



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Desarrollo de un sistema para el
monitoreo del consumo eléctrico usando
internet de las cosas**

TESIS
QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN
MÓDULO DE HARDWARE

P R E S E N T A

JULIO CÉSAR MARTÍNEZ CASTILLO

DIRECTOR DE TESIS

M. EN C. JESÚS CRUZ NAVARRO



Ciudad Universitaria, Cd. Mx. , AÑO 2020



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas

Tesis Digitales

Restricciones de uso

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradezco a

... a mi Madre y a mi Padre, por convertirme en el hombre que soy actualmente.

... a ti Ely, por siempre estar en los momentos más importantes a mi lado.

... a mí asesor Jesús, por toda su ayuda y consejos.

... A mis sinodales, por sus sabios consejos durante este profeso y ayuda en cada momento.

...a todo el laboratorio de BioRobótica, por todo lo que me ha entregado en este tiempo.

... al equipo Justina, por toda su amistad incondicional.

... a CONACyT, a través del proyecto 245491, "Laboratorio de Movilidad e Infraestructura Verde para la Eficiencia Energética en Ciudades", por el apoyo recibido en la realización de este documento.

Contenido

1 Introducción.....	11
1.1 Objetivo.....	16
1.2 Planteamiento del problema.....	17
1.3 Hipótesis.....	17
1.4 Motivación.....	17
1.5 Descripción del documento.....	18
2 Marco teórico.....	19
2.1 Medición de la corriente por medio del sensor.....	19
2.1.1 Física acerca del sensor.....	19
2.1.2 Protecciones y manejo del sensor.....	24
2.1.3 Explicación e implementación del sensor.....	27
2.2 Microcontroladores, Microcomputadoras y componentes electrónicos.....	29
2.2.1 Microcomputadoras (Raspberry Pi 3 B).....	30
2.2.2 Microcontroladores (Arduino Nano).....	34
2.2.3 Manejo de señales digitales y analógicas (ADC).....	36
2.2.4 Regulador de voltaje.....	38
2.3 Arquitectura de los sistemas IoT.....	39
2.3.1 Dispositivos embebidos con internet de las cosas.....	41
2.3.2 Servicios Web (RESTful).....	45
2.3.3 Uso de base de datos.....	49
2.3.4 Sistema de visualización de datos.....	51
3 Desarrollo.....	53
3.1 Arquitectura del sistema.....	53
3.2 Dispositivo embebido.....	55
3.2.1 Sensor de gancho.....	57
3.2.2 Control de la fuente de alimentación.....	60

3.2.3 Electrónica del sistema embebido.....	63
3.2.4 Cálculos de la corriente eléctrica.....	66
3.2.5 Interconexión Arduino-Rapsberry.....	67
3.2.6 Software del sensor de corriente.....	69
3.3 Sistema de adquisición de datos (API).....	71
3.3.1 Comunicación dispositivo - API con Flask y Gevent.....	71
3.3.2 Base de datos con MongoDB.....	75
3.3.3 Algoritmos de búsqueda.....	76
3.3.4 Cálculo del consumo.....	78
3.3.5 Protocolos de respuesta.....	80
3.4 Sistema de visualización.....	80
3.4.1 Sistema gráfico de los datos.....	81
3.4.2 Búsqueda y filtrado de datos.....	86
3.4.3 Validación de los datos desde la vista.....	89
4 Pruebas y resultados.....	91
4.1 Pruebas en ambiente controlado.....	91
4.1.1 Análisis lineal con el foco incandescente.....	92
4.1.2 Análisis no lineal con el foco fluorescente.....	95
4.2 Comparación del sistema completo con un multímetro de gancho comercial.....	98
4.3 Pruebas con el sistema en ambiente real.....	103
5 Conclusiones y trabajo futuro.....	111
5.1 Conclusiones.....	111
5.2 Trabajo futuro.....	113
6 Bibliografía.....	117

Índice de figuras

Figura 1.1 Crecimiento de generación bruta de electricidad 2017-2018.....	12
Figura 1.2 Porcentajes de generación bruta durante el 2018.....	14
Figura 2.1 Funcionamiento de un sensor eléctrico.....	20
Figura 2.2 Ejemplo de un sensor de corriente.....	22
Figura 2.3 Diagrama funcional de un transformador.....	23
Figura 2.4 Conexión de la resistencia de carga.....	24
Figura 2.5 Integrado LM358.....	26
Figura 2.6 Manejo de un sensor de corriente no invasivo.....	28
Figura 2.7 Diagrama de una Raspberry PI 3.....	31
Figura 2.8 Pinout de la Raspberry Pi 3B.....	32
Figura 2.9 Estructura del sistema operativo Raspian basado en Linux.....	33
Figura 2.10 Diagrama básico de un microcontrolador.....	35
Figura 2.11 Arquitectura Básica de un arduino nano.....	36
Figura 2.12 Diagrama de un proceso de conversión analógico/digital.....	37
Figura 2.13 Diagrama de una fuente regulada.....	38
Figura 2.14 Arquitectura IoT.....	41
Figura 2.15 Diagrama físico de un dispositivo embebido.....	42
Figura 2.16 Diagrama de Capas de internet de las cosas.....	44
Figura 2.17 Diagrama de funcionamiento de un Web Service.....	46
Figura 2.18 Diagrama de funcionamiento de un servicio RESTful.....	49
Figura 2.19 Diagrama de funcionamiento de una base de datos.....	50
Figura 2.20 Interacción de una base de datos con una API.....	51
Figura 2.21 Representación de un gráfico.....	52
Figura 3.1 Arquitectura del sistema.....	55
Figura 3.2 Diagrama completo del dispositivo embebido.....	56
Figura 3.3 Conexión del sensor de corriente no invasivo.....	57
Figura 3.4 Sensor SCT-013-030.....	59
Figura 3.5 Foto de regulador marca pololu reg15a.....	60
Figura 3.6 Módulo de carga Adafruit Powerboost 1000C.....	62
Figura 3.7 Diagrama de alimentación del sensor de corriente.....	62
Figura 3.8 Diagrama del sensor al microcontrolador.....	63

Figura 3.9 Diagrama de conexión simbólico del LM358 con el sensor SCT-013-030.....	64
Figura 3.10 Conexión del sensor SCT-013-030 con el LM358.....	65
Figura 3.11 Conexión por pines serial.....	69
Figura 3.12 Conexión por USB serial.....	69
Figura 3.13 Flujo de peticiones web del sistema.....	72
Figura 3.14 Estructura de una función de búsqueda en PyMongo.....	77
Figura 3.15 Estructura de una función de agregación en PyMongo.....	78
Figura 3.16 Ejemplo de Bootstrap.....	82
Figura 3.17 Vista previa de la interfaz AdminLTE.....	82
Figura 3.18 Consulta del consumo eléctrico histórico.....	83
Figura 3.19 Gráfica en tiempo real.....	84
Figura 3.20 Cálculo de la tarifa por parte de CFE.....	85
Figura 3.21 Vista previa de la gráfica de historial.....	87
Figura 3.22 Vista previa del cálculo de la tarifa según lineamientos CFE.....	89
Figura 4.1 Conexión del foco resistivo.....	93
Figura 4.2 Gráfica resultante con el foco resistivo.....	93
Figura 4.3 Consumo del foco resistivo visto desde el sistema.....	94
Figura 4.4 Conexión del foco fluorescente.....	95
Figura 4.5 Diferencia entre señales con y sin armónicas.....	96
Figura 4.6 Gráfica resultante con el foco fluorescente con armónicos.....	97
Figura 4.7 Consumo del foco incandescente visto desde el sistema.....	98
Figura 4.8: Conexión del sistema al taller.....	99
Figura 4.9 Multímetro profesional de gancho RMS con auto rango marca Steren.....	100
Figura 4.10 Gráfica histórica del 31 de Agosto de 2019.....	100
Figura 4.11 Gráfica histórica del 01 de Septiembre de 2019.....	101
Figura 4.12 Gráfica comparativa del 31 de Agosto de 2019.....	102
Figura 4.13 Gráfica comparativa del 01 de Septiembre de 2019.....	103
Figura 4.14 Cableado eléctrico de la casa.....	104
Figura 4.15 Instalación del dispositivo.....	106
Figura 4.16 Periodo de lecturas del sensor de corriente.....	107
Figura 4.17: Gráfica comparativa entre los medidores.....	110

Índice de fórmulas

Fórmula 2.1 Relación de intensidad con el número de espiras.....	23
Fórmula 2.2 Cálculo de Voltaje pico.....	26
Fórmula 2.3 Cálculo de voltaje pico a pico.....	27
Fórmula 2.4 Conversión de corriente eléctrica a valor digital.....	28
Fórmula 2.5 Cálculo de la constante K.....	29
Fórmula 2.6 Regulación de carga.....	39
Fórmula 2.7 Porcentaje de regulación.....	39
Fórmula 3.1 Cálculo de cambio de escala.....	66
Fórmula 3.2 Conversión a corriente.....	66
Fórmula 3.3 Corriente RMS en el dominio continuo.....	67
Fórmula 3.4 Corriente RMS en el dominio discreto.....	67
Fórmula 3.5 Cálculo de las diferencias de las fechas.....	78
Fórmula 3.6 Suma de corrientes RMS.....	79
Fórmula 3.7 Cálculo de la energía eléctrica consumida en Watts-milisegundos.....	79
Fórmula 3.8 Calculo de energía eléctrica consumida en Wh.....	79
Fórmula 3.9 Conversión a kWh.....	80
Fórmula 3.10 Costo total por kilowatt-hora.....	89
Fórmula 4.1 Cálculo de la corriente de un foco.....	92
Fórmula 4.2 Valor del consumo eléctrico con el medidor digital.....	105

1 Introducción

La Comisión Federal de Electricidad (CFE, fundada en 1937) ha llevado acabo la labor de generar y distribuir electricidad por todo México. Pese a todos los obstáculos que esto conlleva, se busca beneficiar a las personas dentro de nuestro país con este servicio. Al paso de los años, se han creado diferentes plantas que generan electricidad por medio de los recursos naturales como los ríos y corrientes de aire, así como la implementación de máquinas a base de combustibles fósiles para generar electricidad.

Actualmente, México cuenta con diferentes plantas generadoras de energía eléctrica y cada una de ellas tiene un tipo de funcionamiento con el cual genera electricidad. Estos tipos de generadores de electricidad que se encuentran en el país son (Sector Eléctrico Nacional, 2016):

- Plantas termoeléctricas con ciclo de vapor.
- Plantas termoeléctricas con turbogas.
- Plantas termoeléctricas con combustión interna.
- Plantas carboeléctricas.
- Plantas geotermoeléctricas.
- Plantas de ciclo combinado
- Plantas duales que operan con carbón o combustóleo.
- Plantas nucleoeleéctricas.
- Plantas hidroeléctricas.
- Plantas eoloeléctricas.

- Plantas que funcionan con celdas fotovoltaicas.

Según el informe anual dado por CFE (CFE, 2018), durante el 2018, el conjunto de plantas generadoras de parte de la empresa produjo una cantidad mayor de 258 mil 684 gigawatts-hora, que significa un aumento del 0.49% comparado al del año 2017, como se observa en la Figura 1.1. En la figura también podemos observar que la mayor parte de producción de energía eléctrica la realizan las plantas generadoras cuyo funcionamiento es a partir de combustibles que alimentan a la maquinaria.

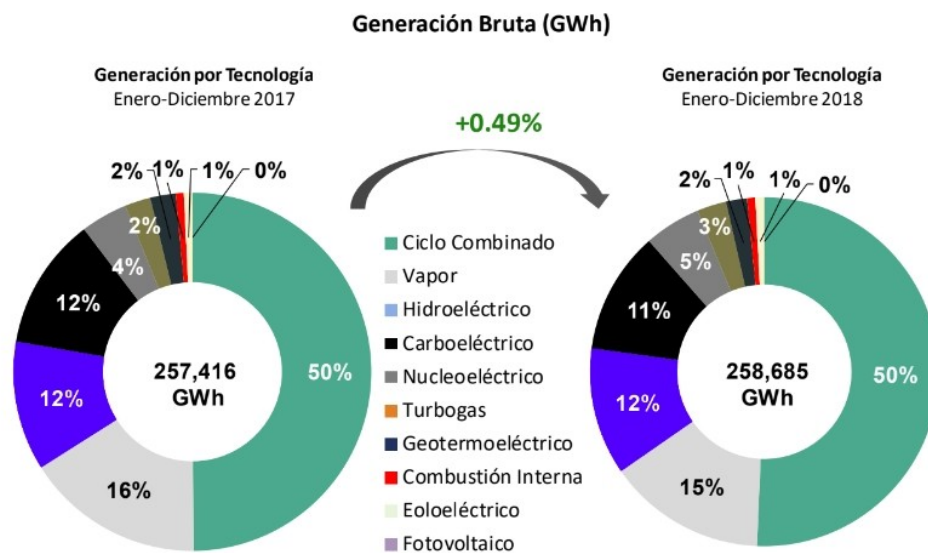


Figura 1.1 Crecimiento de generación bruta de electricidad 2017-2018.

Imagen recuperada de https://infosen.senado.gob.mx/sgsp/gaceta/64/1/2019-05-15-1/assets/documentos/PoderEjecutivo_informe_anual_2018_cfe.pdf pagina 8.

Sin embargo, las plantas en general que utilizan maquinas de combustión son las que producen gases nocivos para el medio ambiente como dióxido de carbono (CO_2), óxidos de nitrógeno (NO_x) y bióxido de azufre (SO_2). Algunas de las consecuencias de la generación de estos gases se muestran en la Tabla 1.1 donde los daños más considerables que se tienen son daños hacia los mares, directamente hacia su flora y fauna, la generación de la lluvia ácida, el deterioro de ecosistemas como lo son los bosques y deterioros a la salud de las personas.

La quema de combustibles fósiles se ha convertido en una opción para producir electricidad de manera sencilla. Esta técnica se usa frecuentemente por las empresas para la generación de energía electricidad. El combustible que utilizan más dentro las plantas

1 Introducción

generadoras es el que utiliza ciclo combinado, ya que son una opción sencilla para instalarse. Los insumos que requieren estas plantas tiene un costo accesible comparándolo con el costo invertido en una planta generadora cuyo funcionamiento se basa en la implementación de energías renovables.

Contaminante	Descripción	Efectos en el ambiente
CO ₂	El dióxido de carbono es un gas que se encuentra naturalmente en la atmósfera, así como resultado de la quema de combustibles fósiles.	<ul style="list-style-type: none">• Contribuye al efecto invernadero• Lluvia ácida• Deterioro de flora y fauna acuática
NO _x	Los óxidos de nitrógeno se generan por el uso de vehículos y plantas termoeléctricas generadoras de energía eléctrica.	<ul style="list-style-type: none">• Daños en los pulmones• Lluvia ácida• Generación de contaminación fotoquímica• Daño directo a los bosques
SO ₂	El bióxido de azufre se produce a consecuencia del funcionamiento de plantas termoeléctricas generadoras de electricidad y refinerías.	<ul style="list-style-type: none">• Daños en los pulmones• Lluvia ácida• Deterioro de flora y fauna acuática• Daño directo a los bosques

Tabla 1.1 Consecuencia de la quema de combustibles fósiles

Durante el 2018 en México, la quema de combustibles fósiles abarcó un 80.17% de la producción de energía eléctrica total del país, mientras que tan sólo el 19.83% fue con energías renovables y energías limpias, como lo muestra la Figura 1.2. Con esto, CFE da a conocer que el implementar centrales eléctricas que funcionan a base de los combustibles fósiles es una de sus opciones más rentables para generar en el país.

En los últimos años, la empresa CFE busca hacer el menor daño ambiental causada por la quema de combustibles, con la creación de proyectos cuya infraestructura es dirigida a la generación y transmisión de energía eléctrica limpia por todo el país, con una mejor calidad, costos bajos y con un menor daño al medio ambiente, disminuyendo la producción de CO₂, NO_x y SO₂ los cuales son nocivos para la capa de ozono.

Dado que la producción eléctrica en México se realiza principalmente con fuentes altamente contaminantes, esto desemboca en una alta demanda de generación, y a su vez está conlleva más contaminación al medio ambiente. Esta alta demanda surge a partir de los usuarios que utilizan la electricidad desmedidamente por medio de sus electrodomésticos, haciendo que las centrales eléctricas trabajen de más para cubrir la demanda de energía eléctrica y no cause bajas de tensión al suministro.

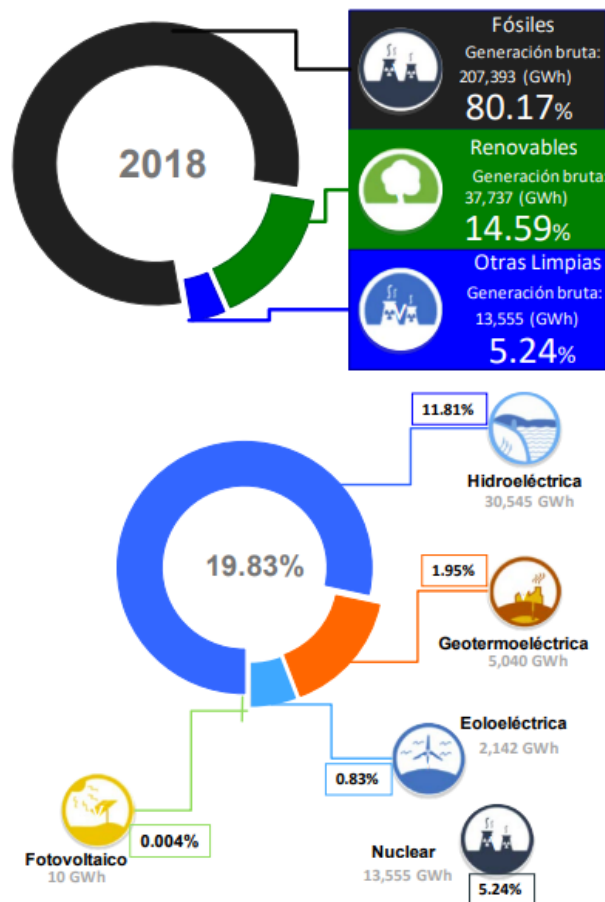


Figura 1.2 Porcentajes de generación bruta durante el 2018.

*Imagen recuperada de
https://infosen.senado.gob.mx/sgsp/gaceta/64/1/2019-05-15-1/assets/documentos/PoderEjecutivo_informe_anual_2018_cfe.pdf pagina 11.*

1 Introducción

Además de contribuir con la contaminación, otro problema que se genera por demanda excesiva de electricidad son los costos elevados que llegan a repercutir en los usuarios, ya que los costos incrementan conforme se aumenta el consumo eléctrico en instalaciones comerciales y residenciales, puesto que existen subsidios gubernamentales a la tarifa. Por lo que CFE ha establecido dos tarifas por consumo dependiendo de la zona de México en la que se habite:

- La primer tarifa subsidiada se aplica dentro de la temperatura media mensual mínima en verano.
- La segunda cuando se rebasa el promedio de la media mensual.

Por tanto, el consumo eléctrico excesivo ha representado una problemática dentro de la sociedad, siendo en parte en cuestiones económicas y en parte cuestiones ambientales, las cuales generan un fuerte impacto dentro de nuestra sociedad mexicana.

Una solución que proponen las compañías de suministro eléctrico es que los usuarios del servicio eléctrico implementen una cultura de ahorro energético en sus hogares y/o negocios; esto para reducir la gran demanda de electricidad hacia las plantas generadoras, ya que estas plantas deben generar una cierta cantidad de electricidad para que no se generen caídas de tensión que dañen a los aparatos eléctricos conectados al cableado de las residencias.

Este tipo de hábitos ayudan a que se reduzca la contaminación causada por la quema de combustibles fósiles y disminuir los costos elevados de los usuarios por su consumo eléctrico. Algunos de los hábitos que la empresa de CFE recomienda hacer para realizar ahorro de energía son (CFE, 2019):

- Utilizar la iluminación natural durante las mañanas y durante las noches utilizar iluminación eléctrica cuando sean necesarias.
- Desconectar aparatos eléctricos que no se estén utilizando.
- Implementar aparatos ahorradores como focos led, o dispositivos con un temporizador para tenerlo suspendido en ciertos tiempos.
- Revisar la instalación eléctrica del hogar.
- Tener un mantenimiento preventivo y correctivo a los electrodomésticos.
- Utilizar correctamente los electrodomésticos como lo indican sus manuales.

El llevar un control del consumo eléctrico para generar ahorros de energía ayuda a disminuir las problemáticas que esto genera. Sin embargo, no todos los usuarios saben como llevar su propio control, o desconocen la forma de leer su medidor de suministro eléctrico e inclusive no cuentan con el tiempo para estar al pendiente del consumo eléctrico durante el día.

Actualmente, la tecnología ha ido evolucionando al paso de los años, hasta tener un gran impacto dentro de la sociedad. Por ejemplo, el internet de las cosas se ha vuelto un término muy común de la vida cotidiana. Esta tecnología permite conectar, por medio de internet, los objetos que utilizamos todos los días como los autos, nuestros electrodomésticos, e inclusive sistemas que interactúan con las casas o negocios. Esto con el fin de monitorear y controlar componentes desde cualquier parte del mundo, dispositivos embebidos con acceso a internet.

Dado lo antes mencionado, se propone un sistema que permita monitorear de manera remota el consumo eléctrico para ayudar a cualquier persona desde cualquier lugar, a consultar su consumo eléctrico aproximado en tiempo real, consultar un histórico en cierto periodo, y obtener un costo aproximado de este consumo, aun sin conocimientos técnicos, para que puedan tomar las debidas acciones preventivas respaldados por información que proviene del sistema.

La implementación masiva de este tipo de sistemas podría incentivar a los usuarios a estar más al pendiente de su consumo y generar sus propias estrategias de ahorro eléctrico, lo que implicaría ahorros económicos para el usuario y una mejora al medio ambiente.

1.1 Objetivo

Desarrollar un dispositivo con tecnología de Internet de las cosas, de fácil instalación, que permita a un usuario consultar y monitorear, en cualquier momento, un estimado de su consumo eléctrico a través de una computadora o un dispositivo móvil por medio de una interfaz gráfica.

Las metas que se plantearon para lograr éste objetivo fueron las siguientes:

- Desarrollar un dispositivo embebido para medir el consumo de energía eléctrica.
- Implementar un sistema que envíe los datos del consumo de corriente desde el dispositivo a un servidor por un medio inalámbrico.
- Desarrollar un sistema que permita el registro, almacenamiento y consulta de los datos vía Internet.
- Desarrollar una interfaz gráfica que permita a un usuario visualizar y analizar los datos registrados en determinados intervalos de tiempo.
- Implementar algoritmos para el filtrado de los datos almacenados.
- Realizar pruebas de campo para validar el uso del sistema.

1.2 Planteamiento del problema

En México, con la instalación de módulos de medición de consumo eléctrico que realiza actualmente la Comisión Federal de Electricidad, no se tiene una forma con la cual los usuarios que no cuentan con conocimientos técnicos puedan consultar y verificar de manera sencilla y amigable el consumo eléctrico dentro de sus casas y negocios.

Esto conlleva a que los usuarios no puedan realizar un control ni una acción preventiva con respecto a su consumo en su hogar. Dado que existen subsidios dentro del costo de la energía eléctrica, esto tiene como consecuencia el pago excesivo de su uso sin un previo aviso; además, la generación eléctrica proveniente de centrales cuyo funcionamiento es por quema de combustibles crea un daño en el ambiente, el cual se incrementa más por el consumo excesivo de la energía eléctrica.

1.3 Hipótesis

Se puede desarrollar un dispositivo embebido con el uso del Internet de las cosas de fácil uso, que le permita a los usuarios monitorear de manera remota con ayuda de una interfaz gráfica, la cual se puede acceder desde sus computadoras y/o dispositivos móviles el consumo eléctrico dentro de sus residencias para que los usuarios tomen acciones preventivas en caso de consumo excesivo.

1.4 Motivación

El presente trabajo se desarrolló dentro del laboratorio de Biorobotica. Este laboratorio participa en el proyecto CONACYT llamado “*Laboratorio de movilidad e infraestructura verde para la eficiencia energética en la ciudades*” con el número de proyecto 245491, el cual consiste en la elaboración de bicicletas mecánicas y eléctricas inteligentes, las cuales se implementaron como una alternativa ecológica de transporte dentro de ciudad universitaria.

En un inicio, la idea de este trabajo nació como una solución para llevar acabo mediciones de manera remota de las estaciones de carga de las bicicletas, puesto que se necesita monitorear cada estación, y observar su rendimiento sin importar en donde se localicen cada una, conforme a la información que es recabada por el sistema.

Sin embargo, se identificó la oportunidad de extender el uso de este sistema a cualquier instalación eléctrica, con la meta de, en un determinado momento, generar un producto comercial. Sin importar esto último se espera que el sistema sea utilizado para el monitoreo del consumo en las estaciones eléctricas del proyecto antes mencionado.

1.5 Descripción del documento

Este trabajo se encuentra organizado de la siguiente manera: dentro del capítulo 2 se presenta teoría básica, conceptos y definiciones que se implementaron para el desarrollo del sistema. En el capítulo 3 se desarrolla con detalle la constitución y el funcionamiento del dispositivo encargado de las lecturas del consumo eléctrico; de igual forma, se detalla el funcionamiento de los servicios web para realizar las acciones de consulta y visualización de los datos recibidos por el dispositivo. En el capítulo 4 se da una explicación con respecto a las pruebas que se llevaron a cabo para ver la calidad de la funcionalidad del sistema completo, comparándolo con otros sistemas comerciales ya existentes, basado en su implementación dentro de un ambiente de uso real. Finalmente, en el capítulo 5 se presentan las conclusiones que se obtuvieron durante las pruebas y se plantea el trabajo futuro.

2 Marco teórico

En este capítulo se menciona la teoría que esta detrás del funcionamiento de las diferentes partes que componen al sistema, que es básicamente el funcionamiento de cada elemento que contiene el dispositivo, en este caso el sensor que utiliza, el microcontrolador que maneja el sensor, y la microcomputadora que realiza el envío de los datos; así como funcionamiento de la parte del software que abarca el servicio web, la base de datos, y el sistema con el cual se visualizan los datos por medio de una página web.

2.1 Medición de la corriente por medio del sensor

Un sensor es un dispositivo que, a partir de la energía del medio donde se mide, da una señal de salida transducible que es función de la variable medida (Pallás, 2004). Existen diferentes tipos de sensores, dependiendo de la magnitud física que miden, por ejemplo, sensores que varían del resultado dependiendo la cantidad de voltaje o corriente, sensores que cambian su valor dependiendo de una presión, sensores que miden la cantidad de fuerza ejercida, sensores que leen el comportamiento de un fenómeno físico o químico e interpretarlos en valores numéricos representativos, etc.

2.1.1 Física acerca del sensor

De los tipos diferentes de sensores antes mencionados, los sensores de tipo eléctrico son los que se utilizan para el sistema. Los sensores eléctricos trabajan por medio de señales eléctricas que varían según la medición del fenómeno físico. Las señales eléctricas que un sensor eléctrico obtiene son de tipo digital y de tipo analógico:

- La señal digital es aquella que solo tiene un número finito de valores posibles (Torrente, 2013). Estos valores se les conoce como variables discretas porque toma solo un valor que pertenecen al rango de voltaje que el sensor puede detectar.
- La señal analógica es aquella que tiene infinitos valores posibles dentro de un rango determinado (Torrente, 2013). Los valores analógicos también son conocidos como variables continuas, que son aquellas que toma cualquier valor dentro de un intervalo.

Los sensores eléctricos tienen un método para poder llevar a cabo las lecturas a partir de la detección de estímulos desde el exterior. Para llevar a cabo este proceso se requieren los siguientes pasos (Figura 2.1) :

- Medición: se mide el fenómeno físico para que sea detectado por el sensor como una respuesta una señal eléctrica. Esta respuesta puede ser proporcional o no al valor físico, puesto que no todo fenómeno físico puede ser representado como lineal.
- Acondicionamiento: es el proceso de una señal de tipo eléctrica se convierte en otro tipo de señal. En este proceso se puede realizar una amplificación, una linearización y/o convertir a la señal de tipo analógica a digital o viceversa, por mencionar algunas.
- Conversión: el sensor tiene un circuito que transforma y/o amplifica la salida, y este a su vez pasa a otro circuito conversor de señal analógica a señal digital para que sea interpretada por el microcontrolador.

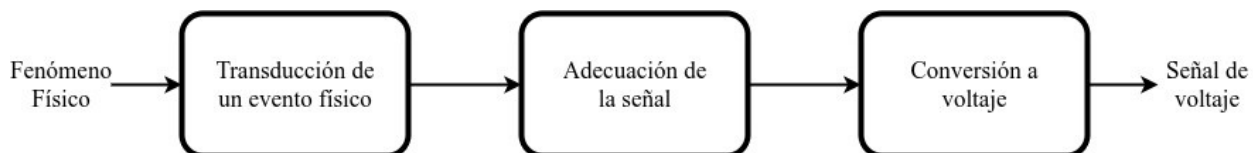


Figura 2.1 Funcionamiento de un sensor eléctrico

Los sensores eléctricos son utilizados comúnmente para el monitoreo de sistemas en el exterior, y de esa forma obtener información del medio. Por ejemplo, sensor distancia, fuerza, corriente, luminosidad. Por lo cual, se implementa el uso de un descriptor. Un descriptor nos ayuda de parámetro para medir su entorno.

Existen dos tipos de descriptores: los estáticos que se enfocan al valor resultante que es invariante en el tiempo; y los dinámicos, que se basan en el resultado que cambia con respecto al tiempo.

2 Marco teórico

Ejemplos de algunos descriptores estáticos son:

- Rango: valores mínimos y máximos que pueden tener la entrada y la salida del sensor.
- Exactitud: la diferencia mínima entre el valor teórico con respecto al valor medido.
- Repetitividad: forma de repetir una medida a partir de una precisión establecida.
- Reproductividad: capacidad de repetir una medición bajo condiciones distintas a la original.
- Resolución: mínimo valor medible del sensor.
- Error: la diferencia del valor real con el obtenido.
- Sensibilidad: relación entre la salida del sensor con su entrada.
- Estabilidad: la aproximación que tiene el sensor para mantener la misma lectura.

Ejemplos de algunos descriptores dinámicos son:

- Tiempo de retardo: tiempo en el que el sensor alcanza la mitad de la medición final.
- Tiempo de subida: es el intervalo de tiempo transcurrido desde que el valor de salida alcanza el 10% de su valor permanente, hasta que llega por primera vez al 90% del mismo valor.
- Tiempo de pico: tiempo en el que el sensor alcanza a llegar al pico de sobre oscilación.
- Pico de sobre oscilación: valor del incremento de la salida de un sensor con el de su valor tope.
- Tiempo de establecimiento: tiempo en el que la salida del sensor alcanza su asentamiento de un 5% de su valor tope para mantenerse.

Un sensor de corriente (Figura 2.2) funciona usando un elemento de conversión magnética insertado en el interior del núcleo magnético o un bobinado creando un núcleo magnético, dependiendo de la corriente que fluya en el conductor bajo prueba. Así pues, cada método de detección esta caracterizado por sus propias ventajas y desventajas, siendo difícil satisfacer todos los requerimientos de medida con un solo método de detección (Ikeda and Masuda, 2016).



Figura 2.2 Ejemplo de un sensor de corriente.

*Imagen recuperada de
<http://www.directindustry.es/prod/honeywell-safety-and-productivity-solutions/product-12365-306124>.*

La forma en la que trabaja un sensor de corriente es la siguiente:

- Dentro del sensor se encuentra un embobinado de alambre de corriente, que, en conjunto con el conductor por donde circula la corriente a medir, forman un transformador (Avelino, 2001) el cuál genera un campo magnético al colocar el flujo de electricidad.
- El transformador de corriente genera una carga eléctrica en el devanado secundario, el cual es proporcional a la carga que se genera en el devanado primario (Figura 2.3).
- El número de espiras que tiene cada devanado tiene que ajustarse de tal manera que cuando se genere un campo magnético por el flujo de corriente en el primario, en el secundario se genere un flujo de corriente muy aproximado al valor primero.
- La fórmula con la que se relaciona el número de espiras con el voltaje y el flujo de corriente se muestra en la fórmula 2.1:

2 Marco teórico

$$\frac{I_s}{I_p} = \frac{V_p}{V_s} = \frac{N_p}{N_s}$$

Fórmula 2.1 Relación de intensidad con el número de espiras.

Recuperado de Avelino Pérez, Pedro (2001). Transformadores de distribución: teoría, cálculo, construcción y pruebas. Editorial Reverte

- En donde:
 - I_s es la intensidad del secundario
 - I_p es la intensidad del primario
 - V_p es la diferencia de potencial del primario
 - V_s es la diferencia de potencial del secundario
 - N_p es el número de espiras en el primario
 - N_s es el número de espiras en el secundario

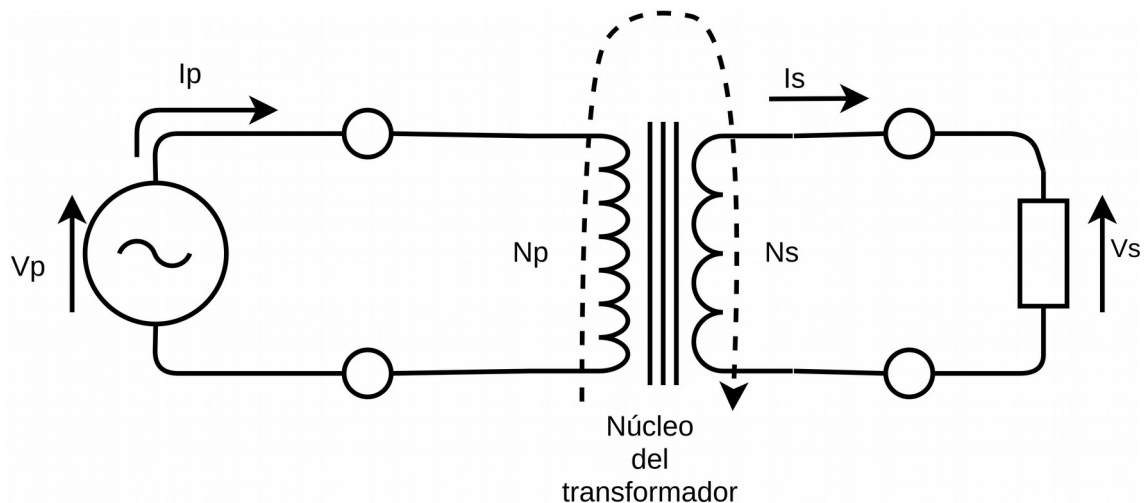


Figura 2.3 Diagrama funcional de un transformador

- El devanado primario se forma de un sola espira por el conductor que se mide. Para el secundario el número de espiras puede variar. Por ejemplo un sensor de corriente utiliza entre 1000 a 2000 espiras.

- En la bobina que tiene colocada el sensor está el imán que funciona como núcleo que potencia el campo magnético. Esto se realiza para que sea más fácil detectar el flujo de corriente provocado por el campo magnético.
- Si no existe un flujo de corriente, no puede generarse un cambio en el campo magnético y la bobina que hace la conversión, no genera una corriente proporcional.
- Los sensores llevan dentro una protección de una resistencia de carga (*burden en inglés*) como se muestra en la figura 2.4 para no dañar al circuito con voltajes altos. Esto se debe a que por ley de Ohm ($V=RI$), ya que el transformador genera una corriente eléctrica acotada (I) y en caso de quedar abierto el circuito, la resistencia (R) tiende a un valor a infinito afectando al voltaje (V) haciéndolo un valor infinito de igual manera;.

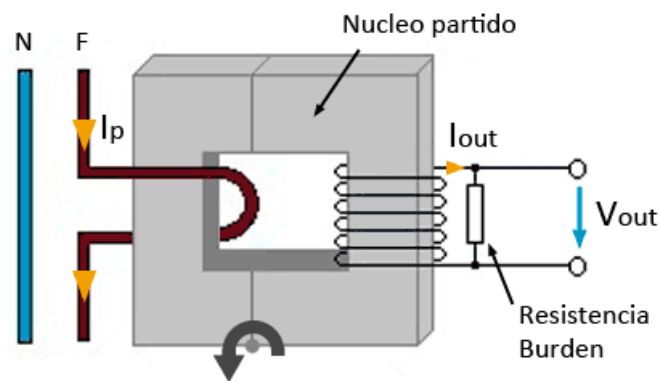


Figura 2.4 Conexión de la resistencia de carga.

Recuperado de <https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>

2.1.2 Protecciones y manejo del sensor

El sensor necesita estar acondicionado previamente para ser utilizarlo en conjunto del microcontrolador. Para ello se tiene que adaptar para su conexión de manera correcta y obtener una lectura. Esto por medio de su conversión de campo magnético a corriente por medio del proceso de inducción del transformador.

2 Marco teórico

Este proceso se lleva a cabo en tres partes que son: La salida del sensor en intensidad, la adaptación de rango de tensión, y las tensiones negativas - positivas.

A la salida de dicha carga se le coloca una resistencia de carga constante, de modo que el voltaje en las terminales es proporcional a la corriente generada por el transformador. Esta resistencia es colocada en los transformadores para que la corriente mantenga su valor de carga de manera constante. Para este ejemplo, podemos considerar que el sensor utiliza una resistencia con un valor de 33 ohm.

Los sensores de corriente trabajan con un rango de voltaje para calcular la corriente equivalente. Este voltaje es calculado por medio de la ley de ohm, que dependiendo de la construcción del sensor varía la lectura de la corriente eléctrica. Supongamos que el rango que el sensor genera de salida varía de 0V a 1V, donde el valor máximo del sensor de 1V es equivalente a 30A y 0V corresponde a 0A. Con la variación del voltaje podemos medir la corriente que se encuentre entre 0A a 30A.

A esta variación de voltaje que se encuentra en el sensor se le conoce como voltaje pico (Jin, Wang, Xu and Zhou, 2009), el cual es el valor máximo o mínimo de voltaje que alcanza un componente electrónico. Este valor puede ser tanto positivo como negativo dependiendo de la aplicación requerida.

Comúnmente, los sensores son utilizados en conjunto de los microcontroladores para controlar tareas en específico. Un microcontrolador obtiene las lecturas digitales desde sus puertos, cuyos picos de voltaje transduce son de 0V a 5V, como es el caso de un PIC o un Arduino.

Un microcontrolador no detecta tal cual tensiones negativas, por lo cual se utiliza un amplificador operacional. Este circuito integrado tiene la función de tomar desde el voltaje entrada y amplificar su salida para ser utilizada. Varias de sus funciones son: temporizador, circuito comparador, o un acondicionador para señales.

El LM358, como lo menciona su fabricante (STMicroelectronics, 2018), es un circuito integrado de potencia baja, para propósitos específicos de circuitos con filtros y de sumadores analógicos (Figura 2.5).

Cuando la medición es de voltaje en alterna se tiene que es un voltaje RMS (root mean square), el cual es el voltaje de alterna que produce el mismo efecto de disipación de valor que su equivalente de voltaje o corriente directa (Cartwright, 2007).

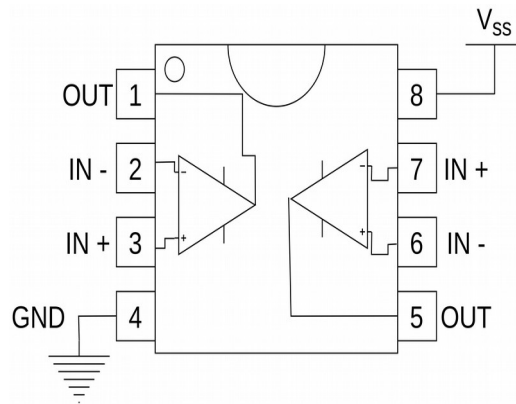


Figura 2.5 Integrado LM358

Por lo cual es necesario transformar el voltaje RMS a voltaje pico para que el microcontrolador pueda trabajar con él. Esto se lleva a cabo con el uso de la fórmula de conversión que se muestra en la fórmula 2.2 (Cartwright, 2007).

$$V_p = \sqrt{2} \cdot V_{rms} = R \cdot I_{rms} = \sqrt{2} \cdot R \cdot I_{rms}$$

Fórmula 2.2 Cálculo de Voltaje pico

Donde:

- V_p es el voltaje pico
- V_{rms} es el voltaje RMS
- R es la resistencia que se
- I_{rms} es la corriente RMS

El voltaje pico es la representación de media onda del voltaje, por lo cual se necesita conocer el voltaje pico a pico, pues es la magnitud completa de la onda, entonces con base a la fórmula anterior se puede obtener la Fórmula 2.3 (Cartwright, 2007).

2 Marco teórico

$$V_{pp} = 2 V_p$$

Fórmula 2.3 Cálculo de voltaje pico a pico

En donde:

- V_p es el voltaje pico
- V_{pp} es el voltaje pico a pico

Una vez que se completa la onda de voltaje pico a pico se puede realizar un acondicionamiento para que el microcontrolador pueda hacer la lectura. Los valores que toma la onda de voltaje abarca la región positiva y la región negativa. Pero como el lector analógico del microcontrolador tiene un rango de voltaje a la entrada del ADC de 0V a 5V, se necesita hacer una rectificación a la señal.

Un método para desfasar la señal dentro de los rangos para muestreo del microcontrolador es agregando un offset de corriente directa. De manera física, con el mismo circuito integrado de la Figura 2.5 mencionado en la conectado en una configuración de offset se puede analizar la señal. En la Figura 3.9 del desarrollo se presenta un esquemático de conexión del sensor con las adecuaciones necesarias para su funcionamiento.

2.1.3 Explicación e implementación del sensor

Para obtener lecturas de un sensor de corriente no invasivo tenemos que tener en cuenta la forma en la que se coloca para obtener las mediciones de corriente como se muestra en la Figura 2.6. Algunos sensores tiene unas marcas en donde se indica la posición en la que el sensor debe ser colocado, esto indicando la dirección del flujo de electricidad, puesto que de eso depende la generación del campo magnético y las lecturas obtenidas sean acorde a las especificaciones que tiene el sensor.

Utilizando las dos terminales (roja y negra de la Figura 2.6) es de donde se obtienen las lecturas. La forma en que los valores obtenido son mostrados es por medio de variaciones en el voltaje, que están dentro del rango ya especificado por el fabricante. Algunos sensores se encuentran ya acondicionados con circuitos adicionales de manera interna para solo colocarse y empezar a obtener sus mediciones por las mismas terminales del sensor.

Resumiendo el proceso de conversión de corriente eléctrica a valores que el ADC puede obtener se representa por medio de la Fórmula 2.4 donde los parámetros utilizados son:

$$I_{ADC} = K * I_A + b$$

Fórmula 2.4 Conversión de corriente eléctrica a valor digital

- I_{ADC} es el número de bits que el ADC obtiene del sensor.
- I_A es el valor de la corriente eléctrica dada por el sensor en un punto A.
- K es la constante de conversión de la corriente a número de bits.
- b es una constante que permite ajustar la corriente del sensor al rango del convertidor analógico/digital.

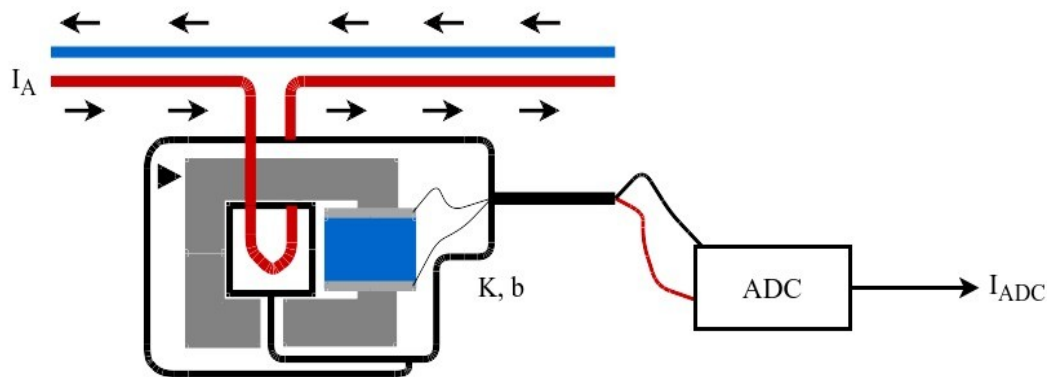


Figura 2.6 Manejo de un sensor de corriente no invasivo

2 Marco teórico

La constante K implementada dentro de la Fórmula 2.4 es obtenida de la siguiente manera:

- Para calcular la constante K por medio de la ley de Ohm, se necesita que el voltaje sea igual a resistencia por corriente eléctrica. Partiendo de ahí tenemos que la corriente eléctrica está dada por la Fórmula 2.4, sólo utilizamos la resistencia; además, si es necesario, considerar algún circuito operacional para ajustar el voltaje del sensor al microcontrolador para que al final se pueda obtener el voltaje en número de bits que resulta de la transducción del ADC. Estos elementos los podemos conjuntar para darnos como resultado la Fórmula 2.5. Por ejemplo, si se utilizan los 33 Ohms de la resistencia de burden, los 1024 números de bits que utiliza el ADC, el voltaje que maneja el sensor de 1V y un escalar de valor 5 que representa el amplificador operacional utilizado para que el microcontrolador lea el voltaje del sensor, tenemos que el valor de K es igual a 6.2.

$$K = \frac{\text{Número de bits}}{\text{Escalar de amplificación} * \text{Voltaje de de sensor} * \text{Resistencia}}$$

Fórmula 2.5 Cálculo de la constante K

2.2 Microcontroladores, Microcomputadoras y componentes electrónicos

En la actualidad, existen dispositivos electrónicos que cuentan con funciones en específico que nos ayudan a llevar acabo diferentes tareas, entre las más comunes el de controlar, interpretar o mostrar resultados que provienen de señales eléctricas que llegan a sus periféricos. Se tienen dispositivos como los microcontroladores o las microcomputadoras que realizan acciones por medio de software a partir de las señales que reciben, y se encuentran los dispositivos que por medio de componentes electrónicos toman la señal eléctrica y la acondicionan para que otros dispositivos puedan trabajar con ella.

2.2.1 Microcomputadoras (Raspberry Pi 3 B)

Una microcomputadora es una computadora que esta compuesta por un microprocesador y al menos una memoria semiconductora (Freiberger,1981). Los microcontroladores tienen circuitos de entrada/salida y de almacenamiento que le sirven para llevar acabo operaciones con otros componentes.

Algunas de las características propias de una microcomputadora son las siguientes:

- Versatilidad a la hora de agregar otros componentes.
- Adaptable en cualquier entorno que requiera una gestión.
- Hardware reducido.
- Orientado para ser de fácil uso.
- Bajo costo.
- Peso y dimensiones mínimas.

Hay que aclarar que al ser microcomputadoras, también comparten características con una computadora que usamos cotidianamente, como lo son las velocidad de procesamiento, visualización de entornos gráficos, realizar tareas al mismo tiempo y/o conectividad hacia internet.

La Raspberry Pi (Figura 2.7) es una tarjeta que hace la función de una microcomputadora a un bajo costo, con el propósito de llevar a cabo proyectos que necesiten ser gestionados un procesamiento de cómputo para realizar algunas tareas en específico. La tarjeta cuenta en todas sus versiones un procesador Broadcom, una memoria RAM, una GPU, puertos USB, HDMI, Ethernet , 40 pines GPIO y un conector para cámara (Upton, 2018).

Utiliza una tarjeta SD y en ediciones posteriores una tarjeta MicroSD que hace la función de almacenamiento interno, en donde se almacenan los recursos que utiliza el sistema operativo con el cual trabaja.

El modelo 3 B es uno de los modelos más completos de las tarjetas en sus versiones anteriores, ya que cuenta con un procesador ARMv8 de 64 bits Quadcore Broadcom BCM2837. Trabaja con una velocidad de procesamiento de 1,2GHz.

Cuenta también con el añadido de un modulo Wi-Fi BCM43438 BUILT-IN, además de un modulo Bluetooth low energy que evita el uso de antenas externas para la comunicación inalámbrica de la Pi. Son muy importantes para el dispositivo pues es como realiza su conexión en una red WLAN (wireless local area network) e interactuar con otro dispositivo conectando en la misma red.

2 Marco teórico

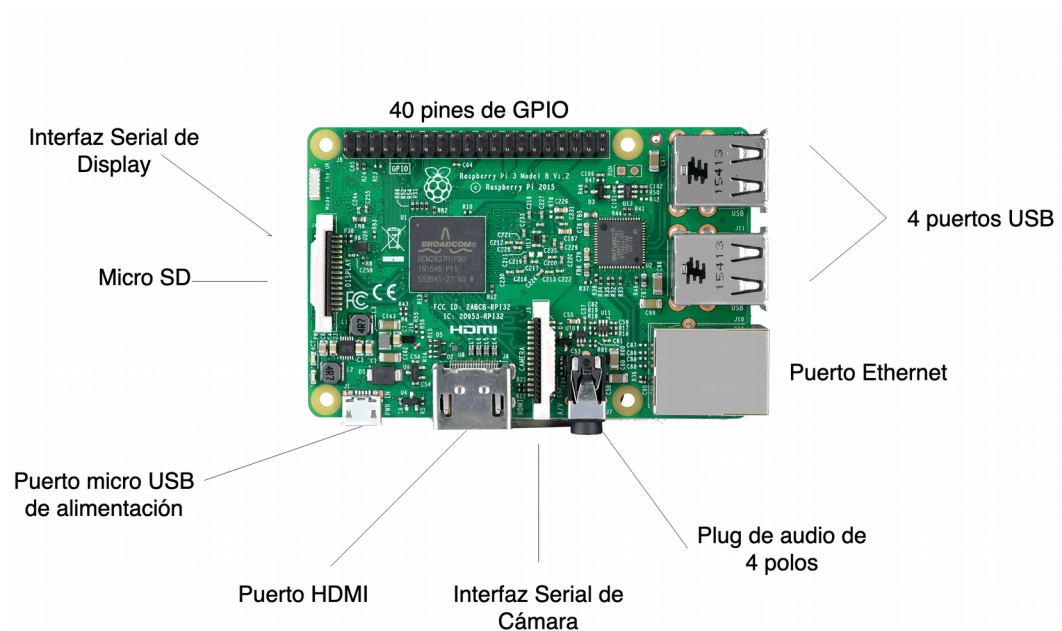


Figura 2.7 Diagrama de una Raspberry PI 3

Los pines con los que cuenta la Raspberry Pi 3 B contiene un número de 40 pines (J8), de los cuales los GPIO (Entrada / Salida de propósito general). Por medio de los pines de la placa se puede establecer una conexión de comunicación con otros componentes para poder enviar, obtener y/o controlar los bits desde software que consta desde el 0 al 27, siendo 0 y 1 los pines reservados. La organización que tienen todos los modelos 3B y sus variantes se puede ver en la figura 2.8.

Las nomenclaturas se asocian (figura 2.8) a los pines de salida tienen tareas específicas como manejar protocolos de comunicación, así como de control. De los cuales son:

- SDA es Serial Data Line
- SCL es Serial Clock Line
- TXD es transmisor asíncrono
- RXD es receptor asíncrono
- SCLK es serial clock
- MOSI es Salida de datos del Master y entrada de datos al Slave

- MISO es Salida de datos del Slave y entrada al Master
- PWM es Modulación de ancho de pulso
- I2C es el protocolo de comunicación esclavo – maestro

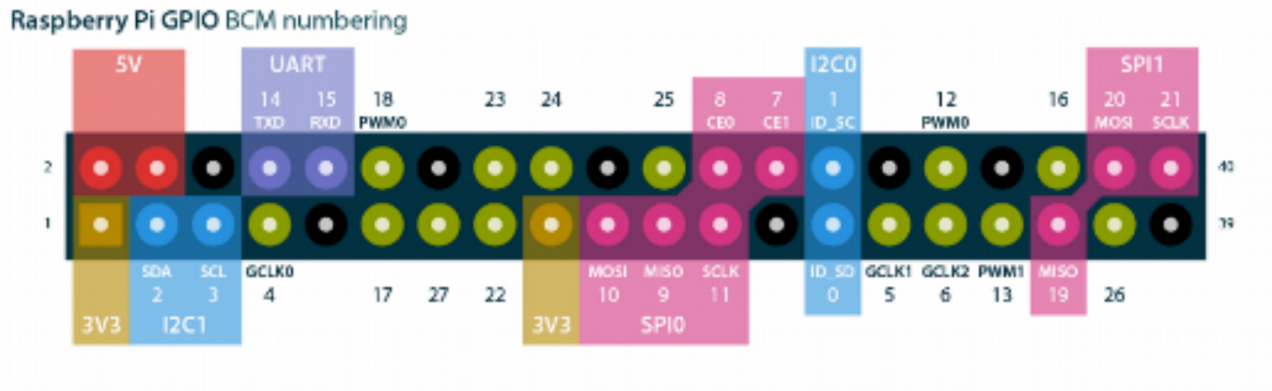


Figura 2.8 Pinout de la Raspberry Pi 3B.

Imagen recuperada de <https://es.pinout.xyz/resources/raspberry-pi-pinout.png>

Para el uso de software dentro de la Raspberry Pi existe una variedad grande de sistemas operativos. Los sistemas operativos más comunes son los que están basados en el sistema operativo GNU/Linux. Muchos programadores y desarrolladores comúnmente se inclinan al uso de Linux. Esto es porque, generalmente, Linux ocupa menos recursos de hardware. Uno de los más usados o conocidos es el sistema operativo llamado Raspbian, basado en Linux.

Raspbian (BYTEMARK HOSTING, 2018) es un sistema operativo gratuito basado en Debian optimizado para el hardware Raspberry Pi. Un sistema operativo es el conjunto de programas básicos y utilidades que hacen funcionar la Raspberry Pi. Sin embargo, Raspbian proporciona más que un sistema operativo puro: viene con más de 35,000 paquetes, software pre compilado incluido en un formato que ayuda para una fácil instalación en su Raspberry Pi.

El desarrollo del sistema operativo Raspbian esta conformado de la misma forma que Debian, por lo cual sus componentes son manejados por directorios que se encargan de administrar diferentes secciones del sistema operativo como se muestra en la Figura 2.9:

- La raíz es la localización principal de todos los directorios que conforman al sistema operativo. Dicha raíz tiene un espacio de memoria dedicado donde se almacenan los directorios.

2 Marco teórico

- Un sistema de archivos el cual ayuda a gestionar a cada uno de los directorios almacenados en el almacenamiento principal.
- Cada directorio dentro de la raíz se encarga de administrar elementos que ayudan al usuario a interactuar con la microcomputadora. Por ejemplo en la figura 2.9, el directorio *home* se dedica a administrar los archivos de los usuarios registrados o el directorio *lib* se encuentran las bibliotecas esenciales de Linux.
- Algunos directorios requieren un almacenamiento secundario donde puedan almacenar elementos como herramientas del sistema, configuraciones, datos de usuarios, bibliotecas y/o registros propios del sistema operativo.

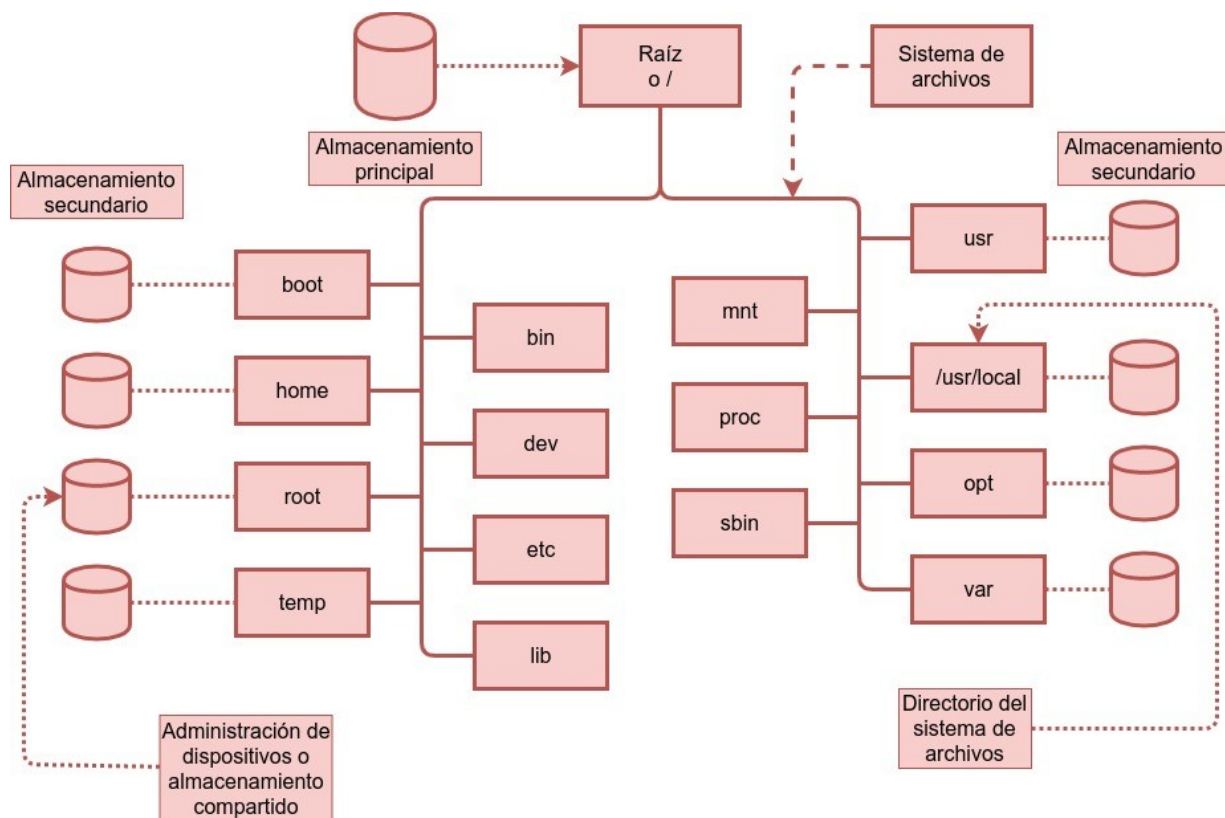


Figura 2.9 Estructura del sistema operativo Raspbian basado en Linux.

La construcción inicial de más de 35,000 paquetes Raspbian, optimizados para un mejor rendimiento en Raspberry Pi, se completó en junio de 2012. Sin embargo, Raspbian aún se encuentra en desarrollo activo con énfasis en mejorar la estabilidad y el rendimiento de tantos paquetes de Debian como sea posible.

2.2.2 Microcontroladores (Arduino Nano)

Un microcontrolador (Torrente, 2013) es un circuito integrado o chip, es decir, es un dispositivo electrónico que integra en un solo encapsulado un gran número de componentes que tiene la característica de ser programable. El circuito integrado tiene la función de ejecutar de forma autónoma una serie de instrucciones previamente definidas por un programador.

Todos los microcontroladores (figura 2.10) cuentan con elementos básicos para su funcionamiento, los cuales son:

- Una unidad central de proceso que se encarga de llevar acabo el control de las instrucciones, manejos de variables en localidades de memoria y de datos que son de entrada o de salida.
- Memoria EEPROM (Electrically Erasable Programmable Read-Only Memory) para alojar todos los datos y las instrucciones que va realizar como tarea el microcontrolador.
- Puertos de entrada/salida para la comunicación del microcontrolador que nos ayudan a tomar lecturas con sensores externos, a comunicar los datos almacenados con otros dispositivos, y mostrar o llevar acabo acciones con actuadores.
- Los microcontroladores cuentan con un circuito para regular el voltaje. Dichos circuitos son un regulador que trabaja a 5V y un puente de diodos interno que lo protege de un corto circuito al microcontrolador.

Para programar un microcontrolador existe software dedicado para hacer la interpretación. El software toma el lenguaje de alto nivel para interpretarlo a bajo nivel por medio de un compilador.

Arduino es una placa de desarrollo basado en el uso de hardware libre que contiene un microcontrolador ATMEL. Este microcontrolador tiene la facilidad de poder programar varias veces para llevar acabo el uso de todas sus funcionalidades. Todas las funciones las hace por medio de sus periféricos de entrada y salida.

Es de fácil uso cuando se programa porque el lenguaje de programación esta basado en processing. Processing es un software editor con un compilador que es de fácil uso, basado en sentencias como las que se encuentran en C para el programador.

2 Marco teórico

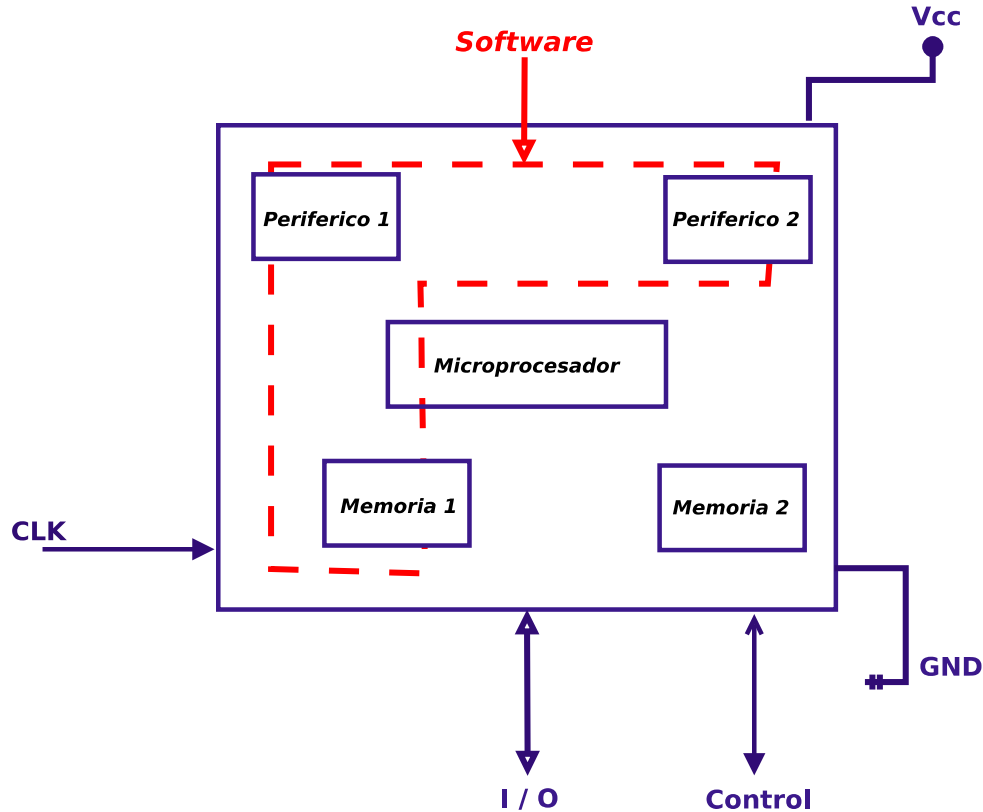


Figura 2.10 Diagrama básico de un microcontrolador

Las placas de Arduino cuentan con varias versiones desarrolladoras puesto que el controlador de Atmel viene en diferentes modelos. Dependiendo del modelo cambian sus características físicas y capacidades de control. Uno de ellos es el Arduino Nano. El Arduino Nano (figura 2.11) cuenta con las características tales como (Arduino, 2019):

- Un ATmega328P de 32KB.
- 14 puertos digitales de propósitos varios: comunicación serial TX – RX, 8 puertos digitales para manejo de PWM, 4 para manejo SPI.
- 6 puertos analógicos que también son para manejo de sensores.
- El voltaje de alimentación del Arduino Nano va de un rango de 6 a 20 Volts de entrada.
- El mismo Arduino suministra una salida interna de 5 Volts a 500 mA como alimentación para algún otro sensor que necesite de 5V para su funcionamiento.

- Cuenta con varios componentes de montaje superficial, lo que le ayuda a reducir su tamaño a comparación del modelo original

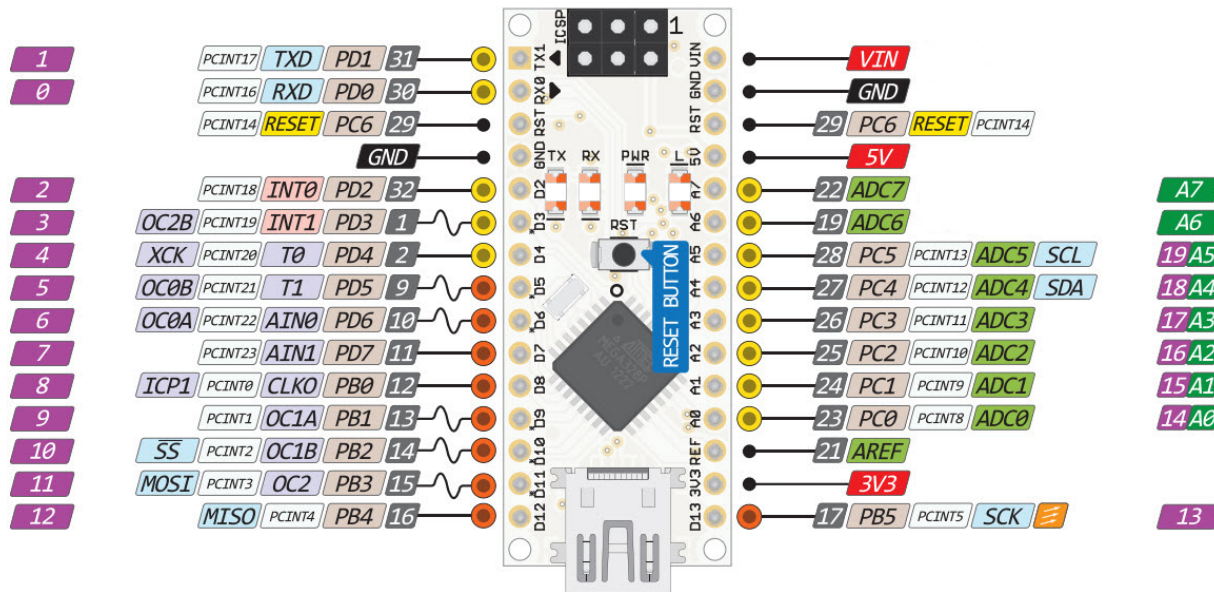


Figura 2.11 Arquitectura Básica de un arduino nano

2.2.3 Manejo de señales digitales y analógicas (ADC)

Los microcontroladores cuentan con periféricos dedicados para manejo de señales digitales y periféricos dedicados para manejo de señales analógicas.

El pinout del Arduino Nano maneja periféricos de entradas y salidas de señales digitales. Cuenta con 14 pines (figura 2.11) que son determinados como pines de propósito general (GPIO), pues depende de la configuración que se le de pueden manejarse como vías en las que se realiza lecturas o se dan salidas de voltaje, en otras palabras se puede leer voltajes de 0V o 5V o responder con los mismos voltajes de 0V o 5V.

El manejo de las entradas analógicas en el Arduino Nano cuenta con seis pines dedicados para las señales analógicas. Los pines pueden leer rangos de voltajes que varían de los 0V hasta los 5V por medio de un circuito convertidor analógico/digital.

El objetivo básico de un ADC es transformar una señal eléctrica análoga en un número digital equivalente (Huircán, 2017). La forma más sencilla de explicar la función que lleva la conversión un ADC se muestra en la figura 2.12.

2 Marco teórico

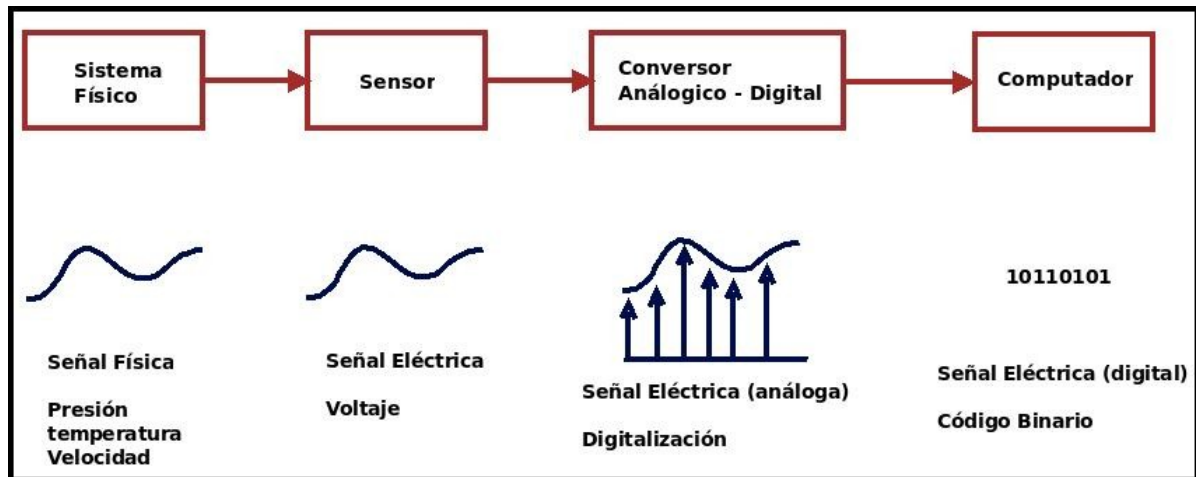


Figura 2.12 Diagrama de un proceso de conversión analógico/digital

Para llevar a cabo este proceso, se puede describir en 4 pasos:

1. A partir de una señal física, por ejemplo, mediciones de temperatura, presión atmosférica, velocidad se obtiene la gráfica de unidad de medición contra intervalo de unidades de tiempo para ver los cambios conforme transcurre el tiempo.
2. Una vez obtenida la señal, se toma todos los valores y se realiza una reconstrucción con valores discretos por medio de muestreo teniendo un valor de frecuencia de muestreo con ayuda del Teorema de Nyquist, en este caso el sensor transforma la señal física en pulsos eléctricos variantes desde 0 a 5 volts, que son interpretados por el circuito convertidor.
3. Cuando se tiene la señal discreta, le podemos asignar un código binario para cada valor que se obtuvo sea representado por números que una computadora puede interpretar, en este caso el circuito convertidor lee cada voltaje dentro de su rango y le asigna un código de 10 dígitos binarios que van desde el 0 hasta el 1023 en decimal.
4. El número binario se le asigna a la computadora para manejar la información que necesite, en este caso el número en binario se le da al microcontrolador para que el Arduino interprete el valor y pueda utilizarlo como lo requiera el usuario.

2.2.4 Regulador de voltaje

Como todo dispositivo electrónico, requieren del uso de voltaje continuo para que su funcionamiento sea el correcto. Antes, para mantener el voltaje continuo, se utilizaba una batería o una fuente de alimentación con voltaje alterno; sin embargo, actualmente también podemos encontrar convertidores de voltaje directo a voltaje directo que hacen que aumente o disminuya la cantidad de voltaje. Se puede ajustar el manejo del voltaje necesario para un dispositivo con la ayuda de circuitos como lo son transformadores y circuitos rectificadores.

Una fuente regulada o regulador de voltaje (Huircán, 2012) tiene una doble misión: debe eliminar el rizado¹ que se genera y hacer una impedancia de salida que sea adecuada para que el voltaje este regulado a la salida. En otras palabras, la salida de de la fuente de alimentación se regulan por medio de un dispositivo de control y de una salida de régimen continuo.

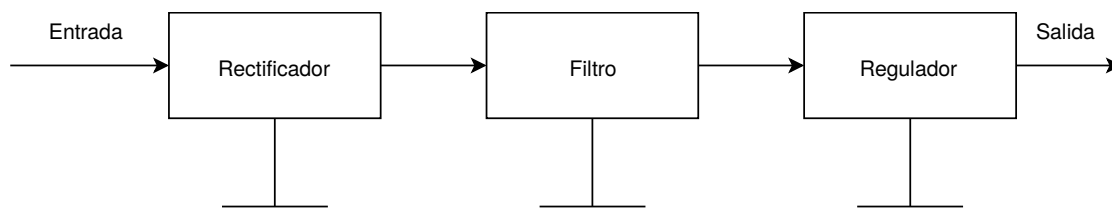


Figura 2.13 Diagrama de una fuente regulada

Los reguladores de voltaje tienen como requerimientos:

- Mantener el voltaje de salida constante independiente de las fluctuaciones de entrada y de temperatura.
- Mantener el voltaje de salida a las exigencias de corriente de carga.
- El voltaje de salida no debe contener componentes alternos (rizados = 0)
- La fuente debe poseer un sistema para limitar la corriente de salida (protección).

En un regulador ideal, la diferencia entre el voltaje de salida sin carga y el voltaje de salida en plena carga es de cero, quedando en la Fórmula 2.6. Esto en la práctica, V_L es siempre menor. A esta variación entre los voltajes se le conoce como regulación de carga.

¹ Una oscilación de pequeña amplitud impuesta sobre un valor constante

$$V_{oc} - V_L = 0$$

Fórmula 2.6 Regulación de carga

Para obtener el porcentaje al que el regulador actúa se obtiene de la Fórmula 2.7:

$$\text{Porcentaje de regulación} = \frac{V_{oc} - V_L}{V_L} \cdot 100$$

Fórmula 2.7 Porcentaje de regulación

2.3 Arquitectura de los sistemas IoT

Los sistemas del internet de las cosas (IoT, por sus siglas en inglés Internet of Things) nos han ayudado para facilitar trabajos en diferentes áreas de manera remota. Con las funciones y las herramientas con las que contienen nos pueden informar y controlar sistemas para un propósito en general. Esto se debe a que se constituyen con varios componentes electrónicos para llevar a cabo la comunicación y el control por medio de una arquitectura que hace que sea un sistema viable.

Traukina, Thomas, Tyagi, Reddipalli (septiembre, 2018) proponen una arquitectura básica para un sistema IoT que consiste en tres partes:

1. Los dispositivos (en inglés devices): son todos los sensores y actuadores, el controlador que gestiona la entrada de datos y salida de tareas que requiere hacer el sistema, y la tecnología de comunicación para transmitir los datos hacia una puerta de enlace.
2. La puerta de enlace (en inglés gateway): es el encargado del tráfico de información de una red en otra por medio de enrutadores que implementan protocolos que hacen funcional la forma en que los datos viajan.

3. La nube (en inglés cloud): son todos los servidores que se encuentran distribuidos por todo el mundo que contienen bases de datos que reciben los datos que provienen de la puerta de enlace y ser almacenados para futuras consultas.

Para que un sistema IoT tenga viabilidad se consideran requerimientos que debe cubrir, los cuales son:

- Debe de tener una conectividad y comunicación con otros sistemas y/o dispositivos que manejen protocolos como HTTP.
- Debe de contar con una gestión y control de los dispositivos. Esto para llevar acabo ciertas acciones que el sistema necesite realizar que son el actualizar software del dispositivo y/o las credenciales de seguridad, habilitar capacidades del hardware de manera remota, reconfigurar los parámetros de tecnologías de comunicación inalámbrica como el GPRS o el Wi-Fi y la localización de dispositivos en caso de que se extravíen, poderlo desconectar de manera remota, y formatear toda la información personal contenida en ella.
- Debe de llevar acabo la recolección, el análisis y el resultado de los datos obtenidos.
- Debe de tener la habilidad de cambiar su configuración de manera sencilla para adaptarse a circunstancias variables. Esto se presenta cuando existe una gran cantidad de dispositivos de hardware y el software para administrar.
- Debe de manejar una gran disponibilidad con el sistema.
- Debe de integrar todos los componentes que constituyen al sistema para su funcionamiento.
- Debe de contener protocolos de seguridad ante riesgos que sean inherentes en el sistema de internet, riesgos propios de los dispositivos IoT, y de riesgos que se presenten por daños.

Una forma de visualizar como tal la estructura de dicha arquitectura IoT es por medio de la Figura 2.14 donde podemos explicar los roles que deben realizar cada uno de los elementos que compone el sistema de la siguiente manera:

- Con ayuda de los sensores medimos el mundo exterior y los fenómenos físicos que lo rodean, para conocer sus lecturas y nosotros poder interpretar la información a nuestras necesidades.
- Una vez teniendo las lecturas con ayuda de dispositivos como controladores o microcomputadoras podemos enviarlas en formatos en específico y por las vías de los

2 Marco teórico

medios de comunicación cableados e inalámbricos con su debido protocolo de envío de datos.

- La información llega a una interfaz de programación de aplicaciones (API por sus siglas) que gestiona la información recibida. Las acciones que gestiona la interfaz es de redirigir los datos hacia servidores donde se encuentran las bases de datos, de el uso de alguna página web que consulta los datos desde cualquier dispositivo con internet y entablar una conexión con los dispositivos para recibir datos y/o enviarles instrucciones.
- Una vez teniendo los datos podemos darles una interpretación para llevar acciones según el propósito del dispositivo, como el de controlar un ambiente con el uso de actuadores implementados en el dispositivo para que modifiquen su medio.

Actualmente la implementación de los dispositivos en diferentes ámbitos de la vida cotidiana se esta volviendo más común, encontrándolos en lugares como hogares, en oficinas, dentro de las industrias y el sector médico, e inclusive dentro de los vehículos.

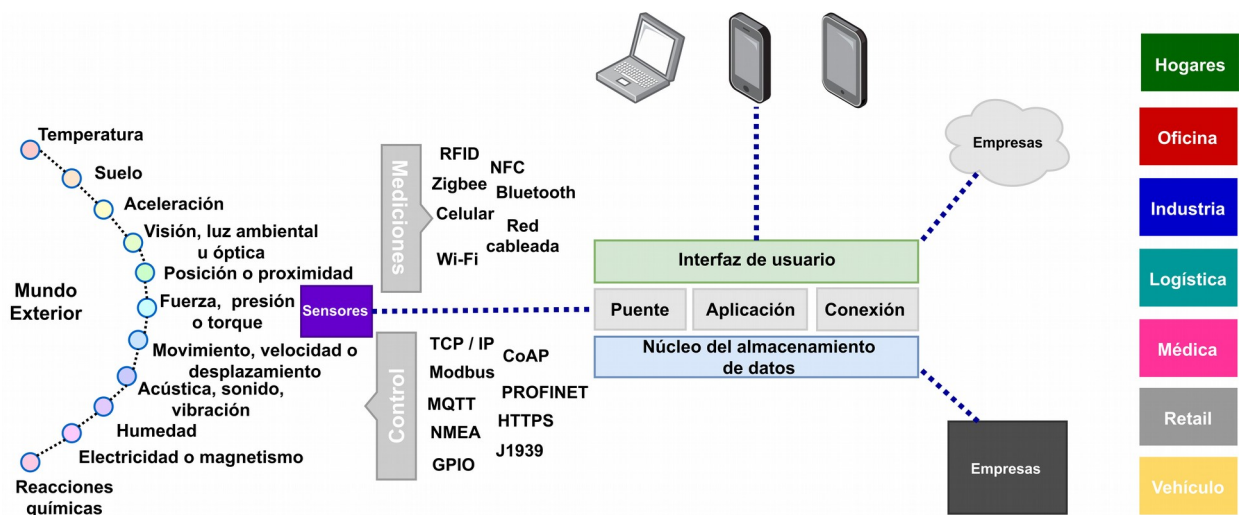


Figura 2.14 Arquitectura IoT

2.3.1 Dispositivos embebidos con internet de las cosas

Es común encontrar un dispositivo embebido para utilizar internet de las cosas. Un sistema embebido (Cayssials, 2014) se trata de un sistema de computación diseñado para realizar una o algunas funciones dedicadas frecuentemente en un sistema de computación en tiempo real.

Los sistemas embebidos son utilizados en nuestra vida día a día para monitorear o controlar alguna necesidad. El dispositivo embebido es el que realiza esta acción por medio de sus sensores para medir el exterior y procesar los datos para llevar a cabo la acción necesaria.

Esta constituido por un microprocesador y un software que se ejecuta sobre él mismo. Sin embargo el software necesita un lugar donde pueda guardarse para luego ejecutarse en el procesador tomando la forma una de memoria RAM o ROM.

El funcionamiento del dispositivo embebido (Figura 2.15) necesita de una cantidad de memoria, la cual se encuentra dentro del mismo encapsulado del procesador. También cuenta con una serie de periféricos que realizan la comunicación entrada/salida con el mundo exterior (Departamento de Ingeniería Eléctrica, Electrónica y de Control de UNED, 2019).

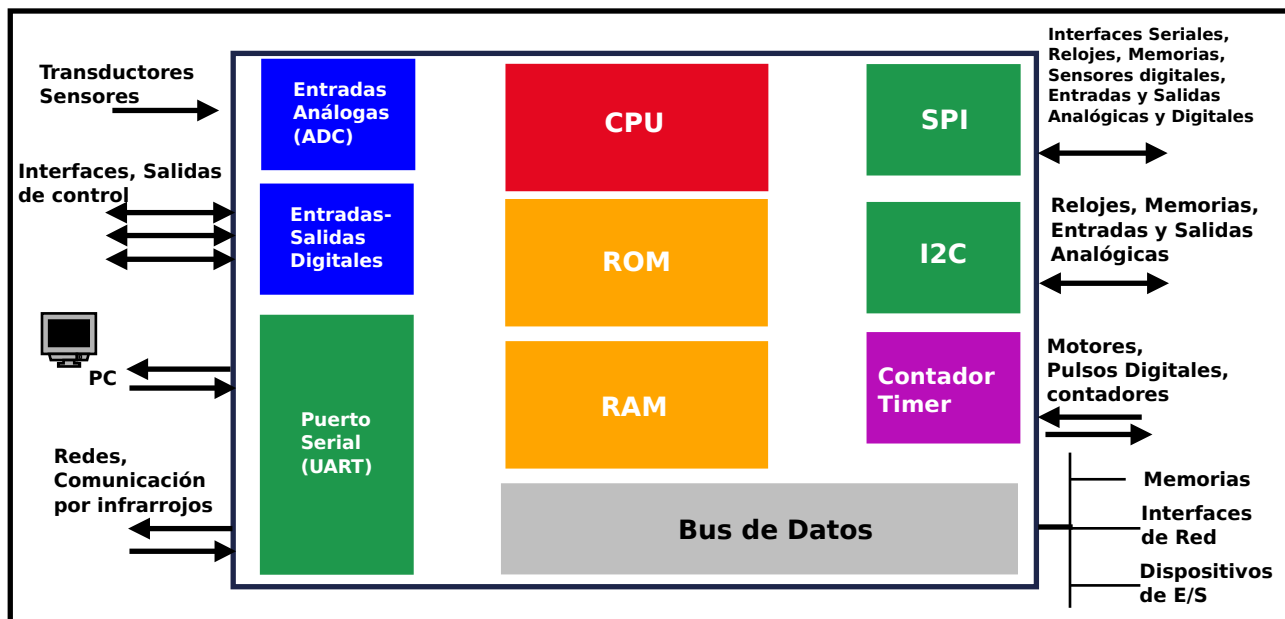


Figura 2.15 Diagrama físico de un dispositivo embebido

Los dispositivos embebidos son utilizados comúnmente para:

- Llevar a cabo control total de un sistema. Los programadores pueden usar su propio código para modificar su interfaz, su funcionalidad y/o tareas que desempeñan cada uno de los pines de microprocesador.

2 Marco teórico

- Hacer una conectividad y adaptabilidad, puesto que sus periféricos se pueden conectar a una computadora para extraer datos o de modificar su código, monitorear funcionamientos en específico desde cualquier lugar.
- Hacer una reducción de los costos de producción, porque al utilizar dispositivos embebidos se cambia el uso de PLC (controladores lógicos programables).
- Un corto tiempo de respuesta, puesto que funcionan en tiempo real, y ejecutan sus acciones en espacios de tiempo inmediatos.
- Dan una gran accesibilidad para trabajar con ellos tan solo utilizando en conjunto un microcontrolador, y un microprocesador, creando sistemas de control a costos bajos.

Las aplicaciones más actuales que se le dan a un dispositivo embebido es por el uso de internet de las cosas para llevar a cabo un control remoto desde cualquier parte del mundo.

El internet según la definición de Vermesan & Backquest (2017) es una infraestructura global de la información, que permite la ejecución de servicios avanzados mediante la interconexión física y/o virtual de cosas mediante tecnologías de información y comunicación interoperativas que existen actualmente, pero que evolucionan a lo largo del tiempo.

Internet de las cosas interactúa con muchos elementos que se relacionan entre ellos:

- Las personas que se conectan por medio de internet desde diferentes partes del mundo y con varios dispositivos sociales para propósitos educativos, de negocios, comerciales y sociales.
- Los datos que se utilizan para consultar información, evaluarla, y tomar decisiones desde cualquier parte en el momento que sea.
- Las cosas son objetos físicos que se pueden conectar tanto a Internet como a la gente a través de sensores, proporcionando información para ayudar a las personas y máquinas a tomar decisiones relevantes y valiosas (Roseroa, Núñez, Realpea, Alveara, Beltrán, & Rosado, 2017).
- Los procesos dan significado a los datos y a las cosas que la persona consultó para ser entregados en tiempo y forma.

Internet de las cosas trabaja bajo las 5 capas de la Figura 2.16:

1. La capa de los objetos contiene a los sensores y los actuadores, pues la información del ambiente que se desea consultar se envía ya digitalizada hacia la capa de abstracción de objetos a través de canales de comunicación seguros.

2. En la capa de abstracción transmite los datos recolectados desde la capa de objetos de un punto hacia otro. Usa tecnologías de comunicación como lo son RFID, 3G, GSM, UTMS, Wi-Fi, Bluetooth, infrarrojos y ZigBee.
3. Durante la capa de gestión de los servicios maneja las conexiones de un servicio con el solicitante basado en la dirección y nombre. Procesa los datos que recibe para dar los servicios requeridos por medio de protocolos de red. La capa de gestión es utilizada por los programadores, puesto que en esta capa se realizan las aplicaciones de IoT donde se utilizan los recursos de los servidores de manera remota para las API.
4. En la capa de de aplicación se proporcionan los servicios inteligentes que solicita el cliente para que se lleve acabo la acción con los sensores y actuadores.
5. Al final en la capa de negocios abarcan todas las aplicaciones y servicios de gestión de IoT. Estos van de crear gráficos, modelos de negocios, informes ejecutivos hasta tablas de datos proporcionados por los análisis de datos, para que las personas puedan tomar decisiones con respecto a las estrategias de negocio o planes de trabajo.

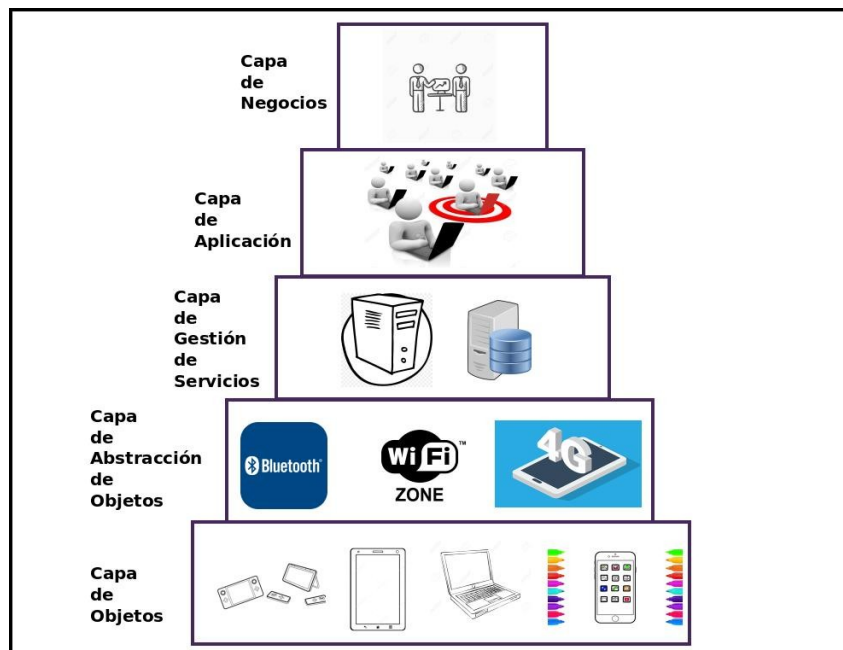


Figura 2.16 Diagrama de Capas de internet de las cosas

2 Marco teórico

Mucho tiene que ver el uso de los sensores dentro del manejo del internet de las cosas, creando así un concepto llamado red de sensores inalámbricos, que es una red ad-hoc formado por una gran colección de dispositivos muy sencillos que combinan la detección, la computación y la capacidad de comunicación (Bokareva, Bulusu y Jha, 2017) ya de manera remota, se puede consultar los datos externos sin necesidad de estar cerca del dispositivo y en cualquier parte del mundo.

Para ello se ocupan los dispositivos embebidos para control de sensado en el ambiente. Se comunica el dispositivo embebido con un dispositivo ajeno por medio de las capas de IoT hasta el usuario que los solicitó previamente.

2.3.2 Servicios Web (RESTful)

Un servicio web (en inglés web service), según la definición del *W3C Working Group* (Febrero, 2004) un servicio web es un sistema de software diseñado para admitir la interacción interoperable de máquina a máquina a través de una red con Internet o con una Intranet, siendo este un mecanismo de invocación totalmente independiente de la plataforma y del lenguaje de programación en el que se haya implementado internamente dicho servicio.

Los servicios web trabajan por medio de los protocolos de envío con el uso de peticiones web que llegan en una comunicación cliente - servidor. Existen dos formas de realizar esta comunicación que son de las más utilizadas actualmente. Estos son los protocolos de tipo SOAP y la arquitectura basada en RESTful.

El funcionamiento del protocolo SOAP es que el sistema se comunica por medio de mensajes basados en XML estandarizado; y al ser estandarizado sus componentes se basan en uso el protocolo de SOAP, el WSDL que es el lenguaje descriptivo de un web service, y el UDDI que lleva acabo la función de un directorio para las interfaces de un servicio web:

- El protocolo SOAP (Simple Object Access Protocol) esta desarrollado en XML para el intercambio de los mensajes entre aplicaciones por vía internet. Define la forma en que la información podrá acceder dentro del servicio web.
- El WSDL (Web Service Description Language) es un lenguaje hecho en XML el cual se encarga de hacer una descripción del servicio web y de la forma en la que se accede dentro.

- UDDI (Universal Description, Discovery and Integration) se trata de un estándar que describe, publica y encuentra los servicios web. Con ayuda del WSDL crea un directorio con la información del servicio web para que se puedan comunicar con otro servicio por medio de SOAP.

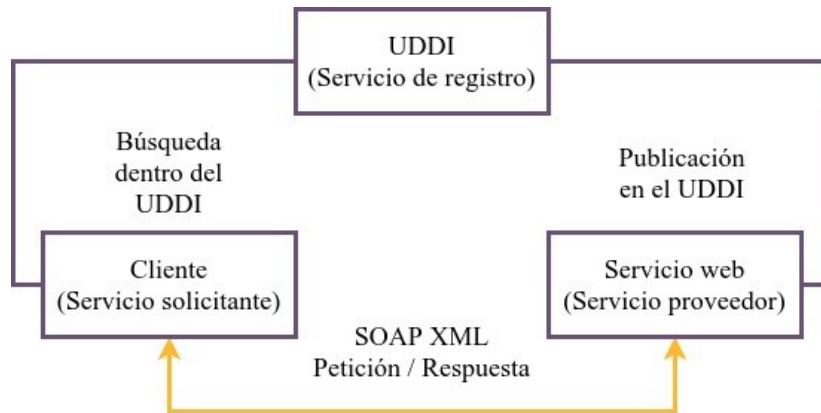


Figura 2.17 Diagrama de funcionamiento de un Web Service

La forma por la cual un Web Service funciona es de la siguiente manera (Figura 2.17):

- Existe un servicio proveedor el cual se encarga de generar el WSDL que hace una descripción del Web Service y hacer su registro dentro del directorio de UDDI.
- Otro servicio se encarga de ser la aplicación cliente, y se establece el contacto con el UDDI para poder localizar al servicio web con el cual se realiza el envío.
- Una vez que el cliente tenga la descripción dada por el WSDL, se envía una petición de request hacia un servicio en específico, el cual se encarga de tomar y redirigir todos los mensajes que le llegan en el formato SOAP.
- El servicio web analiza el mensaje tipo request del SOAP para poder procesar y atender la petición, dando como una respuesta un mensaje en tipo SOAP y enviarlo hacia el cliente.
- Una vez entregado el mensaje al cliente, analiza y procesa la petición. En dado caso que exista un error durante la interpretación, genera un error y lo comunica.

2 Marco teórico

Todo servicio web está basado en una misma arquitectura, la cual consiste en cuatro partes la cuales son:

1. Service Discovery: este servicio se encarga de canalizar a los Web Service dentro de un directorio para registrarlos, y dar funcionalidad de publicarlos y de buscarlos. Se apoya de UDDI para llevar a cabo el registro.
2. Service Description: este servicio realiza las características, las operaciones que puede soportar y como dar de alta un servicio web una vez ya localizado. Para poder realizarlo se apoya con el uso del WSDL.
3. Service Invocation: es un servicio que llama al servicio web para transmitir los mensajes entre el cliente y el servidor; esto con ayuda de SOAP que hace de manera específica el cómo se debe dar formato a los mensajes tipo request hacia el servidor y de los mensajes respuesta desde el servidor.
4. Transport: es un servicio por el cual lleva en ella los mensajes que son transmitidos entre cliente y el servidor. Utiliza el protocolo de HTTP (HyperText Transfer Protocol) puesto que es el más utilizado actualmente por navegadores y paginas web.

En contra parte de SOAP, existe una arquitectura cuya función es realizar el envío de datos implementando peticiones web en una comunicación cliente - servidor por medio del internet, el cual se denomina REST.

REST (Representational State Transfer), basado en la terminología de IBM support (2013) es un patrón de diseño para interactuar con los recursos almacenados en un servidor. Cada recurso tiene una identidad, un tipo de datos y da soporte a un conjunto de acciones. RESTful trabaja conjuntamente con el protocolo HTTP, por lo cual puede manejar los métodos de peticiones web GET, PUT, POST y DELETE para la transmisión de datos como lo haría un web service.

Estos cuatro métodos definen las cuatro operaciones que un servicio REST puede realizar:

- Si se habla de un método POST es para crear el recurso. Los datos son definidos en su entrada para ser consultados por el servicio. Este método embebe el contenido del mensaje haciéndolo más seguro de ataques.
- Si se habla de un método GET es para llevar acabo una consulta por medio de los atributos expuestos dentro de la dirección URL. Cuando se realiza el POST, muestra el resultado con los datos de la búsqueda se notifica por el GET.
- Si se realiza un método PUT hace modificaciones, cambiando el contenido que se da de entrada.
- Si se realiza un método DELETE solo elimina el recurso.

Hay un quinto elemento que se puede utilizar llamado HEAD, que este hace la función de un GET, pero a diferencia de este, solo regresa la cabecera HTTP. Normalmente se realiza esta operación solo para comprobar si hay existencia de algún recurso.

Ahora la implementación de la arquitectura REST en los servicios se les conoce como RESTful, son servicios implementados en la web bajo la forma que la define REST. Los servicios RESTful son comúnmente encontrados en aplicaciones con interfaz de programa, o mejor conocidos como API.

Una API (por sus siglas en inglés Application Programming Interface) es una interfaz que implementa un código para que dos programas puedan comunicarse entre sí de manera más fácil sin afectar la información dentro de una aplicación.

La unión de una API y una arquitectura REST resulta en una API RESTful, que es un código de comunicación basado en la metodología REST antes mencionada para utilizar las cuatro operaciones de peticiones web para la comunicación.

Una API RESTful hace un análisis de los datos que se piden desde un cliente en algún dispositivo por medio de internet y buscarlos desde el dispositivo servidor con la información; esto es de gran utilidad puesto que no es necesario tener la base de datos en el mismo lugar que el servicio RESTful, sino que hace la búsqueda por medio de las peticiones web.

RESTful tiene la versatilidad de enviar cualquier tipo de información, puesto que el tipo de dato es definido dentro del Header Content-Type, y administrar el envío de archivos en XML ya antes mencionado, JSON, o en binarios como imágenes o documentos.

JSON por sus siglas JavaScript Object Notation es un formato ligero de intercambio de datos. Convierte la información que un ser humano lee y escribe en una estructura para que las máquinas sea simple interpretarlo y generarlo (JSON.org, 2018). Está constituido por 2 tipos de estructuras: por una colección de pares de nombre/valor.

En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo (JSON.org, 2018) y por una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Para ejemplificar el funcionamiento de una arquitectura RESTful nos podemos guiar por medio de la Figura 2.18: Desde la aplicación A se hace una petición HTTP de las cuatro posibles (POST, GET, PUT y DELETE), en este caso se realiza un POST para que la aplicación B pueda crear un recurso nuevo con el tipo información dentro del Header Content-type, el cual envía un código en formato JSON y el recurso creado contendrá la información del JSON almacenada en la aplicación B.

2 Marco teórico

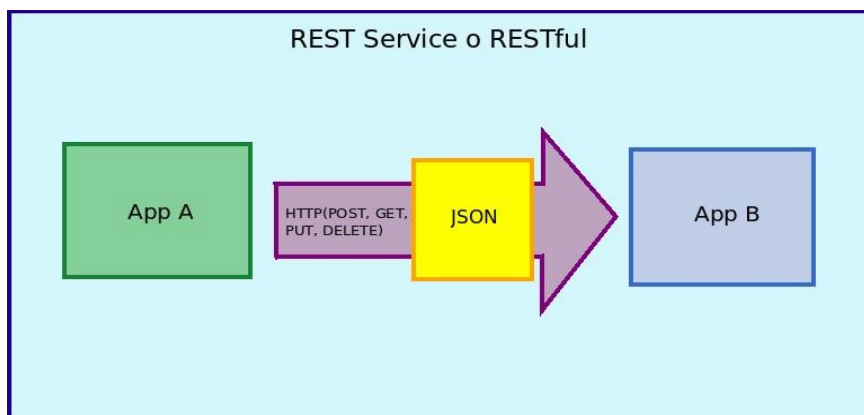


Figura 2.18 Diagrama de funcionamiento de un servicio RESTful

2.3.3 Uso de base de datos

Una base de datos es una colección de información que se encuentra almacenada y organizada de alguna manera con fin determinado (Méndez, Carlos Francisco, 2018). Un sistema de base de datos computacional se diseña para que pueda manejar grandes cantidades de información, hacer manipulación con los datos que tienen que ver con definición de estructuras para poder almacenar y manejar información, y dar una seguridad de que la información se encuentra íntegra ante fallas en el sistema o de intentos no autorizados a la base de datos (Figura 2.19).

Las bases de datos se utilizan en nuestra vida cotidiana y en diferentes aspectos de la sociedad. Se utilizan para almacenar datos importantes que alguna organización o algún establecimiento en cuestión requiere manejar para hacer consultas y llevar un registro de sus recursos. Por ejemplo las podemos encontrar en bancos para almacenar los datos de los clientes, sus números de cuenta y cantidad de manejo que cuentan y llevar acabo transacciones bancarias; otra utilidad es en centros de investigación donde deben almacenar datos de los proyectos que llevan a cabo para su estudio y llevar un seguimiento.

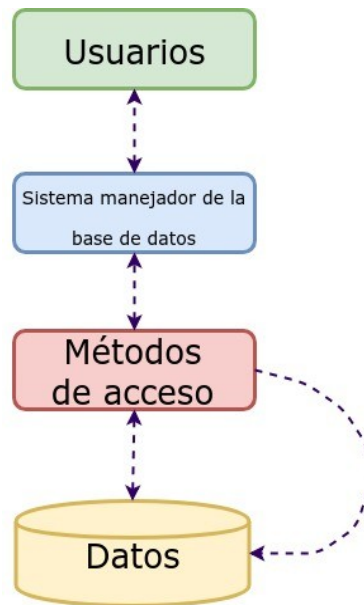


Figura 2.19 Diagrama de funcionamiento de una base de datos

Con ayuda de una API podemos administrar el manejo de la base de datos remotamente. La API y la base de datos se comunican por medio del uso de comandos que la misma base de datos interpreta y de esa manera poder manejar la información que contiene dentro.

El uso de una API se debe a que ésta establezca conectividad con los datos externos para que sean redirigidos, ser vincularlos hacia la API, y poder ser consultados sin importar el lugar dónde se encuentre la base de datos.

La manera en la que lleva acabo el almacenamiento de los datos por medio de una API que se representa en la Figura 2.20 es la siguiente:

- Por medio de una interfaz de usuario se maneja la inserción, la actualización, la consulta, o la eliminación de los datos sin necesidad de estar conectado directamente a la base de datos.
- Los datos pasan por la conexión de la API para ser dirigidos a un micro servicio que actúa como intermediario y realizar las operaciones con la base de datos.
- Los micro servicios se encargan de administrar los datos que provienen de la interfaz, indicar a la base de datos las acciones que debe hacer con los datos recibidos y notificar a la API que la acción se realizó de manera efectiva.

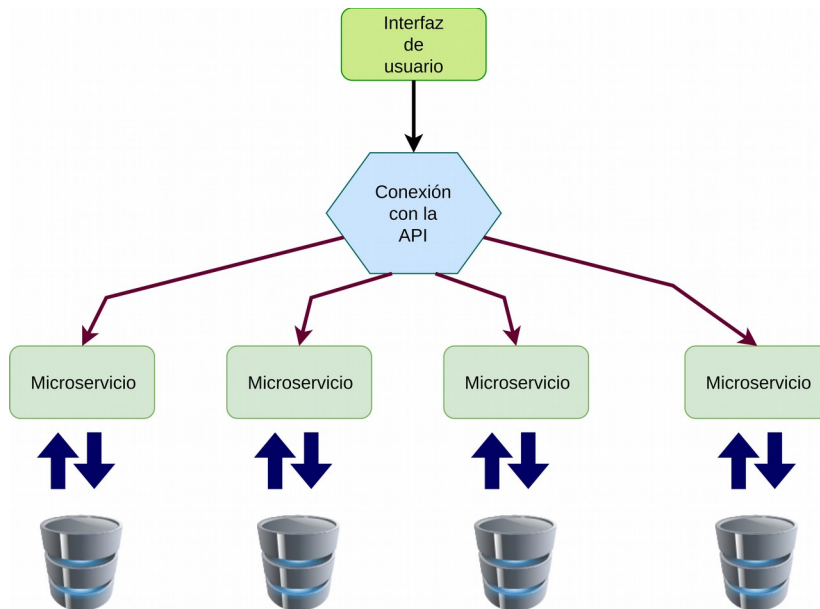


Figura 2.20 Interacción de una base de datos con una API

2.3.4 Sistema de visualización de datos

La visualización de datos es vista por muchas disciplinas como un equivalente moderno de la comunicación visual. Esto representa la creación y el estudio de la representación visual de datos (Friendly, Agosto 2009). Nosotros comunicamos los datos como información que sea clara y eficiente.

Las formas de visualizar los datos por medio de gráficos son de utilidad para comprender mejor la información acerca de su comportamiento. Los datos se pueden representar en formas de puntos, de barras o de líneas. Estos elementos se encuentran en un sistema de referencia de variable dependiente y variable independiente.

Una de las formas más sencillas de hacer un sistema de visualización es por medio de una gráfica. Una gráfica (Jensen, Anderson, 1992) es una representación visual de datos, en la que la información es representada mediante figuras o símbolos, como barras rectangulares colocadas vertical (Figura 2.21) o como líneas y puntos sobre un sistema de coordenadas o como divisiones y subdivisiones dentro de un círculo.

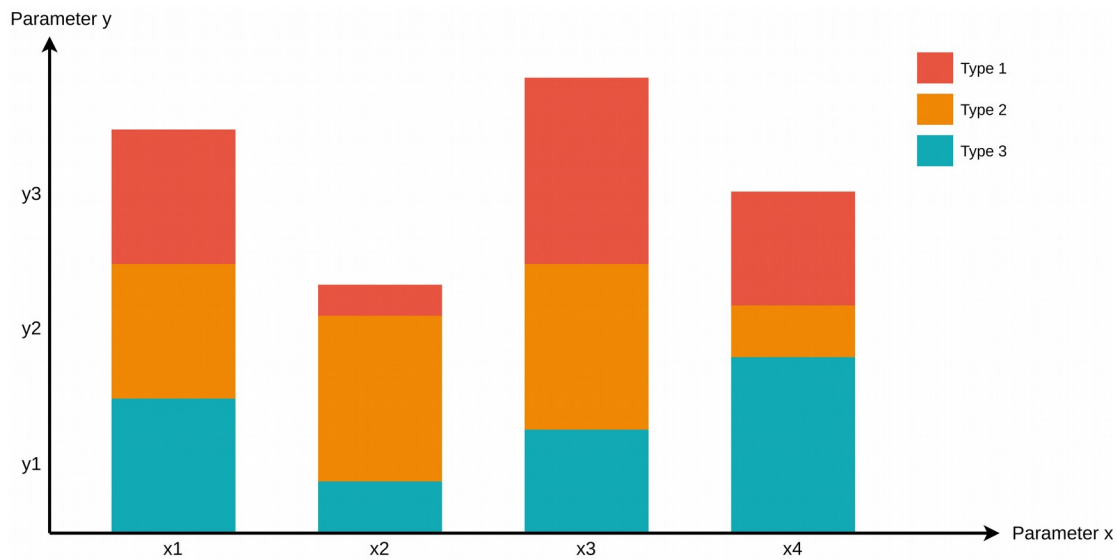


Figura 2.21 Representación de un gráfico

Una vez conociendo a detalle cada uno de los conceptos y las funciones que realiza cada uno de los componentes dentro del sistema, podemos empezar a explicar cómo estos elementos interactúan entre sí para el funcionamiento completo. En el capítulo siguiente, se muestra la arquitectura principal que conforma el sistema de medición, así como una descripción en base al marco teórico de cada modulo que lo integra.

3 Desarrollo

En este capítulo se describe el diseño y el funcionamiento del sistema propuesto en el presente trabajo. En la sección 3.1 se detalla en grandes rasgos como está constituido el sistema completo, la parte física y de programación que las constituyen y el cómo funcionan cada uno de sus componentes. En la sección 3.2 se describen los componentes físicos que tiene integrado el dispositivo para poder realizar una lectura y enviarla hacia la API, que van desde el sensor que mejor se adapta a la necesidad de uso fácil, la conexión del sensor hacia el pin analógico el microcontrolador, así como la comunicación entre microcontrolador y micro computadora. En la sección 3.3 se explica el funcionamiento del software del sistema que utiliza el dispositivo de manera interna para generar los archivos con extensión .json y sean enviados hacia el servidor donde la API los alojará en la base de datos. Relacionado con la sección anterior, en la sección 3.4 se detalla el uso de la página web para visualizar desde cualquier lugar con conexión a internet desde una computadora, celular o tableta se pueda monitorear el consumo eléctrico medido, así como obtener un precio aproximado según la tarifa de CFE.

3.1 Arquitectura del sistema

Para lograr el objetivo del presente trabajo, fue necesario diseñar e implementar un sistema que se compone de diferentes desarrollos de hardware y software, cada uno con una tarea en específico para lograr que un usuario visualice el consumo eléctrico de una casa o negocio desde cualquier lugar y en cualquier momento.

Las desarrollos que se implementaron en el sistema se enlistan a continuación:

- Medidor de corriente. Es el dispositivo embebido que se diseñó y se encarga de realizar las mediciones de la corriente y enviarlas a través de Wi-Fi hacia la API.
- API. La API RESTful se encargó de recibir los archivos JSON con la información de la corriente que fue registrada por el medidor de corriente y de esa manera almacenar los valores a la base de datos. Además, se encarga de procesar y enviar la información necesaria al sistema de visualización.
- Base de Datos. La base de datos es el lugar donde se almacenó cada archivo JSON para ser consultados por medio de la API.
- Visualizador de datos. Por medio de una página web el usuario consulta los datos de consumo por medio de gráficas que funcionan por parámetros que el mismo usuario establece.

Todos estos elementos interactuaron entre ellos a través de internet por medio del protocolo HTTP, enviando y transmitiendo los archivos JSON. Este proceso es descrito a continuación:

- El medidor de corriente se encarga de obtener los valores del consumo eléctrico por medio del sensor de gancho, para luego ser organizadas cada una en un archivo con formato JSON.
- El medidor de corriente se conecta a internet por medio de cualquier punto de acceso que tenga una configuración adecuada, para el envío de los archivos previamente creados llegando así a la API que se encuentra instalada en un servidor.
- La API se encarga de atender las peticiones web que recibe del dispositivo o de la página web, para manejar los datos y procesarlos dependiendo de la tarea que es requerida.
- El visualizador de datos alojado en el servidor realiza una petición a la API para obtener datos o información procesada en un navegador desde cualquier parte del mundo por medio de una conexión a internet. Los datos son mostrados en gráficas dependiendo de la consulta que realice el usuario.

La Figura 3.1 es un diagrama de la arquitectura descrita anteriormente. En esta se puede observar el medidor de corriente conectado a Internet mediante un punto de acceso. Esto permite enviar las lecturas de corriente a través de internet hacia la API, quien almacena la información en la Base de Datos. Asimismo, se observa que los dispositivos que utilicen la Web de Visualización la hacen a través de internet, donde el servidor mostrará la Aplicación Web con la información procesada.

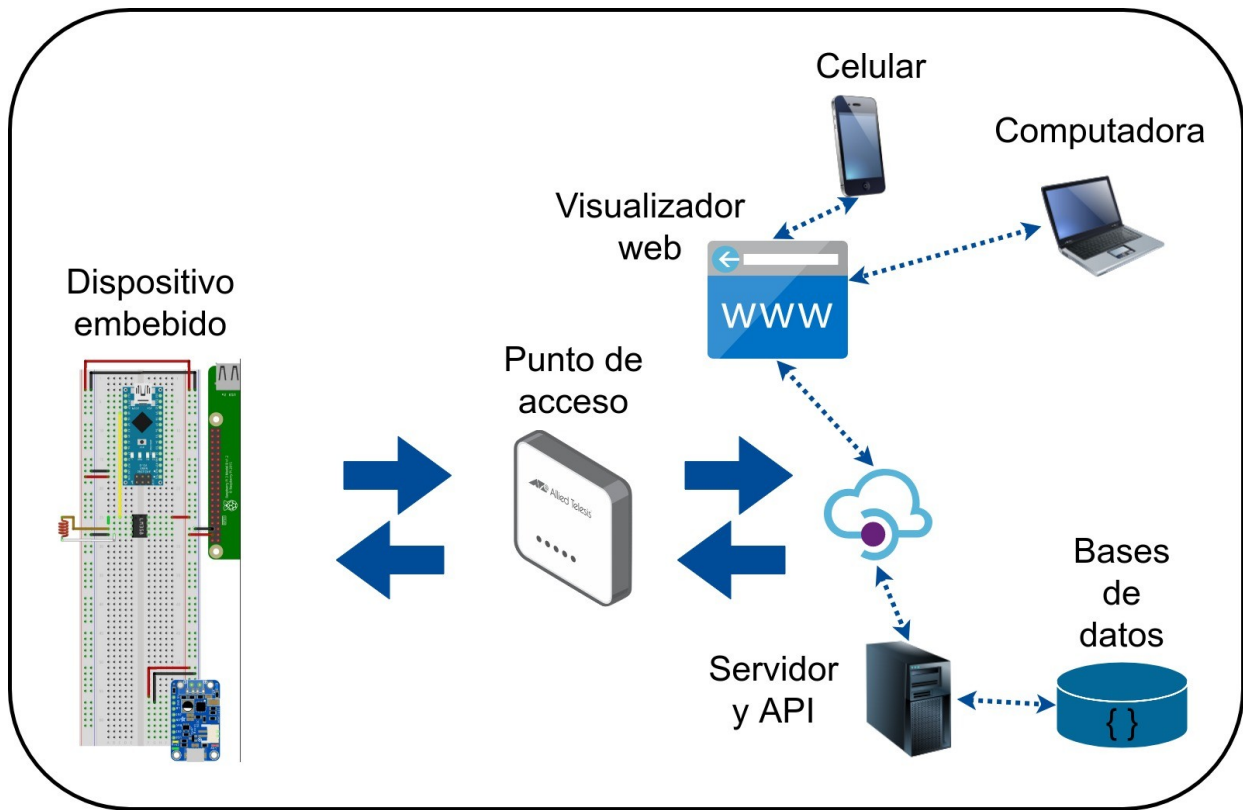


Figura 3.1 Arquitectura del sistema

3.2 Dispositivo embebido

Para leer, acondicionar y enviar los datos registrados por el sensor se diseñó un dispositivo embebido el cual administra cada tarea para obtener las señales eléctricas sean convertidas en corriente I_{RMS} , almacenarlas en un archivo y enviarlas hacia la API. El dispositivo embebido con el cual se realizaron las lecturas está constituido por los siguientes elementos (figura 3.2):

- Como fuente de alimentación, se utilizó un circuito que funciona como convertidor DC a DC, regulador y como cargador de baterías el cual provee de energía a todo el medidor de corriente. Se usó una batería de polímero litio para energizar todo el circuito para que pueda funcionar un cierto tiempo sin necesidad de conectarlo a la línea. La batería elegida

fue con una que trabaja a 3.7V a 4.2Ah la cual da un funcionamiento por un tiempo prolongado.

- El sensor SCT-013-030 es el encargado de obtener los valores del consumo de corriente en una instalación eléctrica. Para poder acondicionar las señales de lectura hacia el Arduino, se utilizó el integrado LM358 que es un amplificador operacional en configuración que rectifica y adecúa la señal a un cierto voltaje para que pueda ser leída por el microcontrolador.
- El microcontrolador Arduino Nano ayuda para poder hacer las lecturas del sensor, tomar los valores analógicos para convertirlos por medio de cálculos en el valor de la corriente que le corresponde y enviarlos por comunicación serial hacia la Raspberry.
- La Raspberry Pi funciona como una mini computadora que, por medio de un software, toman los valores de corriente que llegan por su puerto serial, los almacena temporalmente en archivos JSON y gestiona la comunicación por medio del Wi-Fi. Cuando existen una cierta cantidad de archivos, los envía hacia la API de manera inalámbrica.

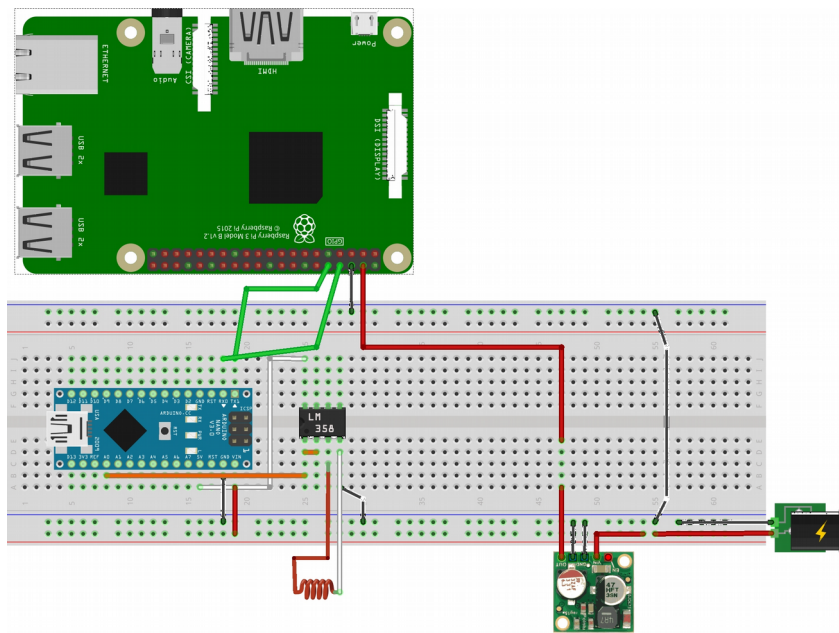


Figura 3.2 Diagrama completo del dispositivo embebido

3 Desarrollo

3.2.1 Sensor de gancho

En el mercado existen diferentes sensores de corriente no invasivos, y cada uno varía según la magnitud de corriente a medir. Los modelos varían sus magnitudes de lectura mínimas rondando desde los 3A hasta corrientes de 100A, dependiendo de la información de construcción del fabricante de cada uno de los modelos.

Para los propósitos del presente trabajo, se analizaron modelos que brindan diferentes características en cuestión del límite que pueden medir la corriente eléctrica, así como el método en el que se conecta. Los modelos óptimos para desarrollar el sistema se encuentran en la Tabla 3.1 con sus características de medición y límites de lectura.

Entre todas las opciones se buscó que el sensor sea lo más sencillo de conectar, puesto que el medidor de corriente debe ser de fácil uso para cualquier persona al momento de conectarlo, y posteriormente poder consultar en tiempo real su consumo eléctrico sin tener que dañar o alterar la instalación eléctrica.

Para ello, los sensores de corriente no invasivos con gancho son los de más fácil uso, puesto que su conexión es muy sencilla: se tiene que colocar en uno de los cables de alterna de donde se desea medir la corriente eléctrica, de preferencia en el inicio de la conexión donde están conectados todos los dispositivos, electrodomésticos, equipos y/o máquinas a medir su consumo. La conexión se puede observar en la Figura 3.3.

Aunque también existen sensores de corriente que se colocan dentro del cableado eléctrico para poder medir el consumo eléctrico, se prefiere u opta usar este tipo de conexión no invasivo para consulta de consumo eléctrico en sistemas que funcionan en alterna, puesto que puede ser colocado en una de las terminales sin tener que modificar el cableado eléctrico que se desea medir, lo que facilita enormemente su instalación.

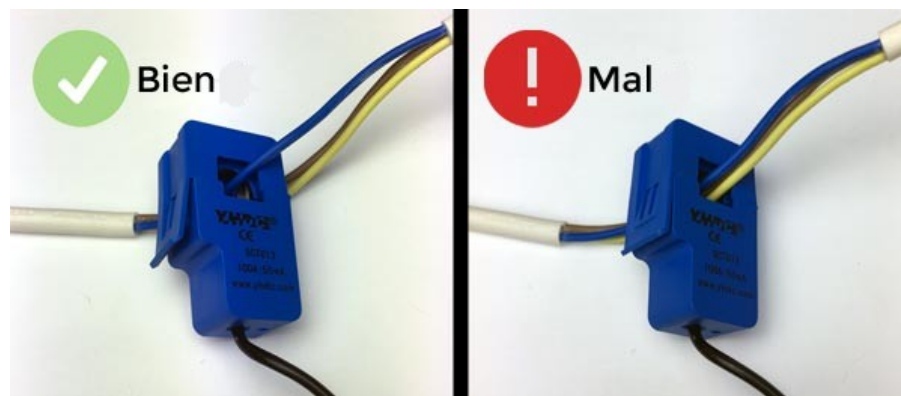


Figura 3.3 Conexión del sensor de corriente no invasivo

Nombre del Sensor	Imagen	Precio	Características
Sensor de Corriente No Invasivo 30A		\$350.00 MXN	<ul style="list-style-type: none"> • 30A de medición • 250Ohm internos • 300mV de lectura
Mini Transformador De Corriente Sensor Transductor 5a / 5ma		\$60.00 MXN	<ul style="list-style-type: none"> • 0-5A de medición • Corriente de salida 0-50mA • Relación 1:100 • Aislamiento eléctrico: 3000V • Temperatura de trabajo: -40 a +70°C
Sensor De Corriente Analógico 5a Inductivo Arduino Avr Pic		\$170.00 MXN	<ul style="list-style-type: none"> • Coeficiente de transformación: 1000:1 • Corriente de entrada: 5A. • Resistencia de muestreo: 200 Ohms. • Voltaje de muestreo: 1V. • Frecuencia de trabajo: 20 Hz a 20 KHz. • Temperatura de operación: -55 a 85°C.
Sensor de Corriente No Invasivo 30A , SCT013		\$320.00 MXN	<ul style="list-style-type: none"> • Frecuencia: 50HZ-150KHZ • Conmutación tipo transformador SCT-013 • corriente nominal de entrada • 30A de medición • Tipo: SCT-013 • Corriente de entrada: 60A/100A

Tabla 3.1 Tabla comparativa de sensores de corriente.

3 Desarrollo

Cada una de las opciones antes mostradas se analizaron sobretodo por la lectura máxima que obtienen al colocarse en un cableado eléctrico. También se consideró los lugares en los que el sensor será colocado para medir la corriente, es decir, que el consumo eléctrico de hogares y/o residencias, la cuales no rebasará los 30A. El sensor elegido fue el SCT-013-03 porque cumplió los requisitos anteriormente mencionados. Además que este modelo del sensor es muy utilizado en proyectos de internet de las cosas aplicados con Arduino.

Los sensores SCT-013-03 (Figura 3.4) son transformadores de corriente que se ocupan como dispositivos de instrumentación. Estos sensores dan una medición que es proporcional a la intensidad de corriente que atraviesa un circuito en corriente alterna. El método por el cual hace la medición es por inducción electromagnética.

Cuenta con las siguientes características (YHDC, 2015):

- La precisión del sensor puede ser de 1-2%, para ello es muy importante que el núcleo ferromagnético se cierre adecuadamente. Un pequeño hueco de aire puede introducir desviaciones del 10%.
- Soporta corrientes máximas de 30A (30A/1V) y salida en tensión de 1V.
- El rango de frecuencia con la que trabaja es de 50HZ-150KHZ.
- Su diseño es accesible para colocarse en el lugar a medir corriente.



Figura 3.4 Sensor SCT-013-030.

Imagen recuperada de <https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>

3.2.2 Control de la fuente de alimentación

Una de las opciones para regular la fuente de alimentación del dispositivo fue con el uso del regulador reg15a de la marca Pololu. Este regulador de (Figura 3.5) funciona como un conmutador síncrono de paso hacia abajo que puede hacer una tensión de entrada de un límite de 38V y de manera eficiente reduce el voltaje hasta 5V. La eficiencia del regulador maneja varía del 85% a 95% para poder alimentar cargas moderadas como sensores o motores pequeños.

Algunas de las especificaciones generales del regulador (Pololu, 2019) son:

- Voltaje de funcionamiento mínimo: 6 V
- Tensión máxima de funcionamiento: 38 V
- Corriente de salida continua: 2.5 A
- Tensión de salida: 5 V
- Protección de voltaje inverso

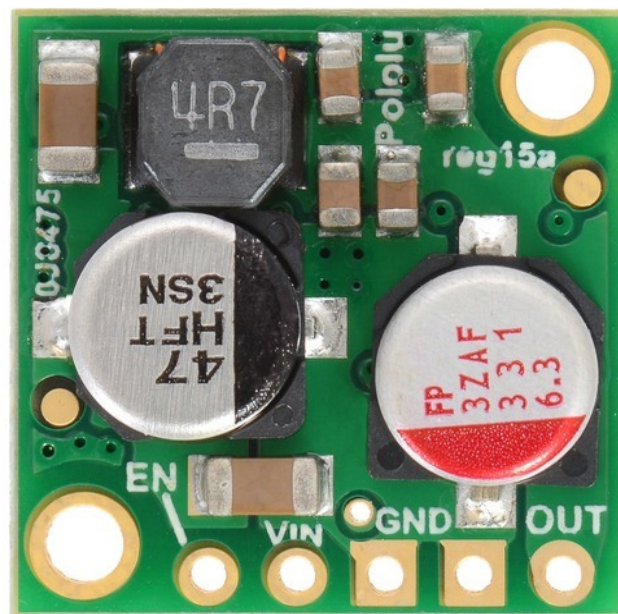


Figura 3.5 Foto de regulador marca pololu reg15a.

Imagen recuperada de
<https://www.pololu.com/product/2850/specs>

3 Desarrollo

Las desventajas de utilizar la batería en conjunto con el regulador es tener que recargar la batería siendo desconectada para poder recargarse y perder las mediciones que pasen mientras la batería se encuentra cargando. El dispositivo utilizó una batería interna para que siempre este funcionando. El problema que se presentó con el circuito fue el de desconectar la batería solo para volver a cargar, pues en ese intervalo de tiempo se pierden lecturas.

La solución fue usar otro tipo de regulador con la opción de cargar la batería al mismo tiempo. El Modulo PowerBoost 1000C (Figura 3.6) es un módulo convertidor DC / DC boost puede alimentar cualquier batería de 3.7V LiIon / LiPoly, y convierte la salida de la batería a 5V DC para alimentar proyectos que trabajen a dicho voltaje. Además cuenta con un convertidor elevador TPS61090 de TI. Este chip convertidor de refuerzo tiene algunos extras realmente interesantes, como detección de batería baja, interruptor interno 2A, conversión síncrona, excelente eficiencia y operación de alta frecuencia de 700 KHz. Las características de este modulo son (Adafruit, 2019):

- Lleva a cabo una operación síncrona, en otras palabras, significa que puede desconectar la salida del suministro de carga completamente conectando el pin ENABLE con la tierra del circuito y esté a su vez sea apagado.
- Contiene un interruptor interno de 2A (~ 2.5A de limitación de pico), que habilita a la batería de 3.7V LiPoly / LiIon para poder cargarse o alimentar un circuito.
- Un indicador LED de batería baja que se ilumina en rojo cuando el voltaje cae por debajo de 3.2V, para optimizar el uso de la batería LiPo / LiIon.
- Implementa una eficiencia operativa del 90% (en la mayoría de los casos) y una corriente baja de reposo que hacen que el circuito dure más tiempo; esto funciona de la siguiente manera: 5mA cuando está habilitado y el LED de alimentación está encendido, 20uA cuando está deshabilitado (la alimentación y el LED de batería baja están apagados).

En conjunto al uso del módulo, se utiliza una batería de polímero de litio de 3.7V a 4.2A para tener una fuente auxiliar de alimentación, quedando la configuración como se muestra en la figura Figura 3.7. De esta forma el sensor de corriente puede estar alimentado con la batería o la línea de alimentación alterna sin tener el problema de apagarse de repente.

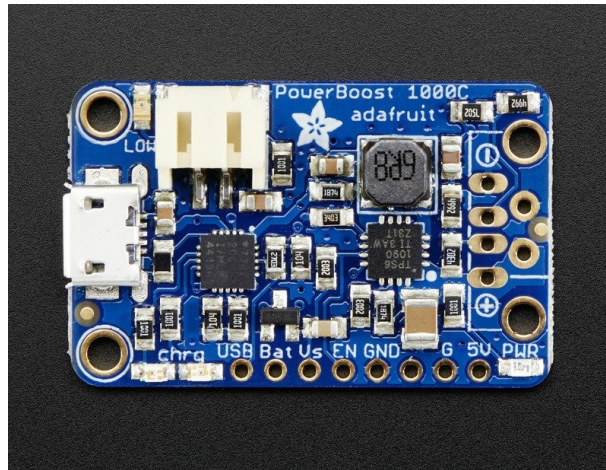


Figura 3.6 Módulo de carga Adafruit Powerboost 1000C.

*Imagen recuperada de
<https://learn.adafruit.com/adafruit-powerboost-1000c-load-share-usb-charge-boost>*

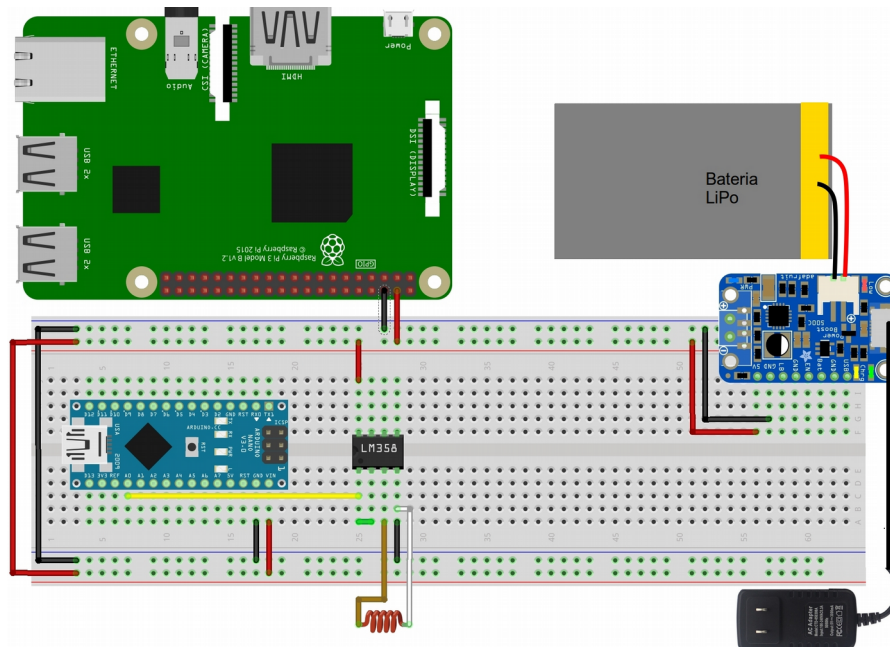


Figura 3.7 Diagrama de alimentación del sensor de corriente.

3 Desarrollo

3.2.3 Electrónica del sistema embebido

El circuito básico para la lectura se basó en la conexión entre sensor SCT-013 con el encapsulado LM358. El sensor tiene una salida desde los -1V hasta 1V, mientras que el ADC del Arduino requiere un voltaje de entrada de 0V a 5V. Por lo cual, se requiere acondicionar dicha señal del sensor para que el Arduino pueda interpretarla. El proceso para acondicionar una señal fue el siguiente:

- Teniendo el rango de voltaje de -1V a 1V, se necesitó escalar la señal hasta un valor que el Arduino pueda interpretar. Se multiplica por un escalar que hace que modifique su tamaño hasta un valor deseado, en este caso por 5V.
- La escala resultante fue de -5V a 5V. Este rango de valores son los que toma la señal senoidal. Una onda senoidal tiene una parte positiva y una negativa, pero el Arduino solo convierte de 0V a 5V. Por ende se necesita rectificar la señal para el ADC con ayuda de un rectificador de onda.
- Una vez acondicionada y rectificada la señal, entra al ADC del Arduino para realizar el muestreo y convierta la señal del dominio continuo al dominio discreto.

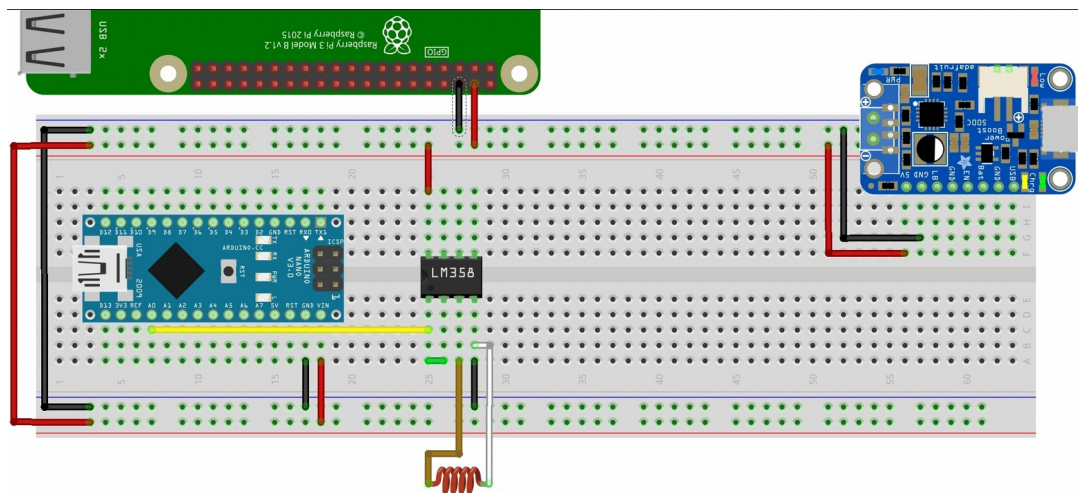


Figura 3.8 Diagrama del sensor al microcontrolador

La forma en la que se realiza el acondicionamiento de la señal sea leída por el ADC del Arduino es rectificando la señal de entrada hacia el sensor para que de esa manera se trabaje con la parte positiva de la señal (en este caso el conector beige del sensor de la Figura 3.8) con el LM358, ya que este circuito integrado trabaja con dicha señal en una de sus entradas y de esa manera hacer una rectificación de media onda como resultado.

El amplificador operacional LM358 tiene una configuración de seguidor de voltaje (que actúa como diodo ideal) para colocarse esta forma, y que corta la parte negativa para poder ser leída por el ADC. Entonces con ayuda del Arduino se configuran los pines de lectura análoga para que trabajen con una referencia de 1.1V, es decir que el valor máximo a leer por el pin analógico es de 1.1V de entrada y el ADC del Arduino pueda trabajar con los valores que le llegan a su pin. Esta configuración se hace porque el valor máximo del sensor es de 1V (a 30A), dando como resultado que el voltaje máximo del ADC, fuera de 1.1V para dejar 0.1V de tolerancia. En la Figura 3.9 se observa la conexión del amplificador operacional con el sensor de corriente, donde se muestra físicamente la conexión.

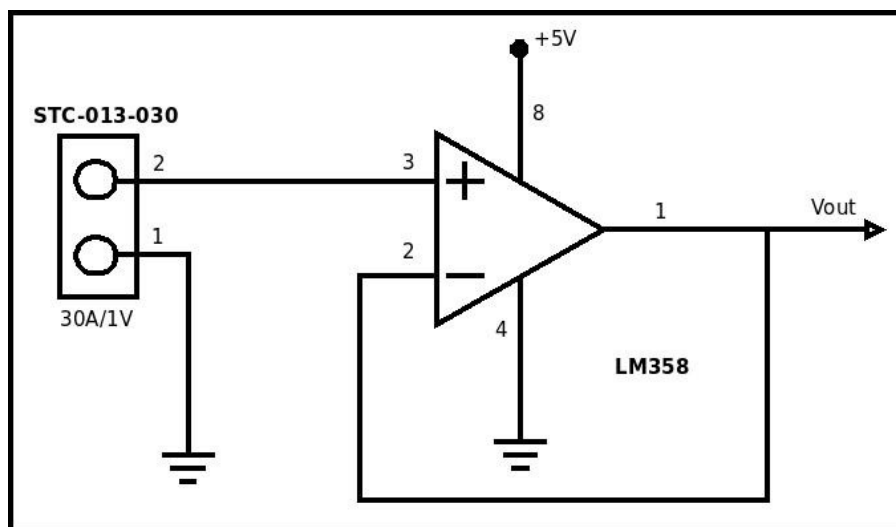


Figura 3.9 Diagrama de conexión simbólico del LM358 con el sensor SCT-013-030

El circuito completo es alimentado por una batería de LiPo de 5V. Tanto el Arduino como la Raspberry pueden funcionar bien con 5V. El Arduino puede ser alimentado por medio de una fuente externa que puede ir de los 5V a los 12V para su funcionamiento sea el adecuado. El módulo PowerBoost que se utiliza especialmente mantiene la carga en 5V para que tanto el Arduino como la Raspberry funcionen adecuadamente.

3 Desarrollo

El Arduino se alimentó en la fuente externa por su pin de V_{in} (voltaje de entrada) y la tierra compartida con el circuito de lectura, con la tierra de el regulador y la tierra de la Raspberry para tener una tierra en común. En el pin de 5V del Arduino se utilizan para poder alimentar al LM358 y se encuentre en funcionamiento constante. El pin A0 que es el pin analógico 0 se conecta la salida V_{out} o el pin 1 del LM358 de donde sale la señal rectificada del sensor para poder leerla con el ADC interno del Arduino como lo indica en la Figura 3.10.

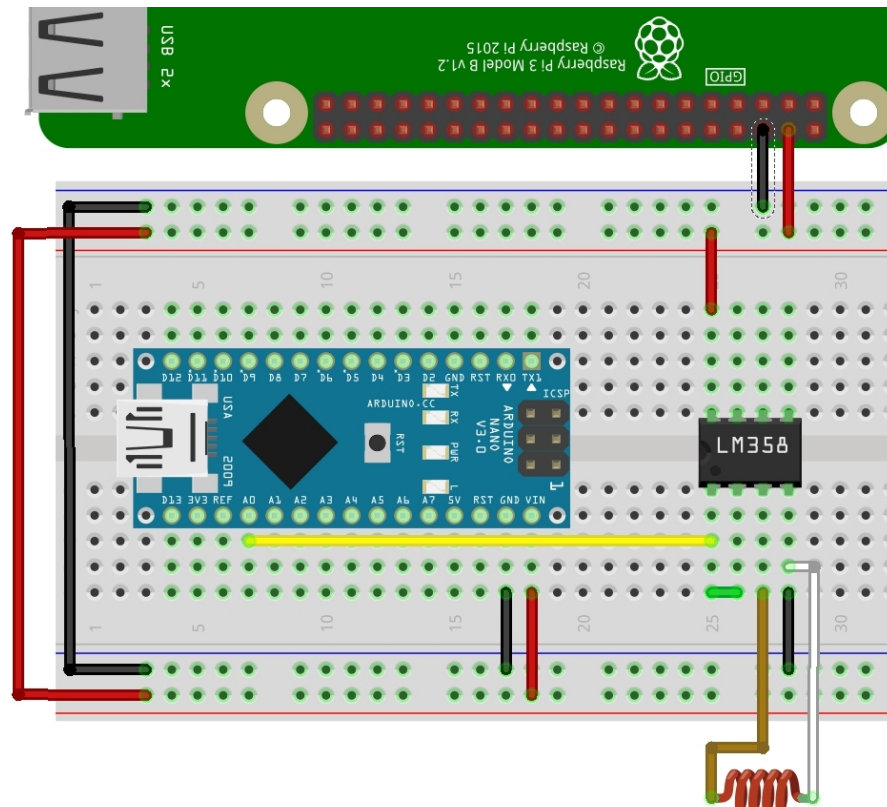


Figura 3.10 Conexión del sensor SCT-013-030 con el LM358

Estas adecuaciones se realizaron partiendo de la suposición de que la señal que genera el sensor de corriente es simétrica, por lo cual se puede tomar el rango de valores de voltaje ($0V$ a $5V$) permitidos por el microcontrolador Arduino solamente tomando la parte positiva de la onda. En general, las señales no siempre son simétricas debido a que las cargas no siempre son lineales.

3.2.4 Cálculos de la corriente eléctrica

El convertidor analógico digital (*ADC, analog-to-digital converter por sus siglas en inglés*) que utiliza el Arduino tiene una resolución de 10 bits que dan 1024 valores, estos se interpretan como un 0 siendo el mínimo valor en binario hasta un 1023 en binario que es el máximo valor que maneja el convertidor ADC, convirtiendo el binario a decimal. Dado que el voltaje máximo que otorga el sensor es de 1.1V, primero se realizó una conversión de escala para ocupar toda la resolución del convertidor analógico digital, como se muestra en la Fórmula 3.1.

$$\text{VoltajeMedido} = \text{ValorAnalogico} \left(\frac{1.1V}{1023} \right)$$

Fórmula 3.1 Cálculo de cambio de escala

El valor analógico es multiplicado por un factor de conversión donde 1023 que es el valor máximo corresponde a 1.1V de la referencia interna del Arduino y de esa manera tener un voltaje para interpretarlo por el Arduino. Dado que el sensor maneja una relación de 1V/30A, es decir, que cada 30A nos da 1V en la medición, se requiere convertir la medición de voltaje a su representación de corriente que se obtiene por la Fórmula 3.2.

$$\text{CorrienteMedida} = \text{VoltajeMedido} \left(\frac{30A}{1V} \right)$$

Fórmula 3.2 Conversión a corriente

Con esto se obtiene la corriente de medición con el voltaje previamente obtenido multiplicado por la relación del sensor donde cada 1V obtenido se refiere a 30A leídos para tener una lectura deseada.

Una vez teniendo la corriente en cada instante se puede obtener la corriente eficaz o corriente RMS que proviene del sensor. La corriente eficaz es la corriente que es capaz de producir el mismo trabajo que el valor en corriente directa. A partir de la función de corriente la corriente eficaz se puede obtener mediante la Fórmula 3.3.

3 Desarrollo

$$I_{rms} = \sqrt{\frac{1}{T} \int_0^T i^2(t) dt}$$

Fórmula 3.3 Corriente RMS en el dominio continuo

En donde T es el periodo en el que se muestrea la corriente e i es una función de la corriente medida con el sensor en cada instante de tiempo. Dado que la medición del sensor no es una medición continua, se tienen muestras discretas cada cierto tiempo, entonces la formula se tiene que llevar al dominio discreto, quedando la Fórmula 3.4. Para el valor de N se toma el número de muestras dentro de un periodo de lecturas del sensor.

$$I_{rms} = \sqrt{\frac{1}{N} \sum_{n=0}^N i_n^2}$$

Fórmula 3.4 Corriente RMS en el dominio discreto

Estos cálculos fueron de ayuda, porque cada valor de corriente RMS que fue registrado por el sensor debe ser almacenado en un archivo para poder enviarlo al servidor y el usuario pueda consultarlo en cualquier momento. También los cálculos fueron implementados dentro del Arduino Nano para obtener la corriente y hacer la conversión a RMS, puesto que previamente se mencionó que el Arduino obtiene valores pico.

3.2.5 Interconexión Arduino-Raspberry

Dado que la Raspberry Pi no tiene ADC, se utilizó el Arduino como interfaz entre la Raspberry Pi y el sensor. Además, la Raspberry Pi es la encargada de conectarse por Wi-Fi a la API, almacenar los datos temporalmente y enviarlos por medio de peticiones web al servidor donde se encuentra la API.

Para la comunicación entre la Raspberry Pi y el Arduino se utilizó el puerto serial que ambos tienen en común. El Arduino, una vez teniendo la corriente eficaz, envía los datos por su puerto serial con el comando de “*println()*” que básicamente envía el dato seguido de un salto de línea “\n”, esto para que la Raspberry Pi pueda detectar un valor nuevo para almacenarlo en el JSON.

Una vez ya alimentando la Raspberry Pi con el voltaje de 5V del regulador en los pines de alimentación (pin 2 para +5V y pin 4 para GND), la Raspberry Pi comienza a funcionar. Lo primero que la Raspberry hace una vez encendida es correr el script que tiene el programa para la recepción y el envío de los valores de la corriente. El script está contenido en un archivo llamado *rc.local*, el cual es una ruta dentro del sistema operativo Linux (En este caso de Raspian) donde ejecuta comandos de la terminal. El script ejecuta las instrucciones del código para abrir el puerto serie y detectar cada salto de línea que le llega por su puerto serial estableciendo una conexión de comunicación asíncrona con el Arduino. La conexión se puede llevar a cabo de dos formas:

- En la primera se utiliza los pines de comunicación serial tanto del Arduino como los de la Raspberry. En el caso del Arduino el pin digital 0 es también el pin de recepción de datos (RX) y el pin digital 1 es el pin transmisor de datos (TX). Los pines de la Raspberry para la comunicación serial es el pin 8 para transmisión y pin 10 para recepción. Por tanto, el pin 0 del Arduino va hacia el pin 8 de la Raspberry, y el pin 1 del Arduino hacia el pin 10 de la Raspberry y de esa manera se puedan comunicar. Un ejemplo de esto se puede ver en la Figura 3.11.
- La segunda forma es hacer la conexión por puertos serial entre el Arduino y la Raspberry usando los puertos USB serial que contiene cada una de las placas por medio de un cable USB como lo muestra en Figura 3.12. Esta segunda opción, dado que es la más sencilla y práctica, es la que se implementó en el sensor de corriente.

3 Desarrollo

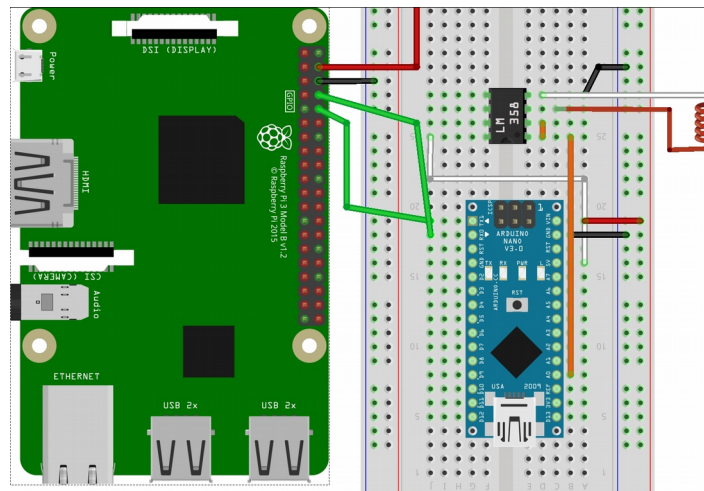


Figura 3.11 Conexión por pines serial

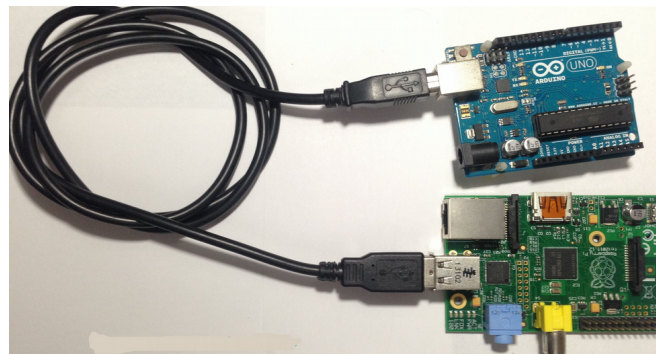


Figura 3.12 Conexión por USB serial

3.2.6 Software del sensor de corriente

Con el sistema operativo de Raspian se implementó un programa que es el encargado de hacer el manejo de los datos. Los datos son recibidos por el puerto serial de donde llegan por medio del Arduino, se almacenan temporalmente y son enviados en formato de archivos JSON hacia el servidor a través de Wi-Fi consumiendo los servicios web expuestos por la API para almacenar la información a la base de datos.

El programa se escribió en Python usando un paradigma multihilo, el cual lleva acabo múltiples tareas, ejecutando una función en específico en cada hilo. Un hilo o un Thread en programación es un elemento que permite a las aplicaciones ejecutar múltiples operaciones de forma concurrente dentro del mismo proceso principal. El programa contiene tres hilos donde se realizan subtareas para adquirir y enviar datos, más un hilo principal donde maneja cada hilo anterior. Los hilos anteriores tienen su propia tarea:

- El primer hilo se encarga de la recepción de los datos provenientes del bus serial. Los datos son recibidos por la Raspberry Pi cada medio segundo. Para almacenarlos junto a su marca de tiempo en formato (año-mes-día hora:minuto:segundo.milisegundos), la cual se adquirió para que la Raspberry Pi siempre este leyendo y evitar perder datos en el proceso.
- El segundo hilo se encarga de la creación de los archivos JSON con la información que proviene del Arduino para almacenar 10 muestras para cada archivo además de el numero del dispositivo de donde se obtuvieron los datos para después guardarlos en su carpeta destino.
- El tercer hilo hace el recorrido de la carpeta donde se almacenaron los archivos, tomar un paquete con veinte lecturas en un archivo JSON para enviarlos hacia el servicio web y una vez recibida la confirmación de que se entregaron, borra los archivos que fueron enviados satisfactoriamente.

Todos los hilos fueron manejados dentro de un hilo principal que los ejecuta iniciando el programa. Además, el hilo principal genera un único identificador de dispositivo que se obtiene de la dirección MAC de la Raspberry y que es anexado en cada lectura; de esta forma se consultan los datos de manera sencilla y se asignan los valores dependiendo del dispositivo. En dado caso de que se cancele el hilo principal los hilos internos detectan el cambio por medio de un demonio para dejar de ejecutarse.

Un demonio (daemon en inglés por sus siglas Disk And Execution MONitor) es un proceso que se ejecuta en un segundo plano, y a su vez puede ser ejecutado varias veces o reiniciado en caso de matar el proceso según el sistema lo requiera, en este caso el demonio se encargó de matar todos los procesos o hilos antes mencionados que dependan del proceso principal.

El medidor de corriente también cuenta con un algoritmo implementado en otro código que corre de manera concurrente². Este algoritmo lo que hace es contar 500 archivos JSON que se encuentren en el sistema para poder eliminarlos del medidor de corriente. Esto lo hace para que el medidor de corriente no se sature de archivos JSON que no se hayan podido enviar por algún error de conexión.

2 Los dos procesos no se ejecutan de manera simultánea, sino que se alternan su turno para ser atendidos.

3.3 Sistema de adquisición de datos (API)

El servicio web se conectó a la base de datos por la misma dirección web de www.labmovilidad.unam.mx/monitor solo que en este caso es por un puerto en específico de comunicación hacia la base de datos de MongoDB. Los procedimientos para insertar o extraer información a la base de datos son atendidos y realizados por medio del puerto 27017 que corre de manera local en el servidor.

La API se necesita conectar a la base de datos para poder agregar los datos recabados por el medidor de corriente. Dentro de la base de datos existe una partición destinada para almacenar de corriente que contiene cada archivo JSON. La API funciona como un cliente y se conecta a esta partición dentro de la base de datos. Esta partición dedicada se llama raspi.

3.3.1 Comunicación dispositivo - API con Flask y Gevent

La API se instaló en un servidor dentro de la unidad de posgrado de la Facultad de Ingeniería de la UNAM. Desde ahí, atiende cada una de las peticiones web que le llegan. Cada petición web que proviene desde el dispositivo embebido hacia la API es recibida por la IP pública que se encuentra en otro servidor que cuenta el Laboratorio de Biorrobótica de la Facultad de Ingeniería de la UNAM. Después, ese servidor redirige cada una de las peticiones hacia el servidor que tiene la API. De igual manera, si se necesita enviar alguna petición desde la página web hacia la API, la petición la redirige hacia el servidor y viceversa como se muestra en la Figura 3.13.

La API fue desarrollada con Flask, que es un framework en Python dedicado para crear aplicaciones web de manera sencilla. Esta constituido en parte con la especificación WSGI (Web Server Gateway Interface), que es una convención de llamada simple para que servidores web puedan reenviar solicitudes hacia aplicaciones web o a frameworks escritos en Python (Simionato, 2007). Para renderizar las páginas web se apoya de Jinja, que es un motor de plantillas para Python.

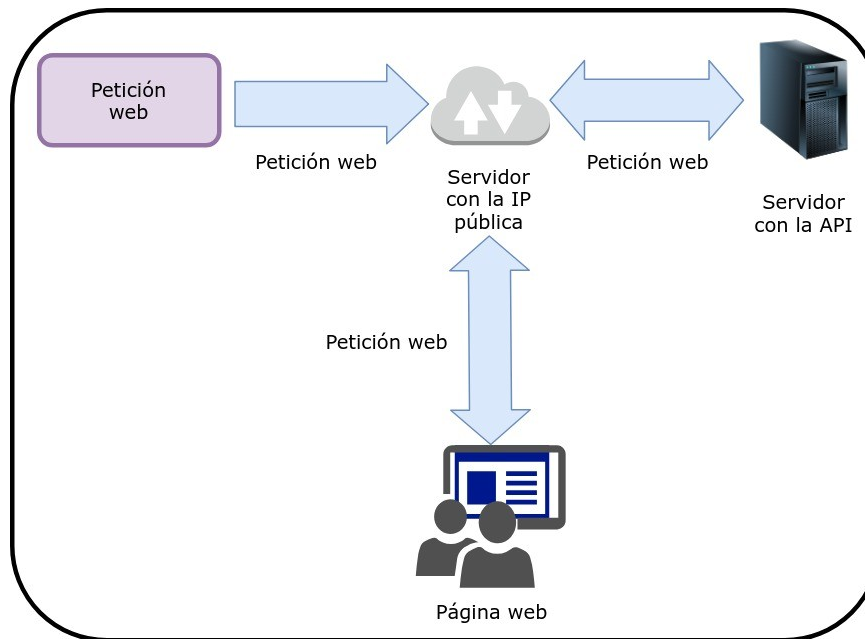


Figura 3.13 Flujo de peticiones web del sistema

Algunas de las características que Flask contiene y que se tuvieron en cuenta para la realización de la API del sistema de manera sencilla son:

- Contiene un servidor para desarrollo y depuración.
- Capacidad de atender solicitudes RESTful.
- Soporte para cookies seguras en sesiones desde el lado del cliente.
- Compatibilidad con WSGI versión 1.0.
- Extensiones disponibles para mejorar sus características que se necesiten.

Sin embargo, Flask es un entorno para desarrollador y depuración, lo que hace que su funcionalidad se limite si se utiliza dentro un entorno de producción. Por lo cual otra herramienta que se utiliza en conjunto para el funcionamiento de la API es Gevent. Gevent es una biblioteca de Python, la cual gestiona los eventos que recibe la API de manera síncrona, lo que permite una alta escalabilidad y robustez (Gevent, 2019).

Gevent cuenta con una herramienta llamada pywsgi que tiene como función crear servidores que reenvían las solicitudes hacia aplicaciones web escritos en Python (servidores WSGI), de tal manera que puedan atender las peticiones http (Gevent, 2019). El uso de Flask y Gevent dentro del servicio interactúan de la siguiente manera: con Flask se declaran los

3 Desarrollo

parámetros bajo peticiones HTTP, mientras que con Gevent levanta el API para hacer la comunicación entre las peticiones y la API.

La API cuenta con diferentes acciones dependiendo del tipo de petición web que recibe, ya sea de la página web o desde el dispositivo y es redirigida a su sección dependiendo de la url que se asignó por cada petición. En la Tabla 3.2 se aprecia la implementación de las peticiones GET y POST para realizar tareas de inserción, consulta y manejo de los datos.

Los pasos en los que se fue desarrollando la API conforme fue establecida la estructura de gestión para los archivos son descritos a continuación:

- La API que se programó en Python con las herramientas de Flask y Gevent se encuentra instalada en el mismo servidor en donde se encuentra la base de datos. Todo el manejo de los datos se lleva a cabo por medio de la dirección `www.labmovilidad.unam.mx`. La dirección HTTP maneja dos apartados: un apartado para la adquisición de datos y uno para búsqueda y cálculo de datos para poder visualizarlos en la gráfica de la página web.
- Para el almacenamiento de datos provenientes del medidor de corriente por vía internet se utilizó el apartado de una dirección fija que se encuentra alojada dentro del servidor. En este caso la dirección destino es `www.labmovilidad.unam.mx/tesismonitor/api/test/data`, en donde el apartado data es para el almacenamiento de todos los datos a la base de datos.
- Para realizar la búsqueda de los datos se utiliza la dirección HTTP de `www.labmovilidad.unam.mx/tesismonitor/api/consult`, donde consult es el apartado global donde se realizan las búsquedas. Las búsquedas son de manera específica, ya que se necesita buscar los datos por medio de un rango de fechas; esto para que el sistema pueda visualizar por periodos el consumo eléctrico. Los parámetros de búsqueda por medio de la dirección son el identificador del medidor de corriente (id), una fecha mínima(fmin) y una fecha máxima(fmax).
- El id es para identificar el dispositivo del cual se tienen las lecturas, pues este sistema nos da la facilidad de monitorear varios dispositivos puestos en diferentes lugares. El valor por el cual busca es con la dirección MAC antes obtenida del medidor de corriente. Ejemplo `id=b8:27:eb:1f:34:8b`.
- Los parámetros fmin y fmax son los rangos de fechas a buscar, es decir, son la fecha inicial y la fecha final a buscar para la base de datos. El formato de la fecha está dado de la siguiente manera: Año-Mes-día hora:minuto:segundo.milisegundo. Ejemplo `fmin=2018-10-13%2019:00:00.000000`, donde %20 es un espacio para protocolos http.

URL	/tesismonitor/api/consult	/tesismonitor/api/price	/tesismonitor/api/test/data
Tipo de petición	GET	GET	POST
Parámetros	<ul style="list-style-type: none"> • Id • fmin • fmax 	<ul style="list-style-type: none"> • Id • fmin • fmax 	No aplica
Tarea	Realiza la consulta de los datos para obtener cada valor de la corriente.	Realiza la consulta de los datos para obtener el kilowatt hora.	Inserta los archivos JSON a la base de datos.
Descripción	Con los parámetros tomados de la URL se hace la búsqueda de los valores de la corriente que fueron almacenados para su dibujado en la página web.	Con los parámetros tomados de la URL se hace la búsqueda de los valores de la corriente que fueron almacenados y calcula los kilowatt hora.	Cada una de las peticiones que recibe tienen el archivo JSON embebido dentro del cuerpo de la petición, los obtiene y la API los va insertando dentro de la base de datos.
Ejemplo de respuesta	<pre>{ "results": [{ "fecha": "2019-10-27 20:00:14.639873", "irms": 2.061 }, { "fecha": "2019-10-27 20:00:30.120763", "irms": 2.088 }] }</pre>	<pre>{ "results": [{ "_id": null, "sumIRMS": 2.7440589434262477 }] }</pre>	No aplica

Tabla 3.2 Respuestas de las peticiones web con la API

- De tal manera que para que el servicio web realice una búsqueda completa de las lecturas del dispositivo cuya dirección MAC es b8:27:eb:1f:34:8b con la fecha del 13 de octubre de las 19:00hrs a las 21:00hrs es de la siguiente manera:
www.labmovilidad.unam.mx/tesismonitor/api/consult?id=b8:27:eb:1f:34:8b&fmin=2018-10-13%2019:00:00.000000&fmax=2018-10-13%2021:00:00.000000.

3 Desarrollo

Para consultar los consumo energético se requieren los mismos parámetros de id, fmin y fmax que son colocados en la url www.labmovilidad.unam.mx/tesismonitor/api/price. Por ejemplo, si se quiere consultar el consumo del medidor de corriente con dirección MAC es b8:27:eb:1f:34:8b con la fecha del 13 de octubre de 2019 de las 19:00:00hrs a las 21:00:00hrs es de la siguiente manera: www.labmovilidad.unam.mx/tesismonitor/api/price?id=b8:27:eb:1f:34:8b&fmin=2018-10-13%2019:00:00&fmax=2018-10-13%2021:00:00.

3.3.2 Base de datos con MongoDB

Uno de los manejadores de bases de datos con estructura no relacional³ más comúnmente utilizado es MongoDB. La base de datos MongoDB como lo menciona su documentación (MongoDB, 2018) es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consultas e indexación avanzado. La base de datos MongoDB brinda al usuario:

- Almacenamiento de documentos en tipo JSON flexibles, que cada documento contiene diferentes campos y sus estructuras de datos se pueden ir modificando.
- Su modelo de documentos concuerda con los objetos que existen en el código de la aplicación para facilitar trabajar con los datos.
- Realiza sus consultas ad-hoc, lleva a cabo la indexación y agrega los datos en tiempo real permitiendo acceder a datos y analizarlos de manera sencilla.
- Da portabilidad de que la base se encuentre en cualquier parte, conectándola a la nube utilizando sus recursos sin restringir el acceso.
- Puede entregar la información hacia las aplicaciones pese a enfrentar alguna adversidad de manera mundial por medio de la replicación y de auto recuperación.
- Realiza las cargas de trabajo que son analíticas y operativas dentro de un mismo clúster.

MongoDB facilita su manejo, pues da a los desarrolladores todas las funcionalidades para cubrir los requisitos que se necesiten manejar, dando drivers para al menos 10 lenguajes de programación diferentes. Uno de ellos es con Python llamado PyMongo, que es una biblioteca que contiene herramientas para trabajar con MongoDB, y es la forma recomendada de trabajar con MongoDB de Python (MongoDB, 2019). PyMongo hace que los datos recibidos por el servicio web sean agregados cada uno hacia a la base de datos.

3 Es una base de datos la cual no tienen un identificador que sirva de relación entre un conjunto de datos y otros.

Las inserciones hacia la sección de la base de datos se llevan acabo con un método llamado *insert_many*, que lo que hace es recorrer el archivo para obtener todos los datos que se encuentren en formato JSON y después insertar los archivos dentro de la base de datos. La ventaja de que sean datos en formato JSON es que MongoDB acepta la estructura del JSON para añadirlo como un objeto con atributos.

La estructura de MongoDB es de una base de datos no relacional, es decir, que es una base de datos diseñada para modelos de datos que no son específicos y tienen esquemas flexibles para la creación de aplicaciones modernas. Cada archivo contenido en la base de datos de MongoDB tiene un id único (Object Id) proporcionado por la misma base de datos para identificar cada archivo dentro.

3.3.3 Algoritmos de búsqueda

Existen dos métodos de búsqueda que la API realiza dentro del servidor, cada uno implementado en una petición GET con diferente url:

- La url que contiene la ruta */tesismonitor/api/consult* lleva acabo una búsqueda dentro de la base de datos con el uso de la función *find* de mongoDB.
- La url que contiene la ruta */tesismonitor/api/price* realiza una función de agregación donde por parámetros se filtran y se realizan los cálculos para obtener el total de consumo de corriente en el lapso de tiempo con la función *aggregate*.

Las búsquedas con la función *find* se llevan acabo con los los valores de los parámetros dados en el la dirección http (*id, fmin, fmax*) el sistema los interpreta como sus valores de búsqueda de mongoDB. Las búsquedas en mongoDB son realizadas como en otras bases de datos con operadores lógicos relacionales para encontrar los datos que se necesitan consultar (Figura 3.14).

En este caso el operador lógico es *AND*, que lo que hace es unir los parámetros de búsqueda como una sola instrucción. Para la búsqueda por rangos se utiliza dos operadores que son *gte* (*greater than or equal*) y *lte* (*less than or equal*) que son el máximo y mínimo valor a buscar.

La función de *find()* recibe de parámetro toda la sentencia ya formada para que haga la búsqueda ya con los operadores relacionales. Y como respuesta de la búsqueda devuelve los valores en formato JSON que entran entre los rangos que se solicitan.

3 Desarrollo

Como resultado de la búsqueda, regresa una lista que contiene una copia de cada diccionario en formato JSON dentro del rango impuesto por la instrucción *AND* en la base de datos. Los diccionarios que se encuentran en lista retornan solamente el valor de la corriente y la fecha que se obtuvo la lectura por parte del dispositivo.

Al final los archivos se generan con la información de cada diccionario que tiene la lista y son enviados hacia la página web por medio de la API para ser bosquejados dentro de la gráfica que contiene la página web.

```
{
  "$and": [
    {"id": "b8:27:eb:1f:34:8b"},
    {"fecha": {"$gte": "2019-09-01 00:00:00"}},
    {"fecha": {"$lte": "2019-09-25 00:00:00"}}
  ]
}
```

Figura 3.14 Estructura de una función de búsqueda en PyMongo

La función de agregación es una consulta que se realiza para agrupar los datos y se calculen valores a partir de ellos. Para agrupar los datos se pueden colocar parámetros para restringir los límites, esto para realizar los cálculos solo a los datos que se encuentren dentro del rango (Figura 3.15).

Los parámetros que limitan los datos a agrupar se colocan después de la instrucción *\$match*. Esta instrucción puede implementar la misma instrucción *AND* y los rangos de *gte* y *lte* que se implemento en el *find* para acortar su rango de búsqueda.

Seguido del acotar los rangos, se crea la lista con los diccionarios en formato JSON resultantes de la búsqueda, para luego recorrer la lista y obtener cada valor de la corriente y de las fechas de cada lectura.

Aparte de la instrucción *\$match*, se utilizó otras dos que son: *\$sample*, que su función es de tomar todos los datos que se encontraron con el *\$match* para obtener un conjunto de submuestras que depende del parámetro de tamaño (en este caso se uso un tamaño de 100000 submuestras); y *\$sort*, que es una instrucción la cual toma un conjunto de datos y los ordena a partir de uno de los valores que contiene el diccionario de manera ascendente (1) o descendente (-1) (ordena el subconjunto anterior ascendente).

```
[
  { "$match": {
    "$and": [
      { "id": "b8:27:eb:1f:34:8b" },
      { "fecha": { "$gte": "2019-09-01 00:00:00",
        "$lte": "2019-10-01 00:00:00" } } },
    ]
  } },
  { "$sample": { size: 100000 } },
  { "$sort": { "fecha": 1 } }
]
```

Figura 3.15 Estructura de una función de agregación en PyMongo

3.3.4 Cálculo del consumo

Con las fechas obtenidas se realizó el cálculo de diferencias de tiempo entre una lectura y otra. Contenidas dentro de la lista, se recorre cada dos valores utilizando la Fórmula 3.5, donde Δt es el valor resultado por cada intervalo de tiempo y t_n y t_{n-1} los valores de las fechas en la lista.

$$\Delta t = t_n - t_{n-1}$$

Fórmula 3.5 Cálculo de las diferencias de las fechas

Con los valores de la corriente I_{rms} de cada diccionario, se hizo el calculo de la suma de todos los valores como se muestra en Fórmula 3.6 donde I_{total} es el resultado de la suma de multiplicar el valor I_{rms} cada lectura con el Δt obtenido anteriormente, n el número de elementos en la lista y m es el último elemento de la lista.

3 Desarrollo

$$I_{total} = \sum_{n=0}^m I_{rms_n} * \Delta t_n$$

Fórmula 3.6 Suma de corrientes RMS

Una vez obtenido el valor de I_{total} , se procedió a calcular la energía consumida tomando en cuenta un voltaje RMS (Fórmula 3.7). El cálculo consistió en realizar la multiplicación de I_{total} y V_{rms} , considerando el voltaje RMS como 127V establecido en todo el país, resultando en la energía eléctrica consumida (EEC_{msec}) con unidades de Watt-milisegundos.

$$EEC_{msec} = I_{total} * V_{rms}$$

Fórmula 3.7 Cálculo de la energía eléctrica consumida en Watts-milisegundos

Para obtener los Watt-hora(Wh) a partir de la EEC_{msec} se consideró la fecha donde inician las lecturas hasta la última fecha correspondiente al ultimo valor de corriente para conocer el tiempo transcurrido. El valor de la variable EEC_{msec} se encuentra en unidades Watt-milisegundos.

Por lo cual se realizó una conversión de milisegundo a horas y a la vez multiplicar. Esta relación se obtiene por medio de la Fórmula 3.8 donde EEC_{hrs} es igual a la EEC_{msec} multiplicado por el factor de conversión de los milisegundos a horas.

$$EEC_{hrs} = EEC_{msec} * \left(\frac{1 \text{ Segundo}}{1000 \text{ Milisegundos}} \right) * \left(\frac{1 \text{ Minuto}}{60 \text{ Segundo}} \right) * \left(\frac{1 \text{ Hora}}{60 \text{ Minutos}} \right)$$

Fórmula 3.8 Calculo de energía eléctrica consumida en Wh

Como los datos se necesitan estar en kilowatt-hora se calcula una última conversión para pasar los watt-hora previamente obtenidos a kilowatt-hora. La conversión se realizó con la Fórmula 3.9 donde kW_{hrs} que resultó de la operación al multiplicar EEC_{hrs} con el factor de conversión de los watt-hora a kilowatts-hora.

$$kW_{hrs} = EEC_{hrs} * \left(\frac{1 \text{ kWatt}}{1000 \text{ Watt}} \right)$$

Fórmula 3.9 Conversión a kWh

3.3.5 Protocolos de respuesta

Cada acción que realiza la API está seguida por una petición web en HTTP. Las peticiones para añadir datos a la base de datos son por método POST para recibir el archivo que se envió con la petición. Si se recibe con éxito el archivo se responde con un JSON con el código 200 (*status ok en HTTP*). Si se genera un error, se regresa el código 500 (error al procesar la información en http).

Para solicitar los datos se utilizan los métodos GET implementados para que la búsqueda retorne como respuesta todos los JSON resultantes. De igual manera se manejó el código 200 para envío correcto y 500 si se genera un error.

Si el mensaje fue entregado de la manera correcta desde la Raspberry hacia el servidor, los datos son insertados a la base de datos. De igual manera, si los datos consultados por la pagina web son recibidos correctamente, se regresan los datos filtrados y muestreados, para luego ser bosquejados.

3.4 Sistema de visualización

El sistema propuesto requiere de un método donde los datos que se registraron mediante las lecturas que genera el medidor de corriente sean mostrados a los usuarios. Esto permite tener un control de nuestro consumo eléctrico de manera visual. Por ello, se desarrolló una página web donde el usuario puede consultar tanto en tiempo real como en lapsos de tiempo definidos por el mismo, el consumo eléctrico de su hogar, así como un precio aproximado de su consumo mensual utilizando la tarifa que CFE brinda a sus usuarios, dependiendo del subsidio.

El propósito de crear la página web es que se brinde al usuario de la interfaz un fácil uso, sin tener que tener conocimientos técnicos de electricidad así como su funcionamiento. Los parámetros que utilizó la página web se pueden establecer de manera sencilla para que cualquier persona lo pueda utilizar.

3.4.1 Sistema gráfico de los datos

La página web implementó el uso de HTML para crear cada uno de los elementos gráficos que ayudan a facilitar las consultas con la API. La dirección web donde se encuentra la interfaz gráfica es la www.labmovilidad.unam.mx/monitor. La API recibe los parámetros provenientes de la URL que contiene los lineamientos compatibles con los tipos de comandos de peticiones web de la API para llevar a cabo las consultas. La herramienta que nos ayuda para la elaboración de los elementos visuales de la página web es un framework llamado Bootstrap.

Bootstrap (Figura 3.16) es una biblioteca multi plataforma de herramientas creada para el desarrollo de sitios o de aplicaciones web. Utiliza plantillas de diseño con tipografía, formularios, botones, cuadros, menú de navegación y de elementos de diseño basado en HTML, CSS y extensiones de JavaScript.

Las características de Bootstrap (Cochran, 2012) cuentan con soporte en HTML5 y en CSS3, tiene compatibilidad con sitios web o con aplicaciones dedicados a dispositivos móviles y de navegadores de escritorio, el diseño gráfico de la página web se ajusta de manera dinámica con las especificaciones del dispositivo ya sean computadoras, tabletas, teléfonos móviles.

Algunas de las funciones que tiene Bootstrap (Cochran, 2012) son:

- Un sistema de cuadrilla de 940 píxeles de ancho, este puede ser ajustable dentro de sus parámetros y un diseño sensible a los reajustes de tamaño de manera automática (diseño adaptativo o responsivo).
- Una compresión de la hoja de estilo CSS para proveer definiciones básicas de estilos para componentes de HTML; dando una uniformidad dentro del navegador y del sistema de anchura.
- Tiene componentes que Bootstrap reutiliza de interfaces previamente uso, como botones con ciertas características avanzadas, etiquetas, capacidades avanzadas de tipografías en miniatura, formatos para mensajes de alerta y barras de progreso.
- Plug-in de JavaScript que están basados en jQuery, interfaces de usuario como tooltips o cuadros de diálogo, y soportar plug-ins como Modal, Scrollspy, Tab, Popover, Alert, Button, Collapse y Typeahead.

Existen actualmente plantillas basadas en Bootstrap con elementos ya predefinidos para que los desarrolladores puedan crear fácilmente una página web. Una de las plantillas que existen en internet es Admin LTE.

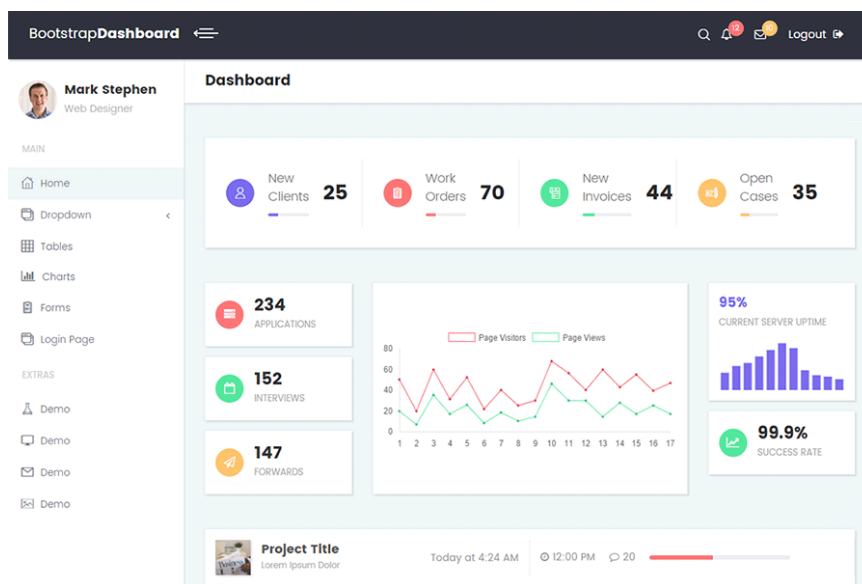


Figura 3.16 Ejemplo de Bootstrap

AdminLTE (Figura 3.17) es una plantilla de aplicación Web con código abierto para crear paneles de control y dashboards. Es una plantilla hecha en HTML que se basa en el marco de CSS Bootstrap 3 que utiliza todos sus elementos que contiene para su diseño y modificar los complementos de uso para así crear un diseño coherente que se puede usar como interfaz de usuario para aplicaciones.

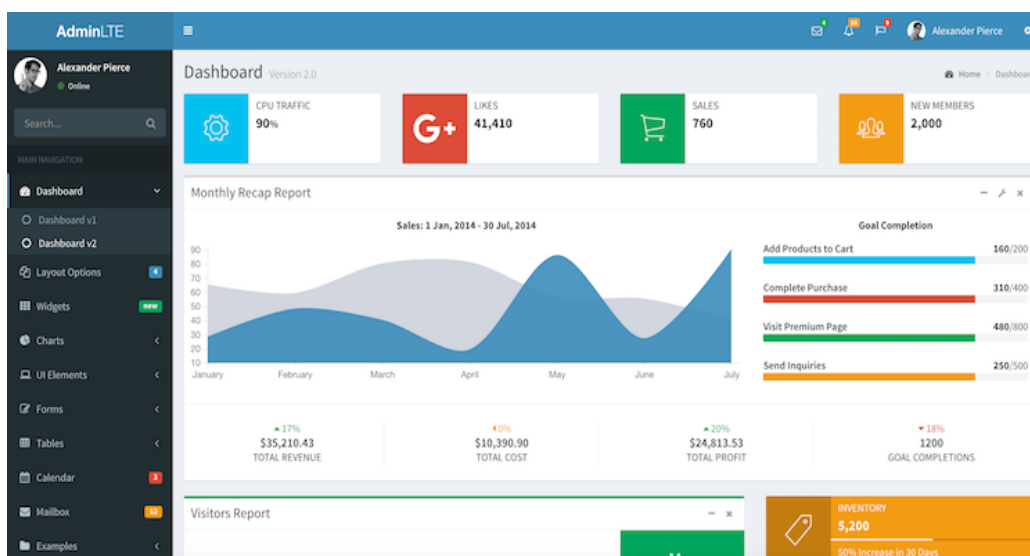


Figura 3.17 Vista previa de la interfaz AdminLTE.

Imagen recuperada de <https://adminlte.io/>

3 Desarrollo

Una vez accedido al sistema de la manera correcta, el sistema redirige al usuario hacia la página de consulta y visualización de los datos. La interfaz esta constituida por lo siguiente:

- Un bloque para visualizar de manera gráfica el historial de consumo (Figura 3.18) durante un cierto lapso de tiempo; esto por medio de introducir el id del dispositivo del cual se desea conocer las mediciones, la fecha y hora mínimas que requiere consultar, así como su fecha y hora máxima para obtener el rango requerido. Los datos son dibujados en una gráfica dedicada a mostrar el historial (Gráfica Historial) la cual tiene una relación intensidad de corriente contra tiempo.

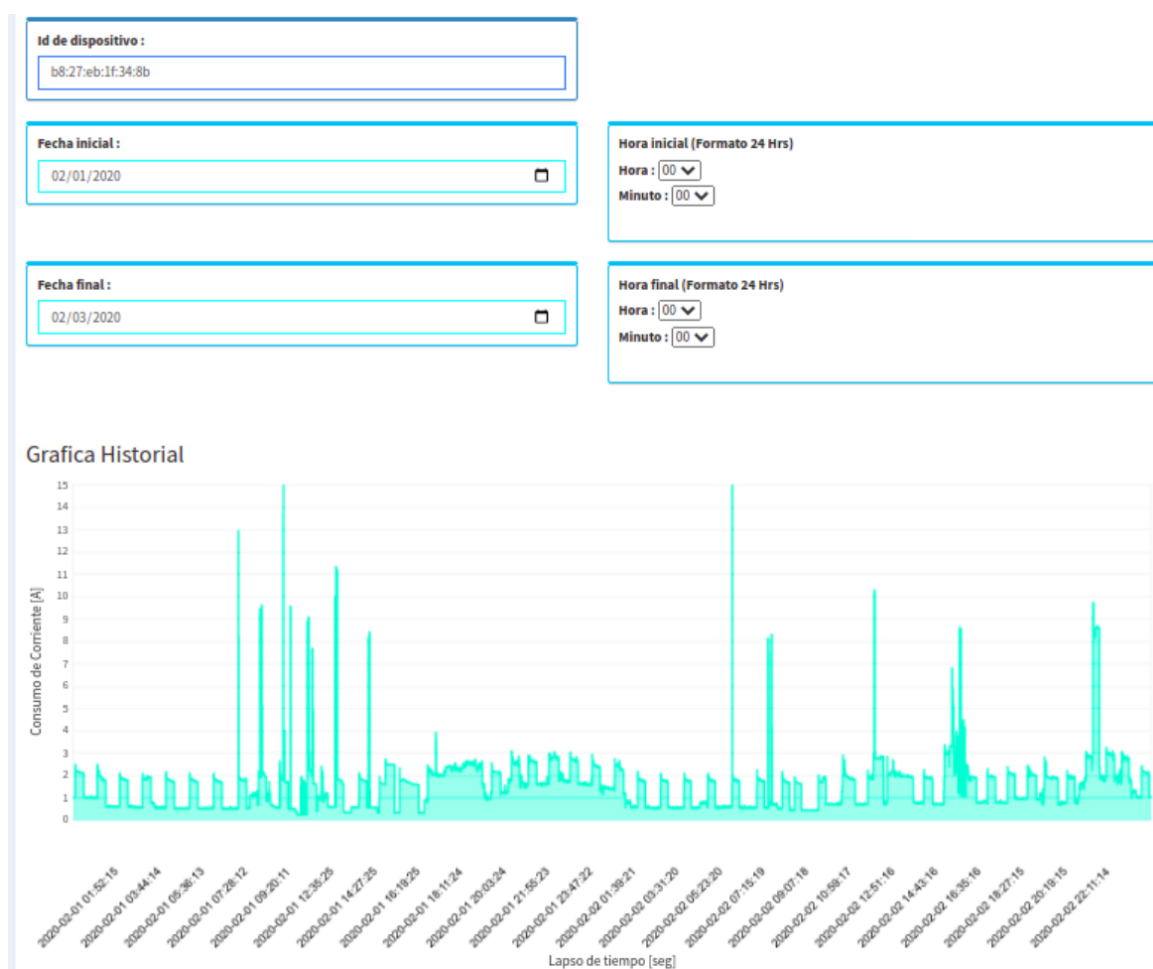


Figura 3.18 Consulta del consumo eléctrico histórico

- El siguiente bloque se utiliza para visualizar en tiempo real el consumo eléctrico que se activa una vez obtenido la id propia del dispositivo a consultar. Los datos resultantes son visualizados en otra gráfica dedicada a mostrar en tiempo real (Figura 3.19). Esta gráfica se actualiza cada 30 segundos con los últimos datos recolectados por el dispositivo en intervalos de una hora.

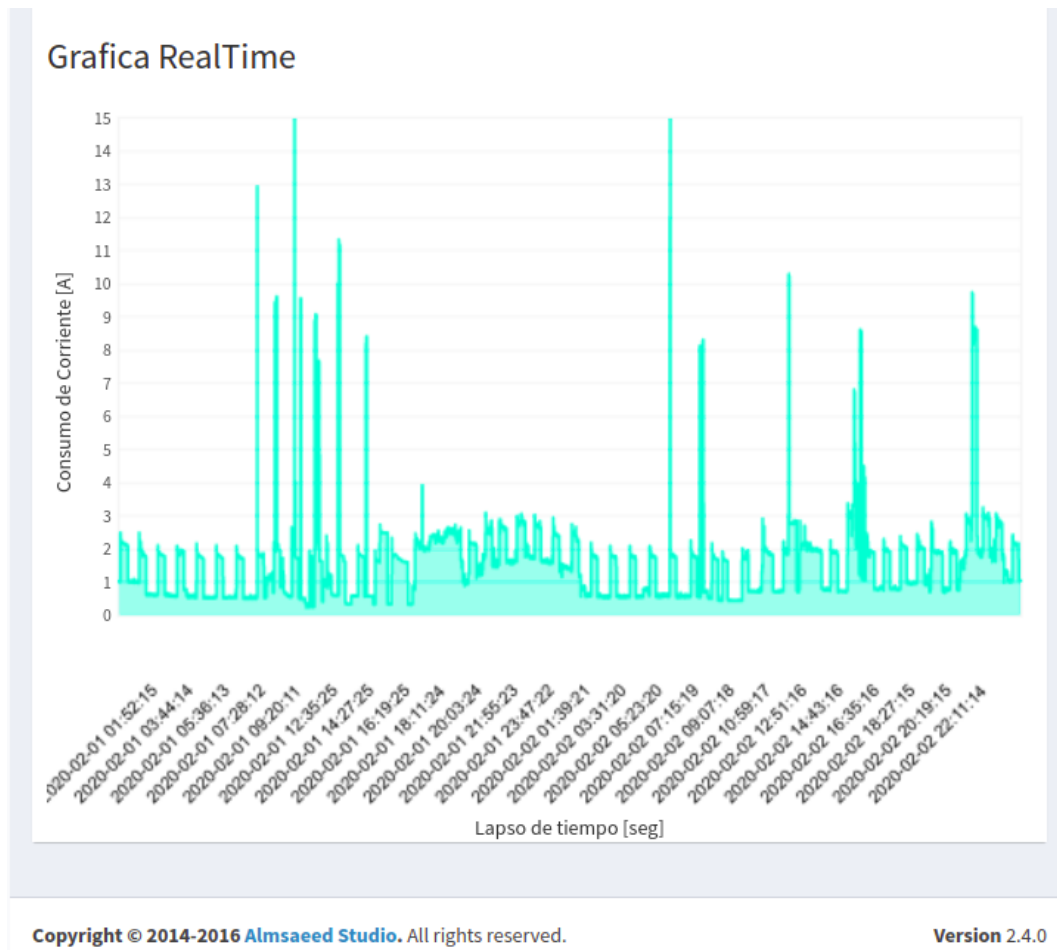


Figura 3.19 Gráfica en tiempo real

3 Desarrollo

- El último bloque se dedica a obtener un precio estimado basado en el cálculo que realiza CFE (Figura 3.20) respecto al consumo eléctrico por medio intervalos de tiempo; es decir, que se encarga de calcular el adeudo monetario con la tarifa con la que trabaja la empresa en ese momento y dependiendo del consumo en cierto tiempo sera el monto aproximado que tendría que pagar. Los rangos de tiempo se obtienen de la consulta histórica, por lo cual se debe primero obtener los valores de las fechas. Tiene un campo donde se ingresa el valor monetario con el que se desea calcular y tomando en cuenta las fechas y horas tanto mínima como máxima, se obtiene el consumo en kWh (kilowatts-hora) y su equivalente aproximado en tarifa a pagar.

Precio Nominal segun CFE

Precio basico

Precio intermedio

Precio excedente

Tu KWh hasta el momento es 9.433KWh Y tu costo es 7.74 pesos M.N.

Si necesita consultar la tarifa correspondiente al periodo, consulte [aqui](#) la página de CFE

Figura 3.20 Cálculo de la tarifa por parte de CFE

3.4.2 Búsqueda y filtrado de datos

Bootstrap al trabajar en conjunto JavaScript también puede utilizar recursos y herramientas para llevar acabo ciertas funciones. AJAX (por sus siglas en inglés Asynchronous JavaScript And XML) es una herramienta para desarrollo web en creaciones de aplicaciones interactivas. Este funciona con un cliente que se mantiene en el navegador la comunicación asíncrona con el servidor en segundo plano, esto es para que la página se actualice sin necesidad de recargar desde el navegador. Esto ayuda a que la página sea más fluida para llevar acabo las consultas.

AJAX esta constituido de cuatro tecnologías que están en la actualidad:

- HTML para el diseño que acompaña a la información.
- Document Object Model accedido con un lenguaje de scripting de parte del usuario.
- XMLHttpRequest que hace el intercambio de datos de forma asíncrona con el servidor web.

AJAX es utilizado tanto en la consulta histórica del consumo eléctrico como en la gráfica en tiempo real, pues se tiene una búsqueda diferente de fechas para ser mostradas en las gráficas.

Para ambas funciones el sistema de solicitud de los datos hacia la base de datos se realiza con ayuda de AJAX para realizar la petición hacia la API. Se tienen ya predeterminado la url ‘https://www.labmovilidad.unam.mx/tesismonitor/api/consult?’ a la cual se le añaden los parámetros id de dispositivo, fecha mínima y fecha máxima para introducir los rangos a la búsqueda de su consulta.

La forma en la que se lleva acabo una consulta de consumo histórico y sea mostrada en la gráfica de historial (Figura 3.21) es la siguiente:

- Se tienen un campo para escribir el id del dispositivo, un campo para la fecha y otro para la hora mínimas a consultar, así como otros campos para la fecha y otro para la hora máximas a consultar.
- Una vez teniendo los campos previos con la información a consultar, se da un click al botón de Get History, el cual ejecuta la función de construir la url con los datos obtenidos.
- La función con AJAX hace una petición GET con la url construida. Si la búsqueda dio un resultado correcto, la api devuelve un arreglo con los datos que se encontraron de acuerdo el rango de fechas establecido, así como su id asociado. Dentro de la misma función se genera un arreglo solo con los datos I_{rms} para ser bosquejados en la gráfica.

3 Desarrollo

- El arreglo con los datos pasa a una función que gráfica cada uno de los valores I_{rms} como puntos en la gráfica corriente eléctrica contra tiempo, mostrando como parámetros las fechas solicitadas en el eje del tiempo y el valor de la corriente eléctrica en el eje del consumo.

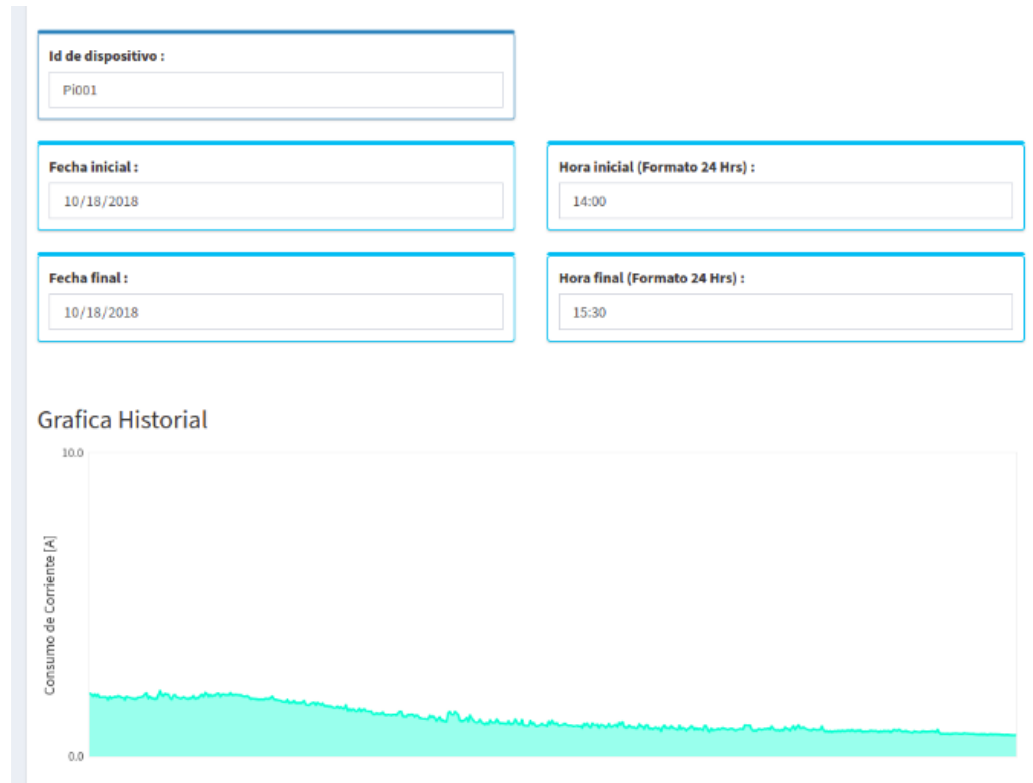


Figura 3.21 Vista previa de la gráfica de historial

La consulta en tiempo real y la gráfica en tiempo real se realizan de la siguiente manera:

- Se requiere que el id del dispositivo esté dentro del campo que esta asociado al id, para que se realice la consulta en tiempo real.
- Con el botón E/D Update se activa la consulta en tiempo real. Una vez teniendo el id, la misma página genera dos fechas: una que se obtiene con el día y hora actual del sistema desfasado por cinco minutos que corresponde a un valor máximo, y a partir de esa misma fecha se obtiene la fecha mínima restando una hora, para la fecha mínima.

- Los parámetros de fecha mínima y fecha máxima se colocan en la url de búsqueda y AJAX lleva acabo la consulta como en la consulta histórica. De misma manera, se genera el arreglo con los datos recuperados por la búsqueda de la API.
- Con los datos obtenidos en el arreglo, se bosquejan los datos en su gráfica de tiempo real. Pasados unos 10 segundos, el sistema vuelve a consultar para recabar los nuevos datos en el rango antes mencionado para volver a graficar. De esa manera da un funcionamiento en tiempo real con los últimos datos recolectados.

Para el cálculo de la tarifa dada por CFE, se requiere de dar ciertos parámetros para que funcione. La forma en la que se lleva a cabo es descrita a continuación:

- Se tiene un campo dedicado para obtener el parámetro de costo que brinda CFE. Este valor puede ser consultado directamente de la página de CFE (la misma página contiene un link que te redirige a la página de CFE), o del tipo de tarifa que aparece en el recibo impreso (los tipos que manejan son 1, 1A, 1B, 1C, 1D, 1E, 1F; y cada uno de los tipos maneja tarifas básica, intermedia y excedente (CFE, 2019)) y según sea la cantidad de kilowatts-hora (kWh).
- Teniendo el valor de la tarifa, con el botón de calcular tarifa toma el arreglo obtenido de la consulta de consumo histórico para tener la lectura de kWh y hace el cálculo del costo correspondiente.
- La página solicita a la API que envíe el archivo JSON con los kilowatt-hora ya calculados para poder calcular el costo dependiendo de la cantidad.
- Para obtener el precio total como lo hace CFE, se realiza conforme las tarifas dependiendo de los kilowatt-hora por medio de multiplicaciones con la cantidad de kilowatt-hora previamente calculada y con los precios que CFE maneja. Se calcula por medio de una de las formulas (Fórmula 3.10) que se aplican dependiendo de el consumo mensual: Si se maneja un consumo menor o igual a los 75kWh se aplica costo total básico; si el rango esta entre los 75 y los 140kWh se maneja un costo total Intermedio; y si el rango sobrepasa los 140kWh se aplica el costo total excedente. Teniendo el resultado con una de las fórmulas es mostrado en un mensaje debajo del botón de calcular tarifa como se muestra en la Figura 3.22.

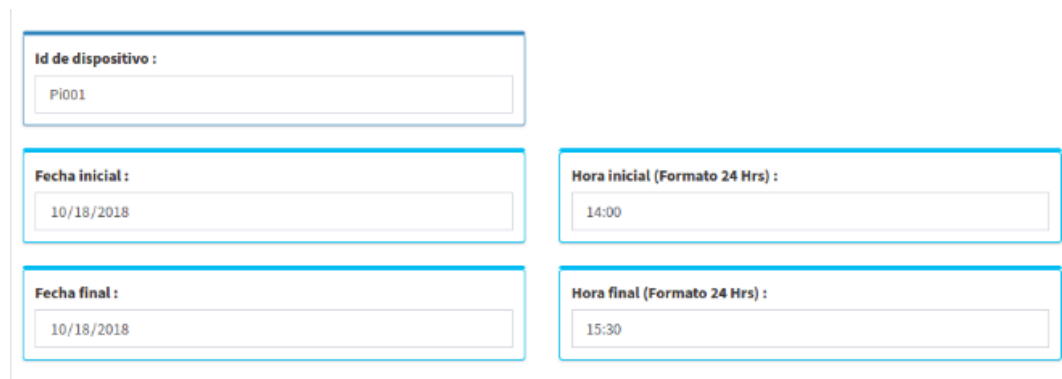
3 Desarrollo

$$\text{CostoTotalBasico} = kWh * \text{Tarifabasica}$$

$$\text{CostoTotalIntermedio} = (75 kWh * \text{TarifaBasica}) + (kWh \text{ restante} * \text{TarifaIntermedia})$$

$$\text{CostoTotalExcedente} = (75 kWh * \text{TarifaBasica}) + (65 kWh * \text{TarifaIntemedia}) + (kWh \text{ restante} * \text{TarifaExcedida})$$

Fórmula 3.10 Costo total por kilowatt-hora



Formulario de cálculo de tarifa según lineamientos CFE. El formulario está dividido en dos columnas y tres filas. La primera fila contiene el campo 'Id de dispositivo :' con el valor 'PI001'. La segunda fila contiene los campos 'Fecha inicial :' con el valor '10/18/2018' y 'Hora inicial (Formato 24 Hrs) :' con el valor '14:00'. La tercera fila contiene los campos 'Fecha final :' con el valor '10/18/2018' y 'Hora final (Formato 24 Hrs) :' con el valor '15:30'.

Figura 3.22 Vista previa del cálculo de la tarifa según lineamientos CFE

3.4.3 Validación de los datos desde la vista

La página tiene funciones en caso de no seguir las instrucciones de uso, en caso de dar algún dato erróneo, y/o dejar algún campo en blanco. Esto es para que la página contenga alternativas en cuestión de hacer un método equivocado. Algunas de ellas son:

- En cuestión de dejar campos en blanco, el sistema lleva acabo ciertas acciones. En caso de que el campo de id se encuentra en blanco al momento de realizar una búsqueda de historial o de tiempo real marca un mensaje de error con el id vacío, puesto que necesita el número del dispositivo para realizar la búsqueda.

- La misma API cuenta con una función para limitar los datos consultados. La forma en la que hace una restricción de los datos es tener un mínimo de datos a mostrar, y si ese intervalo excede por los rangos de las fechas, muestra un dato por intervalo, es decir, que cada cierto valor es enviado cada sub intervalo de la función. Por ejemplo, si los datos que se piden es durante un mes, la función regresa los valores significativos, escalando la gráfica.
- Para la consulta histórica, el sistema solo muestra los valores que se hayan obtenido por el dispositivo desde el día en el que se colocó hasta la última lectura obtenida por el dispositivo, y de esta forma fechas que no se encuentren se informa al usuario por medio de un mensaje que dichas fechas no se encuentran en la base de datos por medio de la página que indica que sus lecturas no se encuentran en la búsqueda.
- Se implementó una validación de los parámetros de búsqueda id, fmin y fmax con el cual se previene insertar cualquier otro texto o caracteres que no tengan el formato correcto. Para ellos se implementó una expresión regular tanto para el id y como la fecha: en el id su expresión regular es `xx:xx:xx:xx:xx:xx` donde las x toman valores de 0 a 9 y de a hasta f; mientras que para las fechas la expresión regular es `xxxx-xx-xx xx:xx:xx` donde las restricciones son año-mes-día hora-minuto-segundo y los valores de x van de 0 al 9 con el formato de fechas en número.
- El sistema se implementó para que las consultas históricas se lleven acabo por mes, porque los costos de CFE son por mes. Entonces dentro de la página web tiene una validación en donde el máximo de días que consulta es de 31 días. Si en caso de que se solicite a la página más de los 31 días, la misma página le comunica al usuario que esta excediendo sus parámetros de consulta y que vuelva a realizar una nueva consulta dentro de los limites permitidos.
- En dado caso que se realice una consulta histórica solo con la id, el sistema regresa los datos de el mes en curso, esto para facilitar el uso de las consultas históricas mensuales. Si la consulta es por medio del id y la fmin o del id y la fmax, las búsquedas se auto-completan con el la fecha actual en la combinación id y fmin, y el primer día del mes para la combinación de id y fmax.
- Los valores de ambas gráficas se acotan a mostrar 20 fechas en la parte del tiempo, porque siendo muchos datos bosquejados no tienen el suficiente espacio para mostrarse dentro de la gráfica. Tomando toda la cantidad de los datos se dividen entre 20 para ver a los cuantos datos mostrar la fecha en la gráfica.

4 Pruebas y resultados

En este capítulo se detalla la implementación del sistema de consumo eléctrico en ambientes reales y su eficiencia dentro del funcionamiento comparado con sistemas que se encuentran en el mercado, o herramientas que hacen la misma función como los medidores de CFE. Además de hacer pruebas de costos aproximados dependiendo del nivel de consumo eléctrico en ciertos periodos de tiempo a partir de los costos y las tarifas que maneja la compañía de CFE.

4.1 Pruebas en ambiente controlado

Una de las pruebas que se llevaron a cabo fue para corroborar la precisión y exactitud de las mediciones del dispositivo embebido, es decir, como la placa Arduino y el sensor de gancho funcionan cuando se calcula la corriente RMS. Para ello se utilizó un ambiente controlado, en el cual se colocaron dos tipos de carga: una lineal que se representa con un foco incandescente resistivo y una no lineal con un foco fluorescente.

Cada uno de los focos tiene información que viene de fabrica acerca de la energía que consumen. A partir de los datos, se necesitó conocer la corriente que usan cada uno para tomarlos como valores teóricos a comparar con las lecturas del sensor. Se consideraron los valores de potencia de cada foco: para el foco fluorescente 40Wh de energía eléctrica que consume, mientras que para el foco incandescente resistivo de 100Wh de consumo eléctrico.

Dado que ninguno de los focos contiene información de la corriente que consumen, se calculó la intensidad de corriente por medio de la energía eléctrica consumida del foco incandescente resistivo con la Fórmula 4.1 dando como resultado 0.7874A para el foco resistivo y 0.3149A para el foco fluorescente.

$$I_{foco} = \left(\frac{EnergíaConsumida_{foco}}{127 V} \right)$$

Fórmula 4.1 Cálculo de la corriente de un foco

Los valores que resultaron a partir de la fórmula fueron comparados con las lecturas que obtuvo el Arduino acondicionado que contiene la fórmulas de la sección 3.2.4 del desarrollo para calcular corriente RMS y verificar que tan fiable son las lecturas que provienen del sensor.

4.1.1 Análisis lineal con el foco incandescente

Teniendo el valor teórico del foco incandescente resistivo, se prosiguió a conectarlo en una base previamente construida y colocar el sensor en uno de las terminales del cableado como se muestra en la Figura 4.1. El gancho fue colocado en el sentido del flujo de corriente, así como se indicó en la Figura 3.3. Una vez conectado el sensor, el Arduino empieza a leer las señales eléctricas que provienen del gancho, pasando por el amplificador operacional que acondiciona la señal.

Las señales eléctricas que mide el sensor son senoidales a una frecuencia de 60 Hz. El algoritmo implementado en el Arduino toma entre 434 a 435 muestras cada medio segundo, y las procesa para obtener un valor de corriente RMS. Tomando en cuenta el teorema de Nyquist, la frecuencia de muestreo utilizada por el algoritmo es suficiente para poder reconstruir la señal.

Los valores pico que retornó el sensor por medio del Arduino rondan desde los 0.756A hasta los 0.772A. Este rango de valores se acercan al valor teórico del foco incandescente que es 0.7874A. Por lo cual, el error relativo resultante de las lecturas del sensor con respecto al foco va desde 1.94% al 4.1%, que es un porcentaje de error mínimo.

También se graficó la señal eléctrica obtenida a partir de las lecturas del sensor, como se muestra en la Figura 4.2. Se tiene que tener en cuenta que la señal original fue alterada con el amplificador operacional para compensar la parte negativa de la señal senoidal, por lo cual se observó como dentro de la gráfica están recortados los picos que representan a los valores correspondientes de la parte negativa.

4 Pruebas y resultados

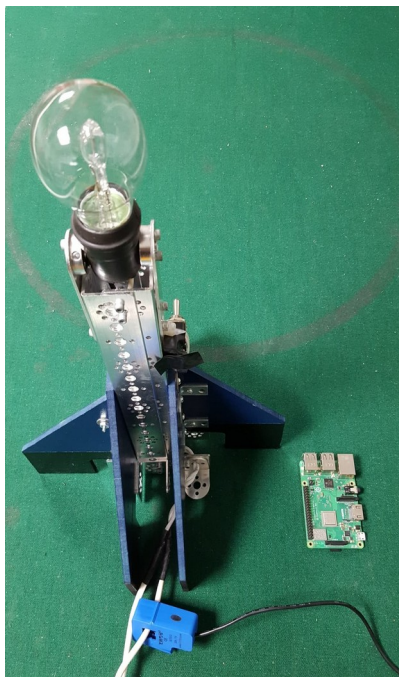


Figura 4.1 Conexión del foco resistivo

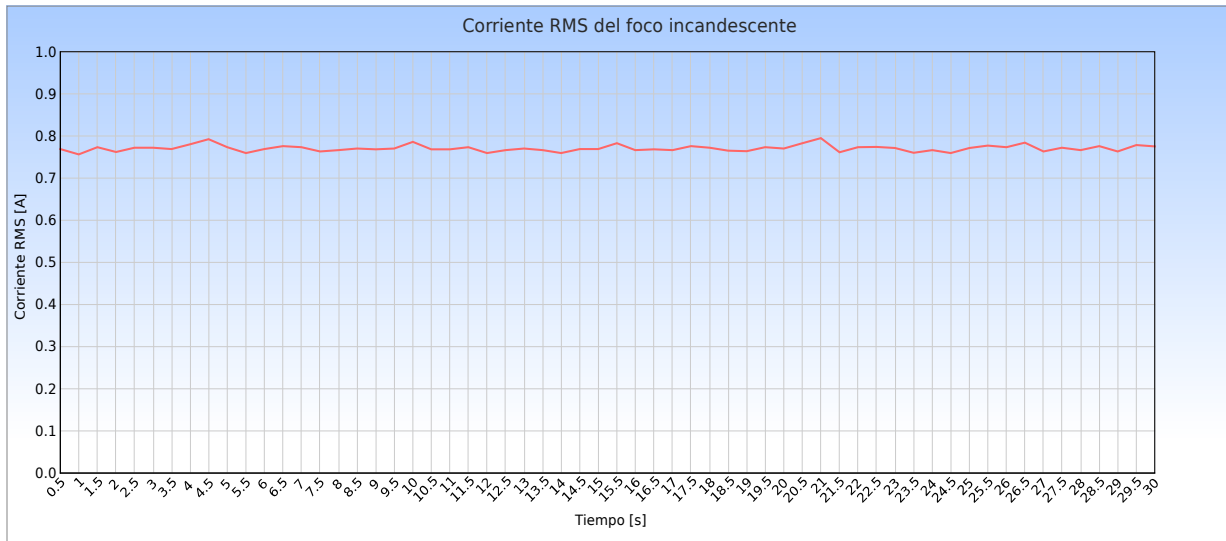


Figura 4.2 Gráfica resultante con el foco resistivo

4 Pruebas y resultados

Calculando la potencia consumida, se tuvo en cuenta que la duración del foco resistivo encendido durante una hora. Multiplicando el valor máximo y el mínimo valor de las corrientes previamente obtenidas con los 127V de voltaje nominal, podemos considerar una potencia que ronda desde los 96.012Wh hasta los 98.044Wh que son muy próximos o cercanos a los 100Wh del foco resistivo. También se pudo calcular la energía consumida teniendo en cuenta una hora que se utilizó el foco, lo cual nos da un total de 100Wh de energía consumida. Por lo cual, se pudo calcular la energía consumida con el sensor de corriente dando así un rango de valores que van desde los 96.012Wh hasta los 98.044Wh. Todos estos cálculos se realizaron por medio de a página web, como se muestra en la Figura 4.3.

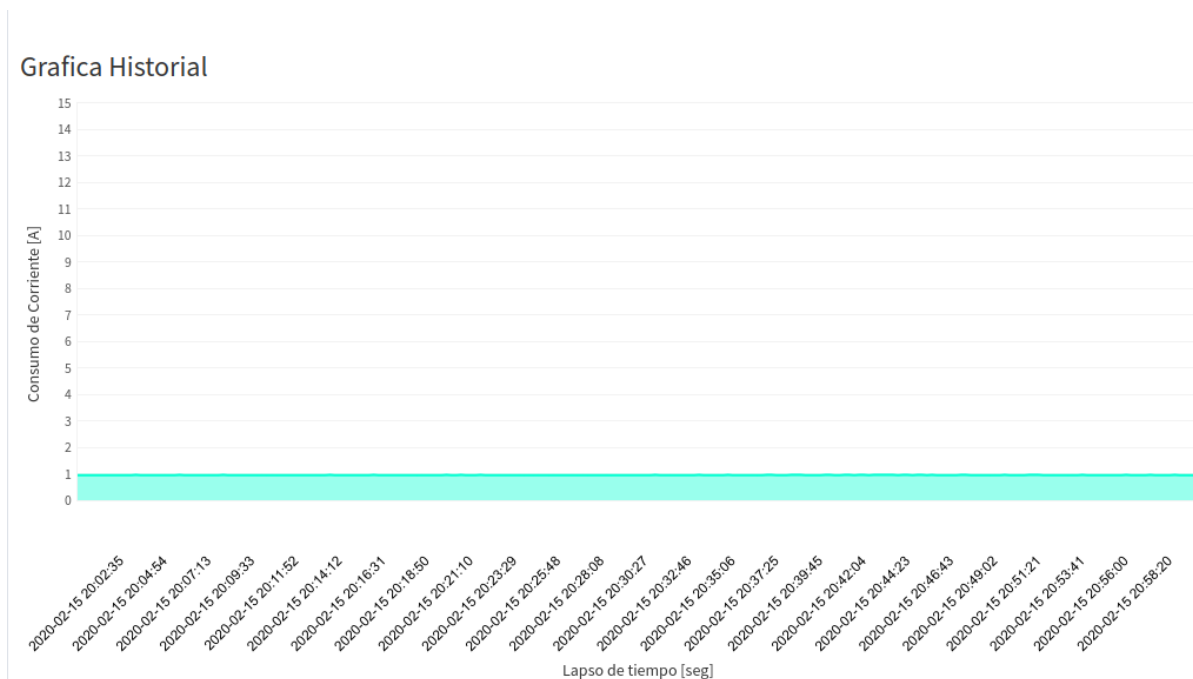


Figura 4.3 Consumo del foco resistivo visto desde el sistema

Basado en todos los resultados obtenidos, el uso del sensor tiene poco margen de error comparándolo con el consumo de corriente del foco incandescente dentro de un ambiente controlado lineal. Con base en el sistema puramente resistivo tuvo poco margen de error en los resultados, se prosiguió a realizar el mismo análisis pero con el foco fluorescente en la base.

4 Pruebas y resultados

4.1.2 Análisis no lineal con el foco fluorescente

El análisis no lineal con el foco fluorescente se realizó con la misma base para foco utilizada anteriormente como se muestra en la Figura 4.4. Siguiendo la misma conexión de la Figura 3.3 se procedió a obtener los valores provenientes de la corriente del foco fluorescente.

A diferencia del foco resistivo, el foco fluorescente genera en sus señales distorsiones armónicas. Un armónico se presenta cuando la forma de la onda de voltaje o de la corriente no es completamente senoidal. Las armónicas se presentan en varios aparatos que contienen circuitos eléctricos, como los focos fluorescentes que contienen un circuito que hace reaccionar el gas dentro del foco.

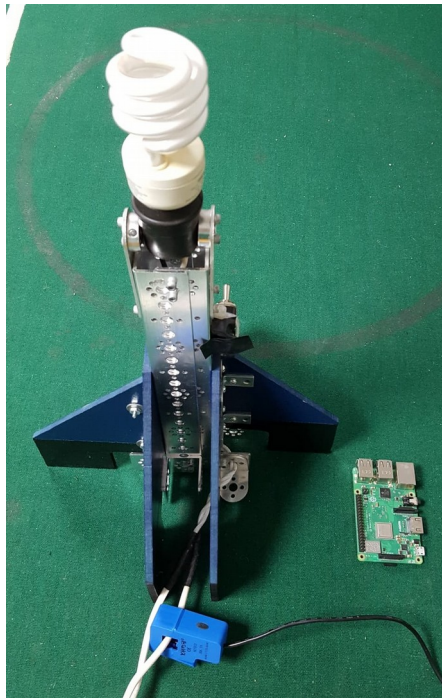


Figura 4.4 Conexión del foco fluorescente

En la Figura 4.5 se observan las diferencias de las señales que genera un foco resistivo y uno fluorescente. Podemos observar que la señal eléctrica de carga lineal de un foco resistivo genera una onda senoidal a una sola frecuencia y por lo tanto la señal no se distorsiona. Mientras tanto, un foco fluorescente que es una carga no lineal, genera una señal eléctrica modificada al sumarse otras frecuencias que provienen de la circuitería interna del foco. En el ejemplo de la gráfica se observa la señal senoidal alterada dando como resultado las magnitudes de los cuatro primeros armónicos.

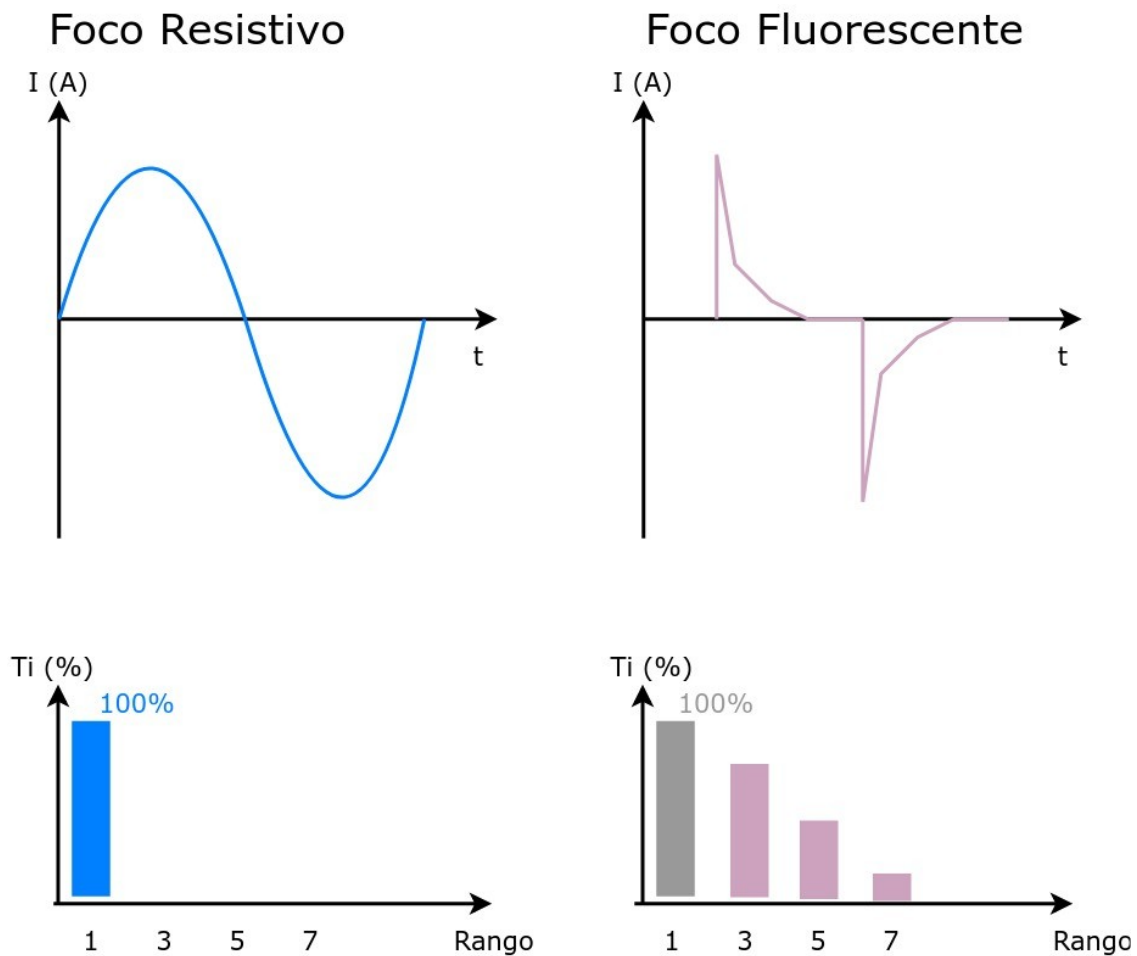


Figura 4.5 Diferencia entre señales con y sin armónicas

4 Pruebas y resultados

Dicho lo anterior, la gráfica que se obtuvo a partir del sensor y el Arduino con el foco fluorescente se muestra en la Figura 4.6. Se puede observar que las armónicas que obtuvo el sensor se asemejan en gran parte a las que aparecen en la Figura 4.5 anteriormente vista. Esto indica que el uso del sensor y el Arduino pudo recrear aproximadamente una señal de carga no lineal. Estos armónicos que se presentan en el foco fluorescente forman parte de los factores externos que hacen que el resultado de la corriente RMS varíe.

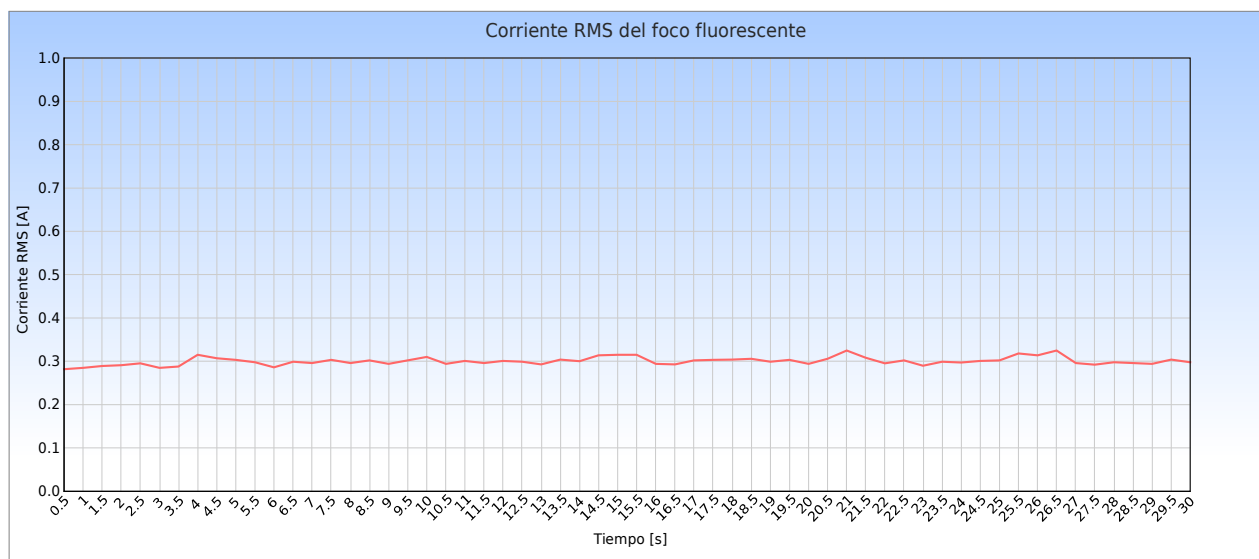


Figura 4.6 Gráfica resultante con el foco fluorescente con armónicos

Basado en lo anterior, los valores de la corriente que registró el Arduino junto con el sensor conectado al cableado del foco rondaron entre los 0.34A y los 0.289A. Comparando con el valor teórico del foco fluorescente, tenemos una diferencia que va desde los 4.97% hasta el 8.97% de error. Al igual que el foco resistivo, se colocó el foco fluorescente durante una hora para obtener la potencia y la energía consumida.

El foco tiene una potencia teórica de 40Wh, mientras que los valores de potencia obtenidos por el gancho rondan desde los 36.703Wh hasta los 38.1Wh, muy próximos al valor teórico. De la misma manera, el consumo eléctrico que consumió el foco fluorescente obteniéndolo con su valor teórico resultó en 40Wh consumidos en una hora, así como su contra parte del gancho se obtuvo un consumo eléctrico de 36.703Wh hasta los 38.1Wh. Los cálculos se realizaron de igual manera con ayuda de la página web (Figura 4.7).

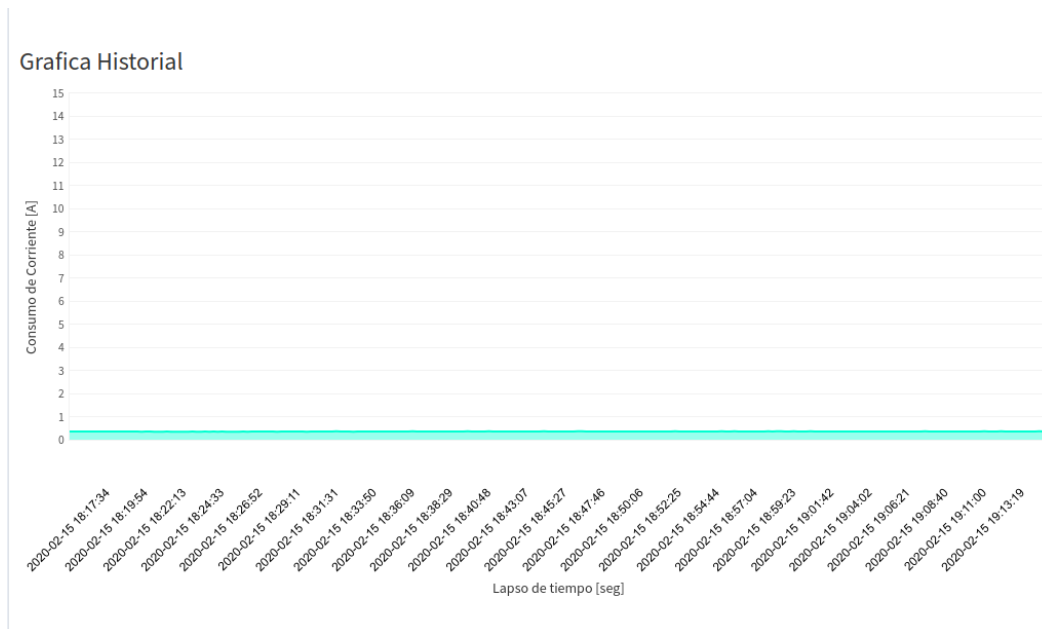


Figura 4.7 Consumo del foco incandescente visto desde el sistema

4.2 Comparación del sistema completo con un multímetro de gancho comercial

Para corroborar la funcionalidad de sistema de lectura, se tomó como puntos de comparación contra sistemas que ya existen en el mercado o que se colocan por parte de compañías de servicio eléctrico que funcionan de manera correcta. Las pruebas constituyen la comparación del sistema con un multímetro con gancho para medir la corriente en uno de los cables de las terminales eléctricas.

Para estas pruebas se utilizó el multímetro profesional de gancho RMS con auto rango de la marca steren (Steren, 2019) el cual cuenta con dos tipos de rangos: el primero de ellos tiene escala de valores entre los cero a los 20 A, mientras que el segundo rango que maneja es de valores entre los 0 a los 600A. Al ser un multímetro con gancho es de fácil uso para conectarlo a una terminal de la alimentación como se muestra en la Figura 4.9.

4 Pruebas y resultados

Este gancho se colocó en un cuarto dedicado a la costura dentro de un hogar. Se colocó el gancho para medir dicho cuarto que se dedica a la costura de prendas de mezclilla donde se utilizan maquinas de coser que según las especificaciones, cada motor consume hasta 4.9 A de consumo en trabajo constante; también se conectaron cargadores de celular a 1A de consumo además que de en ciertos periodos de conectó el cargador que recarga la batería del dispositivo (Figura 4.8).



Figura 4.8: Conexión del sistema al taller

Estas mediciones se pudieron llevar acabo gracias al uso de una extensión eléctrica la cual cuenta con las dos terminales separadas para que los ganchos se puedan colocar de manera sencilla ambos dispositivos y poder verificar que tan fiable es la lectura del sistema embebido comparándolo con un medidor de uso comercial. Las pruebas se llevaron acabo durante dos días donde se pusieron a funcionar ambos sistemas al mismo tiempo en un ambiente real para probar la precisión que tienen las lecturas entre sí.

4 Pruebas y resultados

Los días que se tomaron en cuenta fueron el 31 de agosto de 2019 y el 01 de septiembre de 2019 desde las 10:00 horas hasta las 20:00 horas dando como resultado los valores que se encuentran en la aplicación web del sistema (para el día 31 de Agosto de 2019 en la Figura 4.10 y para el día 01 de Septiembre de 2019 en la Figura 4.11) en donde se pueden observar los picos de consumo durante los periodos antes mencionados.



Figura 4.9 Multímetro profesional de gancho RMS con auto rango marca Steren

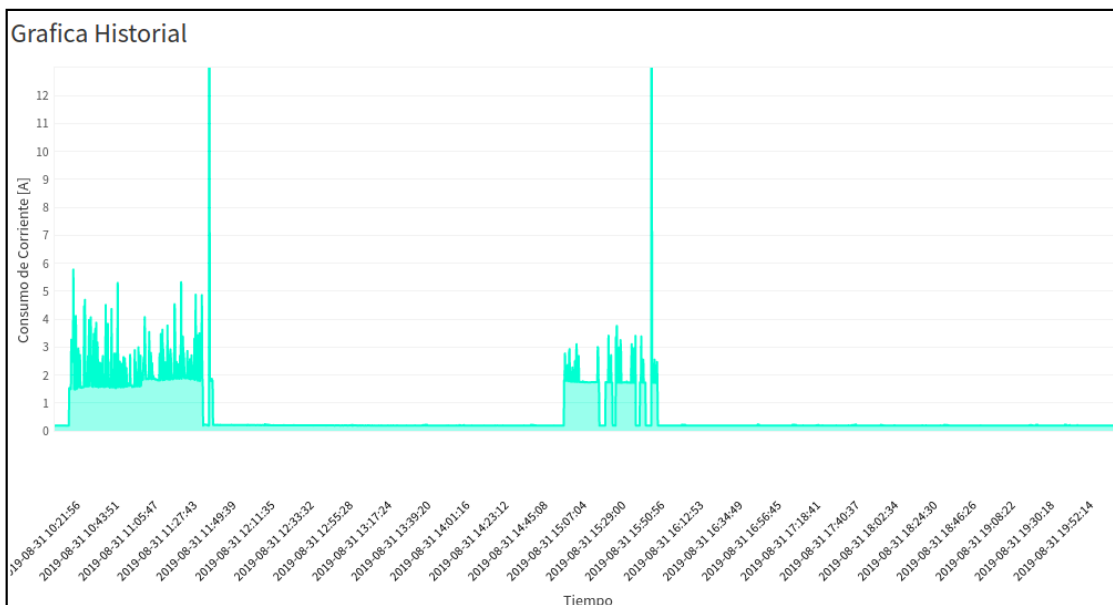


Figura 4.10 Gráfica histórica del 31 de Agosto de 2019

4 Pruebas y resultados

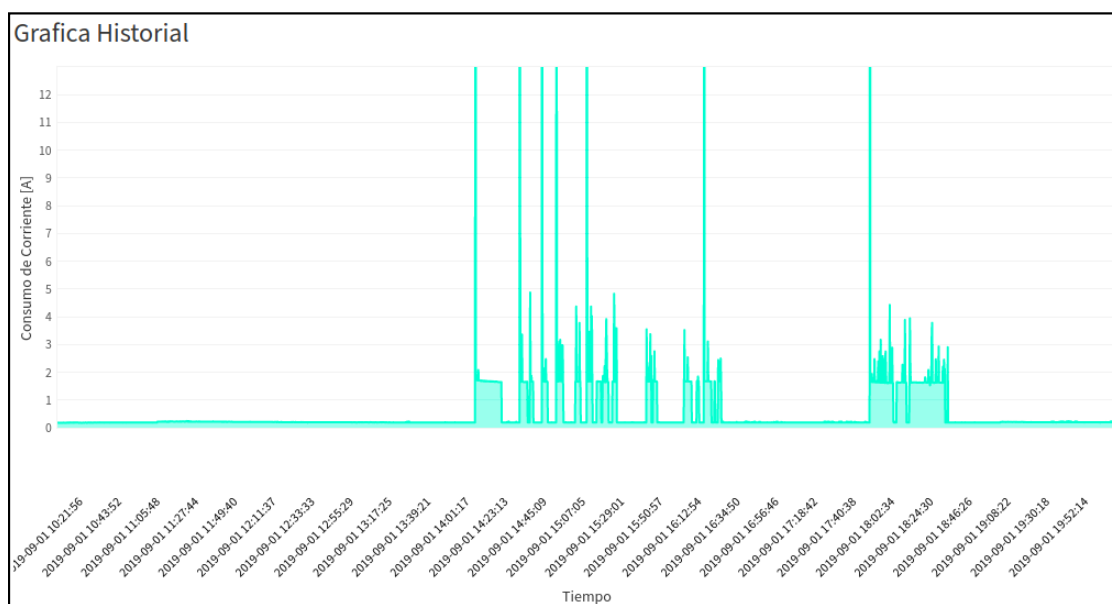


Figura 4.11 Gráfica histórica del 01 de Septiembre de 2019

Como resultado de las mediciones entre los dos dispositivos colocados en el cuarto de costura, se pudo apreciar que entre las mediciones del sistema embebido y el gancho comercial de la marca Steren existen variaciones en el cálculo del error absoluto que van aproximadamente entre los 0.01A a los 0.2A como se aprecia en la Tabla 4.1.

La diferencia entre las dos mediciones se puede apreciar mejor en las gráficas de la Figura 4.12 y la Figura 4.13, pues las dos se asemejan una con la otra, teniendo en cuenta que una de ellas es la referencia con la que se compara el sistema. Esto hace que el sistema tenga lecturas muy parecidas a las que realiza un gancho comercial a pesar de tener un margen de error.

4 Pruebas y resultados

Día	Hora [hrs]	Medición Corriente del gancho comercial [A]	Medición Corriente del sistema [A]	Error Absoluto [A]	Día	Hora [hrs]	Medición Corriente del gancho comercial [A]	Medición Corriente del sistema [A]	Error Absoluto [A]
31 de Agosto de 2019	10:00	0.017	0.186	0.17	01 de Septiembre de 2019	10:00	0.083	0.183	0.1
	11:00	3.131	3.231	0.1		11:00	0.095	0.222	0.13
	12:00	0.106	0.201	0.1		12:00	0.052	0.207	0.16
	13:00	0.027	0.193	0.17		13:00	0.033	0.194	0.16
	14:00	0.023	0.183	0.16		14:00	1.755	1.696	0.06
	15:00	1.780	1.742	0.04		15:00	0.020	0.189	0.17
	16:00	0.018	0.188	0.17		16:00	0.017	0.191	0.17
	17:00	0.017	0.191	0.17		17:00	0.017	0.187	0.17
	18:00	0.017	0.188	0.17		18:00	1.677	1.635	0.04
	19:00	0.016	0.19	0.17		19:00	0.051	0.207	0.16
	20:00	0.016	0.19	0.17		20:00	0.031	0.195	0.16

Tabla 4.1 Tabla comparativa en el cuarto de costura

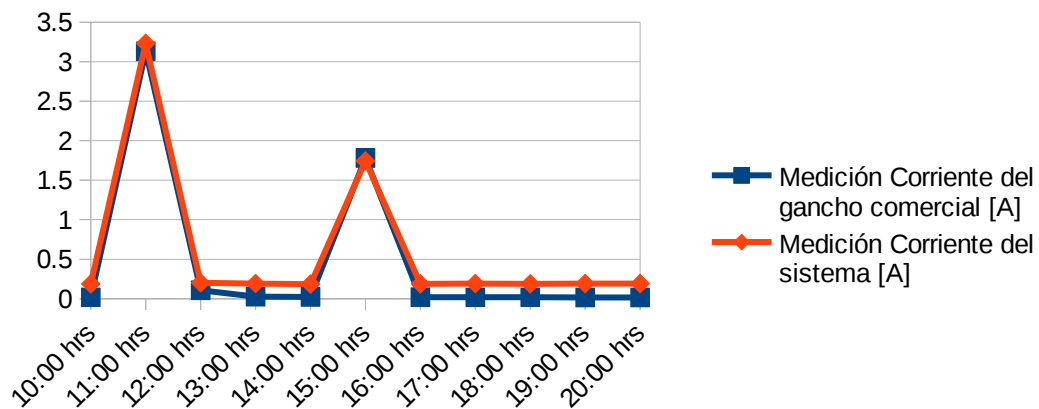


Figura 4.12 Gráfica comparativa del 31 de Agosto de 2019

4 Pruebas y resultados

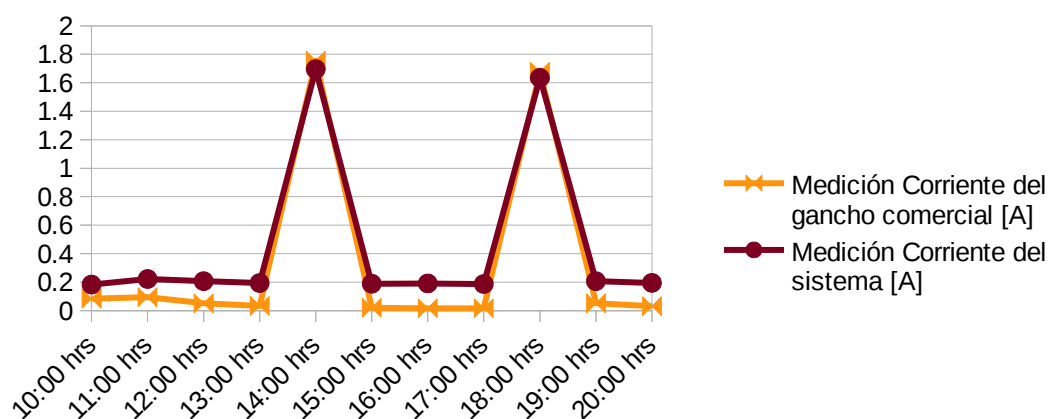


Figura 4.13 Gráfica comparativa del 01 de Septiembre de 2019

Una vez viendo los resultados de la Tabla 4.1 y de las gráficas de las Figura 4.12 y Figura 4.13, se puede apreciar que los valores obtenidos por el sistema son muy diferentes teniendo en cuenta que el error que marca en la tabla es de 0.17A aproximadamente. También se debe tener en cuenta que las pruebas fueron realizadas para un consumo pequeño donde había un motor de máquina de coser y cargadores de celular.

4.3 Pruebas con el sistema en ambiente real

Una vez realizadas las pruebas con sistemas que se encuentran en el mercado, se prosiguió a realizar las pruebas de medición dentro de ambientes en donde se necesitan monitorear el consumo eléctrico. La propuesta del ambiente en el cual se realizaron las pruebas que el sistema consultó su consumo eléctrico fue dentro un hogar, con el cual se desea conocer el consumo que aproximadamente tendrá en un cierto tiempo.

El hogar considerado para las pruebas consiste de una planta baja y un primer piso que se utilizó previamente en las pruebas anteriores. El hogar se considera que las habitaciones tanto de la planta baja como del primer piso. Cada una de las habitaciones cuentan con diferentes aparatos eléctricos, en su mayoría electrodomésticos, que funcionan en cualquier momento en el día, creando así un gasto eléctrico que es detectado por medio del medidor colocado por CFE.

Dentro del hogar se encuentran diferentes aparatos eléctricos tanto domésticos, de usos en específico y de uso laboral:

- Entre los domésticos ellos se encuentran un refrigerador, un horno de microondas, una lavadora-secadora, una licuadora, varios televisores y radios. Estos aparatos eléctricos son utilizados en diferentes horas del día.
- Para los de uso laboral se tienen varias máquinas máquinas de coser y máquinas cortadoras de tela que se encuentran en un taller de costura textil dentro del hogar. Estas máquinas tienen un horario diario de operación desde las 8:00 hrs hasta las 19:00 hrs.
- Para los de uso en general se tienen diferentes componentes eléctricos que se dedican a varias tareas del hogar como lo son una bomba de agua, los focos fluorescentes y/o lamparas de neón; también se tienen otros componentes eléctricos que son cargadores para celular, para laptop y para bocinas portátiles.

Esta instalación se encuentra cableada como indica la Figura 4.14: El cableado viene desde los postes de luz llegan la fase y el neutro a la terminal del medidor de luz hacia la casa, de ahí se continua el cableado hacia un apagador de pastilla, el cual ayuda en que caso de que se genere un corto la pastilla se bota, de ahí el cableado continua hacia dos cajas de fusibles que van hacia la planta baja y el primer piso de la casa respectivamente.

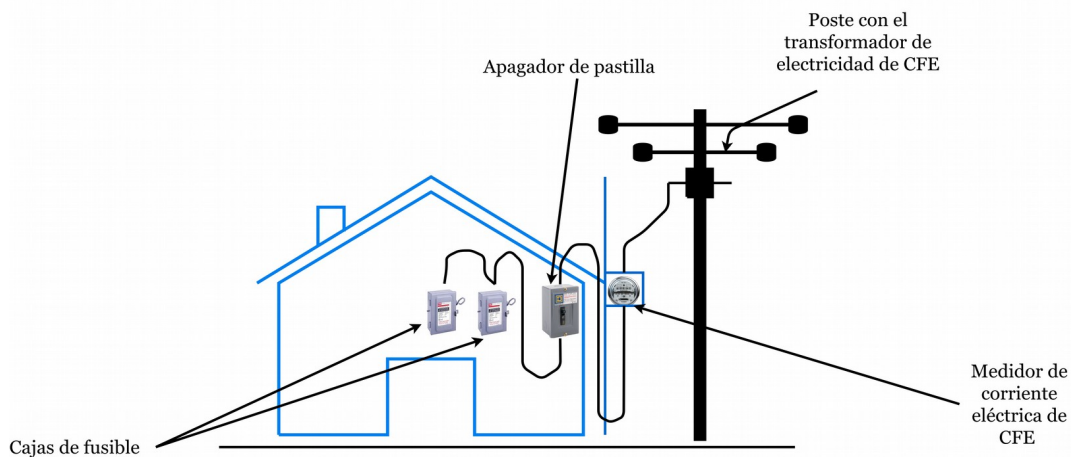


Figura 4.14 Cableado eléctrico de la casa

4 Pruebas y resultados

Para obtener la lectura en kWh por intervalos de tiempo del medidor digital de la compañía CFE, se realizó por medio de una sencilla resta como lo muestra la Fórmula 4.2. El valor del medidor que deseamos consultar es resultado de la diferencia entre la lectura actual con la anterior. Los valores fueron obtenidos de la siguiente forma:

- Teniendo en cuenta un periodo de poco más de una semana (9 días), se consideró la medición de cada día que el sensor de corriente estuvo colocado. Cada valor que fue tomado desde medidor de corriente colocado por CFE con un horario donde las actividades diarias que involucran el uso de los electrodomésticos comienzan desde las 10:00 hrs. Para este caso se consideran los intervalos de 24hrs para corroborar su funcionalidad de la medición de kWh.

$$\text{MediciónCFE} = \text{LecturaAnterior} - \text{LecturaActual}$$

Fórmula 4.2 Valor del consumo eléctrico con el medidor digital

Otro aspecto que se tuvo en cuenta es que se tiene un solo medidor encargado de las mediciones de toda la casa, además de que trabaja solo en kWh, por lo cuál las mediciones que realiza el sistema se calculan de manera interna y mostrando el consumo eléctrico en kWh.

Y una vez identificado la instalación eléctrica de la casa, se procedió a colocar el medidor de corriente en un punto de la terminal principal que pasa por caja de fusibles de toda la casa, la cual tiene una bifurcación en su cableado eléctrico dentro de la instalación. Lo cual ayudó para facilitar la colocación del gancho que tiene el medidor de corriente.

Los valores obtenidos del medidor digital del CFE se encuentran en la tabla, donde se muestran lo que se consumió durante los 9 días, y a partir de esos valores obtener el consumo con la Fórmula 4.2. Los valores registrados que se encuentran en la Tabla 4.2 son los valores de kWh que se obtuvieron por medio del dispositivo al conectarlo a dicha terminal del cableado. Se debe tener en cuenta que el medidor de CFE nos muestra los valores enteros, es decir, que no nos muestra si es un valor exacto con decimales y más bien esta redondeado. Por lo cual los valores del medidor que son redondeados, pueden generar problemas al tratar de compararlos con los datos que arroje el sensor de corriente.

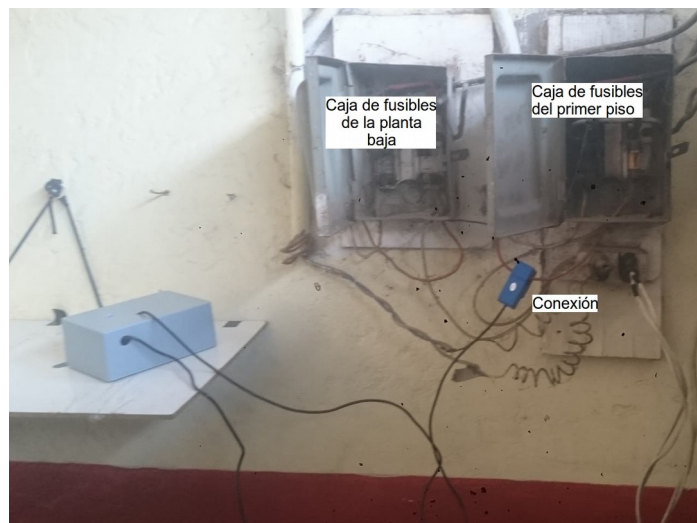


Figura 4.15 Instalación del dispositivo

Día	Consumo en el medidor CFE [kWh]
07 de Enero de 2020	4552 kWh
08 de Enero de 2020	4556 kWh
09 de Enero de 2020	4561 kWh
10 de Enero de 2020	4566 kWh
11 de Enero de 2020	4571 kWh
12 de Enero de 2020	4575 kWh
13 de Enero de 2020	4581 kWh
14 de Enero de 2020	4587 kWh
15 de Enero de 2020	4591 kWh
16 de Enero de 2020	4595 kWh

Tabla 4.2 Consumo eléctrico acumulado obtenido desde medidor CFE

4 Pruebas y resultados

Con ayuda de la aplicación web se visualizaron los datos que registró el sensor de corriente, y a su vez se pudieron obtener la cantidad de energía eléctrica consumida en cada día que transcurrió el sensor de corriente conectado y midiendo el consumo. De igual manera, se muestra un costo aproximado del consumo que se generó en cada día; estos datos se pueden visualizar en la Figura 4.16. La tarifa que brindó CFE están en un precio aproximado en pesos dependiendo de la energía eléctrica consumida (costo por kWh). Los costos utilizados corresponden al mes de enero del 2020 y son los siguientes: 8.31 centavos M.N. para el consumo básico, \$1.006 M.N. para el consumo intermedio y \$2.941 M.N. para el consumo excedente.

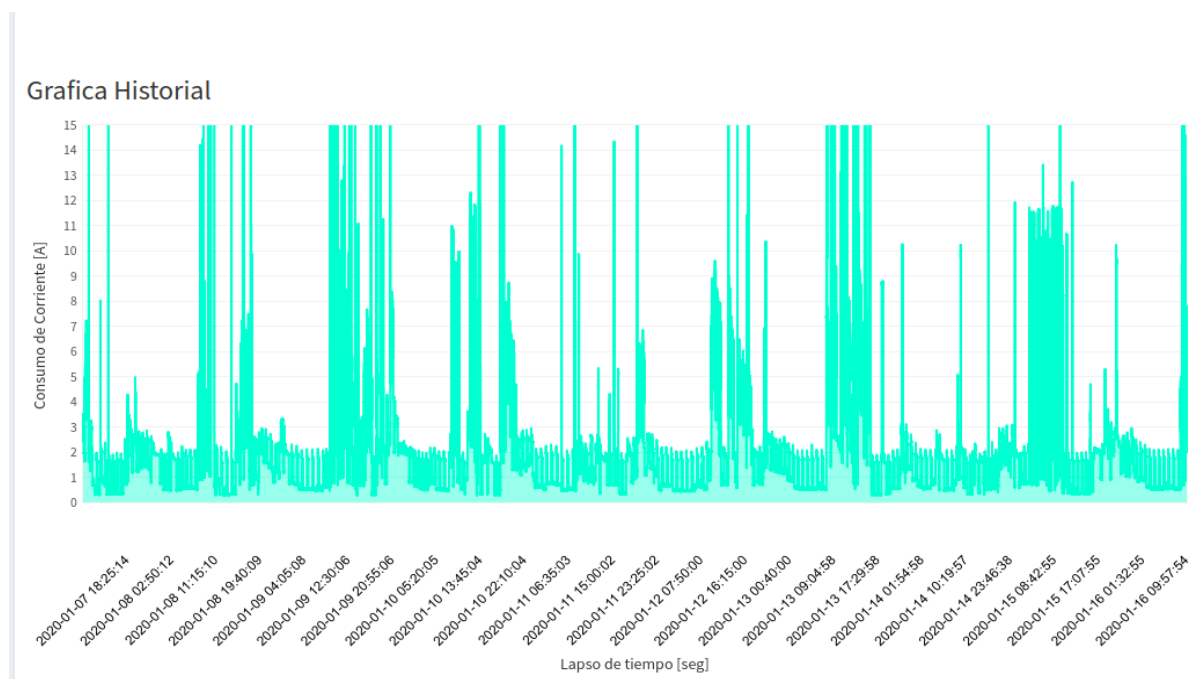


Figura 4.16 Periodo de lecturas del sensor de corriente

Día	Consumo con el sensor de corriente [kWh]	Costo aproximado
07 de Enero de 2020 - 08 de Enero de 2020	5.011 kWh	\$4.16 M.N.
08 de Enero de 2020 - 09 de Enero de 2020	6.603 kWh	\$5.49 M.N.
09 de Enero de 2020 - 10 de Enero de 2020	6.844 kWh	\$5.69 M.N.
10 de Enero de 2020 - 11 de Enero de 2020	5.808 kWh	\$4.83 M.N.
11 de Enero de 2020 - 12 de Enero de 2020	4.407 kWh	\$3.66 M.N.
12 de Enero de 2020 - 13 de Enero de 2020	7.270 kWh	\$6.04 M.N.
13 de Enero de 2020 - 14 de Enero de 2020	7.657 kWh	\$6.36 M.N.
14 de Enero de 2020 - 15 de Enero de 2020	4.979 kWh	\$4.14 M.N.
15 de Enero de 2020 - 16 de Enero de 2020	4.866 kWh	\$4.04 M.N.

Tabla 4.3 Lecturas del medidor de corriente obtenidas con el sistema desarrollado.

Al momento de comparar los valores obtenidos tanto del dispositivo, como de la lectura del medidor de CFE, se puede observar diferencias entre ambas lecturas. De tal manera que el sensor de corriente muestra lecturas que son próximas a un medidor digital que coloca la compañía CFE. En la Tabla 4.4 podemos observar los valores que se tienen tanto del medidor como del sistema, y que tan cercanos son entre sí. También se puede ver la semejanza entre ambas gráficas que tiene el medidor de corriente contra el los medidores de CFE en la Figura 4.17.

Se puede observar que hay errores dentro de las lecturas diarias entre $\sim 0.5\text{kWh}$ y $\sim 2\text{kWh}$. Estos valores obtenidos pueden variar debido a múltiples factores como lo son:

- La resolución del medidor de CFE no permite obtener lecturas precisas (con decimales), pues solo se puede ver diferencias enteras en kWh.
- Para los cálculos fue considerado como valor fijo del Voltaje 127V, pero este Voltaje puede variar dentro de un rango debido a factores externos.

4 Pruebas y resultados

Sin embargo, aunque pueden existir diferencias notables en la gráfica, se puede observar que existe una correlación muy acertada de los datos.

Día	Consumo en el medidor CFE [kWh]	Consumo con el sensor de corriente [kWh]
07 de Enero de 2020 - 08 de Enero de 2020	4 kWh	5.011 kWh
08 de Enero de 2020 - 09 de Enero de 2020	5 kWh	6.603 kWh
09 de Enero de 2020 - 10 de Enero de 2020	5 kWh	6.844 kWh
10 de Enero de 2020 - 11 de Enero de 2020	5 kWh	5.808 kWh
11 de Enero de 2020 - 12 de Enero de 2020	4 kWh	4.407 kWh
12 de Enero de 2020 - 13 de Enero de 2020	6 kWh	7.270 kWh
13 de Enero de 2020 - 14 de Enero de 2020	6 kWh	7.657 kWh
14 de Enero de 2020 - 15 de Enero de 2020	4 kWh	4.979 kWh
15 de Enero de 2020 - 16 de Enero de 2020	4 kWh	4.866 kWh

Tabla 4.4 Resultados de comparación entre los dos medidores.

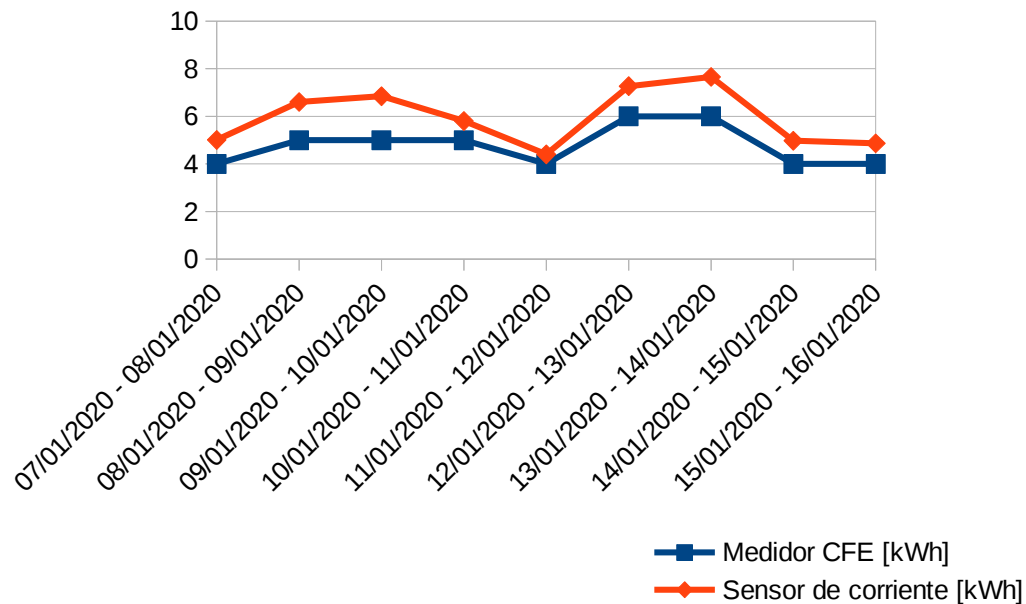


Figura 4.17: Gráfica comparativa entre los medidores

Basados en los resultados de la Tabla 4.4 y en la gráfica de la Figura 4.17, y pese a los factores previamente mencionados, se puede ver una mejoría en las lecturas del sistema con respecto a los valores que se obtuvieron en la sección 4.2 donde la carga era menor. Por lo cual, se puede comprobar que el sistema de medición con respecto a las cargas grandes de un hogar da mejores lecturas a comparación de las cargas mínimas.

Otro aspecto que también se considero es que el medidor de corriente hace las mediciones y las multiplica por el voltaje nominal para obtener el consumo eléctrico. Éste cálculo corresponde a potencia aparente, que en la mayoría de los casos es muy cercana a la potencia activa, si se conectan cargas inductivas, como es el caso de los motores de costura que se utilizaron durante las pruebas, el factor de potencia comienza a disminuir y esto hace que la potencia activa sea más pequeña que la potencia aparente. CFE realiza sus mediciones con la potencia activa consumida y el medidor de corriente implementado realiza mediciones con potencia aparente. Con lo cual, este factor puede explicar el por qué existe una diferencia entre los resultados obtenidos en la Tabla 4.4.

5 Conclusiones y trabajo futuro

En este capítulo se resumen los resultados que fueron obtenidos durante la realización del presente trabajo, además de mencionar algunas consideraciones que se presentaron en la implementación del sistema completo.

De igual manera, se plantean algunas propuestas que se encuentran fuera del alcance del presente trabajo para seguir mejorando el sistema para poder llevar una versión a un área de producción.

5.1 Conclusiones

Se logró integrar todos los componentes de IoT implementados en el sistema medición de corriente propuesto en este trabajo. Con la implementación del medidor de corriente capaz de almacenar los valores de corriente y enviarlos hacia el servidor, y con ayuda de la API para gestionar los datos guardados en la base de datos, se pudo consultar los datos de manera gráfica por medio de la página web, así como estimar un precio aproximado con respecto al consumo mensual obtenido. Como se pudo observar en el capítulo 4, las pruebas resultantes del sistema completo nos muestran resultados estimados cercanos al medidor de CFE.

Se comprobó el desarrollo de un dispositivo embebido puede medir la corriente eléctrica con la implementación en conjunto de un Arduino y una Raspberry para la detección, almacenamiento y envío de los datos hacia un servidor. La detección de la corriente eléctrica por medio de un sensor de gancho fue de gran ayuda para hacer más sencilla la conexión al cableado eléctrico sin tener que hacer una intervención en la instalación eléctrica, ni modificación alguna, ni siquiera se tuvo que desconectar algo, lo que hace que el dispositivo sea de fácil instalación.

Con respecto a la exactitud del sensor, se comparó con otros sensores comerciales para ver que tan parecidos los valores de corriente que obtienen tanto para cargas menores con pocos aparatos eléctricos, como para cargas mayores que se genera con los electrodomésticos del hogar. Además se consideró en las pruebas que existen las potencias activas que se obtienen del medidor de CFE que el medidor de corriente construido, ya que este último considera la potencias aparentes.

Los resultados de ambas mediciones del sistema nos muestran una diferencia con respecto a los medidores comerciales y/o impuestos por compañías de electricidad son diferentes entre sí, con algunas diferencias más notables en algunos casos. Por lo cual, se concluyó que el sistema funciona mejor en los casos donde existen cargas grandes como en un hogar o residencia a comparación de colocarlo en un punto donde la carga es menor por la poca cantidad de componentes y aparatos eléctricos.

A pesar de hacer simplificaciones de los cálculos y sin tomar mediciones de Voltaje, se corroboró la efectividad de la Raspberry Pi implementada al momento de almacenar y enviar los datos de corriente eléctrica que provienen del Arduino. La Raspberry Pi recolecta cada valor que proviene de su puerto serial, almacenando cada cierto tiempo un conjunto de datos para ser colocados en un archivo JSON. A parte, la Raspberry Pi envía cada archivo por medio de un punto de acceso a internet y elimina ese archivo hasta que reciba un código de entrega exitosa. En caso de presentarse un error de conexión con el servidor, la Raspberry Pi hace un reintento de enviar los archivos hasta que los envíe y en dado caso de que acumulen los archivos dentro de del dispositivo, los borra para no saturar la memoria.

Se verificó la implementación de un sistema que maneja el registro, almacenamiento y consulta de los datos que llegan a través de internet por medio de la API. La API gestiona los archivos JSON para almacenarlos en la base de datos de MongoDB que se encuentra en el servidor o solicite los datos para que sean mostrados en la página web. Las peticiones que recibe la API son atendidas en una estructura de cola siendo colocadas en orden en que fueron recibidas, y de esa manera no se sature de varias peticiones. Durante las pruebas, la API no presentó problemas al momento de peticiones que eran recibidas, en cambio lo que si se presentó fue una espera de varios segundo cuando se solicitaban lecturas con rangos de fechas de un mes. Hay que aclarar que este tipo de retardó se considera aceptable dentro de un sistema prototipo. Mientras que la inserción de los datos primero revisa que sean archivos con el formato JSON válido para no hacer que la API se detenga.

5 Conclusiones y trabajo futuro

Se comprobó que el desarrollo de la interfaz gráfica permite a un usuario visualizar y analizar la información resultante de las consultas en intervalos de tiempo. Con el uso de los cuadros de texto y botones implementados en la página, el usuario puede poner tanto el identificador de su dispositivo, los rangos de fechas que desee consultar, así como colocar las tarifas actuales que CFE le asigna a cada residencia. Al colocar dichos datos la página web retorna una gráfica del historial de corriente consumida aproximadamente, así como un costo estimado cercano al consumo eléctrico calculado.

Se verificó la eficiencia de los algoritmos implementados para obtener la información de los costos de consumo por medio de los valores de corriente almacenados. Los algoritmos de búsqueda y cálculo del costo energético eléctrico se implementaron dentro de la API, donde los parámetros que introduce el usuario marcan los límites de los datos a encontrar en la base de datos y con los datos previamente encontrados obtener el total de corriente para calcular la potencia eléctrica, además de calcular el consumo eléctrico. Gracias a las funciones de agregación que permite MongoDB, se pueden realizar consultas muy poderosas antes de hacer la presentación de los resultados, para que los cálculos sean más rápidos.

Los resultados que se aprecian en las pruebas son las gráficas que muestran la corriente eléctrica que se registró en el periodo de tiempo solicitado y por medio de un mensaje en la página web muestra el consumo eléctrico con su respectivo costo estimado.

Durante las pruebas se lograron buenos resultados bajo las consideraciones del presente trabajo. El sistema completo fue implementado en la residencia para corroborar su funcionamiento. Cabe aclarar que el sistema se encuentra funcionando actualmente con la posibilidad de obtener las lecturas y visualizar los datos recolectados hasta el momento.

Aunque se presentaron errores dentro de los consumos, se pudo observar claramente que existen momentos en los que hay consumos más altos o mas bajos, lo que permitiría a los usuarios concientizar acerca de las actividades que realizan los usuarios tienen un mayor impacto energético, lo que le permitiría o ayudaría a seguir estrategias de ahorro de energía, como los aconsejados por CFE.

5.2 Trabajo futuro

En el presente trabajo se realizó un sistema básico que consulta, analiza, y muestra el consumo eléctrico de un hogar utilizando el internet de las cosas, sin embargo dicho sistema al tener implementado componentes específicos hace que requiera contemplar otros aspectos para mejorar el sistema. Estas mejoras abarcan tanto la parte del hardware, así como el software para dar un mejor funcionamiento al sistema.

El trabajo futuro que concierne al área de hardware del sistema son algunas mejoras en cuanto a su construcción y algunos aditamentos con los cuales harían que el dispositivo tenga un mejor funcionamiento para la detección de los valores de corriente eléctrica y para obtener el cálculo del consumo eléctrico.

Los componentes electrónicos, como el amplificador operacional, que contiene el medidor de corriente tienen variaciones en la práctica, es decir, que los valores que tienen nominalmente pueden variar, lo cual hace que el resultado pueda no ser exacto. Por ende, se propone cambiar estos componentes electrónicos por unos con más precisión en su uso, que hace que el resultado sea más preciso para el dispositivo.

El medidor de corriente se conecta por defecto a una red inalámbrica para enviar los datos en archivos JSON al servidor. Esta red inalámbrica está predefinida dentro de la Raspberry Pi y por automático se conecta a internet. Para mejorar su usabilidad, se hará una aplicación móvil, en donde se pueda definir a que red inalámbrica se puede conectar el medidor de corriente. Esto para poder configurar el medidor de corriente y cambiar la red a cual conectarse de manera sencilla.

El hardware del medidor de corriente tiene considerado un gancho para realizar sus mediciones por medio del cableado eléctrico, pero no un sensor el cual mida el voltaje que se encuentra en sistema eléctrico. El hardware fue diseñado contemplando sensores no invasivos para que sea fácil de colocar por el usuario, y los sensores que miden voltaje son meramente invasivos. El voltaje se mide por medio del flujo de electricidad, haciendo que el sensor se conecte en una parte del cableado. La idea que se tiene para un trabajo futuro es tener un sensor o un dispositivo que pueda medir el voltaje y que se pueda adaptar al cableado eléctrico de manera sencilla.

La parte exterior del hardware es otra mejora que se considera a trabajo futuro. La cubierta que contiene actualmente consiste en una caja sencilla de plástico con un par de orificios donde pasan los cables USB, lo cual expone más el circuito con el ambiente. Por lo cual se propone realizar, dentro del trabajo futuro, una cubierta exterior que es capaz de proteger al circuito de los cambios de temperatura y los ambientes húmedos.

En cambio, la parte del software se pueden mejorar aspectos de seguridad y de organización en la página web. Para acceder a la página web sólo se necesita la dirección web para consultar el consumo eléctrico. Por lo cual se desea elaborar un sistema de autenticación de usuarios por medio de un número de usuario y contraseña. Cada usuario puede tener varios dispositivos asociados en su cuenta y a su vez el número de usuario estará asociado a los identificadores de cada dispositivo. Así el usuario puede entrar o salir de la página web sin ningún problema.

5 Conclusiones y trabajo futuro

Otro aspecto que se quiere mejorar dentro de la página web es hacer más fácil la consulta del precio estimado del consumo eléctrico. El sistema requiere de introducir los precios de manera manual. Entonces se buscaría que la misma página web consulte los precios desde la información de la página web de CFE dependiendo al mes correspondiente. Esto también ayudaría a actualizar los precios en dado caso que CFE tenga que hacer ajustes.

Del mismo modo, el sistema no contempla recomendaciones hacia al usuario en cuestión de la cantidad de energía que consumió en el mes, es decir, que el sistema pueda detectar en los horarios que se consume excesivamente la energía eléctrica para poder tomar otras medidas preventivas ahorrrativas. Se tiene como mejora futura la implementación de un algoritmo que pueda recabar los datos de meses anteriores para compararlos con los que se tienen medidos recientemente, y de haber cambios notorios que indiquen un exceso del consumo eléctrico, el usuario sea notificado a su computadora o dispositivo móvil a través de notificaciones push, sms o e-mails para que tome sus debidas precauciones.

Se espera desarrollar más dispositivos de medidor de corriente para ser implementados dentro del proyecto *“Laboratorio de movilidad e infraestructura verde para la eficiencia energética en la ciudades”*, el cual se tienen estaciones de carga para suministrar energía a las bicicletas eléctricas. El objetivo de esto es consultar el consumo de cada estación y verificar su rendimiento en uso para monitorear cada estación en caso de que se presente algún fallo.

6 Bibliografía

Adafruit (2019). *Adafruit Powerboost 1000C*. Recuperado de <https://learn.adafruit.com/adafruit-powerboost-1000c-load-share-usb-charge-boost>

Arduino (2019). *ARDUINO NANO*. Recuperado de <https://store.arduino.cc/usa/arduino-nano>

Avelino Pérez, Pedro (2001). *Transformadores de distribución: teoría, cálculo, construcción y pruebas*. Editorial Reverte.

T. Bokareva, N. Bulusu y S. Jha, «A Performance Comparison of Data Dissemination Protocols for Wireless Sensor Networks,» de Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE, 2004.

BYTEMARK HOSTING (2018). *Welcome to Raspbian*. <https://www.raspbian.org/>

Cartwright, Kenneth V (Fall 2007). *Determining the Effective or RMS Voltage of Various Waveforms without Calculus*(PDF). *Technology Interface*.(1): 20 pages.

Cayssials, Ricardo (2014). *Sistemas embebidos en fpga*. Editorial Alfaomega grupo editor. Número de páginas 340. ISBN 9789871609697.

CFE (2018) *Informe Anual 2018 [PDF file]*. Recuperado de https://infosen.senado.gob.mx/sgsp/gaceta/64/1/2019-05-15-1/assets/documentos/PoderEjecutivo_informe_anual_2018_cfe.pdf

CFE (2019) *CFE. Comisión Federal de Electricidad*. <https://app.cfe.mx/Aplicaciones/CCFE/Tarifas/TarifasCRECasa/Tarifas/Tarifa1.aspx>

Cochran, David (12 de noviembre de 2012). *Twitter Bootstrap Web Development* (1st edición). Packt Publishing. p. 100. ISBN 978-1849518826.

Departamento de Ingeniería Eléctrica, Electrónica y de Control de UNED (2019). Master degree: Industrial Systems Engineering, Asignatura ISE5: Controladores industriales de diseño de alto nivel. Módulo 3: sistemas embebidos. [PDF file]. Recuperado de

http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_ISE5_3_1.pdf.

Dept. Ciencia de la Computación e IA de la universidad de Alicante (2012-2013) . *Servicios Web y SOA [PDF file]*. Recuperado de <http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/wholesite.pdf> Pagina 4

Flask (2019). *Flask*. <https://palletsprojects.com/p/flask/>

Freiberger, Paul (1981-10-05). *"Some Confusion at the Heart of IBM Microcomputer / Which Operating System Will Prevail"*. InfoWorld. pp. 50–51. Retrieved 1 January 2015.

Friendly, Michael (24 de Agosto de 2009). *Milestones in the history of thematic cartography, statistical graphics, and data visualization*. Recuperado de <http://www.math.yorku.ca/SCS/Gallery/milestone/milestone.pdf>

Gevent (2019). *What is gevent?*. Recuperado de <http://www.gevent.org/>

Huircán, Juan Ignacio (2017). *Conversores Análogo-Digital y Digital-Análogo: Conceptos Básicos [PDF file]*. Recuperado de <https://core.ac.uk/download/pdf/47281302.pdf>

Huircán, Juan Ignacio (Diciembre 3, 2012). *Reguladores de Voltaje [PDF file]*. Recuperado de <https://www.acomee.com.mx/clasificaciones/REGULADORES%20DE%20VOLTAJE.pdf>

IBM support (2013) Serviciosweb RESTful. Recuperado de https://www.ibm.com/support/knowledgecenter/es/SSGMCP_5.1.0/com.ibm.cics.ts.mobileextensions.doc/concepts/concepts_restful.html

Ikeda, k. and Masuda, h. (2016). *Tecnología en Sensores de Corriente altamente estables, Alta Precisión y Banda Ancha*. Madrid: HIOKI. Available at: <http://idm-instrumentos.es/wp-content/uploads/2016/02/Tecnolog%C3%AD-a-en-Sensores-de-Corriente.pdf> [Accessed 13 Feb. 2019].

Jensen, Cary, Anderson, Loy (1992). *Harvard graphics 3: the complete reference*. Osborne McGraw-Hill ISBN 0-07-881749-8 p.413

Jin, Yanyan, Wang, Jinping, Xu, Jianping & Zhou, Gouhua (2009) *Comparison study on digital peak current, digital peak voltage, and digital peak voltage/peak current Controlled Buck Converter*. 2009 4th IEEE Conference on Industrial Electronics and Applications. IEEE. Xi'an, China.

JSON.org (2018). *Introducción a JSON*. Recuperado de <https://www.json.org/json-es.html>

Lathi, B.P. (1998). *Modern Digital and Analog Communication Systems* (3rd edition). Oxford University Press.

Méndez Cruz, L.I. Carlos Francisco. UNAM, Facultad de contaduría (2018). *Apuntes de*

6 Bibliografía

la materia de Bases de Datos [PDF file]. Recuperado de http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/6/bases_datos.pdf.

MongoDB (2018). ¿Qué es MongoDB?. Recuperado de <https://www.mongodb.com/es/what-is-mongodb>

MongoDB (2008-2019). PyMongo 3.7.2 Documentation. Recuperado de http://api.mongodb.com/python/current/index.html?_ga=2.179351524.1187306997.1547593242-2115167947.1534527717

Pallás Areny, Ramón(2004). *Sensores y acondicionadores de señal*. Editorial Marcombo. 494 Páginas

Pololu (2019). *Pololu 5V, 2.5A Step-Down Voltage Regulator D24V25F5*. Recuperado de <https://www.pololu.com/product/2850/specs>

Roseroa, Paul, Núñez, Santiago, Realpea, Stalin, Alveara, Vanessa, Beltrán, Luis y Rosado, Christian. Universidad Técnica del Norte. Instituto Tecnológico Superior 17 de Julio (Octubre 2017). INTERNET DE LAS COSAS Y REDES DE SENSORES INALÁMBRICOS: REVIEW [PDF File]. Recuperado de https://www.researchgate.net/publication/316438631_INTERNET_DE_LAS_COSAS_Y_REDES_DE_SENSORES_INALAMBRICOS_REVIEW

Sector Eléctrico Nacional (2016). *Datos técnicos de las principales centrales de CFE en operación en 2016*. Recuperado de <http://egob2.energia.gob.mx/portal/electricidad.html>

STMicroelectronics (2018) LM358. Recuperado de <https://www.st.com/en/amplifiers-and-comparators/lm358.html>

Simionato, Michele (June 11, 2007). *An Introduction to Web Programming with WSGI*. <http://www.phyast.pitt.edu/~micheles/python/europython07/talk.html>

Steren (2019). *MULTÍMETRO PROFESIONAL DE GANCHO RMS CON AUTO RANGO*. Recuperado de: https://www.steren.com.mx/multimetro-profesional-de-gancho-rms-con-auto-rango.html?gclid=CjwKCAjwkqPrBRA3EiwAKdtwk9EQ9WiinB_laoxVn3A0S9EHFAr0NSZzjPrSYBu9JHZg2Vag5rMzmRoC7I4QAvD_BwE

Torrente Arterro, Oscar. ARDUINO Curso practico de formación. Primera Edición. Alfaomega grupo editor. México, Febrero 2013.

Traukina, Alena; Thomas, Jayant; Tyagi, Prashant; Reddipalli, Kishore (29 September 2018). *Industrial Internet Application Development: Simplify IIoT development using the elasticity of Public Cloud and Native Cloud Services (1st ed.)*. Packt Publishing. p. 18.

T. Bokareva, N. Bulusu y S. Jha, *A Performance Comparison of Data Dissemination*

Protocols for Wireless Sensor Networks, de Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE, 2004.

Upton, Eben (14 March 2018). "*Raspberry Pi 3 Model B+ on Sale at \$35*". Raspberry Pi Blog. Raspberry Pi Foundation. Retrieved 2018-05-04.

Vermesan, Ovidiu & Backquest Joël (2017). Cognitive Hyperconnected Digital Transformation: Internet of Things Intelligence Evolution. EU, Belgium: River Publishess.

W3C Working Group (February 2004). *Web Services Architecture*. Retomado de <https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis>

YHDC (2019-8-7). *Product Specification STC013-30*. Retomado de <http://statics3.seeedstudio.com/assets/file/bazaar/product/101990028-SCT-013-030-Datasheet.pdf>