

## TRABAJO FINAL DE GRADO

**Grado en Ingeniería Electrónica Industrial y Automática**

# **SISTEMA DE ADQUISICIÓN DE DATOS CON ESP32**



## **Memoria**

**Autor/a:** Ikiss, Jamal  
**Director/a:** Casellas Beneyto, Francisco  
**Convocatoria:** Setiembre 2020



## Resumen.

El presente trabajo final de grado, TFG, muestra el procedimiento seguido para diseñar un sistema de adquisición de datos empleando el microcontrolador (MCU, siglas del inglés: *Micro-Controller Unit*) ESP32, se utiliza este módulo para realizar medidas de variables eléctricas como la corriente y la tensión. A partir de las variables anteriores calcula; la frecuencia de red, la energía y la potencia consumida por una máquina eléctrica o microred.

Para este trabajo se utilizan 2 placas ESP32; una se conecta a los circuitos de muestreo de las variables eléctricas, contruidos en una placa de circuito impreso (o PCB), esta placa sensa los datos y los envía por wifi a otra placa que está conectada a un ordenador. A parte del diseño de los circuitos, este documento también abarca la programación de los microcontroladores para realizar las funciones correspondientes. Los MCU se programan mediante el *software* de libre disposición Arduino IDE (del inglés: *Integrated Development Environment*), y para la representación de los datos se utiliza el lenguaje de programación LabVIEW de National Instruments.

Mientras se va conformando el TFG, se priorizará que el sistema esté totalmente aislado eléctricamente. Solo se debe manipular cuando el dispositivo esté desconectado de la red eléctrica, así se evita un posible choque eléctrico. También se va a dar importancia a la protección de los equipos, por este motivo hay aislamiento galvánico en algunas partes del circuito. Con el equipo final se harán pruebas y ensayos para obtener la precisión de este y se comparará con un multímetro comercial.

## Resum.

El present treball de final de grau, TFG, mostra el procediment seguit per dissenyar un sistema d'adquisició de dades emprant el microcontrolador (MCU, sigles del anglès: *Micro-Controller Unit*) ESP32, s'utilitza aquest mòdul per a realitzar mesures de variables elèctriques com el corrent i la tensió. A partir de les variables anteriors calcula; la freqüència de xarxa, l'energia i la potència consumida per una màquina elèctrica o microxarxa.

Per a aquest treball s'utilitzen 2 plaques ESP32; una es connecta als circuits de mostreig de les variables elèctriques, construïts en una placa de circuit imprès (o PCB), aquesta placa sense les dades i les envia per wifi a una altra placa que està connectada a un ordinador. A part del disseny dels circuits, aquest document també abasta la programació de les plaques per a realitzar les funcions corresponents. Els MCU es programen mitjançant el programari de lliure disposició Arduino IDE (sigles del anglès: *Integrated Development Environment*), i per a la representació de les dades s'utilitza el llenguatge de programació LabVIEW de National Instruments.

Mentre es va conformant el TFG, es prioritzarà que el sistema estigues totalment aïllat elèctricament. Només s'ha de manipular quan està desconnectat de la xarxa elèctrica, així s'evita un possible xoc elèctrica. També es donarà importància a la protecció dels equips, per aquest motiu hi ha aïllament galvànic en algunes parts del circuit. Amb l'equip final es faran proves i assajos per obtenir la precisió d'aquest i es compararà amb un multímetre comercial.

## Abstract.

To sum up, the current report shows the procedure followed to design a data acquisition system using the microcontroller called “ESP32”, using this module to perform measurements of electrical variables such as the current and the voltage. From the above variables I must calculate: the grid frequency, the energy and the power consumed by an electric machine or micro grid.

Moreover, two motherboards “ESP32” are used in this project: one is connected to the sampling circuits of the electrical variables, built on a protoboard (or PCB), this plate census the data and sends them by WI-FI to another plate that is connected to a computer. Aside from the design of the circuits, this document also covers the codes of the motherboards to perform the corresponding functions. The microcontrollers (C) are programmed using the freeware “Arduino IDE (Integrated Development Environment)”, for the data representation will be carried out through the programming language “LabVIEW of National Instruments”.

While the project is correctly checked over, it will be prioritized that the system stays fully electrically isolated, only it should be handled when the device is disconnected from the power grid, thus avoiding and preventing a possible electric shock. Device protection is also going to be important due to the galvanic isolation in some parts of the circuit. With the final device we will make test results and trials to achieve the accuracy and it will be compared with a commercial multimeter.

## Agradecimientos.

El primer agradecimiento va dedicado al tutor del proyecto; Francisco Casellas Beneyto, por su ayuda y compromiso para la realización del TFG, ha hecho de guía y me ha facilitado el material correspondiente para el diseño y construcción de este trabajo, teniendo en cuenta en todo momento las complicaciones causadas por la pandemia médica actual y las necesidades específicas del proyecto.

También quiero agradecer a mis compañeros de clase por las ideas y la ayuda que me han dado a lo largo del TFG y mis estudios de Grado de Ingeniería Electrónica Industrial y Automática.

Por último, agradecer agradezco a mis amigos y familiares el apoyo que me han dado durante toda la duración de este TFG y a lo largo de los estudios de ingeniería, ya que sin su apoyo emocional y motivación esto no hubiese sido posible.

## Palabras clave

- Microcontrolador.
- Esp32.
- Instrumentación electrónica.
- Sistema automatizado.
- DAQ.
- Multímetro.
- Arduino IDE.
- LabVIEW.
- Wifi.
- Comunicaciones inalámbricas.
- Interface hombre-máquina.
- Transductor.
- Sonda de Corriente.
- Filtro analógico.

## Glosario.

MCU: Microcontrolador, por sus siglas en inglés: *Micro-Controller Unit*.

CPU: Microprocesador, por sus siglas en inglés: *Central Processing Unit*.

IoT: internet de las cosas, por sus siglas en inglés: *Internet of Things*.

PLC: controlador lógico programable, en inglés: *Programmable Logic Controller*.

DAQ: sistema de adquisición de datos, siglas del inglés: *Data Acquisition*.

ADC: convertidor analógico-digital. Sus siglas en inglés: *Analog-Digital Converter*.

SoC: sistema en chip. Sus siglas en inglés: *System on a Chip*

PWM: modulación por ancho de pulso, por sus siglas en inglés: *Pulse-Width Modulation*.

TFT LCD: pantalla de cristal líquido de transistores de película fina, por sus siglas del inglés *Thin Film Transistor-Liquid Crystal Display*.

GPIO: entrada/salida de propósito general, por sus siglas del inglés: *General Purpose Input/Output*.

LPF: filtro pasa bajos, por sus siglas del inglés: *Low Pass Filter*.

IDE: Entorno de desarrollo integrado, por sus siglas del inglés: *Integrated Development Environment*

HMI: Interfaz hombre- maquina, por sus siglas del inglés: *Human Machine Interface*





# Índice.

<b>RESUMEN.</b>	<b>I</b>
<b>RESUM.</b>	<b>II</b>
<b>ABSTRACT.</b>	<b>III</b>
<b>AGRADECIMIENTOS.</b>	<b>IV</b>
<b>GLOSARIO.</b>	<b>VI</b>
<b>1. PREFACIO.</b>	<b>1</b>
1.1. Origen del trabajo.....	1
1.2. Motivación.....	1
1.3. Requerimientos previos.....	2
<b>2. INTRODUCCIÓN.</b>	<b>3</b>
2.1. Planteamiento del problema.....	3
2.2. Objetivos del trabajo. ....	3
2.3. Alcance del trabajo. ....	4
<b>3. ESTADO DEL ARTE.</b>	<b>5</b>
3.1. Efergy Engage Solar Kit: .....	6
3.2. Akozon multímetro inalámbrico DC.....	7
3.3. TFG: Implementación de un analizador de redes con Arduino. ....	8
<b>4. DISEÑO DEL PROTOTIPO PARA MONITORIZACIÓN ELÉCTRICA.</b>	<b>10</b>
4.1. Adquisición de variables.....	10
4.1.1. Arduino Uno Wifi Rev2 y Arduino Yun .....	11
4.1.2. Raspberry Pi 3 y Raspberry Pi Zero W .....	12
4.1.3. ESP32-WROOM.....	13
4.1.4. Comparativa y elección.....	14
4.2. Especificaciones de Diseño.....	16
4.3. Transductor de tensión.....	18
4.4. Circuito Transductor de tensión.....	19
4.5. Calculo componentes del circuito para la medida de tensión.....	21
4.6. Simulaciones transductor de tensión.....	25
4.7. Transductor de corriente.....	28
4.8. Cálculo componentes del circuito para la medida corriente.....	31

4.9. Simulaciones transductor de corriente. ....	32
4.10. Circuito generador de Offset. ....	36
4.11. Alimentación. ....	38
4.12. Diseño Final. ....	39
<b>5. SOFTWARE Y COMUNICACIONES. ....</b>	<b>41</b>
5.1. Canal de comunicaciones. ....	41
5.2. Protocolo de comunicaciones.....	42
5.2.1. ESP-NOW.....	43
5.2.2. Protocolo de transferencia de hipertexto (HTTP).....	44
5.3. El entorno de desarrollo y programación del MCU.....	45
5.4. Configuración Arduino IDE.....	47
5.5. Controlador servidor.....	47
5.6. Controlador cliente. ....	53
5.7. Interface hombre máquina.....	58
<b>6. ANÁLISIS DEL IMPACTO AMBIENTAL. ....</b>	<b>63</b>
<b>CONCLUSIONES. ....</b>	<b>64</b>
<b>PRESUPUESTO. ....</b>	<b>65</b>
<b>BIBLIOGRAFÍA. ....</b>	<b>67</b>
<b>ANEXOS. ....</b>	<b>70</b>
A1. <i>Datasheet</i> sondas amperimétricas SCT-013.....	71
A2. Especificaciones técnicas fuente de alimentación IRM-03-3.3.....	72
A3. Instalación del ESP32 en el Arduino IDE.....	73
A4. Código del ESP32 emisor utilizando protocolo ESP-NOW. ....	77
A5. Código del ESP32 receptor utilizando protocolo ESP-NOW. ....	80



# 1. Prefacio.

## 1.1. Origen del trabajo.

La idea de este trabajo proviene de querer automatizar una tarea, siguiendo la tendencia que está tomando la industria actual, de ahí ha surgido la idea de monitorizar el estado y los consumos de una máquina eléctrica. Esta funciona normalmente la realiza una persona especializada en el tema con unos equipos específicos que tiene coste de cientos de euros, para ello se ha pensado en el diseño y construcción de un multímetro digital inalámbrico que pueda utilizar cualquiera persona con unos conocimientos mínimos de electricidad.

También está relacionado con el crecimiento de las tecnologías del ámbito de las *IoT* (siglas del inglés: *Internet of Things*) ha dado la idea de hacer un proyecto de interconexión digital entre objetos, intercambiando datos mediante radiofrecuencias, y que se pueda aplicar en muchos campos.

Reducir el riesgo de contacto eléctrico para las personas y equipos, en actividades laborales y/o educativas, como las practicas realizada en la asignatura de sistemas electrónicos del grado de ingeniería electrónica industrial y automática, sobre todo para las personas que no tienen tantos conocimientos sobre electricidad/electrónica.

Uno de los orígenes de trabajo, es el uso de eficiente de la energía, controlando el consumo energético de una máquina, se puede realizar un estudio de eficiencia energética, valorando así su rendimiento e impacto ambiental.

## 1.2. Motivación.

La motivación que me ha llevado a realizar este proyecto es para mejorar la calidad de vida de las personas que quieran realizar una tarea relacionada con la medición de variables eléctricas. Diseñar un producto que pueda competir con los que hay actualmente, crear un sistema con un presupuesto baja aprovechando las nuevas tecnologías. Reducir al mínimo

posible el riesgo de sufrir una descarga eléctrica mejorando la seguridad, protegiendo personas y equipos. Ayudar a la monitorización remota de equipos.

Por último, realizar un trabajo final de grado con los problemas sanitarios actuales, superando las limitaciones que esta supone.

### **1.3. Requerimientos previos.**

Para poder realizar este proyecto es muy importante haber adquirido los conocimientos del Grado de Ingeniería Electrónica Industrial y Automática, ya que con este proyecto se tratan muchos campos relacionados con estos estudios.

Empezando por conocimientos de electricidad y electrónica, como análisis de circuitos y temas ya más complejas como filtros y tratado de señales. También se requieren nociones de comunicaciones cableadas como: USB; e inalámbricas como: wifi y Bluetooth, y más importantes conocimientos sobre protocolos de comunicaciones. Así mismo, es necesario saber tratar datos y visualizar estos de la manera más sencilla e intuitiva posible.

Es muy importante aplicar las normas de seguridad para trabajar con los sistemas eléctricos presentes, sobre todo cuando se realicen ensayos, protegiendo a las personas de choques eléctricos y a los equipos electrónicos de cortocircuitos y sobretensiones.

## 2. Introducción.

### 2.1. Planteamiento del problema.

Actualmente el ser humano está viviendo una época con grandes avances tecnológicos, entre los cuales se encuentra la automatizando de actividades industriales. Un sistema automatizado utiliza elementos y componentes especiales de *hardware* y *software* dedicados a implementar sistemas de control y monitoreo como; ordenadores, PLC, robots y otras tecnologías para manejar diferentes procesos productivos, y control de maquinaria industrial. Con el fin de eliminar o evitar la intervención de la mano de obra en el proceso, evitando que las personas realicen tareas peligrosas y/o repetitivas. Con estos sistemas industrial también se obtiene una mayor precisión, calidad y mayor rendimiento consiguiendo una producción más alta y eficiente.

En un control automatizado, se pueden adquirir simultáneamente un amplio número de variables de proceso como temperatura, caudal, presión, distancia, niveles de líquido y variables eléctricas como pueden ser; corriente, tensión, potencia, etc. Las variables se adquieren, procesan y controlan por sistemas de procesamiento de datos, y posteriormente realizar acciones en función de los resultados obtenidos.

### 2.2. Objetivos del trabajo.

El proyecto va a consistir en realizar un sistema automatizado. Para diseñarlo, se han establecido los objetivos, que se presentan a continuación:

- El objetivo más importante de este trabajo, es monitorizar las variables eléctricas de interés para cualquier carga en corriente alterna, que trabaje con una potencia límite preestablecida.
- Diseñar un sistema de medición inalámbrico, separando el equipo de donde se muestra la interface HMI (siglas del inglés: *Human Machine Interface*) del equipo de sensado de las variables, esto permitirá aislar el equipo de medida del equipo donde se muestran los datos, ya que estos últimos suelen ser más caros. También hay que proteger a las personas de posibles contactos eléctricos, por ello se estudiará las diferentes opciones disponibles para medir y estudiar la manera de aislar el sistema de cualquier contacto físico.

- El siguiente objetivo es crear un sistema de mediación replicable fácilmente, es decir un sistema sencillo que se pueda construir desde cero, para poder crear otro sistema que mida los consumos de otra carga de manera sencilla y rápida. Estudiar la posibilidad de medir variables eléctricas de diferentes cargas a la vez, para esto se tiene que crear una red de comunicación que se adecue a las necesidades específicas del proyecto.
- Como último objetivo, crear un equipo para sensar las variables de interés, con un sistema de comunicaciones y un posterior tratamiento de datos pensando en la reducción consumo energético.

## 2.3. Alcance del trabajo.

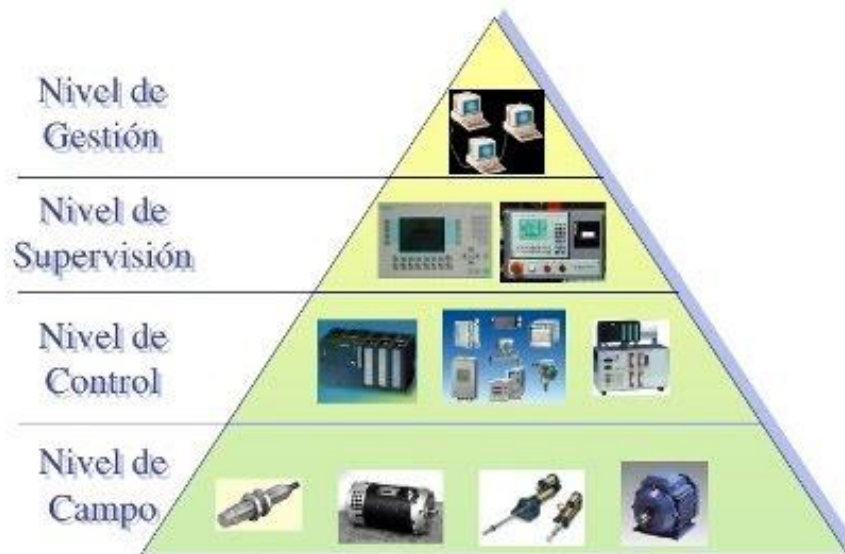
Después de haber descrito los objetivos del trabajo, con el presente TFG se pretende diseñar y construir un sistema de adquisición de datos que resuelva algunos problemas de monitorización de consumo eléctrico en la industria actual. Al ser un DAQ (siglas del inglés: *Data Acquisition*) que mida variables eléctricas, se diseñarán los correspondientes circuitos para medir corriente y tensión desde una placa con un MCU. También se elaborará un sistema de comunicaciones inalámbrico para transferir información desde el sistema de medida al receptor que estará conectado a un ordenador donde se van a tratar los datos, por lo tanto, el alcance del trabajo cubre también el diseño de una interface, donde se van a mostrar los datos medido y algunas variables de interés, por lo cual, abarcará la programación de los MCU en las placas y de la interfaz gráfica, donde se muestran los datos de interés en el ordenador.

El diseño a implementar tendrá unos límites de potencia que están pensados para redes de baja tensión monofásicas, aunque se podrían hacer cambios en el diseño para estudiar redes de alta tensión, pero ya las normas de seguridad son más estrictas. Los componentes para realizar el sistema serán escogidos pensando en un equipo de bajo consumo, y que los componentes tengan la mejor eficiencia posible, reduciendo al mínimo las pérdidas energéticas.



### 3. Estado del arte.

Este proyecto va a consistir en elaborar un sistema automatizado. Para estar en contexto en este TFG, hay que estudiar y entender las diferentes partes de este. Para entender su estructura, en la Figura 1, se presenta el esquema piramidal de un sistema automatizado, sirve para explicar de una manera simple y visual las diferentes partes de estos sistemas:



**Figura 2:** Pirámide de un sistema automatizado [2].

Esta se organiza por niveles. Empezando por la cota más baja, está el nivel de campo, donde se encuentran las máquinas como; motores actuadores, válvulas, y junto a estos también se ubican los sensores. Los equipos de campo están conectados de manera directa o indirecta con PLC y/o MCU mediante buses de comunicaciones, se encuentran en el nivel de control, en estos se ejecutan los programas que determinan las funciones y acciones de los equipos que hay en el área de campo, controlando las entradas y salidas de este. En el nivel de supervisión se encuentra los ordenadores con un software especial para visualizar y parametrizar los datos del proceso facilitados por el equipo de control. Por último, en la categoría más alto de la pirámide se encuentra el nivel de gestión, está formado por servidores que guardan la información del proceso a controlar, también está formado por ordenadores que se dedican a mostrar la información más importante y resumida del proceso, y donde se sacan las conclusiones del proceso automatizado.

Paralelamente a estos niveles se encuentra a la alimentación de cada uno de ellos, pues la fuente de energía que necesita cada equipo puede ser diferente, y los diferentes elementos de seguridad que lo conforman.

Por lo tanto, el presente proyecto va a tener una estructura piramidal parecida a la mostrada en la Figura 3. En este caso, habrá sensores que captan variables eléctricas en el nivel de campo, la función de estos va a consistir en captar las variables y acondicionar las señales, para que las pueda leer e interpretar el microcontrolador que se encuentra en el nivel de control. La función de este último consiste en leer las variables analógicas enviada por los sensores, convertirlas a digitales y enviarlas al equipo de supervisión. El equipo de supervisión, muestra los datos recibidos de una manera clara, también debe hacer las operaciones necesarias a partir de los valores recibidos. Por último, se hace la gestión de la información final, ya sea guardando está en un servidor o disco duro, o mostrar un resumen en un servidor web.

Se van a analizar 2 productos de diferentes fabricantes, para determinar las características que puedan interesar en el trabajo, y conocer los productos que hay actualmente en el mercado. Después se muestra el TFG de otro autor que implementa un analizador de redes mediante Arduino.

### 3.1. Efergy Engage Solar Kit:

Efergy Solar Kit es un producto de la compañía Engage, va acompañado de un sistema de monitorización online pensado especialmente para la producción de energía mediante palcas fotovoltaicas. Con este kit se puede monitorizar la demanda de energía eléctrica de la carga y la generada mediante los paneles fotovoltaicos, con la restricción de que este tiene que estar conectado con la red eléctrica, pues no es apto para sistemas fotovoltaicos aislados.

El kit que se muestra en la figura 2, está formado por un equipo de medida (DAQ), este va conectado a la salida de las placas solares para medir la potencia generada, y se conecta otro DAQ en el cuadro de medida a la salida del inversor (DC/AC) del sistema para medir la potencia consumida por la carga. Para visualizar los datos el equipo dispone de una central (receptor) que se conecta al *router* para poder subir los datos a la nube y visualizarlos desde una página web. Los datos entre el DAQ y la central se transmiten por wifi. También da la

posibilidad de visualizar la información en un teléfono móvil en una APP, ya que los emisores no disponen de pantalla.

Precio: 107,95€.



**Figura 2.** Efergy Solar Kit [3].

### 3.2. Akozon multímetro inalámbrico DC.

Siguiendo la línea del producto anterior tenemos Akozon, un multímetro inalámbrico (comunicación por wifi en la banda de 2,4 GHz) para corriente continua. Es un medidor DC multifunción, este kit separa el equipo de adquisición de las variables eléctricas, del monitor donde se muestran la información, está especialmente pensado para monitorear el estado de baterías.

El módulo utiliza una pantalla TFT LCD (siglas del inglés: *Thin Film Transistor-Liquid Crystal Display*) a color (figura 3), muestra una información completa, fácil de usar y una mejor interacción. La pantalla de visualización con la interfaz predeterminada muestra una información general, y si se pulsa el botón, cambiaría mostrando más información. El *display* tiene una carcasa metálica, resistente al agua y al polvo, y el equipo que va conectado al circuito es de fácil integración.

Akozon se puede utilizar para medir parámetros diversos como voltaje, corriente, potencia, capacidad de carga y descarga, energía consumida, tiempo y temperatura. El instrumento puede identificar automáticamente la dirección de la corriente y la capacidad de la batería puede monitorearse en tiempo real.

Al ser un producto comercial, este consta de algunos elementos de seguridad como protección de sobreintensidad y protección de altos voltajes. Los accesorios incluyen línea de transmisión de datos USB y sensor de temperatura.

Precio: 47,69€.



**Figura 3:** Multímetro inalámbrico Akozon [4].

### 3.3. TFG: Implementación de un analizador de redes con Arduino.

Este es un proyecto realizado por el estudiante: Angel El Barbáchano Cirión en la universidad Pública de Navarra. Ha diseñado un analizador de redes con Arduino, este consiste de un equipo para estudiar el consumo de electricidad doméstica.

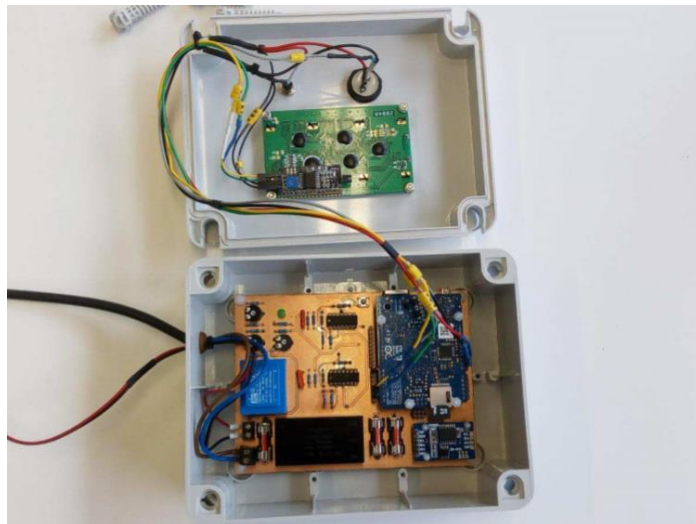
En este trabajo utiliza un Arduino Yun como microcontrolador, y mediante transformadores mide la tensión y la intensidad. En el mismo equipo trata los datos y la información más importante la muestra en una pantalla. Las muestras de los valores de las señales adquiridas

las sube a la nube, utilizando un servidor web, a parte también guarda la información en una tarjeta memoria SD de 8 GBytes.

Como se observa en la figura 4, el equipo está integrado en una caja central de electricidad, este equipo se alimenta mediante una fuente pequeña para PCB. A parte mostrar los datos en la pantalla, crea un servidor web para mostrar los datos más importantes, ya que el Arduino Yun lo permite.

Este equipo permite monitorear valores instantáneos de voltaje e intensidad, potencia instantánea y factor de potencia. A parte permite visualizar y controlar los valores históricos de energía consumida, y los picos máximos de las variables eléctricas.

Precio: 132,52€.



**Figura 4:** TFG realizado por: Angel El Barbáchano Cirión [5].

## 4. Diseño del prototipo para monitorización eléctrica.

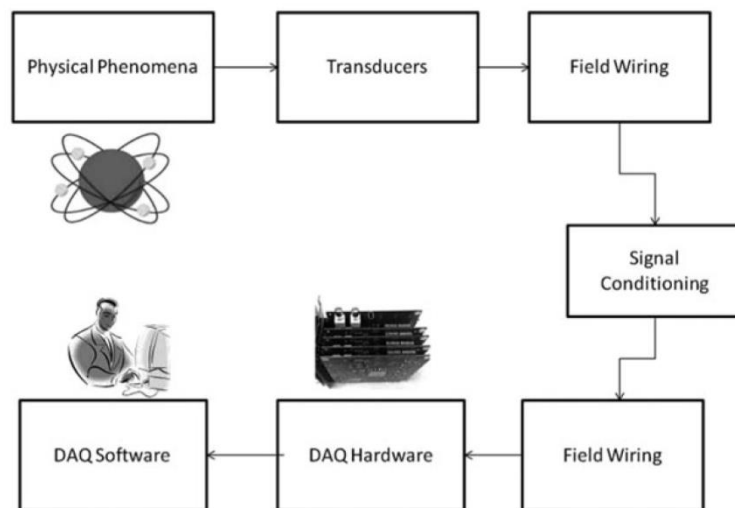
### 4.1. Adquisición de variables.

El muestreo de datos es una acción en la cual variables físicas de interés (temperatura, radiación, distancia a la que está situado un objeto, tensión, corriente, etc.), generalmente analógicas, se captan con sensores, y son convertidas en una señal eléctrica. Guardando una proporcionalidad entre la variable eléctrica y el fenómeno físico, posteriormente se acondiciona y filtra para que pueda ser tratada por los sistemas electrónicos encargados de captarla. La señal resultante es captada y convertida en datos digitales mediante ADC (siglas del inglés: *Analog-Digital Converter*), estos convertidores discretizan la señal, es decir, leen un valor de esta periódicamente, pasando de tener una señal continua en el tiempo a una discreta (digitalizada). Posteriormente estos datos son leídos por una tarjeta de adquisición de datos, y los comunica a un PC o servidor, donde se procesan, operan y presentan. En definitiva, un sistema de adquirente consiste en adquirir, procesar y presentar una variable de un determinado fenómeno físico, dependiendo de la finalidad también hay que registrar los datos.

El ciclo que forma el conjunto, también es conocido como cadena de medida. Un sistema de adquisición de datos automatizado (figura 5) tiene las siguientes partes,

- Transductores.
- Acondicionamiento de señal (amplificación y filtrado).
- Comunicaciones/ Transmisión de datos.
- Controlador DAQ.
- Presentación en PC (*software* especializado).

Para diseñar este sistema se va a empezar por el estudio de la tarjeta de adquisición de datos, esta es la parte que captura y muestrea los datos provenientes de los sensores. Se ha empezado por esta parte el estudio, porque para diseñar los sensores y el acondicionamiento de señal se necesitan algunos datos y características del microcontrolador encargado de adquirir los datos.



**Figura 5:** Esquema sistema de adquisición de datos [7].

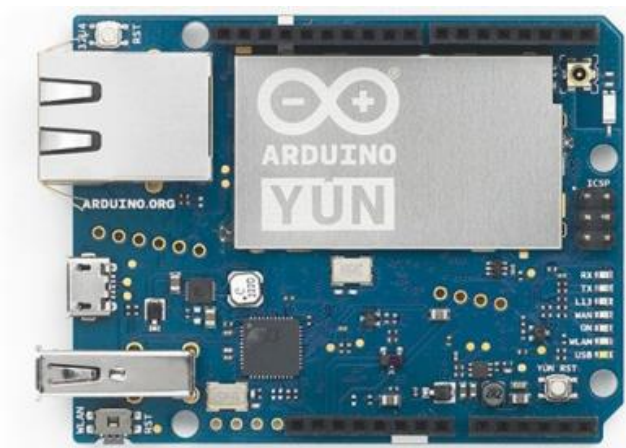
El *hardware* DAQ está formado por un dispositivo microcontrolador, capaz de convertir las señales analógicas facilitadas por los sensores a señales en formato digital (ADC), muestrear las señales y posteriormente se envía la información a un PC donde es tratada y almacenada. Por aspectos de seguridad y versatilidad en este caso los datos se envían al PC de forma inalámbrica. En el mercado actual se pueden encontrar diferentes microcontroladores que permiten hacer la función de capturar los datos.

Un MCU es un circuito integrado programable, capaz de realizar acciones guardadas en su memoria, por lo tanto, programar un MCU con los periféricos adecuados es capaz hacer la función de módulo de adquisición de datos. El microcontrolador escogido para realizar el proyecto es imprescindible que tengan comunicaciones inalámbricas (WIFI y/o Bluetooth). A continuación, se presentan las opciones disponibles en el mercado.

#### 4.1.1. Arduino Uno Wifi Rev2 y Arduino Yun.

Los CPU (siglas del inglés: *Central Processing Unit*) de estos productos los fabrica ATMEL, son integrados en placas de prototipado con determinados periféricos por otros productores. Gran parte de los productos de este fabricante están diseñados para persona que se van a iniciar en el mundo de la electrónica y la automatización, generalmente sus no incorporan comunicaciones bluetooth o wifi, para estos casos si se necesita de esta opción, hay que adquirir el módulo wifi ESP8266 (ESP01) y conectarlo a la placa Arduino.

Dada la gran necesidad de las comunicaciones sin cable y las facilidades que nos otorgan, el fabricante pone a disposición de los clientes modelos con wifi integrado como el Arduino Uno Wifi Rev2 o Arduino Yun (figura 6), este último está especialmente diseñado y fabricado para realizar proyectos relacionados con las IoT, es especialmente interesante, ya que incorpora un procesador (ATmega32u4) con la suficiente potencia como para integrar el sistema operativo Openwrt-Yún, esta es una distribución de Linux que se adapta a la placa, haciendo que sea ideal para interconectar dispositivos, conectarse a internet y/o crear un punto de acceso.



**Figura 6:** Arduino Yun [9].

Siguiendo con las especificaciones del Arduino Yun, encontramos que aparte de tener wifi, tiene soporte para Ethernet, puerto USB-A y micro USB para comunicaciones serie y ranura para tarjeta micro-SD para guardados datos relacionados con el proyecto tratado. Tiene 20 pines de entradas/Salidas, de los cuales 7 se pueden usar como salidas PWM (siglas del inglés: *Pulse-Width Modulation*) y 12 como entradas analógicas (tiene un ADC de 10 bits).

#### **4.1.2. Raspberry Pi 3 y Raspberry Pi Zero W.**

Pasando a otro microcontrolador capaz de realizar la función de tarjeta de adquisición de datos, que cumpla con los requisitos mínimos, encontramos los modelos Pi 3, pensado para proyectos de grandes que requieren un gran número de puertos y una mayor potencia de cálculo, puesto que la CPU que incorpora es de 4 núcleos. También fabrican el modelo



Raspberry Pi Zero W (figura 7), más sencillo, con menores dimensiones y un precio más contenido, haciéndolo interesante para la integración de dispositivos e IoT.

Pasando a las especificaciones técnicas del Pi Zero W, encontramos que monta un procesador ARM1176JZ (1 núcleo) que va a una velocidad de 1GHz, este módulo tiene comunicaciones inalámbricas bluetooth 4.1 y wifi 802.11n. Tiene 28 *pin*s GPIO (siglas del inglés: *General Purpose Input/Output*), cuenta con 2 puertos Micro USB, incorpora las funciones electrónicas de comunicaciones SPI, I2C y UART.



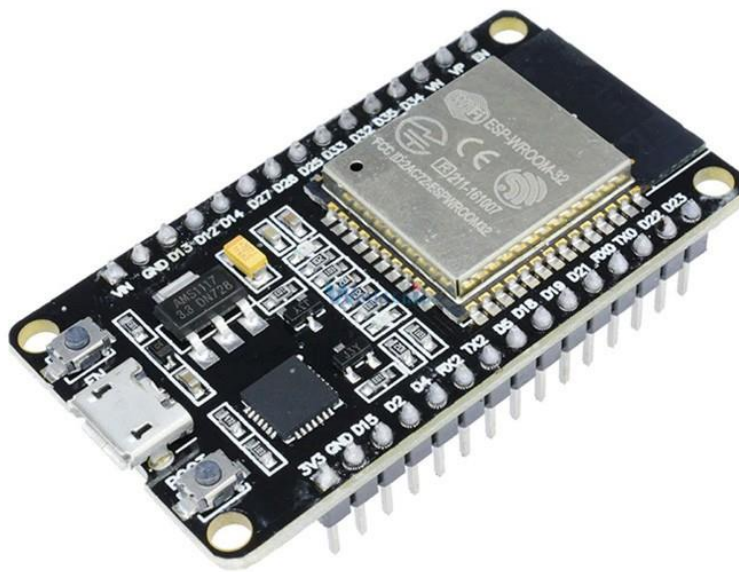
**Figura 7:** Raspberry Pi Zero W [10].

#### **4.1.3. ESP32-WROOM.**

El fabricante chino Espressif Systems produce los SoC (del inglés: *System On a Chip*) ESP32, la versión mejorada del ESP8266. Incorporan tecnología bluetooth y wifi. Con sus pequeñas dimensiones es ideal para interconectarse con otros dispositivos, fácil de integrar en placas de circuitos PCB o de circuito impreso. Las placas ESP32 incorporan el procesador *Xtensa LX6* de 2 núcleos que le otorga versatilidad para hacer 2 procesos a la vez. Son MCU de bajo consumo energético (tiene modo de bajo consumo *deep sleep*) y coste, el sistema de comunicaciones inalámbricas está especialmente cuidado porque incorpora algunas tecnologías como amplificador de potencia, amplificador receptor de bajo ruido, filtros, y módulos de administración de energía para reducir el consumo de esta cuando se esté usando.

Existen diversos modelos de placas ESP32 como: ESP32-WROOM (figura 8), ESP32-CAM, ESP32-DEVKITC, ESP32-WROVER, etc. Las placas tienen la misma base, que es el microprocesador, lo que cambia son los periféricos que incorpora, por lo tanto, la finalidad de cada módulo cambia según el modelo. Estos microcontroladores se pueden programar entre otros con Arduino IDE (siglas del inglés: *Integrated Development Environment*), característica que facilita su uso.

Resumiendo, las especificaciones técnicas encontramos, que el procesador de 32 bits tiene una velocidad máxima de 260 MHz. A parte de tener 4 comunicaciones SPI, 2 de I2C, 2 de I2S y 3 de UART y Bus CAN 2.0, tiene 36 pines GPIO; de los cuales 16 se pueden utilizar como salidas PWM, 18 pines pueden actuar como entradas analógicas. Para tratar los puertos analógicos utiliza 2 ADC de entrada múltiple. Sus convertidores son de 12 bits, característica que le otorga una mayor resolución para leer las señales analógicas.



**Figura 8:** ESP32-WROOM [13].

#### 4.1.4. Comparativa y elección.

A continuación, se presenta una tabla extensa (tabla 1) con los datos técnicos más importantes de los MCU analizados anteriormente.

**Tabla 1:** Comparación de datos técnicos de Arduino Yun, Raspberry Pi Zero W y ESP32-WROOM.

Características	Arduino Yun	Raspberry Pi Zero W	ESP32-WROOM
Procesador	ATmega32u4	ARM1176JZF-S	Tensilica Xtensa X36
Nº bits	8	64	32
Nº Núcleos	1	1	2
Velocidad CPU	400 MHz	1 GHz	160 MHz-240 MHz
Memoria	EEPROM 1 KB	Micro SD	-
SRAM	2,5 KB + 32 KB (Flash)	512 MB	512 KB
Alimentación	5 V	5 V	3 V- 3,6 V
Frecuencia reloj	16 MHz	19,2 MHz	40 MHz
Consumo	Desconocido	350 mA	80 mA-225 mA (2.5 uA en <i>deep sleep</i> )
Wifi	802.11b/g/n	802.11 b/g/n 2.4GHz	802.11 b/g/n 2,4 ~ 2,5 GHz
Bluetooth	no	V4.1 LE	v4.2 BR/EDR
Pines E/S	20	26	36
Salidas PWM	8	2	16
ADC (nº puertos/ bits)	1 / 12/ 10	1/ 16/ 10	2 / 18 / 12
DAC (nº puertos/ bits)	1/ 7/ 8	1/ 2/ 8	1/ 2/ 8
UART	-	1	4
I <sup>2</sup> C	1	2	2
I <sup>2</sup> S	-	1	2
SPI	1	2	4
Sensor <i>hall</i> , <i>touch</i> y temperatura	si	no	Si
Temperatura trabajo	-5 °C - +45 °C	Max 85 °C	-40 °C - 85 °C
Precio	58,8€	24,08€	8,99€

Después de analizar las características técnicas de los 3 MCU, el escogido para hacer este proyecto es el ESP32. La elección se debe a que tiene especificaciones técnicas más equilibradas. Analizando los MCU, la Raspberry a pesar de tener un solo núcleo, trabaja con un vector de datos de 64 bits a una mayor velocidad que el resto. El ESP32, aunque no sea el más rápido, vemos que es el único que tiene 2 núcleos, ideal para dedicar un núcleo al muestreo de señales y otro a las comunicaciones. Respecto a los canales de comunicaciones inalámbricas están bastante igualados, menos Arduino Yun que no tiene Bluetooth, pero en cambio tiene conexión Ethernet. El ESP32 tiene mayor número de pines de entradas y salida, además de que tiene mejores ADC, porque utiliza 12 bits, haciendo que tenga mayor resolución para la lectura de entradas analógicas, ideal para este proyecto.

Analizando el precio de venta en la tienda online Amazon, vemos que en relación calidad-precio, el mejor situado es el ESP32, ya que tiene unas grandes características a un precio muy competitivo, seguido de la Raspberry que tiene un precio más elevado porque tiene mayor potencia de cálculo y es más rápido. El precio del Arduino Yun es más elevado porque dispone de un segundo procesador para ejecutar el sistema operativo Linux, y también dispone de más canales de comunicaciones (Ethernet, USB, micro USB). Gracias a este precio tan competitivo del ESP32, va a permitir adquirir 2 placas ESP32 por un precio inferior al que vale una Raspberry Pi Zero W.

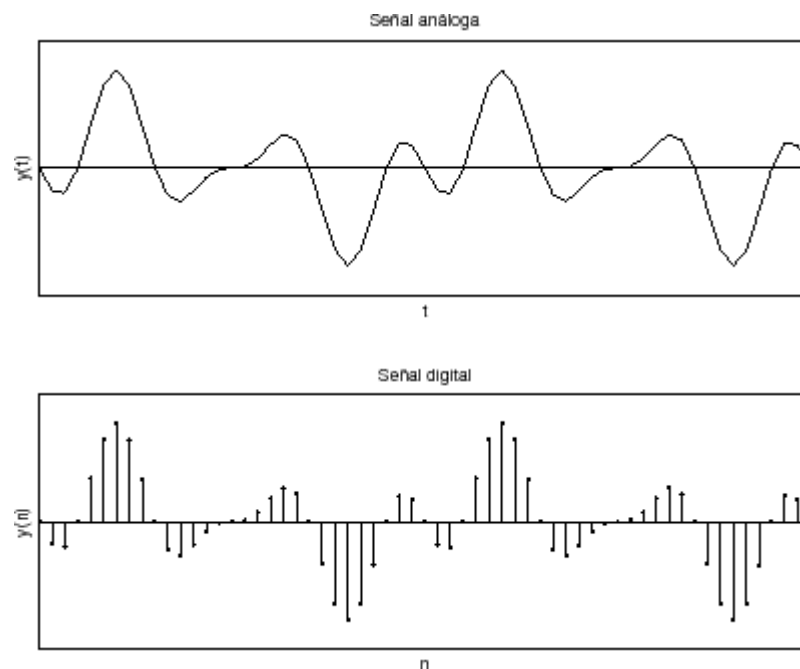
Mediante 2 placas ESP32-WROOM se diseñará un sistema automatizado que monitoree el consumo y el estado de la alimentación eléctrica de un equipo, un módulo se utilizará para adquirir y muestrear las señales eléctricas de corriente y tensión, y los envíe a otra placa conectada a un ordenador donde se analizan y tratan los datos.

## 4.2. Especificaciones de Diseño.

Antes de diseñar los transductores/sensores de corriente y tensión, cabe destacar y aclar algunas características del microcontrolador ESP32, ya que en función de estas se van a diseñar los circuitos.

Empezando por la definición básica de un ADC. Un conversor A/D es un chip que proporciona una representación digital de una señal de entrada analógica en un instante de tiempo, como se observa en la figura 9. En la práctica, las señales analógicas varían continuamente con el

tiempo y un conversor A/D realiza "muestras" periódicas de la señal a una frecuencia determinada. Posteriormente estas muestras se transfieren a un PC, donde es reconstruida para obtener una señal equivalente a la original. El conversor del ESP32 es de 12 bits, lo que quiere decir que la entrada puede tomar hasta 4095 valores diferentes. Por otro lado, como ya sabemos la tensión y la corriente de la red eléctrica es alterna, por lo que tiene valores negativos y positivos, y en cambio el microcontrolador solo trabaja con valores positivos entrada entre 0 V y 3,3 V, por lo tanto, hay que reducir, filtrar y añadir un *offset* a las señales de los circuitos de sensado para que entreguen unas señales dentro de esos márgenes. La placa nos permite obtener suficientes muestras de las señales analógicas para su posterior reconstrucción, ya que la frecuencia del cristal que incorpora es de 40 MHz.



**Figura 9:** Grafica de señal analógica con su correspondiente muestreo digital [14].

Los circuitos se diseñarán para medir como máximo tensiones de red eléctrica monofásica, y dejando un margen, es decir, para analizar fuentes de energía eléctrica con una tensión eficaz de 230 V, y permitiendo medir una corriente eficaz de hasta 25 A. por lo tanto el equipo se podrá usar para monitorizar máquinas de hasta 5,75 kW. El equipo también va a dar la posibilidad de monitorizar fuentes de energía continua.

### 4.3. Transductor de tensión.

El voltímetro es un instrumento de medición que se conecta en paralelo con una carga eléctrica y se utiliza para medir la diferencia de potencial eléctrico (voltaje) entre sus conexiones. El voltaje se conoce como la energía potencial eléctrica por unidad de carga, es responsable de la conducción de una corriente de un electrón a otro electrón. Al ESP32 no se le pueden conectar tensiones superiores a 3,3 V como por ejemplo tensión de red (230 V a 50 Hz) de manera directa, ya que lo destruirían inmediatamente. Para este proyecto se necesita diseñar el circuito para medir tensiones, disponemos de diferentes opciones, entre ellas:

#### **Transformador eléctrico.**

Un transformador es un componente eléctrico que emplea campos magnéticos para adaptar niveles de tensión y corriente entre dos circuitos. Además de ser utilizados para fabricar fuentes de alimentación, también pueden ser empleados como elementos de instrumentación. Se puede utilizar un transformador para adaptar la tensión de circuitos a un rango manejable para componentes electrónicos y microcontroladores.

Este componente supone un aislamiento galvánico entre el circuito primario y el secundario, no obstante, hay un acoplamiento magnético entre ellos, por lo cual el circuito secundario sigue siendo susceptible a grandes picos de tensión del primario, por este motivo hay que añadir un condensador de filtro en el secundario. Al aplicar una ganancia (relación de espiras) entre el primario y el secundario mediante 2 bobinados implica que se introduce un desfase entre las dos señales, complicando así el cálculo del coseno de factor de potencia. Depende de la calidad del transformador puede haber no linealidad de la ganancia entre los 2 circuitos al variar la tensión del primario. Generalmente suelen tener unas dimensiones y un precio considerables en comparación con otras opciones.

#### **Divisor de tensión**

Empleando un simple divisor de tensión con la ganancia adecuada, se puede convertir una señal de entrada independientemente de su naturaleza (AC y/o DC) como la tensión de red, a unas señales con las que pueden trabajar microcontroladores como el ESP32. Es una opción que no tiene aislamiento galvánico, pero se le puede incorporar filtros analógicos, para evitar que el componente de adquisición reciba grandes picos de tensión y sobrecargas de

altas frecuencias. Este circuito se caracteriza por utilizar componentes pasivos (resistencias y condensadores), por lo cual la ganancia del circuito en las frecuencias que deja pasar el filtro es lineal, también aporta la ventaja de que son componentes de bajo coste, dejando un circuito final de pequeñas dimensiones.

### Amplificador VFOA

También se puede utilizar un amplificador operacional, para diseñar un circuito con la ganancia adecuada para altas tensiones que no pueden soportar directamente las tarjetas de adquisición. Empleando un circuito basado en un VFOA (operador operacional) con los correspondientes componentes resistivos y capacitivos se puede conseguir una ganancia y una señal filtrada para medir diferencias de potencial como las de la red eléctrica. Estos componentes son lineales, por ellos las características son independientes de la potencia de entrada. La gran desventaja que tiene es que hay que alimentar el VFOA con una tensión simétrica de entre  $\pm 10$  V y  $\pm 30$  V.

**Tabla 2:** Comparativa de transductores de tensión.

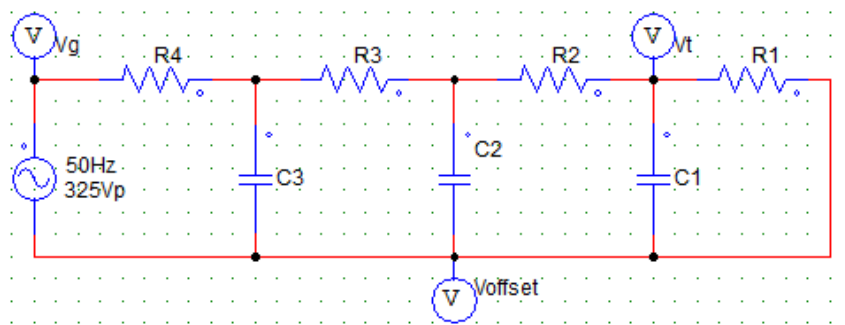
Características	Transformador	Divisor de tensión	VFOA
AC/DC	Solo AC	si	si
Aislamiento galvánico	si	no	no
Linealidad	no	si	si
Tamaño	grande	pequeño	medio
Precio	alto	bajo	medio

Después de indicar las ventajas y desventajas de cada tipo de circuito, encontramos que la mejor opción es utilizar un circuito divisor de tensión con un filtro pasa bajos integrados, la única desventaja que tiene es que no dispone de aislamiento galvánico.

## 4.4. Circuito Transductor de tensión.

Con la primicia de medir la tensión mediante un circuito divisor de tensión, se procede a diseñar el circuito. A parte de tener la ganancia adecuada para reducir los 650 V de pico a pico de la tensión de red a 3,3 V de pico a pico, que es con los que se puede trabajar el

ESP32, también hay que añadir un filtro pasa-bajos, para reducir los picos de alta frecuencia que se generan al conectar el equipo, transitorios de conexión y desconexión. Cuando se conecta un equipo a una fuente de tensión, en el caso de la red eléctrica se generan picos de tensión del orden de 10 kV con una duración de microsegundos. También hay que filtrar el paso de sobretensiones producidas por el impacto de un rayo u otro fenómeno atmosférico a la red eléctrica que podrían atacar al microcontrolador, estos suelen ser del orden de decenas de kilovoltios y de una duración de microsegundos, y también elimina las posibles perturbaciones y sobretensiones por fallos de la red eléctrica. Para ello se ha escogido un LPF de tercer orden, formado por 3 filtros LPF de primer orden en cascada como se observa en la figura 9, es la forma más simple de hacer un filtro pasa bajo de 3er orden, y este orden ya nos permite obtener una pendiente de ganancia de -60 dB por década, suficiente para atenuar las frecuencias altas. En cuanto a la fase de la señal adquirida, como máximo se va a utilizar para realizar medidas con señales que tienen una frecuencia de 50 Hz y 60 Hz, a estas frecuencias el desfase que introduce el circuito en la señal medida respecto a la original es mínimo, posteriormente se cuantifica en las simulaciones. El circuito final es el siguiente:



**Figura 10:** Circuito base para sensado de tensión.

En el circuito de la figura 10 podemos observar la fuente de alimentación a analizar, seguida de tres filtros RC que hacen de divisor de tensión con la resistencia R1. El pin Vt es la señal que va a muestrear el módulo. Por último, tenemos que el circuito en vez de estar conectada a tierra, está conectado a una tensión de *offset*, esto es para evitar que la señal a leer no tenga valores negativos. El uso de 3 filtros en cascada hace que la banda de paso sea lineal, como se observa en la figura 14, que tiene una atenuación despreciable, cosa que no ocurriría si se utilizasen 3 filtros iguales con la misma frecuencia de corte.

Con el diseño del circuito hecho, falta cuantificar el valor de los componentes. En el siguiente apartado se muestra el proceso y los cálculos realizados para determinarlos.



## 4.5. Cálculo componentes del circuito para la medida de tensión.

Dado el diseño del circuito transductor de la tensión de la figura 10, se procede a calcular el valor que deben tener los componentes para cumplir con su función. Para determinar el valor de los elementos del circuito, se comienza por los filtros. Empezando por determinar las frecuencias de corte de los 3 filtros, si se requiere que el conjunto de los filtros tenga una ganancia estable y constante, se fija la frecuencia de corte del primer filtro a 10 veces (ecuación 1) la de red, 500 Hz, se ha escogido esta porque los picos de tensión y ruido se producen a mayores frecuencias y también para asegurar mantener las frecuencias bajas, en la banda de paso entre las frecuencias bajas y los 100 Hz se introduce una ligera atenuación, posteriormente en las simulaciones se especifica el valor de esta. La frecuencia del siguiente filtro tiene que ser el doble que la anterior (ecuación 2 y 3), dejando así las siguientes ecuaciones:

$$f_{c1} = 10 \cdot f_R = 500 \text{ Hz} \quad (\text{Ec. 1})$$

$$f_{c2} = 2 \cdot f_{c1} = 1000 \text{ Hz} \quad (\text{Ec. 2})$$

$$f_{c3} = 2 \cdot f_{c2} = 2000 \text{ Hz} \quad (\text{Ec. 3})$$

Gracias al análisis de circuito de filtro analógicos realizados en la asignatura de electrónica analógica, se determina que la frecuencia de corte de un filtro pasa bajos RC, tiene la siguiente formula:

$$\omega_c = \frac{1}{RC} \quad (\text{Ec. 4})$$

$$f_c = \frac{1}{2\pi RC} \quad (\text{Ec. 5})$$

Sabiendo las frecuencias de corte de los filtros, y fijando un valor alto a las resistencias, para que la corriente que circule por los circuitos sea la mínima posible, por este motivo se ha seleccionado primero el valor de las resistencias, aunque se podría hacer a la inversa, tanteando con el valor de los condensadores hasta obtener un valor grande las resistencias. El valor de los condensadores se obtiene mediante la ecuación 6:

$$C = \frac{1}{2\pi f_c R} \quad (\text{Ec. 6})$$

$$R_1 = 5 \text{ k}\Omega; f_{c1} = 500 \text{ Hz} \rightarrow C_1 = \frac{1}{2\pi \cdot 500 \cdot 5 \cdot 10^3} = 63,6 \text{ nF}$$

$$R_2 = 50 \text{ k}\Omega; f_{c1} = 1 \text{ kHz} \rightarrow C_2 = \frac{1}{2\pi \cdot 1 \cdot 10^3 \cdot 50 \cdot 10^3} = 3,18 \text{ nF}$$

$$R_3 = 70 \text{ k}\Omega; f_{c1} = 2 \text{ kHz} \rightarrow C_3 = \frac{1}{2\pi \cdot 2 \cdot 10^3 \cdot 70 \cdot 10^3} = 1,14 \text{ nF}$$

Obtenidos los componentes de los filtros falta determinar el valor de la resistencia R4. Esta última resistencia se va a obtener mediante el cálculo de la ganancia del divisor de tensión formado por las resistencias. El módulo ESP32 como máximo lee valores de 3,3 V y no trabaja con valores negativos, y como se desea muestrear hasta tensiones de red y dejando un margen de seguridad, hay que poner una ganancia para reducir los 325 V de pico que tiene la red y poder ser capturada por el microcontrolador, hay que añadir una tensión DC (offset) con el valor de la mitad de la tensión de alimentación (ecuación 7) del MCU, centrando así la señal analizada en 1,65 V. Con las anteriores consignas fijamos la ganancia en 1/250, la tensión (Vd) de pico que se obtendría en el punto Vt sin la tensión de offset se calcula mediante la ecuación 8:

$$V_{offset} = \frac{V_{cc}}{2} = \frac{3,3}{2} = 1,65 \text{ V} \quad (\text{Ec. 7})$$

$$V_d = G \cdot V_g \quad (\text{Ec. 8})$$

$$V_t = G \cdot V_g + V_{offset} \quad (\text{Ec. 9})$$

$$G = \frac{1}{250} \rightarrow V_d = \frac{325}{250} = 1,3 \text{ V}$$

Por lo tanto, si se está analizando la tensión doméstica, el microcontrolador lee en Vt un valor máximo (ecuación 10) y mínimo (ecuación 11) de tensiones calculados a continuación:

$$V_{tm\acute{a}x} = V_{offset} + V_d = 1,65 + 1,3 = 2,95 \text{ V} \quad (\text{Ec. 10})$$

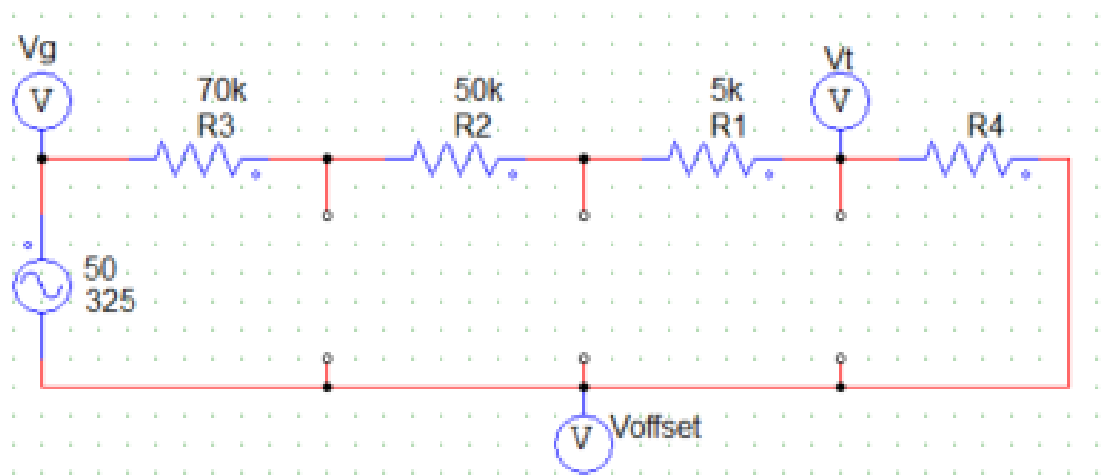
$$V_{tm\acute{i}n} = V_{offset} - V_d = 1,65 - 1,3 = 0,35 \text{ V} \quad (\text{Ec. 11})$$

Con esta ganancia, se dejaría un margen de lectura superior e inferior de 0,35 V, que equivalían a poder analizar tensiones de hasta 412,5 V como se indica en la ecuación 12:

$$V_{gm\acute{a}x} = \frac{V_{offset}}{G} = 1,65 \cdot 250 = 412,5 \text{ V} \quad (\text{Ec. 12})$$

El equipo va a permitir medir diferencias de potencial de hasta 412,5 V de pico, esto es 82,7 V más que la red doméstica. Este margen hace que el sistema sea más seguro para medir tensiones de 230V eficaces, y permite analizar fuentes de alimentación variables, siempre que su tensión de pico no pase los 412,5 V.

Después de determinar la ganancia, para calcular el valor de la resistencia R4, para eso hay que analizar el circuito en corriente continua, ya que el circuito divisor de tensión permite también monitorizar una fuente de corriente continua. Si se analiza el circuito en DC (figura 11), los condensadores serían circuitos abiertos:

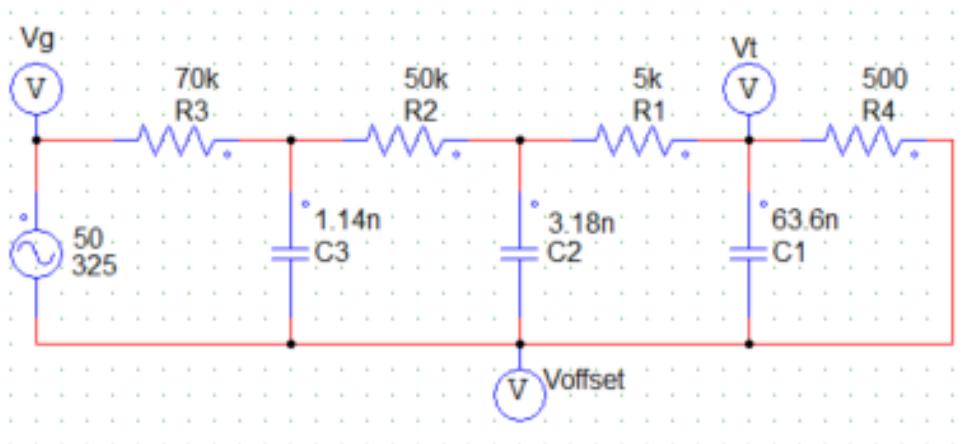


**Figura 11:** Circuito sin condensadores, en DC o a bajas frecuencias, por ejemplo 50 Hz.

Por lo tanto, el valor de R4 se puede calcular a partir de la ecuación 13;

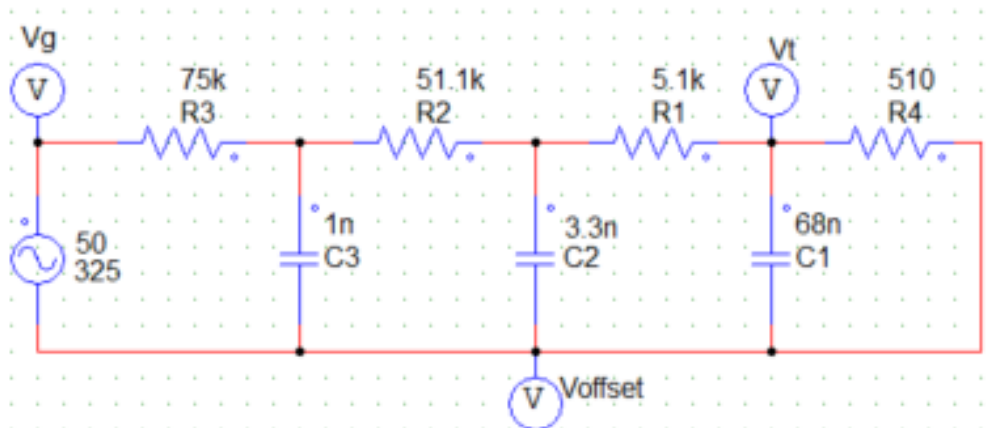
$$G = \frac{R_4}{R_1 + R_2 + R_3 + R_4} = \frac{1}{250} = 4 \cdot 10^{-3} \quad (\text{Ec. 13})$$

Con el cálculo de todos los componentes realizado, se obtiene el siguiente circuito de la figura 12:



**Figura 12:** Circuito final con valor de los componentes sin normalizar

Después de obtenerlos, se han normalizado a la serie estándar E96 que tiene una tolerancia del 1%, obteniendo así el resultado de la figura 13:

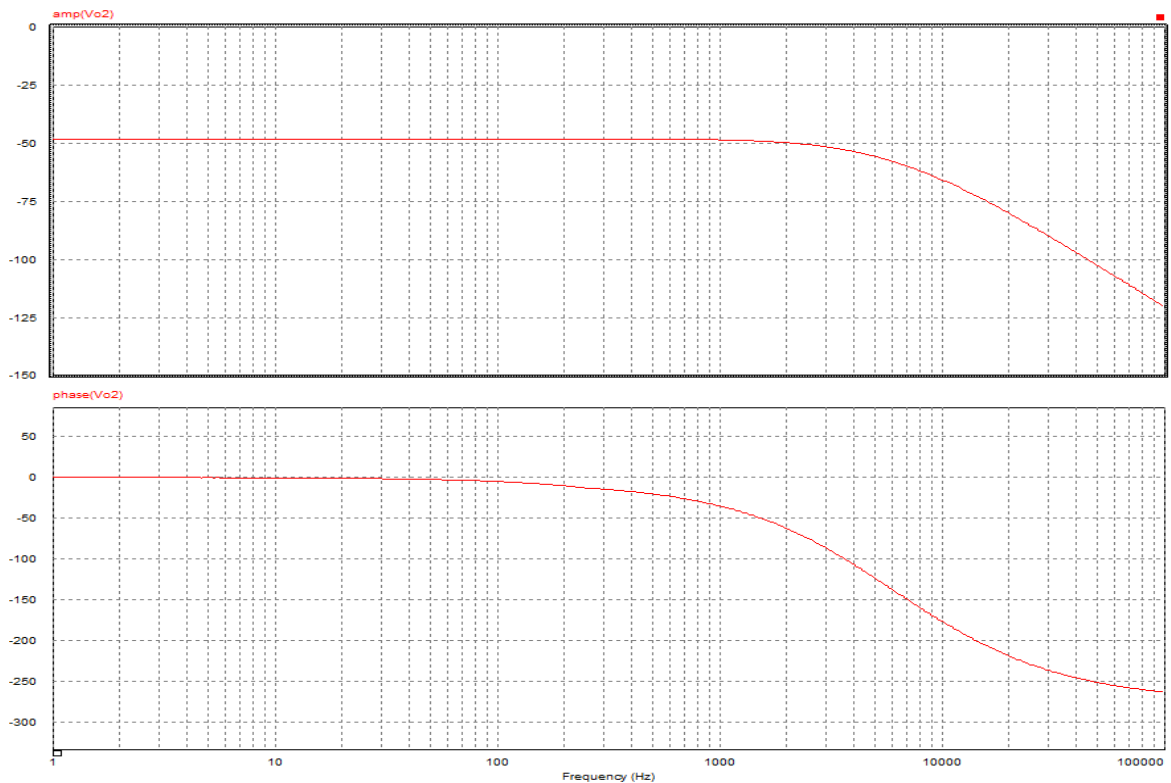


**Figura 13:** Circuito final del sensor de tensión.

Mediante simulaciones se ha obtenido la potencia que deben tener los resistores, y la tensión máxima que deben soportar los capacitores del circuito, para poder medir la máxima tensión posible. Mediante el análisis de una entrada de 412V a 50Hz y midiendo la caída de tensión y la corriente que circula por cada componente, y posteriormente se multiplican los valores eficaces de cada uno para obtener la potencia. Obteniendo así una potencia de 0,25 W para los resistores, y los condensadores C1, C2 y C3 pueden llegar a tener caídas de tensión entre sus bornes de 5 V, 15 V y 150 V respectivamente.

## 4.6. Simulaciones transductor de tensión.

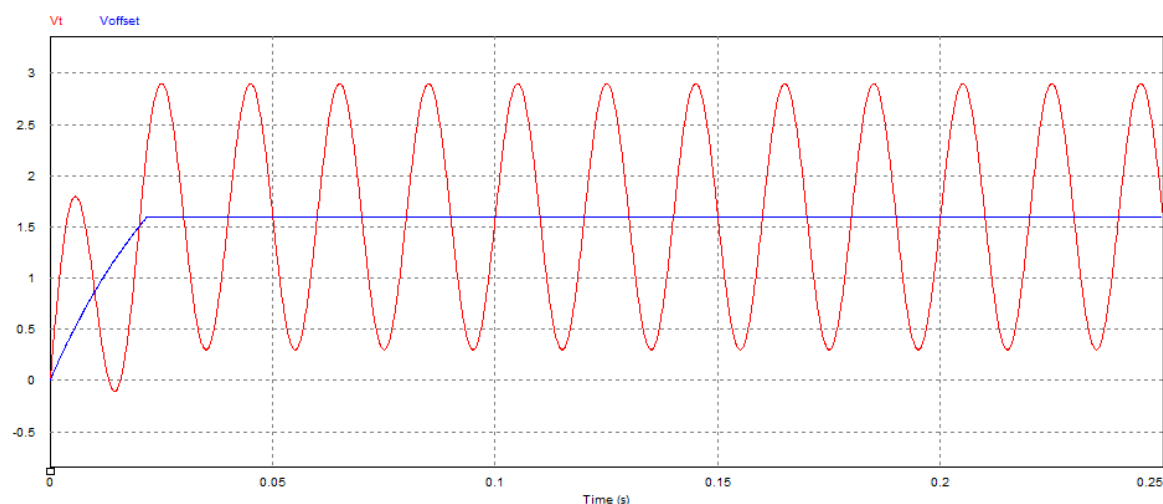
Con el circuito determinado, para comprobar su correcto funcionamiento se realizan simulaciones mediante la demo del software PSIM11.1.3. A partir del circuito de la figura 13 se ha realizado un barrido de frecuencias sobre la fuente de alimentación, de ahí se obtiene el siguiente bode (figura 14) de la señal  $V_t$ :



**Figura 14:** Bode de la señal  $V_t$ .

La ganancia en bajas frecuencia se mantiene linealmente en un valor de -47,9 dB, a la frecuencia de 500 Hz que tiene una ganancia de -48,06 y un desfase de 20,5 grados, a partir de esta frecuencia la ganancia empieza a decaer considerablemente, consiguiendo una caída de 3 dB respecto al valor inicial en la frecuencia de 2750 Hz. También podemos observar que la pendiente de la ganancia en la banda rechazada es de -60 dB/Dec. debido a que el filtro es de tercer orden.

Si se modeliza la fuente a analizar con una tensión de 230 Vac a 50 Hz, en  $V_t$  se obtiene la siguiente señal:



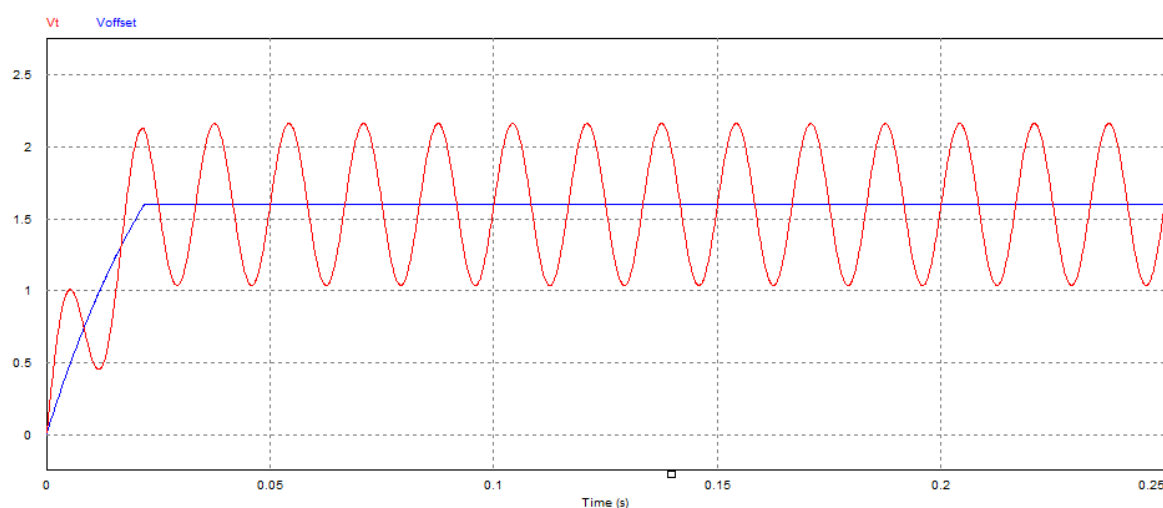
**Figura 15:** Lectura de  $V_t$  cuando la tensión a analizar es de 230 Vac a 50 Hz.

Después transitorio se obtienen los siguientes valores en la señal  $V_t$ :

**Tabla 3:** Valores de la señal  $V_t$  después del transitorio para una entrada de 230V a 50Hz.

$V_t$ mín.	$V_t$ medio	$V_t$ máx.	$V_t$ rms
0,302 V	1,6 V	2,898 V	2,518 V

Midiendo la tensión de 100 Vac a 60 Hz se obtiene la siguiente señal en  $V_t$ :



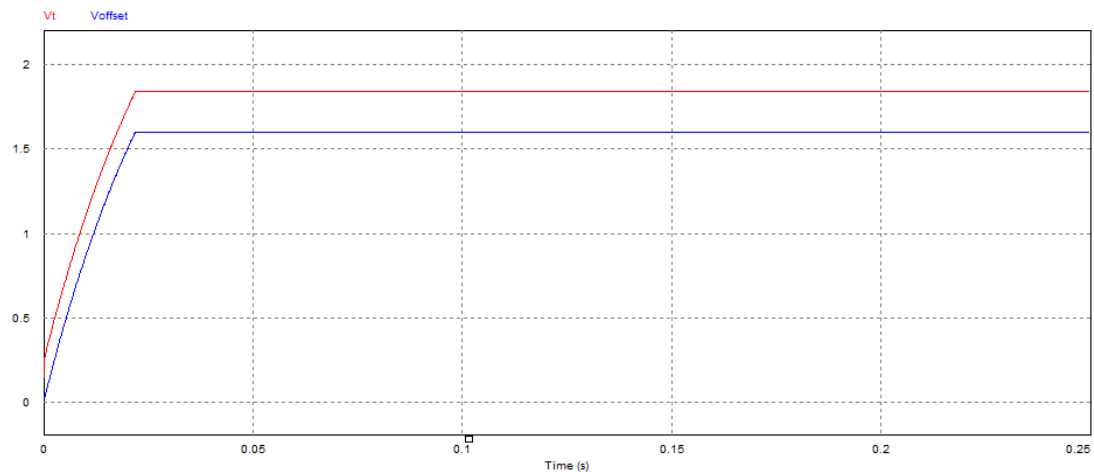
**Figura 16:** Lectura de  $V_t$  cuando la tensión a analizar es de 100 Vac a 50 Hz.

Con los siguientes valores de la señal estabilizada:

**Tabla 4:** Valores de la señal  $V_t$  después del transitorio para una entrada de 100 V a 50Hz.

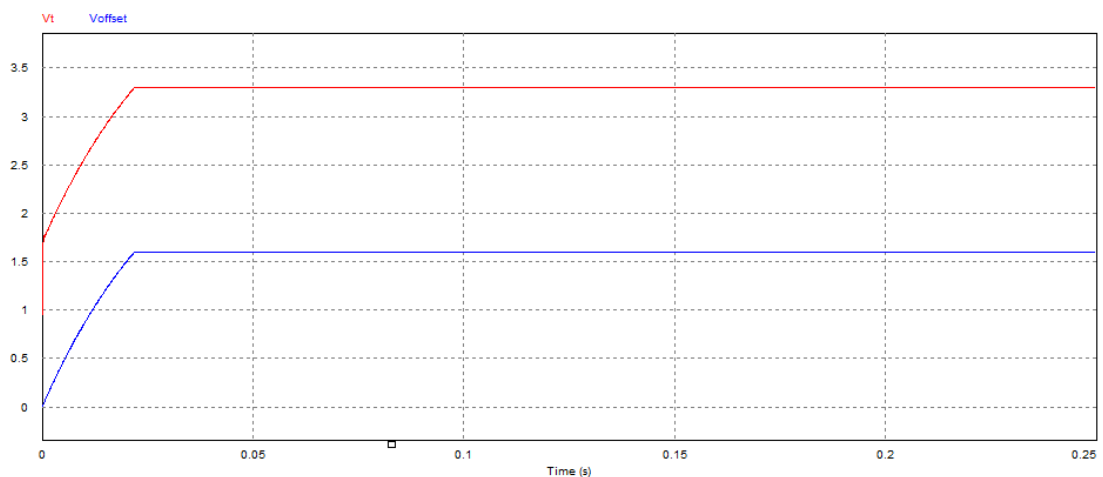
$V_t$ mín.	$V_t$ medio	$V_t$ máx.	$V_i$ rms
1,037 V	1,6 V	2,163 V	1,998 V

Midiendo una tensión continua de valor de 60 Vdc, se obtiene la siguiente salida:  $V_t=1,84$  V



**Figura 17:** Lectura de  $V_t$  cuando la tensión a analizar es de 60 Vdc.

Midiendo una tensión continua de valor de 430 Vdc, se obtiene la siguiente salida:  $V_t=3,3$  V.



**Figura 18:** Lectura de  $V_t$  cuando la tensión a analizar es de 430 Vdc.

Con la misma tensión, pero en negativo (-430 V), el valor de  $V_t$  después del transitorio es de -0,1 V. Esta tensión el ESP32 no podría medir, porque es negativa, pero tampoco es un riesgo para el módulo, ya que no supera los -0,3 V.

## 4.7. Transductor de corriente.

La intensidad que circula por una carga eléctrica se puede medir mediante un amperímetro conectado en serie con esta, para que la corriente atravesase dicho amperímetro. Generalmente los amperímetros utilizan una resistencia de *shunt* para medir la caída de tensión provocada por la corriente en ella, pero hay otros instrumentos como las pinzas amperimétricas, que se basan en inducción electromagnética para realizar la medida.

### Resistencia *shunt*

Generalmente la resistencia *shunt* es conocida por su precisión y es utilizada para determinar la intensidad de corriente eléctrica que fluye a través de esta carga, mediante la medición de la diferencia de voltaje a esta, utilizando la ley de Ohm ( $I = V/R$ ). Es un método intrusivo, pues se está agregando un resistor al circuito a analizar. Habitualmente los multímetros utilizan 2 o más fondos de escala para medir la corriente, por ello tienen 2 o más conexiones, sin contar con la entrada común, una se utiliza para corrientes del orden de miliamperios y otro para valores del orden de unidades de amperios, de esta manera utilizan 2 o más resistencias *shunt* para medir la intensidad y no perder la resolución.

Se debe tener especial cuidado a la hora de realizar medidas de intensidad utilizando este método, ya que se tiene que abrir el circuito para conectarlo, se puede entrar en contacto directo con el circuito y existiendo la posibilidad de recibir una descarga eléctrica.

### Transformadores de corriente

Transductor *Flux Gate*, este es el método que utilizan las pinzas amperimétricas para realizar medidas de corriente. El circuito equivalente de la pinza es un transformador que tiene una espira en el primario (cable donde circula la corriente a medir) y en el secundario que tiene un gran número de espiras (1000; 2000), donde se conecta una resistencia. Cuando circula corriente por el primario se genera un campo magnético en el núcleo de la pinza o



transformador, induciendo así una corriente equivalente en el secundario de menor valor, de acuerdo a la relación de espiras, y se mide la caída de tensión en la resistencia del circuito secundario. Recibe el nombre de pinza porque consta de un sensor, en forma de pinza, que se abre y abraza el cable cuya corriente queremos medir.

Este método de medición permite obviar el inconveniente de tener que abrir el circuito en el que se quiere medir la corriente para colocar un amperímetro clásico, por este motivo también es sumamente seguro para la persona que realiza la medida, ya que se evita un contacto directo con el circuito. También tiene la ventaja de no ser intrusivo, porque no se introduce ningún elemento en el circuito a analizar.

Después de valorar las ventajas y desventajas de cada método. Para este proyecto se utilizará un transformador de corriente, concretamente en forma de pinza flux. Con las opciones disponibles en el mercado se va a utilizar las pinzas SCT-013, como la que se muestra en la figura 19, del fabricante YHDC, en el anexo 4 se encuentra el *datasheet* de la serie SCT-013. Para un correcto uso de estas, hay que elegir un modelo de sonda dependiendo del rango de corrientes que se quiera medir, para no perder la resolución de la medida, por ejemplo, si se van a analizar cargas que consumen aproximadamente 3 A, es conveniente utilizar una pinza que permita medir hasta 5 A en RMS ya que se aprovecha bien el rango de valores que puede medir la pinza, y no conviene utilizar una pinza que pueda medir hasta 25 A ya que se perdería resolución. Para este proyecto se han adquirido 2 sondas: el modelo SCT-013-005 para medir corrientes de hasta 5 A; y el modelo SCT-013-025 para medir corrientes de hasta 25 A que es un valor ya más grande.

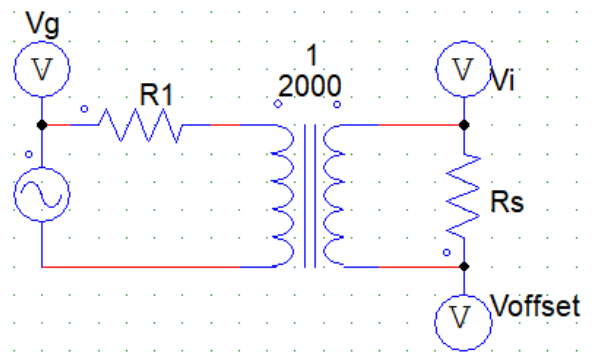


**Figura 19:** Pinza SCT-013 [17].

Independientemente del modelo utilizado, si se está midiendo la corriente máxima para el que está diseñada la pinza, en la salida entrega una tensión de 1 V,

Para la conexión al circuito del módulo dispone de un conector Jack 2,5 mm, este tiene 3 contactos; 2 conectados a cada extremo del bobinado del circuito secundario y un no conectado. Para poder conectarlo con el circuito, únicamente hay que introducir el conector en el *jack* hembra, sin necesidad de abrir ningún circuito.

En la figura 20 se muestra el circuito equivalente a una pinza amperimétrica:



**Figura 20:** Equivalente circuito para sensado de corriente.

Como se observa, hay una carga conecta a una fuente de energía eléctrica en serie a la carga ( $R1$ ) está conectado el transformador de corriente, este tiene una relación de espiras de  $1/2000$ . En el circuito secundario hay una resistencia ( $R_s$ ), en el siguiente apartado se muestra el procedimiento y los cálculos realizado para modelizar el circuito de la figura 20, y obtener el valor de la resistencia  $R_s$ , que lleva integrada la pinza. Esta tiene un valor de  $400\ \Omega$  para el modelo SCT-013-005 y  $80\ \Omega$  para el SCT-013-025. También se puede observar en el circuito está conectado a una tensión de offset de valor  $1,65\text{ V}$ , para evitar que la señal  $V_i$  tenga valores negativos.

Si se supera el valor de corriente para el que está diseñada una sonda, esta nunca entregara más de un  $1\text{ V}$  a la salida, ya que se satura el núcleo de ferrita de la pinza, tal que el module ESP32 leería  $2,8\text{ V}$  de pico a pico, dejando así un margen para las sobre corriente que pueda tener la carga.

## 4.8. Cálculo componentes del circuito para la medida corriente.

Se coloca una resistencia en las bobinas secundario porque no se puede dejar a circuito abierto, si esto se da, en el secundario circularía una corriente determinada por una resistencia infinita, provocando así una caída de tensión infinita entre los bornes del secundario del transformador, destruyendo este. Las sondas SCT-013 ya incorporan esta resistencia, para determinar su valor y hacer las correspondientes simulaciones se hacen los cálculos que se muestran a continuación.

A la circular corriente por el bobinado primero, se induce una corriente en el bobinado secundario, su valor se obtiene de la ecuación 15:

$$\frac{V_p}{V_s} = \frac{N_p}{N_s} = \frac{I_s}{I_p} \quad (\text{Ec. 14})$$

$$I_s = I_p \cdot \frac{N_p}{N_s} \quad (\text{Ec. 16})$$

Para determinar el valor de esta resistencia, se sustituye la ecuación 16, en la ley de Ohm, obteniendo la ecuación 17:

$$R_s = \frac{V_i}{I_s} = \frac{V_i}{I_p} \cdot \frac{N_s}{N_p} \quad (\text{Ec. 17})$$

Donde  $V_i$  es la caída de tensión entre el bobinado secundario. Utilizando una sonda que permite medir intensidades de hasta 5 A eficaces, en la salida nos entregaría una tensión de una tensión máxima de 1 V eficaz, por lo tanto, el valor de la resistencia se puede calcular mediante la ecuación 19:

$$I_p = 5A_{rms} \rightarrow V_i = 1V_{rms} \quad (\text{Ec. 18})$$

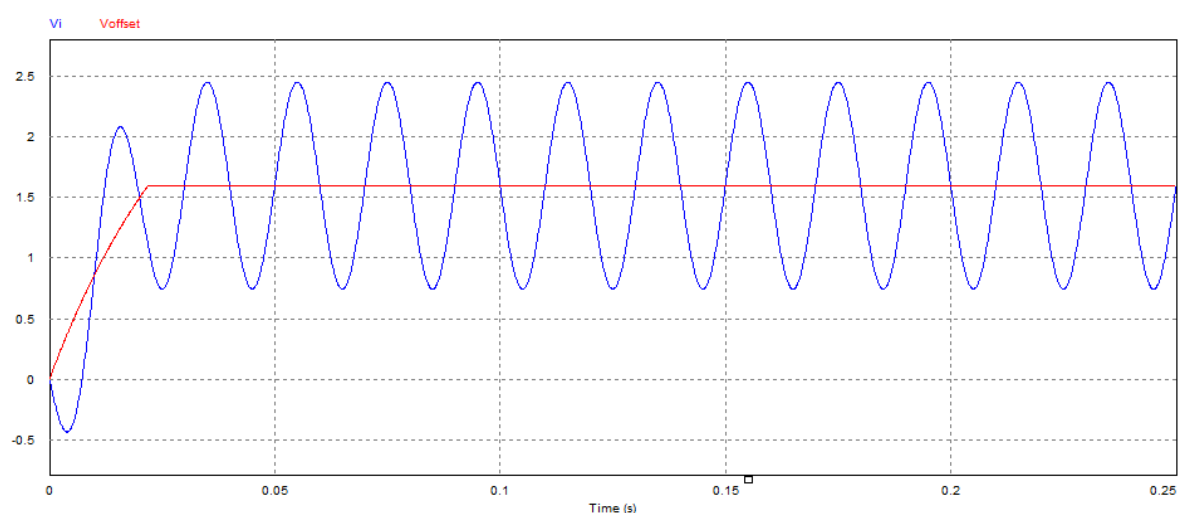
$$R_s = \frac{1}{5} \cdot \frac{2000}{1} = 400\Omega \quad (\text{Ec. 19})$$

El valor de la resistencia que incorporar internamente la sonda SCT-013-005 es de 400  $\Omega$ , para el modelo SCT-013-025 la resistencia interna es de 80  $\Omega$ .

## 4.9. Simulaciones transductor de corriente.

Para analizar el funcionamiento del circuito sensor de corriente de la figura 12, se realizan las simulaciones que se muestran a continuación.

La primera prueba se hace una medida de corriente de 3 A eficaces AC (50 Hz) con la sonda de 5 A, se obtiene lo siguiente:



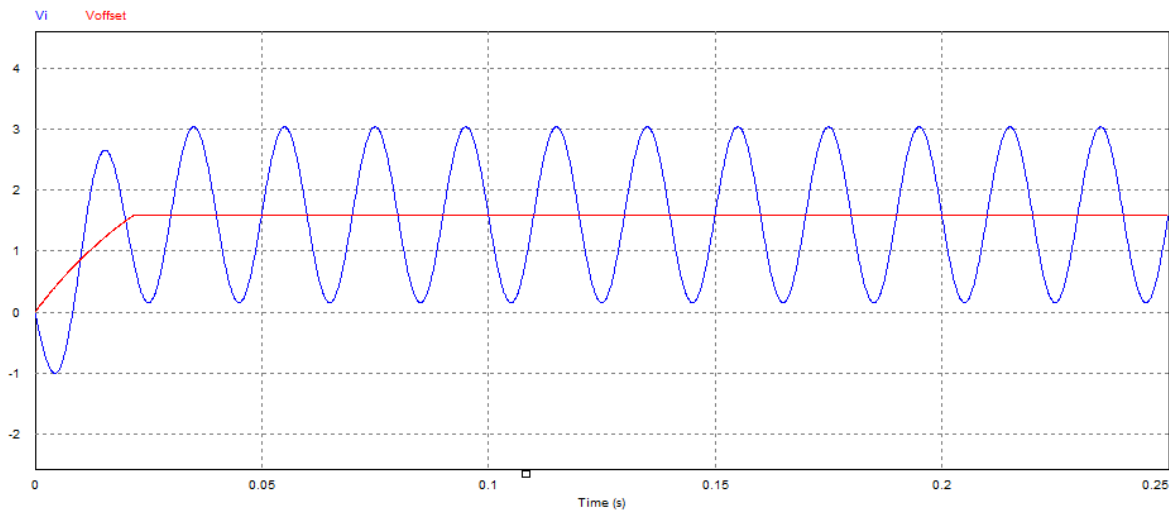
**Figura 21:** Lectura de  $V_i$  cuando la corriente a analizar es de 3 Aac a 50 Hz con SCT-013-005.

Después del transitorio, se obtienen los siguientes valores:

**Tabla 5:** Valores de la señal  $V_i$  después del transitorio para una entrada de 3 A a 50Hz.

$V_i$ mín.	$V_i$ medio	$V_i$ máx.	$V_i$ eficaz
0,745 V	1,6 V	2,45 V	2,205 V

Simulando una corriente de 5 A eficaces AC (50 Hz) con la sonda de 5 A se obtiene lo siguiente en la salida:



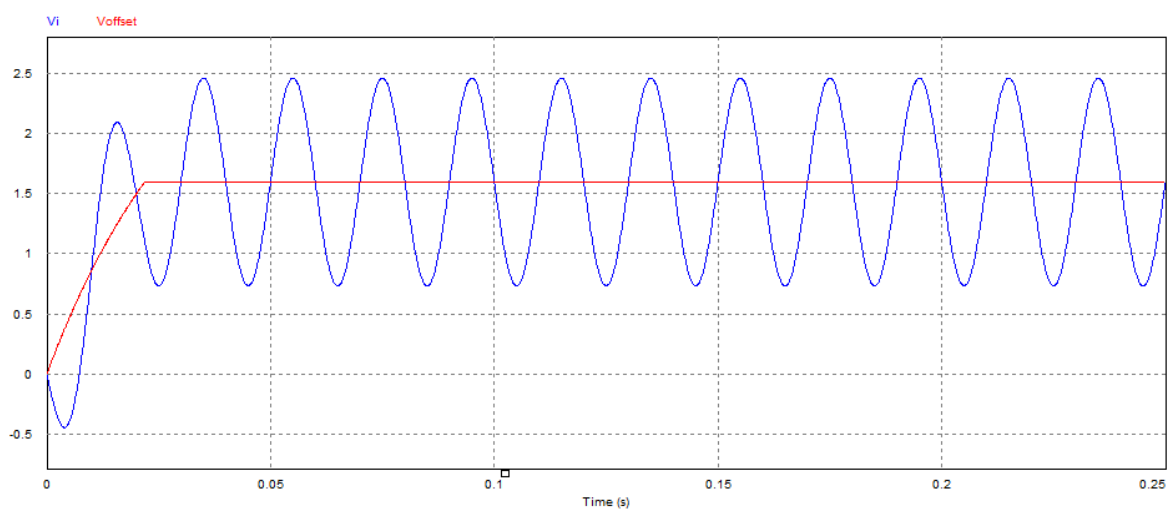
**Figura 22:** Lectura de  $V_i$  cuando la corriente a analizar es de 5 Aac a 50 Hz con SCT-013-005.

Cuando la señal  $V_i$  se estabiliza se obtienen los siguientes valores utilizando la sonda SCT-013-005:

**Tabla 6:** Valores de la señal  $V_i$  después del transitorio para una entrada de 5 A a 50Hz.

$V_i$ mín.	$V_i$ medio	$V_i$ máx.	$V_i$ eficaz (si Voff)
0,145 V	1,6 V	3,044 V	2,622 V

Medir una corriente de 15 A eficaces AC (50 Hz) con la pinza de 25 A, se obtiene la siguiente señal en la salida:



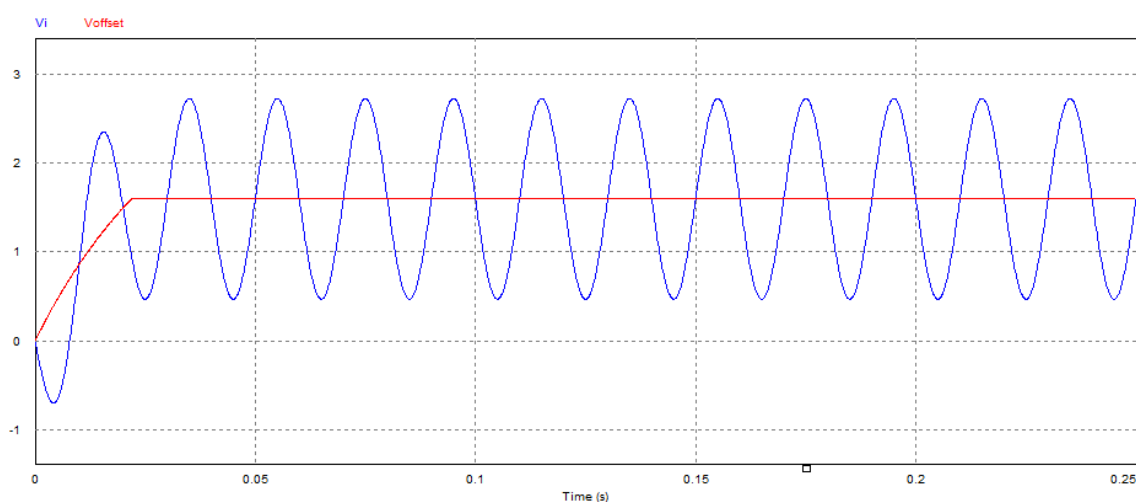
**Figura 23:** Lectura de  $V_i$  cuando la corriente a analizar es de 15 Aac a 50 Hz con SCT-013-025.

Cuando la señal  $V_i$  se estabiliza se obtienen los siguientes valores utilizando la sonda SCT-013-025:

**Tabla 7:** Valores de la señal  $V_i$  después del transitorio para una entrada de 15 A a 50Hz.

$V_i$ mín.	$V_i$ medio	$V_i$ máx.	$V_i$ eficaz
0,733 V	1,6 V	2,465 V	2,213 V

Medir una corriente de 25 A eficaces AC (50 Hz) con la son de 25 A se obtiene lo siguiente en la salida:



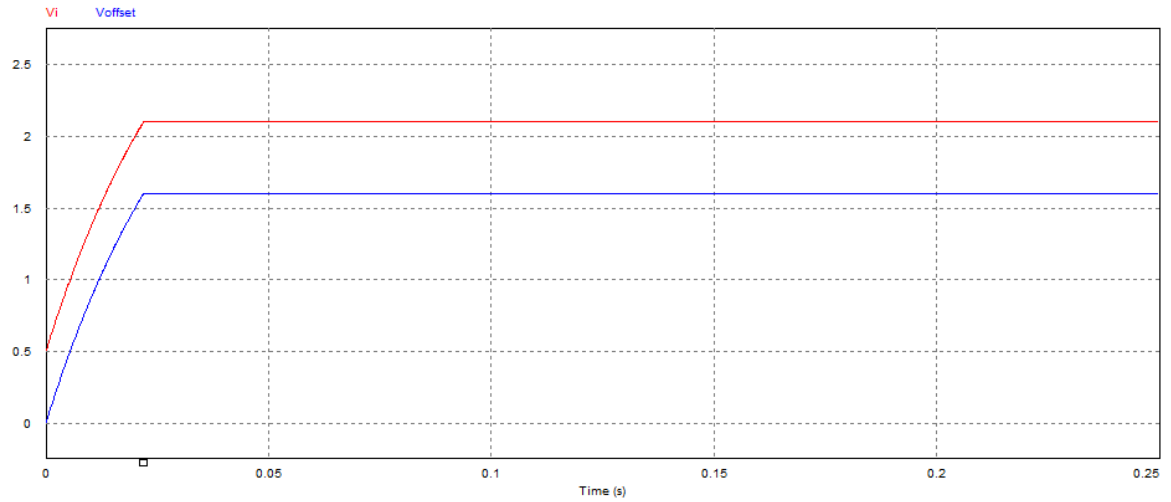
**Figura 24:** Lectura de  $V_i$  cuando la corriente a analizar es de 25 Aac a 50 Hz con SCT-013-025.

Después del transitorio, se obtienen los siguientes valores utilizando la sonda de SCT-013-025.

**Tabla 8:** Valores de la señal  $V_i$  después del transitorio para una entrada de 25 A a 50Hz.

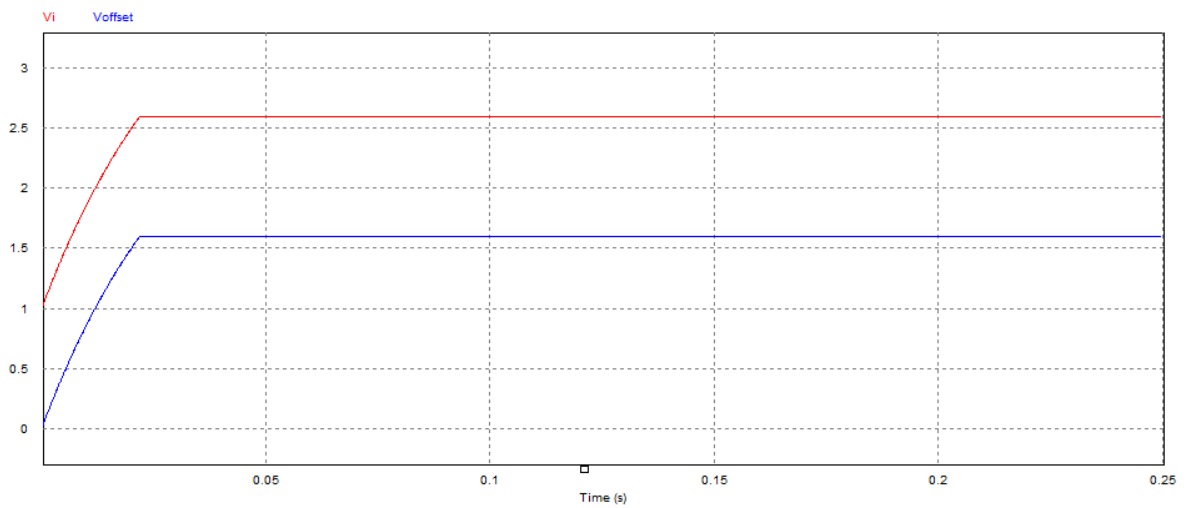
$V_i$ mín.	$V_i$ medio	$V_i$ máx.	$V_i$ eficaz
0,471 V	1,6 V	2,73 V	2,399 V

Medir una corriente de 2,5 A continua con la sonda de 5A se obtiene lo siguiente en la salida:  $V_i=2.1V$  como se muestra en la figura 25:



**Figura 25:** Lectura de  $V_i$  cuando la corriente a analizar es de 2,5 A con SCT-013-005.

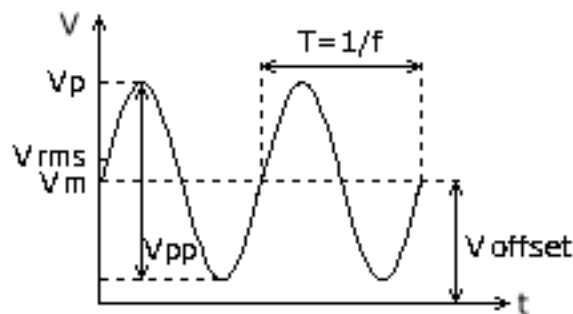
Simulando una corriente de entrada de 25 A continua, con la sonda de 25 A se obtiene la siguiente salida:  $V_i=2.6$  V



**Figura 26:** Lectura de  $V_i$  cuando la corriente a analizar es de 25 A con SCT-013-025.

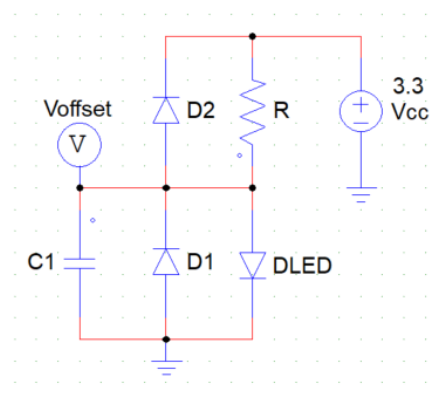
## 4.10. Circuito generador de Offset.

Como ya se ha comentado anteriormente, el microcontrolador ESP32-WROOM no puede trabajar con tensiones negativas, para eso debe añadir una tensión continua, también conocida como tensión de offset, a las señales que se van a medir, como se ve en la figura 27. Debe tener un valor de la mitad de la tensión de alimentación del MCU ESP32, tal que centraría la señal medida en valores positivos.



**Figura 27:** Señal alterna más una tensión de *offset* [18].

La manera más fácil y simple de obtener esta componente continua es emplear un divisor de tensión, con resistencias iguales, y añadiendo condensadores en serie con estas para filtrar los picos, que pueda tener la señal medida evacuar las corrientes de polarización. También se podría utilizar un amplificador operacional no inversor un con una ganancia de 0,5, pero este método ya es más complejo porque requiere alimentación. Para este diseño se utilizará el circuito de la figura 28, formado por una resistencia y un diodo LED (siglas del inglés: *Light-Emitting Diode*).



**Figura 28:** Circuito generador de voltaje para compensación de entrada y protecciones.



Para tener una tensión media de la tensión de alimentación, el diodo LED a utilizar debe ser uno rojo de bajo brillo, como se puede ver en la tabla, es el que tiene una caída de tensión entre sus bornes en directa de 1,6 V.

**Tabla 9:** Valores de tensión umbral de diodos LED según su color [19].

COLOR DEL LED	TENSIÓN UMBRAL
Rojo	1,6V
Rojo alta luminosidad	1,9V
Amarillo	1,7V a 2V
Verde	2,4V
Naranja	2,4V
Blanco brillante	3,4V
Azul	3,4V
Azul 430nm	4,6V

Para el cálculo de resistencia en serie con el diodo LED, se desprecian los diodos y el condensador que son elementos de seguridad, el valor de la resistencia limitadora mínima que se obtiene de la ecuación:

$$R = \frac{V_{cc} - V_{DLED}}{I_{DLEDm\acute{a}x}} = \frac{3,3 - 1,6}{20 \cdot 10^{-3}} = 85\Omega \quad (\text{Ec. 20})$$

Por lo tanto, esta es la resistencia mínima que hay que poner en serie con el diodo para que circule por la máxima corriente, no obstante, es mejor utilizar una resistencia más grande, así se fatigara menos el menos el LED, así brilla y consume menos, también se calienta menos y tiene menos derivas térmicas por la variación de la tensión en directa. Para este diseño se utilizará una resistencia de 1 kΩ. Paralelamente a la resistencia se colocará un diodo *schottky*, y con diodo LED rojo otro diodo *schottky* en inversa y un condensador. El condensador produce que se tenga una arrancada suave de la señal y elimina los picos de corriente que se genera al conectar una carga a una fuente de alimentación. Los diodos sirven para evacuar corriente que provengan de la sonda y no derivarla a la placa ESP.

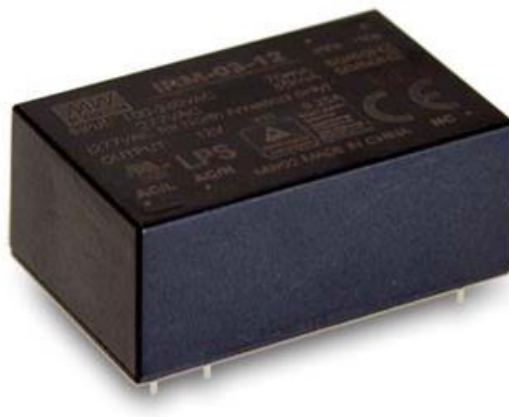
## 4.11. Alimentación.

Para alimentar el conjunto del circuito y el ESP32, se ha recurrido a una fuente de alimentación pequeña para placas de topes, que entrega 3,3 V en DC a la salida. El elemento que más va a consumir es el microcontrolador, trabajando a máximo rendimiento va a necesitar una potencia de 743 mW, y sumada la potencia disipada por el LED de 6,6 mW, a partir de la suma de los resultados de las ecuaciones 21 y 21, se obtiene que va a hacer falta una fuente de 750 mW.

$$P_{uCmáx} = I_{uCmáx} \cdot V_{CC} = 0,225 \cdot 3,3 = 0,743 \text{ W} \quad (\text{Ec. 21})$$

$$P_{LED} = I_{LED} \cdot V_{LED} = 0,002 \cdot 3,3 = 0,007 \text{ W} \quad (\text{Ec. 22})$$

Mirando un valor estándar, una fuente de 1 W es suficiente para alimentar el conjunto. Por falta de *Stock* en la tienda, se ha adquirido una fuente de 3 W, concretamente el modelo IRM-03-3.3 (figura 29) del fabricante Mean Well, en el anexo 4: Especificaciones técnicas fuente de alimentación IRM-03-3.3, se encuentran los datos técnicos del modelo.



**Figura 29:** Fuente de alimentación IRM-03 [20].

## 4.12. Diseño Final.

El diseño final de los circuitos se ha realizado en una placa de prototipos, colocando estratégicamente cada parte para tenga el menor tamaño posible. El diseño final del DAQ tiene la forma de la figura 30, y la parte posterior se puede ver en la figura 31.

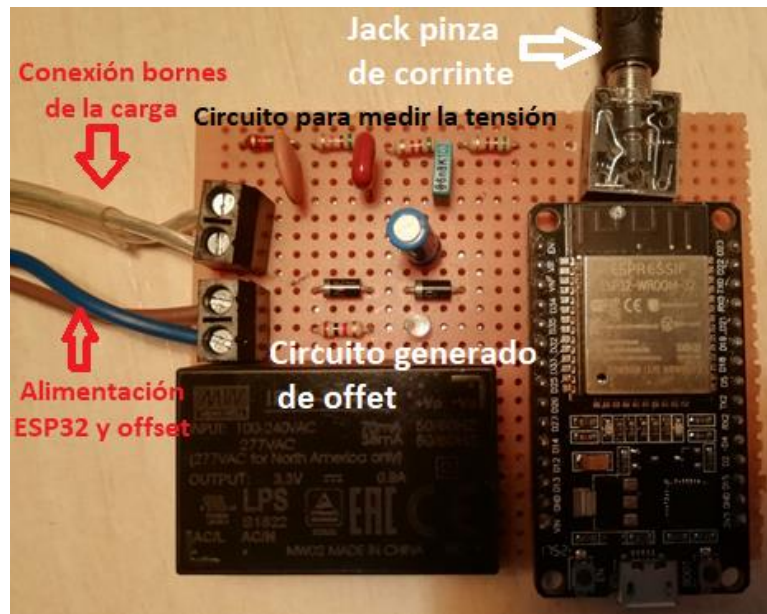


Figura 30: Partes del diseño final del DAQ.

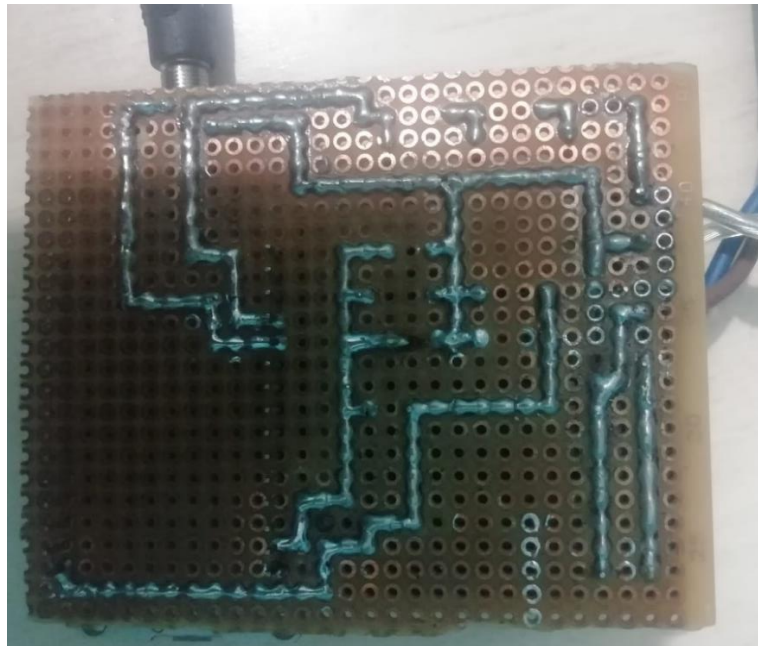


Figura 31: Vista posterior del diseño final del DAQ.

Dependiendo de la alimentación de la carga que se pretende monitorizar, si esta es alimentada con la tensión de red, los conectores de la alimentación de la PCB (enchufe negro) y de los que se utilizan para medir la tensión de la carga (enchufe gris) se pueden juntar en el demultiplexor de enchufes, como el de la figura 32.



**Figura 32:** Sistema de adquisición.

## 5. Software y comunicaciones.

### 5.1. Canal de comunicaciones.

Después de haber realizado el diseño del circuito encargado de transformar y acondicionar las señales para que las pueda leer el equipo de adquisición, toca estudiar el canal y el protocolo de comunicaciones, por el que se realiza la transferencia de datos desde el DAQ al PC, donde se muestran los datos.

Se han descartado las comunicaciones serie por cable USB, debido a su falta de seguridad, ya que, si el equipo encargado del muestreo sufre un fallo eléctrico y se daña, el PC al que está conectado por cable podría sufrir el mismo problema, pudiendo llegar a producir daños costosos en el equipo. El problema se podría solucionar con un aislante de puerto USB, pero son más caros que una placa ESP32, por lo que no es viable, aunque si se usan las comunicaciones alámbricas haría que prescindir del microcontrolador receptor conectado en el PC. Por ello es mejor utilizar comunicaciones no cableadas.

Las comunicaciones inalámbricas habitualmente se realizan por radiofrecuencia, los dispositivos que tienen este sistema no tienen por qué estar alineados, por lo tanto, por este medio pueden comunicarse entre sí cuando se encuentren dentro su alcance, dando también la ventaja de que se pueden realizar la transmisión de datos aunque estén en habitaciones separadas si tiene suficiente potencia de transmisión.

En el caso del microcontrolador ESP32 dispone de Wifi 802.11 b/g/n que trabaja en la banda de los 2,4Ghz y soporta los modos de comunicaciones WFA, WPA, WPA2 y WAPI. Admite velocidades de transferencia de datos de hasta 150 Mbps, utiliza un amplificador para comunicaciones inalámbrica de alcance máximo para tener una ganancia de potencia de salida de 22 dBm.

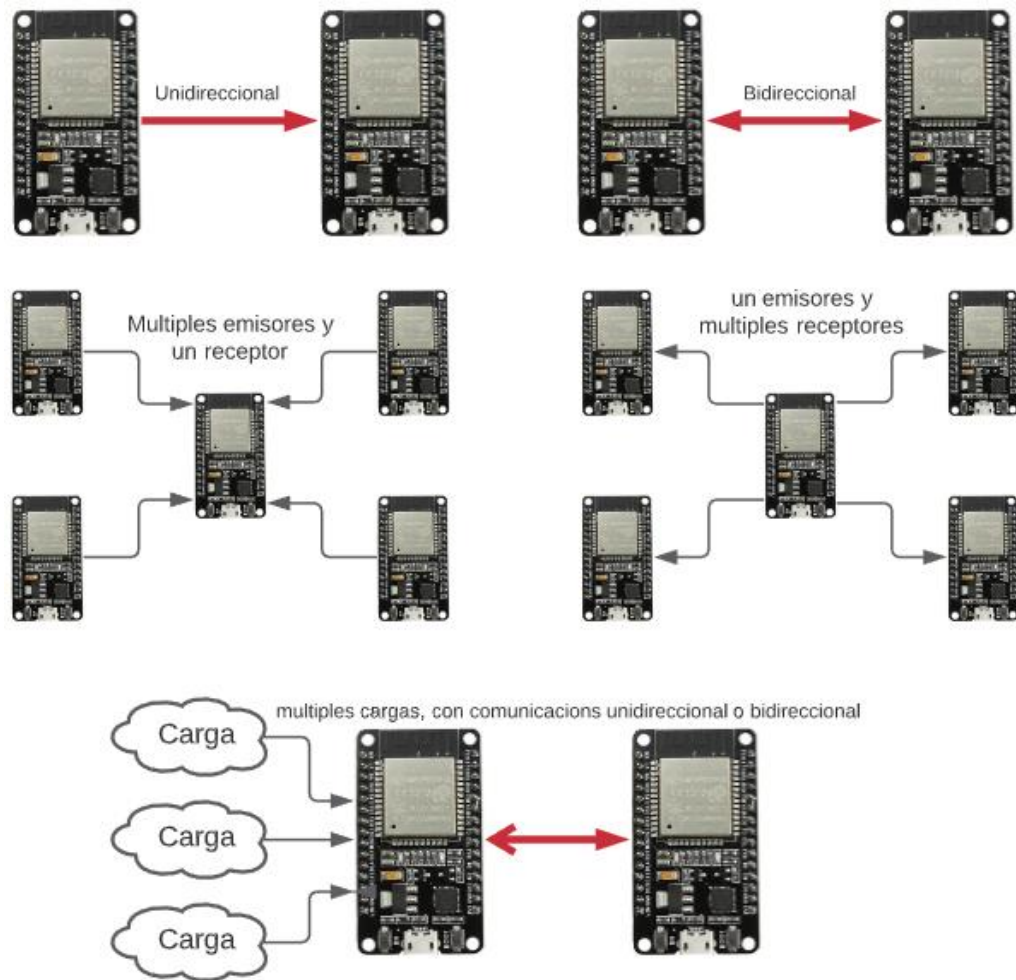
También tiene bluetooth, la versión v4.2 BR/EDR y bluetooth de bajo consumo energético (BLE del inglés: *Bluetooth Low Energy*), utiliza la banda de 2,4 GHz, esta versión es capaz de transmitir datos a una velocidad desde 25 Mbps a 32 Mbps. El bajo consumo eléctrico hace que sea de baja potencia 2,5 mW por ello tiene un alcance máximo de 10 m.

Viendo las características de cada medio, se ha escogido la comunicación inalámbrica por wifi, ya que daría más libertad para ubicar el equipo de visualización más lejos del módulo de adquisición, también tiene la ventaja de que soporta mayor velocidad de envío de datos, aun que tiene la desventaja de consumir más.

## 5.2. Protocolo de comunicaciones.

La versión de wifi que incorpora el microcontrolador ESP32 permite realizar intercambio de datos entre diferentes placas mediante diversos protocolos, como se observa en la figura 30. Tales como conexiones unidireccionales y bidireccionales seguras, con un bajo consumo de energía, largo alcance de comunicación, bajas velocidades de datos y de baja frecuencia de transmisión. Permite la interconexión entre objetos inteligentes sin la necesidad de instalaciones locales complejas, y además le permite al usuario tener una amplia libertad de uso, también da facilidades para el desarrollador y a las empresas que quieran instalar su propia red para Internet de las Cosas.

Para simplificar el proyecto solo se va a realizar el estudio del consumo de una carga, utilizando así únicamente 2 placas ESP32 (1 para adquirir los datos y enviarlos por wifi a otro módulo conectado al PC). Se podría extender a estudiar más de una carga (figura 33), pudiendo adquirir el consumo de diferentes cargas mediante diferentes microcontroladores ESP32, y enviar los datos a un módulo conectado a un ordenador. También se puede hacer a la inversa, mostrando los datos de una carga en diferentes ordenadores. Sin embargo, la manera más fácil es muestrear los datos de diferentes cargas eléctricas mediante un solo módulo ESP32 y replicando los circuitos de transducción de tensión y corriente diseñados anteriormente, y utilizar de visualizador un único ordenador.



**Figura 33:** Esquemas de comunicaciones [22].

A continuación, se presentan los métodos estudiados para realizar la transferencia de datos utilizados en este proyecto:

### 5.2.1. ESP-NOW.

Es un protocolo de comunicación para transferir datos entre placas ESP32 de forma inalámbrica, desarrollado por el fabricante de los mismos módulos *Espressif Systems*.

La conexión entre MCU es persistente, es decir, que cuando se apaga o reinicia un dispositivo por cualquier motivo, al volver a encenderse se conectan automáticamente. Este método permite realizar intercambio de paquetes cortos (250 Bytes) de información entre dispositivos de forma sencilla. Permite encriptar los datos enviados, para evitar que alguien no deseado



los utilice. Por último, permite configurar una llamada de respuesta para confirmar o denegar si el dispositivo receptor ha recibido los datos. Como máximo admite crear una red de comunicaciones entre 10 pares cifrados.

En este caso si se desea enviar los datos adquiridos por un MCU ESP32 a otro, se puede realizar mediante una canal unidireccional, como se puede observar en la figura 34. Esta configuración es muy sencilla y fácil de implementar, ideal para enviar datos de sensores, estado de interruptores, órdenes para controlar los GPIO. Para Enviar datos el emisor únicamente necesitas saber la dirección MAC del receptor, y también tener definida la misma estructura de datos en ambos MCU.



**Figura 34:** Estructura de comunicaciones unidireccional mediante ESP-NOW [22].

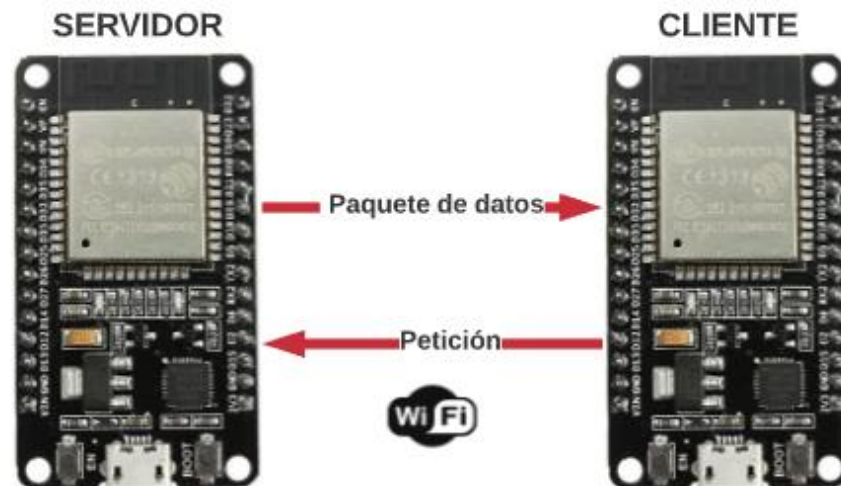
### 5.2.2. Protocolo de transferencia de hipertexto (HTTP).

Mediante un sistema Servidor-Cliente se puede realizar envío de datos entre MCU, utilizando solicitudes HTTP (siglas del inglés: *Hypertext Transfer Protocol*). Permite intercambiar datos a través de wifi sin conexiones a internet, sin la necesidad de utilizar un enrutador. Este método utiliza una placa que actúa como cliente y otra como servidor. El protocolo de comunicaciones por transferencia de Hipertexto, permite realizar transferencia de información en la *World Wide Web*, y no guarda la información intercambiada en operaciones anteriores.

Para crear el sistema, la placa que actúa como servidor genera su propia red inalámbrica (punto de acceso, AP), otros dispositivos (clientes) se pueden conectar a esta, introduciendo el SSID (siglas del inglés: *Service Set Identifier*) y la contraseña del punto de acceso. Para que un cliente reciba datos, le tiene que enviar una solicitud determinada al servidor, indicando



la dirección IP del servidor seguida de la información que desea, el servidor recibe las solicitudes entrantes y envía la respuesta correspondiente a través de HTTP (figura 35). Por lo tanto, la comunicación entre los equipos es bidireccional.



**Figura 35:** Esquema de comunicaciones mediante HTTP [23].

Después de haber valorado y probado los métodos, ambos funcionan correctamente, y cumplen con la función de enviar datos entre dispositivos de forma inalámbrica, resaltando que la velocidad de envío y el tamaño de paquetes del protocolo ESP-NOW es más crítico, puesto que hay que enviar paquetes de muestras más pequeños, si no se pierden datos. En los anexos A4: Código del ESP32 emisor utilizando protocolo ESP-NOW y A5: Código del ESP32 receptor utilizando protocolo ESP-NOW, se presenta el código para el emisor y el receptor utilizando este método. En el método de transferencia de hipertexto es el que se ha utilizado para las pruebas, más adelante se muestran los programas utilizados para el servidor y el cliente (apartado 5.5 y 5.6).

### 5.3. El entorno de desarrollo y programación del MCU.

El entorno de desarrollo integrado, IDE (siglas del inglés: *Integrated Development Environment*) es cualquier programa informático que permita a un desarrollador programar y editar un código. Consta de herramientas como depuradores, herramientas de construcción automáticas y editor de código.

Los microcontroladores de *Espressif Systems* se puede programar de mediante diferente software, entre estos se encuentran los siguientes: *Arduino IDE*, *PlatformIO*, *Visual Studio Code*, *Thonny IDE*, *Python*, etc.

Se utilizará *Arduino IDE*, por los conocimientos previos y la experiencia obtenida en otras asignaturas del grado, también tiene la gran ventaja de que muchas tareas se pueden simplificar utilizando librerías. Al ser un programa conocido y muy utilizado también es fácil encontrar información de interés para realizar determinadas funciones. Es un software programado en java, preparado para ser utilizado en diferentes sistemas operativos. Este se utiliza para escribir, editar y cargar programas en la memoria de placas compatibles con *Arduino*, por lo que consiste de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Con los drivers y ayuda de otros desarrolladores, se puede utilizar *Arduino IDE* para programar módulos de otros fabricantes que no sean *Arduino*.

Técnicamente el programa informático es de libre disposición, está publicado bajo la licencia publica general de GNU, actualmente tiene la versión 1.8.13. Los códigos creados en el entorno de desarrollo de *Arduino* se escriben bajo las reglas y estructuras especiales de los lenguajes de programación C y C++. Los códigos escritos por este *software* solo requieren 2 funciones básicas y esenciales; la función *setup()* donde se inicia el estado de los periféricos de la placa; y la función *loop()*, donde se escribe el código cíclico del programa. También cabe destacar que el programa admite la utilización de librerías para facilitar la descripción de las tareas. Los programas escritos en el *Arduino IDE* generan un solo fichero con extensión “.ino” que se guarda en una carpeta con el mismo nombre que el archivo.

Ahora bien, para diseñar los códigos en *Arduino IDE* para los microcontroladores de *ESP32* no basta con solo instalar la última versión este, ya que el nombre de los módulos *ESP* no aparece en el apartado de gestor de placas en la barra de herramientas del programa. Para ello hay que descargar también las últimas versiones del código emulador *Arduino* para *ESP32*, para eso es necesario instalar el entorno gráfico *GIT*. En el anexo 3: se explica el proceso de instalación y la configuración del entorno.

## 5.4. Configuración Arduino IDE.

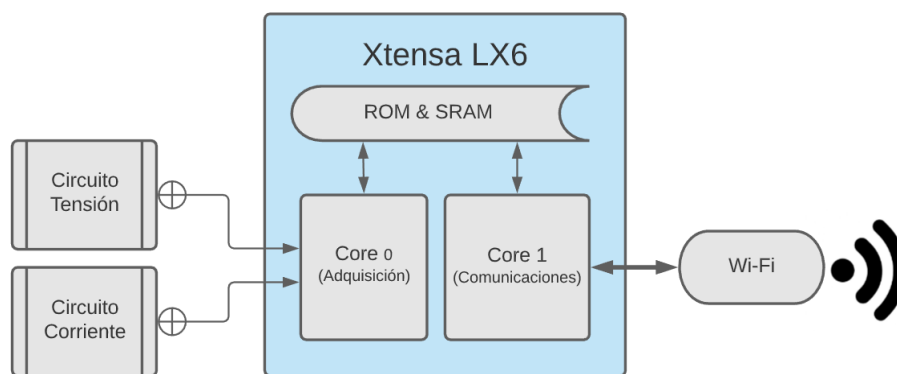
Un paso previo a la programación de los chips, consiste en determinar las características con las que va a trabajar el MCU que se pretende programar, indicando el modelo de la placa en el apartado de herramientas del Arduino IDE, y sus correspondientes características, que se pueden ver en la figura 36:

```
Placa: "ESP32 Dev Module"  
Upload Speed: "115200"  
CPU Frequency: "240MHz (WiFi/BT)"  
Flash Frequency: "80MHz"  
Flash Mode: "QIO"  
Flash Size: "16MB (128Mb)"  
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"  
Core Debug Level: "Cap"  
PSRAM: "Disabled"
```

**Figura 36:** Configuración ESP32.

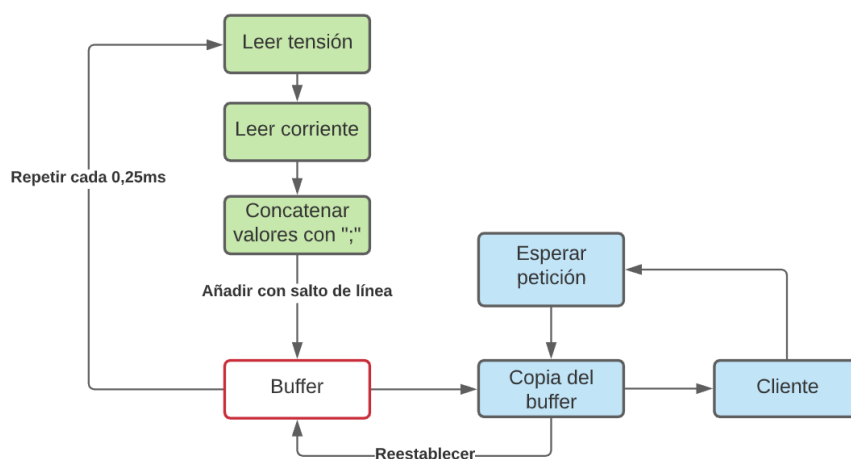
## 5.5. Controlador servidor.

El módulo que adquiere las muestras de los circuitos de corriente y tensión es el que va actuar como servidor, su función consiste en obtener medidas de corriente y tensión cada cierto intervalo de tiempo, y lo guarda en un *buffer*, cuando reciba una petición del cliente enviarle los datos y vaciarlo. Para aprovechar los núcleos que tiene el ESP32, se dedica el principal para escuchar a los clientes y enviar los datos, por tanto, encargarse de las comunicaciones, y el otro para leer y guardar los datos como se observa en la figura 37. Evitando así interrumpir la adquisición para enviar los datos, puesto que, si solo se trabajase con un núcleo para hacer ambas tareas, cuando el ESP32 tenga que enviar datos, interrumpe la lectura, pudiendo perderse información muy valiosa.



**Figura 37:** Esquema de datos del ESP32

En la figura 38 se muestra el diagrama de flujo correspondiente al programa implementado en el controlador servidor, las acciones de color verde son las que se van a ejecutar en núcleo 0, y las de color azul en el núcleo 1, entre ellos los interconecta la memoria que es donde se guarda el buffer que se rellena con las muestras.



**Figura 38:** Diagrama de flujo del controlador servidor.

A continuación, se presenta el código utilizado para programar la placa ESP32 y que realice las funciones determinadas:

## Servidor3

```

#include "WiFi.h"
#include "ESPAsyncWebServer.h"

const char* ssid = "ESP32-Access-Point";
const char* password = "123456789";

int Voltimetro = 36;
int Amperimetro = 34;
String V;
String I;
String VI="";
String cVI="";

TaskHandle_t Task1;
AsyncWebServer server(80);

void setup()
{
    Serial.begin(115200);

    xTaskCreatePinnedToCore(
        loop2,
        "Task_1",
        1000,
        NULL,
        1,
        &Task1,
        0);

    WiFi.softAP(ssid, password);
    IPAddress IP = WiFi.softAPIP();
    Serial.print("Dirección IP: ");
    Serial.println(IP);

    server.on("/VI", HTTP_GET, [] (AsyncWebServerRequest *request){
        request->send_P(200, "text/plain", readVI().c_str());
    });
    server.begin();
}

void String readVI()
{
    Serial.print(VI);
    cVI= VI;
    VI="";
    return cVI;
}

void loop() {}

```

```
void loop2(void *parameter)
{
    for(;;)
    {
        V =analogRead(Voltimetro)+10000;
        I =analogRead(Amperimetro)+10000;
        VI = (VI + String(V) + ";" + String(I)+"\n");
        delay(0.25);
    }
}
```

El programa funciona de la siguiente forma:

Primeramente, hay que descarga las librerías necesarias para simplificar la programación del ESP32, e incluirlas en el IDE.

```
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
```

El siguiente paso es definir el punto de acceso de la red wifi, determinando las credenciales necesarias para acceder a dicha red.

```
const char* ssid = "ESP32-Access-Point";
const char* password = "123456789";
```

Seguido de la definición de todas las variables a utilizar, de las cuales Voltímetro y Amperímetro guardan los valores de lectora de los pines 36 y 34, el ADC proporciona valores de 12 bits por lo tanto pueden tomar valores de 0 a 4095. Las variables "V" y "I" corresponden a la conversión en texto de las 2 anteriores. Por último, el *String* "VI" que corresponde al paquete de datos que se envía al otro módulo, y su correspondiente copia "cVI". Es importante que sean variable principales y no locales, por eso se han definido fuera de las funciones, ya que así se puede acceder a ellas y editarlas desde cualquier función

```
int Voltimetro = 36;
int Amperimetro = 34;
String V;
String I;
String VI="";
String cVI="";
```

Seguidamente se crea el nombre de la tarea a ejecutar en el núcleo 0, y el servidor web asíncrono en el puerto 80.

```
TaskHandle_t Task1;  
AsyncWebServer server(80);
```

La función *setup()* es la primera que se ejecuta en el programa, y se utiliza para configurar, inicializar variables, como usar las librerías, etc. En definitiva, se utiliza para el control activo de la placa. Primeramente, en esta se inicia el monitor serial a 115200 Baudios.

```
Serial.begin(115200);
```

Aquí también se crea el segundo lazo, este se dedica a leer los valores de los pines 34 y 36. Para definir el lazo, hay que indicar el nombre de la función, identificar el nombre de la tarea creada anteriormente, indicar el tamaño de la pila, poner “*NULL*” si no se le pasan valores a este segundo lazo, prioridad, nombre de la tarea y el núcleo donde se va a ejecutar, haciendo un inciso en que el *setup()* y el *loop()* se ejecutan en el núcleo 1, y el *loop2()* en el núcleo 0.

```
xTaskCreatePinnedToCore(  
    loop2,  
    "Task_1",  
    1000,  
    NULL,  
    1,  
    &Task1,  
    0);
```

Después se define el punto de acceso en el ESP32, indicando las credenciales creadas anteriormente. También se busca la dirección IP del servidor y la muestra por pantalla, ya que es necesaria para que un cliente le envíe peticiones.

```
WiFi.softAP(ssid, password);  
IPAddress IP = WiFi.softAPIP();  
Serial.print("Dirección IP: ");  
Serial.println(IP);
```

En esta parte del código se activa el modo de escucha del servidor, esperando solicitudes del cliente. Cuando un cliente le envía un dirección URL terminada en “/VI”, el MCU ejecuta la función *readVI()*, y le envía una respuesta con el paquete de datos. Posteriormente se inicia el servidor.

```

server.on("/VI", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", readVI().c_str());
});
server.begin();
}

```

Cuando se reciba una solicitud, el núcleo 1 ejecuta la siguiente función. Muestra los datos de la variable “VI” y los copia en “cVI”, posteriormente vacía el *buffer* “VI”, ya que se han copiado los datos, y *core* 0, lo puede volver a rellenar. Por último, la función devuelve “cVI”, que se envía al cliente. El *loop()* está vacío, ya que en el núcleo 1 no se requieren tareas cíclicas, porque el sistema es asíncrono.

```

void String readVI()
{
    Serial.print(VI);
    cVI= VI;
    VI="";
    return cVI;
}

void loop(){}

```

Finalmente, en el núcleo 0, se ejecuta el *loop2()*, no requiere parámetros de otras funciones. En este se define un bucle *for* infinito, pues se van a hacer lecturas cíclicamente. En este se leen los valores de los pines 34 y 36, y se les suma 10000, esto facilitará separar las variables cuando estén en el ordenador, ya que cada variable siempre va a tener 5 cifras significativas. Por cada iteración del bucle, al anterior valor de “VI” se le añade las muestras de “V” y “I” separados por “;” con salto de fila después de cada muestra. Se hacen lecturas cada 0,25 ms, obteniendo un total de 80 muestras por ciclo en una señal eléctrica de 50 Hz, suficiente para su posterior representación, con 80 muestras por periodo se obtiene una reconstrucción de la señal aceptable, con una buena definición.

```

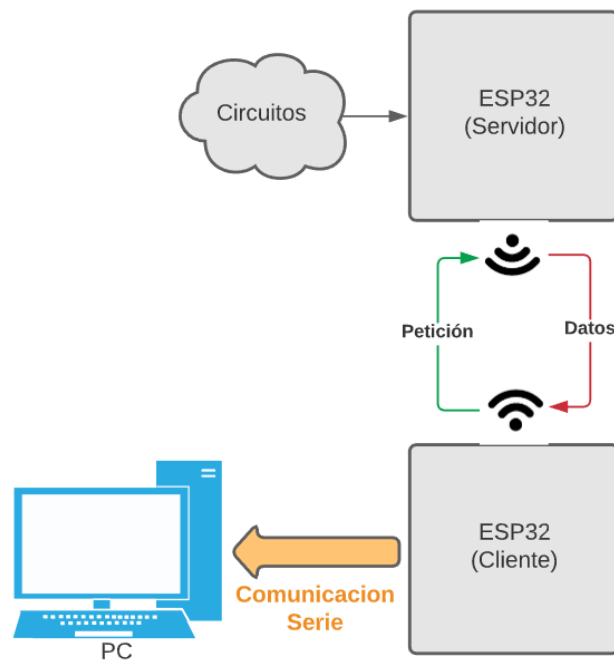
void loop2(void *parameter)
{
    for(;;)
    {
        V =analogRead(Voltimetro)+10000;
        I =analogRead(Amperimetro)+10000;
        VI = (VI + String(V) + ";" + String(I)+"\n");
        delay(0.25);
    }
}

```



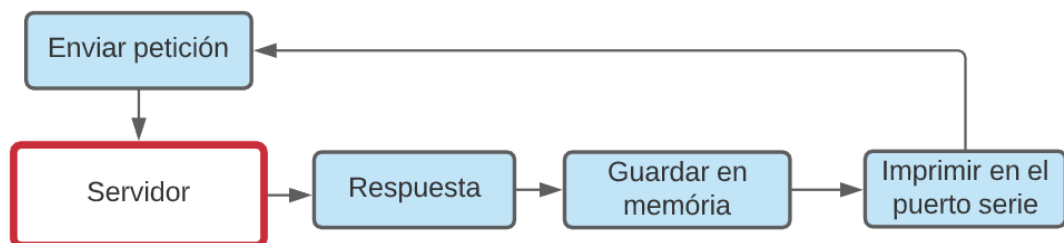
## 5.6. Controlador cliente.

La función de esta segunda placa consiste en enviar solicitudes cada intervalo de tiempo y esperar a que el servidor le envíe los datos (figura 39), cuando reciba los datos enviarlos por el puerto serie al que está conectado al PC.



**Figura 39:** Flujo de datos del sistema.

En el diagrama del flujo que se muestra en la figura 40, se muestran las acciones que realiza el cliente, su función consiste en enviar peticiones al servidor, y la información recibida mostrarla por el puerto serie.



**Figura 40:** Diagrama de flujo del controlador cliente.

Después de describir el funcionamiento y las tareas que debe realizar, a continuación, se presenta el código de Arduino IDE utilizado para programar el ESP32.

#### Receptor

```
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "ESP32-Access-Point";
const char* password = "123456789";

const char* serverNameVI = "http://192.168.4.1/VI";
String VI;
unsigned long anteriorMilis = 0;
const long intervalo = 200;

void setup()
{
    Serial.begin(115200);

    WiFi.begin(ssid, password);
    Serial.println("Conectando");
    while(WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("Wifi conectado a la direccion IP: ");
    Serial.println(WiFi.localIP());
}

void loop()
{
    unsigned long actualMilis = millis();

    if(actualMilis - anteriorMilis >= intervalo)
    {
        if(WiFi.status() == WL_CONNECTED )
        {
            VI = httpGETRequest(serverNameVI);
            Serial.print(VI);
            anteriorMilis = actualMilis;
        }
        else
        {
            Serial.println("Wifi Disconectado");
        }
    }
}
```

```
String httpGETRequest(const char* serverName)
{
    HTTPClient http;
    http.begin(serverName);
    int httpStatusCode = http.GET();
    String Datos = "--";

    if (httpStatusCode>0)
    {
        Datos = http.getString();
    }

    else
    {
        Serial.print("Error codigo: ");
        Serial.println(httpStatusCode);
    }

    http.end();
    return Datos;
}
```

El programa del cliente funciona de la siguiente manera:

Como en el caso anterior, primero hay que descargar e instalar las librerías necesarias para el correcto funcionamiento del código, las 2 librerías se encuentran en el gestor de librerías de Arduino.

```
#include <WiFi.h>
#include <HTTPClient.h>
```

Seguidamente se crean las variables con las credenciales para conectarse al AP del servidor, nombre de la red y contraseña.

```
const char* ssid = "ESP32-Access-Point";
const char* password = "123456789";
```

Posteriormente se crean las variables que se utilizan en el programa, estas son la dirección URL que le envía al servidor con la dirección IP de este. Los datos recibidos se guardan en la variable de texto "VI", las ultimas variables se utilizan para el temporizador de peticiones, en este caso cada 200 ms envía una solicitud al servidor,

```
const char* serverNameVI = "http://192.168.4.1/VI";
String VI;
unsigned long anteriorMillis = 0;
const long intervalo = 200;
```

Dentro de la función principal *setup()*, primeramente se inicia el monitor serie y se fija la velocidad de este a 115200 baudios.

```
Serial.begin(115200);
```

Después se procede a conectar a la red con el SSID correspondiente. Mientras no esté conectado, se muestra “conectando” y un “.” cada medio segundo en el monitor serial. Cuando esté conectado se muestra el mensaje con la dirección IP del *server*.

```
WiFi.begin(ssid, password);
Serial.println("Conectando");
while(WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.print("Wifi conectado a la direccion IP: ");
Serial.println(WiFi.localIP());
```

Ahora bien, dentro del *loop()* se va a estar ejecutando continuamente lo siguiente, la duración de cada ciclo del lazo lo marca el contador con la operación, cada vez que se cumpla la condición y el cliente esté conectado a la red del servidor, se va a ejecutar la función *httpGETRequest*, que hace una petición de datos, los valores devueltos se guardan en la variable “VI” y se envían por el puerto serie. En cambio, si se cumple condición del contador, pero la conexión se ha detenido por algún motivo, se muestra por el monitor “Wifi Desconectado”.

```

unsigned long actualMillis = millis();
if(actualMillis - anteriorMillis >= intervalo)
{
    if(WiFi.status() == WL_CONNECTED )
    {
        VI = httpGETRequest(serverNameVI);
        Serial.print(VI);
        anteriorMillis = actualMillis;
    }
    else
    {
        Serial.println("WiFi Desconectado");
    }
}

```

Cuando se llama a la función *httpGETRequest*, se le pasa el nombre del servidor, posteriormente se le hace una petición de datos a este, indicando el dominio con la ruta URL definida al principio, también se crea la variable “Datos” donde se va a guardar la información recibida.

```

String httpGETRequest(const char* serverName)
{
    HTTPClient http;
    http.begin(serverName);
    int httpResponseCode = http.GET();
    String Datos = "--";

```

Sí se reciben valores, se asignan a la variable “Datos”, esta se vacía cada vez que se ejecuta la función.

```

    if (httpResponseCode>0)
    {
        Datos = http.getString();
    }

```

Si no se recibe información, se imprime por el monitor sería “Error código” y el error que se ha producido. Finalmente, la función devuelve el valor de “Datos”.

```

    else
    {
        Serial.print("Error codigo: ");
        Serial.println(httpResponseCode);
    }

    http.end();
    return Datos;
}

```

## 5.7. Interface hombre máquina.

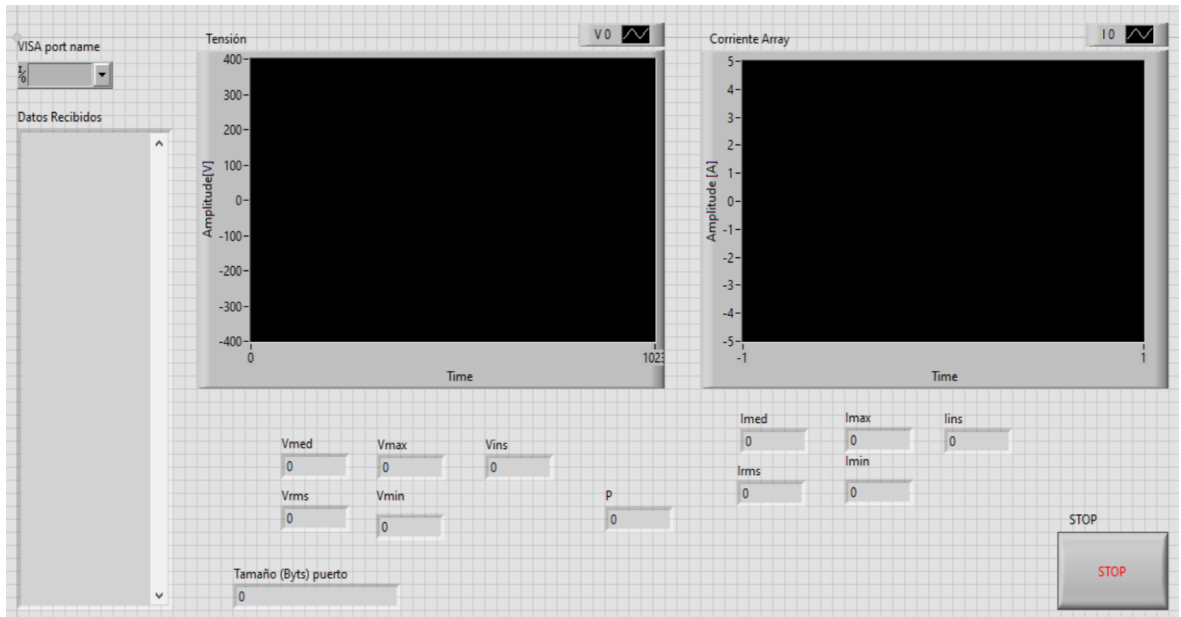
Para la visualización de los datos se necesita un entorno gráfico, para facilitar la comprensión de la información, para ello se debe presentar en una interfaz. Actualmente existen diversas plataformas de programación que permiten hacerla. Para este proyecto se ha valorado las siguientes plataformas: Visual Basic, C# y LabVIEW.

La elección ha sido LabVIEW (acrónimo de *Laboratory Virtual Instrument Engineering Workbench*). Este entorno de National Instruments es un lenguaje de programación gráfico para diseñar sistemas de control, diseño y pruebas. Es un entorno amigable e intuitivo, que ya se ha utilizado en la asignatura de instrumentación electrónica. Mientras que los otros entornos son más complejos, puesto que se requiere más conocimientos para edición de códigos fuente, y están más enfocados para crear eventos y aplicaciones.

Para crear aplicaciones en LabVIEW se dispone de dos partes diferentes:

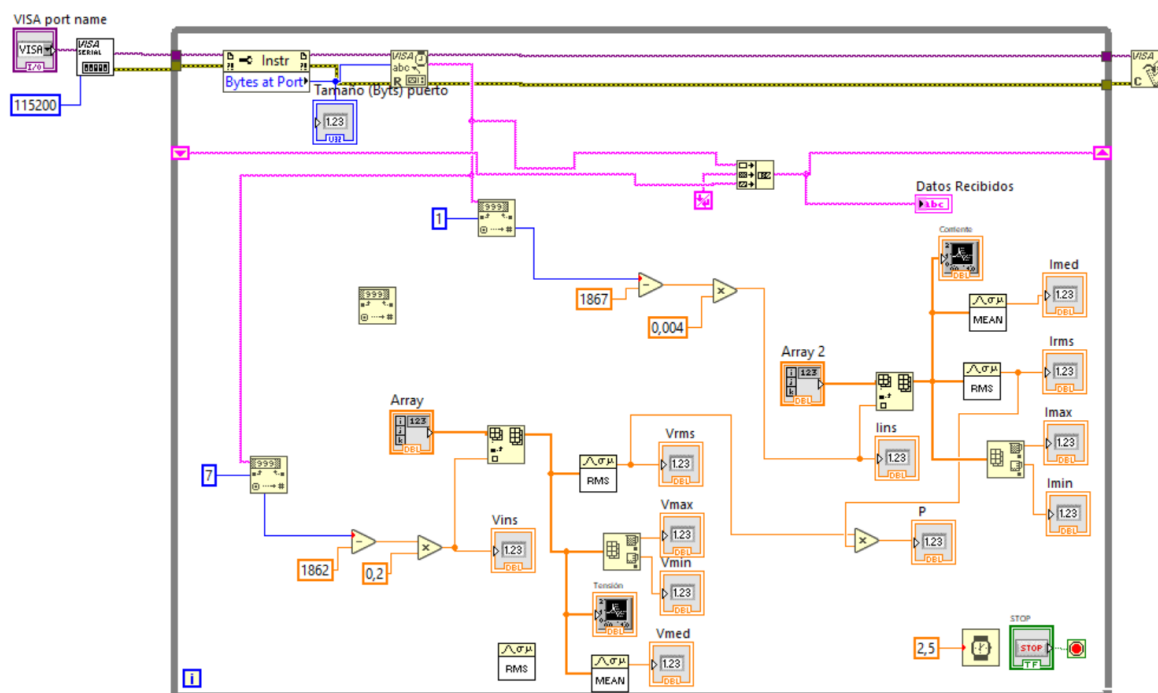
- Panel frontal (figura 41): es donde se muestra la interfaz con el usuario (HMI). Cuando se ejecuta el programa el usuario puede interactuar con este, y muestra la información y los controles en tiempo real. La interfaz de usuario debe ser fácil de entender e intuitiva.
- Diagrama de bloques (figura 42): aquí se designa y determina gráficamente la función de cada elemento que aparece en el panel frontal. Se interconectan iconos con determinadas funciones para realizar una tarea más compleja.

A continuación, se presentan las 2 partes del programa hecho en LabVIEW para mostrar los datos gráficamente del instrumento electrónico.



**Figura 41:** HMI del sistema de adquisición.

Analizando el panel frontal, se puede ver en la puerta superior izquierda el “VISA por name”, aquí se escoge el puerto COM al que está conectado el ESP32 cliente, debajo de este se muestran los datos entrantes en puerto, en formato de columna, y alado, el tamaño en Byts del *buffer* del puerto. En la parte superior se encuentran las 2 gráficas que muestran la forma de onda de la tensión y la corriente medida, es decir, grafican las muestras recibidas. En la parte inferior se muestran valores medios, eficaces, máximos, mínimos e instantáneos de la tensión y la intensidad, también la potencia. Por último, en la parte inferior derecha se encuentra un botón de “Stop” para detener el programa.



**Figura 42:** HMI del sistema de adquisición.

El análisis del diagrama de bloques empieza por la selección del puerto de entrada de datos en el bloque “VISA port name”, aquí se indica la velocidad de transmisión de datos del puerto, en este caso es 115200 Bd. La información recibida se transmite al interior del bucle *while* (recuadro gris), se guardan los datos en el primer bloque, y cada muestra se opera con ella en los siguientes bloques. La siguiente función (figura 44) consiste en separar los datos correspondientes a la tensión y la corriente. La información que llega por el puerto tiene formato que se muestra en la figura 43:

	Tensión						Corriente					
1	X	X	X	X	;	1	X	X	X	X		Vector de datos
0	1	2	3	4	5	6	7	8	9	10		Posiciones

**Figura 43:** Esquema del vector de datos.

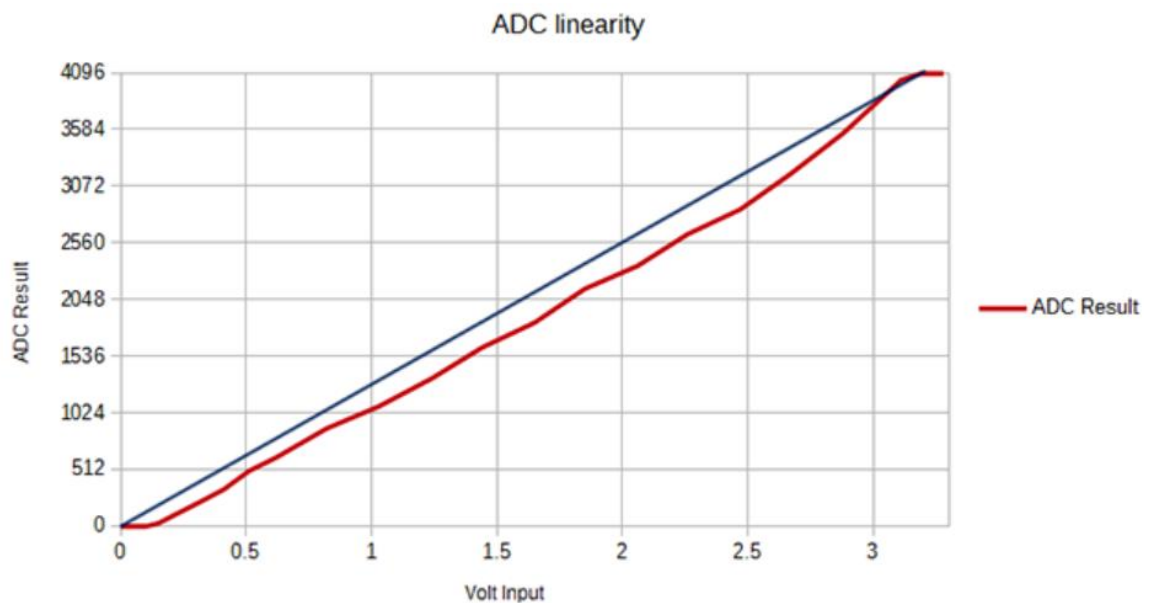
Para separar las muestras, se hace uso del bloque “*Decimal String to Number*” (figura 44), como su nombre indica convierte la información entrante de formato texto a numérico, se le indica desde que posición hace conversión (1 en el caso del voltaje, y 7 en el caso de la corriente), y hasta que encuentra el “;” es el primero valor.





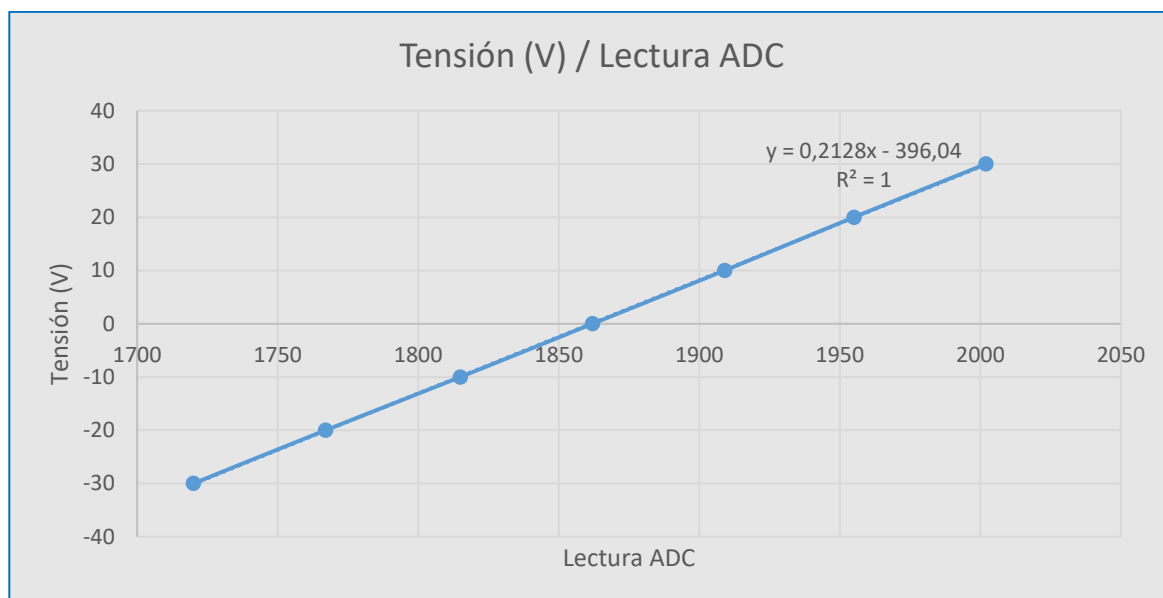
**Figura 44:** Bloque de LabVIEW “*Decimal String to Number*”.

Posteriormente se hace la conversión de cada dato, de valores decimales de entre 0 y 4095 leídos por el ADC, al valor correspondiente en tensión o corriente. El convertidor que incorpora el módulo no está bien calibrado, puesto que los valores de tensión leídos en las entradas no corresponden con los valores teóricos que debería entregar el ADC, como se observa en la figura 45, los valores reales entregados por el convertidor son ligeramente inferiores a los valores teóricos, y se pierde ligeramente la linealidad.



**Figura 45:** Relación voltaje de entrada y resultado del ADC real y teórica [25].

Con la ayuda de una fuente de alimentación, se ha generado diversos valores de tensión, y capturado la salida del ADC con el ESP32, y con un multímetro se mide la tensión que entrega la fuente. Posteriormente con una hoja de cálculo se ha hecho la regresión lineal, obtenido así la gráfica la que se muestra en la figura 46, como se puede ver, el resultado la relacional es perfectamente lineal ( $R^2=1$ ).



**Figura 46:** Relación voltaje medido y lectura del ADC.

Después de hacer la conversión a valores de tensión y corriente medidos, con las muestras se forma un vector de datos, y a partir de estos se representan en las gráficas y se calculan los valores eficaces, medios, máximos, etc.

## 6. Análisis del impacto ambiental.

Como indica en ficha técnica de las sondas amperimétricas SCT-013, los ESP32 y la fuente de alimentación IRM-03-3.3 están fabricados bajo el certificado verde RoHS (de las siglas en inglés *Restriction of Hazardous Substances*), integrado el junio de 2011 en la Directiva 2011/65/UE, esta restringe el uso de ciertos materiales peligrosos en la fabricación de equipos electrónicos y/o eléctricos. A nivel nacional esta traspuesta en el Real Decreto 219/2013. Esta Directiva que entró en vigor en 2006 restringe el uso de los siguientes materiales: Plomo, Mercurio, Cadmio, Cromo VI, PBB y PBDE. Para que un equipo tenga este certificado, cada parte de este no puede llevar más de un 0,1% de concentración de estos materiales (0,01% en el caso del cadmio).



**Figura 47:** Sonda de corriente con logo RoHS [26].

Al tener este certificado, estos componentes cuando sean residuos de equipos eléctricos y electrónicos, tendrán poca concentración de sustancias peligrosas, y reduciendo el impacto que puedan tener en la naturaleza.

Proyecto que se puede utilizar para mejorar la eficiencia energética, teniendo un impacto positivo en el medio ambiente. Durante su uso y lo largo de su vida útil el producto no genera residuos, ni emite gases. Cuando deje de utilizarse, ya sea porque se ha dañado algún parte o ya no tiene utilidad, los componentes que no estén dañados y funcionen correctamente, se pueden aprovechar para realizar otro proyecto, y la otra parte se debe reciclar como chatarra electrónica.

Equipo final con un consumo contenido que varía entre 450 mW y 550 mW, esta potencia varía según la demanda de la CPU del módulo cuando tiene que enviar la información.

## Conclusiones.

Con este proyecto se ha conseguido crear un sistema automatizado de adquisición de datos que monitoriza el estado de variables eléctricas de una carga. Con todo lo que conlleva, en el proyecto se ha tratado diversos temas relacionados con la ingeniería electrónica industrial y automáticas, relacionados con el desarrollo y diseño de *hardware* (circuitos y filtros) y temas de *software* como la programación y comunicaciones

A parte de aplicar los conocimientos adquiridos durante los estudios de ingeniería, este proyecto me ha hecho reforzar y adquirir conocimiento y experiencia sobre temas como las comunicaciones y la programación de microcontroladores.

Sin duda alguna, la parte más compleja que ha tenido este proyecto ha sido la programación, ya que el módulo utilizado es nuevo y está en desarrollo. La complejidad también se debe a que se trabaja con grandes paquetes de muestras, pues no es lo mismo monitorizar la tensión y la corriente en que se requieren muchas muestras, que medir la temperatura y la humedad de un lugar, porque se requieren menor manejo de datos y el muestreo no es tan crítico. Por ello gracias a este proyecto se ha adquirido conocimientos y experiencia en las comunicaciones y tratamiento de datos.

El diseño final cumple con los objetivos fijados inicialmente. A este se le pueden aplicar diversas mejoras. La primera mejora que se debería hacer a los circuitos es introducir más elementos de seguridad, por ejemplo, colocar diodos Zener en las entradas de los puertos GPIO del ESP32 para limitar la tensión máxima que pueden adquirir, y que conduzcan en inversas si se supera una tensión determinada. También aislar los circuitos de la placa PCB, mediante una estructura de plástico que evita un contacto directo.

Respecto al tratamiento de datos, se puede mejorar la indexación de estos, es decir, codificarlos de tal manera que se pueda saber si se han perdido datos al hacer muestreo y/o envío de datos entre placas. La parte que más se puede mejorar es la interfaz gráfica, hacer más cálculos como por ejemplo de energía frecuencia, introducir una opción para poder cambiar de fondo de fondo de escala de la corriente al usar una pinza concreta.

## Presupuesto.

Los componentes utilizados para construir este proyecto se compraron en la tienda Diotronic S.A, menos la MCU ESP32 y las pinzas de Corriente que se han obtenido en la tienda *online* Amazon. Cabe destacar que la cantidad necesaria de resistencias para realizar los circuitos es una unidad, pero el mínimo que subministra la tienda son 5 uds. En la tabla X, se presenta la lista de material que compone el proyecto, con sus respectivos precios.

**Tabla 9:** Presupuesto invertido en el TFG.

Componente	Valor	Característica	Cantidad	Precio
Resistencia	75 k $\Omega$	¼ W	1	0,12 €
Resistencia	51 k $\Omega$	¼ W	1	0,12 €
Resistencia	5,1 k $\Omega$	¼ W	1	0,12 €
Resistencia	510 $\Omega$	¼ W	1	0,12 €
Resistencia	1 k $\Omega$	¼ W	1	0,12 €
Condensador	68 nF	5V AC 2Vrms	1	0,1 €
Condensador	3,3 nF	15V AC 9Vrms	1	0,05 €
Condensador	1 nF	150Vac 100Vrms	1	0,05 €
Diodo LED	1,6 V	Rojo bajo brillo	1	0,34
Diodo Schottky	Vd baja		2	0,1 €
Condensador	33 $\mu$ F	35V	1	0,05
Sonda Corriente	SCT-013-005		1	7,64 €
Sonda Corriente	SCT-013-025		1	7,64 €
ESP32-WROOM			2	15,1 €
Fuente de alimentación	3 W 3,3 V		1	6,64 €
			<b>Total</b>	<b>38,31 €</b>

El precio total de los componentes es 38,31 €. La universidad ha subministrado el soldador (valorado en 45€) y el estaño (3€) y la placa de prototipos (valorada en 4,1€) donde se han construido los circuitos y también los cables correspondientes para realizar las conexiones del sistema DAC a la carga que se pretende estudiar.

El presupuesto invertido en el proyecto se puede recuperar rápidamente, haciendo uso de este, y ahorrando en consumo energético.

El precio final es inferior al de los productos analizados en el estado del arte, siendo inferior a Akozon, pero con unas funcionalidades mucho más extensas. Cabe destacar que este precio puede variar según el número pinzas se adquieran, en este caso se han utilizado 2, y sube el precio hasta 38,31€.

A este presupuesto hay que añadirle el coste humano que lleva su investigación y diseño, en total se le han dedicado aproximadamente 600h; 650h.

## Bibliografía.

- [1]. FELIPE MATEOS MARTÍN. *Ingeniería de automatización*. Universidad de Oviedo, Departamento de ingeniería de sistemas y automática, Gijón, 2018.
- [2]. ESTEFANI MARCANO y ILIANA PARIS. *Introducción a la Automatización Industrial*. Equipo OPC, Universidad de Oriente, Santiago de Cuba, 2016 [Consulta 8/8/2020]. Disponible en:  
<https://sites.google.com/site/equipoopcaccpic4/seminario-eai/introduccion-a-la-automatizacion-industrial>
- [3]. EFERGY TECHNOLOGIES SL. *Engage solar kit*. Barcelona, 2019 [Consulta 10/8/2020], disponible en:  
<https://efergy.com/engage/#over>
- [4]. AKOZON. *Akazon voltímetro multímetro CC probador inalámbrico del voltaje del metro del poder del amperímetro de la corriente del LCD de la pantalla color (VAT-1100, 0-100A 0-100V)*. 2018 [Consulta 10/8/2020]. Disponible en:  
[https://www.amazon.es/Akazon-volt%C3%ADmetro-mult%C3%ADmetro-inal%C3%A1mblico-amper%C3%ADmetro/dp/B07GTLMMBB/ref=sr\\_1\\_9?tag=9-lacompra-21](https://www.amazon.es/Akazon-volt%C3%ADmetro-mult%C3%ADmetro-inal%C3%A1mblico-amper%C3%ADmetro/dp/B07GTLMMBB/ref=sr_1_9?tag=9-lacompra-21)
- [5]. DAVID EGIDO NIETO. *Diseño y construcción de un equipo de medida de energía eléctrica de bajo coste*. Director: Pablo Frías Marín e Ignacio Egido Cortés. ICAI – Universidad Pontificia Comillas, Madrid, 2015.
- [6]. RAMÓN PALLÀRS ARENY. *Adquisición y distribución de señales*. Universidad Politécnica de Catalunya. Editorial Marcombo Boixareu Editores, Barcelona, 1993. ISBN: 8426709184
- [7]. MAURIZIO DI PAOLO EMILIO. *Data Acquisition Systems: From fundamental to applied Design*. Editorial Springer. Pescara, Italia, 2013. ISBN 978461442141.
- [8]. DAVID ASHLOCK y ANJELICA WARREN. *Guía de acondicionamiento de señales para ingenieros*. National Instruments, 2015.
- [9]. ARDUINO YUN. *Datos técnicos*. 2015 [Consulta 15/8/2020]. Disponible en página web oficial del fabricante:  
<https://store.arduino.cc/arduino-yun>
- [10]. LUCY HATTERSLEY. *10 Pi Zero W projects*. Revista The Magpi. Edición ISSUE 61, Londres, septiembre 2017.

- [11]. RASPBERRY PI ZERO W. *Datos técnicos*. 2016 [Consulta 15/8/2020]. Disponible en página web oficial del fabricante:  
<https://www.raspberrypi.org/products/raspberry-pi-zero-w/>
- [12]. ESPRESSIF SYSTEMS. *ESP32 Series Datasheet*. Version V3.4, 2019.
- [13]. ESPRESSIF. *ESP32*. 2016 [Consulta 15/8/2020]. Disponible en:  
<https://www.espressif.com/en/products/socs/esp32>
- [14]. DANIEL PRIETO. *Sistemas y señales: Sistemas discretos y sistemas continuos* [Imagen Gráfica]. Madrid, 2012 [Consulta 17/8/2020]. Disponible en:  
<http://danielprieto-sys-2012.blogspot.com/2012/05/sistemas-discretos-y-sistemas-continuos.html>
- [15]. SERGIO FRANCO. *Diseño con amplificadores operacionales y circuitos integrados analógicos*. Editorial McGraw-Hill, México, 2005. ISBN 9701045955.
- [16]. G. VELASCO, M. ROMÁN, R. PÉREZ, F. JEREZ, P. COLOMER Y V. REPECHO. *Transductor Flux gate para la Medida de Corriente Continua y Corriente Alterna de Bajo Consumo*. Universidad Politécnica de Catalunya (UPC), Departamento Técnico Grupo Premo S.A. Barcelona 2017.
- [17]. LUIS LLAMAS. *Sensor de corriente eléctrica no invasivo con Arduino y sct-013*. 2017 [Consulta 25/4/2020]. Disponible en:  
<https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>
- [18]. UNIVERSIDAD NACIONAL DE LA PATAGONIA SAN JUAN BOSCO. *Tutorial Generador de funciones*, Rivadavia, Argentina 2002 [Consulta 4/9/2020]. Disponible en:  
<http://www.ing.unp.edu.ar/electronica/asignaturas/ee016/tutoriales/generador/generador.htm>
- [19]. Aprendiendo Arduino [Tabla tensiones LED]. 2019 [Consulta 25/8/2020]. Disponible en:  
<https://aprendiendoarduino.wordpress.com/tag/led-rgb/>
- [20]. MEAN WELL. *Datasheet IRM-03*. 2019.
- [21]. ARDUINO. *Arduino IDE*. 2018 [Consulta 30/8/2020]. Disponible en:  
<https://www.arduino.cc/en/Guide/Introduction>
- [22]. SARA SANTOS. *Getting Started with ESP-NOW (ESP32 with Arduino IDE)*. Random Nerd Tutorials. 2020 [Consulta 25/3/2020]. Disponible en:  
<https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/>



- [23]. SARA SANTOS. *ESP32 Client-Server Wi-Fi Communication Between Two Boards*. Random Nerd Tutorials. 2020 [Consulta 15/4/2020]. Disponible en:  
<https://randomnerdtutorials.com/esp32-client-server-wi-fi/>
- [24]. SARA SANTOS. *ESP32 with LoRa using Arduino IDE – Getting Started*. Random Nerd Tutorials. 2020 [Consulta 10/9/2020]. Disponible en:  
<https://randomnerdtutorials.com/esp32-lora-rfm95-transceiver-arduino-ide/>
- [25]. HENRY CHEUNG. *ESP32: adc calibrate*. 2020 [Consulta 25/9/2020]. Disponible en:  
<https://github.com/e-tinkers/esp32-adc-calibrate>
- [26]. EDDING. *RoHS: la directiva y su cumplimiento*. Alcalá de Henares, 2018[Consulta 25/9/2020]. Disponible en:  
<https://www.edding.com/es-es/temas/seccion-wiki/rohs/>
- [27]. Librerías Arduino IDE
- Wifi:  
<https://github.com/espressif/arduino-esp32/tree/master/libraries/WiFi>
  - ESPAsyncWebServer:  
<https://github.com/me-no-dev/ESPAsyncWebServer/archive/master.zip>
  - HTTPCliente:  
<https://github.com/amcewen/HttpClient>

## **Anexos.**

### **TRABAJO FINAL DE GRADO**

**Grado en Ingeniería Electrónica Industrial y Automática**

## **SISTEMA DE ADQUISICIÓN DE DATOS CON ESP32**



## **Anexos**

**Autor/a:** Ikiss, Jamal  
**Director/a:** Casellas Beneyto, Francisco José  
**Convocatoria:** Setiembre 2020



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

## A1. Datasheet sondas amperimétricas SCT-013.

**Model: SCT-013**
**Rated input current: 5A/100A**

**Characteristics:** Opening size: 13mm\*13mm,  
 Non-linearity±3% (10%—120% of rated input current)  
 1m leading wire, standard Φ3.5 three core plug output.  
 Current output type and voltage output type (voltage output type built-in sampling resistor)

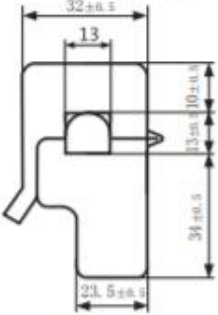
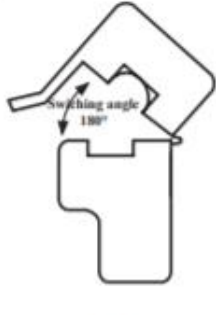

**Purpose:** Used for current measurement, monitor and protection for AC motor, lighting equipment, air compressor etc

**Core material:** ferrite

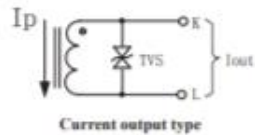
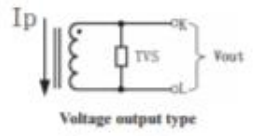
**Mechanical strength:** the number of switching is not less than 1000 times(test at 25℃)

**Safety index:** Dielectric strength(between shell and output)1000V AC/1min  
 Fire resistance property: In accordance with UL94-Vo  
 Work temperature: -25℃~+70℃

**Outline size diagram: (in mm)**

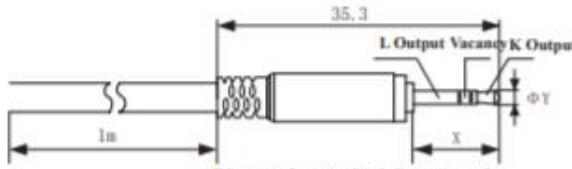
**Schematic diagram**

**Three-core plug size**

	X	Y	select
2.5mm Audio Plug	11.9	2.5	and use
3.5mm Audio Plug	15.0	3.5	standard

**Diagram for standard three-core plug**



**Table of technical parameter:**

Model	SCT-013-000	SCT-013-005	SCT-013-010	SCT-013-015	SCT-013-020
Input current	0-100A	0-5A	0-10A	0-15A	0-20A
Output type	0-50mA	0-1V	0-1V	0-1V	0-1V

Model	SCT-013-025	SCT-013-030	SCT-013-050	SCT-013-060	SCT-013-000V
Input current	0-25A	0-30A	0-50A	0-60A	0-100A
Output type	0-1V	0-1V	0-1V	0-1V	0-1V

※ Output type: voltage output type built-in sampling resistor, current output type built-in protective diode.

Beijing Yaohuadechang Electronic Co., Ltd  
 Web (China): [www.yhdc.com](http://www.yhdc.com) Web (Latvia): [www.yhdc.lv](http://www.yhdc.lv)

## A2. Especificaciones técnicas fuente de alimentación IRM-03-3.3.



3W Single Output Encapsulated Type

IRM-03 series

### SPECIFICATION

MODEL		IRM-03-3.3	IRM-03-5	IRM-03-9	IRM-03-12	IRM-03-15	IRM-03-24
OUTPUT	DC VOLTAGE	3.3V	5V	9V	12V	15V	24V
	RATED CURRENT	900mA	600mA	333mA	250mA	200mA	125mA
	CURRENT RANGE	0 ~ 900mA	0 ~ 600mA	0 ~ 333mA	0 ~ 250mA	0 ~ 200mA	0 ~ 125mA
	RATED POWER	3W	3W	3W	3W	3W	3W
	RIPPLE & NOISE (max.) <small>Note.2</small>	100mVp-p	100mVp-p	100mVp-p	150mVp-p	200mVp-p	240mVp-p
	VOLTAGE TOLERANCE <small>Note.3</small>	± 2.5%	± 2.5%	± 2.5%	± 2.5%	± 2.5%	± 2.5%
	LINE REGULATION	± 0.5%	± 0.5%	± 0.5%	± 0.5%	± 0.5%	± 0.5%
	LOAD REGULATION	± 1.0%	± 0.5%	± 0.5%	± 0.5%	± 0.5%	± 0.5%
	SETUP, RISE TIME	600ms, 30ms/230VAC      600ms, 30ms/115VAC at full load					
HOLD UP TIME (Typ.)	40ms/230VAC      8ms/115VAC at full load						
INPUT	VOLTAGE RANGE	85 ~ 305VAC 120~430VDC					
	FREQUENCY RANGE	47 ~ 63Hz					
	EFFICIENCY (Typ.)	68%	72%	77%	78%	78%	80%
	AC CURRENT (Typ.)	70mA/115VAC	40mA/230VAC	35mA/277VAC			
	INRUSH CURRENT (Typ.)	10A/115VAC	20A/230VAC				
	LEAKAGE CURRENT	< 0.25mA/277VAC					
PROTECTION	OVERLOAD	105%~260% rated output power Protection type : Hiccup mode, recovers automatically after fault condition is removed					
	OVER VOLTAGE	3.8 ~ 4.9V	5.2 ~ 6.8V	10.3 ~ 12.2V	12.6 ~ 16.2V	15.75 ~ 20.3V	25.2 ~ 32.4V
		Protection type : Shut off o/p voltage, clamping by zener diode					
ENVIRONMENT	WORKING TEMP.	-30 ~ +85℃ (Refer to "Derating Curve")					
	WORKING HUMIDITY	20 ~ 90% RH non-condensing					
	STORAGE TEMP., HUMIDITY	-40 ~ +100℃, 10 ~ 95% RH					
	TEMP. COEFFICIENT	± 0.03%/℃ (0 ~ 50℃)					
	VIBRATION	10 ~ 500Hz, 5G 10min./1cycle, period for 60min. each along X, Y, Z axes					
	LEAD TEMPERATURE	260℃, 10s (max.)					
SAFETY & EMC	SAFETY STANDARDS	UL62368-1, TUV EN62368, TUV EN60335-1, EAC TP TC 004, BSMI CNS14336-1 approved, design refer to EN61558-2-16 , IEC60601-1 (By request)					
	WITHSTAND VOLTAGE	I/P-O/P:3KVAC					
	ISOLATION RESISTANCE	I/P-O/P:100M Ohms / 500VDC / 25℃/ 70% RH					
	EMC EMISSION	Compliance to EN55032 (CISPR32) Class B, EN61000-3-2,-3, EAC TP TC 020, CNS13438 Class B					
	EMC IMMUNITY	Compliance to EN61000-4-2,3,4,5,6,8,11, EN55024, heavy industry level (surge L-N : 1KV), criteria A, EAC TP TC 020					
OTHERS	MTBF	2137.6Khrs min. MIL-HDBK-217F (25℃)					
	DIMENSION	PCB mounting style : 37*24*15mm (L*W*H)      SMD style : 37*24*16mm (L*W*H)					
	PACKING	PCB mounting style : 0.026Kg;560pcs/15.3Kg/0.96CUFT      SMD style :0.026Kg;560pcs/15.3Kg/0.96CUFT					
NOTE	1. All parameters NOT specially mentioned are measured at 230VAC input, rated load and 25℃ of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation. 4. The ambient temperature derating of 3.5℃/1000m with fanless models and of 5℃/1000m with fan models for operating altitude higher than 2000m(6500ft).						

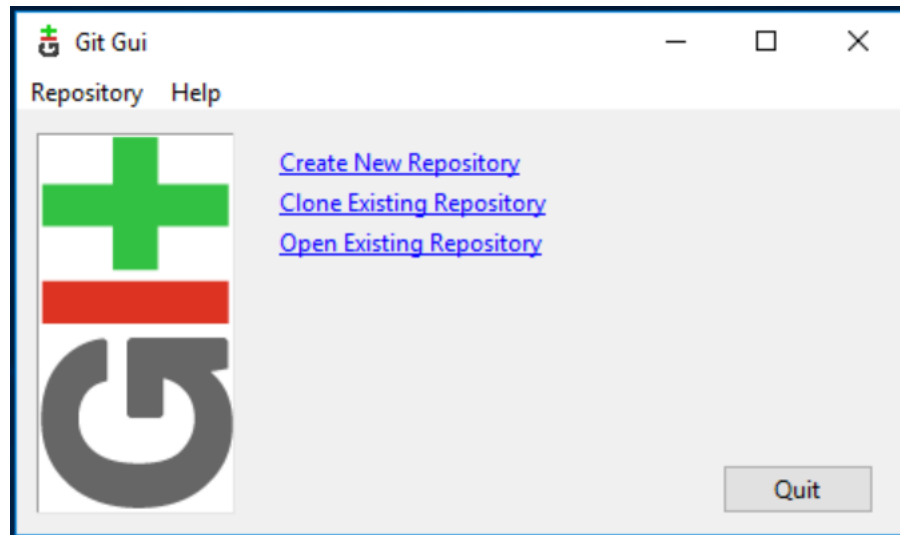
File Name:IRM-03-SPEC 2019-12-09


 UNIVERSITAT POLITÈCNICA DE CATALUNYA  
 BARCELONATECH  
 Escola d'Enginyeria de Barcelona Est

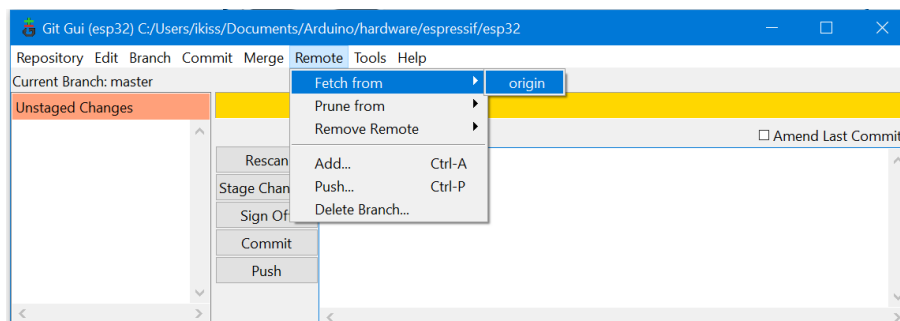
### A3. Instalación del ESP32 en el Arduino IDE.

A continuación, se muestran los pasos a seguir para poder programar las placas ESP32 con el entorno de desarrollo integrado de Arduino.

1. Descargar e Instalar la última versión de Arduino IDE,  
<https://www.arduino.cc/en/Main/Software>
2. Descargar e instalar el entorno grafico GIT desde <https://git-scm.com/download/win>  
GIT es un repositorio organizado de código en desarrollo por programadores voluntarios que contiene los archivos esenciales para programar ESP32 con Arduino IDE.
3. Iniciar GIT GUI y seguir los siguientes pasos
4. En la pestaña que aparece seleccionar Clone Existing Repositor



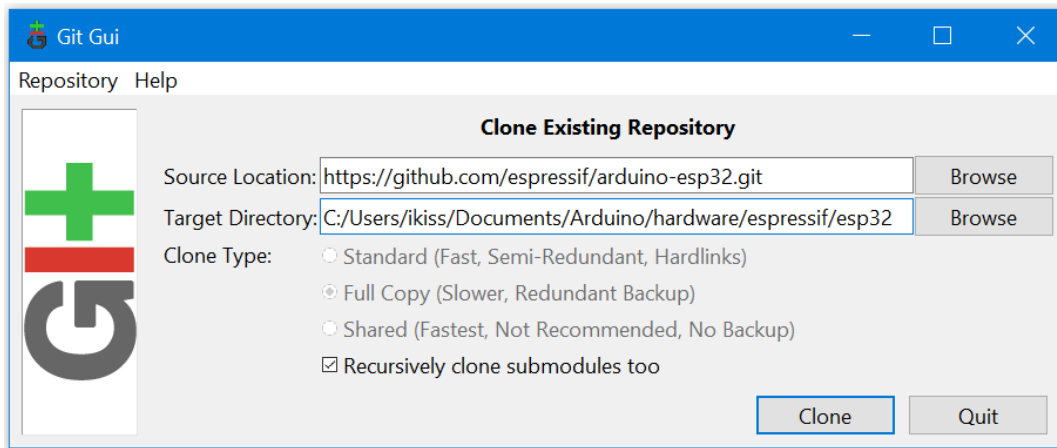
Si se quiere actualizar el repositorio se puede hacer desde el desplegable Remote<Fetch from<origin



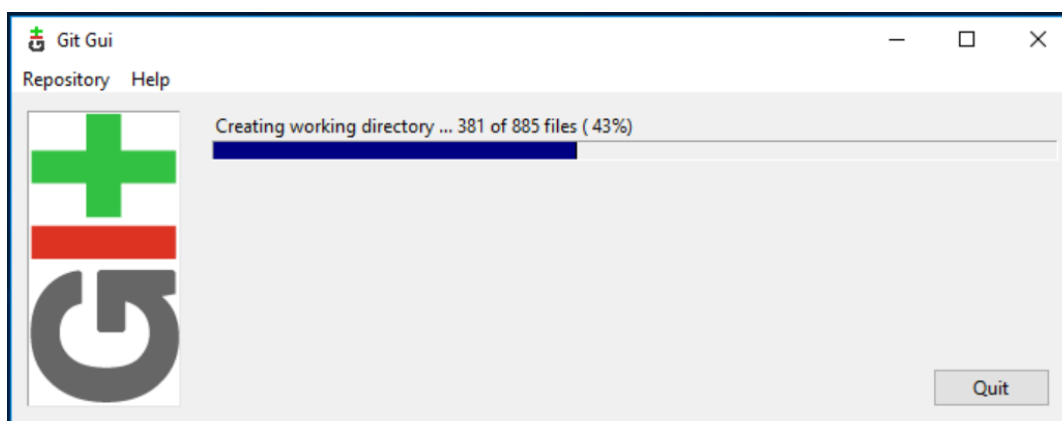
5. En la siguiente pestaña introducir el origen y el destino de la clonación:
  - Origen: <https://github.com/espressif/arduino-esp32.git>

- El destino va a ser dentro de la carpeta de Arduino que normalmente se genera en la carpeta de documentos del PC al instalar Arduino IDE, por lo tanto, el destino va ser:

C:/Users/ [Nombre-Usuario-del-PC]/Documents/Arduino/hardware/espressif/esp32



6. Clicar en Clone para clonar el repositorio, y esperar a que se complete la descarga.



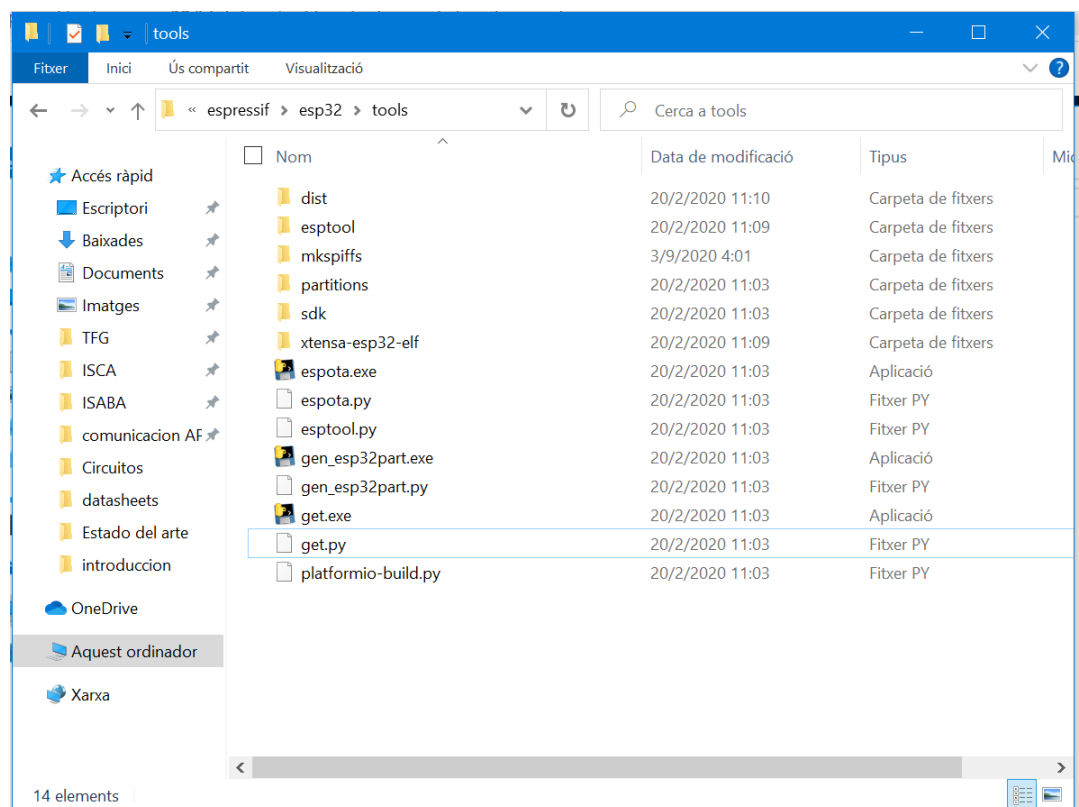
7. Entra en el directorio que se ha creado anteriormente, y en la carpeta tools ejecuta el archivo get.exe C:/Users/ikiss/Documents/Arduino/hardware/espressif/esp32/tools  
Esperar a que el ejecutable termine y se cerrará la pantalla.

```

C:\Users\vikiss\Documents\Arduino\hardware\espressif\esp32\tools\get.exe
System: Windows, Info: Windows-10-10.0.19041
Platform: i686-mingw32
Tool xtensa-esp32-elf-win32-1.22.0-80-g6c4433a-5.2.0.zip already downloaded
Extracting xtensa-esp32-elf-win32-1.22.0-80-g6c4433a-5.2.0.zip
Tool esptool-2.6.1-windows.zip already downloaded
Extracting esptool-2.6.1-windows.zip
Tool mkspiffs-0.2.3-arduino-esp32-win32.zip already downloaded
Extracting mkspiffs-0.2.3-arduino-esp32-win32.zip

```

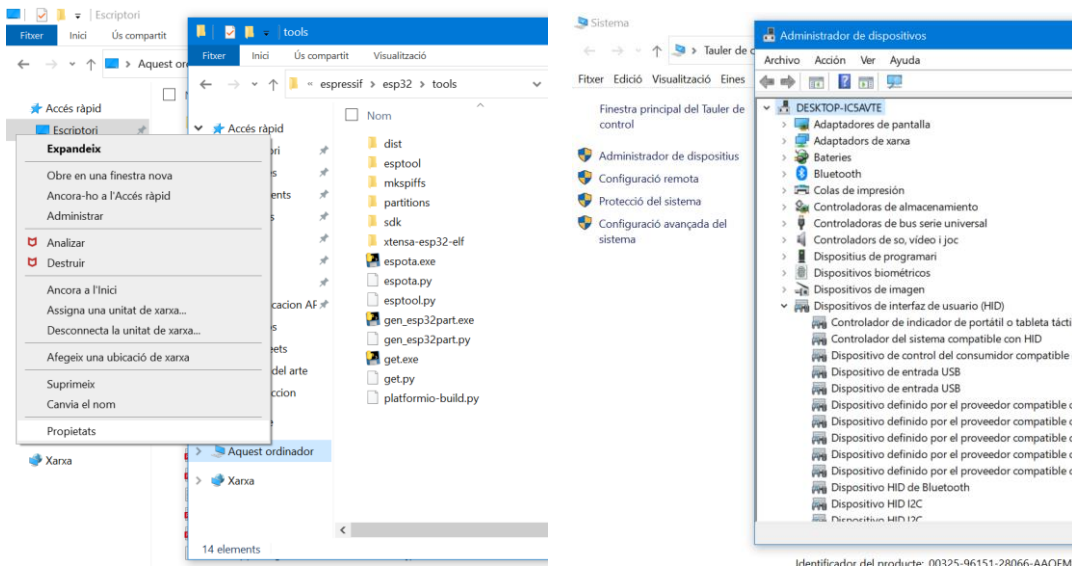
8. Cuando get.exe termine y se cierre la pantalla, se generan los siguientes archivos en la misma carpeta:



9. Conecta la placa ESP32 al PC, y es espera que se instalen los drivers del chip de comunicación serie, en caso de que no se instalen satisfactoriamente, los puedes descargar desde la web e instalarlos manualmente.



Después de esto la placa aparece en el administrador de dispositivos sin un triángulo amarillo con un signo de exclamación, y también aparece el puerto COM en el que está conectado. Al administrador de dispositivos se puede acceder dando clic derecho en la carpeta de mi ordenador y entrando en la opción de propiedad, en la pantalla que aparece a la izquierda está el administrador de dispositivos.



10. Ahora iniciar Arduino IDE.

11. En el menú Herramientas<Placa seleccionamos la placa, en este caso ESP32 Dev Kit

12. Configurar los datos de la placa.



## A4. Código del ESP32 emisor utilizando protocolo ESP-NOW.

```

masterespnow2
#include <esp_now.h>
#include <WiFi.h>

// Direccion MAC donde se van a enviar los datos.
uint8_t broadcastAddress[] = {0xA4, 0xCF, 0x12, 0x81, 0xE8, 0x08};

//definicion de la tarea a ejecutar en el core 0
TaskHandle_t Task1;

//variable global donde se guarda los datos a transferir entre cores, y estructura global para enviar
bool listo = false;
typedef struct struct_message
{
    String a;
} struct_message;

// creacion de una estructura de datos mydate, datos a enviar
struct_message myData;

// devolución de llamada cuando se envían datos
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status)
{
    Serial.print("\r\nEstado del ultimo paquete enviado:\t");
    Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Enviado con exito" : "Fallo de envio");
}

void setup()
{
    // inicial monitor serial y pines como entradas
    pinMode(36, INPUT);
    pinMode(34, INPUT);
    Serial.begin(512000);

    // Configurar el ESP32 como estacion de wifi.
    WiFi.mode(WIFI_STA);

    // Iniciar ESP-NOW
    if (esp_now_init() != ESP_OK)
    {
        Serial.println("Error al iniciar ESP-NOW");
        return;
    }

    //Una vez que ESPNow se inicie con éxito, nos registraremos para
    // obtener el estado del paquete enviado, estado de envio
    esp_now_register_send_cb(OnDataSent);

```

```

// Registrar par
esp_now_peer_info_t peerInfo;
memcpy(peerInfo.peer_addr, broadcastAddress, 6);
peerInfo.channel = 0;
peerInfo.encrypt = false;

// agregar par
if (esp_now_add_peer(&peerInfo) != ESP_OK){
    Serial.println("fallo al añadir para");
    return;
}

//Creacion del segundo loop en el core0
xTaskCreatePinnedToCore(
    loop2, //identificacion del loop
    "Task_1", //nombre de la tarea
    1000, //tamaño de la pila
    NULL, //no se le pasan valores a este loop
    1, //prioridad 1 del segundo loop.
    &Task1, //nombre de la tarea
    0); //nucleo en el que se va a ejecutar.
}

void loop()
{
    if (listo == true)
    {
        // enviar paquete via ESP-NOW
        esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
        if (result == ESP_OK){Serial.println("Enviado con exito");}
        else{Serial.println("Error en el envio");}
        Serial.println(myData.a);
        listo = false;
    }
}

void loop2(void *parameter) //actividad que se ejecuta en el segundo loop
{
    //variables de los pines de las variables
    int Voltimetro = 36;
    int Amperimetro = 34;
    //assignacion de las lecturas de los pines
    int V;
    int I;

```

```
for(;;)
{
    String VI="";
    for(int i=0; i<5; i++)
    {
        //se ejecuta continuamente la adquisicion de las variables de los puerto
        //y se guardan en el string VI
        V =analogRead(Voltimetro);
        I =analogRead(Amperimetro);
        //"\n" indicar salto de linea despues de cada muestra
        VI = (VI + String(V) + ";" + String(I));
        delay(500); //delay que permite obtener 80 muestra en un periodo de 50hz
    }
    myData.a = VI;
    listo = true;
}
}
```

## A5. Código del ESP32 receptor utilizando protocolo ESP-NOW.

### Slave\_2

```
#include <esp_now.h>
#include <WiFi.h>
// Definición de la estructura para recibir datos
// Debe coincidir con la estructura del master
typedef struct struct_message {
    String a;
} struct_message;
//Crea un struct_message llamado myData
struct_message myData;
// función de devolución de llamada que se ejecutará cuando se reciban datos
void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len) {
    memcpy(&myData, incomingData, sizeof(myData));}

void setup()
{
    // Inicializar Monitor Serial
    Serial.begin(9600);

    // Configurar el dispositivo como una estación Wi-Fi
    WiFi.mode(WIFI_STA);

    // Iniciar ESP-NOW
    if (esp_now_init() != ESP_OK)
    {
        {
            Serial.println("Error al iniciar ESP-NOW");
            return;
        }
    }

    // Una vez que ESPNow se inicie con éxito, nos registraremos para recibir respues de estado
    // obtener informacion del paquete recibido
    esp_now_register_recv_cb(OnDataRecv);
}

void loop() {
    Serial.println(myData.a);
}
```