



RESUMEN ANALÍTICO DE INVESTIGACIÓN -RAI-

Estudio y Diseño de Módulo de Transmisión Autónoma de Lectura de Contadores de Energía Eléctrica Integrado a IoT*

OTAVO, Yeison, ROJAS, Javier.

PALABRAS CLAVE

Energía eléctrica; Energía eléctrica; Transmisión de datos; Base de datos.

DESCRIPCIÓN

El proyecto tiene como finalidad diseñar e implementar un módulo encargado de tomar lecturas de consumo eléctrico de energía, las cuales son transmitidas haciendo uso de las tecnologías móviles de acceso a internet actuales (GSM e IoT) a un servidor web en el cual podrán ser consultadas por el cliente haciendo uso de un Login de ingreso a las lecturas (Usuario y Contraseña). Su funcionamiento se basa el uso de una tarjeta microcontroladora la cual esta unificada a unos dispositivos de medición de tensión y amperaje los cuales están previamente calibrados haciendo una medición comparativa de estos mismos valores en la red eléctrica en actual operación. Los datos son finalmente enviados al servidor alojado en la web haciendo uso de un módulo de conexión GSM asociado a una tarjeta SIM previamente registrada con cualquiera de los operadores móviles en operación en el país.

FUENTES

Se consultaron un total de 15 referencias bibliográficas distribuidas así: sobre el tema alusivo a conceptos eléctricos 2 libros, 1 artículo y 1 tesis (Universidad Nacional de Colombia); sobre dispositivos de medición y transmisión 3 libros, 3 artículos; sobre desarrollo web y bases de datos 5 libros.

CONTENIDO

Dentro del presente trabajo se desglosa la realización del trabajo de investigación dirigido (TID), el cual se comprende de seis capítulos. En el capítulo 1(Introducción) se explica de forma breve los conceptos asociados a corriente eléctrica alterna, voltaje y corriente eficaces y

potencia aparente, fundamentos en desarrollo de bases de datos y programación Web fundamentales para el entendimiento del desarrollo del proyecto. En el capítulo 2(Diseño e implementación) se seleccionan los dispositivos de medición y transmisión de la información de acuerdo con los parámetros que se determinan aptos para su escogencia e integración en el proyecto (especificaciones técnicas y método de operación. En el capítulo 3(Resultados) se describen las pruebas de funcionamiento, así como las calibraciones realizadas e informes con los márgenes de error en las lecturas obtenidas a través de la toma de valores comparativos con instrumentos de medición de referencia (voltímetro, pinza amperimétrica); dentro del capítulo 4(Diseño Físico y Costos) se desglosan los costos asociados al diseño y creación del prototipo, así como el diseño e impresión de la tarjeta PCB, el proceso de diseño de la página web, la adquisición de servicios de hosting y la visualización del prototipo final del proyecto; para el capítulo 5 (Conclusiones) se plasma la información referente al cumplimiento de los objetivos del proyecto, resultado final y procesos de mejora a plantear y por último en el capítulo 6 (Bibliografía) se referencian las fuentes consultadas para la realización del trabajo investigativo.

METODOLOGÍA

El trabajo investigativo se genero haciendo uso de una metodología de desarrollo por fases, la cual se conforma de una serie de etapas, de acuerdo con una planificación previa con el fin de llevar un control de la ejecución de la investigación y el cumplimiento de los objetivos propuestos durante la radicación de la propuesta investigativa. Este proyecto se desarrolló

*Trabajo de Investigación Dirigido (TID)

teniendo en cuenta el siguiente orden de implementación:

- Fase I Estado del arte
- Fase II Análisis y selección de dispositivos
- Fase III Pruebas básicas de funcionamiento de los dispositivos
- Fase IV Generación del código de conversión Análogo-Digital
- Fase V Configuración del servidor.
- Fase VI Pruebas de conectividad y envío de datos al servidor.

CONCLUSIONES

La contribución principal del siguiente trabajo tenía como pilar principal el diseño y realización de un prototipo con la capacidad de obtener mediciones de amperaje y tensión eléctrica que permitieran la obtención de una potencia aparente que permitiera el modelamiento de un esquema de obtención de consumo eléctrico de energía en tiempo real que fuera transmitido a través de la infraestructura de red móvil actual, mediante tecnología GSM a un servidor dedicado al almacenamiento, procesamiento y visualización de la información obtenida del prototipo.

Dentro del presente trabajo investigativo se corrobora la toma de las mediciones haciendo uso de los conceptos de corriente y tensión eléctrica alterna para las tareas de calibración de los sensores utilizados, gracias al uso de cálculos teóricos que permitieron el diseño eléctrico requerido para el funcionamiento del prototipo y la reducción del margen de error en comparación con los instrumentos de medición convencionales (multímetro, pinza amperimétrica).

Se evidenció la toma correcta de los datos y su transmisión a través de la red móvil GSM mediante el uso del monitor serial a través del Arduino, así como el protocolo de comunicación AT y los servicios de bases de datos implementados en el diseño del servidor dedicado, los cuales fueron posibles mediante el uso de los conceptos y aplicación de las estructuras PHP, en integración con MySQL

La visualización de la página web integrada a la base de datos fue posible a través del manejo de conceptos de programación y desarrollo fundamentales para su funcionamiento como Javascript, CSS, librerías interactivas como Plotly, las cuales permiten que el usuario final interactúe de forma más sencilla con los datos obtenidos del prototipo.

El diseño de la tarjeta PCB requerida cumple con lo planificado dentro de los esquemáticos del diseño y funcionamiento de esta al integrar los circuitos y componentes requeridos dentro del espacio contemplado para su integración con la placa Arduino, así mismo los sensores de medición utilizados son acordes con las especificaciones técnicas y de operatividad informadas por el fabricante lo que dio buen término a los valores censados.

De la misma manera la estructura implementada en el diseño y desarrollo del presente proyecto se puede abarcar en buena medida en otros campos de la industria como el sector médico a través de tele medición, los sistemas de control SCADA, así como en el ámbito de agricultura donde se cubren grandes distancias, todo esto es posible con la estructura principal del presente trabajo.

ANEXOS

No se anexa información externa al trabajo final.

**ESTUDIO Y DISEÑO DE MÓDULO DE TRANSMISIÓN AUTÓNOMA DE
LECTURA DE CONTADORES DE ENERGÍA ELÉCTRICA INTEGRADO**

A IOT

JAVIER ROJAS CÁRDENAS

YEISON OTAVO VARGAS

AUTORES

CORPORACIÓN UNIVERSITARIA UNITEC

ESCUELA DE INGENIERÍA

INGENIERÍA DE TELECOMUNICACIONES

BOGOTÁ, D.C., NOVIEMBRE 2019

**ESTUDIO Y DISEÑO DE MÓDULO DE TRANSMISIÓN AUTÓNOMA DE
LECTURA DE CONTADORES DE ENERGÍA ELÉCTRICA INTEGRADO
A IOT**

JUAN FELIPE GUTIÉRREZ GÓMEZ

DIRECTOR

CORPORACIÓN UNIVERSITARIA UNITEC

ESCUELA DE INGENIERÍA

INGENIERÍA DE TELECOMUNICACIONES

BOGOTÁ, D.C., NOVIEMBRE 2019

Dedicatoria

*Este trabajo es dedicado a Dios quien nos
ha dado la vida, el pensamiento, el
razonamiento para el desarrollo de este,
después a nuestros padres y familia
quienes son nuestra motivación día a día y
finalmente a nuestros docentes quienes
nos transmitieron todo su conocimiento y
experiencia a lo largo de nuestra carrera.*

CONTENIDO

RESUMEN	6
ABSTRACT	6
1 CAPÍTULO – INTRODUCCION	7
1.1 Corriente eléctrica.....	7
1.2 Corriente continua (DC).....	7
1.3 Corriente alterna (AC).....	8
1.4 Valor eficaz.....	11
1.5 Tensión	12
1.7 Sensor de corriente.....	14
1.8 Sensor de voltaje	15
1.9 Microcontrolador.....	16
1.10 Comunicación móvil.....	17
1.11 SIM Card.....	19
1.12 GSM.....	20
1.13 GPRS	20
1.14 Capas de modelo OSI.....	21
1.15 MySQL.....	24
1.16 PHP	26
1.17 HTML.....	27
1.18 CSS	28
1.19 Javascript.....	29
2 CAPITULO DISEÑO E IMPLEMENTACIÓN.....	31
2.1 Arduino.....	31
2.2 Sensor de amperaje.....	32
2.3 Módulo de transmisión de datos SIM800L	36
2.4 Programación en Arduino.....	36
2.5 Comunicación con el módulo Sim800.....	39
2.6 Configuración información recibida del microcontrolador.....	41
2.7 Visualización de la información.....	47
3 RESULTADOS.....	53

3.1	Calibración del sensor de corriente SCT-013.....	53
3.2	Calibración del sensor de voltaje ZMPT101B	55
3.3	Validación de datos capturados VS la medición con multímetro.....	56
3.4	Transmisión de los datos obtenidos a través de la red GSM.....	57
4	DISEÑO FÍSICO Y COSTOS	61
4.1	Diseño.....	61
4.2	Costos	66
5	CONCLUSIONES	67
6	CAPITULO BIBLIOGRAFIA.....	70

RESUMEN

El proyecto tiene como finalidad diseñar e implementar un módulo encargado de tomar lecturas de consumo eléctrico de energía, las cuales son transmitidas haciendo uso de las tecnologías móviles de acceso a internet actuales (GSM e IoT) a un servidor web en el cual podrán ser consultadas por el cliente haciendo uso de un Login de ingreso a las lecturas (Usuario y Contraseña). Su funcionamiento se basa el uso de una tarjeta microcontroladora la cual esta unificada a unos dispositivos de medición de tensión y amperaje los cuales están previamente calibrados haciendo una medición comparativa de estos mismos valores en la red eléctrica en actual operación. Los datos son finalmente enviados al servidor alojado en la web haciendo uso de un módulo de conexión GSM asociado a una tarjeta SIM previamente registrada con cualquiera de los operadores móviles en operación en el país.

ABSTRACT

The project has as an objective to design and implement a module responsible for taking readings of electricity consumption, which are transmitted using the current mobile internet access technologies (GSM and IoT) to a web server and it can be consulted by the client using a Login to read the readings (Username and Password). Its operation is based on the use of a microcontroller card which is unified to some voltage and amperage measuring devices which are previously calibrated by making a comparative measurement of these same values in the current mains. The data is finally sent to the server hosted on the web using a GSM connection module associated with a SIM card previously registered with any of the mobile operators in operation in the country.

1 CAPÍTULO – INTRODUCCION

1.1 Corriente eléctrica

De todas las formas de energía que se utilizan hoy en día, la energía eléctrica es la más versátil, cómoda y limpia en su consumo, aunque no en su producción. La corriente eléctrica es fácil de manejar y se puede transportar de un lugar a otro fácilmente por medio de conductores eléctricos. No obstante, si en algo destaca la energía eléctrica, respecto a otras, es en la facilidad con que se transforma. Existen múltiples ejemplos de la transformación de energía eléctrica en la vida cotidiana, entre las que destacan:

- Transformación en energía calorífica a través de un radiador eléctrico.
- Transformación en luz (radiación) a través de una lámpara.
- Transformación en energía mecánica por medio de los motores.
- Transformación en ondas sonoras por medio de los altavoces.

1.2 Corriente continua (DC)

Es aquella cuyos valores instantáneos a lo largo del tiempo son de la misma magnitud. Suele estar suministrado por pilas, baterías, dinamos, fuentes de alimentación de corriente continua etc.

Una de las características fundamentales de la corriente continua es que tiene polaridad: Uno de los conductores es el positivo (de color rojo) y el otro el negativo (de color negro), también llamado masa. Esto implica que los receptores deben conectarse de acuerdo con esa polaridad, de lo contrario se podrían obtener consecuencias no deseadas, y en el mejor de los casos no funcionarían.

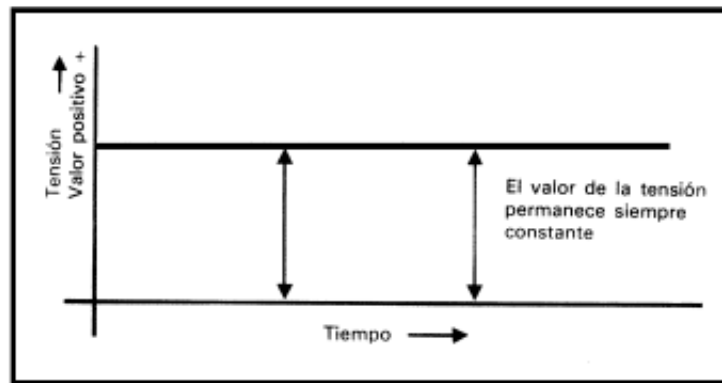


Figura 1 Gráfica de corriente continua.

1.3 Corriente alterna (AC)

Una corriente que cambie de sentido a intervalos de tiempo recibe el nombre de corriente alterna. La corriente que se tiene en las bases de enchufe de una casa doméstica se denomina corriente alterna senoidal. La forma de la onda senoidal es periódica, ya que se reproduce idénticamente en intervalos de tiempo iguales.

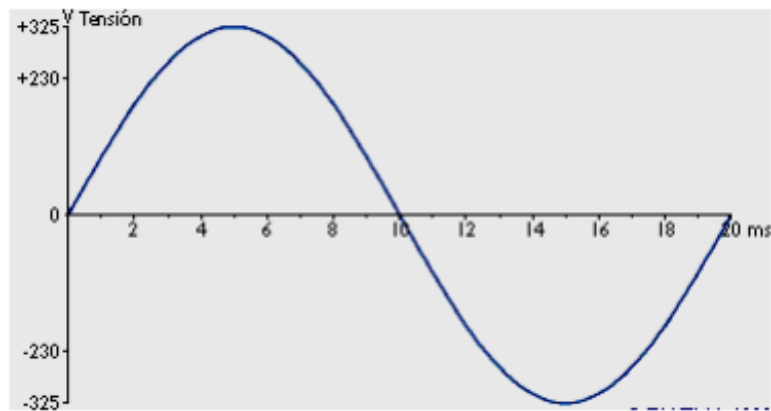


Figura 2 Grafica de corriente alterna.

Dentro de una corriente alterna senoidal se consideran los siguientes parámetros fundamentales:

- **Frecuencia** número de veces que la señal alterna se repite en un segundo. La unidad de frecuencia es el hertzio (Hz), que equivale a un ciclo por segundo (c.p.s). Se representa por la letra f . En toda Europa la frecuencia de la corriente eléctrica de la red de alimentación a viviendas e industrias es de 50 Hz. En América esta frecuencia es de 60 Hz.

- **Período** es el tiempo necesario para que una señal alterna se repita. El período se mide en segundos y se representa por la letra T . El período y frecuencia son dos cantidades inversas ya que, si en un segundo se repite f veces la señal, el tiempo necesario para completarse una vez será:

$$T = \frac{1}{f} \quad f = \frac{1}{T}$$

Ecuación 1 determinar periodo y frecuencia de la señal respectivamente.

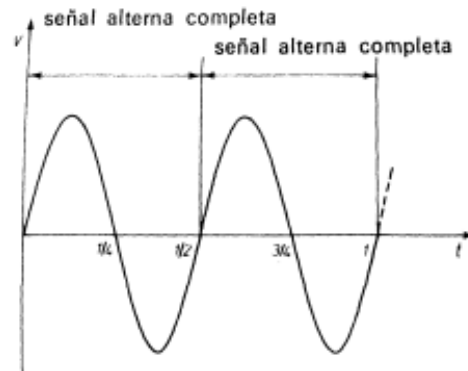


Figura 3 Ilustración de periodo y frecuencia en una señal sinusoidal.

- **Valor máximo** de todos los valores instantáneos comprendidos en un período, se denomina valor máximo al mayor de ellos. También a este valor se le denomina amplitud de la señal alterna y, otras veces, valor de cresta. Al igual que el valor instantáneo, su unidad depende de la magnitud considerada.

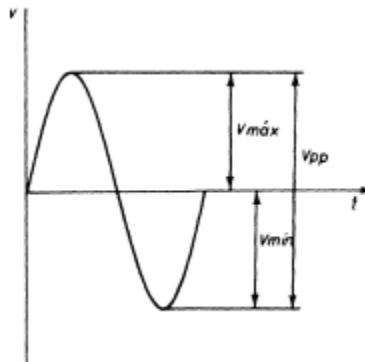


Figura 4 Visualización de parámetros de Valor máximo, mínimo y de pico.

En las señales alternas senoidales, el valor máximo coincide, en valor absoluto, con el valor mínimo. A cualquiera de estos valores se les designa también con el nombre de valor de pico. Interesante, a veces, en el tratamiento

de la señal alterna, es el valor comprendido entre dos picos consecutivos, denominado valor de pico a pico. (Garrigós, 2011)

1.4 Valor eficaz

Se dice que el voltaje disponible para un enchufe de pared en una resistencia debe ser de 110 V. Desde luego, este no es el valor medio del voltaje sinusoidal porque se sabe que el voltaje debería ser cero. Tampoco es el valor instantáneo o el valor máximo del voltaje.

El valor efectivo de un voltaje es una medida de su efectividad al transmitir potencia a un resistor de carga. El concepto de un valor efectivo viene desde un deseo de contar con un voltaje (o corriente) senoidal. El objetivo es encontrar un valor eficaz de cd que transmita la misma potencia promedio al resistor como la debería transmitir una fuente variable de manera periódica como se muestra a continuación: (Drof & Svoboda, 2011)

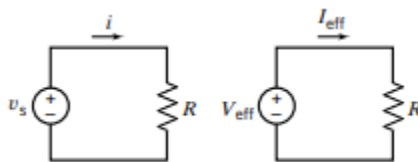


Figura 5 Misma potencia promedio al resistor como la debería transmitir una fuente variable de manera periódica

$$A = \sqrt{\frac{1}{T} \int_{t_0}^{t_0+T} f^2(t) dt}$$

Ecuación 2 Calculo de valor eficaz.

1.5 Tensión

Magnitud física que cuantifica la diferencia de potencial eléctrico entre dos puntos. También se puede definir como el trabajo por unidad de carga ejercido en el campo eléctrico sobre una partícula cargada para moverla entre dos posiciones determinadas. Se puede medir con un voltímetro. Su unidad de medida es el voltio. La tensión es independiente del camino recorrido por la carga y depende exclusivamente del potencial eléctrico de los puntos A y B en el campo eléctrico. Clasificación de las redes de distribución de acuerdo con voltajes nominales.

En Colombia existen varios voltajes de diseño para circuitos secundarios. Los siguientes son los voltajes de diseño de redes urbanas y rurales que permiten abastecer al servicio residencial, comercial, a la pequeña industria y al alumbrado público cuando estos 2 últimos son alimentados por la red secundaria (aunque esto no es deseable).

- Monofásico trifilar 240/120 V con punto central a tierra.
- Trifásico tetra filar 208/120 V con neutro a tierra y 220/127 V con neutro a tierra. Hoy existe en el sector un sector intermedio 214/123 V.
- Trifásico en triángulo con transformadores monofásicos, de los cuales uno solo tiene conexión a tierra 240/120 voltios. (Ramírez Castaño, 2004)

1.6 Potencia

A continuación, se listan las clases de potencias:

- **La potencia activa** es la más familiar porque es la que usan los equipos para efectuar un trabajo útil (calor, movimiento, luz, etc.). Esta potencia activa se mide en watts (W), y se representa con la letra P.
- **La potencia reactiva** es aquella que se hace evidente por sus efectos, ya que es la que establece el campo magnético de un motor, o bien el campo electrostático de un condensador o capacitor. La potencia reactiva se mide en VAR, (voltamperes reactivos), y se representa con la letra Q (de Quadratur).
- **La potencia aparente** es la que describe el total, o aparente flujo de potencia. La potencia aparente se mide en VA (volts-amperes), y se representa con la letra S (de Scheinleistung - Potencia Aparente).

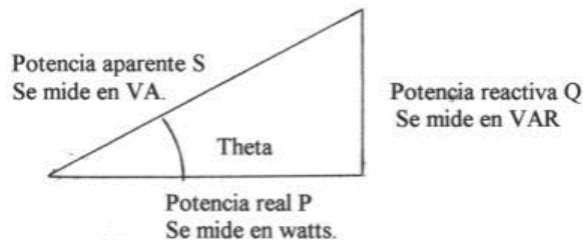


Figura 6 Representación algebraica de los valores de potencia real, reactiva y aparente.

Donde la potencia aparente es igual a la raíz cuadrada de la suma del cuadrado de la potencia activa más el cuadrado de la potencia reactiva. (Ruelas Gómez, Roberto)

$$S = (P^2 + Q^2)^{0.5}$$

Ecuación 3 Cálculo de la potencia aparente.

1.7 Sensor de corriente

La familia SCT-013 son sensores de corrientes no invasivos que permiten medir la intensidad que atraviesa un conductor sin necesidad de cortar o modificar el conductor. Es posible emplear estos sensores con un procesador como Arduino para medir la intensidad o potencia consumida por una carga.

Los sensores SCT-013 son transformadores de corriente, dispositivos de instrumentación que proporcionan una medición proporcional a la intensidad que atraviesa un circuito. La medición se realiza por inducción electromagnética.

Los sensores SCT-013 disponen de un núcleo ferromagnético partido (como una pinza) que permite abrirlo para arrollar un conductor de una instalación eléctrica sin necesidad de cortarlo.

La precisión del sensor puede ser de 1-2%, pero para ello es muy importante que el núcleo ferromagnético se cierre adecuadamente. Hasta un pequeño hueco de aire puede introducir desviaciones del 10%.

Como desventaja, al ser una carga inductiva, el SCT-013 introduce una variación del ángulo de fase cuyo valor es función de la carga que lo atraviesa, pudiendo llegar a ser de hasta 3°. (Llamas, 2017)



Figura 7 Sensor de Corriente SCT-013. Referencia de imagen (Llamas, 2017)

1.8 Sensor de voltaje

El módulo transformador de voltaje alterno ZMPT101B permite medir voltaje alterno, este voltaje AC no puede ser medido directamente por el ADC del Arduino pues escapa al rango de entrada (0V a 5V).

El módulo ZMPT101B soluciona el problema reduciendo el voltaje AC de entrada a un voltaje menor que pueda ser leído por el Arduino o cualquier otro microcontrolador.

El módulo está integrado por un transformador que cumple la función de aislamiento galvánico para mayor seguridad en el uso. El lado primario del transformador se conecta al voltaje alterno que deseamos medir. En el lado secundario del transformador se encuentra un divisor de tensión y un circuito con amplificador operacional (OPAMP LM358) para adicionar un desplazamiento (offset) a la salida análoga.

Soporta voltajes de entrada de hasta 250VAC y entrega una onda senoidal de amplitud regulable por un potenciómetro en placa. La onda senoidal de salida está

desplazada positivamente para que la onda no tenga voltajes negativos y así poder leer la onda completamente con el ADC. El desplazamiento depende del voltaje con el que se alimenta el módulo: si el voltaje de alimentación es de 5V el desplazamiento será de 2.5V y si se alimenta el módulo con 3.3V el desplazamiento será de 1.65V. El circuito de acondicionamiento de señal permite que el voltaje de salida del módulo pueda ser leído por cualquier microcontrolador con entrada analógica (ADC), de esta forma es posible leer el voltaje instantáneo y realizar cálculos de energía, como: voltaje pico a pico (V_{pp}) y voltaje eficaz (V_{rms}). (Naylamp Mechatronics, s.f.)

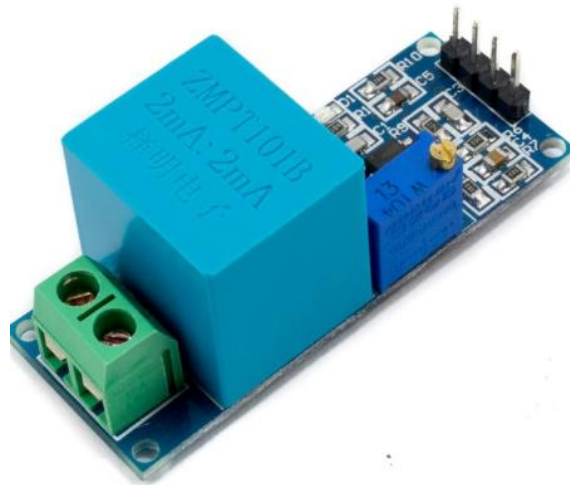


Figura 8 Sensor de Voltaje ZMPT101B Referencia de imagen (Naylamp Mechatronics, s.f.)

1.9 Microcontrolador

El Arduino Leonardo es una placa de microcontrolador basada en el ATmega32u4 (hoja de datos). Tiene 20 pines digitales de entrada / salida (de los cuales 7 pueden usarse como salidas PWM y 12 como entradas analógicas), un cristal de 16 MHz oscilador, una conexión micro USB, un conector de alimentación, un encabezado

ICSP y un botón de reinicio. Contiene todo necesario para soportar el microcontrolador; simplemente basta con conectarlo a una computadora con un cable USB o encenderlo con un Adaptador de CA a CC o batería para comenzar.

Leonardo difiere de todas las placas anteriores en que el ATmega32u4 tiene comunicación USB incorporada, eliminando la necesidad de un procesador secundario. Esto permite que el Leonardo aparezca a una conectada computadora como mouse y teclado, además de un puerto serie / COM virtual (CDC). (Arduino, s.f.)



Figura 9 Tarjeta Microcontroladora Arduino. Referencia de imagen (Arduino, s.f.)

1.10 Comunicación móvil

En algunos sistemas móviles es posible que un terminal hable directamente con otro, pero en la mayoría de los sistemas de comunicación se hace a través de un elemento intermedio al que se le llama estación base. Las estaciones base son visibles

gracias a las antenas que se sitúan en los tejados de los edificios o en lo alto de los mástiles que se observan por las ciudades y el campo. En algunos casos las estaciones base existen, pero disimuladas. Cada una de estas bases es capaz de atender a los usuarios próximos.



Figura 10 Antenas de transmisión y recepción en red celular. Tomada referencia (Casademont, 2010)

Al territorio que atiende una estación base se le llama celda. Es posible imaginar una celda como una baldosa, de modo que las celdas se colocan una al lado de otra para cubrir todo el suelo.

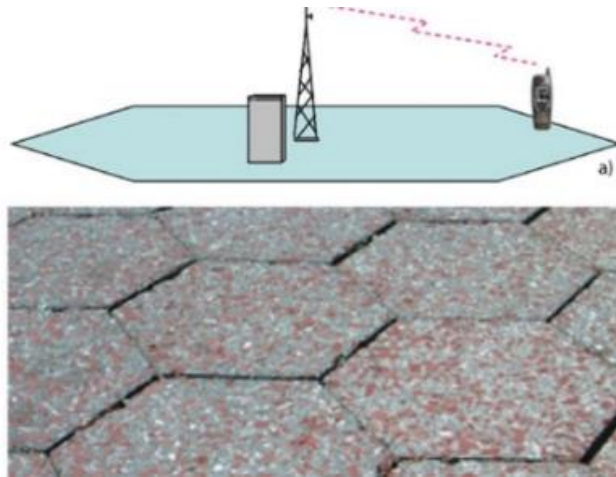


Figura 11 Estructura de Célula en la red móvil. Tomada referencia (Casademont, 2010)

Idealmente todo el territorio está cubierto por celdas y cuando se realiza un desplazamiento en coche o a pie se va pasando de una celda a otra. En algunos casos las celdas no encajan unas con otras y se tienen huecos de cobertura.

En cada una de las celdas están disponibles los canales para poder realizar las comunicaciones vía radio. Los canales no pueden ser los mismos en cada celda. Si fuese de esta forma dos usuarios hablando en celdas vecinas se interferirían y la comunicación sería mala o imposible. Los canales se distribuyen entre las celdas de forma que dos canales iguales no estén demasiado cerca y se puedan llegar a molestar. (Casademont, 2010)

1.11 SIM Card

Cuando una persona se hace cliente de un operador (se suscribe), este le da permiso de usar la red. Realmente lo que hace el operador es dar permiso a una tarjeta electrónica llamada SIM. Esta tiene todos los datos necesarios para que el usuario, al insertar esta tarjeta en una terminal, pueda acceder con esa terminal a la red. La SIM

tiene dos tipos de información. Por un lado, tenemos la propia del usuario con capacidad de almacenar SMS o números de teléfono. Otra información que almacena es la relativa al funcionamiento del sistema. Se guarda un número único a nivel mundial que identifica a la SIM. Este número es imprescindible para poder hacer itineraria o roaming.

1.12 GSM

La red GSM (Global System Mobile), es un sistema de comunicación móvil digital que está enfocado a la transmisión de texto y voz mediante la implementación de GPRS, es el estándar de telecomunicación celular más utilizado el mundo con cerca de un rango de penetración del 80 %. Su frecuencia de operación oscila en los 900 MHz y los 1900 MHz para el estándar implementado en EE. UU.

1.13 GPRS

Tecnología implementada en la tercera generación de la del estándar móvil GSM; soporta la transmisión de texto y voz a través de un método de transmisión digital. Permite tasas de transmisión de hasta 384 kbit/s. (Huidobro Moya, 2019)

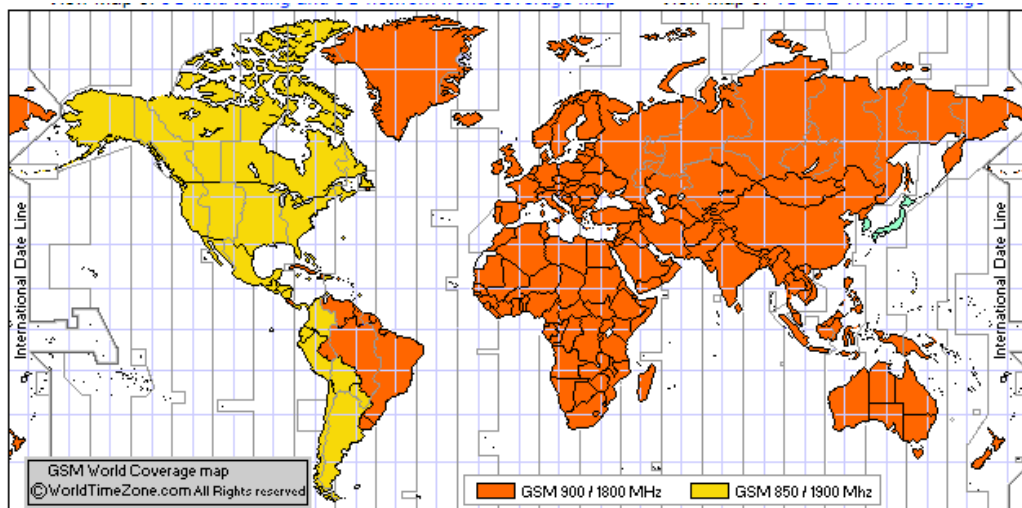


Figura 12 Mapa de cobertura mundial de la red GSM en las bandas de frecuencia. Referencia de imagen (Huidobro Moya, 2019)

1.14 Capas de modelo OSI

Afronta el problema de las comunicaciones de datos y las redes informáticas dividiéndola en niveles. Cada participante de la comunicación incorpora como mínimo uno de los mismos, y los equipos terminales los incorporan a todos.

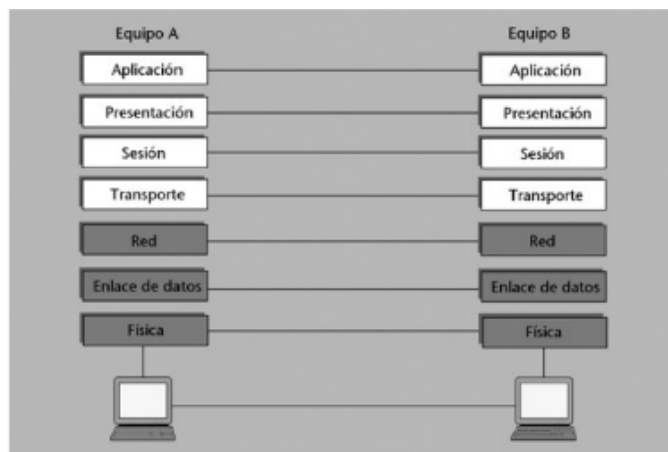


Figura 13 Estructura de funcionamiento del modelo OSI. Imagen tomada de (Íñigo Griera, y otros, 2008)

- **Nivel Físico:** Se encarga de las tareas de transmisión física de las señales eléctricas (o electromagnéticas) entre los diferentes sistemas. Las limitaciones del nivel físico (equipos de transmisión, recepción, medios de transmisión, amplificadores, etc.) imponen otras al resto del sistema, por otro lado, hacen aparecer una probabilidad de error, el porcentaje de bits erróneos que llegan a destino.
- **Nivel de Enlace:** Proporciona un servicio similar al nivel físico, mejorando las características de fiabilidad de la transmisión. Añade bits adicionales a los que forman el mensaje para poder detectar errores de transmisión en el mismo y poder pedir su retransmisión. Para ello, es preciso conferir una estructura a los bits: se agrupan en bloques denominados tramas, que contienen los bits de mensaje, los bits añadidos para detectar errores y diferentes campos de control, tales como el número de tramas.
- **Nivel de Red:** Las redes de conmutación de paquetes constituyen el tipo de red más eficiente para transmitir datos desde diferentes puntos de vista. En este nivel se distingue entre estaciones terminales y nodos de conmutación. Los nodos de conmutación disponen de diferentes enlaces hacia otros nodos o hacia terminales, y son los que permiten que los paquetes viajen por la red desde una terminal a otra.

La asignación de direcciones (addressing) es uno de los conceptos básicos del nivel de red. Permite a la red, como sistema distribuido, pero único, decidir cuál de los múltiples terminales es el destino final.

El encaminamiento (routing) constituye el procedimiento que permite a este sistema distribuido conducir la información por los diferentes nodos de origen a destino, minimizando el trayecto y el tiempo de tránsito, optimizando recursos.

Dentro de los protocolos que se destacan en esta capa se destacan IP, ICMP y ARP.

- **Nivel de transporte:** La función principal de este nivel consiste en asegurar la calidad de transmisión entre los terminales que utilizan la red, lo que implica recuperar errores, ordenar correctamente la información, ajustar la velocidad de transmisión de la información (control de flujo), etc.

Los protocolos que usa esta capa son TCP y UDP.

- **Nivel de sesión:** En teoría, el encargado de gestionar las conexiones de larga duración, la recuperación de caídas de red de manera transparente y protocolos de sincronía entre aplicaciones.

Algunos de los protocolos que maneja esta capa son SSL, NetBios, RPC.

- **Nivel de presentación:** Se encarga de conseguir que las diferentes plataformas (sistemas operativos, procesadores, etc.) se puedan entender al conectarse por medio de la misma red. Dicho de otra manera, es él encarga de definir una manera universal de codificar la información. Dicha codificación puede poseer propiedades de eficiencia (por medio de la criptografía), etc.
- **Nivel de aplicación:** En este residen los programas. En este nivel se pueden encontrar servidores, clientes que acceden a estos últimos, aplicaciones que trabajan según un modelo simétrico (peer to peer), etc.

Algunos de los protocolos que usa esta capa son HTTP, SMTP, FTP, Telnet.
(Íñigo Griera, y otros, 2008)

1.15 MySQL

Es un sistema de administración de bases de datos relacionales rápido robusto y fácil de usar. Se adapta bien a la administración de datos en un entorno de red, especialmente en arquitecturas cliente/servidor. Se proporciona con muchas herramientas y es compatible con muchos lenguajes de programación. Es el más célebre del mundo Open Source, en particular gracias a su compatibilidad con el servidor de páginas Web Apache y el lenguaje de páginas Web dinámicas PHP.

MySQL se difunde como Open Source. Así, cualquiera que utilice MySQL tiene acceso al código fuente y puede modificar el código para adaptarlo a sus propias necesidades.

Debido a su licencia GNU General Public Licence (GPL), toda persona puede descargar, instalar y utilizar MySQL libremente sin ninguna aportación económica.

Una base de datos permite guardar los datos con independencia del formato físico de los archivos de almacenamiento, el usuario solo tiene que gestionar el formato lógico. El rendimiento se optimiza. Además, ofrece posibilidades de control de la integridad de datos. El lenguaje SQL aporta una serie de funciones uniformes que permiten trabajar con los datos sin que el desarrollador necesite escribir sus propias funciones.

Una base de datos permite guardar y relacionar datos por medio de tablas compuestas de columnas (llamadas campos) y de líneas (llamadas registros) que permiten guardar datos uniformes.

El siguiente esquema representa gráficamente una base de datos (Thibaud, 2006):

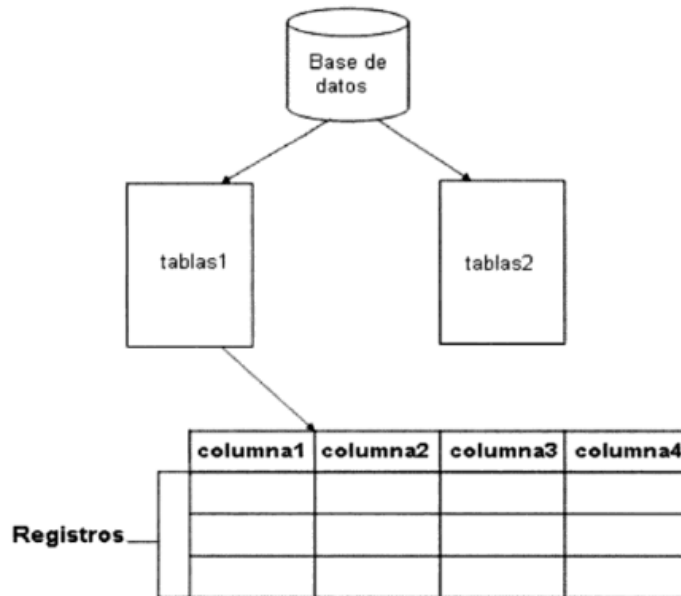


Figura 14 Esquema de representación de base de datos. Imagen tomada de (Thibaud, 2006)

1.16 PHP

Acrónimo de “Hypertext Processor” es un lenguaje de “código abierto” interpretado, de bajo nivel, embebido en páginas HTML y ejecutado en el servidor.

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la maquina cliente, es que el código PHP es ejecutado en el servidor. Si se tuviese un script similar al del ejemplo en el servidor, el cliente solo recibirá el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido. El servidor Web puede ser incluso configurado para que procese todos los archivos HTML con PHP.

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales.

PHP puede hacer cualquier cosa que se pueda hacer con un script, como procesar la información de los formularios, generar páginas con contenido dinámico o enviar o recibir cookies.

Existen tres campos en los que se usan scripts escritos en PHP:

- **Scripts del lado del servidor:** Es el campo más tradicional. Se necesitan tres cosas para que esto funcione. El intérprete (CGI o módulo), un servidor Web y un navegador. El resultado puede obtenerse a través de un navegador, conectándose con el servidor Web.
- **Scripts en la línea de comando:** Se puede crear un script PHP y correrlo sin ningún servidor web o navegador. Es ideal para ejecutarlos normalmente en planificador de tareas de Windows o tareas simples de procesamiento de texto. (Fossati, 2018)

1.17 HTML

Es un lenguaje artificial que los ordenadores son capaces de interpretar y diseñado para que los programadores redacten instrucciones que los navegadores ejecutan para originar la página web. Es decir, HTML es un lenguaje de programación, o “un idioma que la maquina entiende y procesa para dar respuesta”.

Los lenguajes de programación se utilizan para crear programas que especifiquen el comportamiento que debe tener una máquina, para expresar algoritmos con precisión, para realizar cálculos complejos, etc.

Las siglas HTML significan Hypertext Markup Language (lenguaje de marcas hipertexto). El hipertexto en una computadora es texto que posee referencias (hipervínculos, links o enlaces) a otro texto. Para simplificar se puede decir que el hipertexto es aquel texto que se pulsa con el ratón del ordenador y conduce a otro texto cuando se utiliza internet. Pero además de texto, el hipertexto puede estar formado por tablas, imágenes u otros elementos.

En esencia HTML sirve para estructurar documentos (títulos, párrafos, listas, etc), pero no describe la apariencia o el diseño de un documento, sino que ofrece las herramientas necesarias para dar formato, según la capacidad del servidor web en el que se almacenan las páginas web y la capacidad del navegador (tamaño de la pantalla, fuentes instaladas, etc.). Por esta razón y para no caer en la frustración si el aspecto no es el que se preveía, no se deben diseñar los documentos basándolos en cómo se muestran en el navegador, sino que hay que centrarse en proporcionar un contenido claro y bien proporcionado que resulte fácil de leer y entender. (Equipo Vértice, 2009)

1.18 CSS

Las siglas CSS son “Cascade Style Sheet”, en español hojas de estilo cascada. CSS es un lenguaje que permite otorgar atributos a los elementos de los documentos realizados en HTML, CSS permite realizar una separación del diseño (formatos y

estilos) de los contenidos de las páginas web. Es importante resaltar que, hay características que ofrece CSS y que también se pueden realizar con HTML, pero CSS tiene una mayor cantidad de opciones para personalizar los elementos de la página web en general.

Para ver las funcionalidades principales de HTML y CSS, se puede imaginar que se encuentra en la construcción de un edificio, las columnas y las paredes del edificio, serían la estructura de una página web desarrollado con HTML y la presentación del edificio sería como realizar el diseño de la página web desarrollado con CSS; para el desarrollo de la página web se separa el diseño del contenido, esta separación hace que el mantenimiento y manipulación del sitio web sea más fácil. (Condor Tinoco & Soria Solis, 2019)

1.19 Javascript

El lenguaje HTML permite dotar las páginas web de una forma atractiva información visual, no obstante, la falta de cierto grado de interactividad para el usuario, es decir, en la mayoría de los casos, una página web es mero escaparate (más o menos atractivo). Para dotar una página web de más interactividad podemos insertar, por ejemplo, applets java; un applet java es una mini aplicación construida en este lenguaje.

No obstante, Java es un lenguaje bastante complicado, en especial si no se tienen conocimientos en C++, o programación orientada a objeto. Así pues, no se trata de un instrumento fácil de utilizar para los que no conocen técnicas de programación.

Para llenar el vacío existente entre la sencillez y poca interactividad de HTML, y el grado de interactividad y dificultad de Java, las empresas de software se dispusieron a desarrollar órdenes fáciles de utilizar. De esta manera Netscape Communications introdujo el lenguaje Livescript, poco después la empresa Sun Microsystems (creadora del lenguaje de Java), se unió a Netscape para conseguir que Livescript fuese adoptado como lenguaje estándar de internet para la escritura de órdenes acerca de la web. Puesto que Livescript tenía muchas semejanzas con Java, el lenguaje fue renombrado Javascript.

Javascript es pues, un lenguaje de programación dirigido a los creadores de páginas web. (Pacheco Escribano, 2015)

2 CAPITULO DISEÑO E IMPLEMENTACIÓN

2.1 Arduino

Para el proceso de captura y procesamiento de los datos capturados referentes a tensión y amperaje, fue necesario el uso de una tarjeta microcontroladora, se optó por el Arduino referencia Leonardo. A continuación, se muestran sus especificaciones principales.

Tabla 1 Especificaciones básicas de funcionamiento para el Arduino Leonardo

Microcontrolador	ATmega32u4
Tensión de funcionamiento	5V
Alimentación recomendada	7-12V
Pines I/O Digitales	20
Canales PWM	7
Entradas analógicas	12
Corriente Máxima de los pines I/O	40 mA
Corriente Máxima de los pines 3.3V	50 mA
Memoria Flash	32 KB (4 KB usados para el bootloader)
SRAM	2.5 KB
EEPROM interna	1 KB
Velocidad	16 MHz

Dispositivos para la captura de los datos requeridos

Luego de escoger la tarjeta microcontroladora, la siguiente validación a realizar fue la selección de los sensores que se iban a utilizar teniendo en cuenta que se necesitaba tomar los valores de tensión y amperaje reales con la finalidad de obtener la potencia y con esta el consumo de Vatios realizado con respecto al tiempo, para lo cual se lleva a cabo la escogencia de los dos dispositivos a mencionados a continuación.

2.2 Sensor de amperaje.

Para la toma de los valores de amperaje se hace uso del sensor SCT-030-000 de la compañía YHDC, el cual permite obtener una lectura de manera no invasiva, con lo cual no se hace necesario generar un corte o modificación en la acometida eléctrica actual del lugar donde se toma la medición.

Tabla 2 Características técnicas del sensor SCT-030-000

Model	SCT-013-000
Input current	0-100A
Output type	0-50mA

El sensor tiene la capacidad de recibir corrientes entre los 0 y 100 A con una salida de esta de entre 0 y 50 mA. Su ventaja sobre otras referencias de la marca es que no cuenta con una resistencia de carga interna, lo que nos permite asignar una resistencia acorde al consumo que queremos medir.

Para obtener la resistencia de carga se debe identificar un valor promedio de consumo, para ello se deben tener los valores de consumo de alguno de los electrodomésticos en el domicilio. Para el proyecto se utilizaron los valores de consumo de una bombilla eléctrica que esta alrededor de los 40 Watts.

La resistencia de carga se obtuvo llevando a cabo las siguientes validaciones:

Cálculo de la corriente a medir.

Se tomó el valor de consumo de una bombilla de 40 Watts, teniendo en cuenta que el valor teórico de tensión suministrado para una red doméstica común en Colombia es de 110 V, hacemos uso de la siguiente ecuación.

$$I_{RMS} = \frac{P}{V} = \frac{40 \text{ W}}{110 \text{ V}} = 0.363 \text{ A}$$

Obtención de corriente Pico.

$$I_{pico} = I_{RMS} * \sqrt{2} = 0.363 \text{ A} * \sqrt{2} = 0.513 \text{ A}$$

Corriente pico del devanado secundario.

$$I_s = N_p * I_p / N_s = 1 * 0.513 \text{ A} / 2000 = 0.00025 \text{ A}$$

Luego de calcular los valores teóricos de amperaje debemos conocer la referencia interna de la entrada analógica AREF, para el Arduino, la cual hace referencia al voltaje máximo que se puede medir en las entradas analógicas de nuestra tarjeta el cual está comprendido entre los 0 y 5 Voltios para nuestro caso particular.

Al haber generado los cálculos de una corriente pico se debe dividir en dos el valor AREF para obtener un rango de medición de 2.5V a -2.5V de forma teórica aclarando que Arduino no tiene la capacidad de censar valores negativos.

$$R_{Carga} = \frac{A_{REF}}{2 * I_{Pico}} = \frac{5 \text{ V}}{2 * 0.00025 \text{ A}} = 10 \text{ k}\Omega$$

Luego de las mediciones de los cálculos de manera teórica logramos obtener un valor de 10 k Ω como valor teórico para nuestra resistencia de carga. Por lo tanto, debemos hacer uso de una resistencia con este valor.

Desplazamiento de nuestra Señal.

Luego de obtener nuestros cálculos teóricos se hace necesario determinar la manera de obtener un muestreo general de los datos, teniendo en cuenta que el amperaje es alterno, por lo cual tiene picos positivos y negativos y como fue aclarado previamente Arduino solo puede medir valores iguales o superiores a cero se hizo necesario la realización de un circuito Off SET.

Para el diseño de nuestro circuito Off SET hacemos uso de un condensador y un divisor de tensión, la utilidad del circuito es poder desplazar nuestros picos negativos y transponerlos al rango positivo de la medición.

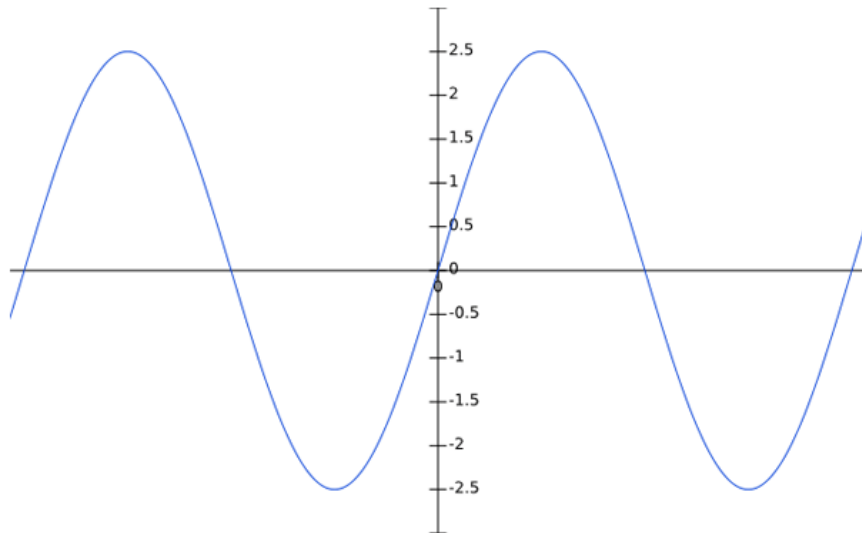


Figura 15 Señal sinusoidal normal

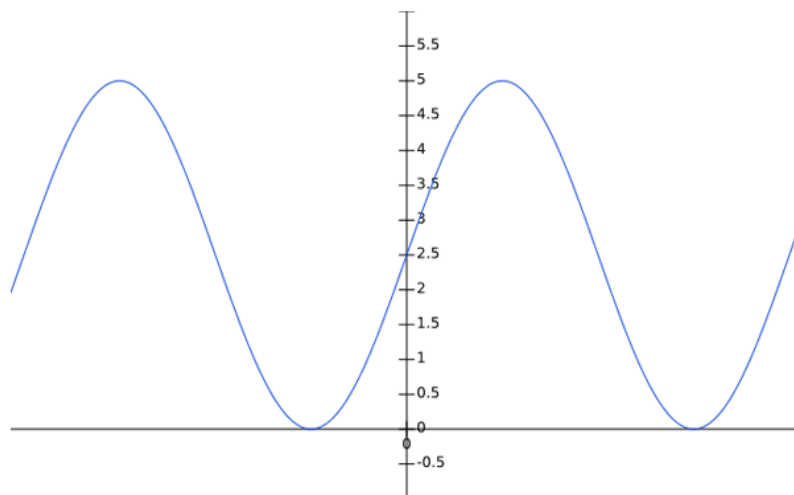


Figura 16 Señal sinusoidal luego de aplicado el Off SET

Por último, se obtuvo el esquema del circuito con las cargas de resistencia y del circuito Off SET.

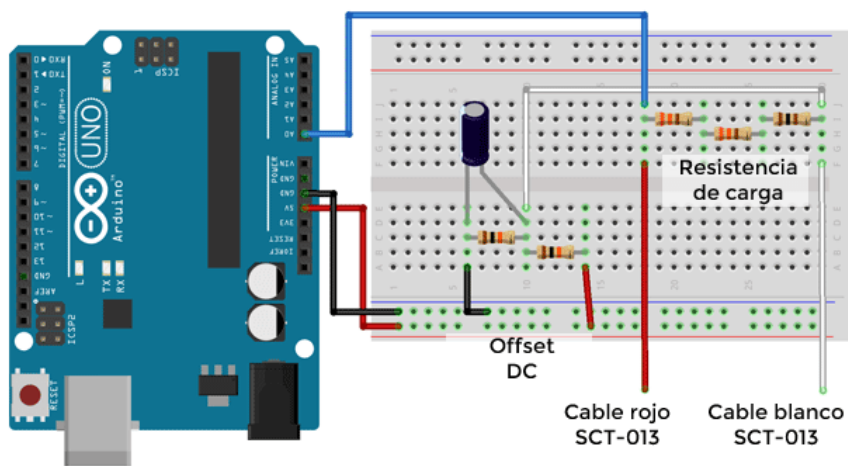


Figura 17 Esquema del circuito con los componentes de carga de resistencia y Off SET.

2.3 Módulo de transmisión de datos SIM800L

Para el proceso de transmisión de los datos capturados a través de los distintos sensores integrados al proyecto se hace uso de un módulo de transmisión de datos haciendo uso de la red móvil GSM (Global System Mobile), teniendo como uso la tecnología 2G. A continuación, las características técnicas del dispositivo:

Tabla 3 Características básicas del módulo SIM800L.

Especificación	Descripción
Voltaje de alimentación	3.4 V- 4.4 V
Bandas de frecuencia	850, 900, 1800, 1900 (MHz)
Modo de programación	Comandos AT
Temperatura de operación	-40°C a 85°C
Dimensiones	1.5 cm * 1.8 cm * 0.25 cm
Tasa de descarga de datos máxima	85.6 Kbps
Tasa de subida de datos máxima	85.6 Kbps

La estructura mediante la cual se genera la activación de GPRS, la conexión a la red móvil y el proceso de encapsulamiento de los datos se tramita a través del uso de los comandos AT, mediante los cuales fue posible dar las instrucciones correspondientes.

2.4 Programación en Arduino.

El paso siguiente fue generar la realización de un código, con el que se pretendió que el sensor tomara las lecturas de voltaje y de corriente.

Estructura del código de programación implementado.

Como primer paso dentro de la estructura del código de programación implementado se determinan las librerías a utilizar para dar funcionamiento óptimo al programa.

```
#include <SoftwareSerial.h>
#include <String.h>
#include "EmonLib.h"
```

Figura 18 librerías utilizadas en el diseño del programa.

Las librerías que se implementaron permiten realizar lo siguiente:

- “SoftwareSerial.h”: La librería permite implementar una comunicación de manera serial con la placa Arduino a través de los puertos COM del ordenador utilizado para el cargue del código.
- “String.h”: La librería permite manipular las variables capturadas por los sensores de medición a un formato de texto el cual es maleable dentro de la base de datos alojada en el servidor.
- “Emonlib.h”: Permite traer la información necesaria para realizar los cálculos de corriente y voltaje eficaces como la toma de una serie de muestras configuradas a elección del programador.

Selección de pines para la ejecución de tareas.

Dentro del código realizado se debió asignar los pines que en la placa Arduino permitirían realizar la toma de los valores de medición y procesos de comunicación con el puerto serial del ordenador utilizado.

- Los pines 9 y 10 son asignados como entrada y salida respectivamente para la comunicación serial con la tarjeta Arduino.
- Los pines A0 y A1 se utilizaron como entradas para recolección de las lecturas tomadas por los sensores de corriente y voltaje respectivamente.

Comunicación Serial

La comunicación serial es de vital importancia, porque permite visualizar y corroborar la información capturada por los sensores de medición, y que a su vez es transmitida a través del módulo sim implementado.

El primer paso dentro de la estructura del código es determinar la velocidad que será establecida para la comunicación serial la cual es determinada en Baudios, la velocidad seleccionada es de 19200 Baudios con los pines de comunicación 9 (RX) y 10 (TX).

```
SoftwareSerial Sim800Serial(9,10);
```

Dentro de la información cargada en la estructura principal del código. Se asignan dos instancias de comunicación serial, una enfocada a la comunicación entre el ordenador y la placa Arduino y una segunda instancia inicia una comunicación con el módulo Sim800.

```
Sim800Srial.begin(19200);//Arduino se comunica con el SIM800 a una velocidad de 19200bps  
Serial.begin(19200);//Velocidad del puerto serial de arduino
```


Estructura del código principal.

Dentro de la estructura principal ejecutada dentro de la función “Void Setup ()” y “Void Loop ()” se realizó el formato de comunicación con el módulo sim800 y la ejecución de la función “comandosAT ()”, encargada del encapsulamiento de los datos censados para ser enviados a través de la red GSM.

2.5 Comunicación con el módulo Sim800.

La comunicación con el módulo Sim se realiza mediante la activación de una serie de tareas que en conjunto permiten activar el módulo, enlazarlo a la red móvil y prepararlo para activar el envío de datos a través de la red móvil del operador seleccionado.

```
Sim800Serial.println("AT+SAPBR=3,1,\"Contype\", \"GPRS\");  
delay(3000);  
mostrarDatosSeriales();  
  Sim800Serial.println("AT+SAPBR=3,1,\"APN\", \"\"moviletb.net.co\");  
delay(3000);  
mostrarDatosSeriales();  
Sim800Serial.println("AT+SAPBR =1,1");  
delay(3000);  
mostrarDatosSeriales();
```

La estructura de comunicación inicial permite iniciar la configuración de acceso a la red a través de la tecnología GPRS, se ingresan los datos de APN (Access Point Name) del operador móvil y se activa el servicio de transmisión de datos en la red móvil. La función “mostrarDatosSeriales()” tiene como finalidad obtener una respuesta visualizada en el monitor serie a la tarea ejecutada dentro de los comandos AT para corroborar si esta fue realizada con éxito.

```
void mostrarDatosSeriales ()  
{  
  while (Sim800Serial.available () !=0)  
    Serial.write (Sim800Serial.read ());  
}
```

Función “comandosAT()”

El uso de la función permitió generar una estructura capaz de enlazar los datos de medición obtenidos con los dispositivos de lectura al módulo Sim que se encarga de enviarlos vía internet al servidor a través de la red del operador móvil.

```
Sim800Serial.println ("AT+HTTPIPINIT");  
delay (3000);  
mostrarDatosSeriales ();  
Sim800Serial.println ("AT+HTTTPARA=\ "CID\ ",1");  
delay (300);  
mostrarDatosSeriales ();
```

Como se evidencia en la figura anterior el proceso de transmisión de los datos capturados se genera mediante una conexión a través del puerto HTTP (80) con lo cual se generó la apertura de la conexión y se debe obtener una respuesta afirmativa de conexión a través del monitor serial.

Arreglo del encapsulamiento de los datos a transmitir.

Una de la tareas que se ejecutaron dentro de la función comandosAT(), es el esquema para el envío de los datos el cual se planteó de la siguiente manera:

```
Sim800Serial.println ("AT+HTTTPARA=\ "URL\ ",\ "http://www.modulomedicion.online/lecturas.php?pass=
```

El primer paso de la línea de programación donde se encapsulan los datos es determinar la conexión con la URL de acceso al servidor para la cual se declaró el

nombre de la página web, así como también se informa el nombre de la base de datos a consultar “lecturas.php” y la contraseña de acceso a esta base.

```
&id_sensor=34AB
```

El siguiente paso fue el ingreso de las variables consultables dentro de la base de datos comenzando por la variable encargada de asignar una identificación al módulo desde el cual se están enviando los datos.

```
&corriente="+ String(Irms) + "&voltaje=" + String(voltaje) + "&potencia=" + String(potencia) + "\"";
```

Por último, se ingresaron los datos de las variables en las cuales se almacenaron los datos a transmitir referentes a corriente, voltaje y potencia, estos valores fueron transformados a un formato de texto para que se pudieran manipular en el servidor.

2.6 Configuración información recibida del microcontrolador

Para la recepción de los datos en la nube se optó por adquirir un Hosting ya que localmente las empresas de comunicaciones no ofrecen el servicio de IPs públicas para este tipo de proyectos. Luego de realizar la verificación de las diferentes empresas que ofrecen el servicio de hosting se optó por escoger Hostinger.com ya que este además de lo económico y facilidades de pagos, contaba con los servicios y herramientas que se requerían para la implementación del proyecto y adicionalmente tenía una configuración entendible para poner en puesta la ejecución e implementación el proyecto.

Después de realizados los pagos, lo primero que se configuro fue el dominio, esta es la dirección web con la cual se da a conocer el proyecto en la web, en este caso

se le dio el nombre “modulomedicion.online” y este enlace es el más importante porque es aquí donde deben llegar cada una de las lecturas obtenidas del sensor.

En la imagen que se relaciona a continuación se puede evidenciar que el dominio este activo, se tiene un registro del tiempo de caducidad, se puede hacer el registro de otros dominios y se tienen otras opciones de administración como logs de la actividad y las configuraciones de las Zonas DNS.



Figura 19 Estado del hosting

Luego de realizada la configuración del dominio al ingresar a la URL desde un navegador se pudo evidenciar que la página ya se encontraba en internet con lo que se determinó que el servicio ya estaba correctamente configurado, tal como se muestra en la figura 20 a continuación:

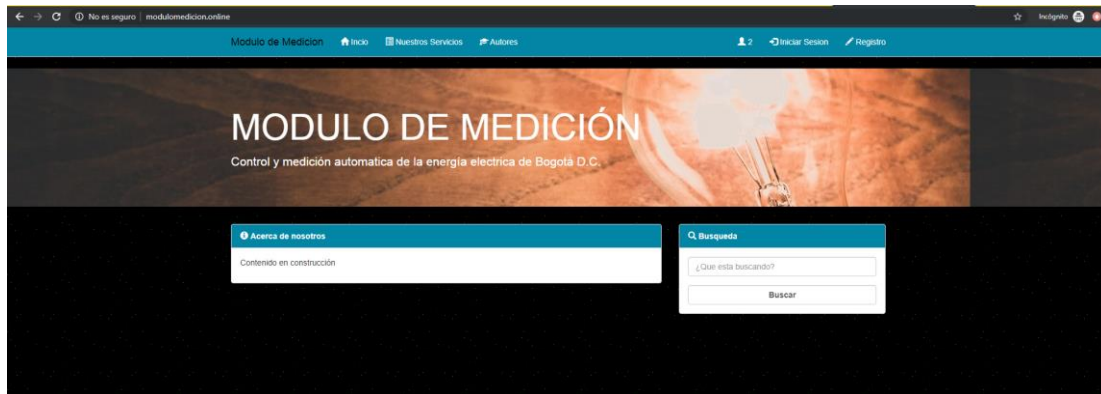


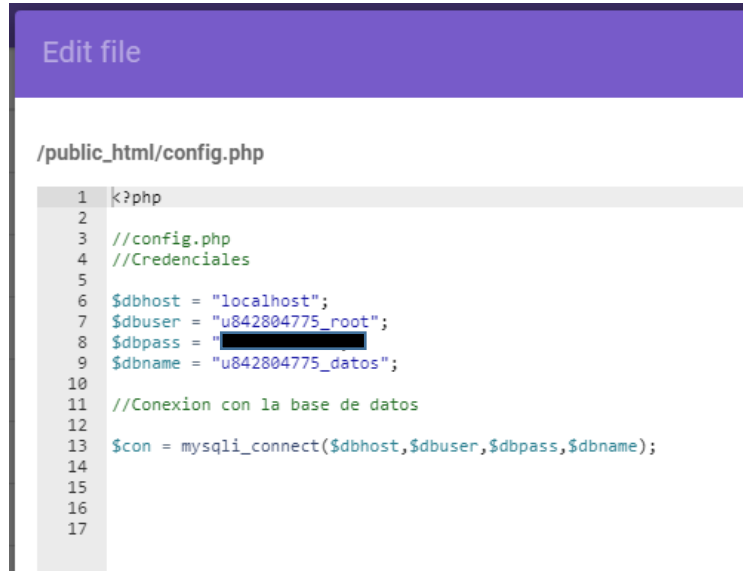
Figura 20 Página en funcionamiento en internet

Enseguida se realizó la elaboración de las tablas que componen la base de datos, esta también importante para los datos que se reciben del sensor. La conformación de las tablas se diseñó de la siguiente forma, la primera llamada “usuarios” donde se almacenan todos los datos de las personales y datos del predio de las personas que se inscriban por la página web para la obtención del servicio y una segunda tabla que contiene todos los datos del sensor (marca serial) y almacenan los datos que recibe el sensor (corriente, voltaje y potencia).

u842804775_datos lecturas	u842804775_datos usuarios
id : int(255)	id : int(11)
fecha : timestamp	fecha_registro : datetime
id_sensor : longtext	nombre : varchar(25)
# corriente : float	apellidos : varchar(255)
# voltaje : float	ciudad : varchar(255)
# potencia : float	barrio : varchar(255)
	direccion : varchar(255)
	# estrato : bigint(255)
	id_sensor : varchar(255)
	correo : varchar(255)
	password : varchar(255)
	# activo : tinyint(4)

Figura 21 Tablas MySQL

Finalmente, a nivel de servidor se realizó la copia del código que previamente se diseñó en el servidor de pruebas y se validó la funcionalidad de los servicios previamente elaborados, teniendo en cuenta los cambios en algunos archivos que contenían direcciones ip a nivel local, actualizando estas al nuevo dominio adquirido. En la imagen que a continuación se puede apreciar uno de los archivos más importante a nivel de conexión, ya que es aquí donde indicamos la conectividad a la base de datos y de esta forma lograr que los datos recibidos puedan ser almacenados correctamente. También la importancia de este archivo radica en que desde cualquier entorno de programación y si se necesita traer información de la base de datos, con llamar este archivo se tiene una conexión directa al mismo sin tener que sobre escribir la información y sin hacerla visible a personas no autorizadas.



```
1 |<?php
2 |
3 | //config.php
4 | //Credenciales
5 |
6 | $dbhost = "localhost";
7 | $dbuser = "u842804775_root";
8 | $dbpass = "██████████";
9 | $dbname = "u842804775_datos";
10 |
11 | //Conexion con la base de datos
12 |
13 | $con = mysqli_connect($dbhost,$dbuser,$dbpass,$dbname);
14 |
15 |
16 |
17 |
```

Figura 22 Archivo Config.php

El archivo config.php que se evidencia en la figura 22 se compone de la siguiente estructura:

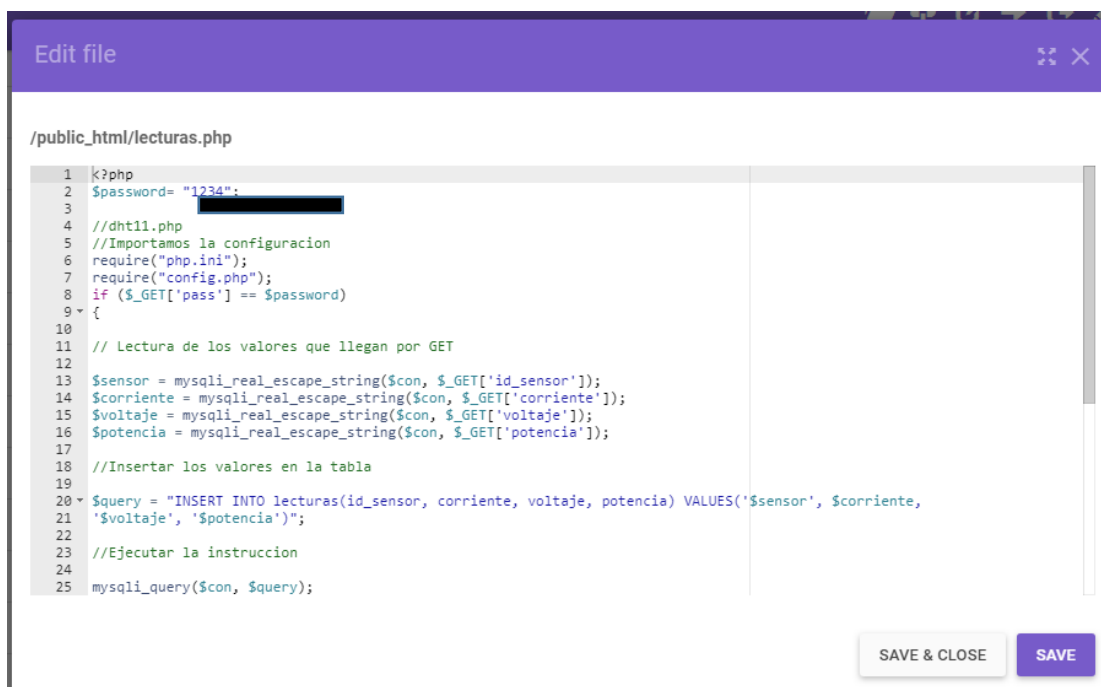
- Nombre del servidor
- Usuario de ingreso al servidor
- Contraseña de ingreso al servidor
- Nombre de la base de datos

Otro de los archivos importantes en la recepción de los datos es `lecturas.php` y su importancia radica en que es a este archivo donde se apunta específicamente desde el Arduino para el almacenamiento de los datos, este se compone de los siguientes parámetros:

Inicialmente se tiene la conexión con el archivo `config.php` explicado anteriormente con el fin de obtener la conexión a la base de datos, en el “if” se configura una seguridad es decir solo quien tenga la contraseña puede enviar a esa URL.

Seguidamente se configuran las variables llamando la conexión a la base de datos y utilizando los métodos GET para obtener cada una de las variables que vienen desde el Arduino (`id_sensor`, corriente, voltaje, potencia).

Enseguida se le ordena al sistema por medio de una sentencia MySQL que inserte cada uno de los valores almacenados en las variables a la tabla `lecturas` que se configuro previamente en la base de datos.



```
1 <?php
2 $password= "1234";
3
4 //dht11.php
5 //Importamos la configuracion
6 require("php.ini");
7 require("config.php");
8 if ($_GET['pass'] == $password)
9 {
10
11 // Lectura de los valores que llegan por GET
12
13 $sensor = mysqli_real_escape_string($con, $_GET['id_sensor']);
14 $corriente = mysqli_real_escape_string($con, $_GET['corriente']);
15 $voltaje = mysqli_real_escape_string($con, $_GET['voltaje']);
16 $potencia = mysqli_real_escape_string($con, $_GET['potencia']);
17
18 //Insertar los valores en la tabla
19
20 $query = "INSERT INTO lecturas(id_sensor, corriente, voltaje, potencia) VALUES('$sensor', $corriente,
21 '$voltaje', '$potencia')";
22
23 //Ejecutar la instruccion
24
25 mysqli_query($con, $query);
```

Figura 23 Configuración archivo lecturas.php

Finalmente, la última línea de código le ordena al sistema que se ejecute las instrucciones previamente dadas. Y es de esta forma en que se almacenan los datos que se envían desde el microcontrolador en la base, se puede evidenciar en la figura 24 una muestra de algunas lecturas que se tomaron durante el funcionamiento del módulo y que se fueron almacenando según la instrucción dada anteriormente.



Mostrando filas 0 - 24 (total de 4409. La consulta tardó 0.0009 segundos.)

```
SELECT * FROM `lecturas`
```

Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la cl

	id	fecha	id_sensor	corriente	voltaje	potencia
<input type="checkbox"/>	101	2019-05-15 23:07:29	34AB	0.45	70	0.34
<input type="checkbox"/>	102	2019-05-15 23:14:38	34AB	0.36	110	40.1
<input type="checkbox"/>	103	2019-05-15 23:15:48	34AB	0.36	110	39.18
<input type="checkbox"/>	104	2019-05-15 23:16:59	34AB	0.36	110	39.28
<input type="checkbox"/>	105	2019-05-15 23:18:11	34AB	0.01	110	0.88

Figura 24 Lecturas almacenadas

Se puede apreciar que a la izquierda de la imagen las tablas por las que se compone la base de datos, en este caso “lecturas” en donde se está almacenando la información recibida, y en la parte derecha podemos apreciar el id_sensor, y los valores de corriente, voltaje y potencia recibidos.

2.7 Visualización de la información

Dentro de la creación de la página se creó un entorno el cual permite que el usuario que quiera usar el servicio complete un corto formulario con información básica, con el fin de que se pueda visualizar la información de una forma coherente y clara tal como se muestra en la siguiente imagen.

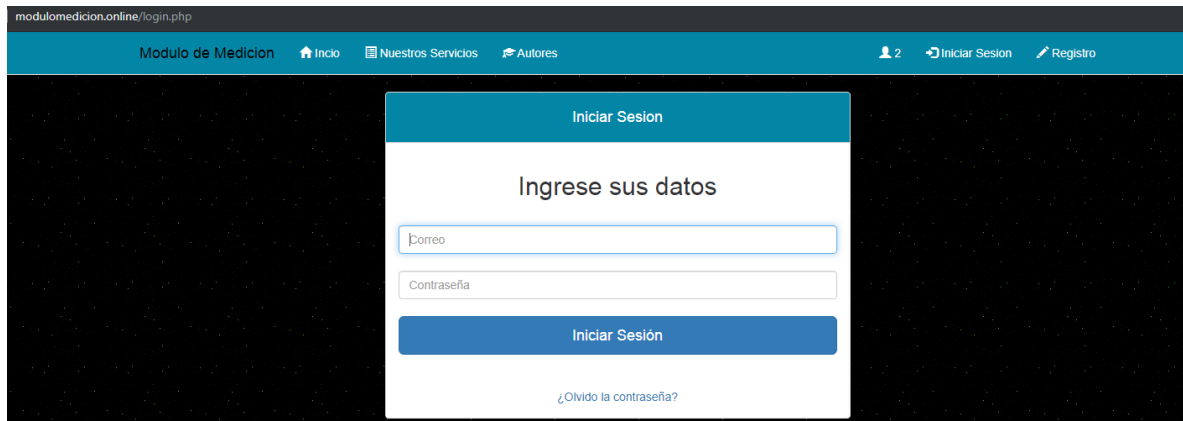


Figura 25 Login de usuario

Después que la persona se ha registrado e ingresado, se optó por mostrar la siguiente información:

Datos personales

En módulo de la página se muestra la información del usuario, esta información se encuentra almacenada en el servidor de bases de datos como “contador.php” y a

continuación se detalla la conformación del código, en el cual podemos apreciar algunas etiquetas html.

```
/public_html/contador.php
18 <div class="panel panel-default">
19 <div class="panel panel-heading" >
20 <h4 class="text-center">DATOS PERSONALES</h4>
21 </div>
22 <?php
23 require("config.php");
24 include_once 'config.php';
25 $sql = "SELECT * FROM usuarios WHERE nombre = 'Javier'";
26 $result = mysqli_query($con, $sql);
27
28 while ($mostrar = mysqli_fetch_array($result)) {
29     ?>
30     <table class="table">
31     <tr>
32         <th>Nombre</th>
33         <td><?php echo $mostrar['nombre'] ?></td>
34     </tr>
35     <tr>
36         <th >Apellidos</th>
37         <td><?php echo $mostrar['apellidos'] ?></td>
38     </tr>
39     <tr>
40         <th>Ciudad</th>
41         <td><?php echo $mostrar['ciudad'] ?></td>
42     </tr>
```

SAVE & CLOSE SAVE

Figura 26 Configuración archivo contador.php

Seguidamente tenemos una conexión por php donde se realiza una consulta al servidor de base de datos, esta consulta nos lista toda la información del usuario relacionado y la almacena en un array. Después se llaman del array los valores que se quieren mostrar en la página. Para el proyecto se enlisto la información que se muestra en la siguiente imagen.



DATOS PERSONALES	
Nombre	Javier
Apellidos	Rojas Cárdenas
Ciudad	Bogota
Barrio	Alqueria
Dirección	Calle 3 # 12 78
Estrato	1
Correo	rojasjavier@outlook.com
Serial Sensor	34AB

Figura 27 Datos información de usuarios

Grafica en barras

Se optó que en esta misma plantilla “contador.php” se realizara la visualización de la información recibida por gráficas, esto con el fin de que el usuario tenga de una forma más entendible la información que recibe de los sensores. Para el diseño a nivel de programación de este tipo de gráficas, se usa la librería llamada plotly, esta es una librería de código abierto obtenida de <https://plot.ly/r/>, en donde se realizó la configuración con la consulta para que se obtuvieran los datos de los consumos por meses.

```
/public_html/contador.php
90 <table class="table">
91 <div class="col-sm-12">
92 <meta http-equiv="refresh" content="5">
93 <?php
94 $sql2 = "SELECT MONTHNAME(Fecha) Mes, SUM(potencia) total_mes FROM lecturas GROUP BY Mes ORDER BY
    Fecha ASC";
95 $result2 = mysqli_query($con, $sql2);
96 $valoresX2 = array(); //Fechas
97 $valoresY2 = array(); //Potencia
98 while ($ver2 = mysqli_fetch_row($result2)) {
99     $valoresY2[] = $ver2[1];
100     $valoresX2[] = $ver2[0];
101 }
102
103 $datosX2 = json_encode($valoresX2);
104 $datosY2 = json_encode($valoresY2);
105 ?>
106 <div id="graficabarras"></div>
107 <script type="text/javascript">
108     function crearCadenaBarras(json) {
109         var parsed = JSON.parse(json);
110         var arr = [];
111         for (var x in parsed) {
112             arr.push(parsed[x]);
113         }
114     }
115 </script>
116 </div>
117 </table>
```

Figura 28 Sentencia para llamar información por mes

Como se puede apreciar en la imagen anterior, en la consulta al servidor se le indica que traiga los datos de fecha y la suma de la potencia (que es lo que realmente se le factura al usuario final) de la tabla lecturas y los ordene por mes de forma descendente. De esta forma se asigna una variable al eje x (fecha) y a la variable y la suma de las potencias. Ya con estas variables definidas se realizó la inclusión dentro del código como resultado la siguiente gráfica.



Figura 29 Grafica de consumo por mes

En esta grafica el usuario tiene la facilidad de poder evidenciar gráficamente el consumo de energía en el mes actual y adicionalmente cuenta con los datos históricos de los meses anteriores para que de esta forma se pueda tener un control y así el usuario puede determinar en qué periodos las lecturas han tenido un mayor consumo y de esta forma poder regular el consumo de energía.

Grafica de consumo en tiempo real

En la plantilla “contador.php” se realizó el desarrollo de un código que permitiera la visualización de la información que se está recibiendo de los sensores en tiempo real. Para lograrlo se tomó como base la librería llamada Ploly obtenida de <https://plot.ly/r/>, la cual permite la gráfica en tiempo real de estos datos. En la siguiente imagen se puede apreciar como a través de una consulta a la base de datos se le indica que liste los valores de los campos fecha y potencia de forma descendiente y que tenga un límite de 10 registros.

```
/public_html/contador.php
188 <div class="row">
189   <div class="col-sm-12">
190     <div class="panel panel-primary">
191       <div class="panel-heading"><h4 class="text-center">GRÁFICA DE CONSUMO</h4></div>
192     </div>
193     <div class="panel panel-body">
194       <div class="row">
195         <div class="col-sm-12">
196           <div id="graficalineal"></div>
197           <?php
198             $sql = "SELECT fecha, potencia FROM lecturas ORDER BY fecha DESC LIMIT 10";
199             $result = mysqli_query($con, $sql);
200             $valoresX = array(); //Fechas
201             $valoresY = array(); //Potencia
202             while ($ver = mysqli_fetch_row($result)) {
203               $valoresY[] = $ver[1];
204               $valoresX[] = $ver[0];
205             }
206
207             $datosX = json_encode($valoresX);
208             $datosY = json_encode($valoresY);
209             <?
210             <script type="text/javascript">
211               function crearCadenaLineal($json) {
212                 var nParsed = JSON.parse($json);
```

Figura 30 Sentencia MySQL para llamar datos de consumo

Tanto la fecha como la potencia toman los valores de los ejes x y y, los cuales se adjuntan al código de la gráfica y que finalmente nos da como resultado el muestreo de los datos que llegan de microprocesador y los muestre de la siguiente forma.

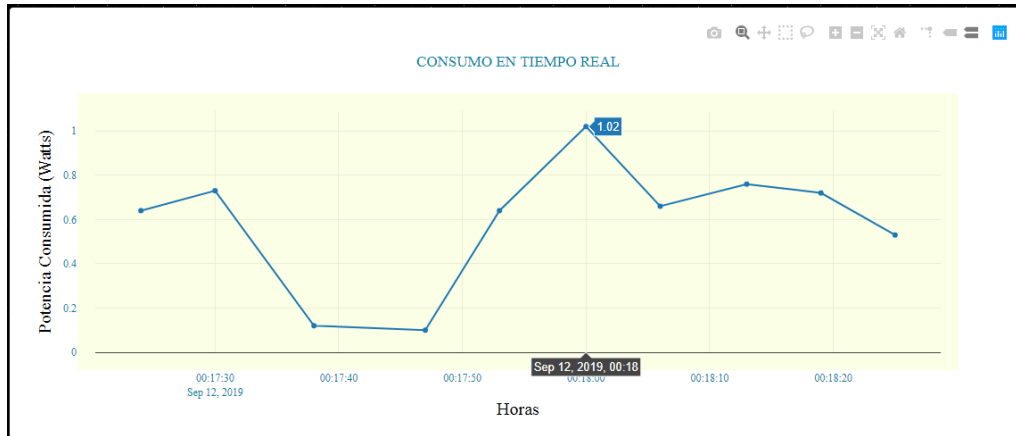


Figura 31 Grafica de consumo en tiempo real

Esta grafica le será muy útil al usuario ya que en la misma primero puede apreciar que el módulo está en funcionamiento tomando las lecturas, y principalmente en ella se pueden apreciar en que momentos ocurren picos de potencia cuando se genera un consumo fuera de lo normal.

3 RESULTADOS

Las mediciones realizadas en el proyecto se obtuvieron con una serie de procesos calibración que a la vez eran comparadas con la medición real dentro de la toma eléctrica y el circuito diseñado para la obtención de pruebas, las cuales se mostraran a continuación.

3.1 Calibración del sensor de corriente SCT-013

Para conseguir un valor de calibración acorde a las lecturas de amperaje suministradas por la acometida eléctrica diseñada se debe conocer las características técnicas del sensor y como se genera la toma de las lecturas.

El sensor SCT-013 maneja una toma de lecturas que son reconocidas a través de los pines analógicos de la tarjeta microcontroladora. El sensor refleja lecturas que son recibidas desde un rango de 0 a 1024, siendo el valor máximo correspondiente con el voltaje de entrada máximo del microcontrolador que es de 5V.

La lectura de un amperaje alterno para obtener un valor de corriente eficaz (RMS), requiere generar un muestreo de valores tanto positivos como negativos, la tarjeta microcontroladora solo tiene la capacidad de recibir valores positivos motivo por lo cual las lecturas que tomen los valores negativos de la señal serán siempre tomados como cero.

La solución implementada consistió en diseñar un circuito OFF SET que permitió generar el corrimiento de la señal recibido con la finalidad de poder ser medida dentro del rango de voltaje de entrada máximo permitido (0-5V).

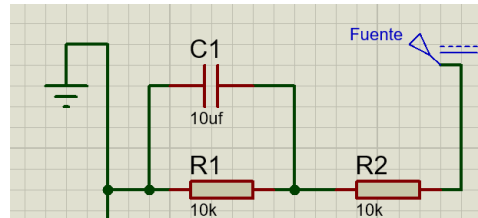


Figura 32 Circuito OFF SET implementado

Por ende, luego de diseñar el circuito OFF SET fue necesario generar un código programación que permitió tomar la lectura del pin analógico y que fueran visualizadas a través del monitor serie.

```
sensor=analogRead(A0);  
Serial.println(sensor);
```

La aplicación de las líneas de mostradas en la anterior figura permitió capturas los valores del sensor y luego imprimirlos en el monitor serial de la aplicación Arduino.

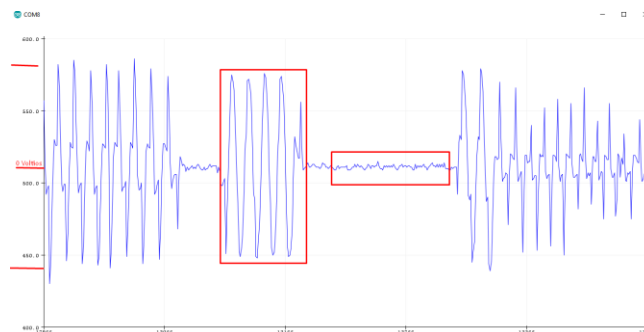


Figura 33 Señal reflejada en el monito serie.

Como se puede visualizar en la figura, el corrimiento permite obtener los valores de la señal de amperaje determino la ausencia de corriente en un rango de una toma de lectura de 512 que interpretado como un valores de voltaje de entrada de Arduino toma un valor de 2.5 voltios, es decir el punto medio por lo cual se pudo ver reflejado en el monitor serial tanto los valores positivos como negativos de la señal de entrada porque la señal fue acoplada al rango de medición de la tarjeta Arduino.

3.2 Calibración del sensor de voltaje ZMPT101B

La ventaja del sensor de voltaje que se implementó por sobre el sensor de amperaje , fue que las especificaciones técnicas del datasheet permitieron aclarar que las lecturas tomadas por el mismo permitieron identificar que en la entrada analógica de nuestra placa los valores de voltaje del sensor que van de los 0 voltios a máximo 250 voltios, podían ser interpretados en la entrada analógica como un valor de 512 para una lectura de 0 voltios y de 1024 para un valor máximo de 250 voltios por lo cual no fue necesario implementar un circuito OFF SET.

```
Sensor = analogRead(A1)  
Serial.print( Sensor );
```

Visualizar la señal de voltaje a través del monitor serie se logra mediante el uso de las dos líneas mostradas anteriormente, en la primera línea toma una lectura de la entrada analógica y la segunda línea imprime la información en el monitor serie.

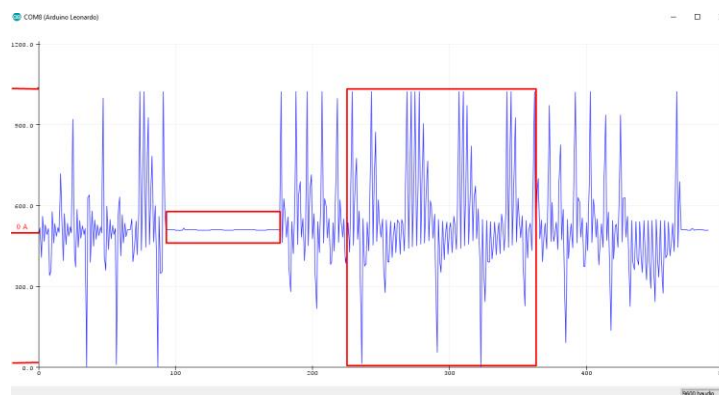


Figura 34 Señal del voltaje reflejada en el monitor serie.

La medición del voltaje obtenido permitió observar que los rangos de los picos de la señal tanto positivos como negativos manejaban un punto intermedio en el valor asignado como 0 voltios comprendido en la entrada analógica como 512.

3.3 Validación de datos capturados VS la medición con multímetro

Luego de haber realizado la calibración de los sensores de medición que se utilizaron es necesario corroborar cual el porcentaje de error, para ello se toman una serie de muestras tanto del sensor de tensión como del sensor de amperaje y se genera un comparativo con una medición hecha a través de un dispositivo fiable.

Tabla 4 Tabla comparativa de medición y margen de error para valores de voltaje (Voltios).

MARGEN DE ERROR LECTURAS DE VOLTAJE			
No. de muestra	Valor de referencia (Multímetro)	Valor Censado (ZMPT101B)	Margen de Error
1	0,4	0,9	12,50%
2	42	41,9	0,24%
3	46,9	45,5	2,99%
4	49,3	48,6	1,42%
5	56,7	56,8	0,18%
6	63,2	63,8	0,95%
7	69	69,08	0,12%

8	73,4	72,24	1,58%
9	79,3	79,66	0,45%
10	88,3	89,27	1,10%
11	93,5	93,7	0,21%
12	96,7	96,32	0,39%
13	100,8	100,69	0,11%
14	102,5	102,13	0,36%
15	107,4	106,97	0,40%
16	112,4	111,9	0,44%
17	114,9	115,05	0,13%
18	116,9	117,08	0,15%
19	117,4	117,62	0,19%

Tabla 5 Tabla comparativa de medición y margen de error para valores de corriente (Amperios).

MARGEN DE ERROR LECTURAS DE CORRIENTE			
No. de muestra	Valor de Referencia (Multímetro)	Valor Censado (SCT-013)	Margen de Error
1	0	0,18	10,00%
2	0,2	0,37	4,59%
3	0,3	0,51	4,12%
4	0,4	0,61	3,44%
5	0,5	0,67	2,54%
6	0,6	0,76	2,11%
7	0,7	0,84	1,67%
8	0,8	0,91	1,21%
9	0,9	0,95	0,53%

3.4 Transmisión de los datos obtenidos a través de la red GSM

Se realizaron una serie de pruebas, mediante las cuales se pudo determinar la recepción de la información enviada, durante el proceso de construcción del servidor

de alojamiento de los datos se necesitó generar unas pruebas de envío que pudieran comprobar el funcionamiento del módulo.

La primera prueba realizada consistió en realizar una captura de datos y transmitirla a un servidor de alojamiento de datos gratuito bajo el dominio <https://thingspeak.com/>, esto permitió corroborar el funcionamiento correcto del módulo.

API Requests

Write a Channel Feed

```
https://api.thingspeak.com/update?api_key=945VSQ4DFNOGYKX7&field1=0
```

Figura 35 Estructura para la escritura de los datos en Thingspeak.com

Luego de ingresar los datos para la estructura de la información en el código de programación de Arduino, se compilan los datos con los cual se genera la transmisión de los datos. Tal como se evidencia en la gráfica mostrada dentro de la página en la figura 35, con las lecturas recibidas en un rango de tiempo determinado.

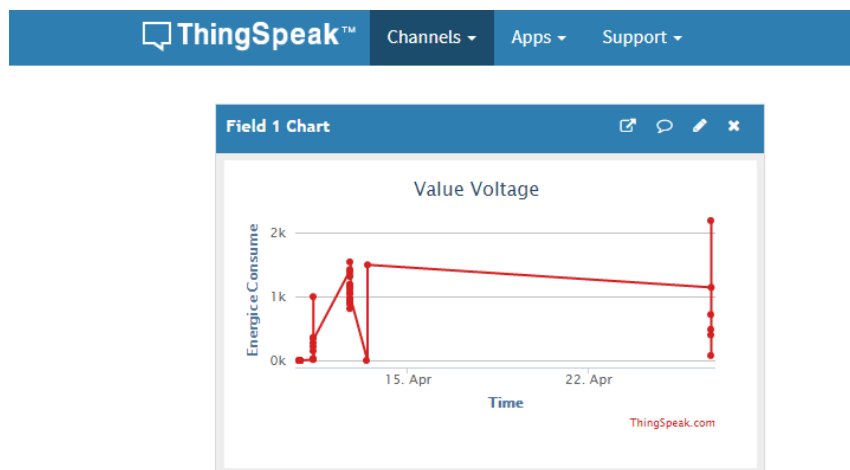


Figura 36 Gráfica de los datos recibidos en Thingspeak.com

Al corroborar esta información se procede a enviar la información al servidor dedicado diseñado para el almacenamiento de datos haciendo uso de una estructura similar al utilizado para la transmisión de datos en Thingspeak.com.

```
COM8 (Arduino Leonardo)
AT+SAPBR=3,1,"Contype","GPRS"
OK
AT+SAPBR=3,1,"APN","@cmoviletb.net.co"
OK
AT+SAPBR=1,1
OK
AT+SAPBER=2,1
ERROR
AT+CCLK?
+CCLK: "19/10/03,19:04:49-20"

OK
AT+HTTPINIT
OAT+HTTPPARA="CID",1
OK
Voltaje= 5.67
Potencia = 0.04 Irms = 0.01
AT+HTTPPARA="URL","http://www.modulomedicion.online/lecturas.phAT+HTTPACTION=0
OK
+HTTPACTION: 0,200,198
Voltaje= 17.87
Potencia = 18.41 Irms = 1.03
AT+HTTPPARA="URL","http://www.modulomedicion.online/lecturas.phAT+HTTPACTION=0
OK
```

Figura 37 Visualización en el monitor serial de la estructura de transmisión de datos al servidor dedicado.

Como se evidencia en la figura 36, se recibe una respuesta positiva al encapsulamiento de los datos y la transmisión de estos a la URL parametrizada.



Figura 38 grafica en tiempo real de los datos recibidos en el hosting dedicado.

Finalmente, los datos transmitidos logran ser recibidos en el servidor de alojamiento para su posterior tratamiento y con ello la visualización del consumo en Watts de manera correcta, en lo referente a tasas de transmisión de datos estas alcanzan velocidades de hasta 85.6 Kbps (SIMCom, 2013), de acuerdo a la capacidad técnica del módulo SIM8001 sobre la banda de frecuencia ofrecida por el operador móvil seleccionado (ETB), el cual opera sobre la banda de frecuencia de los 1900 MHz, banda de operación del módulo de conexión implementado en la realización del prototipo.

4 DISEÑO FÍSICO Y COSTOS

4.1 Diseño

El módulo se compone de una placa PBC personalizada la cual cuenta con tres entradas, una para la alimentación, otra para el sensor de corriente y una más para el sensor de voltaje.

El diseño de la placa PCB requerida fue realizado mediante el uso de una herramienta con licencia libre conocida bajo el nombre de KiCad; mediante esta se logró diseñar el esquemático del circuito que era necesario implementar en el diseño del prototipo. Como se aprecia en la figura 38, el esquema mostrado comprende las distintas conexiones que se unen a los pines de la tarjeta Arduino, tanto para suministro de energía como pines utilizado en la toma de datos y comunicación serial con el módulo de transmisión de datos. Lo siguiente a tener en cuenta fue la organización de las distintas conexiones y elementos utilizados dentro del espacio físico reservado para su creación, lo cual fue posible haciendo uso de una herramienta de diseño de tarjetas incluidas en la suite de PCB.

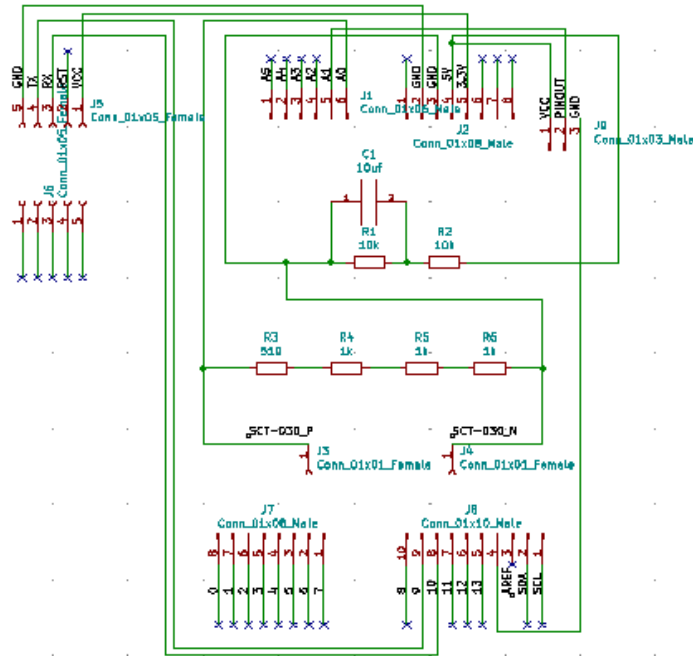


Figura 39 Diseño del esquema electrónico de la PCB

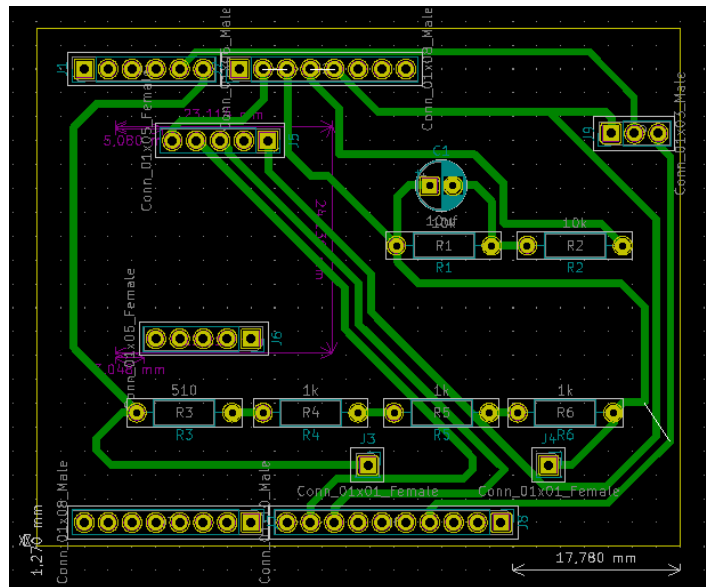


Figura 40 Diseño físico de la PCB

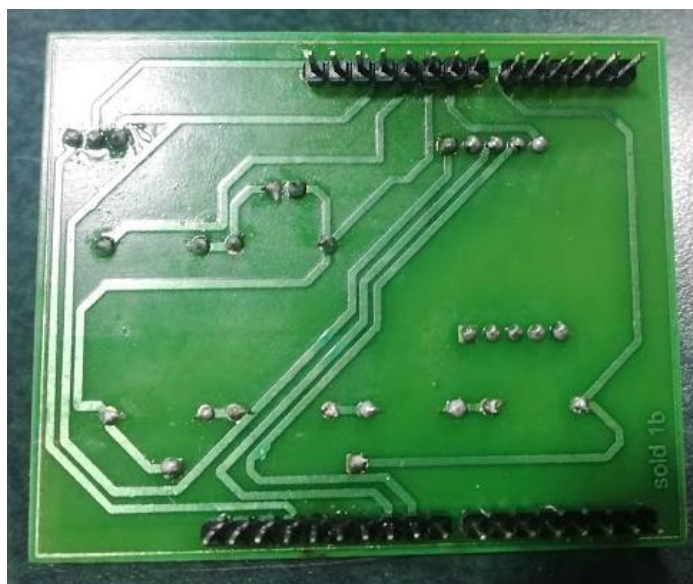


Figura 41 Circuito impreso finalizado

Por último, se obtiene el diseño realizado de manera física visto en la figura 35, en el cual es apreciable los componentes utilizados comprendidos por la resistencia de carga, el circuito off set y los pines de conexión con el módulo SIM800l, la placa Arduino y los sensores de medición de corriente y voltaje; así como también el diseño de las pistas para la interconexión entre cada uno de los elementos del circuito final.

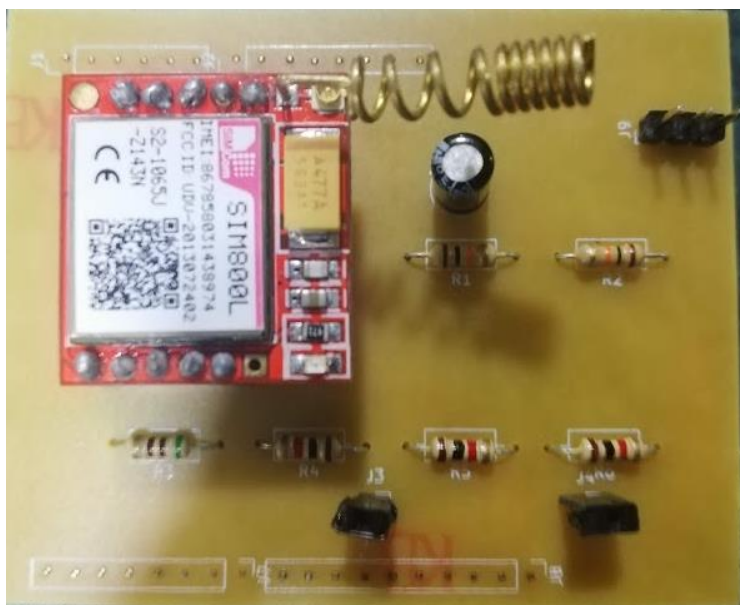


Figura 42 Elementos que componen el circuito instalados

Para el diseño de la caja se realizó en acrílico transparente la cual cuenta con tres entradas, a continuación, se muestra algunas imágenes de su diseño por medio de la aplicación Google Sketchup.

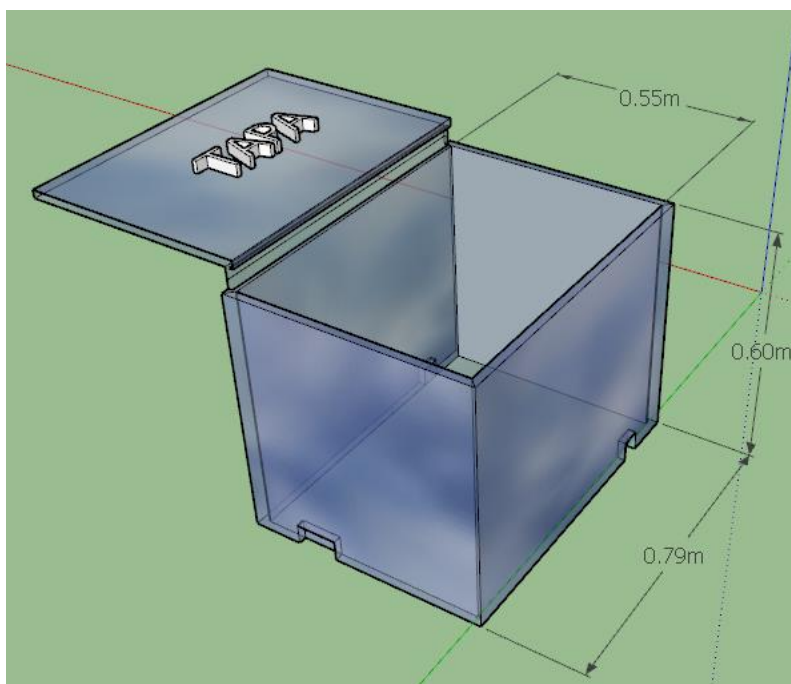


Figura 43 Diseño digitalizado de la caja



Figura 44 Diseño final de caja



Figura 45 Diseño final de la caja en acrílico.

4.2 Costos

A continuación, se ven reflejados los costos totales que se requirieron para el diseño del prototipo.

Tabla 6 Costo de fabricación del prototipo diseñado para la realización del proyecto.

Cantidad	Componente	Costo
1	Tarjeta Arduino	\$ 80.000
1	Módulo SIM800L	\$ 30.000
1	Sensor de Corriente SCT-013	\$ 30.000
1	Sensor de Voltaje ZMPT101B	\$ 25.000
1	Cubierta de acrílico personalizada	\$ 20.000
1	Placa PCB personalizada	\$ 20.000
1	Condensador electrolítico	\$ 500
1+	Jumpers	\$ 10.000
1+	Resistencias	\$ 2.000
2	Bombillas incandescentes	\$ 4.000
1	Tarjeta SIM	\$ 50.000
1	Servicio hosting	\$ 50.000
1	Dimmer	\$ 25.000

Total	\$ 346.500
--------------	------------

5 CONCLUSIONES

La contribución principal del siguiente trabajo tenía como pilar principal el diseño y realización de un prototipo con la capacidad de obtener mediciones de amperaje y tensión eléctrica que permitieran la obtención de una potencia aparente que permitiera el modelamiento de un esquema de obtención de consumo eléctrico de energía en tiempo real que fuera transmitido a través de la infraestructura de red móvil actual, mediante tecnología GSM a un servidor dedicado al almacenamiento, procesamiento y visualización de la información obtenida del prototipo.

Dentro del presente trabajo investigativo se corrobora la toma de las mediciones haciendo uso de los conceptos de corriente y tensión eléctrica alterna para las tareas de calibración de los sensores utilizados, gracias al uso de cálculos teóricos que permitieron el diseño eléctrico requerido para el funcionamiento del prototipo y la reducción del margen de error en comparación con los instrumentos de medición convencionales (multímetro, pinza amperimétrica).

Se evidencio la toma correcta de los datos y su transmisión a través de la red móvil GSM mediante el uso del monitor serial a través del Arduino, así como el protocolo de comunicación AT y los servicios de bases de datos implementados en el diseño del servidor dedicado, los cuales fueron posibles mediante el uso de los conceptos y aplicación de las estructuras PHP, en integración con MySQL1

La visualización de la página web integrada a la base de datos fue posible a través del manejo de conceptos de programación y desarrollo fundamentales para su funcionamiento como Javascript, CSS, librerías interactivas como Plotly, las cuales permiten que el usuario final interactúe de forma más sencilla con los datos obtenidos del prototipo.

El diseño de la tarjeta PCB requerida cumple con lo planificado dentro de los esquemáticos del diseño y funcionamiento de esta al integrar los circuitos y componentes requeridos dentro del espacio contemplado para su integración con la placa Arduino, así mismo los sensores de medición utilizados son acordes con las especificaciones técnicas y de operatividad informadas por el fabricante lo que dio buen término a los valores censados.

De la misma manera la estructura implementada en el diseño y desarrollo del presente proyecto se puede abarcar en buena medida en otros campos de la industria como el sector médico a través de tele medición, los sistemas de control SCADA, así como en el ámbito de agricultura donde se cubren grandes distancias, todo esto es posible con la estructura principal del presente trabajo.

Como trabajo de mejora futura se planea el diseño en la página web de un espacio donde puede ser observable el valor tarifario correspondiente al consumo promedio generado durante el mes por los usuarios en kWh, partiendo del valor base manejado (\$/kWh), por las compañías energéticas en la actualidad. Así mismo la medición busca ser optimizada con la finalidad de tener la capacidad de hacer lecturas sobre acometidas con conexión eléctrica trifásica, la cual es implementada en el sector

industrial debido al enorme consumo energético de los elementos utilizados, lo que permitiría a estas industrias tener mayor control sobre el consumo energético realizado, como también sobre la cantidad de energía que retribuyen a las compañías debido al uso de grandes motores que funcionan a partir del uso de grandes embobinados eléctricos.

6 CAPITULO BIBLIOGRAFIA

- Garrigós, J. (2011). *Introducción a la electricidad*. I.E.S. Aandrés de Vandelvira.
- Arduino. (s.f.). *Arduino*. (Arduino) Recuperado el 19 de Octubre de 2019, de <https://store.arduino.cc/usa/leonardo>
- Casademont, J. (2010). *Redes de comunicaciones. De la telefonía móvil a internet*. Barcelona, España: Edicions de la Universitat Politècnica de Catalunya.
- Condor Tinoco, E., & Soria Solis, I. (2019). *Programación Web con CSS, JavaScript, PHP y AJAX*. Universidad Nacional José María Argueda.
- Drof, R. C., & Svoboda, J. A. (2011). *Circuitos Electricos*. Mexico D.F.: Alfaomega Grupo Editor, S.A.
- Equipo Vértice. (2009). *Diseño básico de páginas web en HTML*. España: Publicaciones Vértice.
- Fossati, M. (2018). *Introducción a PHP y HTML*.
- Huidobro Moya, J. M. (2019). *Comunicaciones Móviles. Sistemas GSM, UMTS y LTE*. Ra-Ma Editorial.
- Íñigo Griera, J., Barceló Ordinas, J. M., Cerda Alabern, L., Peig Olivé, E., Abella i Fuentes, J., & Corral i Torruella, G. (2008). *Estructura de Redes de Computadores*. Barcelona: Editorial UOC.
- Llamas, L. (24 de Enero de 2017). *Luis LLamas*. (Creative Commons License BY-NC-SA.) Recuperado el 19 de Octubre de 2019, de <https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>
- Naylamp Mechatronics. (s.f.). *Naylamp Mechatronics*. Recuperado el 2019 de Octubre de 2019, de <https://naylampmechatronics.com/sensores-corriente-voltaje/393-transformador-de-voltaje-ac-zmpt101b.html>
- Pacheco Escribano, L. F. (2015). *JavaScript. Ejercicios tutorizados*. Lulu.com.
- Ramírez Castaño, S. (2004). *Redes de Distribución de Energía*. Manizalez: Centro de Publicaciones Universidad Nacional de Colombia.
- Ruelas Gómez, Roberto. (s.f.). *Ruel S. A. de C. V*. Recuperado el 19 de 10 de 2019, de <https://www.ruelsa.com/notas.html>;
https://www.ruelsa.com/notas/rt/rt126_factordepotencia.pdf

SIMCom. (2013). SIM800L Hardware Design. Shanghai: SIMCom Wireless Solutions Ltda.

Thibaud, C. (2006). *Recursos Informáticos MYSQL 5*. Barcelona: Ediciones ENI.

Por intermedio del presente documento en mi calidad de autor o titular de los derechos de propiedad intelectual de la obra que adjunto, titulada **Estudio y Diseño de Módulo de Transmisión Autónoma de Lectura de Contadores de Energía Eléctrica Integrado a IoT**, autorizo a la Corporación universitaria Unitec para que utilice en todas sus formas, los derechos patrimoniales de reproducción, comunicación pública, transformación y distribución (alquiler, préstamo público e importación) que me corresponden como creador o titular de la obra objeto del presente documento.

La presente autorización se da sin restricción de tiempo, ni territorio y de manera gratuita. Entiendo que puedo solicitar a la Corporación universitaria Unitec retirar mi obra en cualquier momento tanto de los repositorios como del catálogo si así lo decido.

La presente autorización se otorga de manera no exclusiva, y la misma no implica transferencia de mis derechos patrimoniales en favor de la Corporación universitaria Unitec, por lo que podré utilizar y explotar la obra de la manera que mejor considere. La presente autorización no implica la cesión de los derechos morales y la Corporación universitaria Unitec los reconocerá y velará por el respeto a los mismos.

La presente autorización se hace extensiva no sólo a las facultades y derechos de uso sobre la obra en formato o soporte material, sino también para formato electrónico, y en general para cualquier formato conocido o por conocer. Manifiesto que la obra objeto de la presente autorización es original y la realicé sin violar o usurpar derechos de autor de terceros, por lo tanto, la obra es de mi exclusiva autoría o tengo la titularidad sobre la misma. En caso de presentarse cualquier reclamación o por acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión asumiré toda la responsabilidad, y saldré en defensa de los derechos aquí autorizados para todos los efectos la Corporación universitaria Unitec actúa como un tercero de buena fe. La sesión otorgada se ajusta a lo que establece la ley 23 de 1982.

Para constancia de lo expresado anteriormente firmo, como aparece a continuación.

Firma



Nombre Yeison Otavo Vargas
CC. 1.012.422.108



Nombre Javier Rojas Cárdenas
CC. 1.015.404.154