

## Sistema remoto de monitoreo eléctrico para una central telefónica

*Remote electrical monitoring system for a telephone exchange*

 <https://doi.org/10.52948/mare.v3i2.563>

FABIO ALEXIS RODRÍGUEZ LAZARTE  
Universidad Privada del Valle (Bolivia)  
rlf1008720@est.univalle.edu

EDGAR ROBERTO RAMOS SILVESTRE  
Universidad Privada del Valle (Bolivia)  
eramoss@univalle.edu

### Artículo de investigación

**Recepción:** 1 de septiembre de 2021

**Aceptación:** 22 de febrero de 2022

### Cómo citar este artículo

F. Rodríguez Lazarte y E. Ramos Silvestre, «Sistema remoto de monitoreo eléctrico para una central telefónica», *mare*, vol. 3, n.º 2, may 2022.

## Resumen:

El presente artículo se basa en un sistema de monitoreo de la alimentación eléctrica en una central telefónica, la cual se encuentra a una distancia considerable del centro de la ciudad de Cochabamba, Bolivia. El sistema realiza únicamente la lectura de parámetros de voltaje en corriente alterna proveniente de la red eléctrica, por un lado, y corriente continua resultante de un banco de baterías del sistema de respaldo, de otro; con el objetivo de detectar cortes o fallas en la alimentación eléctrica en los equipos de la central telefónica. Dichos parámetros de voltaje son adquiridos localmente mediante circuitos de acondicionamiento de señales, tales como conversores, atenuadores, filtros y amplificadores. Se basan en instrumentos de medición de voltaje convencionales para que posteriormente las señales acondicionadas puedan ser introducidas en un sistema de procesamiento. Este se encargará de procesar dichos valores y visualizarlos localmente mediante un display de cristal líquido (LCD), y remotamente a través de una aplicación Android o una página web. Para el monitoreo en tiempo real, se implementó una arquitectura cliente-servidor basados en consultas HTTP para publicar y recuperar datos. Para ello se usó un módulo ethernet que interactúa con el módulo de procesamiento. El módulo ethernet utiliza determinados protocolos del modelo TCP/IP para conectarse a internet. Para almacenar los datos se utilizó ThingSpeak, una plataforma de internet de las cosas con funciones de agregación, análisis y visualización de un flujo de datos en la nube en tiempo real.

**Palabras clave:** central telefónica; sistema de respaldo; banco de baterías; acondicionamiento de señales; protocolos de comunicación; internet de las cosas.

## Abstract:

This article is based on a power supply monitoring system in a telephone exchange found at a considerable distance from the center of the Cochabamba-Bolivia city. The system only performs the reading of voltage parameters in alternating current from the electrical network and direct current from the backup system battery bank, with the aim of detecting cuts or failures in the electrical supply in the equipment of the telephone exchange. These voltage parameters are locally acquired using signal conditioning circuits such as converters, attenuators, filters, and amplifiers, which are based on conventional voltage measuring instruments so that later these conditioned signals can be introduced to a processing system that it's in charge of processing these values and displaying them locally through a display and remotely from an Android application or a web page. For real-time monitoring, it implements a client-server architecture based on HTTP queries to publish and retrieve data. For this, it uses an ethernet module that interacts with the processing module. The ethernet module applies the TCP/IP model to connect to the Internet and to store the data. It uses ThingSpeak as an Internet of Things platform for aggregation, analysis, and visualization functions for real-time cloud data flow.

**Keywords:** telephone exchange; backup system; battery bank; signal conditioning; communication protocols; internet of things.

## Introducción

En los últimos años, las telecomunicaciones fueron avanzando a pasos bastante grandes. Actualmente es casi imprescindible el uso de servicios de comunicaciones tales como telefonía, televisión por cable e

internet; siendo el más importante y utilizado el servicio de internet, en el cual convergen los tres (3) servicios.

Para una compañía que brinda servicios de telecomunicaciones es importante que tenga centrales telefónicas distribuidas, las cuales generalmente están localizadas a distancias considerables del centro de la ciudad. Allí están principalmente las interconexiones departamentales y rurales (redes de transporte), así como de usuario final (redes de acceso o de última milla). Estas centrales telefónicas albergan equipos activos para mantener dichas interconexiones, por lo que requieren constantemente alimentación eléctrica para la alimentación de los equipos. El servicio que contratan los abonados de determinada empresa de telecomunicaciones debe ser ininterrumpido. Por tal motivo las centrales telefónicas cuentan generalmente con sistemas de respaldo de energía (proveniente de un banco de baterías, grupos electrógenos u otros) [1], que por determinado tiempo evitan una interrupción en la alimentación eléctrica en los equipos encargados de brindar servicios en telecomunicaciones. Debido a la alta demanda de servicios de comunicación, se debe asegurar la disponibilidad de los servicios y tener un soporte de personal técnico inmediato en la central telefónica, en caso de algún corte de energía en la red eléctrica. De esta manera, evitando el agotamiento de energía almacenada en los sistemas de respaldo.

La implementación de un sistema de monitoreo eléctrico local y remoto en una central telefónica es el principal objetivo del presente artículo. Se busca realizar un monitoreo de parámetros de alimentación eléctrica; el parámetro principal de monitoreo es el voltaje, tanto de corriente alterna como de corriente continua. Para ello se tomará en cuenta de-

terminados umbrales de voltaje de funcionamiento de los equipos (máximos y mínimos), antes de generar una interrupción total de energía en los equipos de la central telefónica, ya sea por corte, sobrevoltaje, o caída de voltaje. Debido a la distante ubicación en las centrales, es importante un almacenamiento de datos en la nube; es necesario implementar protocolos de comunicación del modelo TCP/IP; además, ejecutar una arquitectura cliente-servidor con el motivo de almacenar y recuperar los datos mediante consultas web en tiempo real y generar alarmas en caso de eventos inesperados.

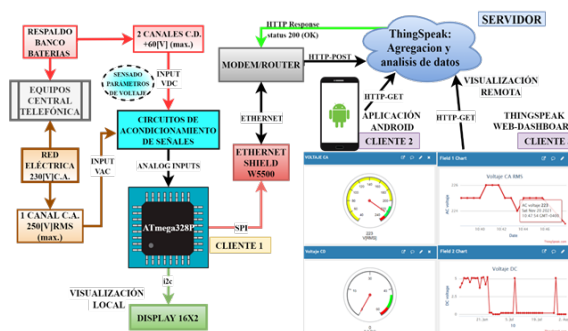


Figura 1. Diagrama general del sistema. Elaboración propia.

Para ello se utilizó el servicio en la nube ThingSpeak como plataforma de internet de las cosas (IoT) en su versión gratuita. ThingSpeak está basado en canales (públicos y privados) para recuperar y almacenar datos de diferentes dispositivos usando la transferencia del estado representacional, llamada API REST (get, post, update and delete). Estas solicitudes son gestionadas a través del protocolo HTTP.

En la figura 1 se puede observar el diagrama general del sistema. El microcontrolado (cliente 1) se encarga de procesar e interpretar las señales de entrada que se obtienen mediante los circuitos de acondicionamiento de señal. Una vez realizada la adecuada interpre-

tación, calibración y ajuste de los valores de voltaje mediante algoritmos en la programación del microcontrolador, estos se visualizan en una pantalla localmente. Al mismo tiempo se realiza la publicación de los datos con una conexión a internet mediante el módulo ethernet hacia los canales de voltaje respectivos (corriente alterna y corriente continua) de ThingSpeak, usando el método POST del protocolo HTTP. La plataforma tiene la función de servidor y está encargado de almacenar datos y responder solicitudes provenientes del cliente 1 (figura 1) para que, posteriormente un dispositivo Android (cliente 2), o un navegador web (cliente 3), que requieran leer los datos en un determinado canal de voltaje, puedan consultarlos mediante el método GET del protocolo HTTP en un formato JSON (notación de objetos de JavaScript); un formato estándar basado en texto de fácil manipulación, lectura y escritura.

## Metodología

### Principio de obtención de parámetros eléctricos básicos con un multímetro.

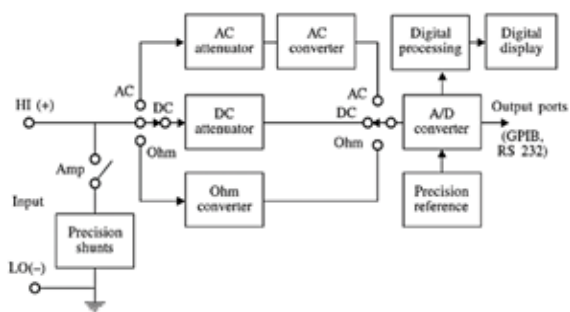
La forma de medir las tres (3) magnitudes eléctricas principales en un circuito (voltaje, corriente y resistencia) puede ser de forma analógica o digital. Los dispositivos de medición analógica utilizan el movimiento electromecánico de un puntero sobre una escala calibrada para mostrar el valor de determinada cantidad conocida como movimiento d'Arsonval [2].

Los dispositivos de medición digital utilizan pantallas LED numéricas o alfanuméricas, así como de cristal líquido (LCD) para mostrar la lectura de la cantidad medida.

El voltaje es esencial en cualquier tipo de circuito eléctrico. Es la energía potencial de la carga eléctrica requerida para

que el circuito trabaje. La corriente también es necesaria para que operen los circuitos eléctricos, pero se requiere voltaje para producirla [3]. Alexander y Sadiku [4] indican que, si la corriente no cambia con el tiempo, sino que permanece constante, se conoce como corriente directa (CD). En cambio sí una corriente varía sinusoidalmente con el tiempo es denominada corriente alterna (CA). La forma de onda sinusoidal es el tipo fundamental de corriente alterna como se muestra en la figura 4, debido a su bajo costo, fácil generación y transporte para un suministro energético a hogares, fábricas laboratorios, etc. [3]. Como el objetivo del presente artículo se basa en la detección de cortes de energía, solo se analizará el principio de funcionamiento de lectura de voltaje eficaz en CA y voltaje en CD.

Un medidor digital utiliza conversores analógicos a digital y otros circuitos de procesamiento digital para la obtención de parámetros eléctricos previamente acondicionados mediante conversores, atenuadores, amplificadores y filtros [2]. En la figura 2 se puede ver el diagrama básico de un multímetro digital básico.

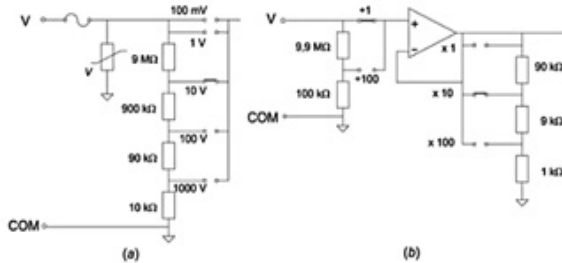


**Figura 2.** Diagrama de bloques de un multímetro digital básico. [2]

### Principio de medición de voltaje CD.

En el diagrama de bloques de la figura 2 los atenuadores son utilizados para re-

ducir el voltaje de entrada CD. Luego es interconectado con un conversor analógico digital que lo convierte en una forma digital. Esta señal es procesada y mostrada en un despliegue [2]

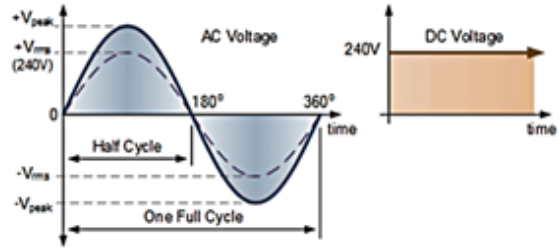


**Figura 3.** Atenuadores y amplificadores de entrada para multímetros digitales. [5]

Areny [5] indica que el atenuador es el elemento que establece la impedancia equivalente de entrada del instrumento; le protege frente a sobrevoltajes y sobre corrientes, y fija el valor máximo del voltaje aplicado al conversor analógico digital (ADC). Puede ser un circuito meramente pasivo mostrado en la figura 3, parte a (izquierda), o puede estar combinado con un amplificador como lo muestra la parte b de la figura (derecha). En ambos casos se emplean redes de resistencias con una relación entre valores bien definida y estable.

**Principio de medición de voltaje AC RMS o eficaz.**

El valor eficaz o Root Mean Square (RMS) de un voltaje alterno es el valor que producirá el mismo efecto de calor en una resistencia que un voltaje o corriente directa de la misma magnitud [6], [7].}



**Figura 4.** Representación gráfica del voltaje RMS de una onda sinusoidal y el valor en corriente directa. [7]

Hay distintos métodos de medir voltaje eficaz de una señal. El más simple es el voltímetro de valor promedio, el que idealmente está construido para señales sinusoidales perfectas. Está basado en un rectificador de onda completa (puente de diodos) el cual no es muy confiable y generalmente es implementado en voltímetros de bajo costo. Por otro lado, se tienen los multímetros True RMS (Valor verdadero eficaz), que medirán el valor efectivo de cualquier forma de onda sin importar distorsiones o armónicos en una señal.

El valor RMS (voltaje o corriente) se puede calcular mediante dos (2) métodos: método analítico y método gráfico los cuales se explican a continuación.

**Método analítico.**

Dada una función periódica continua  $V(t)$ , donde:

$$V(t) = V_{max} \sin(\omega t). \tag{1}$$

El voltaje RMS esta dado por la siguiente relación:

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T V(t)^2 dt}. \tag{2}$$

Reemplazando la ecuación (1) en 2 se tiene:

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T (V_{max} \sin(\omega t))^2 dt.}$$

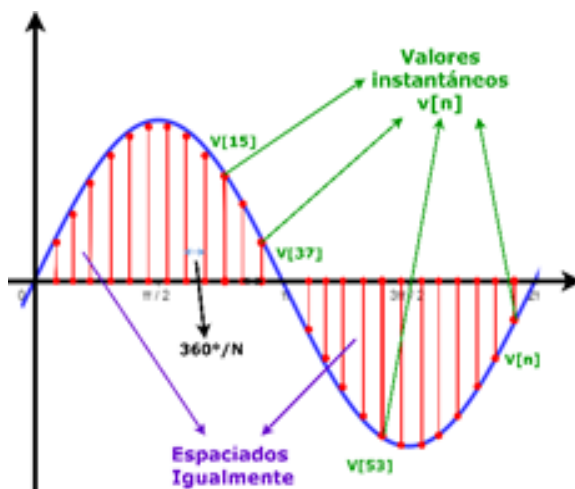
Resolviendo la integral para dicha función (sinusoidal), tomando como periodo  $T=2\pi$ , se tiene el siguiente resultado:

$$V_{rms} = \frac{V_{max}}{\sqrt{2}}. \quad (3)$$

La función  $V(t)$  en uno (1), representa una señal de voltaje en CA, y la relación con tres (3) solo es para una forma de onda sinusoidal.

### Método gráfico.

Cálculo del valor RMS en dominio el dominio de tiempo discreto de una señal sinusoidal mediante el método gráfico.



**Figura 5.** Cálculo del voltaje RMS mediante el método gráfico (muestreo). [7]

Para señales muestreadas en el tiempo, el cálculo del valor RMS está definido como la raíz cuadrada del valor medio de los cuadrados de los valores instantáneos de una cantidad que varía periódicamente, promediado en un ciclo completo. La ecuación de tiempo discreto para calcular el valor RMS es:

Implica elevar al cuadrado la señal, tomar el promedio de la suma de estos cuadrados y obtener la raíz cuadrada del promedio, donde:

$$V_{rms} = \sqrt{\frac{\sum_{n=0}^{N-1} (v[n])^2}{N}}. \quad (4)$$

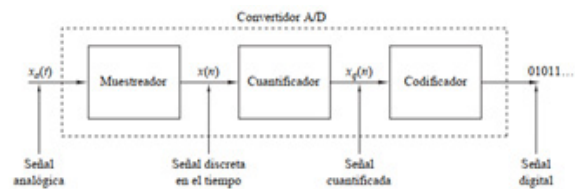
$v[n]$  = muestras de valores instantáneos.

$N$  = muestras de valores instantáneos.

### Parámetros de voltaje mediante el conversor analógico digital (ADC) del microcontrolador Atmega328p.

En los sistemas de censado, Ramírez [8] indica que los conversores analógico-digitales juegan un papel importante. Son el nexo entre las señales de salida de diversos sensores y los elementos digitales de procesamiento de la información.

La conversión analógica digital es un proceso de tres (3) pasos: muestreo, cuantificación y codificación, el cual se presenta en la figura 6.



**Figura 6.** Diagrama general de un convertidor analógico-digital(A/D). [9]

Para la lectura de parámetros de voltaje se utilizará el ADC del microcontrolador Atmega328p. Este periférico maneja un método de conversión de valores analógicos a digitales mediante aproximaciones sucesivas [10]. Este tipo de conversión se encarga de representar

una señal continua en una representación digital binaria usando una búsqueda binaria mediante todos los niveles de cuantización posibles antes de obtener una salida digital a la salida en cada conversión.

Parámetro	Símbolo	Mínimo	Normal	Máximo	Unidad
Resolución	n	-	10		bits
Voltaje de alimentación analógico	AVcc	Vcc - 0.3	-	Vcc + 0.3	V
Voltaje de referencia	Vref	1	-	AVcc	V
Voltaje de entrada	Vin	GND	-	Vref	V
Referencia de voltaje interno	Vint	1	1.1	1.2	V

**Tabla 1.** Características eléctricas ADC Atmega328p. [10]

Internamente el ADC del microcontrolador Atmega328p tiene una resolución de diez (10) bits (tabla 1); esta cifra indica el número de bits que se asigna para cuantificar la entrada [11], es decir asignar un valor digital de diez (10) bits a un valor analógico en la entrada (en este caso voltaje). Para calcular la resolución del ADC, es decir, el tamaño de paso de voltaje por cada cambio de bit menos significativo se utilizará la relación en (5).

$$Resolución = \frac{ADC(Vref)}{2^n - 1} \quad (5)$$

Donde:

$n$  = número de bits para cuantificar la entrada.

$2^n$  = número de combinaciones en el código de salida.

$ADC(Vref)$  = voltaje de referencia del ADC.

$2^n - 1$  = código más grande que puede ser leído.

Para el ADC del microcontrolador Atmega328p se revisó las hojas de especificaciones (tabla 1) y se tiene los siguientes valores:

$$n = 10 \text{ bits.}$$

$$ADC(Vref) = 5[v].$$

$$2^n = 2^{10} = 1024$$

Reemplazando los valores en (5) se tiene:

$$Resolución = \frac{5 [v]}{2^{10} - 1} = 4.8875 \times 10^{-3} [v]$$

$$Resolución = 4.8875 [mv]$$

Este valor de resolución indica que por cada cambio de bit menos significativo en las lecturas hay una variación de 4.8875 (mV) en un rango de número de códigos de 0 (0b0000000000) a 1023 (0b1111111111), que corresponden a un valor mínimo de 0 (v.) y un máximo de 5 (v.) respectivamente [11]. Por lo tanto, el ADC del microcontrolador se encargará de tomar lecturas de niveles de voltaje, provenientes de los circuitos de acondicionamiento en un determinado código;

asimismo, poderlo relacionar con un nivel de voltaje de acuerdo con la resolución del ADC del microcontrolador que se calculó en (5). Para relacionar voltaje a la entrada y código que este representa, se tiene la relación en (6) que será utilizado en el algoritmo de programación del microcontrolador.

$$ADC\_INPUT_{volts} = AnalogRead * \frac{ADC(Vref)}{2^n - 1} \quad (6)$$

Reemplazando valores (tabla 1) en (6) se tiene:

$$ADC\_INPUT_{volts} = AnalogRead * \frac{5}{1023}$$

La relación obtenida en (6) calcula el voltaje en la entrada analógica del microcontrolador, de acuerdo con el código que este representa.

### Obtención de voltaje CD.

Para la obtención de voltaje CD de elevada magnitud se implementará un atenuador básico basado en un divisor de voltaje. Los pasos por realizar se pueden ver en el diagrama de flujo de la figura 3.

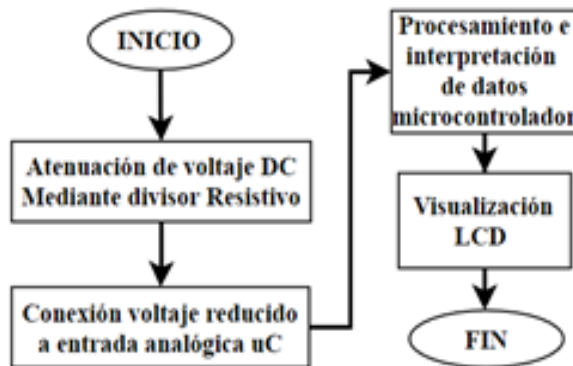


Figura 7. Diagrama de flujo para el censado de voltaje CD. Elaboración propia.

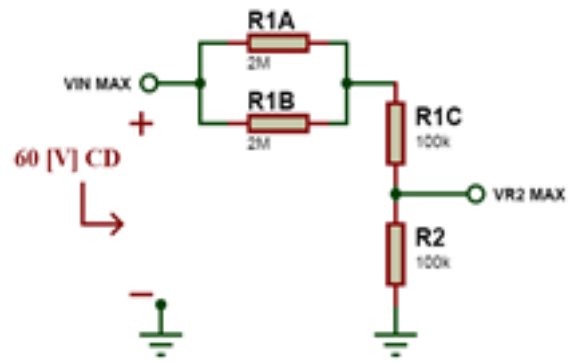


Figura 8. Divisor de voltaje para lectura de voltaje CD. Elaboración propia en software Proteus.

Con base en la configuración de la figura 4 y aplicando análisis de circuitos básico se tiene:

$$V_{R2\ MAX} = \frac{R2}{R1 + R2} * V_{IN\ MAX} \quad (7)$$

$$R1 = \left( \frac{R1A * R1B}{R1A + R1B} \right) + R1C \quad (8)$$

$$V_{IN\ MAX} = \frac{V_{R2\ MAX}}{\frac{R2}{R1 + R2}} \quad (9)$$

$$V_{R2\ MAX} = 5 [v], \quad V_{IN\ MAX} = 60 [v]$$

Reemplazando los valores de R1A, R1B, y R1C en (8) tenemos:

$$R1 = 1.1 [M\Omega]$$

Reemplazando los valores en (9) para hallar R2 se tiene:

$$5 = 60 * \frac{R2}{1.1 [M\Omega] + R2}$$

$$R2 = 100 [k\Omega]$$



El valor que se medirá con el ADC del microcontrolador será el voltaje que cae en la resistencia R2 de la figura 4. Inter-namente el ADC del microcontrolador es nutrido con el voltaje de alimentación de este. Es un valor de referencia a es-cala completa utilizado para una óptima conversión de datos, y según la hoja de especificaciones del microcontrolador (tabla 1) está dado por la relación en (10).

$$ADC(V_{ref}) = V_{cc} \pm 0.3 \quad (10)$$

Los puertos de entrada del ADC tie-nen una tolerancia máxima en su puerto de la misma magnitud que la alimenta-ción del ADC en (10). Por tanto, el volta-je de lectura máxima en CD que puede realizar el sistema es de 60[v] utilizando la relación (7), (9) y (10) y según la figura 4, los datos pueden ser interpretados de la siguiente manera:

$$V_{IN\ MAX} = \text{voltaje CD del banco de baterías.}$$

$$V_{R2\ MAX} = \text{voltaje hacia el sistema microcontrolado.}$$

$$ADC(V_{ref}) = \text{voltaje de referencia del ADC.}$$

$$V_{cc} = \text{voltaje de alimentación del microcontrolador.}$$

Se tabularon los valores resistivos de la figura 7 y se comparó con los valores medidos, con el objetivo de realizar el cen-sado de voltaje DC con mayor preci-sión

Resistencia	Valor esperado	Valor medido	Valores comerciales	Tolerancia
R1	1,1[MΩ]	1,09[MΩ]	R1A=2[MΩ]	± 1%
R2	100[kΩ]	998[kΩ]	R1B=2[MΩ] R1C=100[kΩ] R2= 100[kΩ]	± 5%

**Tabla 2.** Valores de resistencias elegidos. Fuente: Elaboración propia.

Una vez realizado el armado físico del circuito de la figura 4, y así como la pro-gramación del microcontrolador Atme-ga328p aplicando un algoritmo basado en (7) y (9); se detectó inconsistencias en las mediciones, debido principalmente a que la alimentación del microcontro-lador Vcc=5v no es estable. Además, los componentes y/o periféricos que com-parten esta alimentación influyen direc-tamente en la alimentación del voltaje de referencia del ADC, lo cual resulta en lecturas muy variables e inconsistentes. Una solución para este problema es uti-lizar la referencia interna de 1.1 voltios que tiene implementado en su hard-ware el microcontrolador Atmega328p (tabla 1). De esta manera, manipulando registros internos del microcontrolador se puede utilizar esta referencia e imple-mentar una función en nuestro código para obtener el valor de voltaje de ali-mentación del sistema (5 volts); así, en las conversiones necesarias se tendrá valores de conversión más confiables. La función implementada se puede ver en la figura 5.

```

int vRefADC() {
    long result;
    ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
    delay(2);
    ADCSRA |= _BV(ADSC);
    while (bit_is_set(ADCSRA, ADSC));
    result = ADCL;
    result |= ADCH << 8;
    // Calculate Vcc (in mV); 1125300 = 1.1*1023*1000
    result = 1125300L / result;
    return result;
}

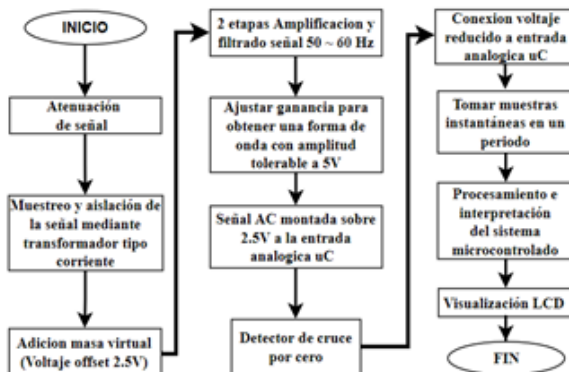
```

**Figura 9.** Función para calcular el voltaje de alimentación del Atmega328P (en miliVolts). [12]

Con esta función los valores fueron bastante cercanos al real, con un porcentaje de error de 0.1% a 0.8% dependiendo de la magnitud de voltaje a la entrada (tabla 3). Para mejorar la lectura de estos valores, se deberán ajustar los valores de las resistencias en la programación del microcontrolador, debido a que las resistencias tienen una tolerancia de un porcentaje de valor máximo y mínimo con respecto a su utilidad comercial.

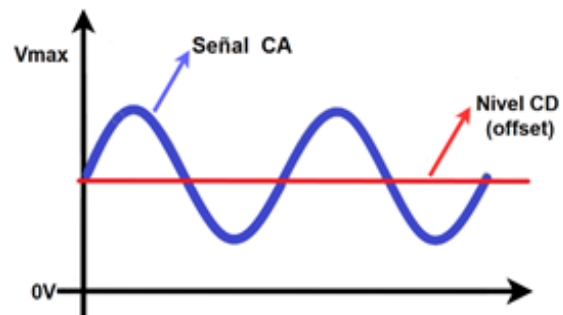
### Obtención de voltaje RMS en CA.

Para la obtención del voltaje en CA se utilizará el método gráfico de la sección del principio de medición de voltaje AC RMS o eficaz. Para esto se desarrolla el diagrama de flujo de la figura 6.



**Figura 10.** Diagrama de flujo para el censado de voltaje AC. Elaboración propia.

Para realizar la medición de voltaje en CA, se implementará el método gráfico visto en la sección ya mencionada en el anterior párrafo. Para aplicar este método en un sistema microcontrolado se debe tomar en cuenta el concepto de voltaje de offset que se ilustra en la figura 10. Es un voltaje en CD que puede ser negativo o positivo; adiciona a la señal alterna un voltaje en CD con el objetivo de cambiar el punto de referencia de 0v. Como resultado tendría un desplazamiento vertical de la señal alterna en un rango de oscilación de la señal solo en voltajes positivos [6]. Esta adición de voltaje de offset es necesario debido a que el microcontrolador Atmega328p tiene una tolerancia de 0v a 5v en CD como máximo en sus entradas (tabla 1). Se elegirá un voltaje de offset de 2.5v como se ilustra en la figura 11 para sobreponer la señal alterna previamente atenuada. Asimismo, que pueda ser muestreada y procesada en el microcontrolador.



**Figura 11.** Desplazamiento de una señal alterna en base a la adición de un voltaje de offset. Elaboración propia con base en [6].

Para el acondicionamiento de señal, se implementó el sensor de voltaje de corriente alterna ZMPT101B (figura 8).



Figura 12. Modulo sensor de voltaje AC. [13]

El sensor ZMPT101B implementa circuitos de acondicionamiento de señal para sistemas de procesamiento de 5v. El sensor tiene un transformador de voltaje de tipo corriente de micro precisión que se encarga de reducir y muestrear la señal de entrada con ayuda de unas resistencias (limitadora y de muestreo) a la entrada y salida del transformador respectivamente.

Posteriormente pasa por dos (2) etapas de un filtro pasa banda activo inversor de ganancia variable. Dicho filtro activo tiene la función de compensar, ajustar y filtrar la señal (a 60 Hz) muestreada en el transformador para tomar pruebas instantáneas en el sistema de procesamiento. Además, añade un voltaje de offset a dicha señal para tomar solo muestras de voltaje positivos. La ventaja de utilizar este módulo es el bajo costo, el tamaño y la calidad de los componentes implementados (condensadores y resistencias). Además, se puede modificar la ganancia total del circuito mediante un potenciómetro de precisión localizado en una etapa intermedia del filtrado de señal.

Para realizar el algoritmo en la programación es necesario realizar un análisis del circuito de acondicionamiento que tiene el sensor para verificar la forma de onda, niveles de amplitud (máximos y mínimos) y adición del voltaje de offset a la salida del sensor. Lo anterior, revisando las hojas de especificaciones del fabricante. El circuito implementado en el sensor es el siguiente:

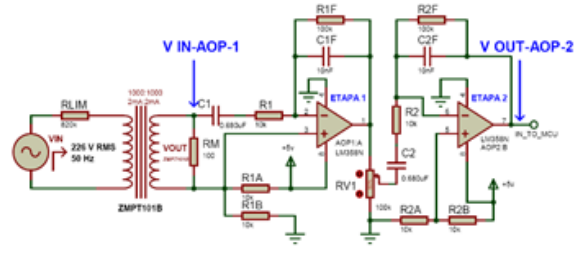


Figura 13. Diagrama esquemático sensor de voltaje ZMPT101B. Software Proteus 8.10. Elaboración propia con base en [14].

A continuación, se analizará el circuito de la figura 9 con el objetivo de obtener la forma de onda en la salida y comparar con el valor real para implementar un algoritmo de programación en el microcontrolador.

**Análisis del transformador de voltaje en el sensor ZMPT101B.**

Una vez verificadas las hojas de especificaciones del transformador implementado en el sensor ZMPT101B, se tiene el circuito de aplicación en la figura 10.

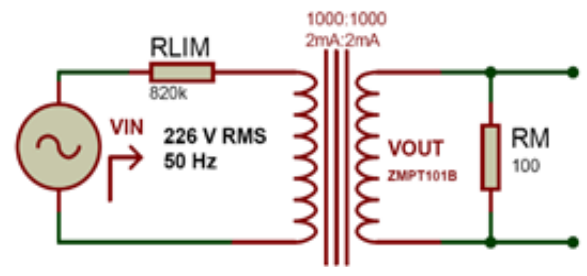


Figura 14. Circuito de aplicación transformador de voltaje de tipo corriente ZMPT101B. Elaboración propia con base en [15].

En la figura 9 y 10 se tiene los siguientes valores:

- $V_{IN}$  = voltaje CA de entrada =  $226[V]_{RMS} \sim 50 [Hz]$ .
- $V_{OUT} = V_{RM}$  = voltaje RMS de salida.
- $V_{OUT\ PICO}$  = voltaje pico de salida.
- $R_{LIM}$  = resistencia limitadora =  $820[\Omega]$ .
- $R_M$  = resistencia de muestreo =  $100[\Omega]$ .

Al aplicar el análisis de circuitos y relaciones de las hojas de especificaciones del transformador ZMPT101B [16] se tiene:

$$R_{LIM} = \frac{V_{IN}}{I_{LIM}} \quad (11)$$

Al despejar (11) para hallar  $I_{LIM}$  y reemplazando los valores se tiene:

$$I_{LIM} = \frac{V_{IN}}{R_{LIM}} = \frac{226}{820K} = 0.2756 \times 10^{-3} [A]$$

$$V_{OUT} = \frac{V_{IN}}{R_{LIM}} * R_M \quad (12)$$

$$V_{OUT} = \frac{V_{IN}}{R_{LIM}} * R_M = V_{OUT} = I_{LIM} * R_M \\ = 0.2756 \times 10^{-3} * 100$$

$$V_{OUT} = 27.56 \times 10^{-3} [V]_{RMS}$$

$$V_{OUT} = V_{RM} = 27.56 \times 10^{-3} [V]_{RMS}$$

$$V_{OUT_{PICO}} = V_{RM_{PICO}} = V_{RM} * \sqrt{2} \quad (13)$$

$$V_{OUT_{PICO}} = V_{RM_{PICO}} = V_{RM} * \sqrt{2} \\ = 27.56 \times 10^{-3} [V] * \sqrt{2}$$

$$V_{OUT_{PICO}} = V_{RM_{PICO}} = 38.9757 \times 10^{-3} [V]$$

### **Cálculo de ganancia de las etapas de amplificación.**

El circuito de la figura 9, posterior al transformador de voltaje previamente analizado, implementa dos (2) etapas de un filtro pasa banda inversor de ganancia variable (que cumple la función de compensación y filtrado de la red eléctrica de CA a 50 Hz o 60 Hz) [17]. Además, el sensor tiene como referencia de 0v un voltaje de offset de 2.5v que se obtiene

$$A_V = -\frac{R_F}{R_i} \quad (14)$$

mediante un divisor de voltaje en cada etapa; la ecuación de ganancia por cada etapa se muestra en (14).

$$A_{VT} = \frac{V_{OUT-AOP-2}}{V_{IN-AOP-1}} \quad (17)$$

$$A_{VT} = \frac{V_{OUT-AOP-2}}{V_{IN-AOP-1}} = 33.333$$

En la figura 9 se tiene los siguientes valores:

$$R_{1F} = R_{2F} = 100k [\Omega], R_1 = R_2 = 10k[\Omega],$$

$$RV_{120\%} = 20k[\Omega].$$

Ganancia de voltaje, primera y segunda etapa:

$$A_{VAOP1} = -\frac{R_{1F}}{R_1} = -\frac{100K}{10K} = -10$$

$$A_{VAOP1} = -10 \rightarrow \text{Ganancia primera etapa}$$

Para la ganancia de la segunda etapa (figura 12) se debe tomar en cuenta el valor del potenciómetro que permitirá ajustar la amplitud de salida del sensor. Está seteado a un 20% de su valor, por lo que se tiene la siguiente relación:

$$A_{VAOP2} = -\frac{R_{2F}}{(R_2 + RV_{120\%})} \quad (15)$$

Reemplazando los valores de resistencias en (15), se tiene:

$$A_{VAOP2} = -\frac{100k}{10k + 20k} = -3.3333$$

$$A_{VAOP2} = -3.333 \rightarrow \text{Ganancia segunda etapa}$$

Ganancia total del circuito

$$A_{VT} = A_{VAOP1} * A_{VAOP2} \quad (16)$$

Reemplazando  $A_{VAOP1}$  y  $A_{VAOP2}$  en (16) se tiene:

$$A_{VT} = (-10) * (-3.333) = 33.333$$

Además, el voltaje de salida en relación con las etapas de los amplificadores operacionales (figura 9) estará dado por la relación entre la salida y la entrada:

$$A_{VT} = \frac{V_{OUT-AOP-2}}{V_{IN-AOP-1}} \quad (17)$$

$$A_{VT} = \frac{V_{OUT-AOP-2}}{V_{IN-AOP-1}} = 33.333$$

### ***Cálculo del voltaje en CA RMS a la salida del sensor ZMPT101B y la forma de onda resultante.***

Tomando como  $V_{IN-AOP-1}$  el voltaje de muestreo en  $V_{RM PICO}$  obtenido en (13) y reemplazando en (17) se tiene:

$$V_{OUT-AOP-2} = 33.333 * 38.9757 \times 10^{-3}$$

$$= 1.299 [v]$$

Aplicando la relación de voltaje pico y RMS de una señal sinusoidal (3):

$$V_{OUT-AOP-2} RMS = 33.333 * \frac{38.9757 \times 10^{-3}}{\sqrt{2}}$$

$$= 0.9185 [V] RMS$$

El valor  $V_{OUT-AOP-2} RMS$  es el voltaje AC RMS a la salida del sensor ZMPT101B (figura 9).

La forma de onda en el sensor se puede observar en la figura 11, y se la puede representar matemáticamente de la siguiente manera:

$$V_{in\ to\ MCU} = V_{OUT-AOP-2} * \sin(2\pi * 50t) + V_{offset} \quad (18)$$

Al reemplazar el valor obtenido en (17), se tiene:

$$V_{in\ to\ MCU} = 1.299 \sin(2\pi * 50t) + 2.5$$

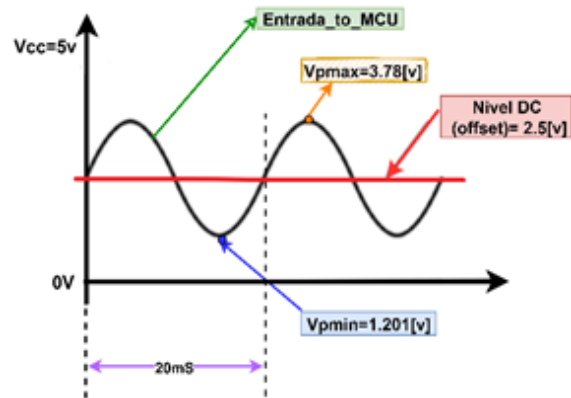


Figura 15. Forma de onda de entrada al microcontrolador usando el módulo de voltaje ZMPT101B. Elaboración propia.

### ***Análisis de los parámetros del filtro.***

Debido a que son dos (2) etapas iguales de un filtro pasa banda [17], se analizará solo una (1) etapa.

#### ***Frecuencia crítica inferior.***

$$f_{c1} = \frac{1}{2\pi * R_3 * C_1} \quad (19)$$

$$f_{c1} = \frac{1}{2\pi * 10k * 0.680\mu F}$$

$$f_{c1} = 23.40Hz$$

### Frecuencia crítica superior.

$$f_{c2} = \frac{1}{2\pi * R_6 * C_2} \quad (20)$$

$$f_{c2} = \frac{1}{2\pi * 100K * 10nF}$$

$$f_{c2} = 159.1549$$

### Frecuencia central.

$$f_0 = \sqrt{f_{c1} * f_{c2}} \quad (21)$$

$$f_0 = \sqrt{23.40Hz * 159.1549}$$

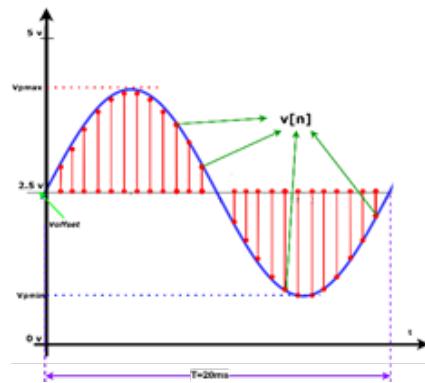
$$f_0 = 61.02Hz$$

El filtro pasa banda solo permitirá pasar la frecuencia de la red eléctrica, en Bolivia la constante de red a nivel nacional es de 50 Hz. Este filtro tiene una banda de frecuencia crítica inferior de 23.40 Hz (20) y una continuidad crítica superior de 159,16 Hz (21), por lo que puede ser implementada en sistemas de 50 o 60 Hz dependiendo al país.

### Método para calcular el valor RMS de la señal con el sensor ZMPT101B.

Para el cálculo del voltaje RMS en CA, se utilizará el método gráfico en el dominio de tiempo discreto de una señal sinusoidal. Consiste en tomar valores instantáneos en un periodo completo. Para ello, mediante algoritmos de programación y manejo de temporizadores en el

microcontrolador, se configurará un detector de cruce por cero, de tal manera que las muestras sean tomadas durante veinte (20) milisegundos (tiempo que tarda la señal en completar un ciclo completo) y se tenga un valor aproximado más cercano al medido con un multímetro comercial. Como se mencionó previamente, el valor RMS está definido como la raíz cuadrada del valor medio de los cuadrados de los valores instantáneos; la ecuación (4) muestra la relación de tiempo discreto para calcular el valor RMS de una señal, la cual la se relaciona con la figura 15.



**Figura 16.** Análisis de la gráfica para calcular RMS de la señal. [18]

Se implementará el algoritmo de programación en el microcontrolador de este método, aplicando un factor de corrección (22), el cual relacionará la entrada de la red eléctrica con la salida del sensor de voltaje ZMPT101B.

$$F_c = \frac{\text{Voltaje RMS red electrica}}{\text{Voltaje RMS salida ZMPT101B}} \quad (22)$$

Al tomar en cuenta los valores calculados y calculados (tabla 4) se reemplaza en (22) y se tiene:

$$F_c = \frac{226 [V]_{RMS}}{0.9185 [V]_{RMS}} = 246.05$$

Este factor de corrección puede variar por la posición donde se configura el potenciómetro (15) que varía la ganancia del circuito de la figura 9; también puede variar dependiendo a la hora del día; se puede ver los valores medidos en diferentes horas del día (tabla 4). Por tanto, este factor de corrección se implementará en el código de programación del microcontrolador. Asimismo, aplicando algunas relaciones dará el valor aproximado de la red eléctrica.

$$\text{Voltaje RMS red eléctrica} = 226 \text{ [V]RMS.}$$

$$\text{Voltaje RMS salida ZMPT101B medido} = 0.88 \text{ [V]RMS}$$

### Implementar la plataforma ThingSpeak como servidor.

ThingSpeak es un servicio en la nube que funciona como una plataforma de IoT. Está basado en canales (públicos y privados) para recuperar y almacenar datos de diferentes dispositivos usando llamadas API REST (get, post, update and delete). Estas solicitudes son gestionadas a través del protocolo HTTP usando un modelo de solicitud-respuesta [19].

Las web API son interfaces de desarrollo que permiten el intercambio de información entre un servicio web y una aplicación sin tener que saber cómo están implementadas. Usualmente el intercambio se realiza a través de peticiones HTTP o HTTPS, por medio de formatos XML o JSON.

Para la monitorización remota se utilizará la plataforma IoT ThingSpeak de manera privada; servirá para almacenar y consultar los datos provenientes del sistema de procesamiento. Además, permite la creación de canales para visualizar los datos de determinados campos. Un canal gratuito puede implementar hasta cuatro (4) canales; un intervalo

de publicación de datos (también llamados mensajes) mínimo de 15 segundos; una capacidad de almacenar tres (3) millones de mensajes como límite anual, y el envío de mensajes no puede exceder un tamaño de tres (3) megabytes [20]. El proceso para la publicación de datos en ThingSpeak se puede ver en la figura 13.

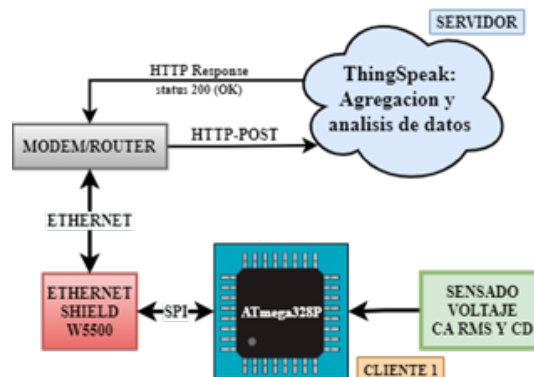
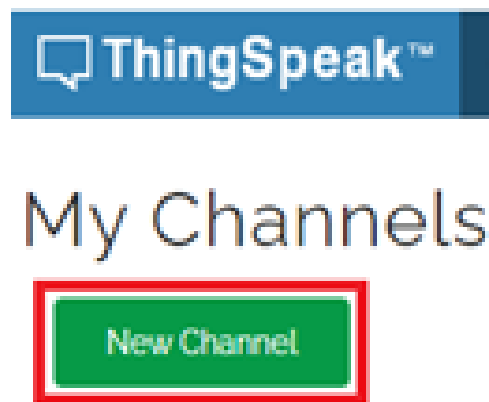


Figura 17. Diagrama de bloques para publicar datos en la plataforma ThingSpeak. Elaboración propia.

Se debe crear una cuenta gratuita en la plataforma y posteriormente crear canales para el almacenamiento de datos de voltaje como se puede ver en la figura 14.



### New Channel

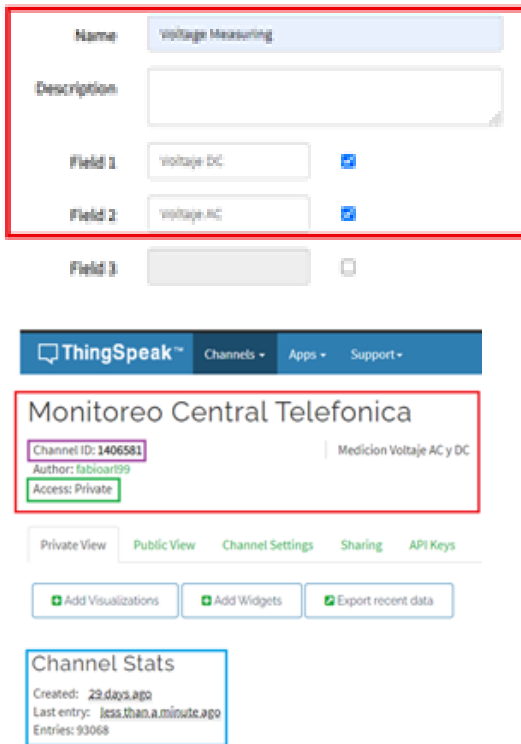


Figura 18. Creación canal ThingSpeak privado con dos (2) campos para el voltaje AC y voltaje DC. Elaboración propia.

### Implementación de modulo ethernet W5500 para el envío de datos mediante una conexión a internet.

Los datos se envían a la plataforma de ThingSpeak por internet, mediante el módulo ethernet W5500 el cual integra el conjunto de protocolos TCP/IP, 10/100 ethernet MAC (capa de enlace de datos) y PHY (capa física). El módulo implementa aplicaciones ethernet usando un programa de socket simple, conexión de dos (2) nodos en una red para comunicarse entre sí, e integra Serial Peripheral Interface (SPI) para una comunicación con un microcontrolador externo [23]. En la figura 13 se puede ver la función que tiene este módulo al momento de publicar datos en la plataforma, y en la figura 15 se puede ver los pines que utiliza SPI.



Figura 19. Módulo de comunicación ethernet w5500. [22]

### Recuperación de datos desde la plataforma ThingSpeak.

Para recuperar los datos del servidor de ThingSpeak se tienen dos (2) opciones: verificar desde la plataforma web de ThingSpeak o recuperar los datos desde una aplicación Android, implementando las API requests que facilita ThingSpeak. Los datos recuperados o leídos están en un formato JSON, un modelo ligero de intercambio de datos que resulta sencillo de interpretar para las máquinas y programadores [23]. En la figura 16 se puede ver el diagrama de bloques para la recuperación de datos.

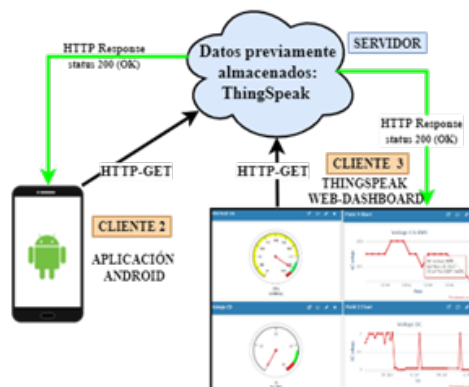


Figura 20. Diagrama de bloques para consultar datos de la plataforma ThingSpeak. Elaboración propia.



En el presente artículo, el objetivo es recuperar únicamente los últimos valores añadidos a los canales de la plataforma ThingSpeak, junto con la fecha y hora de actualización, es decir, recuperar el último valor de voltaje CA (canal 1), el último de voltaje CD (canal 2), junto con la fecha y hora respectiva. ThingSpeak facilita una interfaz de visualización web (figura 23) y consultas API (Requests), haciendo el uso de llaves (API-Keys) tanto para lectura como escritura en los canales. Las API requests de lectura son consultas HTTP (Método Get), mediante un localizador uniforme de recursos (URL) que tienen diferentes parámetros. Por último, a través de la configuración que se tenga en esta URL se pueden obtener valores específicos de los canales de ThingSpeak en un formato de texto JSON. Este servirá para implementar la aplicación Android en el entorno de desarrollo MIT App Inventor [24]. Aquí la URL que se configuró imprime los valores de los canales de interés (figura 17).



Figura 21. Lectura de campos específicos de los canales en formato JSON. Elaboración propia.

### Desarrollo de aplicación Android en el entorno MIT App Inventor.

App Inventor es un entorno de desarrollo intuitivo de programación para dispositivos móviles con un sistema operativo Android e IOS. Está basado en bloques que facilitan la creación de aplicaciones en menos tiempo que haciendo uso de entornos de desarrollo tradicionales [24].

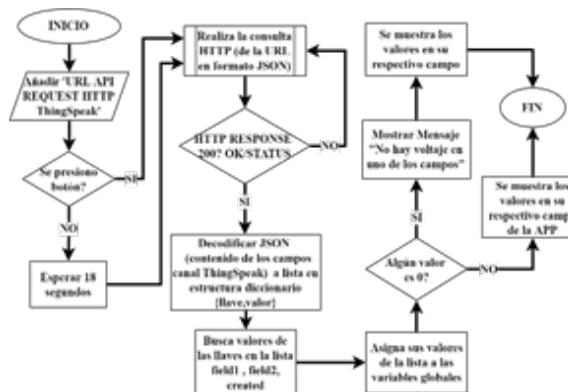


Figura 22. Diagrama de flujo para el desarrollo de la aplicación Android. Elaboración propia.

Se implementó una aplicación Android con base en el diagrama de flujo de la figura 18, con el objetivo de obtener los últimos parámetros censados y la última hora de actualización. En caso de que el sistema de monitoreo deje de actualizar sus datos, la aplicación mostrará una alerta. Esta última refresca los datos cada 18 segundos, o se puede solicitar los últimos datos leídos instantáneamente mediante un botón. La interfaz de usuario de la aplicación Android se puede ver en la figura 19.

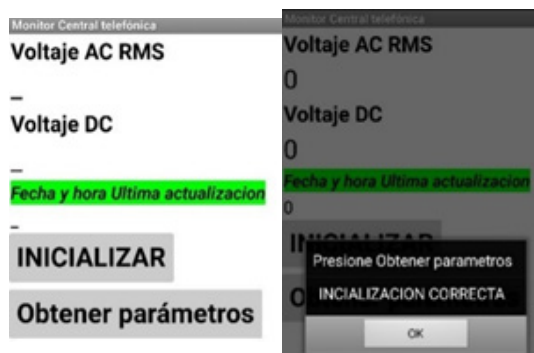


Figura 23. Interfaz de usuario de aplicación Android desarrollada en el entorno MIT App Inventor. Elaboración propia.

### Implementación del sistema completo.

En la figura 20 se presenta el diagrama de conexión del sistema completo, además de los componentes utilizados. En la



Se realizó las respectivas mediciones con dos (2) voltímetros, para corroborar los datos censados por el sistema de procesamiento que se los presenta en un despliegue, como se muestra en la figura 25.



**Figura 26.** Monitoreo de voltaje de la red eléctrica 226 v. RMS en CA y una fuente de CD de 12,4 v. Elaboración propia.

### Implementación de tablero (DashBoard) en la plataforma ThingSpeak y su interfaz de Visualización WEB.

En la plataforma web de ThingSpeak se tienen elementos de interacción en una interfaz gráfica de usuario (Widgets), que se configuran de una manera muy intuitiva y se puede dar un aspecto visual entendible y amigable con el usuario (figura 23). También se pueden desplazar los datos o incluso un promedio de estos.



**Figura 27.** Visualización de widgets implementados en el tablero de ThingSpeak. Elaboración propia.

### Resultados de la aplicación Android desarrollada para la recuperación de datos.

Al realizar pruebas de corte de alimentación en el voltaje en CD se pudo verificar el funcionamiento de las alarmas en la aplicación Android (figura 24).



**Figura 28.** Recuperación de datos en la aplicación desarrollada, teniendo en cuenta que se desconectó el voltaje DC para probar el mensaje de advertencia de caída de voltaje en un campo. Elaboración propia.

## Implementación dispositivo de procesamiento, periféricos externos, sensores, dispositivos de visualización y fuente de alimentación localmente.



**Figura 29.** Implementación en circuito impreso del sistema de procesamiento, fuente de alimentación y dispositivo de visualización local. Elaboración propia.

Para reducir el cableado de las entradas y salidas del sistema, facilitar la alimentación eléctrica, conexión a los dispositivos de visualización locales y la conexión a un modem mediante un cable de red ethernet; se desarrolló una placa de circuito impreso (PCB). Posteriormente se adecuó el sistema de la figura 25 en una caja de metal que se puede instalar de forma local en la central telefónica (figura 26).



**Figura 30.** Caja contenedora del sistema de monitoreo. Elaboración propia.

## Conclusiones

Se implementó el sistema según los requerimientos planteados para un monitoreo de voltaje en corriente alterna de 230[V] RMS y corriente continua de 60[V]. El sistema puede realizar lecturas bastante cercanas a los valores reales tanto de voltaje en corriente continua como en alterna. Puede detectar caídas de voltaje o detección de corte de energía en tiempo real y visualizarlos localmente en un despliegue, remotamente en la plataforma de IoT ThingSpeak o en una aplicación Android, implementando una arquitectura cliente-servidor, por medio de consultas HTTP. Aquí se utilizan los métodos Get (para la obtención de datos) y Post (para el almacenamiento de datos) en la base de datos de ThingSpeak.

Para el desarrollo de la aplicación Android se utilizó la API REST que proporciona ThingSpeak. Mediante la consulta HTTP-Get se pudo obtener los valores de voltaje almacenados en la base de datos en un formato JSON. Los parámetros de interés a recuperar son los valores de voltaje censados, la fecha y hora del último valor agregado en la base de datos para su posterior manipulación en la aplicación Android; así como también la generación de alarmas en la aplicación dependiendo a umbrales de voltaje (máximos y mínimos) permitidos en la central telefónica (figura 24).

Se pudo optimizar el sistema a prueba de errores utilizando el periférico Watchdog Timer del microcontrolador Atmega 328p. Tiene la función de resetear el microcontrolador en caso de fallas por circunstancias inesperadas como alimentación del microcontrolador insuficiente, errores de comunicación con sensores, u otros. Este periférico incorpora un oscilador interno separado del oscilador principal del sistema, el cual puede ser

configurado para que pasado un determinado tiempo el microcontrolador se reinicie por completo. Por tanto, se debe realizar un control de este temporizador reiniciándolo constantemente cada vez que se almacene satisfactoriamente en la base de datos de ThingSpeak.

El análisis de circuitos desarrollado para el sensor de voltaje ZMPT101B en la sección "Obtención de voltaje RMS en CA" fue necesario debido a que no se tiene una hoja de especificaciones de un único fabricante que sea lo suficientemente detallada y que explique la configuración de los circuitos implementados en el sensor. Además, esta investigación permitió poner en marcha el algoritmo de programación del microcontrolador y obtener valores aceptables en comparación a un multímetro comercial de categoría 3 (figura 22).

### Agradecimientos

Un agradecimiento al Ing. Albero Pérez por plantar la idea para el presente proyecto, así como al Ing. Edgar Ramos, quien estuvo pendiente de las correcciones del sistema.

### Referencias

- [1] G. Álvarez, *Estudio de sistemas de respaldo de energía eléctrica para cuarto de telecomunicaciones en la finca limoncito*, Trabajo de grado, Facultad de Educación Técnica para el Desarrollo, Universidad Católica de Santiago de Guayaquil, Guayaquil, Ecuador, 2012. [Internet]. Disponible en <http://repositorio.ucsg.edu.ec/handle/3317/8547>
- [2] M. Anand, *Electronic instruments, and instrumentation technology*. Nueva Deli, India: PHI Learning Private Limited, 2004.
- [3] T. L. Floyd, *Principios de circuitos eléctricos*, 8ª ed., México: Pearson Educación, 2007. [Internet]. Disponible en [http://media.espora.org/mgoblin\\_media/media\\_entries/1455/Principios\\_de\\_circuitos\\_electricos.pdf](http://media.espora.org/mgoblin_media/media_entries/1455/Principios_de_circuitos_electricos.pdf)
- [4] C. K. Alexander and M. N. Sadiku, *Fundamentos de circuitos eléctricos*, México, D.F.: McGraw-Hill, 2013. [Internet]. Disponible en [https://www.academia.edu/42934332/Fundamentos\\_de\\_circuitos\\_electricos\\_5ed\\_Charles\\_K\\_Alexander\\_Espanhol](https://www.academia.edu/42934332/Fundamentos_de_circuitos_electricos_5ed_Charles_K_Alexander_Espanhol)
- [5] R. Pallás, *Instrumentos electrónicos básicos*, 1ª ed., Barcelona: Marcombo S.A, 2006.
- [6] M. Tooley, *Electronic circuits: Fundamentals and Applications*, 3rd ed., Oxford: Elsevier, 2006.
- [7] Electronic Tutorials. (2013, Jun. 25). "RMS Voltage of a Sinusoidal AC Waveform." [Internet]. Disponible en <https://www.electronics-tutorials.ws/accircuits/rms-voltage.html>
- [8] L. Corona, G. Abarca y J. Mares Ramírez, *Sensores y actuadores. Aplicaciones con Arduino*, Azcapotzalco, México, D. F: Grupo Editorial Patria, S. A., 2014. [Internet]. Disponible en [http://www.eet476.edu.ar/documentos/Aula%20Virtual/ABBONIZIO-6TM%20\(2\).pdf](http://www.eet476.edu.ar/documentos/Aula%20Virtual/ABBONIZIO-6TM%20(2).pdf)
- [9] E. Soria, M. Martínez, J. Francés et al., *Tratamiento digital de señales: problemas y ejercicios resueltos*. Madrid: Prentice-Hall, 2003.
- [10] *8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*, Atmel, San José, California, 2015, pp. 294. [Internet]. Disponible en [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)
- [11] A. Kay and T. Green, eds., *Analog Engineer's Pocket Reference*, Dallas, Texas: Texas Instruments, 2015. [Internet]. Disponible en [https://www.academia.edu/24484011/Analog\\_Engineer\\_s\\_Pocket\\_Reference\\_Guide](https://www.academia.edu/24484011/Analog_Engineer_s_Pocket_Reference_Guide)

- [12] Scott. (2012, Jul. 9). "Secret Arduino Voltmeter – Measure Battery Voltage." [Internet]. Disponible en <https://providyourown.com/2012/secret-arduino-voltmeter-measure-battery-voltage/>
- [13] Techtonics. (2019). "ZMPT101B AC Single Phase Voltage Transformer Sensor Relay". [Internet]. Disponible en <https://www.techtonics.in/zmpt101b-ac-single-phase-voltage-transformer-sensor-relay>
- [14] M. Mnati, A. Van den Bossche, and R. Chisab. (2017, apr.). "A Smart Voltage and Current Monitoring System for Three Phase Inverters Using an Android Smartphone Application," *Sensors*. [Internet]. Vol. 17, n.º 4, p.872. Disponible en <https://doi.org/10.3390/s17040872>
- [15] InnovatorsGuru.(2014)."ZMPT101B Micro Precision Voltage Transformer". [Internet]. Disponible en <https://datasheetspdf.com/pdf/1031464/ETC/ZMPT101B/1>
- [16] InnovatorsGuru. (s.f.). "ZMPT101B". [Internet]. Disponible en <https://innovatorsguru.com/zmpt101b/>
- [17] ElectronicsTutorials. (s.f.). "Active Band Pass Filter" - Op-amp Band Pass Filter." [Internet]. Disponible en [https://www.electronics-tutorials.ws/filter/filter\\_7.html](https://www.electronics-tutorials.ws/filter/filter_7.html)
- [18] I. Abubakar, S. N. Khalid, M. W. Mustafa, H. Shareef, and M. Mustapha. (2017). "Calibration of ZMPT101B voltage sensor module using polynomial regression for accurate load monitoring", *ARPJ Journal of Engineering and Applied Sciences*. [Internet]. Vol. 12, n.º 4, pp. 1076-1084. Disponible en [https://www.researchgate.net/publication/316636029\\_Calibration\\_of\\_ZMPT101B\\_voltage\\_sensor\\_module\\_using\\_polynomial\\_regression\\_for\\_accurate\\_load\\_monitoring](https://www.researchgate.net/publication/316636029_Calibration_of_ZMPT101B_voltage_sensor_module_using_polynomial_regression_for_accurate_load_monitoring)
- [19] MathWorks. (s.f.). "REST API". [Internet]. Disponible en <https://la.mathworks.com/help/thingspeak/rest-api.html>
- [20] ThingSpeak. (s.f.). "ThingSpeaks Licensing FAQ". [Internet]. Disponible en [https://thingspeak.com/pages/license\\_faq](https://thingspeak.com/pages/license_faq)
- [21] Wiznet. (s.f.). "W5500". [Internet]. Disponible en <https://www.wiznet.io/product-item/w5500/>
- [22] Leorick. (2018, Jul. 6). "w5500 Ethernet Module with Arduino Mega 2560". [Internet]. Disponible en <https://leoricksimon.blogspot.com/2018/06/w5500-ethernet-module-with-arduino-mega.html>
- [23] JSON. (s.f.). "Introducing JSON." [Internet]. Disponible en <https://www.json.org/json-en.html>
- [24] Appinventor. (s.f.). "About Us". [Internet]. Disponible en <https://appinventor.mit.edu/about-us>