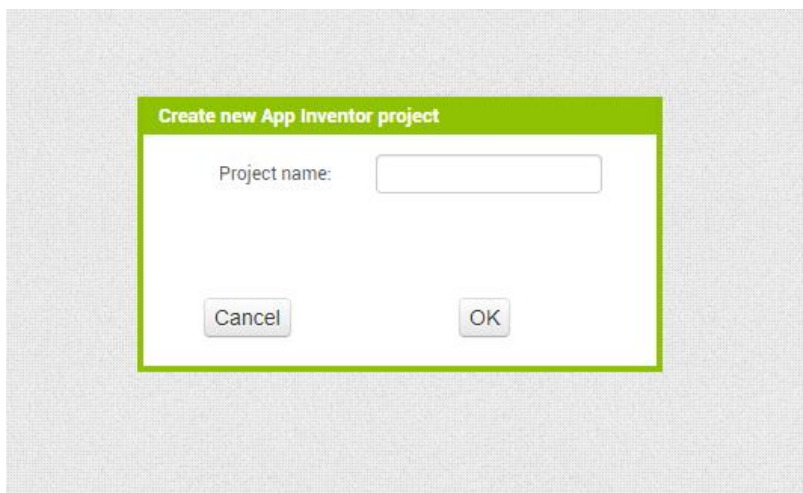


Figura 40: Nuevo proyecto de App Inventor

El próximo paso es empezar a diseñar nuestra aplicación, se opta por un diseño simple e intuitivo para dar facilidades a los usuarios. El diseño consta de dos fases:

- **Designer:** La ventana Designer nos permite llevar a cabo el diseño de la interfaz que tenemos en mente.
- **Blocks:** La opción de blocks no llevará a la parte destinada a la programación de los eventos que queremos que ocurran al interactuar con nuestra app.

La ventana Designer está dividida en cuatro grandes apartados:

- **Palette:** La encontramos a la izquierda de la pantalla y recoge todos los componentes con los que podemos trabajar a la hora de crear nuestras aplicaciones (botones, sprites, imágenes, sonidos, etc...)
- **Viewer:** Simula una pantalla de un dispositivo móvil, aquí iremos añadiendo los diferentes componentes y dando forma al aspecto que tendrá la app que habíamos imaginado.
- **Components:** Este apartado recoge en una lista aquellos componentes que finalmente formarán parte de la aplicación, será a partir de esta lista y el último apartado, properties, como terminaremos nuestro diseño.
- **Properties:** Último gran apartado de esta ventana de diseño. Seleccionando cada uno de los componentes podremos modificar sus

propiedades (modificar textos, cambiar colores, alineaciones, añadir imágenes, etc.).

Nuestra aplicación estará directamente conectada con nuestra cuenta de Firebase, para poder permitir la conexión entre ambas, para ello tenemos que utilizar el token de Firebase del cual hablamos previamente.

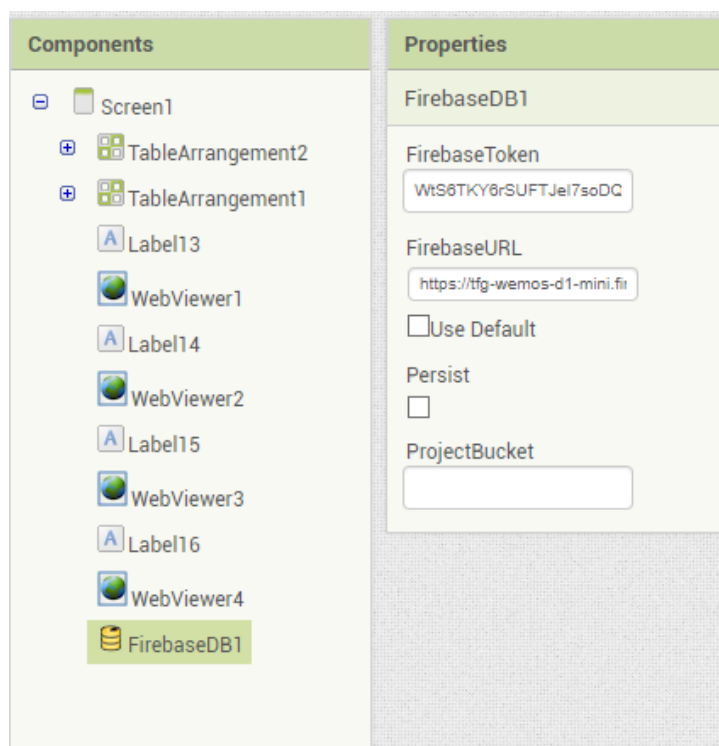


Figura 41: Configuración del componente Firebase en App Inventor

Una vez completado el diseño de nuestra aplicación, pasaremos a la ventana Blocks, desde la cual realizaremos la programación que tendrán los componentes seleccionados, es decir, con esta opción conseguiremos que nuestra app se comporte como deseamos mediante la unión de bloques que permitirán que se ejecuten los diferentes eventos, en nuestro caso, permitiremos ver en nuestra aplicación las medidas a tiempo real de la base de datos de Firebase, así como controlar desde esta el encendido y el apagado de nuestro dispositivo mandando una señal al relé y decidir si resetear el medidor de energía del PZEM-0004T.

Para agilizar el proceso y poder probar nuestra aplicación, se ha descargado en el móvil la aplicación de Android MIT App Inventor, la cual nos va a permitir instalar rápidamente en nuestro móvil la aplicación que hemos creado, ya sea mediante la lectura de un código QR o introduciendo un código de seis dígitos.

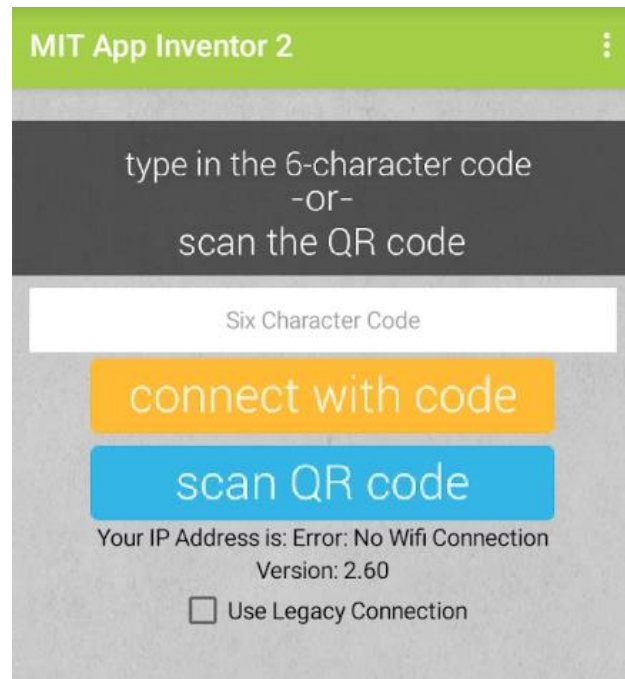


Figura 42: Menú de la app MIT App Inventor

El diseño nuestra aplicación es el siguiente:

Una pantalla principal en la que podemos ver los distintos valores de voltaje, corriente, potencia, frecuencia, energía consumida y factor de potencia, así como consultar las gráficas de Thingspeak para cada variable.

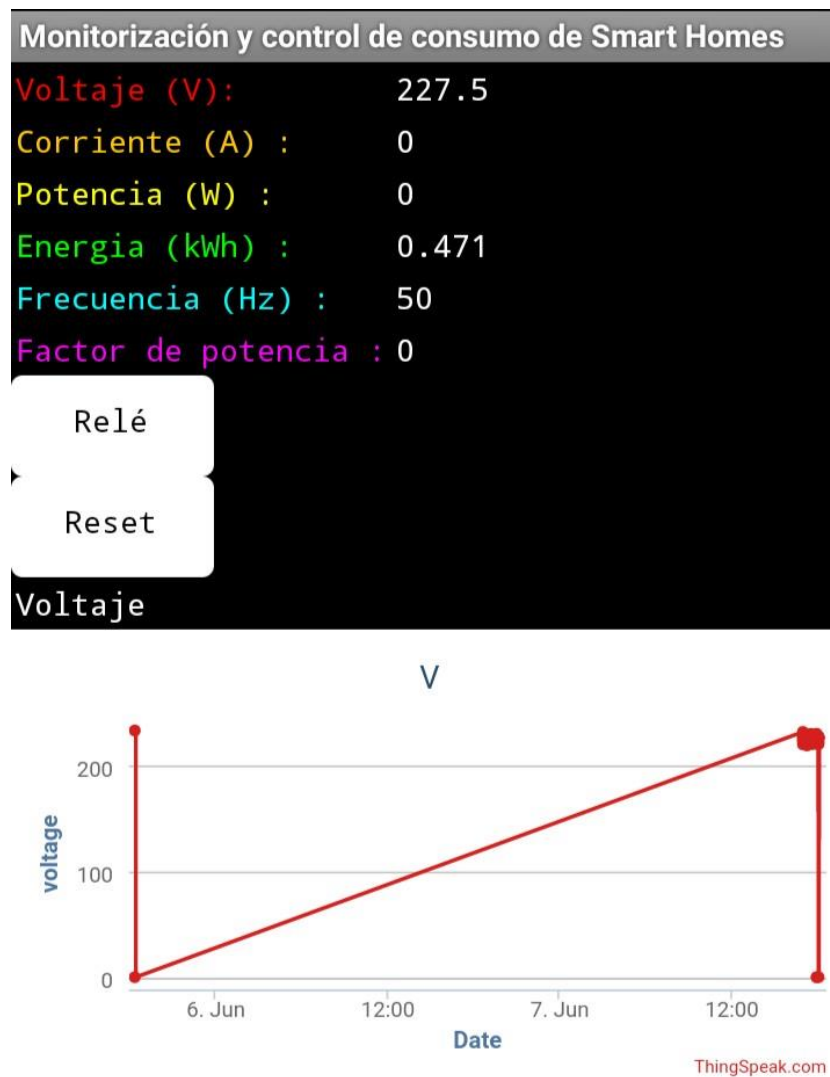


Figura 43: Pantalla principal de nuestra aplicación

Otra pantalla en la que podemos controlar el encendido y apagado del relé, así como su estado actual.

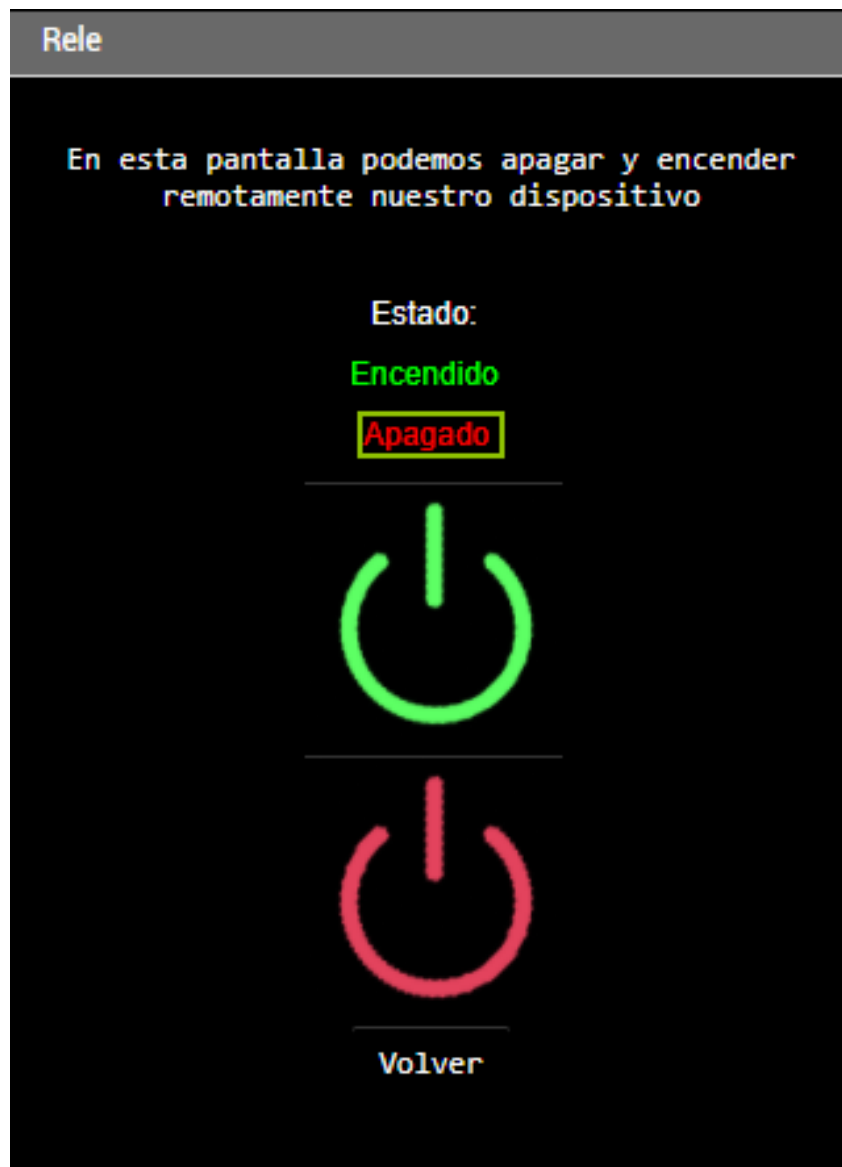


Figura 44: Pantalla de control del relé

Y también cuenta con otra pantalla en la que podemos poner a cero el contador de energía consumida, el cual se encuentra en la pantalla principal.



Figura 45: Pantalla de control de reseteo del contador

Y así quedaría nuestra aplicación, permitiéndonos leer los datos actuales, consultar las gráficas de las mediciones e interactuar con el dispositivo desde cualquier lugar.

4.3. Desarrollo de la placa de circuito impreso (PCB)

Una vez determinados los elementos presentados en el apartado 4.1, es necesario unir todos los componentes junto con la placa de desarrollo. Con la idea de crear un prototipo físico, la mejor solución para interconectar todo el hardware es diseñar una placa de circuito impreso que integre todos los componentes. Junto a esto, veremos cómo se lleva a cabo la fabricación de nuestra placa. También llevaremos a cabo una soldadura del tipo Through-Hole.

4.3.1. Diseño de la PCB

Para este diseño hemos usado el software de simulación y diseño gratuito EasyEDA, el cual puede descargarse o utilizar directamente en su plataforma web.

A continuación, vamos a mostrar el esquemático simple (es decir, se mostrarán los pines de cada elemento y su interconexión) de cada uno de los elementos que van a formar parte de la placa, así como un esquemático final, adjuntado al final de la memoria con mejor resolución.

En primer lugar, nuestra placa de desarrollo, el Wemos D1 Mini, que será el centro de operaciones de la placa, todo está conectado a él.

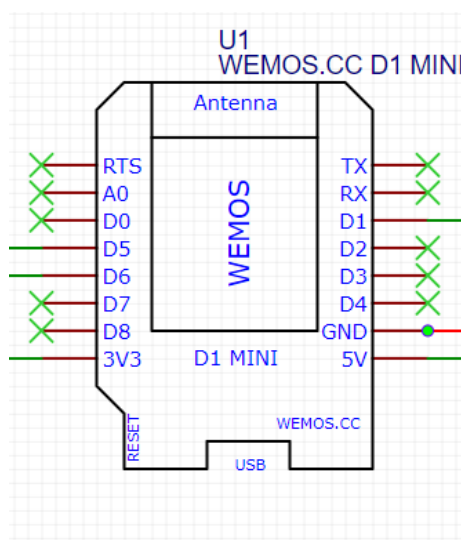


Figura 46: Esquemático simple WeMos D1 Mini

La fuente de alimentación de 5V será la que alimentará el WeMos D1 Mini para otorgar autonomía al sistema y que no dependa de baterías externas, conectándose en los pines de 5V y GND.

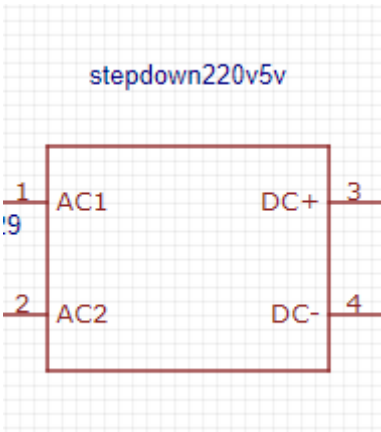


Figura 47: Esquemático simple de los bornes y la fuente de alimentación

El relé, conectado al pin D1 del Wemos D1 Mini y a su alimentación de 3,3V, así como a su GND.

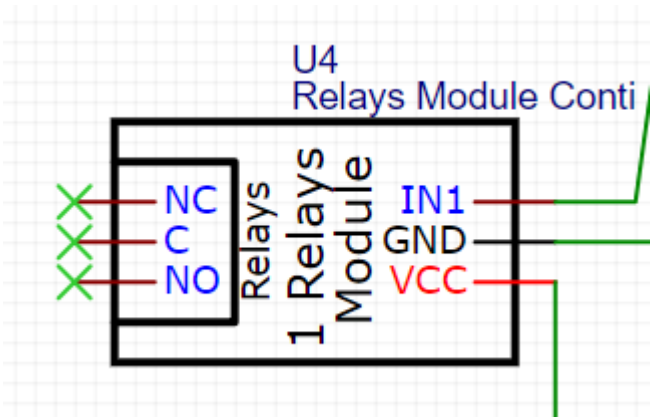


Figura 48: Esquemático simple del relé de un canal

El medidor PZEM-004T también estará alimentado por 5V y GND, sus pines TX y RX estarán conectados respectivamente a los pines D5 y D6 del Wemos D1 Mini.

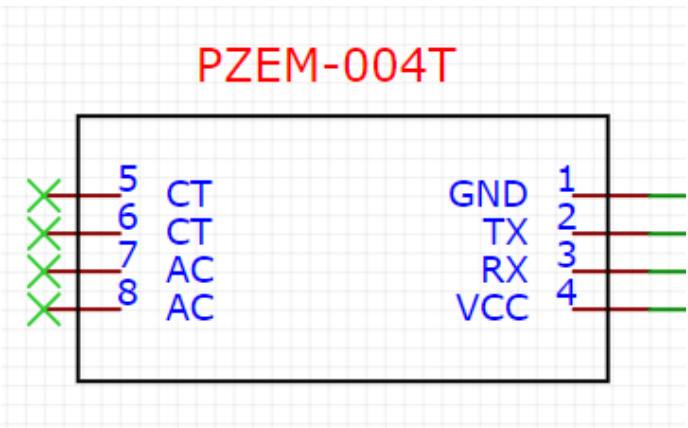


Figura 49: Esquemático simple del medidor PZEM-004T

La alimentación del circuito son unos bornes que estarán directamente conectado a la toma de corriente y que la transportan hasta la fuente reguladora AC-DC.

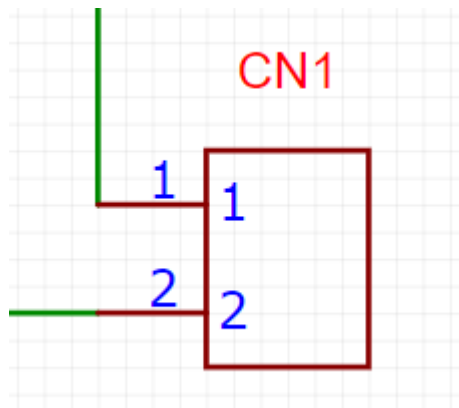


Figura 50: Esquemático simple de los bornes de alimentación

Finalmente, todo el conexionado dentro de la placa quedará así:

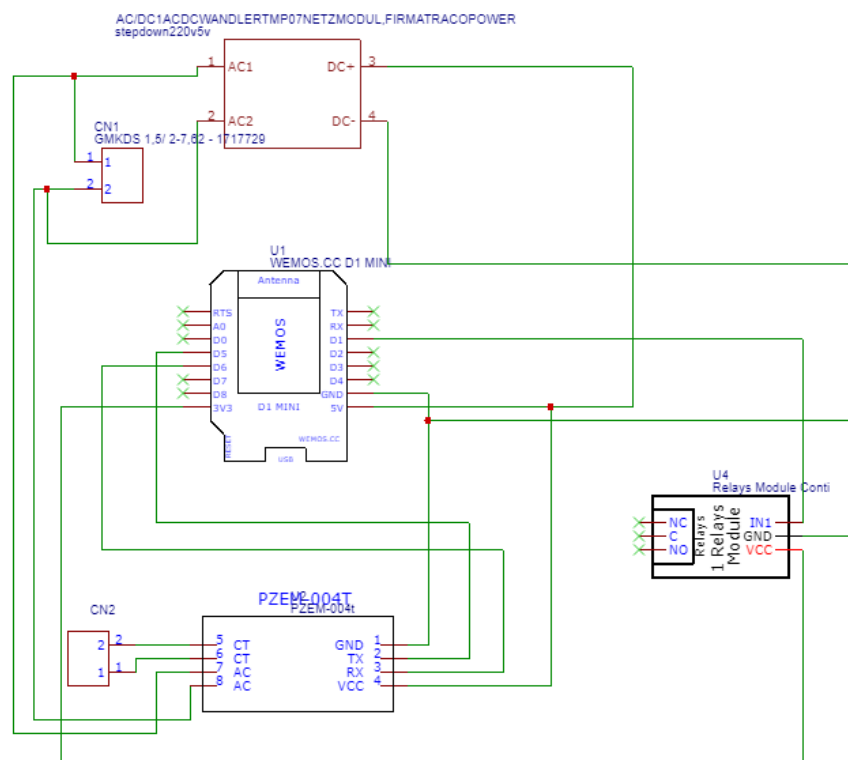


Figura 51: Esquemático total de conexionado de la placa

La Figura 43 representa el esquemático de la parte electrónica de la placa, en cuanto a la parte de tensión de red realizaremos el siguiente conexionado:

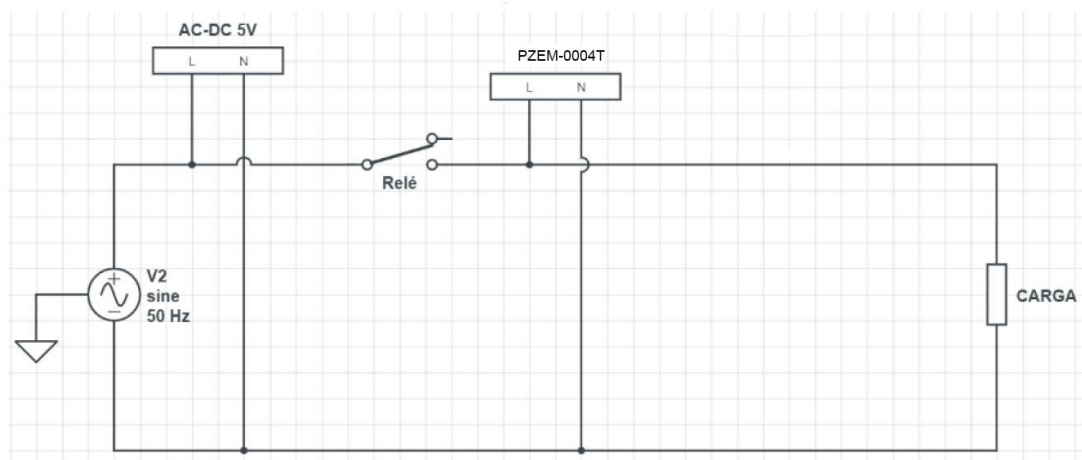


Figura 52: Esquemático del conexionado a tensión de red

Para conectar físicamente el módulo a la red utilizaremos unos conectores como los de la figura 45, el WeMos D1 Mini siempre estará alimentado a 5V, pero el PZEM-004T no realizará lecturas mientras el relé no esté excitado.



Figura 53: Conectores de dos entradas para nuestra placa de circuito impreso

Tras el diseño del esquemático, vamos a realizar el diseño de la placa de circuito impreso.

Para ello, tendremos que asignar footprints a cada elemento. La plataforma dispone de un buscador en el que cada usuario puede subir sus diseños para que cualquier otra persona pueda utilizarlos. En este caso hemos encontrado las footprint del WeMos D1 Mini; del PZEM004-T y de los bornes.

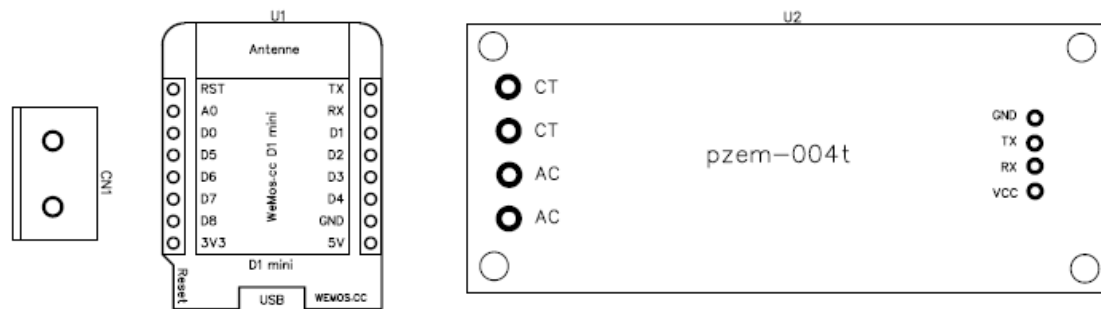


Figura 54: Footprints de los conectores WeMos D1 Mini y PZEM-004T respectivamente

En cambio, otros diseños eran similares en tamaño a los elementos que tenemos, pero con varios cambios en los pines, así que se decidió utilizar otra herramienta del software que nos permite crear nuestros propios footprints, y realizando medidas reales a nuestros elementos, se consiguieron crear los footprint del relé y de la fuente de alimentación de 5V.

AC/DC 5V



Figura 55: Diseño del footprint de la fuente de alimentación de 5V con una dimensión de 30x20mm

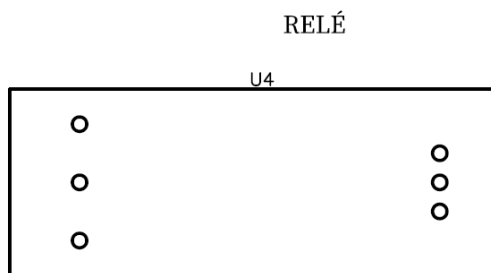
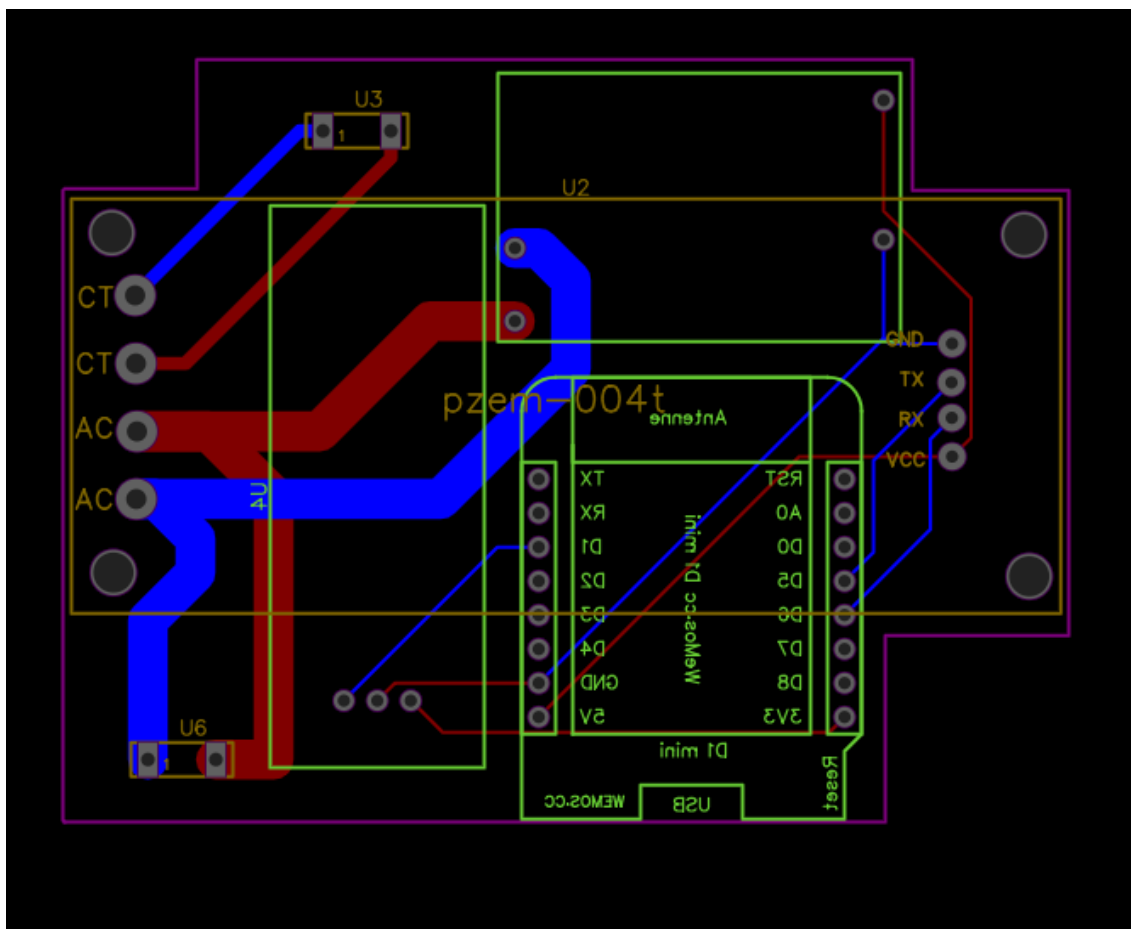


Figura 56: Diseño del footprint del relé con una dimensión de 42x16mm

Una vez tenemos asignados todos los footprint de cada componente, volvemos al esquemático del conexionado y en la barra de arriba pulsamos Design – Convert to PCB.

Ahí aparecerán todos nuestros elementos y podremos modificar también el tamaño de la placa para que se ajuste al deseado. Como para nuestro proyecto se desea que el prototipo sea de un tamaño reducido, se ha decidido soldar los elementos por ambas caras de la placa, ya que se nos permite realizar conexiones por los dos lados y ahorrar mucho espacio.

Una vez hemos colocado los elementos, podemos pulsar la opción de Auto Router y automáticamente nos conectará los elementos. Se ha seleccionado esta opción, pero hemos realizado algunos cambios en las uniones para que resulte más sencillo y se ajuste más a lo deseado. También se ha ajustado el tamaño de los trases según la corriente que vaya a pasar por ellos. La placa tiene un tamaño de 76x82 mm



En la Figura 48, los elementos bordeados de color amarillo se encuentran en la cara superior y los de color verde en la cara inferior de la placa.

Todos los pines representados serán físicamente pines macho, que, como podemos observar, están compuestos de taladro en la placa que serán del tipo THT, lo que nos facilitará el posterior soldado de las hileras de pines en ambas caras de la placa.

4.3.2. Fabricación de la PCB

Para fabricar la PCB se ha recurrido a uno de los tantos fabricantes que podemos encontrar tras realizar una pequeña búsqueda web, tras sondear varios servicios de fabricantes se ha optado por el fabricante chino JiaLiChuang, más conocido como JLCPCB, fabricantes expertos en el sector con más de diez años de experiencia, destacando por su calidad-precio y plazos de entrega excepcionales.

PCB SMT-Stencil

Add your gerber file

Only accept zip or rar,Max:10 M

Instructions for ordering

Upload History >>

Layers 1 2 4 6

Dimensions 100 100 mm

PCB Qty 5

Different Design 1 2 3 4

Delivery Format Single PCB Panel by Customer Panel by JLCPCB

PCB Thickness 0.4 0.6 0.8 1.0 1.2 1.6 2.0

PCB Color Green Red Yellow Blue White Black

Surface Finish HASL(with lead) LeadFree HASL-RoHS ENIG-RoHS

Charge Details

Special Offer: €1.68

Build Time:

PCB: 1-2 days €0.00

Calculated Price: €1.68

Weight: 170g

SAVE TO CART

Shipping Estimate:

€5.72 Via Global Direct Line

Delivery Time: 15-20 business days

Up to 50% off on electronic parts at LCSC.COM

Figura 58: Página web de JLCPCB

El proceso con el cual llevaremos a cabo el pedido de fabricación de la placa es bastante simple.

En primer lugar, antes de realizar el pedido, debemos cambiar el país de envío y la divisa con la que pagaremos para hacernos con la placa, situado en la esquina superior derecha, con el fin de ahorrarnos problemas logísticos o monetarios.

Para crear una cuenta debemos ir al botón Register, introducimos nuestras credenciales y procedemos a realizar el pedido. Si nos fijamos podemos ver que el fabricante elabora tanto placas de circuito impreso (PCB), como chapas guía para aplicación de pasta de soldadora para componentes SMD (SMT-Stencil), en las cuales no se entrará en detalle.

Si nos fijamos en la pestaña PCB podemos ver que la mayoría de parámetros de la placa pueden ser modificados a nuestro gusto, tales como la dimensión, color, número de capas, etc. Todas estas opciones incluyen variaciones en el precio.

Podemos ver que la página nos pide un archivo comprimido llamado GERBER. Para generar este archivo, simplemente tenemos que volver a nuestro software EasyEDA, guardar el documento, y en la pestaña File no dará la opción de crear el archivo GERBER. Antes de generarlo, el programa nos realizará el

DRC (Design Rule Checking) para comprobar que no hay errores en el diseño. Si todo está correcto se procede a crear el archivo.

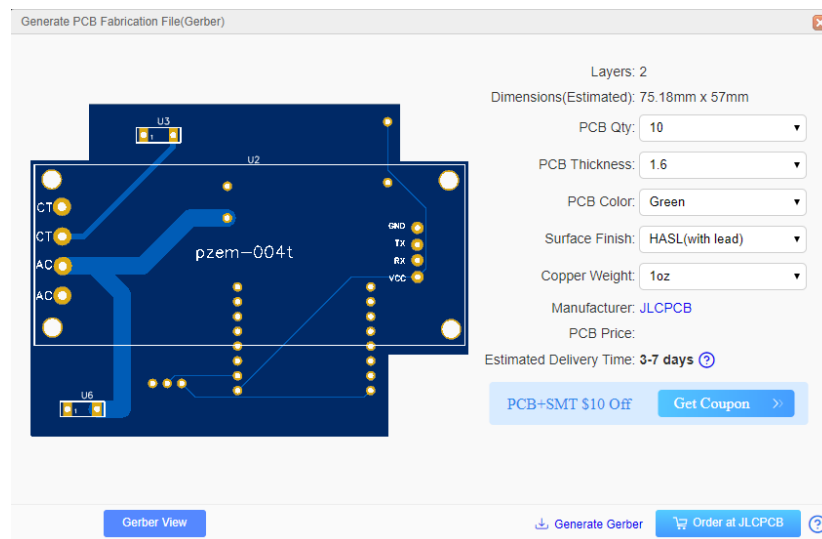


Figura 59: Creación del archivo GERBER en EasyEDA

Como podemos comprobar en la Figura 50, el programa nos da la opción de descargar el archivo .ZIP o hacer directamente el pedido a JLCPCB.

Se ha optado por descargar el archivo ZIP y subirlo nosotros mismos a la página pulsando el botón “Add your Gerber file” para comprobar precios y ver que se ajusta a lo que necesitamos.

Automáticamente con el contenido del archivo comprimido la página web procesa los datos y brinda una pre visualización (Figura 51) con el resultado final de la placa.

Detected 2 layer board of 57x75mm(2.24x2.96 inches) .

Your upload has finished processing. Enter the project details below and we'll move on to checking all the individual layers to make sure that they're correct.

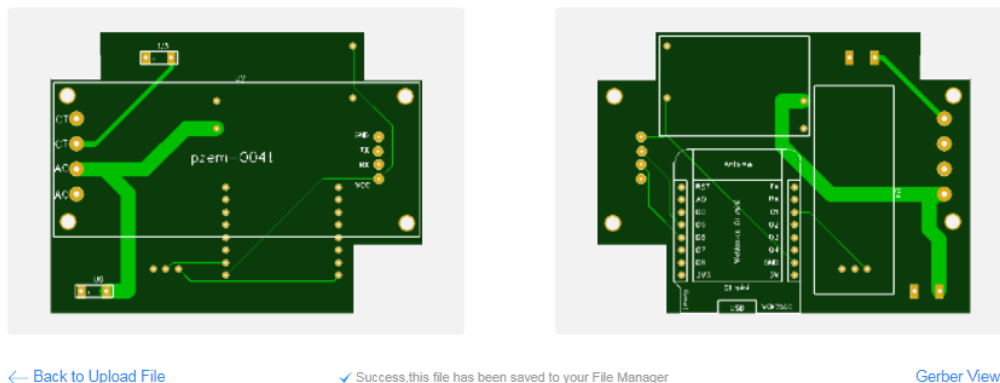


Figura 60: Previsualización del acabado final de la PCB

Si no encontramos conformes con el resultado solo nos quedará pulsar “Save to Cart” para finalizar el pedido introduciendo nuestra dirección y modo de envío. Podemos ver que el precio es de 1.70 euros por cinco placas (el mínimo que ofrece la página) y que los gastos de envío son más elevados. En este caso entre las distintas opciones se ha elegido la opción “Global Direct Lane” que por un precio de 5 euros realizan la entrega en un plazo de 10-15 días.

Al finalizar el pedido, los técnicos comprobarán el diseño de la placa y valorarán la viabilidad de su fabricación, al ser una placa pequeña y sencilla no ha habido ningún problema.

En nuestro caso a lo largo del proyecto se han llegado a desarrollar dos placas distintas. La primera que se diseñó es la siguiente:

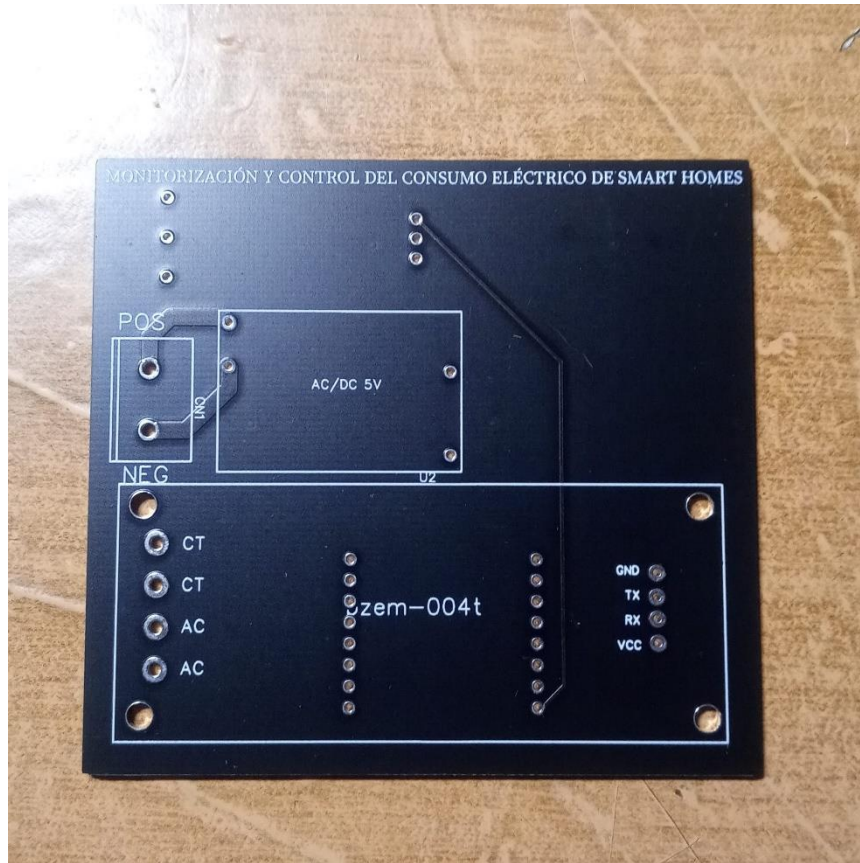


Figura 61: Primera placa Top Layer

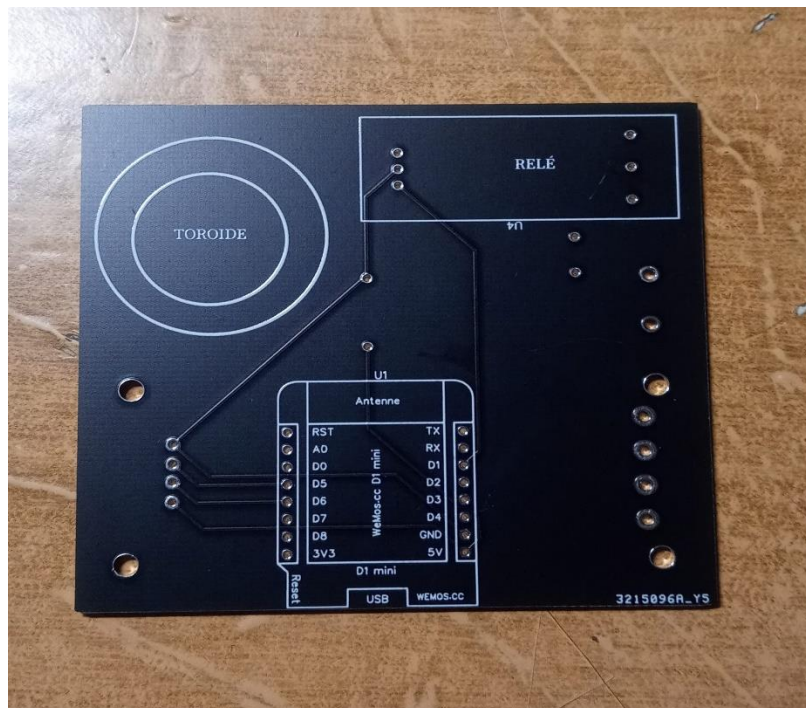


Figura 62: Primera placa Bottom Layer

La segunda PCB y la que hemos utilizado finalmente para la realización de nuestro prototipo es la siguiente:

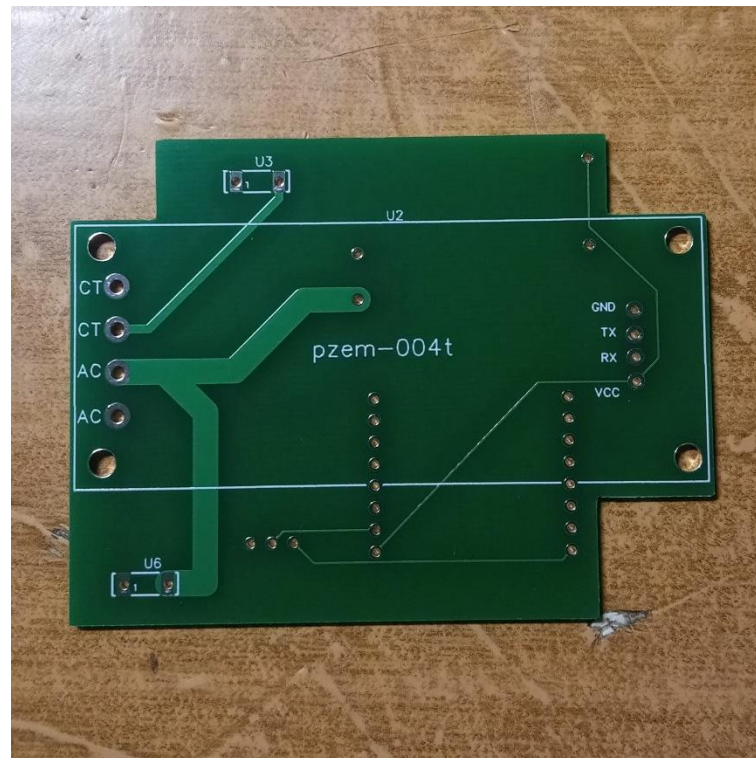


Figura 63: Segunda placa Top Layer

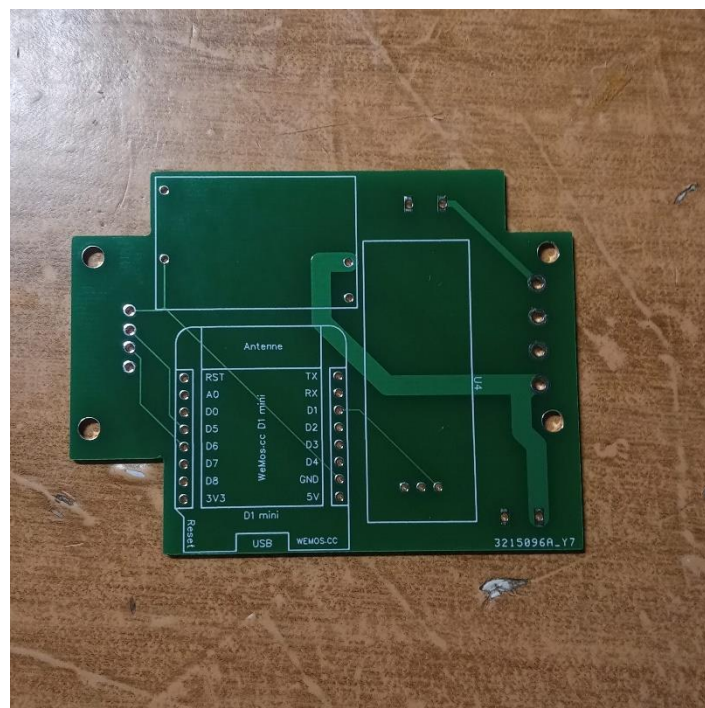


Figura 64: Segunda placa Top Layer

El motivo del rediseño se debe a que en la nueva placa hemos añadido pistas nuevas a la entrada de corriente alterna del PZEM, evitando utilizar más cables a la hora del montar el dispositivo y así poder realizarlo de forma más

sencilla. También se ha reducido considerablemente el tamaño y se ha aprovechado más el espacio disponible.

4.3.3. Montaje del prototipo

Tras recibir las nuevas placas, se procede a realizar la soldadura con estaño de todos los elementos que forman el dispositivo. El Arduino ha sido colocado de forma que puede ser extraído con facilidad si es necesario sustituirlo con otro. La entrada micro USB ha sido orientada hacia el exterior en caso de ser necesario actualizar o cambiar el código.



Figura 65: Soldadura del prototipo

El resultado sería el siguiente:

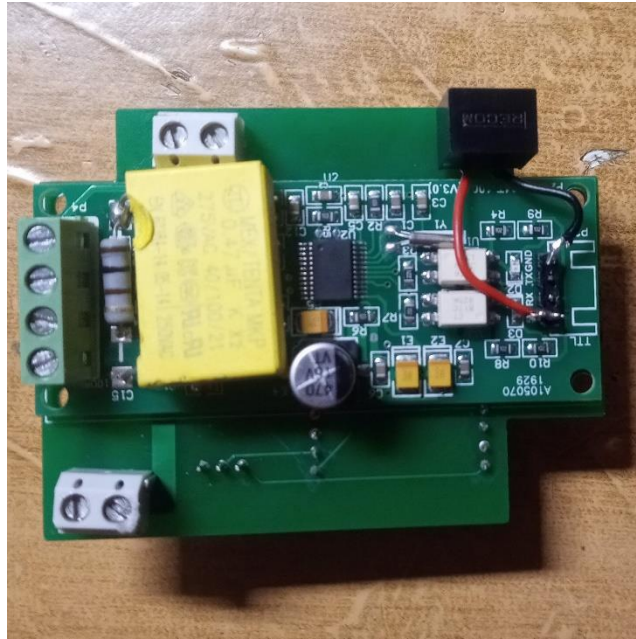


Figura 66: Prototipo Top Layer

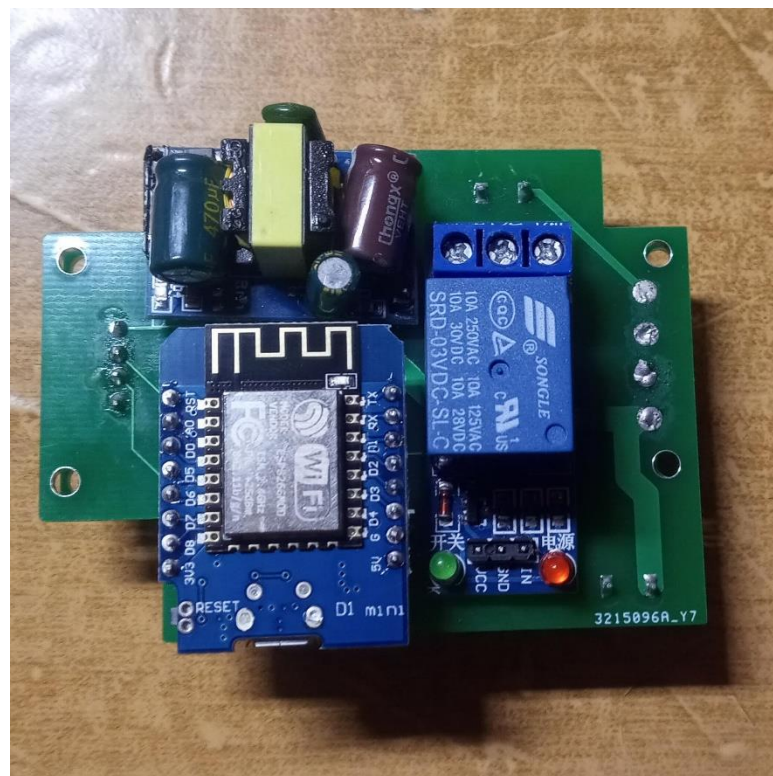


Figura 67: Prototipo Bottom Layer

5. Pruebas Realizadas

A continuación, se van a exponer los casos a estudiar para probar nuestro dispositivo, para ello utilizaremos un multímetro en modo amperímetro y compararemos con los datos obtenidos en nuestra aplicación móvil.

5.1. Estudio con carga resistiva

En las cargas resistivas, el factor de potencia $\cos(\varphi)$ es igual a uno, el seno del voltaje y el seno de la corriente están sincronizados. Esto quiere decir que la corriente es cero cuando el voltaje es cero, y que la corriente es máxima cuando el voltaje es el máximo.

Para este estudio se ha elegido un radiador eléctrico con tres niveles distintos de potencia. Hemos elegido este electrodoméstico porque va desde unos 3,8 Amperios hasta unos 10 Amperios. Realizaremos las medidas para cada uno de los tres casos y calcularemos el error relativo y absoluto en cada caso.



Figura 68: Dispositivo y multímetro conectados a un radiador eléctrico

• Nivel 1. Bajo consumo

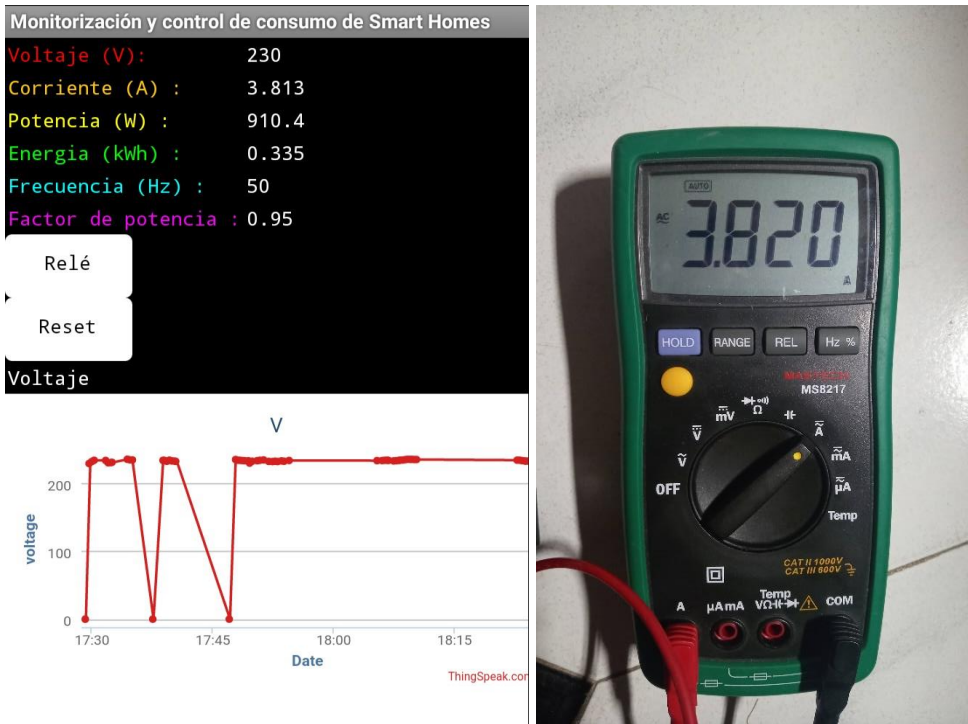


Figura 69: Lecturas del dispositivo y multímetro con el radiador en nivel 1

Medidas realizadas con el multímetro (A)	Medidas realizadas con el dispositivo (A)
3,816	3,841
3,818	3,845
3,789	3,813
3,784	3,82
3,794	3,827
3,777	3,811
3,798	3,815

Tabla 4. Carga resistiva bajo consumo

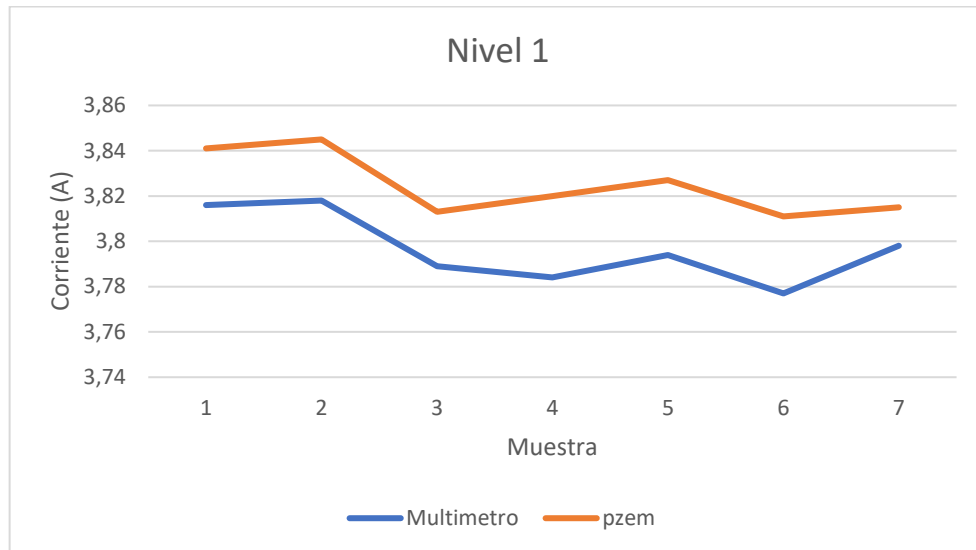


Figura 70: Medidas para el nivel 1 del radiador

Cálculo de errores

$$e_{\text{absoluto}} = |\text{Dispositivo} - \text{Multimetro}| = |26,772 - 26,576| = 0,196$$

$$e_{\text{relativo}} = \frac{e_{\text{absoluto}}}{\text{Multimetro}} * 100 = 0,7375\%$$

- Nivel 2. Consumo medio

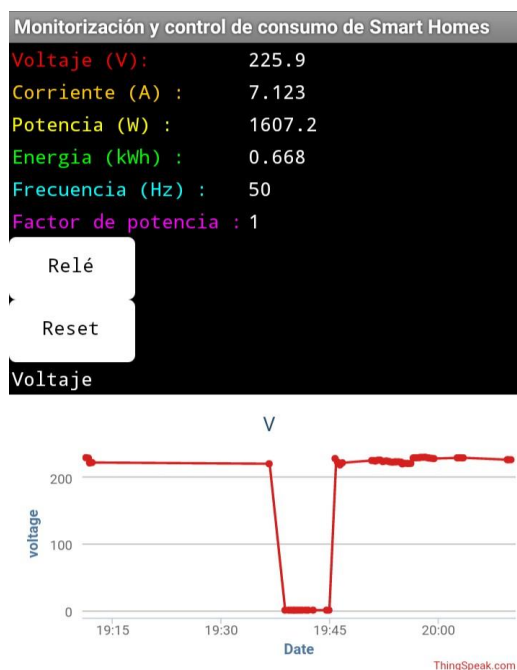


Figura 71: Lectura del dispositivo con el radiador en nivel 2

Medidas realizadas con el multímetro (A)	Medidas realizadas con el dispositivo (A)
7,1	7,141
7,065	7,104
6,98	7,079
6,97	7,054
6,96	7,041
6,96	7,026
6,95	7,013
7,1	7,141

Tabla 5. Carga resistiva consumo medio

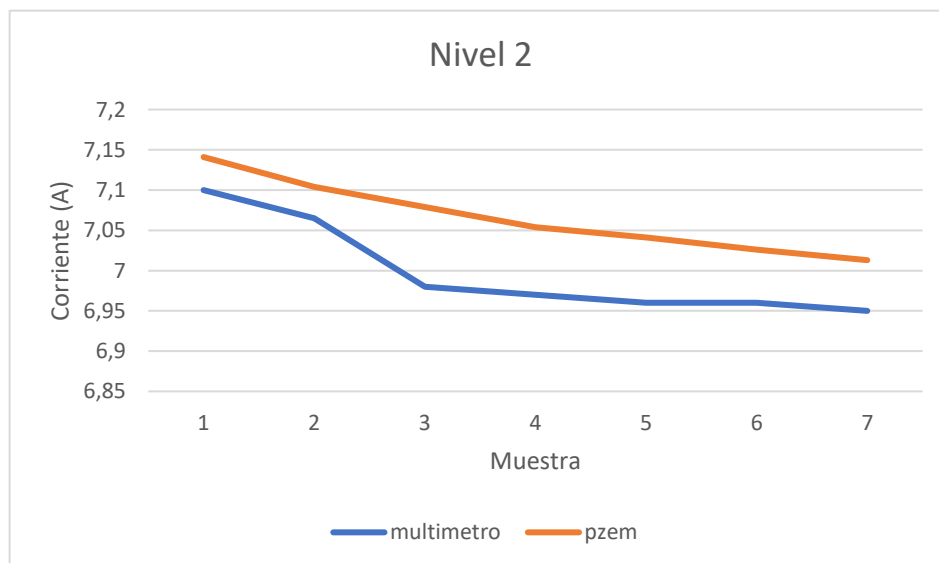


Figura 72: Medidas para el nivel 2 del radiador

Cálculo de errores

$$e_{\text{absoluto}} = |\text{Dispositivo} - \text{Multímetro}| = |49,458 - 49,985| = 0,527$$

$$e_{\text{relativo}} = \frac{e_{\text{absoluto}}}{\text{Multímetro}} * 100 = 1,0543\%$$

- Nivel 3. Alto consumo

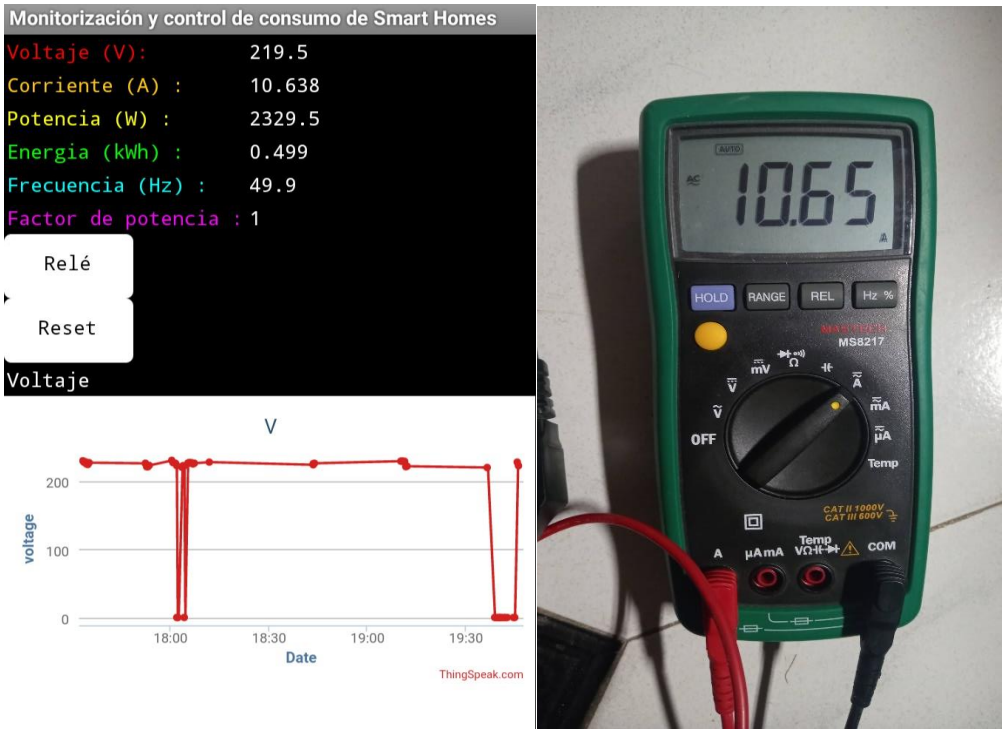


Figura 73: Lectura del dispositivo con el radiador en nivel 3

Medidas realizadas con el multímetro (A)	Medidas realizadas con el dispositivo (A)
10,54	10,611
10,59	10,603
10,52	10,593
10,52	10,591
10,51	10,581
10,38	10,577
10,5	10,585
10,54	10,611

Tabla 6. Carga resistiva alto consumo

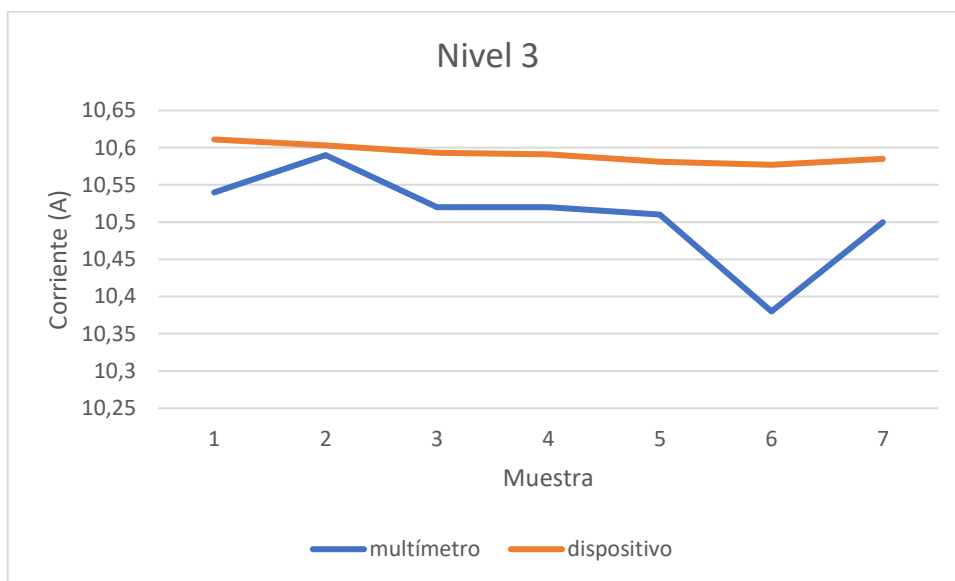


Figura 74: Medidas para el nivel 3 del radiador

Cálculo de errores

$$e_{\text{absoluto}} = |\text{Dispositivo} - \text{Multímetro}| = |49,458 - 49,985| = 0,527$$

$$e_{\text{relativo}} = \frac{e_{\text{absoluto}}}{\text{Multímetro}} * 100 = 1,0543\%$$

Observando las gráficas de las pruebas realizadas a varios niveles en un mismo electrodoméstico podemos comprobar que las medidas obtenidas por el prototipo se asemejan bastante a los resultados mostrados por el multímetro (dentro de un cierto límite de resolución) especialmente a bajos niveles de intensidad. Sin embargo, cuando el consumo de potencia de la carga resistiva es más elevado se puede apreciar que la discrepancia entre ambos valores es ligeramente más elevada.

También podemos observar que el multímetro, al ser más sensible y exacto, muestra unas fluctuaciones algo mayores en la medida mientras que el dispositivo nos muestra una medida más constante.

Como podemos ver el consumo de estos calefactores es muy elevado y resulta sin duda uno de los principales gastos de nuestras facturas durante los meses de invierno, por lo que se recomienda reducir su uso una vez esté caldeada la habitación.

5.2. Estudio con equipos electrónicos

En los equipos electrónicos producen una gran cantidad de armónicos que pueden producir medidas erróneas.

Para este estudio hemos utilizado el cargador de un ordenador portátil.

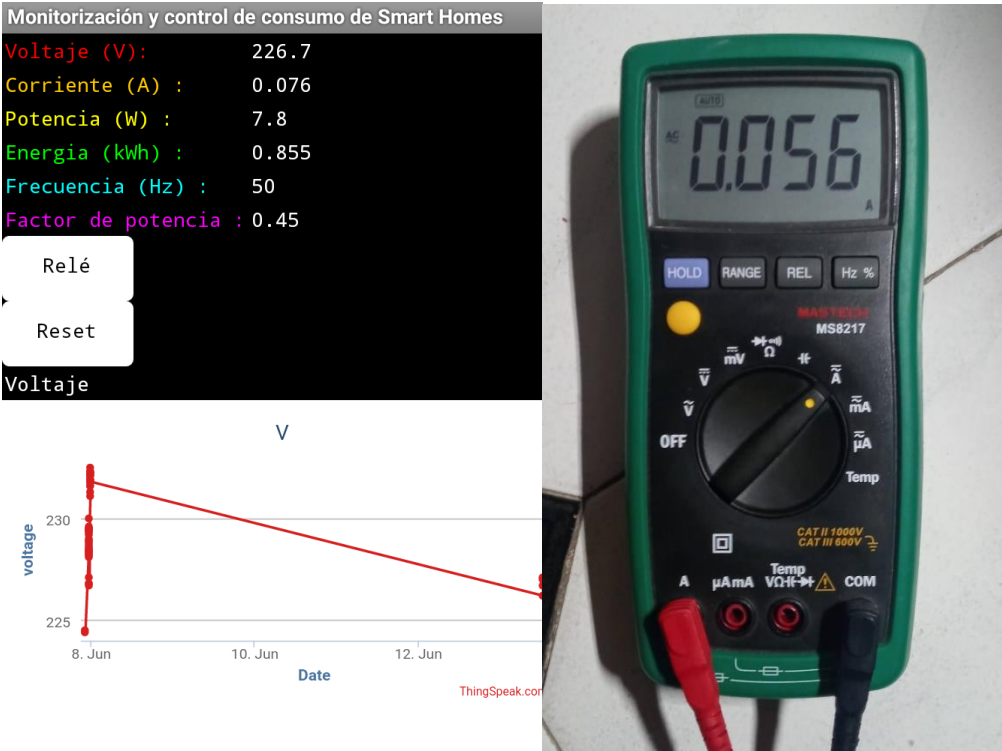


Figura 75: Lectura del dispositivo para la carga electrónica

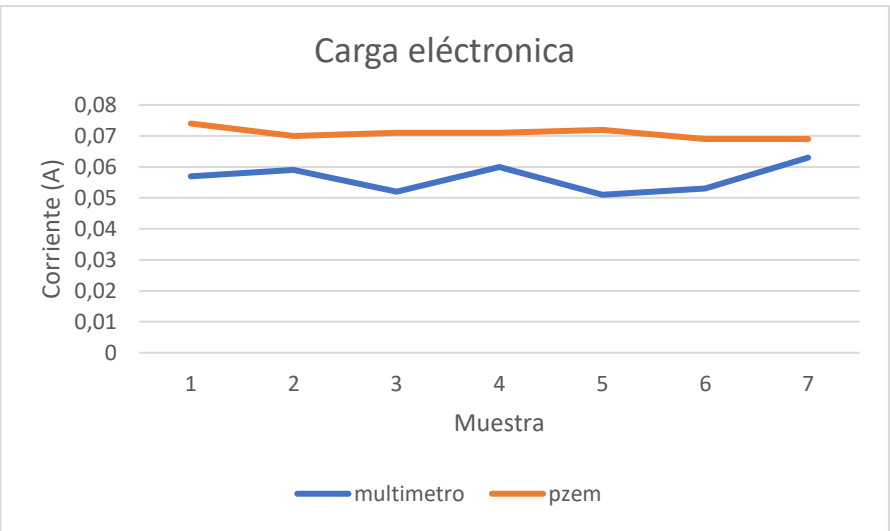


Figura 76: Medidas para carga electrónica

Medidas realizadas con el multímetro (A)	Medidas realizadas con el dispositivo (A)
0,057	0,074
0,059	0,07
0,052	0,071
0,06	0,071
0,051	0,072
0,053	0,069
0,063	0,069

Tabla 7. Carga electrónica

Cálculo de errores

$$e_{\text{absoluto}} = |\text{Dispositivo} - \text{Multímetro}| = |0,496 - 0,395| = 0,101$$

$$e_{\text{relativo}} = \frac{e_{\text{absoluto}}}{\text{Multímetro}} * 100 = 25,53\%$$

En este caso podemos ver que el error a la hora de medir corriente es mayor que en las cargas resistivas, esto se debe a la presencia de este tipo de corrientes parásitas en las instalaciones eléctricas conocidas como no lineales. Aunque la diferencia parezca ser mucho mayor al ser valores de corriente tan pequeños, la variable que realmente nos interesa para el consumo de energía es la potencia, la cual sí es correcta.

Tomando la muestra realizada por nuestro dispositivo podemos calcular la potencia que nos muestra.

$$V * A * \cos(\varphi) = 226,7 * 0,076 * 0,45 = 7,753 \text{ W}$$

Podemos ver que el consumo de estos dispositivos es muy pequeño, pero la mayoría del tiempo son productos dejamos siempre conectados malgastando energía poco a poco, y si son dejamos muchos puede llegar a notarse en la factura.

5.3. Otros ensayos

Se deseaba llevar a cabo otros ensayos con distintos elementos comunes de un hogar, como por ejemplo distintos tipos de luminarias, pero al ser la mayoría de ellas resistivas, podemos hacer una aproximación conforme a los ensayos realizados previamente con este tipo de cargas. También se planeaba utilizar el dispositivo con cargas inductivas con motor, como una batidora, con el objetivo de probar el comportamiento del dispositivo ante distintos factores de potencia. Dichos ensayos se han visto truncados debido a un error en el que nuestro dispositivo ha dejado de realizar sus funciones con normalidad y que requiere una sustitución de hardware, algo difícil de solucionar con rapidez si no disponemos de los elementos necesarios para realizar la sustitución del elemento dañado y nos sometemos de nuevo a los envíos lentos y con retrasos de algunas páginas web de proveedores, debido probablemente al actual desabasto en la producción de microchips debido a la pandemia por el COVID-19.

6. Conclusiones

Todo el desarrollo ha dado lugar a un dispositivo que logra nuestros objetivos, tal como se esperaba, realizando medidas precisas y permitiéndonos que sea monitorizado desde cualquier lugar.

Para ello se han utilizado los elementos más económicos que permitían el desarrollo de este proyecto, así como herramientas de software y desarrollo completamente gratuitas.

Se ha desarrollado una PCB de forma exitosa en la que se ha logrado integrar y conectar todos los componentes de la forma más compacta posible, convirtiéndolo en un dispositivo pequeño y portátil a la vez que funcional.

Hemos conseguido conectar nuestro dispositivo a Internet y enviar los datos de lectura a una base de datos, así como a una plataforma web que muestra gráficas de las lecturas a tiempo real. Todo ello mientras a la vez

podemos consultar dichos datos en la aplicación móvil, la cual ha sido desarrollada desde cero y con la que podemos interactuar con el dispositivo.

Este proyecto ha resultado ser muy enriquecedor al poseer muchos campos de trabajo como programación en Arduino, diseño electrónico y de PCB, soldadura, diseño de aplicaciones en Android y bases de datos. Además, resulta ser una interesante introducción al mundo del Internet de las cosas, así como a la investigación sobre el estado del arte, del hardware y del software actual que se encuentra disponible de forma accesible a prácticamente cualquier persona que se interese en el tema o quiera iniciarse en él.

6.1. Cambios futuros

Con el fin de mejorar el dispositivo actual, se van a proponer algunas mejoras de diseño:

- En la PCB pueden retirarse los conectores ya que los cables de alimentación y la pinza amperimétrica pueden conectarse directamente al PZEM004-T. Los conectores proporcionaban lecturas incorrectas, interrumpiendo el propósito del dispositivo. También reposicionar u orientar algunos elementos para que el cableado sea más sencillo.
- Optimización del código de Arduino para retirar líneas de código que no necesitemos y para que los datos se envíen de forma más rápida y efectiva.
- Integrar en la aplicación móvil la posibilidad de controlar varios dispositivos simultáneamente ya que actualmente está limitado solo a uno.
- Integrar una pantalla LCD para visualización de datos directamente del dispositivo otorgándonos otra forma rápida de poder visualizar las lecturas realizadas
- Realización de diseño para impresión 3D de una carcasa personalizada en la cual incorporar nuestro prototipo.
- Incluir datos de tarifas eléctricas según horas de uso y días para controlar el gasto económico de los elementos conectados.

7. Bibliografía

1. Paul, A., & Jeyaraj, R. (2019). Internet of Things: A primer. *Human Behavior and Emerging Technologies*, 1(1), 37-47.
2. Angulo Usategui, J. M., & Angulo Martínez, I. (2003). *Microcontroladores PIC: diseño practico de aplicaciones*.
3. Carballo B, Yusneyi (2011) *Algoritmos y Programación*
4. Richardson, M., & Wallace, S. (2012). *Getting started with raspberry PI*. "O'Reilly Media, Inc."
5. InnovatorsGuru (2019) *PZEM 004T-V3*
<https://innovatorsguru.com/pzem-004t-v3/> (Accedido: 10 abril 2021)
6. Prometec (2017) *ESP8266 para Arduino IDE*.
<https://www.prometec.net/esp8266-pluggin-arduino-ide/> (Accedido: 10 abril 2021)
7. PrimalCortex (julio 2019) *Measuring home consumption with the PZEM004T and ESP8266*
<https://primalcortex.wordpress.com/2019/07/06/measuring-home-energy-consumption-with-the-pzem004t-and-esp8266/> (Accedido: 13 abril 2021)
8. Llamas, L. (2019) *Cómo conectar un ESP8266 o ESP32 a una red WiFi (modo STA)* <https://www.luisllamas.es/como-conectar-un-esp8266-a-una-red-wifi-modo-sta/> (Accedido: 13 abril 2021)
9. Ude (2017) *Tutorial: Aprender a usar Wemos D1 Mini (Arduino+Wifi)*
<http://www.lunegate.net/2017/09/tutorial-aprender-usar-wemos-d1-mini.html> (Accedido: 15 abril 2021)

10. WEMOS (s.f) <https://www.wemos.cc/en/latest/> (Accedido 15 abril 2021)
11. Del Valle Hernández, L. (2019) *WifiManager librería para configurar red WiFi de un ESP8266* <https://programarfacil.com/esp8266/wifimanager-configura-wifi-esp8266/> (Accedido: 15 abril 2021)
12. Bauer, A. (2018) *ESP8266 + FIREBASE + APP ANDROID + LED* <https://texolab.net/2018/10/29/esp8266-firebaseapp-android-led/> (Accedido: 15 abril 2021)
13. Lledó Sánchez, E. (2012). Diseño de un sistema de control domótico basado en la plataforma Arduino (Doctoral dissertation, Universitat Politècnica de València).
14. Nguyen, D. (2018) *Firebase with realtime Database for IoT* <https://programarfacil.com/esp8266/wifimanager-configura-wifi-esp8266/> (Accedido: 17 abril 2021)
15. Kumar, P. (2018) *Relay Control Using Google Firebase* <https://www.hackster.io/pawan-kumar3/relay-control-using-google-firebase-7817ce> (Accedido: 17 abril 2021)
16. Santos, R. (2018) *Guide for Relay Module with Arduino* <https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/> (Accedido: 17 abril 2021)
17. Patton, E. W., Tissenbaum, M., & Harunani, F. (2019). MIT app inventor: Objectives, design, and development. In Computational thinking education (pp. 31-49). Springer, Singapore.
18. faisalamin (2018) *How to link Firebase to MIT App Inventor and Send or Retrieve data from Firebase database using MIT App Inventor 2* <https://steemit.com/utopian-io/@faisalamin/how-to-link-firebase-to-mit-app-inventor-and-send-or-retrieve-data-from-firebase-database-using-mit-app-inventor-2> (Accedido: 19 abril 2021)

19. Sánchez Acosta, A. (2016.) *La interfaz de App Inventor a fondo*
<https://rootsaid.com/wemos-d1-mini-home-automation-step-by-step-with-pcb-design/> (Accedido 19 abril 2021)
20. Ruiz, C. (2018) *Publicar datos de sensores en Internet a través de ThingSpeak* <https://openlanuza.com/publicar-datos-de-sensores-en-internet-thingspeak/> (Accedido: 20 abril 2021)
21. Tobi, M. D., & Van Harling, V. N. (2021, February). Prototype Design of Wireless Electric Energy Transmission System and Distance Electric Recording Record System Using PZEM004T and NRF24L01 Module. In Journal of Physics: Conference Series (Vol. 1783, No. 1, p. 012074). IOP Publishing.
22. [Engage Hub Kit – Mide tu Energía vía Web y Smartphone App] (s.f.) <https://es.efergy.com/engage-hub-kit/> (Accedido 21 abril 2021)
23. [Wattio Pack Energy] (s.f.) <https://wattio.com/es/tienda/pack-energy-8> (Accedido 21 abril 2021)
24. [OWL Intuition] (s.f.) <https://www.theowl.com/index.php/owl-intuition/> (Accedido 21 abril 2021)
25. [FIBARO Wall Plug] (s.f.) <https://www.fibaro.com/cl/products/wall-plug/> (Accedido 21 abril 2021)
26. [LEOTEC Enchufe inteligente WiFi] (s.f.) <https://www.pccomponentes.com/leotec-enchufe-inteligente-wifi> (Accedido 22 abril 2021)
27. [Fuente de alimentación AC-DC 5V] (s.f.) <https://es.aliexpress.com/item/32451069599.html> (Accedido 25 abril 2021)
28. [Relé de control 3V] (s.f.) <https://es.aliexpress.com/item/1005001994042589.html> (Accedido 25 abril 2021)

29. [ESP8266 WEMOS D1 Mini] (s.f.)
<https://es.aliexpress.com/item/4000420770002.html> (Accedido 26 abril 2021)
30. Koyanagi, F. (2018) Node MCU ESP8266: details and Pinout
<https://www.instructables.com/NodeMCU-ESP8266-Details-and-Pinout/>
(Accedido 26 abril 2021)
31. Jithin (s.f.) *Wemos D1 Mini Home Automation Step by Step with PCB Design* <https://rootsaid.com/wemos-d1-mini-home-automation-step-by-step-with-pcb-design/> (Accedido 2 mayo 2021)
32. [PCB Trace Width Calculator] (s.f.) <https://www.4pcb.com/trace-width-calculator.html> (Accedido 2 mayo 2021)
33. EasyEDA (s.f.) <https://easyeda.com/es> (Accedido 5 mayo 2021)
34. [Fabricación de PCBs JLCPCB] (s.f) (Accedido 6 mayo 2021)
35. Cano Ortega, A., Sánchez Sutil, F. J., & De la Casa Hernández, J. (2019). Power factor compensation using teaching learning based optimization and monitoring system by cloud data logger.
36. Poyatos Lozano, M. (2019). Desarrollo de un dispositivo IoT para la telemonitorización del consumo eléctrico
37. Atlam, H. F., Walters, R., & Wills, G. (2018). Internet of Things: state-of-the-art, challenges, applications, and open issues. *International Journal of Intelligent Computing Research (IJICR)*, 9(3), 928-938.
38. Navas Damas, M. (2018). Diseño, construcción e implementación de un prototipo de producto IoT.
39. Ferreras Astorqui, I.M. (2016) Sensor lot para monitorización de consumo de energía en continua.
40. Carrillo Quero, I. (2015) Sistemas para monitorización y control remoto de dispositivos eléctricos en viviendas.

8. Anexos

8.1. Anexo 1 – Presupuesto

Componente	Precio
Fuente de alimentación 5V	0,79€
Bornes PCB	0,54€
Relé 3V	0,48€
LOLIN WEMOS D1 Mini	3,25€
PZEM004-T V3	6,20€
5 PCB	6,51€
Total	17,77€

Tabla 8: Desglose de coste por material

8.2. Anexo 2 – Hojas de características

PZEM004T-V3 : <https://innovatorsguru.com/wp-content/uploads/2019/06/PZEM-004T-V3.0-Datasheet-User-Manual.pdf>

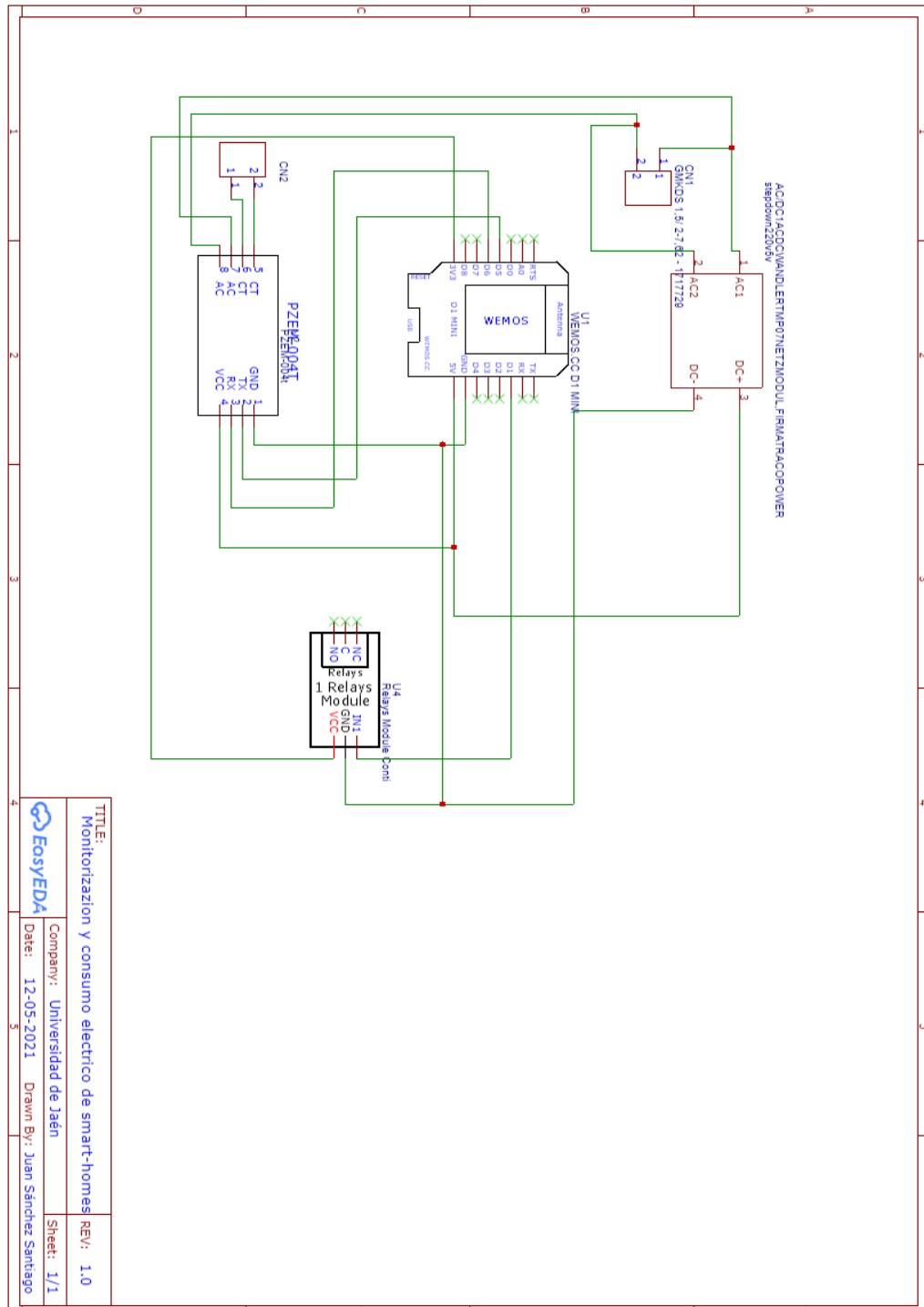
Relé: https://components101.com/asset/sites/default/files/component_datasheet/Relay-Module-Datasheet.pdf

ESP8266: https://www.elecrow.com/download/ESP8266_Specifications_English.pdf

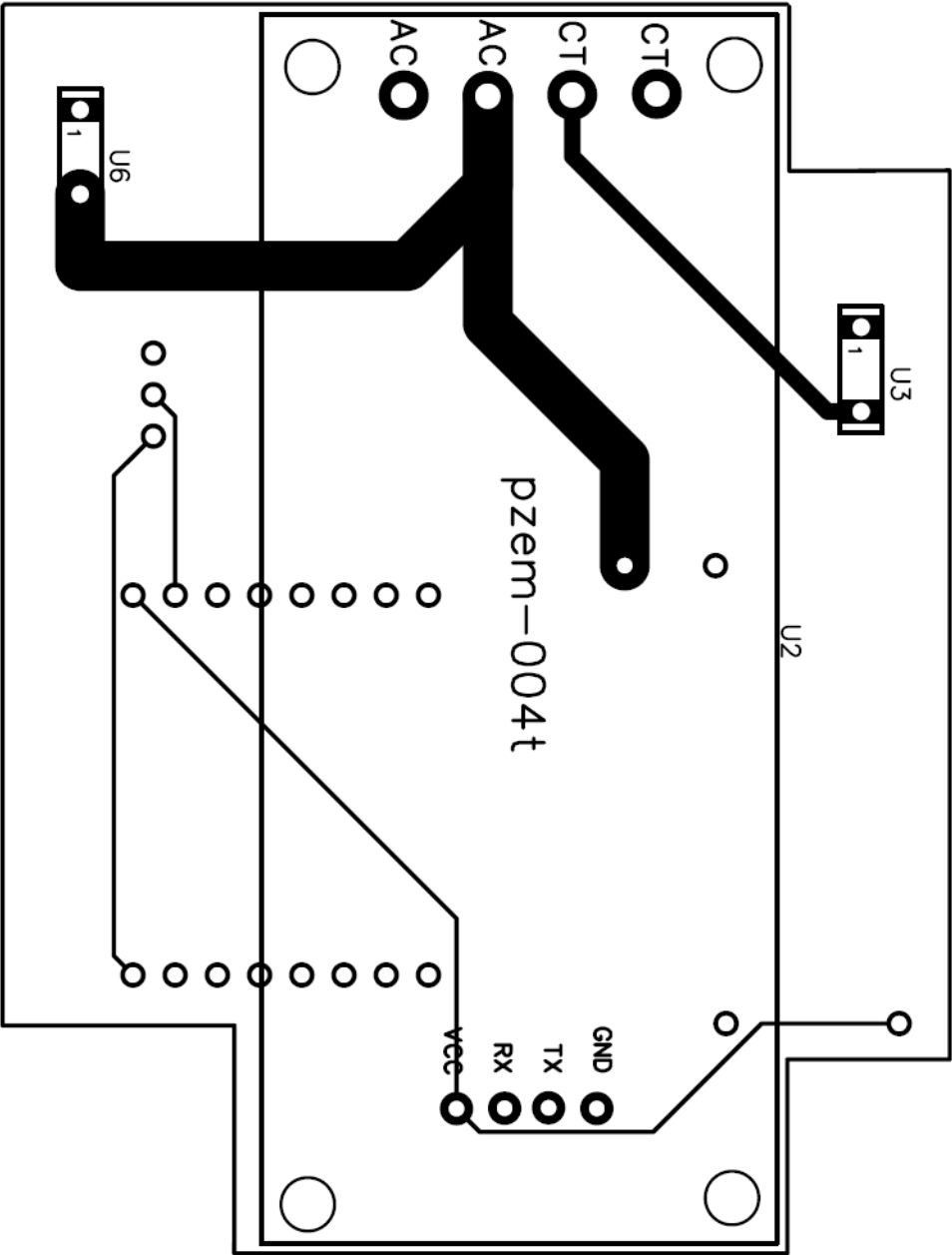
Bornes para PCB: <http://www.farnell.com/datasheets/2617011.pdf>

8.2. Anexo 3 – Esquemáticos

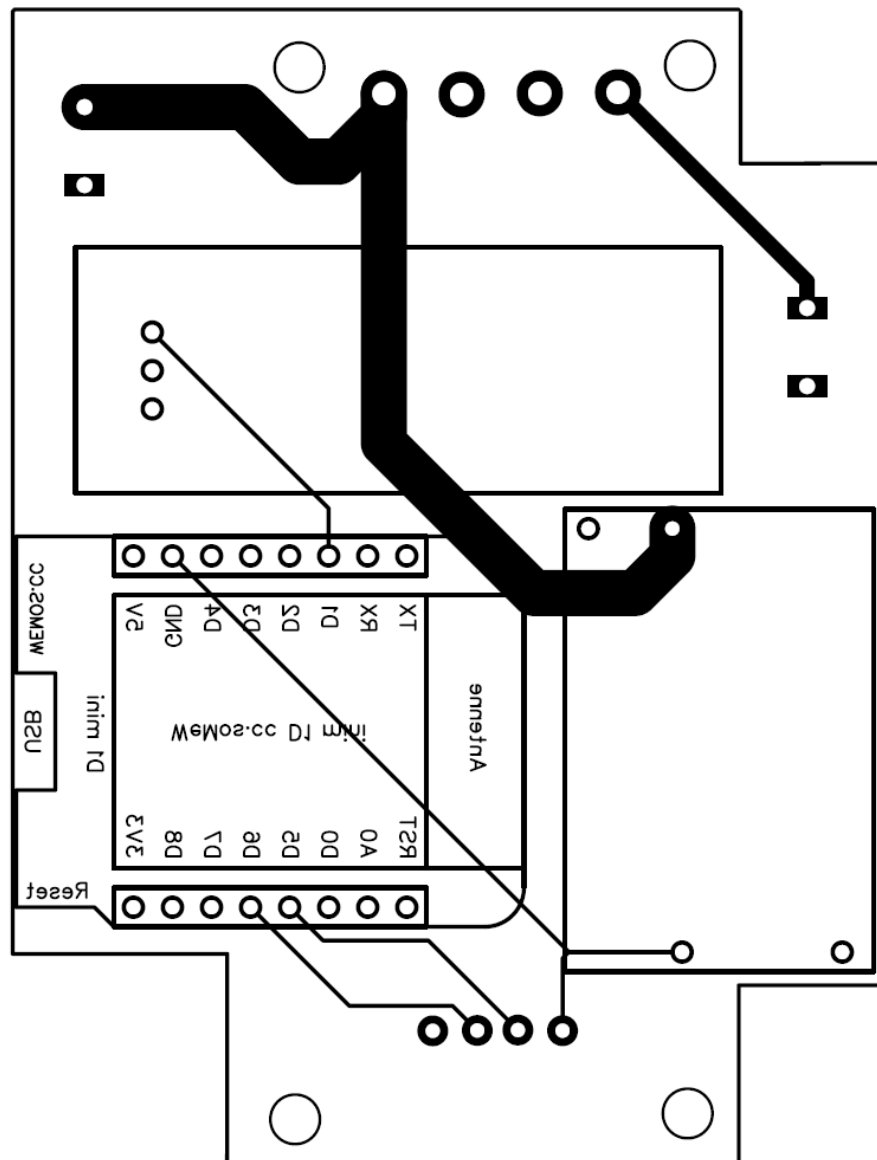
Esquemático de electrónica



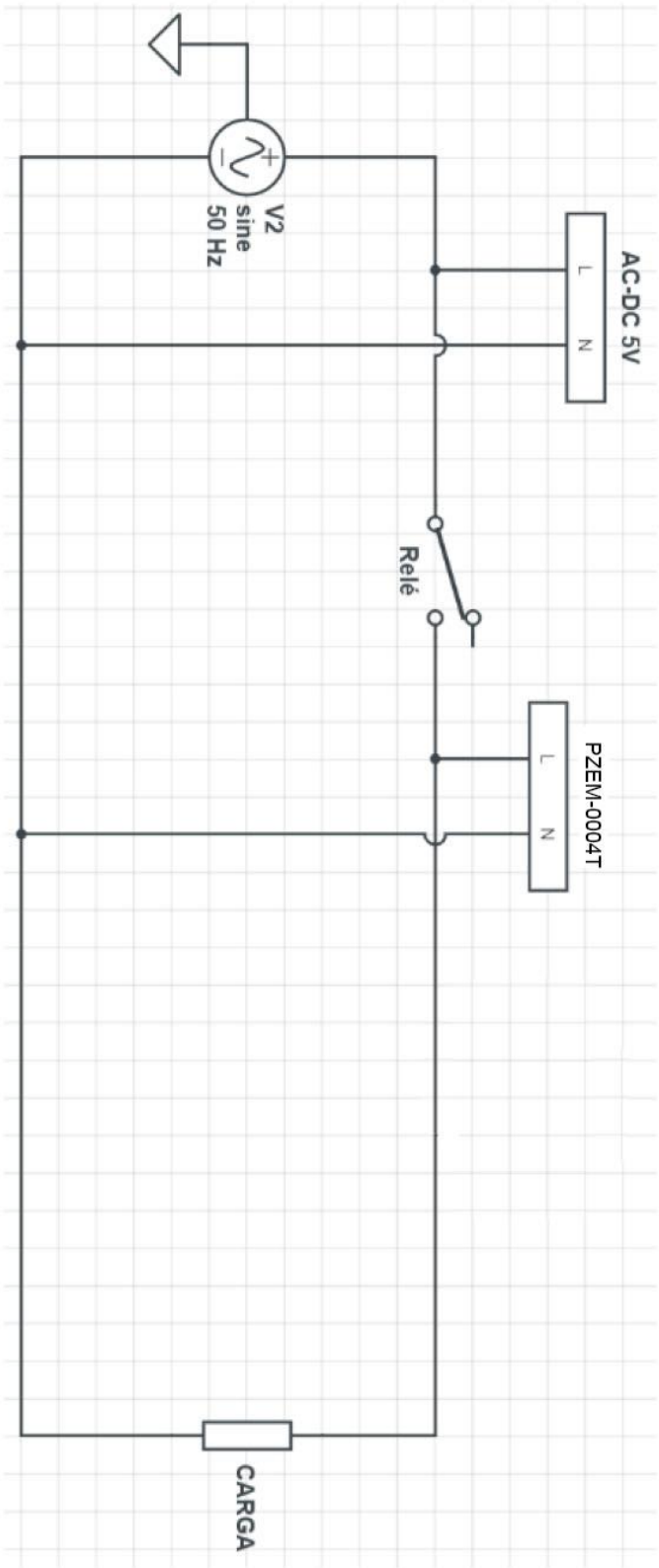
Esquemático enrutado PCB cara Top



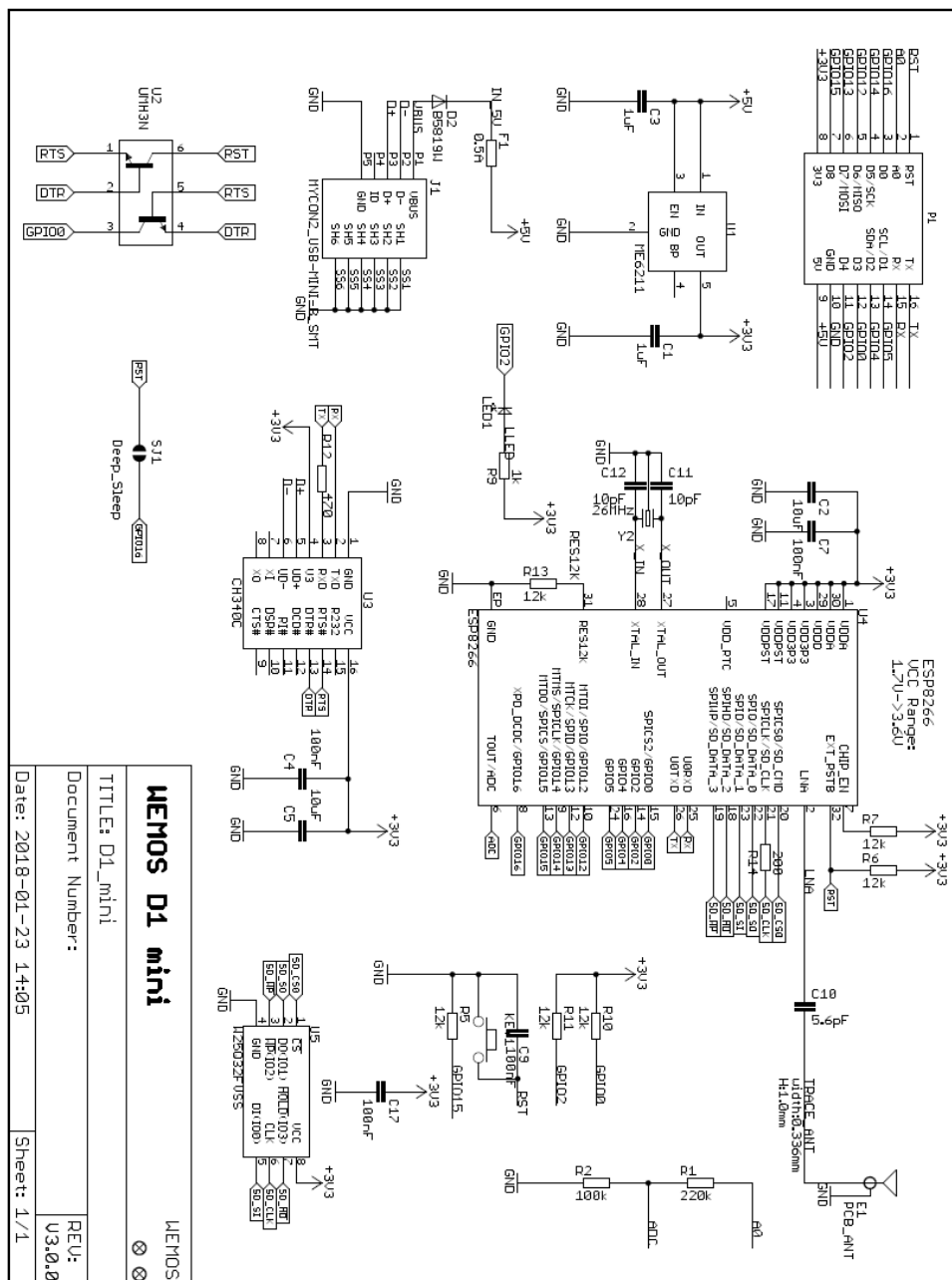
Esquemático PCB cara Bottom



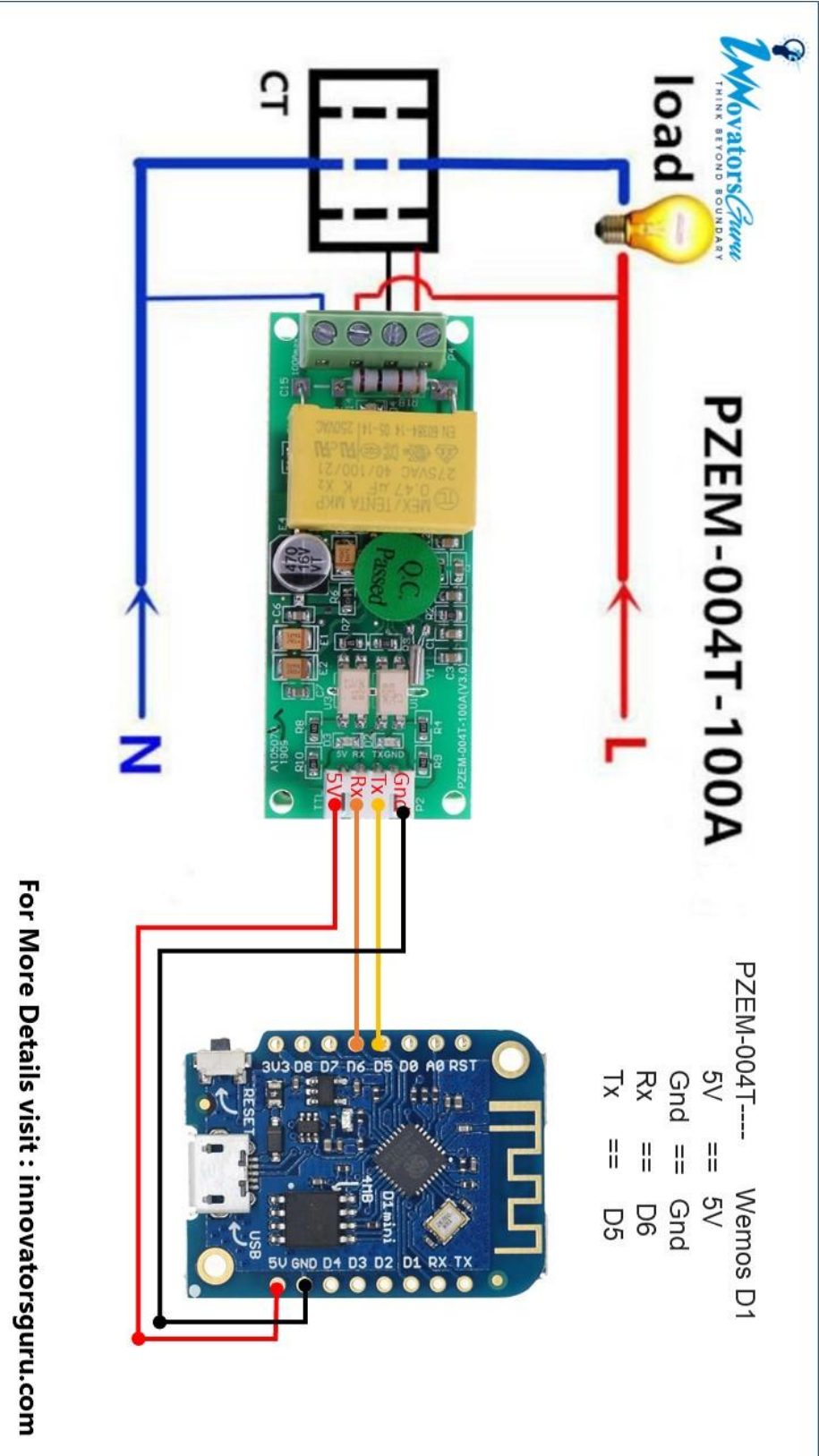
Esquemático del cableado de red



Esquemático WEMOS D1 Mini

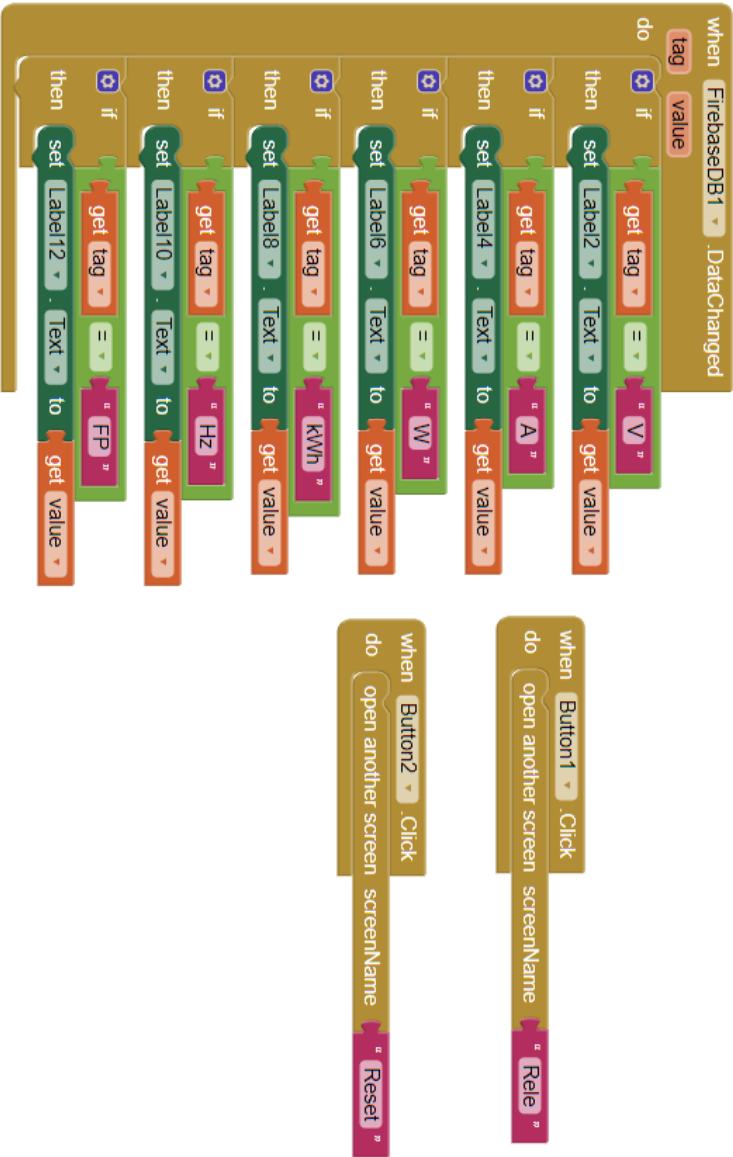


Esquemático conexión PZEM

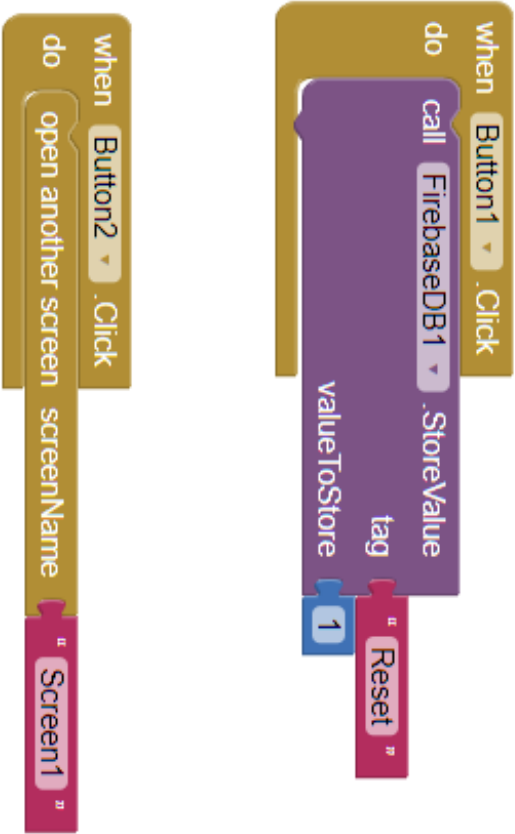


Diagramas de bloques MIT APP Inventor

Pantalla principal



Pantalla de Reset



Pantalla de control del Relé

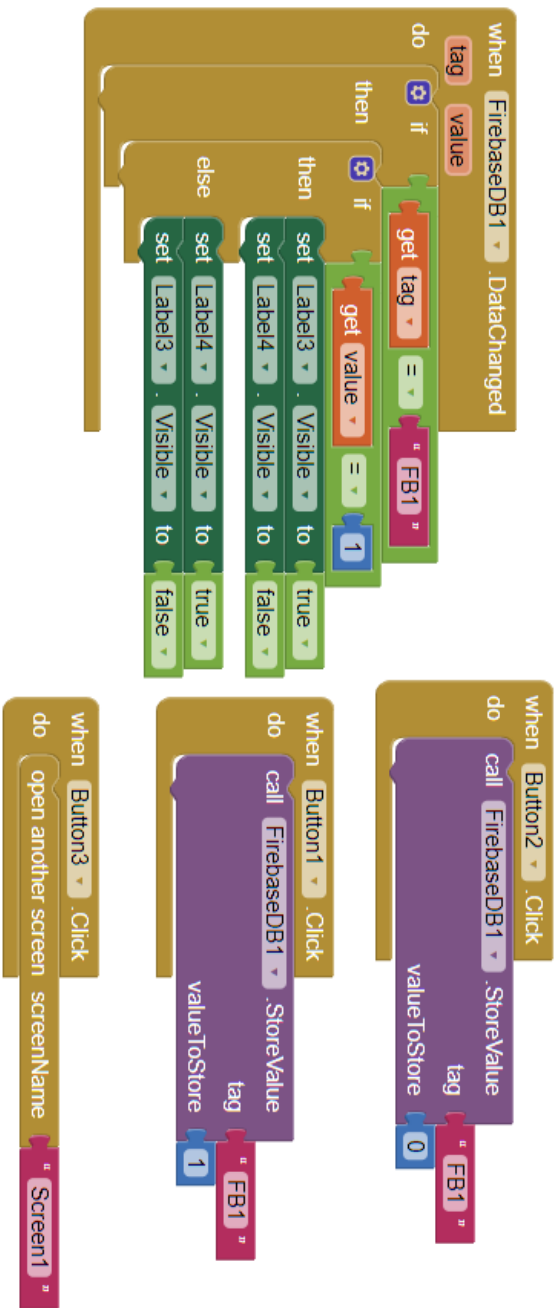


Diagrama de flujo del software de medida